NUMERICAL FIELD MODEL SIMULATION
OF
FIRE AND HEAT TRANSFER
IN A RECTANGULAR COMPARTMENT

by

Kenneth J. Thorkildsen

September 1992

Thesis Advisor  M.D. Kelleher

Approved for public release; distribution is unlimited.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.
Shipboard fires have been the bane of mariners since man's earliest attempts to sail the sea. Understanding the behavior of fire in an enclosed space, such as those found on today's modern seagoing vessels, will greatly enhance the mariner's ability to combat or prevent them. In a joint effort between the Naval Postgraduate School and the University of Notre Dame a computer code has been developed to model a full scale fire in a closed compartment. The code uses a finite volume formulation to obtain numerical solutions to the unsteady, three dimensional conservation equations of mass, momentum and energy. Included are the effects of turbulence, strong buoyancy, surface radiation and wall conduction. The code gives velocities, pressure, temperatures, and densities throughout the field.

This thesis applies that computer code to the U.S. Navy's full scale fire test chamber at Naval Air Warfare Center, China Lake, California. Advanced computer graphics techniques, including color contouring and three dimensional vector field plotting have been applied to make output data more informative. It is hoped that someday this model could provide a useful tool for naval architects in the design of a fire safe ship, and a cost effective means for development evaluation of new firefighting equipment and techniques.
Approved for public release; distribution is unlimited.

Numerical Field Model Simulation
of
Fire and Heat Transfer
in a Rectangular Compartment

by

Kenneth J. Thorkildsen
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1992

Author: 

Kenneth J. Thorkildsen

Approved by: 

M.D. Kelleher, Thesis Advisor

M.D. Kelleher, Chairman,
Department of Mechanical Engineering
ABSTRACT

Shipboard fires have been the bane of mariners since man’s earliest attempts to sail the sea. Understanding the behavior of fire in an enclosed space, such as those found on today’s modern seagoing vessels, will greatly enhance the mariner’s ability to combat or prevent them. In a joint effort between the Naval Postgraduate School and the University of Notre Dame a computer code has been developed to model a full scale fire in a closed compartment. The code uses a finite volume formulation to obtain numerical solutions to the unsteady, three dimensional conservation equations of mass, momentum and energy. Included are the effects of turbulence, strong buoyancy, surface radiation and wall conduction. The code gives velocities, pressure, temperatures, and densities throughout the field.

This thesis applies that computer code to the U.S. Navy’s full scale fire test chamber at Naval Air Warfare Center, China Lake, California. Advanced computer graphics techniques, including color contouring and three dimensional vector field plotting have been applied to make output data more informative. It is hoped that someday this model could provide a useful tool for naval architects in the design of a fire safe ship, and a cost effective means for development evaluation of new firefighting equipment and techniques.
THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.
# TABLE OF CONTENTS

I. INTRODUCTION ................................................................. 1
   A. BACKGROUND ................................................................. 1
   B. COMPUTER MODELING ...................................................... 2
   C. THE FIRE TEST FACILITY ................................................. 3
   D. THE COMPUTER PROGRAM .................................................. 8

II. THE NUMERICAL MODEL ..................................................... 10
   A. THE GOVERNING EQUATIONS .............................................. 10
   B. THE CONTROL VOLUME ...................................................... 11
   C. DISCRETIZATION OF THE CONSERVATION EQUATIONS ............... 13
       1. The Continuity Equation .............................................. 13
       2. The Energy Equation .................................................. 14
       3. The Momentum Equations ............................................. 15
   D. PRESSURE CORRECTIONS ................................................... 18
       1. Global Pressure Corrections .......................................... 18
       2. Local Pressure Corrections ......................................... 19
   E. INITIAL AND BOUNDARY CONDITIONS ................................... 21
       1. Initial Conditions ..................................................... 21
       2. Boundary Conditions ................................................ 21
   F. MODELING OF PHYSICAL PHENOMENA ................................... 22
       1. Wall Conduction ....................................................... 22
       2. Turbulence .......................................................... 22
       3. Radiation ........................................................ 23

III. THE COMPUTER PROGRAMS ................................................ 24
   A. INTRODUCTION ............................................................ 24
   B. PROGRAM FIREBLD ....................................................... 24
   C. PROGRAM FIRE .......................................................... 26
   D. GRAPHICAL ANALYSIS .................................................. 27

IV. CONCLUSIONS AND RECOMMENDATIONS .................................. 49
A. CONCLUSIONS ......................................................... 49
B. RECOMMENDATIONS .................................................... 49

APPENDIX A. PROGRAM FIREBLD ........................................ 50
APPENDIX B. PROGRAM FIRE .............................................. 61
APPENDIX C. PROGRAM ISOTHERM ....................................... 150
APPENDIX D. PROGRAM VELOCITY ....................................... 165

LIST OF REFERENCES ......................................................... 179
INITIAL DISTRIBUTION LIST ............................................... 183
LIST OF TABLES

Table 1. INPUT DATA ............................................. 25
Table 2. WALL DATA ............................................. 26
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fire test chamber at NAWC, China Lake, CA.</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Plan view of fire test chamber at NAWC, China Lake, CA.</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>North south elevations of test chamber at NAWC, China Lake, CA.</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>East west elevations of test chamber at NAWC, China Lake, CA.</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Instrumentation of fire test chamber at NAWC, China Lake, CA.</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>One Dimensional Basic and Staggered Computational Cells.</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>Basic Cell Nomenclature on a Rectangular Grid.</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>Temperature Profile, Plan View, 30 seconds.</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>Temperature Profile, Plan View, 60 seconds.</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>Temperature Profile, Plan View, 90 seconds.</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>Temperature Profile, X-Z Profile, 30 seconds.</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>Temperature Profile, X-Z Profile, 60 seconds.</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>Temperature Profile, X-Z Profile, 90 seconds.</td>
<td>34</td>
</tr>
<tr>
<td>14</td>
<td>Temperature Profile, Y-Z Profile, 30 seconds.</td>
<td>35</td>
</tr>
<tr>
<td>15</td>
<td>Temperature Profile, Y-Z Profile, 60 seconds.</td>
<td>36</td>
</tr>
<tr>
<td>16</td>
<td>Temperature Profile, Y-Z Profile, 90 seconds.</td>
<td>37</td>
</tr>
<tr>
<td>17</td>
<td>Velocity Profile, Plan View, 30 seconds.</td>
<td>38</td>
</tr>
<tr>
<td>18</td>
<td>Velocity Profile, Plan View, 60 seconds.</td>
<td>39</td>
</tr>
<tr>
<td>19</td>
<td>Velocity Profile, Plan View, 90 seconds.</td>
<td>40</td>
</tr>
<tr>
<td>20</td>
<td>Velocity Profile, X-Z Profile, 30 seconds.</td>
<td>41</td>
</tr>
<tr>
<td>21</td>
<td>Velocity Profile, X-Z Profile, 60 seconds.</td>
<td>42</td>
</tr>
<tr>
<td>22</td>
<td>Velocity Profile, X-Z Profile, 90 seconds.</td>
<td>43</td>
</tr>
<tr>
<td>23</td>
<td>Velocity Profile, Y-Z Profile, 30 seconds.</td>
<td>44</td>
</tr>
<tr>
<td>24</td>
<td>Velocity Profile, Y-Z Profile, 60 seconds.</td>
<td>45</td>
</tr>
<tr>
<td>25</td>
<td>Velocity Profile, Y-Z Profile, 90 seconds.</td>
<td>46</td>
</tr>
<tr>
<td>26</td>
<td>Example of “Zoom” feature of ISOTHERM.</td>
<td>47</td>
</tr>
<tr>
<td>27</td>
<td>Example of “Zoom” feature of VELOCITY.</td>
<td>48</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

The author would like to take this opportunity to thank the following persons for their contributions to this work:

To Professor K.T. Yang of the University of Notre Dame, for providing the program FIRE which is central to this work and for making himself available whenever there were questions regarding the code.

To Mr. Kent Farmer of the Naval Air Warfare Center, China Lake, CA for providing the data regarding the fire test chamber.

To Dr. Devadatta Mukutmoni of the Naval Postgraduate School whose assistance with development of the plotting programs and troubleshooting all of the programs was instrumental in completion of this work.
I. INTRODUCTION

A. BACKGROUND

Fire aboard ship is one of the professional mariner's worst fears. With no place to go to escape the heat and smoke the mariner must fight the blaze or face the loss of his vessel. Even though U.S. merchant marine and naval personnel receive some of the best fire fighting training available, annual losses to shipboard fires can easily run into the hundred million dollar range. Ship down time, equipment repair replacement, personnel injuries and casualties all contribute to these costs and result in the degradation of our merchant marine and naval forces. To minimize these losses it is imperative that the phenomena of fire be studied and understood especially in the relatively closed environment found aboard ship. It is only through study and understanding that adequate means may be developed to prevent or mitigate the devastating effects of a shipboard fire.

Fire is a complex phenomena whose study is equally complex, requiring the combined knowledge of a variety of fields including fluid dynamics, heat and mass transfer, and combustion. Research into the mechanics of fire and development of methods to predict its behavior will aid engineers in reducing the probability of its ignition and minimizing its effects once ignited. [Ref. 1]

It is the complexity of fire which makes its study so difficult, especially in a shipboard environment. Small scale fire studies have proven inadequate for predicting the behavior of large scale fires. Therefore, full scale studies are generally the only means for conducting realistic experimentation. However, such full scale experimentation can be very costly and dangerous. Shipboard fires often occur in fully enclosed, airtight spaces where pressures can build up during the fire. These spaces may have restricted accesses, contain electronic equipment, flammables and/or toxics, all of which add to the expense and danger of full scale experimentation. Additional costs will also be incurred as air quality and emission standards are stiffened across the country.

To study the phenomena of shipboard fire, the U.S. Navy has built a test facility at the Naval Research Laboratory in Washington, DC known as Fire-I. This facility is essentially a large cylindrical pressure vessel with spherical endcaps intended for use in full scale modeling of fires inside submarines and/or in closed airtight compartments aboard ships. Missile attacks against British warships during the Falklands crisis and the
Iraqi missile attack on the USS STARK in the Persian Gulf in 1987 prompted the U.S. Navy to build a second fire test facility at the Naval Air Warfare Center (NAWC) in China Lake, CA. The purpose of this facility is to study the effects of fire in a vented compartment [Ref. 2, pp. 7-8]. It is this facility which is modeled in this thesis.

A less expensive, and less dangerous alternative to full scale fire experimentation is the use of computer modeling techniques. The development of high speed, high capacity computers has allowed researchers to model thoroughly the complex fire phenomena and predict fire behavior without the expense of full scale testing. A properly developed computer model, validated against actual full scale test data, can provide a less expensive and safe alternative to full scale experimentation. Furthermore, the inherent flexibility of computer modeling allows it to be used on increasingly more complex geometries. Someday this may lead to modeling of entire ships during the design phase to identify areas particularly susceptible to fire, or for the accurate prediction of the effectiveness of new firefighting techniques.

B. COMPUTER MODELING

Field modeling uses finite difference techniques to subdivide the volume being studied, in this case the simulated shipboard compartment, into small, finite volume elements. Using initial conditions specified by the user and the finite difference forms of the equations for conservation of mass, momentum, and energy, values of temperature, pressure, velocity and density are calculated for each of the individual volumetric elements at discrete time intervals. Additional modeling of physical effects such as radiation, turbulence and wall conduction are included to increase the validity of the simulation. The enormous number of calculations necessary for this type of modeling mandates the use of large amounts of computer memory and high speed processors.

The basis for this thesis is provided by a large number of previous research projects. One of the earliest successful models of this type was developed by Aziz and Hellums [Ref. 3] at Rice University in Houston, Texas. They expressed the Navier-Stokes equations in terms of vorticity and vector potential, then solved the resulting three dimensional finite difference equations using a combination of the alternating direction method and successive over-relaxation (SOR). Later work by Mallinson and de Vahl Davis [Ref. 4], Morrison and Tran [Ref. 5]; Chan and Banerjee [Ref. 6]; Ozeo, Fujii, Lior and Churchill [Ref. 7]; and a host of others have all expanded on the use of finite difference techniques to model convective heat transfer in closed compartments.
In the late 1970's and early 1980's, R.G. Rehm and H.R. Baum began developing equations to describe the buoyant flow induced by large scale fires [Refs. 8, 9, 10 and 11]. Work done at the University of Notre Dame used a two dimensional finite difference field model to predict velocities, temperatures and smoke concentrations in aircraft cabin fires [Ref. 12 and 13]. The development of a two dimensional model of transient, natural convection cooling was developed, and experimentally verified, by Nicolette et al [Ref. 14] using a semi-implicit upwind differencing scheme and global pressure corrections. Still more studies [Refs. 15, 16, 17, 18, & 19] have utilized finite difference methods to solve non-linear, three dimensional partial differential equations for rectangular enclosures.

C. THE FIRE TEST FACILITY

A full scale test facility has been constructed at the Naval Air Warfare Center (NAWC) at China Lake, CA. This facility, designed to simulate full scale shipboard compartments, consists of three chambers [see Figure 1 on page 4 and Figure 2 on page 5]. The main chamber measures 20 feet by 20 feet by 10 feet and is vented to the atmosphere through an opening in a side wall, in a manner intended to simulate the penetration of a missile or other projectile. The second chamber measures 15 feet by 15 feet by 10 feet is located adjacent to the main chamber to the east, while the third chamber, also measuring 15 feet by 15 feet by 10 feet, is located on top of the main chamber. These secondary chambers are intended to provide a means to study the vertical and horizontal heat transmission rates. [Ref. 2, pp. 7-10]

The entire test structure is constructed of 3.8 inch thick steel bulkheads walls reinforced by 5 inch I-beams on 5 foot centers, and 1/2 inch thick steel decks reinforced underneath by 12 inch I-beams on 5 foot centers. Additional support was provided for the overhead in the main chamber by several 5 inch I-beam columns. These columns were intended to eliminate any sagging and or distortion of the overhead as it is repeatedly cycled from ambient conditions to the extreme temperatures expected during the missile fuel tests. Access to all three compartments is through hatches which open to the outside of the structure and are kept closed during all testing. There are no openings between any two adjacent chambers. The ventilation opening is located on the north face of the main chamber to simulate an impact hole, such as a missile strike [see Figure 3 on page 6 and Figure 4 on page 7]. [Ref. 2, pp. 10-12]

Instrumentation is provided within the test chamber to measure radiation heat flux, total heat flux, compartment bulk pressure, wall temperatures and compartment gas.
Figure 1. Fire test chamber at NAWC, China Lake, CA.
Figure 2. Plan view of fire test chamber at NAWC, China Lake, CA.
Figure 3. North/south elevations of test chamber at NAWC, China Lake, CA.
Figure 4. East/west elevations of test chamber at NAWC, China Lake, CA.
temperatures. Thermocouples were arranged in vertical arrays of 10 thermocouples. The lowest thermocouple in each array was placed 6 inches above the deck with subsequent thermocouples spaced at 1 foot intervals ending 6 inches below the ceiling [see Figure 5 on page 9]. [Ref. 2, pp. 14-15]

D. THE COMPUTER PROGRAM

This computer model is a joint project undertaken by the Naval Postgraduate School and the University of Notre Dame. It represents a low cost, safe alternative to full scale fire testing. With proper modifications, and properly validated by full scale experimentation, this program should provide a valuable tool for testing the effectiveness of fire mitigation techniques and evaluation of new ship designs.

Work on this program began at Naval Postgraduate School in 1986 when Nies [Ref. 20] used a cartesian coordinate system to model the cylindrical spherical geometry of the FIRE-I test facility. In 1987 Raycraft [Ref. 21] modified the program to use a cylindrical spherical coordinate system. She also expanded the scope of this project by writing a companion program to calculate view factors and account for surface radiation effects. Using both of Raycraft's programs, Houck [Ref. 22] further expanded the scope of the project to account for the internal ventilation capabilities of the FIRE-I facility. Most recently, in 1991, McCarthy [Ref. 23] used advanced computer graphics techniques to provide a more accurate pictorial representation of the unique geometry of the FIRE-I facility and demonstrated the advantages of using full color displays to represent the three dimensional isotherm and velocity vector field profiles. This development greatly enhanced the presentation of the program output. McCarthy was also the first researcher at NPS to incorporate Raycraft's radiation view factor program as subprograms of the main computer code.

This thesis returns to the cartesian coordinate system used by Nies [Ref. 20], to model the Navy's newest fire test facility at the Naval Air Warfare Center (NAWC) at China Lake, CA. It also advances McCarthy's [Ref. 23] work by applying the enhanced graphics capabilities of the NCAR Graphics software, developed by the National Center for Atmospheric Research [Refs. 24 and 25], for output presentation.
Figure 5. Instrumentation of fire test chamber at NAWC, China Lake, CA.
II. THE NUMERICAL MODEL

A. THE GOVERNING EQUATIONS.

The model used in this code is built on the fundamental laws of conservation of mass (continuity), energy and momentum. One example of these equations is presented by Patankar [Ref. 26, pp. 11-17]. As in that work, this model neglects the pressure work and viscous dissipation terms of the energy equation due to the low flow velocities expected. The volumetric heat generation term will be used to account for heat input by the fire. Patankar’s version of the momentum equations [Ref. 26, p. 14, eq. 2.11] included both a body force and an extra viscous force. This model assumes a newtonian fluid therefore the extra viscous force is neglected and the only body force present is due to gravitational effects acting in the negative z-direction. Thus the body force in the z-momentum equation is set equal to \(- \rho g\) where \(g\) is the local gravitational constant; the negative sign indicates that \(g\) is acting in the negative z-direction; and \(\rho\) is the fluid density.

Nies [Ref. 20, pp. 16-38] expanded the equations presented by Patankar into the three dimensional, cartesian coordinate system used in this model. Following the development pattern of Doria [Ref. 18, pp. 4-7] and making the assumption that air is a homogeneous gas of constant composition (reactive variations due to the fire are neglected), Nies [Ref. 20, pp. 16-17] incorporated the equations of state for an ideal gas with constant specific heats in the form:

\[ P = \rho RT \]

and

\[ h = c_p(T - T_{ref}) \]

where \(P\), \(T\) and \(\rho\) represent the bulk pressure, temperature and density inside the control volume, \(R\) is the gas constant for air, \(h\) is the specific enthalpy, \(c_p\) is the specific heat and \(T_{ref}\) is a suitably chosen reference temperature.

Again, following the procedures developed by Patankar [Ref. 26] and expanded by Doria [Ref. 18], Nies [Ref. 20, pp. 21-26] goes on to place these six equations in their non-dimensional, integral forms. After subdividing the test chamber into a large, but
finite number of control volumes, finite difference techniques are used to solve for the six unknowns of temperature, pressure, density and the three components of velocity.

**B. THE CONTROL VOLUME.**

Each control volume, or cell, is defined by the nodal point contained at its center. Values for temperature, density and pressure are calculated at this central point (designated P) and are then assumed to hold for the entire cell. A secondary, or staggered grid system, is used to determine velocities. This staggered grid is offset from the main grid by one half the length of the cell [see Figure 6 on page 121]. As explained in McCarthy [Ref. 23, pp. 18-19] and Patankar [Ref. 26, pp. 115-120], use of this staggered grid alleviates two problems: first, since velocities are dependent upon pressure differentials, the staggered grid allows pressures to be determined at a more frequent interval thereby reducing the error associated with larger separations between nodal points; second, the stagger increases stability by decreasing and, or eliminating unrealistic, oscillatory velocity fields when adjacent velocities are used to satisfy the continuity equation.

Since Patankar's method uses primitive variables, in lieu of the stream function and vorticity, special attention must be paid to the coupling of equations through the pressure terms. An iterative process is used to calculate the pressure in each cell, then a local pressure correction is calculated to ensure that continuity is satisfied. Additional discussions of this correction are included in both Patankar [Ref. 26, pp. 120-128] and Doria [Ref. 18, pp. 26-32]. A global pressure correction was described by Nicolette, *et al* [Ref. 14] for addressing net energy changes within the closed compartment. This correction has also been incorporated into this model.

As stated in McCarthy [Ref. 23, p. 19], forcing convergence on a non-linear set of equations, like those describing fluid flow, can be difficult. A variety of schemes have been developed, each with its own set of strengths and weaknesses. This model applies iterative techniques to a solution scheme known as "QUICK", or Quadratic Upstream Interpolation for Convective Kinematics, developed by Leonard [Ref. 19]. With the accuracy of a central differencing scheme and the stability of the convective diffusion terms in upwind differencing, the QUICK scheme estimates values and gradients of the transport variables at the cell faces. The utility of this method was demonstrated by H.Q. Yang [Ref. 27] when he used QUICK to solve the coupled momentum and energy equations for three dimensional flow in a tilted rectangular enclosure.

In addition to the center cell described above, neighboring cells will be used in various computations. To keep track of these cells a standardized nomenclature must be
adopted. Assuming the central cell, cell P, to be located at the point \((i, j, k)\) where \(i\) and \(k\) correspond to the standard coordinated axis \(x\), \(y\) and \(z\) respectively, then each neighbor may be described as follows (NOTE: the following directions are for standardization of nomenclature only and do not necessarily correspond to compass directions as shown in Figure 1 on page 4):

- East \((i+1, j, k)\)
- West \((i-1, j, k)\)
- North \((i, j+1, k)\)
- South \((i, j-1, k)\)
- Front \((i, j, k+1)\)
- Back \((i, j, k-1)\)

The nodal point in each direction is designated by the capital letter corresponding to the direction (i.e. E, W, N, S, F, B), while the boundary between cell P and its neighbors is designated by a lower case letter corresponding to that direction (i.e. the boundary between cell P and cell N is designated n, and the boundary between P and F is f). A typical cell array on a rectangular grid is shown in figure Figure 7 on page 13.
As discussed previously, each P node is used to determine values of density, pressure and temperature which are then applied to its entire cell. Velocities are determined at the cell faces using the staggered grid arrangement described above.

C. DISCRETIZATION OF THE CONSERVATION EQUATIONS.

With that brief background, the integral forms of the conservation equations developed Nies [Ref. 20, pp. 25-26] may now be discretized. As discussed by Nies [Ref. 20, pp. 26-38] maximum stability and accuracy of the model is achieved by using three different finite differencing schemes. Forward differencing is used for timewise discretization, central differencing is used to discretize the diffusion terms, and the QUICK scheme is used to discretize the convective terms. Use of these techniques to discretize the governing equations was discussed in detail by Nies [Ref. 20, pp 26-38] and will not be repeated here. The following finite difference forms of the governing equations are taken from his work.

1. The Continuity Equation.

\[
(\rho_p - \rho_0) \frac{\Delta x \Delta y \Delta z}{\Delta t} + (G_e - G_w)\Delta y \Delta z + (G_n - G_s)\Delta z \Delta x + (G_f - G_b)\Delta x \Delta y = S_{mp}
\]
where $\rho_p$ is the density at the current time step; $\rho_p^t$ is the density at node $P$ at the previous time step; $\Delta t$ is the incremental time step; $\Delta x$, $\Delta y$, and $\Delta z$ are the cell dimensions in the indicated direction; and $S_{mp}$ is the residual mass term. The mass flux ($G$) terms are defined as:

$$G_e = \frac{(\rho_E + \rho_p)}{2} u_e$$

$$G_w = \frac{(\rho_p + \rho_w)}{2} u_w$$

$$G_n = \frac{(\rho_n + \rho_p)}{2} v_n$$

$$G_s = \frac{(\rho_s + \rho_p)}{2} v_s$$

$$G_f = \frac{(\rho_f + \rho_p)}{2} w_f$$

$$G_b = \frac{(\rho_p + \rho_b)}{2} w_b$$

It should be noted that in all of these equations $\rho$ refers to the density; $u$, $v$, and $w$ are the three velocity components in the $x$, $y$, and $z$-directions respectively; uppercase subscripts denote values at the indicated nodal point; and lowercase subscripts denote values at cell faces. Also, in order for continuity to be satisfied in this closed system, the residual mass term ($S_{mp}$) would equal zero. However, due to the approximation inherent in the numerical scheme, we will be satisfied if $S_{mp}$ tends toward zero as determined by comparison to an arbitrarily small threshold value.

2. The Energy Equation.

$$\left[H_{Ap} + \rho_p^0 \frac{\Delta x \Delta y \Delta z}{\Delta t}\right] h_p = H_{A_e} h_E + H_{A_w} h_W + H_{A_n} h_N + H_{A_s} h_S + H_{A_f} h_F + H_{A_b} h_B + H_{S_p}$$

where $h$ is the specific enthalpy at the current time step; $h^p$ is the specific enthalpy at the previous time step; and the coefficients (II) are defined as:

$$H_{A_e} = \frac{(1 - G_e - G_p)}{2} \Delta y \Delta z + \left(\frac{1}{Re Pr e}\right) \left(\frac{\Delta y \Delta z}{\Delta x}\right)$$

14
\[ H_{Aw} = \frac{(|G_\omega| + G_\omega)}{2} \Delta y \Delta z + \left( \frac{1}{Re_{Pr_1}} \right) n \frac{\Delta y \Delta z}{\Delta x} \]

\[ H_{An} = \frac{(|G_n| - G_n)}{2} \Delta z \Delta x + \left( \frac{1}{Re_{Pr_1}} \right) n \frac{\Delta z \Delta x}{\Delta y} \]

\[ H_{As} = \frac{(|G_s| + G_s)}{2} \Delta z \Delta x + \left( \frac{1}{Re_{Pr_1}} \right) s \frac{\Delta z \Delta x}{\Delta y} \]

\[ H_{Af} = \frac{(|G_f| - G_f)}{2} \Delta x \Delta y + \left( \frac{1}{Re_{Pr_1}} \right) f \frac{\Delta x \Delta y}{\Delta z} \]

\[ H_{Ab} = \frac{(|G_b| + G_b)}{2} \Delta x \Delta y + \left( \frac{1}{Re_{Pr_1}} \right) b \frac{\Delta x \Delta y}{\Delta z} \]

\[ H_{As} + H_{Aw} + H_{An} + H_{As} + H_{Af} + H_{Ab} \]

\[ H_{Sp} = \rho_0 \theta_0 \Delta x \Delta y \Delta z \]

Also used above are the turbulent Reynolds number \((Re_t)\) and the turbulent Prandtl number \((Pr_t)\). These are defined as

\[ Re_t = \frac{\rho_0 u_0 H}{\mu_{eff}} \]

\[ Pr_t = \frac{\mu_{eff} c_p}{k_{eff}} \]

where \(H\) is the characteristic length (defined as the height of the test chamber in our model); \(\rho\) and \(c_p\) represent the density and specific heat of air, while the subscript \(0\) indicates that the properties are to be evaluated at the initial conditions which existed prior to ignition of the fire; \(u_0\) is the reference velocity (set equal to 1.0 ft/sec); and \(\mu_{eff}\) and \(k_{eff}\) are effective values of viscosity and conductivity as defined by Nies [Ref. 20, p. 39-40].

3. The Momentum Equations.

As stated by Nies [Ref. 20, p. 31] the momentum equations are more complex than the previous equations because of the use of the staggered grid and the addition of the shear stress terms. Staggered grids are determined by shifting the main grid one half cell in the negative direction along one axis at a time. Maintaining the nomenclature-
ture established for the basic, centered cells, the central node of a cell staggered in the x-direction becomes \( w \), the central node in a cell staggered in the y-direction becomes \( s \), and the central node in a cell staggered in the z-direction becomes \( b \). Likewise, the faces of the staggered cell are designated by the capital letters representing the basic nodes through which they pass. Thus, in the staggered grid \( P \) always represents the positive face along the axis where the shift (stagger) is being evaluated. Thus, the x-momentum equation becomes:

\[
\left[ A_w + \rho_w \frac{\Delta x \Delta y \Delta z}{\Delta t} \right] u_w = A_e u_e + A_{nw} u_{nw} + A_{nw} u_{nw} + A_{sw} u_{sw} + A_{fw} u_{fw} + A_{bw} u_{bw} + S_w
\]

where

\[
A_e = \left[ \frac{(\left| G_p \right| - G_p)}{2} + \frac{\left( \frac{1}{Re} \right)_p}{\Delta x} \right] \Delta y \Delta z
\]

\[
A_{nw} = \left[ \frac{(\left| G_{nw} \right| + G_{nw})}{2} + \frac{\left( \frac{1}{Re} \right)_{nw}}{\Delta x} \right] \Delta y \Delta z
\]

\[
A_{nw} = \left[ \frac{(\left| G_{nw} \right| - G_{nw})}{2} + \frac{\left( \frac{1}{Re} \right)_{nw}}{\Delta y} \right] \Delta x \Delta z
\]

\[
A_{sw} = \left[ \frac{(\left| G_{sw} \right| + G_{sw})}{2} + \frac{\left( \frac{1}{Re} \right)_{sw}}{\Delta y} \right] \Delta x \Delta z
\]

\[
A_{fw} = \left[ \frac{(\left| G_{fw} \right| - G_{fw})}{2} + \frac{\left( \frac{1}{Re} \right)_{fw}}{\Delta x} \right] \Delta y \Delta z
\]

\[
A_{bw} = \left[ \frac{(\left| G_{bw} \right| + G_{bw})}{2} + \frac{\left( \frac{1}{Re} \right)_{bw}}{\Delta x} \right] \Delta y \Delta z
\]

\[
A_w = A_e + A_{nw} + A_{sw} + A_{fw} + A_{bw}
\]
\[ S_* = \rho_* u_*^2 \frac{\Delta x \Delta y \Delta z}{\Delta t} - (P_p - P_w) \Delta y \Delta z \]

\[ + (u_p - u_{ww})(\frac{1}{\text{Re}_t})_p \Delta y \Delta z - (u_{ww} - u_{ww})(\frac{1}{\text{Re}_t})_{ww} \Delta y \Delta z \]

\[ + \left[ (v_{NW} - v_{NWw})(\frac{1}{\text{Re}_t})_{nw} - (v_{ww} - v_{ww})(\frac{1}{\text{Re}_t})_{ww} \right] \Delta z \]

\[ + \left[ (w_{wf} - w_{wwf})(\frac{1}{\text{Re}_t})_{wf} - (w_{ww} - w_{ww})(\frac{1}{\text{Re}_t})_{ww} \right] \Delta y \]

and

\[ \rho_w^0 = \frac{\rho_p^0 + \rho_{ww}^0}{2} \]

\[ u_p = \frac{u_e + u_w}{2} \]

\[ u_{ww} = \frac{u_{ww} + u_{ww}}{2} \]

\[ u_{nw} = \frac{u_{nw} + u_{nw}}{2} \]

\[ u_{sw} = \frac{u_{sw} + u_{sw}}{2} \]

\[ u_{fw} = \frac{u_{fw} + u_{fw}}{2} \]

\[ u_{bw} = \frac{u_{bw} + u_{bw}}{2} \]

\[ G_p = \rho_{pp} u_p \]

\[ G_w = \rho_{ww} u_w \]

\[ G_{nw} = \left[ \frac{\frac{\rho_N + \rho_p}{2} v_n + \frac{\rho_{NW} + \rho_w}{2} v_{ww}}{2} \right] \]
Development of the equations for y and z-momentum proceed in a similar fashion and will not be repeated here in the interest of brevity.

D. PRESSURE CORRECTIONS.

1. Global Pressure Corrections.

As described by Nies [Ref. 20, pp. 50-52] and McCarthy [Ref. 23, pp.47-48], in a fixed mass - fixed volume system, overall pressure depends on the net energy added or removed from the system. Nicolette, et al, [Ref. 14] demonstrated that in such a system, with a uniform grid, the sum of the product of density times volume for all of the cells remains fixed at the total mass of the system. Therefore, at any time during the fire, the total mass of the system must equal the initial mass at the equilibrium density which existed before the fire was ignited. This may be expressed as:

\[ \sum \rho_i^n (\Delta x \Delta y \Delta z)_i = \sum \rho_{EQ,j} (\Delta x \Delta y \Delta z)_j \]

Since the grid is uniform, the term (\(\Delta x \Delta y \Delta z\), is a constant, independent of time, it may be divided out of both sides of the equation leaving:

\[ \sum \rho_i^n = \sum \rho_{EQ,j} \]
Now, since we are operating under the assumption that the fluid inside our burn chamber is an ideal gas, and recalling that we are also working in a fixed volume environment, the density ($\rho$) may now be expressed as a function of pressure and temperature only.

$$\rho_i = f(P_i, T_i)$$

The actual pressures and temperatures may now be expressed as the sum of an estimated value ($P^*$ and $T^*$) and a global correction ($P_g$ and $T_g$).

$$P = P^* + P_g$$

$$T = T^* + T_g$$

Now applying the ideal gas law and substituting into the summation relation shown above, we can solve for the global pressure correction as

$$P_g = \frac{\sum_i P_{EQ}(\frac{1}{T_i} - \frac{1}{T^*}) - \sum_i \frac{P^*}{T^*}}{\sum_i \frac{1}{T^*}}$$

This relation is then iterated until a global pressure correction is obtained which conserves mass for all the cells.

2. Local Pressure Corrections.

As explained by both McCarthy [Ref. 23, pp.49-51] and Nies [Ref. 20, pp. 52-54], the method for obtaining the local pressure correction was developed by Patankar [Ref. 26, pp. 120-126] and Doria [Ref. 18, pp. 26-32] and is similar to the method used for determination of the global pressure correction. For this correction, the pressure distribution found during the previous time step is used to estimate the velocity field. Continuity is then applied and residual mass terms ($S_m$) are calculated for each cell. Based on these residual mass terms a pressure correction is estimated and the process is repeated until the values of $S_m$ fall below a previously established threshold value at which point the final value of the correction is now known. As in the determination of the global pressure correction, the total pressure is expressed as the sum of the estimated pressure and the local correction.
\[ p = p^e + p' \]

where \( p \) is the total pressure, \( p^e \) is the estimated pressure, and \( p' \) is the local pressure correction. The local pressure correction may now be expressed in its finite difference form

\[ A_p p' = A_E p'_E + A_W p'_W + A_N p'_N + A_S p'_S + A_F p'_F + A_B p'_B - S_m p \Delta x \Delta y \Delta z \]

where

\[
A_E = \frac{\rho_e (\Delta y \Delta z)^2}{\Delta t (A_e + \rho_e \Delta x \Delta y \Delta z)} \\
A_W = \frac{\rho_w (\Delta y \Delta z)^2}{\Delta t (A_w + \rho_w \Delta x \Delta y \Delta z)} \\
A_N = \frac{\rho_n (\Delta z \Delta x)^2}{\Delta t (A_n + \rho_n \Delta x \Delta y \Delta z)} \\
A_S = \frac{\rho_s (\Delta z \Delta x)^2}{\Delta t (A_s + \rho_s \Delta x \Delta y \Delta z)} \\
A_F = \frac{\rho_f (\Delta z \Delta y)^2}{\Delta t (A_f + \rho_f \Delta x \Delta y \Delta z)} \\
A_B = \frac{\rho_b (\Delta z \Delta y)^2}{\Delta t (A_b + \rho_b \Delta x \Delta y \Delta z)}
\]

\[ A_p = A_E + A_W + A_N + A_S + A_F + A_B \]

At solid boundaries where the mass flux (\( G \)) is zero, the appropriate coefficient (\( A \)) corresponding to that boundary is also set equal to zero.

After the local pressure correction (\( p' \)) is determined, new velocities are determined from the following relations:
\[
\begin{align*}
  u &= u^x + u' \\
  v &= v^x + v' \\
  w &= w^x + w'
\end{align*}
\]

where

\[
\begin{align*}
  u' &= \frac{(P' p - P' W) \Delta y \Delta z}{A_w + \rho_w \frac{\Delta x \Delta y \Delta z}{\Delta t}} \\
  v' &= \frac{(P' p - P' S) \Delta z \Delta x}{A_t + \rho_t \frac{\Delta x \Delta y \Delta z}{\Delta t}} \\
  w' &= \frac{(P' p - P' b) \Delta x \Delta y}{A_b + \rho_b \frac{\Delta x \Delta y \Delta z}{\Delta t}}
\end{align*}
\]

The residual mass \( (S_{mp}) \) is again compared to the threshold value and the entire process is iterated if necessary.

E. INITIAL AND BOUNDARY CONDITIONS.

Before this system of equations can be solved, appropriate initial and boundary conditions must be determined and applied. As in the previous work conducted by Nies, Raycraft, Houck and McCarthy [Refs. 20, 21, 22, and 23], these conditions are established as follows:

1. Initial Conditions.

The initial conditions for the model are determined by the conditions existing inside the test chamber just prior to starting the fire. It is assumed that the air is uniformly at rest, thus all components of velocity are set equal to zero throughout the chamber. The temperature inside the chamber is assumed to be uniform and equal to the ambient temperature outside the chamber, therefore the non-dimensional temperature field is set equal to 1.0 throughout the model. Finally, pressure and density are also assumed to be uniformly distributed and in static equilibrium.

2. Boundary Conditions.

The chamber walls are constructed of standard 3/8 inch sheet steel [Ref. 2, p. 10], therefore they are presumed to be non-porous. This allows all velocity components
to be set equal to zero at the wall (the so called "no slip" condition), and the mass flux across the wall is also set to zero (the "impermeable wall" condition). Since there is no heat generation inside the walls of the chamber, it can be assumed that no discontinuities exist between the temperature of the surface of the wall and the air immediately adjacent to it. Therefore the inside surface temperature of the wall must be identically equal to the temperature of the fluid immediately adjacent to it. These condition may be expressed as follows:

\[ u_{surf} = 0 \]
\[ v_{surf} = 0 \]
\[ w_{surf} = 0 \]
\[ T_{surf} = T_{air} \]

Finally, conservation of energy must also be satisfied at the wall. Therefore

\[ q_r - k_{air} \frac{\delta T_{air}}{\delta n} = - k_{surf} \frac{\delta T_{surf}}{\delta n} \]

where \( n \) represents the inward pointing normal at that location; and \( q_r \) represents the energy transfer due to thermal radiation.

F. MODELING OF PHYSICAL PHENOMENA.

1. Wall Conduction.

Heat losses through the walls are calculated assuming one dimensional, unsteady heat conduction through walls of uniform conductivity. A constant convective heat transfer condition is assumed to exist between the exterior of the wall and the environment.

2. Turbulence.

As explained by McCarthy [Ref. 23, pp. 15-16] and Nies [Ref. 20, pp. 39-40] the turbulence model used in this code is a simple algebraic model first developed by Nee and Liu [Ref. 28]. This model calculates an effective viscosity (\( \mu_{eff} \)) and an effective conductivity (\( k_{eff} \)) for recirculating buoyant flows with widely fluctuating turbulence levels. Development of these equations, as used in this model, is shown in Nies [Ref. 20, p. 39-40].
3. **Radiation.**

The radiation model used in this code considers only surface radiation while considering the gas and smoke to be transparent. Developed and explained in detail by Raycraft [Ref. 21], the model is based on the net radiosity method discussed by Sparrow and Cess [Ref. 29]. As summarized by McCarthy [Ref. 23, p. 17], the model treats both the chamber walls and the flame areas as grey, diffuse surfaces.
III. THE COMPUTER PROGRAMS

A. INTRODUCTION.

Running the computer code for this project, from initial data input to final graphical output, is a three step process. Initial input is accomplished using the program FIREBLD; numerical data output is generated by the program FIRE; and final graphical output is provided by the programs ISOTHERM and VELOCITY, both of which are written to utilize the graphics software developed by the National Center for Atmospheric Research (NCAR).

B. PROGRAM FIREBLD.

Program FIREBLD [Appendix A] is used to build and or modify the data file FIRE.DATA which provides the input data required to run the main code, FIRE. Originally part of SUBROUTINE INPUT in FIRE, FIREBLD represents an effort to improve the “user friendliness” of the code. It accomplishes this by creating an interactive input environment which attempts to minimize the user’s need for detailed knowledge of the internal operation of FIRE. As the user friendliness of the SUBROUTINE INPUT increased, so did its length. It quickly became the largest subroutine in FIRE, and in an effort to streamline that code, it was converted into a separate program. This conversion had two advantageous effects:

1. It increased the run time options for FIRE by eliminating the need for direct operator input while running the program. This allows FIRE to take advantage of the time savings associated with running in background and or batch modes; and

2. Elimination of the necessity for direct operator input has reduced the total run time of the code (FIRE) by placing the slow process of data input in a separate program.

FIREBLD is designed to query the user regarding the various input parameters, indicating the proper units when appropriate. It begins by looking for a previously created input file. If one is found the user is asked if it should be used, a negative response causes program termination, while a positive response causes the data file to be read. The program then continues by displaying the existing data and asking the user if any changes are desired. If a previously created input file is not located, FIREBLD automatically enters input mode and prompts the user for the required information.
Some of the input data used in this project is shown in Table 1 on page 25 and Table 2 on page 26. Additional data which may be input at the user's discretion includes:

Thermocouple data, including number and location. The present code is limited to a maximum of 20 thermocouples, but this may be changed with only minor modifications to the code.

Mass source data, including location and size. This portion of the code was not required for this project and therefore has not been tested.

Internal solids data, including location, size, conductivity, specific heat, and fan speed. The fan speed included here is for internal ventilation contained wholly inside the test chamber. This is similar to the ventilation incorporated into the code by Houck [Ref. 22]. The problems experienced by McCarthy [Ref. 23] in association with this ventilation, have not been seen here. It should also be noted that the external chamber walls are input as internal solids.

<table>
<thead>
<tr>
<th>Table 1. INPUT DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Directions</strong></td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Chamber dimensions</td>
</tr>
<tr>
<td>Length (feet)</td>
</tr>
<tr>
<td>Wall thickness (inches)</td>
</tr>
<tr>
<td>Floor ceiling thickness (inches)</td>
</tr>
<tr>
<td>Number of Computational Cells</td>
</tr>
<tr>
<td>Inside the chamber</td>
</tr>
<tr>
<td>Total number of cells</td>
</tr>
<tr>
<td>Fire location</td>
</tr>
<tr>
<td>Starting node</td>
</tr>
<tr>
<td>Ending node</td>
</tr>
<tr>
<td>Times (seconds)</td>
</tr>
<tr>
<td>Total length of run</td>
</tr>
<tr>
<td>Incremental time step</td>
</tr>
<tr>
<td>Between data saves</td>
</tr>
<tr>
<td>Between data saves for plotting</td>
</tr>
<tr>
<td>Fire start time</td>
</tr>
</tbody>
</table>
Table 2. WALL DATA: The following data is input through FIREBLD to identify the location of the chamber walls.

<table>
<thead>
<tr>
<th>Location</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall #1, starting node</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wall #1, number of nodes</td>
<td>1</td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>Wall #2, starting node</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wall #2, number of nodes</td>
<td>21</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Wall #3 (floor), starting node</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wall #3, number of nodes</td>
<td>21</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>Wall #4 (ceiling), starting node</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>Wall #4, number of nodes</td>
<td>21</td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>Wall #5, starting node</td>
<td>0</td>
<td>21</td>
<td>0</td>
</tr>
<tr>
<td>Wall #5, number of nodes</td>
<td>21</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Wall #6, starting node</td>
<td>21</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wall #6, number of nodes</td>
<td>1</td>
<td>21</td>
<td>11</td>
</tr>
</tbody>
</table>

C. PROGRAM FIRE.

Operational details of the main program FIRE [Appendix B] are provided by McCartney [Ref. 23, pp. 55-56]. The only significant difference between his version of the program and the present version (neglecting the obvious difference of compartment geometry) is that the separate program required by McCarthy for the generation of view factors, is incorporated into the present code as a subroutine, SUBROUTINE VIEW. Also, the present code inputs the wall locations and properties as "internal solids", as noted above.

Heat input from the fire is modeled as a volumetric heat source based on the dimensions of the fire as provided by FIREBLD. The magnitude of this heat source is calculated in SUBROUTINE CALQ. It is based on the known heat of combustion of the fuel used (2600 Btu/lbm, as provided by NAWC, China Lake, CA) and an estimated burn rate of 1.0 lbm/sec. However, due to temperature and velocity instabilities associated with this high heat generation rate, it was necessary to reduce the burn rate by two
orders of magnitude to 0.01 lbm sec in order to achieve stable output data. Increasing the burn rate to its proper level would necessitate reductions in the time step used, and require multiple data runs to confirm stable output. Due to time restrictions placed on completion of this work, those additional data runs were left for later investigators.

In making comparisons to McCarthy's work [Ref. 23, p. 55], it is interesting to note that he used the VAXSTATION 3100 which required approximately 3600 seconds of CPU time to process one second of fire time. The current work required approximately 677 seconds of CPU time per second of fire time, utilizing the AMDAHL mainframe processor here at Naval Postgraduate School. Even when performance is broken down further, to a CPU seconds per fire second per cell basis, the AMDAHL outperforms the VAX by a 5.5 or 6 to 1 margin. Although no attempt was made to determine the source of this improvement, the majority of it can be attributed to the increased clock speed of the AMDAHL machine, which unconfirmed estimates place at approximately 4 to 1 over the VAX.

D. GRAPHICAL ANALYSIS.

The value of graphical analysis was successfully demonstrated by McCarthy [Ref. 23]. This work utilizes a different graphics software package in order to standardize the software used at both Naval Postgraduate School and the University of Notre Dame. The software package chosen, NCAR Graphics, was created at the National Center for Atmospheric Research (NCAR) and is intended specifically for scientific applications. The primary advantage of NCAR over the previously used graphics software is in the flexibility of output presentation built in to the NCAR software.

The two graphics programs used in the present study, ISOTHERM [Appendix C] and VELOCITY [Appendix D] each utilize the three dimensional PLOT.DA TA file created by FIRE to generate two dimensional temperature and velocity profiles as requested by the user. Each program allows the user to specify the desired two dimensional view (or section) (i.e. Plan view, X-Z profile, or Y-Z profile) and the location (or elevation) in the third dimension where that section is to be taken. Each program then performs two linear interpolations, the first to locate the requested section, and the second to convert the grid used by FIRE to a uniform grid as required by the NCAR Graphics software.

Scaling of the output in each program is accomplished in different manners. VELOCITY sets the maximum vector size in any given plot by the maximum interpolated velocity for that section (NOTE: velocities are shown in centimeters per second). While the color scale used in each plot generated by ISOTHERM is determined by dividing the
range from ambient (70 ° F = 21.1 ° C) to the maximum temperature in the non-interpolated data into 14 color zones. These scaling techniques are examples of the flexibility of the NCAR Graphics software, and each user may customize the output to satisfy the current needs desires with minimal effort.

Color isotherm graphics are printed on a DEC LJ250 Companion Color printer using ink jet technology, attached to a VAXStation 3100 M38 standalone workstation. Velocity profiles are printed on a DEC LN03 Laser printer attached to the VAX cluster in the Mechanical Engineering Department at the Naval Postgraduate School. The following figures [Figure 8 on page 29 through Figure 25 on page 46] show the temperature and velocity profiles for each of the available view planes at 30, 60 and 90 seconds of fire time. Elevations are shown on each figure and were chosen to provide a plan view at mid-compartment height and profiles through the fire. Also included in both ISOTHERM and VELOCITY is the option to "zoom" in on any localized region of the plot as desired by the user. Figure 26 on page 47 and Figure 27 on page 48 demonstrate this capability. Note also, that the axis on all of these plots are marked (in feet) to indicate the section being viewed.
Figure 8. Temperature Profile, Plan View, 30 seconds.
Figure 9. Temperature Profile, Plan View, 60 seconds.
Figure 10. Temperature Profile, Plan View, 90 seconds.
Figure 11. Temperature Profile, X-Z Profile, 30 seconds.
Figure 12. Temperature Profile, X-Z Profile, 60 seconds.
Figure 13. Temperature Profile, X-Z Profile, 90 seconds.
Figure 14. Temperature Profile, Y-Z Profile, 30 seconds.
Figure 15. Temperature Profile, Y-Z Profile, 60 seconds.
Figure 16. Temperature Profile, Y-Z Profile, 90 seconds.
THIS PAGE INTENTIONALLY LEFT BLANK
Figure 17. Velocity Profile, Plan View, 30 seconds.
Figure 18. Velocity Profile, Plan View, 60 seconds.
Figure 19. Velocity Profile, Plan View, 90 seconds.
Figure 20. Velocity Profile, X-Z Profile, 30 seconds.
Figure 21. Velocity Profile, X-Z Profile, 60 seconds.
Figure 22. Velocity Profile, X-Z Profile, 90 seconds.
Figure 23. Velocity Profile, Y-Z Profile, 30 seconds.
Figure 24. Velocity Profile, Y-Z Profile, 60 seconds.
Figure 25. Velocity Profile, Y-Z Profile, 90 seconds.
Figure 26. Example of "Zoom" feature of ISOTHERM.
Figure 27. Example of "Zoom" feature of VELOCITY.
IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS.

1. Program FIREBLD represents a marked improvement in user friendly data input to FIRE.

2. Additional improvements can be made in user friendliness of FIREBLD, and thus FIRE. Specifically, all locations and dimensions should be input as length measurements (i.e. feet or meters) by the user.

3. The burn rate used by SUBROUTINE CALQ in program FIRE is artificially low and must be increased before the code can be properly validated against actual test data.

4. The NCAR Graphics software provides increased flexibility in graphics output over the previously used graphics package.

5. The temperature and velocity profiles generated by this work appear to have the expected characteristics of an actual fire. Further testing and comparison to actual test data is necessary for validation of the code.

6. The ink jet printer used to create hard copy graphical printouts of the data in this work is slow and of marginal quality for professional publication. Improved graphics printing capabilities should be pursued.

B. RECOMMENDATIONS.

1. Increase user friendliness of FIREBLD by putting all input in terms easily determined by the user. Specifically:
   - Input locations in terms of length measurements vice nodal locations.
   - User should have his choice of units (SI or English) when inputting data.
   - Input data should not require any manipulation by the user prior to entry.

2. Increase the burn rate used by SUBROUTINE CALQ in program FIRE to realistic levels and adjust size of time step as necessary to achieve stability.

3. Expand FIRE to completely model test chamber at the Naval Air Warfare Center, China Lake, CA. Specifically, include the natural vent and adjacent compartments.

4. Validate model against actual test procedures and results from Naval Air Warfare Center, China Lake.

5. Examine alternative graphical presentation schemes available with the NCAR Graphics software. Specifically, test feasibility of combining isotherm and velocity plots into single output. Also, examine use of NCAR Graphics movie making capabilities to provide a "real time" representation of data.

6. Enhance clarity of graphics printouts by procuring access to a laser quality color printing.
APPENDIX A. PROGRAM FIREBLD

This program is used to build and or modify the input data file required by the main program, FIRE.

PROGRAM FIREBLD
******************************************************************************
**
** THREE-DIMENSIONAL NUMERICAL SIMULATION
** OF A FIRE SPREAD INSIDE A BUILDING
**
** DEVELOPED BY:
** H.Q. YANG AND K.T. YANG
**
** DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING
** UNIVERSITY OF NOTRE DAME
** NOTRE DAME, INDIANA, 46556
**
** DEC. 1986
**
** AMENDED BY
** K.J. THORKILDSEN
** LIEUTENANT, U.S. COAST GUARD
**
** DEPARTMENT OF MECHANICAL ENGINEERING
** NAVAL POSTGRADUATE SCHOOL
** MONTEREY, CALIFORNIA 93942
**
** SEP. 1992
**
**
******************************************************************************
* THIS PROGRAM BUILDS AND/OR MODIFIES THE INPUT DATA FILE REQUIRED FOR *
* THE PROGRAM LISTED ABOVE. *
******************************************************************************
* THIS SUBROUTINE SETS UP REQUIRED VALUES TO BEGIN THE PROGRAM. *
*
* VARIABLES ARE:
*  KRUN = RESTART INDICATOR
*  NCHIP = NUMBER OF INTERNAL SOLID PIECES
*  NMS = NUMBER OF MASS SOURCES
*  NWRP = NUMBER OF TIME STEPS BETWEEN WRITES TO OUTPUT FILE
*  NTHCO = NUMBER OF THERMOCOUPLES TO PRINT OUT
*  TMAX = NONDIMENSIONAL MAXIMUM TIME ALLOWED
*  XTMAX = MAXIMUM TIME ALLOWED (SECONDS)
*  TWRITE = TIME BETWEEN FIELD VARIABLE OUTPUT (SECONDS)
*  TTAPE = TIME INTERVAL BETWEEN PLOTS (SECONDS)
*  DTIME = NONDIMENSIONAL TIME STEP
*  XDTIME = TIME STEP (SECONDS)
*  HSTART = FIRE START TIME (SECONDS)
*  NHSZ(1,1) = STARTING NODE OF HEAT SOURCE, X-DIR

50
* NHSZ(2,1) = Y-DIR
* NHSZ(3,1) = Z-DIR
* NHSZ(1,2) = ENDING NODE OF HEAT SOURCE, X-DIR
* NHSZ(2,2) = Y-DIR
* NHSZ(3,2) = Z-DIR
* ICHPB = FIRST NODE OF INTERNAL SOLID IN X DIR
* JCHPB = FIRST NODE OF INTERNAL SOLID IN X DIR
* KCHPB = Y DIR
* NCHPI = NUMBER OF INTERNAL SOLID NODES IN X DIR
* NCHPJ = Z DIR
* ICMB = FIRST NODE OF INTERNAL SOLID IN X DIR
* JCMSB = FIRST NODE OF INTERNAL SOLID IN X DIR
* KMSB = Y DIR
* NMSI = NUMBER OF INTERNAL SOLID NODES IN X DIR
* NMSJ = Z DIR
* RMS = DIMENSIONLESS MASS SOURCE
  (= CMF/(60.*UO*NMSI*NMSJ*NMSK))
* CX,CY,CZ = THERMOCOUPLE POSITIONS IN X,Y,Z

*(DATA FILES USED IN THIS PROGRAM:
*  FILE # 10 = FIRE DATA B1 : INITIAL SET-UP DATA
*
********************************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
CHARACTER ANS*1
LOGICAL L1
DIMENSION ICHPB(20) ,NCHPI(20) ,JCHPB(20) ,NCHPJ(20) ,KCHPB(20),
& NCHPK(20) ,CPS(20) ,CONS(20) ,WFAN(20) ,ICMB(20) ,NMSI(20),
& JCMSB(20) ,NMSJ(20) ,KMSB(20) ,NMSK(20) ,RMS(20) ,CX(30),
& CY(30) ,CZ(30) ,NHSZ(3,2)
PARAMETER (UO=1.0)
INQUIRE (FILE='/FIRE DATA B1',EXIST=L1)
IF (L1) THEN
  PRINT *, 'INPUT DATA FILE FOUND!'
  PRINT *
  PRINT *, 'DO YOU WISH TO USE IT FOR INPUT?'
  READ (*,*), ANS
  IF(INDEX(ANS,'Y').GT.0.OR,INDEX(ANS,'y').GT.0) THEN
    C *** READ IN DATA FROM EXISTING DATA FILE
      OPEN(10,FILE='/FIRE DATA B1',STATUS='OLD')
      READ(10,*) X,Y,H,TFLR,TWAL,TA
      READ(10,*) NI,NJ,NK
      READ(10,*) NCHIP,NMS,NWRP,NTHCO
      READ(10,*) TMAX,DTIME,TTAPE,TW**TE,HSTART
      READ(10,*) NHSZ(1,1),NHSZ(1,2),NHSZ(2,1),NHSZ(2,2),
        & NHSZ(3,1),NHSZ(3,2)
      IF (NCHIP.LE.0) GOTO 33
      DO 32 N=1,NCHIP
        READ(10,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
          & NCHPK(N),CPS(N),CONS(N),WFAN(N)
  END IF
      C *** READ IN DATA FROM EXISTING DATA FILE
  END IF
C *** END PROGRAM
CONTINUE
IF (NMS.LE.0) GOTO 37
DO 36 N=1,NMS
READ(10,*) IMSB(N),NMSI(N),JMSB(N),NMSJ(N),KMSB(N),
NMSK(N),RMS(N)
&
CONTINUE
DO 38 I1=,NTHCO
READ (10,*) CX(I),CY(I),CZ(I)
&
C *** UPDATE EXISTING DATA
C *** UPDATE COMPARTMENT DIMENSIONS
PRINT *, 'CURRENT DATA:'
PRINT *
PRINT *, 'COMPARTMENT DIMENSIONS'
PRINT *, 'LENGTH (X DIRECTION) : ', X, ' FEET'
PRINT *, 'WIDTH (Y DIRECTION)  : ', Y, ' FEET'
PRINT *, 'HEIGHT (Z DIRECTION) : ', H, ' FEET'
PRINT *, 'WALL THICKNESS      : ', TWAL, ' INCHES'
PRINT *, 'FLOOR/CEILING THICKNESS: ', TFLR, ' INCHES'
PRINT *
PRINT *, 'DO YOU WISH TO CHANGE ANY OF THESE VALUES?'
READ(5,*) ANS
IF(INDEX(ANS, 'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
PRINT *, 'DO YOU WISH TO CHANGE THE LENGTH?'
READ(5,*) X
ENDIF
PRINT *, 'DO YOU WISH TO CHANGE THE HEIGHT?'
READ(5,*) H
ENDIF
PRINT *, 'DO YOU WISH TO CHANGE THE WIDTH?'
READ(5,*) Y
ENDIF
PRINT *, 'DO YOU WISH TO CHANGE THE WALL THICKNESS?'
READ(5,*) TWAL
ENDIF
PRINT *,
'DO YOU WISH TO CHANGE THE FLOOR/CEILING THICKNESS?'
READ(5,*) TFLR
ENDIF
C *** UPDATE NUMBER OF CELLS
PRINT *
PRINT *, 'CURRENT DATA:
PRINT *
PRINT *, 'NUMBER OF CELLS:
PRINT *, 'X DIRECTION: ', NI
PRINT *, 'Y DIRECTION: ', NJ
PRINT *, 'Z DIRECTION: ', NK
PRINT *
PRINT *, 'DO YOU WISH TO CHANGE ANY OF THESE VALUES?
READ(5,*) ANS
IF(INDEX(ANS, 'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
  PRINT *, 'DO YOU WISH TO CHANGE THE NUMBER OF X CELLS?
  READ(5,*) NI
ENDIF
PRINT *, 'DO YOU WISH TO CHANGE THE NUMBER OF Y CELLS?
READ(5,*) NJ
ENDIF
PRINT *, 'DO YOU WISH TO CHANGE THE NUMBER OF Z CELLS?
READ(5,*) NK
ENDIF

C *** UPDATE INTERNAL SOLIDS
PRINT *
PRINT *, 'CURRENT DATA:
PRINT *
PRINT *, 'NUMBER OF INTERNAL SOLID PIECES: ', NCHIP
IF(NCHIP.EQ.0) GOTO 42
PRINT *, 'PIECE STARTING NODES NUMBER OF NODES THERMAL SPECIFIC FAN'
& 'CONDUCTIVITY HEAT SPEED'
DO 40 N=1,NCHIP
  PRINT 2,' ',N,ICHPB(N)-2,JCHPB(N)-2,KCHPB(N)-2,NCHPI(N),
& NCHPJ(N),NCHPK(N),CONS(N),CPS(N),WFAN(N)
  2 FORMAT (2X,I3,2(4X,3(I3,1X)),4X,F9.4,2X,F8.4,2X,F5.1)
40 CONTINUE
42 PRINT *, 'DO YOU WISH TO CHANGE THE NUMBER OF SOLID PIECES?
READ(5,*) NCHIP
IF(NCHIP.EQ.0) GOTO 44
53
DO YOU WISH TO CHANGE POSITION OF THE SOLID PIECES?
READ(5,*) ANS
IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
   DO 46 N=1,NCHIP
      PRINT *, 'FOR SOLID PIECE NUMBER ',N,' ENTER THE'
      PRINT *, 'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
      READ(5,*) ICHPB(N),JCHPB(N),KCHPB(N)
      ICHPB(N)=ICHPB(N)+2
      JCHPB(N)=JCHPB(N)+2
      KCHPB(N)=KCHPB(N)+2
      PRINT *,
      'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
      READ(5,*) NCHPI(N),NCHPJ(N),NCHPK(N)
      PRINT *,'THERMAL CONDUCTIVITY'
      READ(5,*) CONS(N)
      PRINT *,'SPECIFIC HEAT'
      READ(5,*) CPS(N)
      PRINT *,'AND FAN VELOCITY (0 IF NO FAN)'
      READ(5,*) WFAN(N)
   46 CONTINUE
ENDIF

C *** UPDATE MASS SOURCE DATA
44 PRINT *
   PRINT *, 'CURRENT DATA:'
   PRINT *
   PRINT *, 'NUMBER OF MASS SOURCES: ',NMS
   IF(NMS.EQ.0) GOTO 50
   PRINT *, 'SOURCE STARTING NODE NUMBER OF NODES FLOW'
   PRINT *, ' NO. X Y Z X Y Z RATE'
   DO 52 N=1,NMS
      PRINT 4,N,IMSB(N)-2,JMSB(N)-2,KMSB(N)-2,NMSI(N),
      NMSJ(N),NMSK(N),
      & RMS(N)*(60.*H**2*UO*NMSI(N)*NMSJ(N)*NMSK(N))
   4 FORMAT (3X,13,4X,2(3(lX,13),4X),F5.1)
   52 CONTINUE
50 PRINT *, 'DO YOU WISH TO CHANGE THE NUMBER OF SOURCES?'
   READ(5,*) ANS
   IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
      PRINT *, 'ENTER NUMBER OF SOURCES:'
      READ(5,*) NMS
   ENDIF
   IF(NMS.EQ.0) GOTO 54
   PRINT *,
   'DO YOU WISH TO CHANGE THE DATA FOR THE MASS SOURCES?'
   READ(5,*) ANS
   IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
      DO 56 N=1,NMS
         PRINT *
         PRINT *, 'FOR MASS SOURCE NUMBER ',N,' ENTER'
         PRINT *, 'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
         READ(5,*) IMSB(N),JMSB(N),KMSB(N)
         IMSB(N)=IMSB(N)+2
         JMSB(N)=JMSB(N)+2
         KMSB(N)=KMSB(N)+2
      56 CONTINUE
   ENDIF
PRINT *,
'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
READ(5,*) NMSI(N),NMSJ(N),NMSK(N)
PRINT *, 'THE FLOW RATE OF THE MASS SOURCE IN CFM'
READ(5,*) RMS(N)
RMS(N)=RMS(N)/(60.*H**2*UO*NMSI(N)*NMSJ(N)*NMSK(N))
56 CONTINUE
ENDIF
C *** UPDATE THERMOCOUPLE DATA
54 PRINT *
PRINT *, 'CURRENT DATA:
PRINT *
PRINT *, 'NUMBER OF THERMOCOUPLES: ',NTHCO
IF(NTHCO.EQ.0) GOTO 60
PRINT *, 'TC LOCATION'
PRINT *, 'NO. X Y Z'
DO 62 N=1,NTHCO
PRINT ('(1X,12,2X,3(2X,F8.4))'), N,CX(N)*H,CY(N)*H,CZ(N)*H
62 CONTINUE
60 PRINT *, 'DO YOU WISH TO CHANGE THE NUMBER OF THERMOCOUPLES?
READ(5,*) ANS
IF(INDEX(ANS,'Y').NE.O.OR.INDEX(ANS,'y').NE.0) THEN
PRINT *, 'ENTER THE NUMBER OF THERMOCOUPLES:
READ(5,*) NTHCO
ENDIF
IF(NTHCO.EQ.0) GOTO 70
IF(INDEX(ANS,'Y').NE.O.OR.INDEX(ANS,'y').NE.0) GOTO 65
PRINT *, 'DO YOU WISH TO CHANGE THE THERMOCOUPLE LOCATIONS?
READ(5,*) ANS
IF(INDEX(ANS,'Y').NE.O.OR.INDEX(ANS,'y').NE.0) GOTO 65
GOTO 70
65 DO 66 N=1,NTHCO
PRINT *, 'ENTER THE LOCATION (FEET) OF THERMOCOUPLE ',N
READ(5,*) XCX,XCY,XCZ
CX(N)=XCX/H
CY(N)=XCY/H
CZ(N)=XCZ/H
66 CONTINUE
C *** UPDATE HEAT SOURCE DATA
70 PRINT *
PRINT *, 'CURRENT DATA'
PRINT *
PRINT *, 'HEAT SOURCE LOCATION'
PRINT *, 'X DIRECTION: NODE',NHSZ(1,1)-2,' TO',NHSZ(1,2)-2
PRINT *, 'Y DIRECTION: NODE',NHSZ(2,1)-2,' TO',NHSZ(2,2)-2
PRINT *, 'Z DIRECTION: NODE',NHSZ(3,1)-2,' TO',NHSZ(3,2)-2
PRINT *
PRINT *, 'DO YOU WISH TO CHANGE THE STARTING COORDINATES?
READ(5,*) ANS
IF(INDEX(ANS,'Y').NE.O.OR.INDEX(ANS,'y').NE.0) THEN
PRINT *, 'ENTER THE STARTING COORDINATES:
READ(5,*) NHSZ(1,1)
PRINT *, 'Y DIRECTION:'
READ(5,*) NHSZ(2,1)
PRINT *, 'Z DIRECTION:'
READ(5,*) NHSZ(3,1)
DO 71 I=1,3
   NHSZ(I,1)=NHSZ(I,1)+2
CONTINUE
ENDIF
PRINT *, 'DO YOU WISH TO CHANGE THE ENDING COORDINATES?'
READ(5,*) ANS
IF(INDEX(ANS, 'Y').NE.0.OR.INDEX(ANS, 'y').NE.0) THEN
   PRINT *, 'ENTER THE ENDING COORDINATES:'
   PRINT *, 'X DIRECTION:'
   READ(5,*) NHSZ(1,2)
   PRINT *, 'Y DIRECTION:'
   READ(5,*) NHSZ(2,2)
   PRINT *, 'Z DIRECTION:'
   READ(5,*) NHSZ(3,2)
DO 72 I=1,3
   NHSZ(I,2)=NHSZ(I,2)+2
CONTINUE
ENDIF
C *** UPDATE AMBIENT TEMPERATURE
PRINT *
PRINT *, 'CURRENT DATA'
PRINT *
PRINT 3,
   'AMBIENT TEMPERATURE =',TA-459.67,'DEGREES FARENHEIT'
3 FORMAT (1X,A,1X,F6.2,1X,A)
PRINT *
PRINT *, 'DO YOU WISH TO CHANGE THE AMBIENT TEMPERATURE?'
READ(5,*) ANS
IF(INDEX(ANS, 'Y').NE.0.OR.INDEX(ANS, 'y').NE.0) THEN
   PRINT *, 'ENTER THE NEW AMBIENT TEMPERATURE IN DEGREES F:'
   READ(5,*) TAF
   TA=TAF+459.67
ENDIF
C *** UPDATE TIME DATA
PRINT *
PRINT *, 'CURRENT DATA'
PRINT *
PRINT *, 'MAXIMUM RUN TIME =',TMAX*H/U0,' SECONDS'
PRINT *, 'INCREMENTAL TIME STEP =',DTIME*H/U0,' SECONDS'
PRINT *, 'TIME BETWEEN DATA OUTPUT =',TWRITE,' SECONDS'
PRINT *, 'TIME BETWEEN PLOTS =',TTAPE,' SECONDS'
PRINT *, 'FIRE START TIME =',HSTART,' SECONDS'
PRINT *
PRINT *, 'DO YOU WISH TO CHANGE ANY OF THESE TIMES?'
READ(5,*) ANS
IF(INDEX(ANS, 'Y').NE.0.OR.INDEX(ANS, 'y').NE.0) THEN
   PRINT *, 'ENTER MAXIMUM RUN TIME IN SECONDS:'
   READ(5,*) XTMAX
   TMAX=XTMAX*U0/H
   PRINT *, 'INCREMENTAL TIME STEP IN SECONDS:'
READ(5,*), XDTIME
DTIME=XDTIME*UO/H
PRINT *, 'TIME BETWEEN DATA OUTPUT IN SECONDS:'
READ(5,*), TWRITE
PRINT *, 'TIME BETWEEN PLOTS IN SECONDS:'
READ(5,*), TTAE
PRINT *, 'FIRE START TIME IN SECONDS:'
READ(5,*), HSTART
WWRP=INT(TWRITE/XDTIME)/2
ENDIF
ELSE
BEGIN
PRINT *, 'PROGRAM TERMINATING.'
PRINT *, 'PLEASE RENAME FIRE.DAT AND TRY AGAIN.'
GOTO 999
ENDIF

***** CREATING NEW DATA FILE

ELSE
PRINT *, 'FIRE DATA BI NOT FOUND!'
PRINT *, 'CREATING NEW DATA FILE.'
OPEN(10,FILE='/FIRE DATA BI',STATUS='NEW')

***** INPUT NEW DATA *****

C *** INPUT GEOMETRIC DATA
PRINT *, 'INPUT COMPARTMENT LENGTH (X DIRECTION) IN FEET'
READ(5,*), X
PRINT *, 'INPUT COMPARTMENT HEIGHT (Z DIRECTION) IN FEET'
READ(5,*), H
PRINT *, 'INPUT COMPARTMENT WIDTH (Y DIRECTION) IN FEET'
READ(5,*), Y
PRINT *, 'INPUT COMPARTMENT WALL THICKNESS IN INCHES'
READ(5,*), TWAL
PRINT *, 'INPUT COMPARTMENT FLOOR/CEILING THICKNESS IN INCHES'
READ(5,*), TFLR
PRINT *, 'INPUT NUMBER OF CELLS IN THE X DIRECTION:'
READ(5,*), NI
PRINT *, 'INPUT NUMBER OF CELLS IN THE Y DIRECTION:'
READ(5,*), NJ
PRINT *, 'INPUT NUMBER OF CELLS IN THE Z DIRECTION:'
READ(5,*), NK

C *** INPUT INTERNAL SOLIDS DATA
PRINT *, 'INPUT NUMBER OF INTERNAL SOLID PIECES'
READ(5,*), NCHIP
IF (NCHIP.GT.0) THEN
PRINT *, 'INPUT THE LOCATION OF INTERNAL SOLID PIECES'
DO 14 N=1,NCHIP
PRINT *
PRINT *, 'FOR SOLID PIECE NUMBER ',N,' ENTER THE'
PRINT *, 'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
READ(5,*), ICHPB(N),JCHPB(N),KCHPB(N)
ICHPB(N)=ICHPB(N)+2
JCHPB(N)=JCHPB(N)+2
KCHPB(N)=KCHPB(N)+2
14 CONTINUE
PRINT *, 'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
READ(5,*) NCHPI(N),NCHPJ(N),NCHPK(N)
PRINT *, 'THERMAL CONDUCTIVITY'
READ(5,*) CONS(N)
PRINT *, 'SPECIFIC HEAT'
READ(5,*) CPS(N)
PRINT *, 'AND FAN VELOCITY (0 IF NO FAN)'
READ(5,*) WFAN(N)
CONTINUE
ENDIF
C *** INPUT MASS SOURCE DATA
PRINT *, 'INPUT NUMBER OF MASS SOURCES'
READ(5,*) NMS
IF (NMS.EQ.0) THEN
PRINT *
PRINT *, 'ENTER MASS SOURCE DATA:'
DO 18 N=1,NMS
PRINT *, 'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
READ(5,*) IMSB(N),JMSB(N),KMSB(N)
IMSB(N)=IMSB(N)+2
JMSB(N)=JMSB(N)+2
KMSB(N)=KMSB(N)+2
PRINT *, 'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
READ(5,*) NMSI(N),NMSJ(N),NMSK(N)
PRINT *, 'THE FLOW RATE OF THE MASS SOURCE IN CFM'
READ(5,*) RMS(N)
RMS(N)=RMS(N)/(60.*H**2*UO*NMSI(N)*NMSJ(N)*NMSK(N))
CONTINUE
ENDIF
C *** INPUT THERMOCOUPLE DATA
PRINT *, 'INPUT NUMBER OF THERMOCOUPLES'
READ(5,*) NTHCO
PRINT *, 'INPUT THERMOCOUPLE POSITION IN FEET:'
DO 19 I=1,NTHCO
PRINT *
PRINT *, 'FOR THERMOCOUPLE NUMBER ',I
PRINT *, ' X POSITION (FEET):'
READ (5,*) XCX
CX(I)=XCX/H
PRINT *, ' Y POSITION (FEET):'
READ (5,*) XCY
CY(I)=XCY/H
PRINT *, ' Z POSITION (FEET):'
READ (5,*) XCZ
CZ(I)=XCZ/H
CONTINUE
C *** INPUT HEAT SOURCE DATA
PRINT *, 'INPUT FIRST NODE OF HEAT SOURCE'
PRINT *, ' X DIRECTION:'
READ (5,*) NHSZ(1,1)
PRINT *, ' Y DIRECTION:'
READ (5,*) NHSZ(2,1)
PRINT *, ' Z DIRECTION:'

READ (5,*) NHSZ(3,1)
PRINT *, 'INPUT LAST NODE OF HEAT SOURCE IN EACH DIRECTION:'
PRINT *, 'X DIRECTION:'
READ (5,*) NHSZ(1,2)
PRINT *, 'Y DIRECTION:'
READ (5,*) NHSZ(2,2)
PRINT *, 'Z DIRECTION:'
READ (5,*) NHSZ(3,2)

C *** CORRECT HEAT SOURCE LOCATION DUE TO EXTERNAL CELLS
DO 5 I=1,3
   DO 6 J=1,2
      NHSZ(I,J)=NHSZ(I,J)+2
   6 CONTINUE
5 CONTINUE

C *** INPUT AMBIENT TEMPERATURE
PRINT *, 'INPUT AMBIENT TEMPERATURE (DEGREES FARENHEIT)'
READ(5,*) TAF
TA=TAF+459.67

C *** INPUT TIME DATA
PRINT *, 'INPUT MAX RUN TIME FOR FIRE (SECONDS)'
READ(5,*) XTMAX
TMAX=XTMAX*UO/H
PRINT *, 'INPUT SIZE OF TIME STEP (SECONDS)'
READ(5,*) XDTIME
DTIME=XDTIME*UO/H
PRINT *, 'INPUT FIRE START TIME (SECONDS)'
READ(5,*) HSTART
PRINT *, 'INPUT TIME INTERVAL BETWEEN DATA SAVES (SECONDS)'
READ(5,*) TWRITE
PRINT *, 'INPUT TIME INTERVAL BETWEEN PLOTS (SECONDS)'
READ(5,*) TTAPE

C *** DETERMINE NWRP AND ADJUST TWRITE
NWRP=INT(TWRITE/XDTIME)/2
ENDIF

C *** SAVE DATA IN DATA FILE
REWIND(10)
WRITE(10,*) X,Y,H,TFLR,TWAL,TA
WRITE(10,*) NI,NJ,NK
WRITE(10,*) NCHIP,NMS,NWRP,NTHCO
WRITE(10,*) TMAX,DTIME,TTAPE,TWRITE,HSTART
WRITE(10,*) NHSZ(1,1),NHSZ(1,2),NHSZ(2,1),NHSZ(2,2),NHSZ(3,1),
& NHSZ(3,2)
IF (NCHIP.EQ.0) GOTO 20
DO 22 N=1,NCHIP
   WRITE(10,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
& NCHPK(N),CPS(N),CONS(N),WFAN(N)
22 CONTINUE

20 IF (NMS.EQ.0) GOTO 24
DO 26 N=1,NMS
   WRITE(10,*) IMSB(N),NMSI(N),JMSB(N),NMSJ(N),KMSB(N),
& NMSK(N),RMS(N)
26 CONTINUE
26 CONTINUE
24 DO 28 I=1,NTHCO
     WRITE(10,*) CX(I),CY(I),CZ(I)
28 CONTINUE
     REWIND(10)
     CLOSE(10)
999 END
APPENDIX B. PROGRAM FIRE

PROGRAM FIRE

*************************************************************************
**
** THREE-DIMENSIONAL NUMERICAL SIMULATION
**
** OF A FIRE SPREAD INSIDE A BUILDING
**
** DEVELOPED BY:
**
** H.Q. YANG AND K.T. YANG
**
** DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING
**
** UNIVERSITY OF NOTRE DAME
**
** NOTRE DAME, INDIANA, 46556
**
**
** DEC. 1986
**
** AMENDED BY
**
** K.J. THORKILDSEN
**
** LIEUTENANT, U.S. COAST GUARD
**
** DEPARTMENT OF MECHANICAL ENGINEERING
**
** NAVAL POSTGRADUATE SCHOOL
**
** MONTEREY, CALIFORNIA 93942
**
** SEP. 1992
**
**
*************************************************************************

*SET CONSTANTS:
*  CPO : REFERENCE SPECIFIC HEAT OF AIR =
*  GC : GRAVITATIONAL ACCELERATION = 32.17 FT/SEC**2
*  RAIR : UNIVERSAL GAS CONSTANT FOR AIR = 53.34
*  RHO0 : REFERENCE AIR DENSITY (LBM/FT**3) = 0.0714 LBM/FT**3
*  U0 : REFERENCE VELOCITY (FT/SEC) = 1.0 FT/SEC
*
*COMPARTMENT DIMENSIONS (IN FEET):
*  H : HEIGHT IN Z-DIRECTION (USED AS REFERENCE LENGTH)
*  X : LENGTH IN X-DIRECTION
*  Y : WIDTH IN Y-DIRECTION
*
*  NI : NUMBER OF CELLS IN X-DIRECTION
*  NJ : Y-DIRECTION
*  NK : Z-DIRECTION
*
*  CONSRA : TA**3/(RA*CP*U0*H*H)
*  HCONV : HEAT TRANSFER COEFFICIENT TO THE AMBIENT (BTU/H*K*FT**2)
*  TA : REFERENCE TEMPERATURE (R)
*  TINIT : INITIAL TEMPERATURE (0)
*  UR : REFERENCE VELOCITY (CM/S)
*
*HEAT SOURCE DATA:
*  NHSZ(1,1) : STARTING CONTROL VOLUME NUMBER IN X-DIRECTION
*  NHSZ(2,1) : Y-DIRECTION
* NHSZ(3,1) : Z-DIRECTION
* NHSZ(1,2) : LAST CONTROL VOLUME NUMBER IN X-DIRECTION
* NHSZ(2,2) : Y-DIRECTION
* NHSZ(3,2) : Z-DIRECTION
*
*INTERNAL SOLID PIECES:
* NCHIP : NUMBER OF INTERNAL SOLID PIECES
* ICHPB() : STARTING NODE NUMBER FOR SOLID IN X-DIRECTION
* JCHPB() : Y-DIRECTION
* KCHPB() : Z-DIRECTION
* NCHPI() : NUMBER OF NODES OF SOLID IN X-DIRECTION
* NCHPJ() : Y-DIRECTION
* NCHPK() : Z-DIRECTION
*
*TOTAL HEAT:
* QSIN : INPUT FROM THE FIRE
* QSWAL : LOST TO THE WALL (FROM AIR TO THE WALL)
* QSFAN : CARRIED AWAY BY THE VENTILATION
*
*VIEW FACTORS FROM HEAT SOURCE:
* VFHSW(N,J,K) : TO ELEMENT J,K ON WEST WALL
* VFHSE(N,J,K) : EAST WALL
* VFHSN(N,K,I) : TO ELEMENT K,I ON NORTH WALL
* VFHSS(N,K,I) : SOUTH WALL
* VFHSF(N,I,J) : TO ELEMENT I,J ON FRONT WALL
* VFHSB(N,I,J) : BACK WALL

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

*DATA FILES USED IN THIS PROGRAM:
*
* FILE # 10 = FIRE DATA : INITIAL SET-UP DATA
* 11 = CONTINUE DATA : RESTART/CONTINUATION DATA
* 12 = OUTPUT DATA : OUTPUT RESULTS
* 13 = PLOT DATA : DATA FOR PLOTTING

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

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(25),YC(25),ZC(25),XS(25),YS(25),ZS(25),DXXC(25),
& DYYC(25),DZZC(25),DXXX(25),DYYY(25),DZZZ(25)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWAL
COMMON/BL3/F,FR,HSTART
COMMON/BL7/N1,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL12/NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
COMMON/BL16/UO,UGRT,BUOY,CP0,PRD,COND0,VIS0,RHOO,
& TA,DTEMP,TWRITE,TTAPE,TEMP,GC,RAIR,NT
COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
& SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15),
& SIG14(25,25,15),SIG24(25,25,15),SIG34(25,25,15)
COMMON/BL23/RMS(20),NMS,IMSB(20),NMSI(20),JMSB(20),NMSJ(20),
& KMSB(20),NMSK(20)
COMMON/BL26/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
& JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
& COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
& WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
& U(25,25,15), V(25,25,15), W(25,25,15)
COMMON/BL33/TPD(25,25,15), RPD(25,25,15), PPD(25,25,15),
& CPD(25,25,15), UPD(25,25,15), VPD(25,25,15),
& WPD(25,25,15)
COMMON/BL34/HEIGHT(25,25,15), REQ(25,25,15), SMP(25,25,15),
& SMP(25,25,15), PP(25,25,15), DU(25,25,15),
& DV(25,25,15), DW(25,25,15)
COMMON/BL36/AREA(25,25,15), AE(25,25,15), AW(25,25,15), AN(25,25,15),
& AS(25,25,15), AF(25,25,15), SP(25,25,15),
& SU(25,25,15), RI(25,25,15)
COMMON/BL37/VMAX(25,25,15), VMIN(25,25,15), RESORM(40),
& CP(25,25,15), NCH(3,2), NOD(25,25,15)
COMMON/BL38/TCOUP(30), CX(30), CY(30), CZ(30), NTH(30,3), NTHCO
COMMON/BL39/ALEW, CONRA, QSIN, QSWER, QSWAL, QSAR, QSFAN
COMMON/BL40/VFHSW(5,25,15), VFHSE(5,25,15), VFHSS(5,15,25),
COMMON/BL41/VFHSBW(5,8,34,34), VFHSBE(5,8,34,34), VFHSBS(5,8,34,34),
& VFHSBN(5,8,34,34), VFHSBB(5,8,34,34), VFHSBF(5,8,34,34)
COMMON/BL43/QSCONF, QSCONB, QSCONE, QSCONW, QSCONN, QSCONS, QSRADF,
& QSRADB, QSRADW, QSRAIR, QSRADS, WAIR, WWAL, WINS,
& WERR, WWFAN
DATA SORMAX, XTIME, ITMAX/3.00, 0.0, 0.0, 4/

C ***** INITIAL PROGRAM START *****
CALL CPUSR(BEGIN, IPR)

C *** INPUT DATA
CALL INPUT(NSTOP)
IF(NSTOP.GT.0) GOTO 9999

C *** GENERATE GRID SYSTEM
CALL GRID

C *** INITIALIZE THE ALL DATA FIELDS
CALL INIT

C *** OPEN OUTPUT FILES
OPEN(12,FILE='OUTPUT DATA B1', STATUS='UNKNOWN')
OPEN(13,FILE='PLOT DATA B4', STATUS='UNKNOWN',
& FORM='UNFORMATTED')

C *** CALCULATE THE VIEW FACTORS FROM THE FIRE TO THE WALLS
CALL VIEW

***** START CALCULATIONS *****

NT=0
NTIM=0
300 NT=NT+1

C *** ON RESTART NTMAX0 IS SET EQUAL TO OLD VALUE FOR NTREAL
IF(TIME.GE.TMAX) GO TO 277
NTREAL=NT+NTMAX0
TIME=TIME+DTIME
XTIME=TIME*H/U0
C PRINT 3,'CURRENT FIRE TIME IS:',XTIME,'SECONDS'
C 3 FORMAT (1X,A,1X,F10.6,1X,A)
C *** CALCULATE THE HEAT SOURCE IN BTU/SEC
CALL CALQ
C *** START CALCULATIONS
ITER=0
JTERM=0
JJTERM=0
C *** PREDICT VARIABLE FIELDS FOR USE BY CALVIS AND SU(I,J,K)
DO 48 K=1,NK+4
   DO 47 J=1,NJ+4
      DO 48 I=1,NI+4
         TPD(I,J,K)=T(I,J,K)
         CPD(I,J,K)=C(I,J,K)
         RPD(I,J,K)=R(I,J,K)
         UPD(I,J,K)=U(I,J,K)
         VPD(I,J,K)=V(I,J,K)
         WPD(I,J,K)=W(I,J,K)
48 CONTINUE
47 JTERM=JTERM+1
301 NTITER=0
312 NTITER=NTITER+1
C *** IF FIRE HAS STARTED, CALCULATE THE TEMPERATURE
   IF (XTIME.GE.HSTART) CALL CALT
   *****THIS STEP CAN BE SKIPPED WHEN COMPARTMENT IS OPEN TO OUTSIDE*****
C *** CORRECT GLOBAL PRESSURE FOR TOTAL MASS CONSERVATION
CALL GLOBE
C *** CALCULATE DENSITY
DO 100 J=1,NJ+4
   DO 100 I=1,NI+4
      DO 100 K=1,NK+4
         IF (NOD(I,J,K).EQ.1) GOTO 100
         AAAA=BUOY*UGRT*HEIGHT(I,J,K)
         R(I,J,K)=(UGRT*P(I,J,K)+(1./EXP(AAAA)))/T(I,J,K)
100 CONTINUE
   *****THIS STEP CAN BE SKIPPED WHEN COMPARTMENT IS OPEN TO OUTSIDE*****
C *** ITERATE INSIDE TEMPERATURE LOOP TO ASSURE GLOBAL CONSERVATION
   IF (NTITER.LT.2) GOTO 312
C *** PRINT OUT THE ENERGY DISTRIBUTION
   * IF (MOD(NTREAL,NWRP).EQ.0) CALL OUT(4)
C *** CALCULATE THE SMOKE CONCENTRATION
   CALL CALC
C *** CALCULATE TURBULENT VISCOSITY AND CONDUCTIVITY
   CALL CALVIS

64
C *** CORRECT CONDUCTIVITY OF THE SOLID
IF (NCHIP.NE.0) CALL SOLCON

C *** START PRESSURE CORRECTION ITERATIVE LOOP
C *** IT IS THE MAJOR PART OF THE ERROR CONTROL ROUTINE
ITER=ITER+1

C *** CALCULATE THE STRESS AND VELOCITY COMPONENTS U,V,AND W
CALL STRESS
CALL CALU
CALL CALV
CALL CALW

C *** CALCULATE PRESSURE
CALL CALP

C *** IF SOURCE TERM IS LARGER THAN 10.0, STOP PROGRAM
IF (RESORM(ITER).GT.10.0) GOTO 2020

    IF(RESORM(ITER).LE.SORMAX) GO TO 49
    IF(ITER.EQ.1) GO TO 302
    IF(RESORM(ITER) .LE. RESORM(ITER-1)) GO TO 302
GO TO 304

302 IF(JTERM .LT. 2) THEN
    SOURCE=RESORM(ITER)
ELSEIF(RESORM(ITER).LE.SOURCE) THEN
    SOURCE=RESORM(ITER)
ELSE
    GOTO 304
ENDIF

DO 23 K=1,NK+4
DO 23 J=1,NJ+4
DO 23 I=1,NI+4
    TPD(I,J,K)=T(I,J,K)
    CPD(I,J,K)=C(I,J,K)
    RPD(I,J,K)=R(I,J,K)
    UPD(I,J,K)=U(I,J,K)
    VPD(I,J,K)=V(I,J,K)
    WPD(I,J,K)=W(I,J,K)
    PPD(I,J,K)=P(I,J,K)
23 CONTINUE

JJTERM=O
IF(ITER.EQ.ITMAX) GO TO 49
IF(JTERM.EQ.2) GO TO 35
IF(ITER.EQ.4) GO TO 47
35 IF(JTERM.EQ.3) GO TO 58
IF(ITER.EQ.7) GO TO 47
58 JJTERM=O
GO TO 301

304 JJTERM=JJTERM+1
IF(JTERM.EQ.1) GOTO 41
IF(JTERM.EQ.2.AND.JJTERM.EQ.1.AND.ITER.NE.5) GO TO 41
GO TO 82
41 DO 40 K=1,NK+4
   DO 40 J=1,NJ+4
   DO 40 I=1,NI+4
      R(I,J,K)=RPD(I,J,K)
      U(I,J,K)=UPD(I,J,K)
      V(I,J,K)=VPD(I,J,K)
      W(I,J,K)=WPD(I,J,K)
      P(I,J,K)=PPD(I,J,K)
   40 CONTINUE
   IF(ITER.EQ.ITMAX) GO TO 49
   GO TO 47

82 DO 43 K=1,NK+4
   DO 43 J=1,NJ+4
   DO 43 I=1,NI+4
      C(I,J,K)=CPD(I,J,K)
      R(I,J,K)=RPD(I,J,K)
      U(I,J,K)=UPD(I,J,K)
      V(I,J,K)=VPD(I,J,K)
      W(I,J,K)=WPD(I,J,K)
      P(I,J,K)=PPD(I,J,K)
   43 CONTINUE
   IF(ITER.EQ.ITMAX) GO TO 49
   IF((JTERM.EQ.3.AND.ITER.NE.8).OR.JTERM.EQ.2) GO TO 49
   GO TO 301

49 ITERT=ITERT+ITER
   IF (MOD(NTREAL,NWRP).EQ.0) CALL OUT(1)
   C *** FIND TEMPERATURES AT THERMOCOUPLES AND PRINT OUT AT PROPER TIME
   CALL TCP
   IF (MOD(NTREAL,NWRITE).EQ.0) CALL OUT(2)
   C *** OUTPUT FILED VALUES
   IF (MOD(NTREAL,NWRITE).EQ.0) CALL OUT(3)
   IF(TIME.GE.TMAX) GO TO 277
   C *** SHIFT CURRENT TIME VALUES TO PREVIOUS TIME VALUES AND
   C *** LOOP BACK FOR NEXT ITERATION
   DO 305 K=1,NK+4
   DO 305 J=1,NJ+4
   DO 305 I=1,NI+4
      TOD(I,J,K)=T(I,J,K)
      COD(I,J,K)=C(I,J,K)
      ROD(I,J,K)=R(I,J,K)
      UOD(I,J,K)=U(I,J,K)
      VOD(I,J,K)=V(I,J,K)
      WOD(I,J,K)=W(I,J,K)
      POD(I,J,K)=P(I,J,K)
   305 CONTINUE
   C *** OUTPUT TO DATA FILE FOR PLOTTING
   IF(MOD(NTREAL,NTAPE).EQ.0) THEN
      WRITE(13) TIME,T,U,V,W
ENDIF

C *** OUTPUT TO CONTINUATION FILE FOR RESTART
IF(MOD(NTREAL,100).EQ.0) THEN
  WRITE(11) TIME,NTREAL,FR,T,R,U,V,W,P,C
  REWIND 11
ENDIF
GO TO 300

C *** OUTPUT TO CONTINUATION FILE
277 WRITE(11) TIME,NTREAL,FR,T,R,U,V,W,P,C
GO TO 9999

2020 WRITE(12,*)'RESIDUAL MASS IS LARGER THAN 10.0',
& ' PROGRAM STOPS AT TIME = ',XTIME,' SEC'

9999 CALL CPUTIME(END,IPR)
WRITE (12,*) 'CPU RUN TIME = ',(END-BEGIN)*1.E-6,' SECONDS'
STOP
END

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

BLOCK DATA
******************************************************************************

* UO : REFERENCE VELOCITY = 1.0 FT/SEC
* PRT : TURBULENT PRANDTL NUMBER = 1.0
* RHOO : REFERENCE DENSITY OF AIR = 0.0714 LBM/FT**3
* CP0 : REFERENCE SPECIFIC HEAT OF AIR = 0.24 BTU/(LBM*F)
* VIS0 : REFERENCE VISCOSITY = 1.56E-4
* CNT :
* ABTURB : TURBULENCE CONSTANT
* BTURB : TURBULENCE CONSTANT
* GC : GRAVITATIONAL ACCELERATION = 32.17 FT/SEC**2
* RAIR : GAS CONSTANT FOR AIR = 53.34
* ALEW : LEWIS NUMBER = 1.0
******************************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/BL12/NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
COMMON/BL16/UO,UGRT,BUOY,CP0,PRT,COND0,VISO,RHOO,
& TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN

C *** SPECIFY THE INITIAL DATA
DATA U0, PRT, RHOO , CP0, VIS0 , NTMAX0/
& 1.0, 1.0, 0.0714, 0.24, 1.56D-4, 0/
DATA CNT,ABTURB,BTURB/0.2,2.0,1.0/
DATA GC,RAIR,ALEW/32.17,53.34,1.0/
END

******************************************************************************
******************************************************************************
SUBROUTINE CALC
********************************************************************
*THIS SUBROUTINE CALCULATES THE SMOKE CONCENTRATIONS

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
&  DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWL
COMMON/BL7/NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
&  C0D(25,25,15),VOD(25,25,15),VOD(25,25,15),
&  WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
&  U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
&  CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
&  WPD(25,25,15)
COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
&  AS(25,25,15),AF(25,25,15),AR(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
&  CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
COMMON/BL39/ALEW,CONSRA,QQIN,QQER,QQWAL,QQAIR,QQSFAN

C *** CALCULATE COEFFICIENTS
DO 100 K=2,NK+3
  DO 100 J=2,NJ+3
    DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
  DXP1=DXXC(I+1)
  DXI =DXXC(I)
  DXM1=DXXC(I-1)

  DYP1=DYYC(J+1)
  DYJ =DYYC(J)
  DYM1=DYYC(J-1)

  DZP1=DZZC(K+1)
  DZK =DZZC(K)
  DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
  DXN=DXXC(I)
  DXS=DXXC(I)
  DXF=DXXC(I)
  DXB=DXXC(I)

  DYF=DYYC(J)
  DYB=DYYC(J)
  DYE=DYYC(J)
  DYW=DYYC(J)

  DZE=DZZC(K)
  DZW=DZZC(K)

68
DZN=DZZC(K)
DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
DXE=DXX(I+2)
DXE =DXX(I+1)
DXW =DXX(I)
DXWW=DXX(I-1)

DYN=DYYS(J+2)
DYN =DYYS(J+1)
DYS =DYYS(J)
DYSS=DYYS(J-1)

DZFF=DZZS(K+2)
DZF =DZZS(K+1)
DZB =DZZS(K)
DZBB=DZZS(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
DXYF=DXF*DYF
DXYB=DXB*DYB
DYZE=DYE*DZE
DYZW=DYW*DZW
DZXN=DZN*DXN
DZXS=DZS*DXS

VOL=DXI*DYJ*DZK
VOLDT=VOL/DTIME

ZXOYN=DZXN/DYN
ZXOYS=DZXS/DYS
XYOZF=DXYF/DZF
XYOZB=DXYB/DZB
Y2OXE=DY2E/DXE
Y2OXW=DY2W/DXW

C *** DENSITY AT THE SURFACES OF THE CONTROL VOLUME
GN=(R(I,J,K)*DYP1+R(I,J+1,K))*DYJ/(DYP1+DYJ)
GS=(R(I,J,K)*DYMI+R(I,J-1,K))*DYJ/(DYMI+DYJ)
GE=(R(I,J,K)*DXP1+R(I+1,J,K))*DXI/(DXP1+DXI)
GW=(R(I,J,K)*DXM1+R(I-1,J,K))*DXI/(DXM1+DXI)
GF=(R(I,J,K)*DZP1+R(I,J,K+1))*DZK/(DZP1+DZK)
GB=(R(I,J,K)*DZM1+R(I,J,K-1))*DZK/(DZM1+DZK)

CN=GN*V(I,J+1,K)*DZK
CS=GS*V(I,J,K)*DZS
CE=GE*U(I+1,J,K)*DYE
CW=GW*U(I,J,K)*DY2B
CF=GF*W(I,J,K+1)*DXYF
CB=GB*W(I,J,K)*DXYB

C *** DIFFUSIVITY AT THE SURFACES OF THE CONTROL VOLUME
CONDN=(DYP1+DYJ)/(DYJ/COND(I,J,K)+DYP1/COND(I,J+1,K))
CONDS=(DYMI+DYJ)/(DYJ/COND(I,J,K)+DYMI/COND(I,J-1,K))
CONDE=(DXP1+DXI)/(DXI/COND(I,J,K)+DXP1/COND(I+1,J,K))
CONDW = (DXM1 + DXI) / (DXI / COND(I, J, K) + DXM1 / COND(I-1, J, K))
CONDN = (DZP1 + DZK) / (DZK / COND(I, J, K) + DZP1 / COND(I, J, K+1))
CONDB = (DZM1 + DZK) / (DZK / COND(1, J, K) + DZM1 / COND(1, J, K-1))

COND1 = ZXOYN * CONDN * ALEW
COND2 = ZXOYS * CONDS * ALEW
CONDE = YZOXE * CONDE * ALEW
CONDW = YZOXM * CONDW * ALEW

C *** QUICK SCHEME
CEP = (ABS(CE) + CE) * DXP1 * DXI / (DXE * (DXE + DXW)) / 8.
CEM = (ABS(CE) - CE) * DXP1 * DXI / (DXE * (DXE + DXEE)) / 8.
CWP = (ABS(CW) + CW) * DXM1 * DXI / (DXW * (DXW + DXWW)) / 8.
CWM = (ABS(CW) - CW) * DXM1 * DXI / (DXW * (DXW + DXE)) / 8.

CNP = (ABS(CN) + CN) * DYP1 * DYL / (DYN * (DYN + DYS)) / 8.
CNM = (ABS(CN) - CN) * DYP1 * DYL / (DYN * (DYN + DYN)) / 8.
CSP = (ABS(CS) + CS) * DYM1 * DYL / (DYS * (DYS + DYS)) / 8.
CSM = (ABS(CS) - CS) * DYM1 * DYL / (DYS * (DYS + DYN)) / 8.

CFP = (ABS(CF) + CF) * DZP1 * DZK / (DZF * (DZF + DZB)) / 8.
CFM = (ABS(CF) - CF) * DZP1 * DZK / (DZF * (DZF + DZF)) / 8.
CBP = (ABS(CB) + CB) * DZM1 * DZK / (DZB * (DZB + DZBB)) / 8.
CBM = (ABS(CB) - CB) * DZM1 * DZK / (DZB * (DZB + DZF)) / 8.

AE(I, J, K) = -.5 * CE * DXI / DXE + CEP + CEM * (1. + DXE / DXEE) + CWM * DXW / DXE
AW(I, J, K) = .5 * CW * DXI / DXW + CWP * (1. + DXW) + CEP * DXI / DXW
AN(I, J, K) = -.5 * CN * DYL / DYN + CNM + CNM * (1. + DYN / DYN) + CSM * DYS / DYN
AS(I, J, K) = -.5 * CS * DYL / DYS + CSP * (1. + DYS / DYS) + CSM * DYN / DYN

AF(I, J, K) = -.5 * CF * DZK / DZF + CFP + CFM * (1. + DZF / DZF) + CBM * DZB / DZF
AB(I, J, K) = .5 * CB * DZK / DZB + CBM * CBP * (1. + DZB / DZBB) + CFP * DZF / DZF

C *** BOUNDARY CONSIDERATION
IF (I.LT.NI+3)
  AEE = - CEM * DXE / DXEE
  AEE = AEE * CPD(I+2, J, K)
ELSE
  AEE = 0.
  AEE = 0.
ENDIF

IF (I.GT.2)
  AWW = - CWP * DXW / DXWW
  AWW = AWW * CPD(I-2, J, K)
ELSE
  AWW = 0.
  AWW = 0.
ENDIF

IF (J.LT.NJ+3)
  ANN = - CNM * DYN / DYN
  ANN = ANN * CPD(I, J+2, K)
ELSE
  ANN = 0.

70
ANNR=0.
ENDIF

IF (J.GT.2) THEN
  ASS=-CSP*DYS/DYSS
  ASSR=ASS*CPD(I,J-2,K)
ELSE
  ASS=0.
  ASSR=0.
ENDIF

IF (K.LT.NK+3) THEN
  AFF=-CFM*DZF/DZFF
  AFFR=AFF*CPD(I,J,K+2)
ELSE
  AFF=0.
  AFFR=0.
ENDIF

IF (K.GT.2) THEN
  ABB=-CBP*DZB/DZBB
  ABBR=ABB*CPD(I,J,K-2)
ELSE
  ABB=0.
  ABBR=0.
ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
IF (NOD(I-1,J,K).NE.0) THEN
  AWW=0.0
  AWWR=0.0
ENDIF

IF (NOD(I+1,J,K).NE.0) THEN
  AEE=0.0
  AEER=0.0
ENDIF

IF (NOD(I,J-1,K).NE.0) THEN
  ASS=0.0
  ASSR=0.0
ENDIF

IF (NOD(I,J+1,K).NE.0) THEN
  ANN=0.0
  ANN=0.0
ENDIF

IF (NOD(I,J,K-1).NE.0) THEN
  ABB=0.0
  ABBR=0.0
ENDIF

IF (NOD(I,J,K+1).NE.0) THEN
  AFF=0.0
  AFFR=0.0
ENDIF
ENDIF

\[
AP(I,J,K) = AE(I,J,K) + AW(I,J,K) + AN(I,J,K) + AS(I,J,K) + AF(I,J,K) + \\
& \quad AB(I,J,K) + AEE + AWW + ANN + ASS + AFF + ABB + CONDE1 + CONDW1 + \\
& \quad CONDN1 + CONDS1 + CONDF1 + CONDB1
\]

AE(I,J,K) = AE(I,J,K) + CONDE1

AW(I,J,K) = AW(I,J,K) + CONDW1

AN(I,J,K) = AN(I,J,K) + CONDN1

AS(I,J,K) = AS(I,J,K) + CONDS1

AF(I,J,K) = AF(I,J,K) + CONDF1

AB(I,J,K) = AB(I,J,K) + CONDB1

SP(I,J,K) = -ROD(I,J,K) * VOLDT

SU(I,J,K) = -SP(I,J,K) * COD(I,J,K) + AEER + AWWR + ANNR + ASSR + AFFR + ABBR

100 CONTINUE

C *** TAKE CARE OF B.C. THRU AN, AS, AE, AW, AF, AB, SP AND SU

C *** Y DIRECTION
DO 500 I = 2, NI + 3
DO 500 K = 2, NK + 3
SP(I,3,J,K) = SP(I,3,J,K) + AS(I,3,J,K)
SP(I,NJ+2,J,K) = SP(I,NJ+2,J,K) + AN(I,NJ+2,J,K)
AS(I,3,J,K) = 0.
AN(I,NJ+2,J,K) = 0.
500 CONTINUE

C *** X DIRECTION
DO 600 J = 2, NJ + 3
DO 600 K = 2, NK + 3
SP(3,J,K) = SP(3,J,K) + AW(3,J,K)
SP(NI+2,J,K) = SP(NI+2,J,K) + AE(NI+2,J,K)
AW(3,J,K) = 0.0
AE(NI+2,J,K) = 0.0
600 CONTINUE

C *** Z DIRECTION
DO 700 I = 2, NI + 3
DO 700 J = 2, NJ + 3
SP(I,J,3) = SP(I,J,3) + AB(I,J,3)
SP(I,J,NK+2) = SP(I,J,NK+2) + AF(I,J,NK+2)
AB(I,J,3) = 0.
AF(I,J,NK+2) = 0.
700 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
DO 300 K = 2, NK + 3
DO 300 J = 2, NJ + 3
DO 300 I = 2, NI + 3
AP(I,J,K) = AP(I,J,K) - SP(I,J,K)
300 CONTINUE

C *** VOLUMETRIC MASS SOURCE INPUT
VOLT = 0.0
DO 113 I = 2, NI + 3
DO 113 J=2,NJ+3
DO 113 K=2,NK+3
DXI =DXXC(I)
DYJ =DYYC(J)
DZK =DZZC(K)
VOL =DXI*DYJ*DZK*H**3
VOLT=VOLT+VOL
113 CONTINUE

DO 111 I=NHSZ(1,1),NHSZ(1,2)
DO 111 J=NHSZ(2,1),NHSZ(2,2)
DO 111 K=NHSZ(3,1),NHSZ(3,2)
DXI =DXXC(I)
DYJ =DYYC(J)
DZK =DZZC(K)
VOL =DXI*DYJ*DZK
SU(I,J,K)=SU(I,J,K)+VOL*H/(UO*RHO0*VOLT)
111 CONTINUE

C *** SOLVE FOR C
CALL TRID (3,3,3,NI+2,NJ+2,NK+2,C)

C *** Z DIRECTION
DO 74 I=1,NI+4
DO 74 J=1,NJ+4
C(I,J,2 )=C(I,J,3 )
C(I,J,1 )=C(I,J,2 )
C(I,J,NK+3)=C(I,J,NK+2)
C(I,J,NK+4)=C(I,J,NK+3)
74 CONTINUE

C *** Y DIRECTION
DO 84 I=2,NI+3
DO 84 K=2,NK+3
C(I,NJ+3,K)=C(I,NJ+4,K)
C(I,NJ+4,K)=C(I,NJ+3,K)
C(I,2 ,K)=C(I, 3,K)
C(I,1 ,K)=C(I, 2,K)
84 CONTINUE

C *** X DIRECTION
DO 80 J=1,NJ+4
DO 80 K=1,NK+4
C(2 ,J,K)=C( 3,J,K)
C(1 ,J,K)=C( 2,J,K)
C(NI+3,J,K)=C(NI+2,J,K)
C(NI+4,J,K)=C(NI+3,J,K)
80 CONTINUE

RETURN
END
SUBROUTINE CALP

*CALCULATES NODE Pressures

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
 & DYYC(40),DZZC(40),DXXS(40),DYS(40),DZSS(40)

COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL7/N1,NJ,NK,KNUN,NELEOR,NWRP
COMMON/BL12/NWRITE,NTAPE,NTMAXO,NTREAL,TSUM,ITER
COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
 & JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
COMMON/BL23/RMS(20),NMS,IMSB(20),NMS1(20),JMSB(20),NMSJ(20),
 & KMSB(20),NMSK(20)
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
 & COD(25,25,15),VOD(25,25,15),
 & WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
 & U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
 & SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
 & DV(25,25,15),DW(25,25,15)
COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
 & AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
 & SU(25,25,15),RI(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
 & CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
DO 100 K=2,NK+3
DO 100 J=2,NJ+3
DO 100 I=2,NI+3

IF (NOD(I,J,K).EQ.1) GOTO 100

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
DXP1=DXXC(I+1)
DXI =DXXC(I)
DXM1=DXXC(I-1)

DYP1=DYYC(J+1)
DYJ =DYYC(J)
DYM1=DYYC(J-1)

DZP1=DZZC(K+1)
DZK =DZZC(K)
DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
DXN=DXXC(I)
DXS=DXXC(I)
DXF=DXXC(I)
DXB=DXXC(I)

DYN=DYYC(J)
DYS=DYYC(J)
DYE=DYYC(J)
DYW=DYYC(J)

DZE=DZZC(K)
DZW=DZZC(K)
DZN=DZZC(K)
DZS=DZZC(K)

C *** DEFINE AREA OF THE CONTROL VOLUME
DXYF=DXF*DYF
DXYB=DXB*DYB
DYZE=DYE*DZE
DYZW=DYW*DZW
DZNX=DZN*DXN
DZXS=DZS*DXS

VOL=DXI*DYJ*DZK
VOLDT=VOL/DTIME

C *** DENSITY AT THE SURFACES
RN=(R(I,J,K)*DYP1+R(I,J+1,K))*DYJ/(DYP1+DYJ)
RS=(R(I,J,K)*DYM1+R(I,J-1,K))*DYJ/(DYM1+DYJ)
RE=(R(I,J,K)*DXP1+R(I+1,J,K))*DXI/(DXP1+DXI)
RW=(R(I,J,K)*DXM1+R(I-1,J,K))*DXI/(DXM1+DXI)
RF=(R(I,J,K)*DZP1+R(I,J+1,K+1))*DZK/(DZP1+DZK)
RB=(R(I,J,K)*DZM1+R(I,J,K-1))*DZK/(DZM1+DZK)

AN(I,J,K)=RN*DZXN*DV(I,J+1,K)
AS(I,J,K)=RS*DZXS*DV(I,J,K)
AE(I,J,K)=RE*DU(I+1,J,K)
AW(I,J,K)=RW*DU(I,J,K)
AF(I,J,K)=RF*DXYF*DW(I,J,K+1)
AB(I,J,K)=RB*DXYB*DW(I,J,K)

CN=RN*V(I,J+1,K)*DZXN
CS=RS*V(I,J,K)*DZXS
CE=RE*U(I+1,J,K)*DYZE
CW=RW*U(I,J,K)*DYZW
CF=RF*W(I,J,K+1)*DXYF
CB=RB*W(I,J,K)*DXYB

SMP(I,J,K)=-(R(I,J,K)-ROD(I,J,K))*VOLDT-CE+CW-CN+CS-CF+CB
SU(I,J,K)=SMP(I,J,K)
SP(I,J,K)=0.

100 CONTINUE

C *** CONSIDER THE MASS SOURCE INPUT INTO THE CONTROL VOLUME
IF (NMS. GE. 1) THEN
DO 150 M=1,NMS
IB=IMSB(M)
JE=IB+NMSI(M)-1
JB=JMSB(M)
JE=JB+NMSJ(M)-1
KB=KMSB(M)
KE=KB+NMSK(M)-1
DO 160 I=IB,IE-1
DO 160 J=JB,JE-1
DO 160 K=KB,KE-1
SU(I,J,K)=SU(I,J,K)+RMS(M)
160 CONTINUE
150 CONTINUE
ENDIF

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** X DIRECTION
DO 500 K=2,NK+3
DO 500 I=2,NI+3
AS(I,2,K)=0.
AN(I,NJ+3,K)=0.
500 CONTINUE

C *** Y DIRECTION
DO 501 K=2,NK+3
DO 501 J=2,NJ+3
AW(2,J,K)=0.
AE(NI+3,J,K)=0.
501 CONTINUE

C *** Z DIRECTION
DO 502 I=2,NI+3
DO 502 J=2,NJ+3
AB(I,J,2)=0.
AF(I,J,NK+3)=0.
502 CONTINUE

C *** MODIFICATION FOR DECK BOUNDARIES
IF (NCHIP.EQ.0) GOTO 110
DO 101 N=1,NCHIP
  IB =ICHPB(N)
  IE =IB+NCHIP(I)-1
  JB =JCHPB(N)
  JE =JB+NCHIP(J)-1
  KB =KCHPB(N)
  KE =KB+NCHIP(K)-1

DO 102 J=JB,JE-1
DO 102 K=KB,KE-1
AE(IB-1,J,K)=0.0
AW(IE,J,K)=0.0
102 CONTINUE

DO 103 I=IB,IE-1
DO 103 K=KB,KE-1
AN(I,JB-1,K)=0.0
AS(I,JE,K)=0.0
103 CONTINUE

DO 106 I=IB,IE-1
DO 106 J=JB,JE-1
AF(I,J,KB-1)=0.0
AB(I,J,KE)=0.0
106 CONTINUE
CONTINUE

C *** FOR THE CELLS INSIDE OF THE DECKS
DO 104 I=IB,IE-1
   DO 104 J=JB,JE-1
      DO 104 K=KB,KE-1
         SP(I,J,K)=-1.0E2
         AW(I,J,K)=0.
         AE(I,J,K)=0.
         AS(I,J,K)=0.
         AN(I,J,K)=0.
         AB(I,J,K)=0.
         AF(I,J,K)=0.
         SU(I,J,K)=0.
      CONTINUE
   CONTINUE
104 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
110 DO 300 I=2,NI+3
    DO 300 J=2,NJ+3
       DO 300 K=2,NK+3
            & +AF(I,J,K)+AB(I,J,K)
    CONTINUE
300 CONTINUE

C *** SOLUTION OF FINITE DIFFERENCE EQUATION
   CALL TRID (3,3,3,NI+2,NJ+2,NK+2,PP)

C *** CORRECTION FOR VELOCITY U
   DO 600 I=3,NI+3
      DO 600 J=2,NJ+3
         DO 600 K=2,NK+3
       CONTINUE
600 CONTINUE

C *** CORRECTION FOR VELOCITY V
   DO 603 J=3,NJ+3
      DO 603 K=2,NK+3
         DO 603 I=2,NI+3
      CONTINUE
603 CONTINUE

C *** CORRECTION FOR VELOCITY W
   DO 604 K=3,NK+3
      DO 604 I=2,NI+3
         DO 604 J=2,NJ+3
       CONTINUE
604 CONTINUE

C *** CORRECTION FOR PRESSURE P
   DO 606 J=1,NJ+4
      DO 606 I=1,NI+4
         DO 606 K=1,NK+4
            P(I,J,K)=P(I,J,K)+PP(I,J,K)
            PP(I,J,K)=0.
       CONTINUE
606 CONTINUE
C *** RESET THE VELOCITY INSIDE OF DECK
IF (NCHIP.EQ.0) GOTO 121
DO 120 N=1,NCHIP
   IB=ICHPB(N)
   IE=IB+NCHPI(N)-1
   JB=JCHPB(N)
   JE=JB+NCHPJ(N)-1
   KB=KCHPB(N)
   KE=KB+NCHPK(N)-1
   DO 109 I=IB,IE
      DO 109 J=JB,JE
         DO 109 K=KB,KE-1
            U(I,J,K)=0.0
   109 CONTINUE
   DO 118 I=IB,IE-1
      DO 118 J=JB,JE
         DO 118 K=KB,KE-1
            V(I,J,K)=0.0
   118 CONTINUE
   DO 119 I=IB,IE-1
      DO 119 J=JB,JE-1
         DO 119 K=KB,KE
            W(I,J,K)=WFAN(N)
   119 CONTINUE
120 CONTINUE

C *** RECALCULATE THE ERROR SOURCE AFTER CORRECTIONS OF U, V, P
121 SORSUM=0.
   RESORM(ITER)=0.
   DO 700 J=2,NJ+3
      DO 700 I=2,NI+3
         DO 700 K=2,NK+3
            IF (NOD(I,J,K).NE.1) THEN
C ** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
      DXP1=DXXC(I+1)
      DXI =DXXC(I)
      DXM1=DXXC(I-1)
      DYP1=DYYC(J+1)
      DYJ =DYYC(J)
      DYM1=DYYC(J-1)
      DZP1=DZZC(K+1)
      DZK =DZZC(K)
      DZM1=DZZC(K-1)
C ** SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=DXXC(I)
      DXS=DXXC(I)
      DXF=DXXC(I)
   78
DYF=DYYC(J)
DYB=DYYC(J)
DYE=DYYC(J)
DYW=DYYC(J)
DZE=DZZC(K)
DZW=DZZC(K)
DZN=DZZC(K)
DZS=DZZC(K)

C *** DEFINE AREA OF THE CONTROL VOLUME
DXYF=DXF*DYF
DXYB=DXB*DYB
DYZE=DYE*DZE
DYZW=DYW*DZW
DZXS=DZS*DXS

C *** CALCULATE DENSITY
RN=(R(I,J,K)*DYP1+R(I,J+1,K))*DYJ)/(DYP1+DYJ)
RS=(R(I,J,K)*DYM1+R(I,J-1,K))*DYJ)/(DYM1+DYJ)
RE=(R(I,J,K)*DXP1+R(I+1,J,K))*DXI)/(DXP1+DXI)
RW=(R(I,J,K)*DXM1+R(I-1,J,K))*DXI)/(DXM1+DXI)
RF=(R(I,J,K)*DZP1+R(I,J+1,K))*DZK)/(DZP1+DZK)
RB=(R(I,J,K)*DZM1+R(I,J-1,K))*DZK)/(DZM1+DZK)

C *** SORSUM IS ACTUAL MASS INCREASE OR DECREASE FROM CONTINUITY
C *** EQUATION, THIS WILL BE COMPARED TO MASS SOURCE
C *** CONSIDER THE MASS SOURCE INPUT INTO THE CONTROL VOLUME
IF (NMS.GT.0) THEN
  DO 250 M=1,NMS
    IB=IMSB(M)
    IE=IB+NMSI(M)-1
    JB=JMSB(M)
    JE=JB+NMSJ(M)-1
    KB=KMSB(M)
    KE=KB+NMSK(M)-1
    DO 260 II=IB,IE-1
      DO 260 JJ=JB,JE-1
        DO 260 KK=KB,KE-1
          IF (((II.EQ.I).AND.(JJ.EQ.J)).AND.(KK.EQ.K)) THEN
            SMP(I,J,K)=SMP(I,J,K)+RMS(M)
          END IF
  260 CONTINUE
ENDIF

260 CONTINUE
CONTINUE
ENDIF
SORSUM=SORSUM+SMP(I,J,K)

C *** RESORM IS SUM OF THE ABSOLUTE VALUE OF SMP(I,J,K)
    RESORM(ITER)=RESORM(ITER)+ABS(SMP(I,J,K))
ENDIF

700 CONTINUE
RETURN
END

************************************************************************************
************************************************************************************
SUBROUTINE CALQ
************************************************************************************
*
* VARIABLES:
* BR   = MAXIMUM BURN RATE (LBM/SEC)
* F    = MAXIMUM FUEL AVAILABLE (LBM)
* FR   = TOTAL FUEL REMAINING (LBM)
* H    = REFERENCE LENGTH (FT)
* HC   = HEAT OF COMBUSTION (BTU/LBM)
* HSTART= FIRE START TIME (SECONDS)
* Q    = TOTAL HEAT INPUT (BTU/SEC)
* TIME = NONDIMENSIONAL FIRE TIME
* UO   = REFERENCE VELOCITY (FT/SEC)
* XTIME = FIRE TIME (SECONDS)
*
************************************************************************************
*
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWAL
COMMON/BL3/F,FR,HSTART
COMMON/BL12/NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
COMMON/BL16/UO,UGRT,BUOY,CPO,PRT,CONDO,VISO,RHOO,
& TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT

XTIME=TIME*H/UO
HC=2600.0
BR= 0.01

C *** CALCULATE HEAT RELEASE RATE (Q) IN BTU/SEC
C *** NOTE: THESE ALGORITHMS ASSUME A LINEAR INCREASE IN BOTH
C *** HEAT RELEASE AND FUEL CONSUMPTION OVER THE FIRST
C *** TWO SECONDS OF FIRE TIME, AFTER WHICH BOTH ARE AT
C *** MAXIMUM
IF(XTIME.LT.HSTART) THEN
Q=0.0
FR=F
ELSEIF(XTIME.GE.HSTART.AND.(XTIME-HSTART).LE.2.0) THEN
    IF(FR.LE.0.0) THEN
        Q =0.0
        FR=0.0
        ELSE
Q = HC*BR*(XTIME-HSTART)/2.
FR = F-BR*(XTIME-HSTART)**2/2.
ENDIF
ELSEIF((XTIME-HSTART).GT.2.0) THEN
IF(FR.LE.0.0) THEN
Q = 0.0
FR = 0.0
ELSE
Q = HC*BR
FR = F-BR*(XTIME-HSTART)
ENDIF
ENDIF

C *** TAKE RADIATION HEAT FLUX INTO ACCOUNT
Q = Q-QR
IF (Q.LE.0.0) Q = 0.
RETURN
END

**********************************************************************
**********************************************************************
SUBROUTINE CALT
**********************************************************************
**********************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
& DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWAL
COMMON/BL7/NI,NJ,NK,KRUN,NBIOR,NWRP
COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
COMMON/BL16/UO,UGRT,BUOY,CPO,PRT,CONDO,VISO,RHOO,
& TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
COMMON/BL23/RMS(20),NMS,IMS,(20),NMSI(20),JMSB(20),NMSJ(20),
& KMSB(20),NMSK(20)
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
& COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
& WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
& U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL33/TPD(25,25,15),RDPD(25,25,15),PD(25,25,15),
& CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
& WPD(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMPP(25,25,15),
& SMPP(25,25,15),DP(25,25,15),DU(25,25,15),
& DV(25,25,15)
COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
& AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
& SU(25,25,15),RI(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
& CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
COMMON/BL39/ALEW,CONSRA,QSIN,QR,QRWAL,QSAR,QSFAN
COMMON/BL43/QSCONQ,QSCONB,QSCONE,QSCONW,QSCONN,QSCONI,
& QSRADF,GSRADQ,GSRADG,GSRADW,GSRADN,GSRADS,
& WAIR,WWAL,WINS,WERL,WWFA
C *** NONDIMENSIONAL REFERENCE TEMPERATURE
   TINF=TA/TA

C *** CALCULATE COEFFICIENTS
   DO 100 K=2,NK+3
   DO 100 J=2,NJ+3
   DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE TEMPERATURE CONTROL VOLUME
   DXP1=DXXC(I+1)
   DXI =DXXC(I)
   DXM1=DXXC(I-1)
   DYP1=DYYC(J+1)
   DYJ =DYYC(J)
   DYM1=DYYC(J-1)
   DZP1=DZZC(K+1)
   DZK =DZZC(K)
   DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
   DXN=DXXC(I)
   DXS=DXXC(I)
   DXF=DXXC(I)
   DXB=DXXC(I)
   DYF=DYYC(J)
   DYB=DYYC(J)
   DYE=DYYC(J)
   DYW=DYYC(J)
   DZE=DZZC(K)
   DZW=DZZC(K)
   DZN=DZZC(K)
   DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR TEMPERATURE
   DXEE=DXXS(I+2)
   DXE =DXXS(I+1)
   DXW =DXXS(I)
   DXWW=DXXS(I-1)
   DYN=DYYS(J+2)
   DYN =DYYS(J+1)
   DYS =DYYS(J)
   DYSS=DYYS(J-1)
   DZF=DZZS(K+2)
   DZF =DZZS(K+1)
   DZB =DZZS(K)
   DZBB=DZZS(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
   DXYF=DXF*DYF
\[ DXYB = DXB \times DYB \]
\[ DYZE = DYE \times DZE \]
\[ DYZW = DYW \times DZW \]
\[ DZXN = DZN \times DXN \]
\[ DZXS = DZS \times DXS \]
\[ VOL = DXI \times DYJ \times DZK \]
\[ VOLDT = VOL / DTIME \]

C *** FOR CONDUCTION
\[ ZXOYN = DZXN / DYN \]
\[ ZXOYS = DZXS / DYS \]
\[ XYOZF = DXYF / DZF \]
\[ XYOZB = DXYB / DZB \]
\[ YZOXE = DYZE / DXE \]
\[ YZOXW = DYZW / DXW \]

C *** DENSITY AT THE SURFACES
\[ GN = (R(I,J,K) \times DYP1 + R(I,J+1,K) \times DYJ) / (DYJ \times DYP1 + DYP1 \times R(I,J+1,K)) \]
\[ GS = (R(I,J,K) \times DYM1 + R(I,J-1,K) \times DYJ) / (DYJ \times DYM1 + DYM1 \times R(I,J-1,K)) \]
\[ GE = (R(I,J,K) \times DXP1 + R(I+1,J,K) \times DXI) / (DXI \times DXP1 + DXP1 \times R(I+1,J,K)) \]
\[ GW = (R(I,J,K) \times DXM1 + R(I-1,J,K) \times DXI) / (DXI \times DXM1 + DXM1 \times R(I-1,J,K)) \]
\[ GF = (R(I,J,K) \times DZP1 + R(I,J,K+1) \times DZK) / (DZK \times DZP1 + DZP1 \times R(I,J,K+1)) \]
\[ GB = (R(I,J,K) \times DZM1 + R(I,J,K-1) \times DZK) / (DZK \times DZM1 + DZM1 \times R(I,J,K-1)) \]

C *** THE MASS FLUX RATE THROUGH THE SURFACES
\[ CN = GN \times V(I,J+1,K) \times DZXN \]
\[ CS = GS \times V(I,J,K) \times DZXS \]
\[ CE = GE \times U(I+1,J,K) \times DYZE \]
\[ CW = GW \times U(I,J,K) \times DYZW \]
\[ CF = GF \times W(I,J,K+1) \times DXYF \]
\[ CB = GB \times W(I,J,K) \times DXYB \]

C *** CONDUCTIVITY AT THE SURFACES
\[ CONDN = (DYP1 \times DYJ) \times COND(I,J,K) \times COND(I,J+1,K) \times DYJ \times DYP1 / \]
\[ & (DYJ \times COND(I,J,K) + DYP1 \times COND(I,J+1,K)) \]
\[ CONDS = (DYM1 \times DYJ) \times COND(I,J,K) \times COND(I,J-1,K) \times DYJ \times DYM1 / \]
\[ & (DYJ \times COND(I,J,K) + DYM1 \times COND(I,J-1,K)) \]
\[ CONDE = (DXP1 \times DXI) \times COND(I,J,K) \times COND(I+1,J,K) \times DXI \times DXP1 / \]
\[ & (DXI \times COND(I,J,K) + DXP1 \times COND(I+1,J,K)) \]
\[ CONDW = (DXM1 \times DXI) \times COND(I,J,K) \times COND(I-1,J,K) \times DXI \times DXM1 / \]
\[ & (DXI \times COND(I,J,K) + DXM1 \times COND(I-1,J,K)) \]
\[ CONDF = (DZP1 \times DZK) \times COND(I,J,K) \times COND(I,J,K+1) \times DZK \times DZP1 / \]
\[ & (DZK \times COND(I,J,K) + DZP1 \times COND(I,J,K+1)) \]
\[ CONDB = (DZM1 \times DZK) \times COND(I,J,K) \times COND(I,J,K-1) \times DZK \times DZM1 / \]
\[ & (DZK \times COND(I,J,K) + DZM1 \times COND(I,J,K-1)) \]

C *** CONDUCTION COMPONENT
\[ COND1 = ZXOYN \times CONDN \]
\[ CONDS1 = ZXOYS \times CONDS \]
\[ CONDE1 = YZOXF \times CONDE \]
\[ CONDF1 = YZOXW \times CONDW \]
\[ CONDB1 = XYOZB \times CONDB \]

C *** QUICK SCHEME
CEP = (ABS(CE) + CE) * DXP1 * DXI / (DXE * (DXE + DXW) * 8.)
CEM = (ABS(CE) - CE) * DXP1 * DXI / (DXE * (DXE + DXEE) * 8.)
CWF = (ABS(CW) + CW) * DXM1 * DXI / (DXW * (DXW + DXWW) * 8.)
CWM = (ABS(CW) - CW) * DXM1 * DXI / (DXW * (DXW + DXW) * 8.)
CNP = (ABS(CN) + CN) * DYP1 * DYJ / (DYN * (DYN + DYS) * 8.)
CNM = (ABS(CN) - CN) * DYP1 * DYJ / (DYN * (DYN + DYNN) * 8.)
CWP = (ABS(CS) + CS) * DYM1 * DYJ / (DYS * (DYS + DYSS) * 8.)
CSM = (ABS(CS) - CS) * DYM1 * DYJ / (DYS * (DYS + DYN) * 8.)
CFP = (ABS(CF) + CF) * DZP1 * DZK / (DZF * (DZF + DZB) * 8.)
CFM = (ABS(CF) - CF) * DZP1 * DZK / (DZF * (DZF + DZFF) * 8.)
CBP = (ABS(CB) + CB) * DZM1 * DZK / (DZB * (DZB + DZBB) * 8.)
CBM = (ABS(CB) - CB) * DZM1 * DZK / (DZB * (DZB + DZF) * 8.)

AE(I, J, K) = -.5 * CE * DXI / DXE + CEP + CEM * (1 + DXE / DXEE) + CWM * DXW / DXE
AW(I, J, K) = .5 * CW * DXI / DXW + CWM + CWP * (1 + DXW / DXWW) + CEP * DXW / DXE
AN(I, J, K) = -.5 * CN * DYJ / DYN + CNP + CNM * (1 + DYN / DYNN) + CSM * DYS / DYN
AS(I, J, K) = .5 * CS * DYJ / DYS + CSM + CSM * (1 + DYS / DYSS) + CSM * DYN / DYS
AF(I, J, K) = -.5 * CF * DZK / DZF + CFP + CFM * (1 + DZF / DZFF) + CBM * DZB / DZF
AB(I, J, K) = .5 * CB * DZK / DZB + CBM + CBP * (1 + DZB / DZBB) + CFP * DZF / DZB

C *** BOUNDARY CONSIDERATIONS

IF (I.LT.NI+3) THEN
  AEE = -CEM * DXE / DXEE
  AEER = AEE * TPD(I+2,J,K) * CPM(I+2,J,K)
ELSE
  AEE = 0.
  AEER = 0.
ENDIF

IF (I.GT.2) THEN
  AWW = -CWP * DXW / DXWW
  AWWR = AWW * TPD(I-2,J,K) * CPM(I-2,J,K)
ELSE
  AWW = 0.
  AWWR = 0.
ENDIF

IF (J.LT.NJ+3) THEN
  ANN = -CNM * DYN / DYNN
  ANNR = ANN * TPD(I,J+2,K) * CPM(I,J+2,K)
ELSE
  ANN = 0.
  ANNR = 0.
ENDIF

IF (J.GT.2) THEN
  ASS = -CSP * DYS / DYSS
  ASSR = ASS * TPD(I,J-2,K) * CPM(I,J-2,K)
ELSE
  ASS = 0.
  ASSR = 0.
ENDIF

IF (K.LT.NK+3) THEN
AFF = -CFM*DFZ/DFZF
AFFR = AFF*TPD(I,J,K+2)*CPM(I,J,K+2)
ELSE
    AFF = 0.
    AFFR = 0.
ENDIF

IF (K.GT.2) THEN
    ABB = -CBP*DBZ/DBZB
    ABBR = ABB*TPD(I,J,K-2)*CPM(I,J,K-2)
ELSE
    ABB = 0.
    ABBR = 0.
ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES

IF (NOD(I-1,J,K).NE.0) THEN
    AWW = 0.0
    AWWR = 0.0
ENDIF

IF (NOD(I+1,J,K).NE.0) THEN
    AEE = 0.0
    AEER = 0.0
ENDIF

IF (NOD(I,J-1,K).NE.0) THEN
    ASS = 0.0
    ASSR = 0.0
ENDIF

IF (NOD(I,J+1,K).NE.0) THEN
    ANN = 0.0
    ANN = 0.0
ENDIF

IF (NOD(I,J,K-1).NE.0) THEN
    ABB = 0.0
    ABBR = 0.0
ENDIF

IF (NOD(I,J,K+1).NE.0) THEN
    AFF = 0.0
    AFFR = 0.0
ENDIF

             AB(I,J,K) + AEE + AWW + ANN + ASS + AFF + ABB) * CPM(I,J,K) +
             CONDE1 + CONDW1 + COND1 + CONDS1 + CONDF1 + CONDB1

AE(I,J,K) = AE(I,J,K) * CPM(I+1,J,K) + CONDE1
AW(I,J,K) = AW(I,J,K) * CPM(I-1,J,K) + CONDW1
AN(I,J,K) = AN(I,J,K) * CPM(I,J+1,K) + COND1
AS(I,J,K) = AS(I,J,K) * CPM(I,J-1,K) + CONDS1
AF(I,J,K) = AF(I,J,K) * CPM(I,J,K+1) + CONDF1
AB(I,J,K) = AB(I,J,K) * CPM(I,J,K-1) + CONDB1
\[ \text{SP}(I,J,K) = -\text{ROD}(I,J,K) \times \text{VOLDT} \times \text{CPM}(I,J,K) \]

\[ \text{SU}(I,J,K) = -\text{SP}(I,J,K) \times \text{TOD}(I,J,K) + \text{AEER} + \text{AWWR} + \text{ANNR} + \text{ASSR} + \text{AFFR} + \text{ABBR} \]

```
C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** Y-DIRECTION
DO 500 I=3,NI+2
   DO 500 K=3,NK+2
      SU(I,3,K) = SU(I,3,K) + AS(I,3,K) \times T(I,2,K)
      SU(I,NJ+2,K) = SU(I,NJ+2,K) + AN(I,NJ+2,K) \times T(I,NJ+3,K)
      AS(I,3,K) = 0.
      AN(I,NJ+2,K) = 0.
   500 CONTINUE
C *** X-DIRECTION
DO 600 J=3,NJ+2
   DO 600 K=3,NK+2
      SU(3,J,K) = SU(3,J,K) + AW(3,J,K) \times T(2,J,K)
      SU(NI+2,J,K) = SU(NI+2,J,K) + AE(NI+2,J,K) \times T(NI+3,J,K)
      AW(3,J,K) = 0.0
      AE(NI+2,J,K) = 0.0
   600 CONTINUE
C *** Z-DIRECTION
DO 700 I=3,NI+2
   DO 700 J=3,NJ+2
      SU(I,J,3) = SU(I,J,3) + AB(I,J,3) \times T(I,J,2)
      SU(I,J,NK+2) = SU(I,J,NK+2) + AF(I,J,NK+2) \times T(I,J,NK+3)
      AB(I,J,3) = 0.
      AF(I,J,NK+2) = 0.
   700 CONTINUE
C *** CONSIDER THE MASS SOURCE INPUT TO THE CONTROL VOLUME
IF (NMS.GE.1) THEN
   DO 150 M=1,NMS
      IB = IMSB(M)
      IE = IB + NMSI(M) - 1
      JB = JMSB(M)
      JE = JB + NMSJ(M) - 1
      KB = KMSB(M)
      KE = KB + NMSK(M) - 1
      DO 160 I=IB,IE-1
         DO 160 J=JB,JE-1
            DO 160 K=KB,KE-1
               IF (RMS(M).GE.0.0) THEN
                  RMSCPT = RMS(M) \times 1.0 \times CPM(I,J,K)
               ELSE
                  RMSCPT = RMS(M) \times T(I,J,K) \times CPM(I,J,K) \times R(I,J,K)
               ENDIF
   160 CONTINUE
   150 CONTINUE
ENDIF
C *** CONSIDER THE RADIATION HEAT FLUX FROM THE FIRE TO THE BLOCK
CALL RADHT(2)
```
C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
DO 300 K=3,NK+2
DO 300 J=3,NJ+2
DO 300 I=3,NI+2
   AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
300 CONTINUE

C *** VOLUME HEAT SOURCE INPUT
C *** CALCULATE THE TOTAL VOLUME OCCUPIED BY HEAT SOURCE
C *** DISTRIBUTE ENERGY INTO EACH CONTROL VOLUME
C *** QQ/H**3 DIMENSIONLESS HEAT SOURCE
   VOLT=0.0
   DO 113 I=NHSZ(1,1),NHSZ(1,2)
   DO 113 J=NHSZ(2,1),NHSZ(2,2)
   DO 113 K=NHSZ(3,1),NHSZ(3,2)
      QQQ=Q*H/(UO*CPO*RHOO*TA)
      VOL=DXXC(I)*DYYC(J)*DZZC(K)
      VOLT=VOLT+VOL*H**3
      SU(I,J,K)=SU(I,J,K)+VOL*QQQ/VOLT
113 CONTINUE

C *** SOLVE FOR T
   CALL TRID (3,3,3,NI+2,NJ+2,NK+2,T)
   DO 2001 I=1,NI+4
   DO 2001 J=1,NJ+4
   DO 2001 K=1,NK+4
      IF(T(I,J,K).LT.TCOOL) T(I,J,K)=TCOOL
2001 CONTINUE

C *** CALCULATE RADIATION HEAT TRANSFER
C *** HERE SU(I,J,K) IS USED TO STORE THE RADIATIVE HEAT FLUX
   DO 75 I=1,NI+4
   DO 75 J=1,NJ+4
   DO 75 K=1,NK+4
      SU(I,J,K)=0.
75 CONTINUE

C *** CONSIDER THE RADIATION HEAT FLUX FROM THE FIRE TO THE WALL
   CALL RADHT (1)

C *** SUMMATION OF CONDUCTION HEAT FLUX AND RADIATION HEAT FLUX TO WALLS
   QSCONF=0.
   QSCONB=0.
   QSCONE=0.
   QSCONW=0.
   QSCONN=0.
   QSCONS=0.
   QSRADF=0.
   QSRADB=0.
   QSRADE=0.
QSRADW=0.  
QSRADN=0.  
QSRADS=0.  

C *** CALCULATE CONDUCTION, RADIATION & TEMPERATURE ON THE SOLID WALLS  
DO 74 I=3,NI+2  
DO 74 J=3,NJ+2  

C *** ON THE BACK WALL  
DZK =DZCZ(2)  
DZP1=DZCZ(3)  
DXI =DXC(I)  
DYJ =DYYC(J)  
DXY =DXI*DYJ  
VOL =DXY*DZK  
CONDF=(DZP1+DZK)*DZK*DZP1*COND(I,J,2)*COND(I,J,3)/  
      & (DZK*COND(I,J,2)+DZP1*COND(I,J,3))  
QCONF=DXY*CONDF*(T(I,J,3)-T(I,J,2))*2.0/(DZP1+DZK)  
QCONB=DXY*COND(I,J,2)*(T(I,J,1)-T(I,J,2))*2.0/DZK  
QRADB=SU(I,J,2)  
T(I,J,2)=TOD(I,J,2)+DTIME*(QCONF+QCONB+QRADB)/(VOL*CPN(I,J,2))  
& T(I,J,1)=(2.*COND(I,J,2)*T(I,J,2)+HCOEF*TINF*DZK)/  
      & (HCOEF*DZK+2.*COND(I,J,2))  
QSCONB=QSCONB+QCONF  
QSRADB=QSRADB+QRADB  

C *** ON THE FRONT WALL  
DZK =DZCZ(NK+3)  
DZM1=DZCZ(NK+2)  
DXI =DXC(I)  
DYJ =DYYC(J)  
DXY =DXI*DYJ  
VOL =DXY*DZK  
CONDF=(DZM1+DZK)*DZK*DZM1*COND(I,J,NK+3)*COND(I,J,NK+2)/  
      & (DZK*COND(I,J,NK+3)+DZM1*COND(I,J,NK+2))  
QCONF=DXY*CONDF*(T(I,J,NK+3)-T(I,J,NK+2))*2.0/(DZK+DZM1)  
QCONB=DXY*COND(I,J,NK+3)*(T(I,J,NK+4)-T(I,J,NK+3))*2.0/DZK  
QRADB=SU(I,J,NK+3)  
T(I,J,NK+3)=TOD(I,J,NK+3)+DTIME*(QCONF+QCONB+QRADB)/(VOL*CPM(I,J,NK+3))  
& T(I,J,NK+4)=(2.*COND(I,J,NK+3)*T(I,J,NK+3)+HCOEF*TINF*DZK)/  
      & (HCOEF*DZK+2.0*COND(I,J,NK+3))  
QSCONF=QSCONF+QCONF  
QSRADF=QSRADF+QRADF  

74 CONTINUE  

DO 84 I=3,NI+2  
DO 84 K=3,NK+2  

C *** ON THE SOUTH WALL  
DYJ =DYYC(2)
DYP1 = DYYC(3)
DXI = DXXC(I)
DZK = DZZC(K)
DZX = DZK * DXI
VOL = DZX * DYJ

CONDN = (DYP1 + DYJ) * DYJ * DYP1 * COND(I, 2, K) * COND(I, 3, K) /
(DYJ * COND(I, 2, K) + DYP1 * COND(I, 3, K))

QCONN = DZX * COND(I, 2, K) * (T(I, 1, K) - T(I, 2, K)) * 2.0 / (DYP1 + DYJ)
QCONS = DZX * COND(I, 2, K) * (T(I, 1, K) - T(I, 2, K)) * 2.0 / DYJ
QRADS = SU(I, 1, 2, K)

T(I, 1, K) = TOD(I, 1, 2, K) + DTIME * (QCONN + QCONS + QRADS) /
(VOL * CPM(I, 1, 2, K))

QSCONS = QSCONS + QCONN
QSRADS = QSRADS + QRADS

C *** ON THE NORTH WALL
DZK = DZZC(NJ + 2)
DZK = DZK + DZK
DZX = DZK + DXI
VOL = DZX * DYJ

COND = (DZK + DXI) * DYJ * DZK * COND(I, NJ + 2, K) /
(DX1 + DZK)

QCONS = DZK * COND(I, NJ + 2, K) * (T(I, NJ + 2, K) - T(I, NJ + 3, K)) * 2.0 / (DZK + DXI)
QCONN = DZK * COND(I, NJ + 2, K) * (T(I, NJ + 4, K) - T(I, NJ + 3, K)) * 2.0 / DXI
QRADN = SU(I, NJ + 3, K)

T(I, NJ + 3, K) = TOD(I, NJ + 3, K) + DTIME * (QCONS + QCONN + QRADN) /
(VOL * CPM(I, NJ + 3, K))

QSCONN = QSCONN + QCONS
QSRADN = QSRADN + QRADN

84 CONTINUE

DO 80 J = 3, NJ + 2
DO 80 K = 3, NK + 2

C *** ON THE WEST WALL
DXY = DXYC(2)
DXY = DXYC(3)
DZK = DZK + DZK
DYJ = DYJ + DYJ
VOL = DYJ + DXI

COND = (DXY + DXI) * DXI * DXY * COND(2, J, K) * COND(3, J, K) /
(DX1 + DXY)

QCON = DYJ * COND(I, NJ + 3, K) * (T(3, J, K) - T(2, J, K)) * 2.0 / (DX1 + DXY)

QCON = DYJ * COND(2, J, K) * (T(1, J, K) - T(2, J, K)) * 2.0 / DXI
QRADN = SU(2, J, K)

T(2, J, K) = TOD(2, J, K) + DTIME * (QCON + QCON + QRADN) /
(VOL * CPM(2, J, K))

T(1, J, K) = (2.0 * COND(2, J, K) * T(2, J, K) + HCOEF * TINF * DYJ) / 
(HCOEF * DXI + 2.0 * COND(2, J, K))

89
& \( (HCOEF\times DXI+2.0\times COND(2,J,K)) \)

\[ \begin{align*}
QSCONW &= QSCONW + QCONE \\
QSRADW &= QSRADW + QRADW \\
\end{align*} \]

C *** ON THE EAST WALL

\[ \begin{align*}
DXI &= DXXC(NI+3) \\
DXM1 &= DXXC(NI+2) \\
DYJ &= DYYC(J) \\
DZK &= DZZC(K) \\
DYZ &= DYZ*DZK \\
VOL &= DYZ*DXI \\
CONDW &= (DXM1*DXI+DXI*COND(NI+3,J,K)*COND(NI+2,J,K))/ \\
& (DXI*COND(NI+3,J,K)+DXM1*COND(NI+2,J,K)) \\
QCONW &= DYZ*CONDW \times (T(NI+2,J,K)-T(NI+3,J,K))*2.0/(DXI+DXM1) \\
QCONE &= DYZ*COND(2,J,K) \times (T(NI+4,J,K)-T(NI+3,J,K))*2.0/DXI \\
QRADW &= SU(NI+3,J,K) \\
T(NI+3,J,K) &= TOD(NI+3,J,K)+DTIME*(QCONW+QCONE+QRADW)/ \\
& (VOL*CPM(NI+3,J,K)) \\
T(NI+4,J,K) &= (2.0*COND(NI+3,J,K)*T(NI+3,J,K)+HCOEF*TINF*DXI)/ \\
& (HCOEF*DXI+2.0*COND(NI+3,J,K)) \\
\end{align*} \]

\[ \begin{align*}
QSCONE &= QSCONE + QCONW \\
QSRADW &= QSRADW + QRADW \\
\end{align*} \]

80 CONTINUE

C *** CALCULATE THE ENERGY LOST THROUGH (OR CONSUMED BY)

C 1) THE CAVITY WALLS; 2) CAVITY AIR; 3) DUCT

C *** TANK AIR

\[ \begin{align*}
\text{WERR} &= 0. \\
\text{WAIR} &= 0. \\
\text{DO} \ 25 \ I=3,NI+2 \\
\text{DO} \ 25 \ J=3,NJ+2 \\
\text{DO} \ 25 \ K=3,NK+2 \\
\text{IF} \ (\text{NOD}(I,J,K).EQ.1) \ \text{GO} \ \text{TO} \ 25 \\
DXI &= DXXC(I) \\
DYJ &= DYYC(J) \\
DZK &= DZZC(K) \\
\end{align*} \]

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME

\[ \begin{align*}
DXP1 &= DXXC(I+1) \\
DXI &= DXXC(I) \\
DXM1 &= DXXC(I-1) \\
DYP1 &= DYYC(J+1) \\
DYJ &= DYYC(J) \\
DYM1 &= DYYC(J-1) \\
DZP1 &= DZZC(K+1) \\
DZK &= DZZC(K) \\
DZM1 &= DZZC(K-1) \\
\end{align*} \]

C *** SURFACE LENGTH OF THE CONTROL VOLUME

\[ \begin{align*}
DXN &= DXXC(I) \\
DXS &= DXXC(I) \\
\end{align*} \]
DXF = DXXC(I)
DXB = DXXC(I)

DYF = DYYC(J)
DYB = DYYC(J)
DYE = DYYC(J)
DYW = DYYC(J)

DZE = DZZC(K)
DZW = DZZC(K)
DZN = DZZC(K)
DZS = DZZC(K)

C *** DEFINE AREA OF THE CONTROL VOLUME
DXYP = DXF*DYF
DXYP = DXB*DYB
DYZE = DYE*DZE
DYZW = DYW*DZW
DZXN = DZN*DXN
DZXS = DZS*DXS

VOL = DXI*DYJ*DZK
VOLDT = VOL/DTIME

RN = (R(I,J,K)*DYP1 + R(I,J+1,K)*DYJ)/(DYP1 + DYJ)
RS = (R(I,J,K)*DYM1 + R(I,J-1,K)*DYJ)/(DYM1 + DYJ)
RE = (R(I,J,K)*DXP1 + R(I+1,J,K)*DXI)/(DXP1 + DXI)
RW = (R(I,J,K)*DXM1 + R(I-1,J,K)*DXI)/(DXM1 + DXI)
RF = (R(I,J,K)*DZP1 + R(I,J,K+1)*DZK)/(DZP1 + DZK)
RB = (R(I,J,K)*DZM1 + R(I,J,K-1)*DZK)/(DZM1 + DZK)

CN = RN*V(I,J+1,K)*DZXN
CS = RS*V(I,J,K)*DZXS
CE = RE*U(I+1,J,K)*DYZE
CW = RW*U(I,J,K)*DYZW
CF = RF*W(I,J,K+1)*DXYF
CB = RB*W(I,J,K)*DXYB

WERR = WERR + T(I,J,K)*CPM(I,J,K)*SMP(I,J,K)
WRES = WRES + SMP(I,J,K)
WAIR = WAIR + (T(I,J,K)*R(I,J,K) - TOD(I,J,K)*ROD(I,J,K))*CPM(I,J,K)*VOLDT

25 CONTINUE

C *** SUM UP THE TOTAL WALT AT THAT TIME STEP (DIMENSIONLESS)
WINS = QQQ/H**3

C *** SUM UP THE TOTAL WALT LOST TO THE WALLS
QTCON = QSCONF + QSCONB + QSCONS + QSCONN + QSCONW + QSCONE
QTRAD = QSRADF + QSRADB + QSRADS + QSRADN + QSRADW + QSRADE
WWAL = QTCON + QTRAD

C *** EQUIVALENT INTERNAL HEAT SOURCE DUE TO RADIATION
QR = QTRAD*UO*CP0*RHOO*TA*H**2

C *** TOTAL WALT EXHAUSTED THROUGH THE DUCT

91
IF (NMS.EQ.0) THEN
  WWFAN=0.0
ELSE
  WWFAN=8000.*CPM(12,3,12)*(T(12,3,12)-1.0)*R(12,3,12)/
  (60.*H**2*UO)
ENDIF

C *** THE ENERGY CALCULATION
QSIN=QSIN+WINS*DTIME
QSWER=QSWER+WERR*DTIME
QSWAL=QSWAL+WWAL*DTIME
QSAIR=QSAIR+WAIR*DTIME
QSFAN=QSFAN+WWFAN*DTIME
RETURN
END

******************************************************************************
******************************************************************************
SUBROUTINE CALU
******************************************************************************
*CALCULATES THE U COMPONENT OF THE VELOCITY

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
  & DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
  & SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
COMMON/BL22/CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
  & JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
  & COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
  & WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
  & U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
  & CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
  & WPD(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
  & SMPF(25,25,15),PP(25,25,15),DU(25,25,15),
  & DV(25,25,15),DW(25,25,15)
COMMON/BL36/AF(25,25,15),AN(25,25,15),SP(25,25,15),
  & AS(25,25,15),AF(25,25,15),AN(25,25,15),
  & SU(25,25,15),RI(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
  & CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
DO 100 K=2,NK+3
  DO 100 J=2,NJ+3
    DO 100 I=3,NI+3

C *** CENTRAL LENGTH OF THE U CONTROL VOLUME
DXP1=DXXS(I+1)
DXI =DXXS(I)
DXM1=DXXS(I-1)

DYP1=DYYC(J+1)
DYJ =DYYC(J)
DYM1=DYYC(J-1)

DZP1=DZZC(K+1)
DZK =DZZC(K)
DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
DXN=DXXS(I)
DXS=DXXS(I)
DXF=DXXS(I)
DXB=DXXS(I)

DYF=DYYC(J)
DYB=DYYC(J)
DYE=DYYC(J)
DYW=DYYC(J)

DJE=DZZC(K)
DZW=DZZC(K)
DJN=DZZC(K)
DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR U
DXEE=DXXC(I+1)
DXXC(I)
DXW =DXXC(I-1)
DXWW=DXXC(I-2)

DYNN=DYYS(J+2)
DYN =DYYS(J+1)
DYS =DYYS(J)
DYSS=DYYS(J-1)

DZFF=DZZS(K+2)
DZF =DZZS(K+1)
DZB =DZZS(K)
DZBB=DZZS(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
DXYF=DXF*DYF
DXYB=DXB*DYB
DYZE=DYE*DZE
DYZW=DYW*DZW
DZNL=DZN*DXN
DZXS=DZS*DXS

VOL=DXI*DYJ*DZK
VOLDT=VOL/DTIME

ZYOYN=DZN/DYN
ZYOYS=DZS/DYS

93
**USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.**

\[
G_{NE} = \text{SILIN}(R(I,J+1,K),R(I,J,K),D_{Y1},D_{YJ})^*V(I,J+1,K)
\]
\[
G_{NW} = \text{SILIN}(R(I-1,J+1,K),R(I,J,K),D_{Y1},D_{YJ})^*V(I-1,J+1,K)
\]
\[
G_{SE} = \text{SILIN}(R(I,J-1,K),R(I,J,K),D_{MI},D_{YJ})^*V(I,J-1,K)
\]
\[
G_{SW} = \text{SILIN}(R(I-1,J-1,K),R(I,J,K),D_{MI},D_{YJ})^*V(I-1,J-1,K)
\]
\[
G_{E} = \text{SILIN}(R(I+1,J,K),R(I,J,K),D_{XEE},D_{XE})^*U(I+1,J,K)
\]
\[
G_{P} = \text{SILIN}(R(I-1,J,K),R(I,J,K),D_{XEE},D_{XE})^*U(I-1,J,K)
\]
\[
G_{FE} = \text{SILIN}(R(I,J,K+1),R(I,J,K),D_{ZP1},D_{ZK})^*W(I,J,K+1)
\]
\[
G_{FW} = \text{SILIN}(R(I,J,K-1),R(I,J,K),D_{ZM1},D_{ZK})^*W(I,J,K-1)
\]
\[
G_{BE} = \text{SILIN}(R(I,J,K-1),R(I,J,K),D_{ZM1},D_{ZK})^*W(I,J,K)
\]
\[
G_{BW} = \text{SILIN}(R(I,J,K+1),R(I,J,K),D_{ZM1},D_{ZK})^*W(I,J,K+1)
\]

**MASS FLOW RATE**

\[
C_{E} = 0.5^* (G_{E} + G_{P})^*D_{YZE}
\]
\[
C_{W} = 0.5^* (G_{P} + G_{W})^*D_{YZW}
\]
\[
C_{N} = \text{SILIN}(C_{NE},G_{NW},D_{XE},D_{XW})^*D_{ZN}
\]
\[
C_{S} = \text{SILIN}(G_{SE},G_{SW},D_{XE},D_{XW})^*D_{XS}
\]
\[
C_{F} = \text{SILIN}(G_{FE},G_{FW},D_{XE},D_{XW})^*D_{YF}
\]
\[
C_{B} = \text{SILIN}(G_{BE},G_{BW},D_{XE},D_{XW})^*D_{YB}
\]

**VISCOSITY**

\[
V_{ISE} = V(I,J,K)
\]
\[
V_{ISW} = V(I-1,J,K)
\]
\[
V_{ISN} = (V(I,J+1,K)+V(I,J,K)+V(I-1,J+1,K)+V(I-1,J,K))/4.0
\]
\[
V_{ISS} = (V(I,J-1,K)+V(I,J,K)+V(I-1,J-1,K)+V(I-1,J,K))/4.0
\]
\[
V_{ISF} = (V(I,J,K+1)+V(I,J,K)+V(I-1,J,K+1)+V(I-1,J,K))/4.0
\]
\[
V_{ISB} = (V(I,J,K-1)+V(I,J,K)+V(I-1,J,K-1)+V(I-1,J,K))/4.0
\]
\[
V_{ISN} = Z_{XOYN}*V_{ISN}
\]
\[
V_{ISS} = Z_{XOYS}*V_{ISS}
\]
\[
V_{ISE} = Y_{XOYE}*V_{ISE}
\]
\[
V_{ISW} = Y_{XOYW}*V_{ISW}
\]
\[
V_{ISF} = X_{YOZF}*V_{ISF}
\]
\[
V_{ISB} = X_{YOZB}*V_{ISB}
\]

**QUICK SCHEME**

\[
C_{E} = (\text{ABS}(C_{E})+C_{E})^*D_{XE}/(D_{XI}*16.)
\]
\[
C_{EM} = (\text{ABS}(C_{E})-C_{E})^*D_{XE}/(D_{XI}*16.)
\]
\[
C_{WP} = (\text{ABS}(C_{W})+C_{W})^*D_{XW}/(D_{XM}*16.)
\]
\[
C_{WM} = (\text{ABS}(C_{W})-C_{W})^*D_{XW}/(D_{XI}*16.)
\]
\[
C_{NP} = (\text{ABS}(C_{N})+C_{N})^*D_{Y1}/(D_{YN}*(D_{YN}+D_{YS}))
\]
\[
C_{NM} = (\text{ABS}(C_{N})-C_{N})^*D_{Y1}/(D_{YN}*(D_{YN}+D_{YN}))
\]

94
CSP=(ABS(CS)+CS)*DYM1*DYJ/(8.*DYS*(DYS+DYSS))
CSM=(ABS(CS)-CS)*DYM1*DYJ/(8.*DYS*(DYS+DYN))

CFP=(ABS(CF)+CF)*DZP1*DZK/(8.*DZF*(DZF+DZFF))
CFM=(ABS(CF)-CF)*DZP1*DZK/(8.*DZF*(DZF+DZBB))

CBP=(ABS(CB)+CB)*DZN1*DZK/(8.*DZB*(DZB+DZBB))
CBM=(ABS(CB)-CB)*DZM1*DZK/(8.*DZB*(DZB+DZF))

AE(I,J,K) = -.5*CE +CWM*DXW/DXE+CEP+CEM*(1.+DXE/DXEE)+VISE1
AW(I,J,K) = .5*CW +CEP*DXE/DXW+CWM+CWP*(1.+DXW/DXWW)+VISW1
AN(I,J,K) = -.5*CN*DYJ+CSrI*DYS)/DYN+CNP+CNM*(1.+DYN/DYNN)+VISN1
AS(I,J,K) = (.5*CS*DYJ+CN*DNS)/DYS+CSP*CN*DYSS)+VISS1
AF(I,J,K) = (.5*CF*DZK+CBM*DZB)/DZF+CFP+CFM*(1.+DZF/DZFF)+VISFL
AB(I,J,K) = (.5*CB*DZK+CFP*DZF)/DZB+CBM.+CBP*(1.+DZB/DZBB)+VISB1

C *** BOUNDARY CONSIDERATION
IF(I.LT.NI+3) THEN
   AEE=CEM*DXE/DXEE
   AEER=AEE*UPD(I+2,J,K)
ELSE
   AEE=0.
   AEER=0.
ENDIF

IF (I.GT.3) THEN
   AWW=-GWP*DXW/DXWW
   AWWR=AWW*UPD(I-2,J,K)
ELSE
   AWW=0.
   AWWR=0.
ENDIF

IF (J.LT.NJ+3) THEN
   ANN=CNM*DYN/DYNN
   ANNR=ANN*UPD(I,J+2,K)
ELSE
   ANN=0.
   ANNR=0.
ENDIF

IF (J.GT.2) THEN
   ASS=-CSP*DYS/DYSS
   ASSR=ASS*UPD(I,J-2,K)
ELSE
   ASS=0.
   ASSR=0.
ENDIF

IF (K.LT.NK+3) THEN
   AFF=-CFM*DZF/DZFF
   AFFR=AFF*UPD(I,J,K+2)
ELSE
   AFF=0.
   AFFR=0.
ENDIF
IF (K.GT.2) THEN
   ABB=-CBP*DZB/DZBB
   ABBR=ABB*UPD(I,J,K-2)
ELSE
   ABB=0.
   ABBR=0.
ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
IF (NOD(I-2,J,K).NE.0) THEN
   AWW=0.0
   AWWR=0.0
ENDIF
IF (NOD(I+1,J,K).NE.0) THEN
   AEE=0.0
   AEER=0.0
ENDIF
IF (NOD(I,J-1,K).NE.0) THEN
   ASS=0.0
   ASSR=0.0
ENDIF
IF (NOD(I,J+1,K).NE.0) THEN
   ANN=0.0
   ANNR=0.0
ENDIF
IF (NOD(I,J,K-1).NE.0) THEN
   ABB=0.0
   ABBR=0.0
ENDIF
IF (NOD(I,J,K+1).NE.0) THEN
   AFF=0.0
   AFFR=0.0
ENDIF

C *** SU FROM NORMAL STRESS
RE=(SIG11(I ,J,K) -(U(I+1 ,J,K)-U(I ,J,K ))*VISE/DXE)*DYZE
RW=(SIG11(I-1,J,K)-U(I-1,J,K))*VISE/DXW)*DYZW
RN=(SIG12(I,J+1,K )-(U(I,J+1,K)-U(I,J,K))*VISN/DYN)*DZXN
RS=(SIG12(I,J,K )-(U(I,J,K )-U(I,J-1,K ))*VISN/DYS)*DZXS
RF=(SIG13(I,J,K+1 )-(U(I,J,K+1)-U(I,J,K ))*VISF/DZF)*DXYF
RB=(SIG13(I,J,K+1 )-(U(I,J,K)+U(I,J,K-1 ))*VISB/DZB)*DXYB

C *** SU FROM CURVED STRESSES AND ACCELERATIONS
AVG12=0.5*(SIG12(I,J+1,K )+SIG12(I,J,K))
AVG13=0.5*(SIG13(I,J ,K+1)+SIG13(I,J,K))
AVG22=SILIN(SIG22(I,J,K),SIG22(I-1,J,K),DXE,DXW)
AVG33=SILIN(SIG33(I,J,K),SIG33(I-1,J,K),DXE,DXW)
AU1=U(I,J,K)
AU2=BILIN(V(I,J+1,K),V(I,J,K),DYJ,DYJ,
& V(I-1,J+1,K),V(I-1,J,K),DYJ,DXE,DXW)
AU3=BILIN(W(I,J+1,K),W(I,J,K),DZK,DZK,
& W(I-1,J,K),DZK,DXW)
AR=SILIN(R(I,J,K),R(I-1,J,K),DXE,DXE)

ARU12=AR*AU1*AU2
ARU13=AR*AU1*AU3
ARU22=AR*AU2*AU2
ARU33=AR*AU3*AU3

RRY=(AVG12-ARU12)*DZK*(DXN-DXS)
RRZ=(AVG13-ARU13)*DYJ*(DXF-DXB)
RRX=(AVG22-ARU22)*DZK*(DYE-DYW)+(AVG33-ARU33)*DYJ*(DZE-DZW)

AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+ASCI,J,K)+AF(I,J,K)+
& AB(I,J,K)+AAW+ANN+ASS+AFF+ABB

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** Y DIRECTION
DO 500 K=2,NK+3
DO 500 I=3,NI+3
   SP(I,2 ,K)=SP(I,2 ,K)-AS(I,2 ,K)
   SP(I,NJ+3,K)=SP(I,NJ+3,K)-AN(I,NJ+3,K)
   AN(I,NJ+3,K)=0.
   AS(I,2 ,K)=0.
500 CONTINUE

C *** X DIRECTION
DO 502 K=2,NK+3
DO 502 J=2,NJ+3
   AW(3,J,K)=0.0
   AE(NI+3,J,K)=0.0
502 CONTINUE

C *** Z DIRECTION
DO 600 I=3,NI+3
DO 600 J=2,NJ+3
   SP(I,J,2 )=SP(I,J,2 )-AB(I,J,2 )
   SP(I,J,NK+3)=SP(I,J,NK+3)-AF(I,J,NK+3)
   AF(I,J,NK+3)=0.
   AB(I,J,2 )=0.
600 CONTINUE

C *** MODIFICATION FOR DECK BOUNDARIES
IF (NCHIP.EQ.0) GOTO 201
DO 101 N=1,NCHIP
   IB =1CHPB(N)

100 CONTINUE

101 CONTINUE
IE = IB+NCHPI(N)-1
JB = JCHPB(N)
JE = JB+NCHPJ(N)-1
KB = KCHPB(N)
KE = KB+NCHPK(N)-1

DO 102 J=JB,JE-1
  DO 102 K=KB,KE-1
    AE(IB-1,J,K)=0.0
    AW(IE+1,J,K)=0.0
  102 CONTINUE

DO 103 I=IB,IE
  DO 103 K=KB,KE-1
    SP(I,JB-1,K)=SP(I,JB-1,K)-AN(I,JB-1,K)
    AN(I,JB-1,K)=0.0
    SP(I,JE,K)=SP(I,JE,K)-AS(I,JE,K)
    AS(I,JE,K)=0.0
  103 CONTINUE

DO 106 I=IB,IE
  DO 106 J=JB,JE-1
    DO 106 K=KB,KE-1
      SP(I,J,K)=-1.0E2
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      AB(I,J,K)=0.
      AF(I,J,K)=0.
      SU(I,J,K)=0.
  106 CONTINUE

C *** FOR THE CELLS INSIDE OF THE DECKS
DO 104 I=IB,IE
  DO 104 J=JB,JE-1
    DO 104 K=KB,KE-1
      SP(I,J,K)=SP(I,J,K)-SP(I,J,K)
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      AB(I,J,K)=0.
      AF(I,J,K)=0.
      SU(I,J,K)=0.
  104 CONTINUE
101 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
201 DO 301 K=2,NK+3
  DO 301 J=2,NJ+3
    DO 301 I=3,NI+3
      DYJ=DYYC(J)
      DZK=DZZC(K)
      DYZ=DYJ*DZK
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
      DU(I,J,K)=DYZ/AP(I,J,K)
  301 CONTINUE

C *** SOLVE FOR U
CALL TRID (4,3,3,NI+2,NJ+2,NK+2,U)
C *** RESET THE VELOCITY INSIDE OF DECK
IF (NCHIP.EQ.0) GOTO 111
DO 110 N=1,NCHIP
   IB=ICHPB(N)
   IE=IB+NCHPI(N)-1
   JB=JCHPB(N)
   JE=JB+NCHPJ(N)-1
   KB=KCHPB(N)
   KE=KB+NCHPK(N)-1
   DO 108 I=IB,IE
      DO 108 J=JB,JE-1
      DO 108 K=KB,KE-1
         U(I,J,K)=0.0
      108 CONTINUE
   110 CONTINUE
111 RETURN
END

*******************************************************************************
*******************************************************************************
SUBROUTINE CALV
*******************************************************************************
*******************************************************************************
*CALCULATES THE V COMPONENT OF THE VELOCITY

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
& DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
& SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
& JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
& COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
& WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
& U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
& CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
& WPD(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
& SMP(25,25,15),PP(25,25,15),DU(25,25,15),
& DV(25,25,15),DW(25,25,15)
COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
& AS(25,25,15),AF(25,25,15),AB(25,25,15),SA(25,25,15),
& SU(25,25,15),RI(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
& CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
DO 100 K=2,NK+3
DO 100 J=3,NJ+3
DO 100 I=2,NI+3

99
C *** CENTRAL LENGTH OF THE V CONTROL VOLUME
   DXP1=DXXC(I+1)
   DXI =DXXC(I)
   DXM1=DXXC(I-1)

   DYP1=DYYS(J+1)
   DYJ =DYYS(J)
   DYM1=DYYS(J-1)

   DZP1=DZZC(K+1)
   DZK =DZZC(K)
   DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
   DXN=DXXC(I)
   DXS=DXXC(I)
   DXF=DXXC(I)
   DXB=DXXC(I)

   DYF=DYYS(J)
   DYB=DYYS(J)
   DYE=DYYS(J)
   DYW=DYYS(J)

   DZE=DZZC(K)
   DZW=DZZC(K)
   DZN=DZZC(K)
   DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR V
   DXEE=DXXS(I+2)
   DXE =DXXS(I+1)
   DXW =DXXS(I)
   DXWW=DXXS(I-1)

   DYNN=DYYC(J+1)
   DYN =DYYC(J)
   DYS =DYYC(J-1)
   DYSS=DYYC(J-2)

   DZFF=DZZC(K+2)
   DZF =DZZC(K+1)
   DZB =DZZC(K)
   DZBB=DZZC(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
   DXYF=DXF*DYF
   DXYB=DXB*DYB
   DYZE=DYE*DZE
   DYZW=DYW*DZW
   DZN=DZN*DXN
   DZXS=DZS*DXS

   VOL =DXI*DYJ*DZK
   VOLDT=VOL/DTIME

100
C *** USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE

C PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.

GN=SILIN(R(I+1,J,K),R(I,J,K),DXP1,DXI)*U(I+1,J,K)
GWN=SILIN(R(I-1,J,K),R(I,J,K),DXM1,DXI)*U(I,J-1,K)

GP=SILIN(R(I,J+1,K),R(I,J,K),DYN,DYS)*V(I,J+1,K)
GBN=SILIN(R(I,J,K+1),R(I,J,K),DYN,DYS)*V(I,J,K+1)

C MASS FLOW RATE

CN=0.5*(GN+GP)*DZXN
CS=0.5*(GP+GS)*DZXS
CE=SILIN(GPN,GSN,DYN,DYS)*DYZE
CW=SILIN(GWN,GWS,DYN,DYS)*DYZW

C VISCOSITY

VISN=V(I,J,K)
VISI=V(I-1,J,K)

VISE=(V(I+1,J+1,K)+V(I+1,J,K)+V(I+1,J-1,K)+V(I,J-1,K))/4.0
VISW=(V(I-1,J+1,K)+V(I-1,J,K)+V(I-1,J-1,K)+V(I,J-1,K))/4.0
VISF=(V(I,J+1,K+1)+V(I,J+1,K)+V(I,J-1,K+1)+V(I,J-1,K))/4.0
VISB=(V(I,J-1,K-1)+V(I,J-1,K)+V(I,J,K-1)+V(I,J,K))/4.0

VISN=ZXOYN*VISN
VISI=ZXOYS*VISI
VISE=YZOXE*VISE
VISW=YZOXW*VISW
VISF=XYOZF*VISF
VISB=XYOZB*VISB

C QUICK SCHEME

CEP=(ABS(CE)+CE)*DXP1*DXI/(DXE*(DXE+DXW))*8.0
CEM=(ABS(CE)-CE)*DXP1*DXI/(DXE*(DXE+DXW))*8.0
CWP=(ABS(CW)+CW)*DXW1*DXI/(DXW*(DXW+DXWW))*8.0
CWM=(ABS(CW)-CW)*DXW1*DXI/(DXW*(DXW+DXE))*8.0
CNP=(ABS(CN)+CN)*DYN/(DYJ*16.0)
CNM=(ABS(CN)-CN)*DYN/(DYJ*16.0)
CSP=(ABS(CS)+CS)*DYS/(DYJ*16.0)

C}

101
CSM = \( (\text{ABS}(CS) - CS) \times DYS / (DYZ \times 16.) \)

CFP = \( (\text{ABS}(CF) + CF) \times DZP1 \times DZK / (DZF \times (DZF + DZB) \times 8.) \)

CFM = \( (\text{ABS}(CF) - CF) \times DZP1 \times DZK / (DZF \times (DZF + DZFF) \times 8.) \)

CBP = \( (\text{ABS}(CB) + CB) \times DZM1 \times DZK / (DZB \times (DZB + DZBB) \times 8.) \)

CBt1 = \( (\text{ABS}(CB) - CB) \times DZM1 \times DZK / (DZB \times (DZB + DZF) \times 8.) \)

AE(I, J, K) = \((-0.5 \times CE \times DXI + CWM \times DXW) / DXE + CEP + CEM \times (1 + DXE / DXEE) + VISE1\)

AW(I, J, K) = \((0.5 \times CW \times DXI + CEP \times DXE) / DXW + CWM + CWP \times (1 + DXW / DXWW) + VISW1\)

AN(I, J, K) = \(-0.5 \times CN + CSM \times DYS / DYN + CNP + CNII \times (1 + DYN / DYNN) + VISN1\)

AS(I, J, K) = \((-0.5 \times CS + CNP \times DYN / DYS + CSM + CSP \times (1 + DYS / DYSS) + VISS1\)

AF(I, J, K) = \((-0.5 \times CF \times DZK + CBM \times DZB) / DZF + CFP + CFM \times (1 + DZF / DZFF) + VISF1\)

AB(I, J, K) = \((0.5 \times CB \times DZK + CF \times DZF) / DZB + CBM + CBP \times (1 + DZB / DZBB) + VISB1\)

C *** BOUNDARY CONSIDERATION

IF (I.LT.NI+3)
  THEN
    AEE = -CEM*DXE/DXEE
    AEER = AEE*VPD(I+2,J,K)
  ELSE
    AEE = 0.
    AEER = 0.
ENDIF

IF (I.GT.2)
  THEN
    AWW = -CWP*DXW/DXWW
    AWWR = AWW*VPD(I-2,J,K)
  ELSE
    AWW = 0.
    AWWR = 0.
ENDIF

IF (J.LT.NJ+3)
  THEN
    ANN = -CNM*DYN/DYNN
    ANNR = ANN*VPD(I,J+2,K)
  ELSE
    ANN = 0.
    ANNR = 0.
ENDIF

IF (J.GT.3)
  THEN
    ASS = -CSP*DYS/DYSS
    ASSR = ASS*VPD(I,J-2,K)
  ELSE
    ASS = 0.
    ASSR = 0.
ENDIF

IF (K.LT.NK+3)
  THEN
    AFF = -CFM*DZF/DZFF
    AFFR = AFF*VPD(I,J,K+2)
  ELSE
    AFF = 0.
    AFFR = 0.
ENDIF

IF (K.GT.2)
  THEN

102
\[ ABB = -CBP \cdot DZB / DZBB \]
\[ ABBR = ABB \cdot VPD(I, J, K-2) \]

**ELSE**
\[ ABB = 0. \]
\[ ABBR = 0. \]
**ENDIF**

**C *** MODIFICATION FOR DECK BOUNDARIES**

**IF** (NOD(I-1, J, K) .NE. 0) **THEN**
\[ AWW = 0.0 \]
\[ AWWR = 0.0 \]
**ENDIF**

**IF** (NOD(I+1, J, K) .NE. 0) **THEN**
\[ AEE = 0.0 \]
\[ AEE R = 0.0 \]
**ENDIF**

**IF** (NOD(I, J-2, K) .NE. 0) **THEN**
\[ ASS = 0.0 \]
\[ ASS R = 0.0 \]
**ENDIF**

**IF** (NOD(I, J+1, K) .NE. 0) **THEN**
\[ ANN = 0.0 \]
\[ A NN R = 0.0 \]
**ENDIF**

**IF** (NOD(I, J, K-1) .NE. 0) **THEN**
\[ ABB = 0.0 \]
\[ ABBR = 0.0 \]
**ENDIF**

**IF** (NOD(I, J, K+1) .EQ. 0) **THEN**
\[ AFF = 0.0 \]
\[ AFFR = 0.0 \]
**ENDIF**

**C *** SU FROM NORMAL STRESS**

\[ RN = (SIG22(I, J, K) - (V(I, J+1, K) - V(I, J, K)) \cdot VISN / DYN) \cdot DZYN \]
\[ RS = (SIG22(I, J-1, K) - (V(I, J, K) - V(I-1, J, K)) \cdot VISN / DYN) \cdot DZYS \]
\[ RE = (SIG12(I+1, J, K) - (V(I+1, J, K) - V(I, J, K)) \cdot VISE / DCE) \cdot DYZE \]
\[ RW = (SIG12(I, J-1, K) - (V(I, J, K) - V(I-1, J, K)) \cdot VISE / DCE) \cdot DYZW \]
\[ RF = (SIG23(I, J, K+1) - (V(I, J, K+1) - V(I, J, K)) \cdot VISF / DZF) \cdot DXYF \]
\[ RB = (SIG23(I, J, K-1) - (V(I, J, K) - V(I, J, K-1)) \cdot VISB / DZB) \cdot DXYB \]

**C *** SU FROM CURVED STRESSES AND ACCELERATIONS**

\[ AVG12 = 0.5 \cdot (SIG12(I+1, J, K) + SIG12(I, J, K)) \]
\[ AVG23 = 0.5 \cdot (SIG23(I, J, K+1) + SIG23(I, J, K)) \]
\[ AVG11 = SILIN(SIG11(I, J, K), SIG11(I, J, K-1), DYN, DYS) \]
\[ AVG33 = SILIN(SIG33(I, J, K), SIG33(I, J, K-1), DYN, DYS) \]

\[ AU2 = V(I, J, K) \]
\[ AU1 = BILIN(U(I+1, J, K), U(I, J, K), DXI, DXI, \]
\[ U(I+1, J-1, K), U(I, J-1, K), DXI, DXI, DYN, DYS) \]
\[ AU3 = BILIN(W(I, J, K+1), W(I, J, K), DZK, DZK, \]

103
& W(I,J-1,K+1), W(I,J-1,K), DZK, DZK, DYN, DYS)

AR = SILIN(R(I,J,K), R(I,J-1,K), DYN, DYS)

ARU12 = AR * AU1 * AU2
ARU23 = AR * AU2 * AU3
ARU11 = AR * AU1 * AU1
ARU33 = AR * AU3 * AU3

RRX = (AVG12 - ARU12) * DZK * (DYE - DYW)
RRZ = (AVG23 - ARU23) * Dxi * (DYF - DYN)
RRY = (AVG11 - ARU11) * DZK * (DXN - DXS) + (AVG33 - ARU33) * Dxi * (DZN - DZS)

AP(I,J,K) = AE(I,J,K) + AW(I,J,K) + AN(I,J,K) + AS(I,J,K) +
& AF(I,J,K) + AB(I,J,K) + AEE + AWW + ANN + ASS + AFF + ABB
SP(I,J,K) = -(ROD(I,J,K) * DYS + ROD(I,J-1,K) * DYN) * VOLDT / (DYS + DYN)
SU(I,J,K) = SP(I,J,K) * VOD(I,J,K) + DZK * DXI * (P(I,J-1,K) - P(I,J,K)) +
& AEER + AWWR + ANNR + ASSR + AFFR + ABBR + RE - RW + RN - RS + RF - RB + RRX +
& RRZ - RRY

100 CONTINUE

C ** TAKE CARE OF B.C. THRU AN, AS, AE, AW, AF, AB, SP AND SU

C ** Y DIRECTION
DO 500 K = 2, NK + 3
DO 500 I = 2, NI + 3
AS(I, 3, K) = 0.
AN(I, NJ + 3, K) = 0.
500 CONTINUE

C ** X DIRECTION
DO 502 K = 2, NK + 3
DO 502 J = 3, NJ + 3
SP(2, J, K) = SP(2, J, K) - AW(2, J, K)
SP(NI + 3, J, K) = SP(NI + 3, J, K) - AE(NI + 3, J, K)
AW(2, J, K) = 0.0
AE(NI + 3, J, K) = 0.0
502 CONTINUE

C ** Z DIRECTION
DO 600 I = 2, NI + 3
DO 600 J = 3, NJ + 3
SP(I, J, 2) = SP(I, J, 2) - AB(I, J, 2)
SP(I, J, NK + 3) = SP(I, J, NK + 3) - AF(I, J, NK + 3)
AF(I, J, NK + 3) = 0.
AB(I, J, 2) = 0.
600 CONTINUE

C ** MODIFICATION FOR DECK BOUNDARIES
IF (NCHIP.EQ.0) GOTO 201
DO 101 N = 1, NCHIP
IB = ICHPB(N)
IE = IB + NCHP1(N) - 1
JB = JCHPB(N)
JE = JB + NCHP3(N) - 1
KB = KCHPB(N)

104
KE = KB + NCHPK(N) - 1
DO 102 J = JB, JE
    DO 102 K = KB, KE - 1
        SP(IB-1, J, K) = SP(IB-1, J, K) - AE(IB-1, J, K)
        AE(IB-1, J, K) = 0.0
        SP(IE, J, K) = SP(IE, J, K) - AW(IE, J, K)
        AW(IE, J, K) = 0.0
    CONTINUE
DO 103 I = IB, IE - 1
    DO 103 K = KB, KE - 1
        AN(I, JB-1, K) = 0.0
        AS(I, JE+1, K) = 0.0
    CONTINUE
DO 106 I = IB, IE - 1
    DO 106 J = JB, JE
        SP(I, J, KB-1) = SP(I, J, KB-1) - AF(I, J, KB-1)
        AF(I, J, KB-1) = 0.0
        SP(I, J, KE) = SP(I, J, KE) - AB(I, J, KE)
        AB(I, J, KE) = 0.0
    CONTINUE
C *** MODIFICATION FOR THE CELLS INSIDE OF THE DECKS
DO 104 I = IB, IE - 1
    DO 104 J = JB, JE
        DO 104 K = KB, KE - 1
            SP(I, J, K) = -1.0E2
            AW(I, J, K) = 0.
            AE(I, J, K) = 0.
            AS(I, J, K) = 0.
            AN(I, J, K) = 0.
            AB(I, J, K) = 0.
            AF(I, J, K) = 0.
        CONTINUE
    CONTINUE
104 CONTINUE
101 CONTINUE
C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENTIATION EQUATIONS
201 DO 300 K = 2, NK + 3
    DO 300 J = 3, NJ + 3
        DO 300 I = 2, NI + 3
            DXI = DXXC(I)
            DZK = DZZC(K)
            DZX = DZK * DXI
            AP(I, J, K) = AP(I, J, K) - SP(I, J, K)
            DV(I, J, K) = DZX / AP(I, J, K)
        CONTINUE
    CONTINUE
300 CONTINUE
C *** SOLVE FOR V
    CALL TRID (3, 4, 3, NI+2, NJ+2, NK+2, V)
C *** RESET THE VELOCITY INSIDE OF THE DECKS
    IF (NCHIP.EQ.0) GOTO 111
    DO 110 N = 1, NCHIP
        IB = ICHPB(N)
105
**SUBROUTINE CALVIS**

* THIS SUBROUTINE CALCULATES THE TURBULENT VISCOSITY AND UPDATES THE VISCOSITY MATRIX *

**IMPLICIT DOUBLE PRECISION (A-H,O-Z)***

```plaintext
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
  DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40),
  HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX

**C *** CALCULATE LOCAL SHEAR AND VISCOSITY VIS(I,J,K)**

**C *** SPECIFY LOCAL TURBULENT LENGTH SCALES SMPP(I,J,K)**

DO 611 K=3,NK+2
  DO 611 J=3,NJ+2
  DO 611 I=3,NI+2

**C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME**

DXP1 = DXXC(I+1)
DXI = DXXC(I)
DXM1 = DXXC(I-1)

DYP1 = DYYC(J+1)
DYJ = DYYC(J)
DYM1 = DYYC(J-1)
```
DZP1 = DZZC(K+1)
DZK = DZZC(K)
DZM1 = DZZC(K-1)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
DXE = DXXS(I+1)
DXW = DXXS(I)
DYN = DYYS(J+1)
DYS = DYYS(J)
DZF = DZZS(K+1)
DZB = DZZS(K)

C *** CALCULATE DV/DX, D2V/DX2, DU/DX, D2U/DX2, DW/DX AND D2W/DX2
DUDX = (U(I+1,J,K) - U(I,J,K))/DXI
DUDXW = 0.5*(U(I+1,J,K) - U(I-1,J,K))/DXW
DUDXE = 0.5*(U(I+2,J,K) - U(I,J,K))/DXXE
D2UDX2 = (DUDXE - DUDXW)/DXI

DVDXW = 0.5*(V(I,J+1,K) + V(I,J,K) - V(I-1,J+1,K) - V(I-1,J,K))/DXW
DVDXE = 0.5*(V(I+1,J+1,K) + V(I+1,J,K) - V(I,J+1,K) - V(I,J,K))/DXE
DVDX = 0.5*(DVDXE + DVDXW)
D2VDX2 = (DVDXE - DVDXW)/DXI

C *** CALCULATE DU/DY, D2U/DY2, DV/DY, D2V/DY2, DW/DY AND D2W/DY2
DVDY = (V(I,J+1,K) - V(I,J,K))/DYJ
DVDYS = 0.5*(V(I,J+1,K) - V(I,J-1,K))/DYS
DVDYN = 0.5*(V(I,J+2,K) - V(I,J,K))/DYN
D2VDY2 = (DVDYN - DVDYS)/DYJ

DUDYS = 0.5*(U(I+1,J,K) + U(I,J,K) - U(I+1,J-1,K) - U(I,J-1,K))/DYS
DUDYN = 0.5*(U(I+1,J+1,K) + U(I,J+1,K) - U(I+1,J,K) - U(I,J,K))/DYN
DUDY = 0.5*(DUDYN + DUDYS)
D2UDY2 = (DUDYN - DUDYS)/DYJ

C *** CALCULATE DU/DZ, D2U/DZ2, DV/DZ, D2V/DZ2, DW/DZ AND D2W/DZ2
DWDZ = (W(I,J,K+1) - W(I,J,K))/DZK
DWDZF = 0.5*(W(I,J,K+2) - W(I,J,K))/DZF
DWDZB = 0.5*(W(I,J,K+1) - W(I,J,K-1))/DZB
D2WDZ2 = (DWDZF - DWDZB)/DZK

DVDZB = 0.5*(V(I,J+1,K) + V(I,J,K) - V(I,J+1,K-1) - V(I,J,K-1))/DZB
DVDZF = 0.5*(V(I,J+1,K+1) + V(I,J,K+1) - V(I,J+1,K) - V(I,J,K))/DZF
DVDZ = 0.5*(DVDZF + DVDZB)
D2VDZ2 = (DVDZF - DVDZB)/DZK

DUDZB = 0.5*((U(I+1,J,K)+U(I,J,K)-U(I+1,J,K-1)-U(I,J,K-1))/DZB
DUDZF = 0.5*((U(I+1,J,K+1)+U(I,J,K+1)-U(I+1,J,K)-U(I,J,K))/DZF
DUDZ = 0.5*(DUDZF+DUDZB)

D2UDZ2 = (DUDZF - DUDZB)/DZK

C *** CALCULATE THE DENSITY GRADIENT WITH RESPECT TO THE VERTICAL
DRDGA = (R(I,J,K+1)-REQ(I,J,K+1)-R(I,J,K-1)+REQ(I,J,K-1))/
& (DZF+DZB)

C *** CALCULATE STRAIN
STRAIN = DUDY**2 + DVDX**2 + DWDY**2 + DVDZ**2 + DWDY**2 + DUDZ**2

DDO2 = SQRT(STRAIN+DUDZ**2+DUDZ**2+DWDY**2+DUDZ**2)

IF(DDO2.EQ.0. OR. STRAIN.EQ.0.) THEN
  VIS(I,J,K) = VISL
ELSE

C *** CALCULATE TURBULENT LENGTH SCALE SMPP(I,J)
SMPP12 = DUDY**2 + DVDX**2 + DWDY**2 + DVDZ**2 + DWDY**2 + DUDZ**2

SMPP12 = D2UDY2**2 + D2UDY2**2 + D2UDZ2**2 + D2UDZ2**2 + D2WDY2**2 + D2WDY2**2

SMPP(I,J,K) = CNT*(SMP123 + SMPP12)/2.

C *** CALCULATE RICHARDSON NUMBER
RI(I,J,K) = -BUOY*DRDGA/(R(I,J,K)*STRAIN)

ABRIPR = ABTURB + RI(I,J,K)/PRT

IF(ABRIPR.LT.0.) THEN
  VIS(I,J,K) = VISL
ELSEIF(ABRIPR.EQ.0.) THEN
  VIS(I,J,K) = VISMAX
ELSE
  VIS(I,J,K) = VISL + R(I,J,K)*SMPP(I,J,K)**2*
& SQRT(STRAIN)/(BTURB*ABRIPR)

IF(VIS(I,J,K).GT.VISMAX) VIS(I,J,K) = VISMAX
ENDIF
ENDIF

611 CONTINUE

C *** SPECIFY THE VISCOSITY ON THE BOUNDARY POINT
DO 110 I = 1, NI+4
  DO 110 J = 1, NJ+4
    VIS(I,J,NK+3) = VIS(I,J,NK+2)
    VIS(I,J,2) = VIS(I,J,3)
  110 CONTINUE

DO 120 J = 1, NJ+4
  DO 120 K = 1, NK+4
    VIS(NI+3,J,K) = VIS(NI+2,J,K)
    VIS(2,J,K) = VIS(3,J,K)
  120 CONTINUE
DO 130 K=1,NK+4
DO 130 I=1,NI+4
VIS(I,NJ+3,K)=VIS(I,NJ+2,K)
VIS(I,2,K)=VIS(I,3,K)
130 CONTINUE

C *** CALCULATE TURBULENT CONDUCTIVITY
DO 140 I=1,NI+4
DO 140 J=1,NJ+4
DO 140 K=1,NK+4
IF (NOD(I,J,K).NE.1) COND(I,J,K)=VIS(I,J,K)/PRT
140 CONTINUE
RETURN
END

**********************************************************************
**********************************************************************
SUBROUTINE CALW
**********************************************************************
*CALCULATES THE W COMPONENT OF THE VELOCITY

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
& DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,R
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL16/UO,UGRT,BUOY,CPO,PRT,COND,VISO,RHOO,
& TA,DTEMP,TWRITE,TTAPE,TCOLR,GC,RAIR,NT
COMMON/BL20/KSIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
& SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
& JCHPB(20),NCHPB(20),NCHPK(20)
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
& COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
& WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
& U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
& CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
& WPD(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
& SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
& DV(25,25,15),DW(25,25,15)
COMMON/BL36/AN(25,25,15),AF(25,25,15),AN(25,25,15),
& AS(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
& AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
& SU(25,25,15),RI(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
& CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
DO 100 K=3,NK+3
DO 100 J=2,NJ+3
DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE W CONTROL VOLUME
DXP1=DXXC(I+1)
DXI = DXXC(I)
DXM1 = DXXC(I-1)

DYP1 = DYYC(J+1)
DYJ = DYYC(J)
DYM1 = DYYC(J-1)

DZP1 = DZZS(K+1)
DZK = DZZS(K)
DZM1 = DZZS(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
DXN = DXXC(I)
DXS = DXXC(I)
DXF = DXXC(I)
DXB = DXXC(I)
DYF = DYYC(J)
DYB = DYYC(J)
DYE = DYYC(J)
DYW = DYYC(J)
DZE = DZZS(K)
DZW = DZZS(K)
DZN = DZZS(K)
DZS = DZZS(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME
DXEE = DXXS(I+2)
DXE = DXXS(I+1)
DXW = DXXS(I)
DXWW = DXXS(I-1)
DYNN = DYYS(J+2)
DYN = DYYS(J+1)
DYS = DYYS(J)
DYSS = DYYS(J-1)
DZFF = DZZC(K+1)
DZF = DZZC(K)
DZB = DZZC(K-1)
DZBB = DZZC(K-2)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
DXYF = DXF*DYF
DXYB = DXB*DYB
DYZE = DYE*DZE
DYZW = DYW*DZW
DZNX = DZN*DXN
DZXS = DZS*DXS

VOL = DXI*DYJ*DZK
VOLDT = VOL/DTIME

ZXOYN = DZXN/DYN
ZXOYS = DZXS/DYS
XYOZF = DXYF/DZF
XYOZB = DXYB/DZB
YZOXE = DYZE/DXE
YZOXW = DYZW/DXW

C *** USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE
C

PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.

GNF = SILIN(R(I,J+1,K), R(I,J,K), DYP1, DYZ1)*V(I,J+1,K)
GNB = SILIN(R(I,J+1,K-1), R(I,J,K-1), DYP1, DYZ1)*V(I,J+1,K-1)
GSF = SILIN(R(I,J-1,K), R(I,J,K), DYM1, DYZ1)*V(I,J,K)
GSB = SILIN(R(I,J-1,K-1), R(I,J,K-1), DYM1, DYZ1)*V(I,J,K-1)

GF = SILIN(R(I,J,K+1), R(I,J,K), DZF1, DZF1)*W(I,J,K+1)
GP = SILIN(R(I,J,K-1), R(I,J,K), DZF1, DZF1)*W(I,J,K-1)

GEF = SILIN(R(I+1,J,K), R(I,J,K), DXP1, DXP1)*U(I+1,J,K)
GWF = SILIN(R(I-1,J,K-1), R(I,J,K-1), DXP1, DXP1)*U(I-1,J,K-1)
GWB = SILIN(R(I-1,J,K), R(I,J,K), DXP1, DXP1)*U(I-1,J,K)

C *** MASS FLOW RATE

CN = SILIN(GNF, GNB, DZF1, DZF1)*DZXN
CS = SILIN(GSF, GSB, DZF1, DZF1)*DZXS
CE = SILIN(GEF, GEB, DZF1, DZF1)*DYZE
CW = SILIN(GWF, GWB, DZF1, DZF1)*DYZW

CF = 0.5*(GP+GF)*DXYF
CB = 0.5*(GP+GB)*DXYB

C *** VISCOSITY

VISF = VIS(I,J,K)
VISB = VIS(I,J,K-1)
VISN = (VIS(I,J+1,K) + VIS(I,J,K) + VIS(I,J+1,K-1) + VIS(I,J,K-1))/4.
VIS = (VIS(I,J-1,K) + VIS(I,J,K) + VIS(I,J-1,K-1) + VIS(I,J,K-1))/4.

VISN1 = ZXOYN*VISN
VISS1 = ZXOYS*VISS
VISE1 = YZOXE*VISE
VISW1 = YZOXW*VISW
VISF1 = XYOZF*VISF
VISB1 = XYOZB*VISB

C *** QUICK SCHEME

CEP = (ABS(CE) + CE)*DXP1*DXI/(8.*DZE*(DXE+DXW))
CEM = (ABS(CE) - CE)*DXP1*DXI/(8.*DZE*(DXE+DXE))
CWP = (ABS(CW) + CW)*DXM1*DXI/(8.*DXW*(DXW+DXW))
CWM = (ABS(CW) - CW)*DXM1*DXI/(8.*DXW*(DXW+DXE))

CNP = (ABS(CN) + CN)*DYP1*DYJ/(8.*DYN*(DYN+DYS))
CNM = (ABS(CN) - CN)*DYP1*DYJ/(8.*DYN*(DYN+DYN))
CSP = (ABS(CS) + CS)*DYM1*DYJ/(8.*DYS*(DYS+DYSS))
CSM = (ABS(CS) - CS)*DYM1*DYJ/(8.*DYS*(DYS+DYN))

CFP = (ABS(CF) + CF)*DZF/(DZK*16.)
CFM=(ABS(CF)-CF)*DZF/(DZP1*16.)
CBP=(ABS(CB)+CB)*DZB/(DZM1*16.)
CBM=(ABS(CB)-CB)*DZB/(DZK*16.)

AE(I,J,K)=(-.5*CE*DXI+CWM*DXW)/DXE+CEP+CEM*(1.+DXE/DXEE)+VISE1
AW(I,J,K)=(.5*CW*DXI+CEP*DXE)/DXW+CWM+CWP*(I.+DXW/DXWW)+VISW1
AN(I,J,K)=(-.5*CN*DYJ+CSM*DYS)/DYN+CNP+CNM*(1.+DYN/DYNN)+VISN1
AS(I,J,K)=(.5*CS*DYJ+CNP*DYN)/DYS+CSM-$CSP*(I.+DYS/DYSS)+VISS1
AF(I,J,K)= -.5*CF +CBM*DZB /DZF+CFP+CFM*(1.+DZF/DZFF)+VISFI
AB(I,J,K)=.5*CB +CFP*DZF /DZB+CBM+CBP*(1.+DZB/DZBB)+VISB1

C *** BOUNDARY CONSIDERATION
IF (I.LT.NI+3) THEN
AEE=-CEM*DXE/DXEE
AEER=AEE*WPD(I+2,J,K)
ELSE
AEE=0.
AEER=0.
ENDIF

IF (I.GT.2) THEN
AWW=-CWP*DXW/DXWW
AWWR=AWW*WPD(I-2,J,K)
ELSE
AWW=0.
AWWR=0.
ENDIF

IF (J.LT.NJ+3) THEN
ANN=-CNM*DYN/DYNN
ANNR=ANN*WPD(I,J+2,K)
ELSE
ANN=0.
ANNR=0.
ENDIF

IF (J.GT.2) THEN
ASS=-CSP*DYS/DYSS
ASSR=ASS*WPD(I,J-2,K)
ELSE
ASS=0.
ASSR=0.
ENDIF

IF (K.LT.NK+3) THEN
AFF=-CFM*DZF/DZFF
AFFR=AFF*WPD(I,J,K+2)
ELSE
AFF=0.
AFFR=0.
ENDIF

IF (K.GT.3) THEN
ABB=-CBP*DZB/DZBB
ABBR=ABB*WPD(I,J,K-2)
ELSE

112
ABB=0.
ABBR=0.
ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
IF (NOD(I-1,J,K).NE.0) THEN
  AWW=0.0
  AWWR=0.0
ENDIF

IF (NOD(I+1,J,K).NE.0) THEN
  AEE=0.0
  AEER=0.0
ENDIF

IF (NOD(I,J-1,K).NE.0) THEN
  ASS=0.0
  ASSR=0.0
ENDIF

IF (NOD(I,J+1,K).NE.0) THEN
  ANN=0.0
  ANN=0.0
ENDIF

IF (NOD(I,J,K-1).NE.0) THEN
  ABB=0.0
  ABR=0.0
ENDIF

IF (NOD(I,J,K+1).NE.0) THEN
  AEF=0.0
  AFFR=0.0
ENDIF

C **SU FROM NORMAL STRESS
RF=(SIG33(I,J,K)-(W(I,J,K+1)-W(I,J,K))*VISF/DZF)*DXYF
RB=(SIG33(I,J,K-1)-(W(I,J,K)-W(I,J,K-1))*VISB/DZB)*DXYB
RN=(SIG23(I+1,J,K)-(W(I+1,J,K)-W(I,J,K))*VISS/DYS)*DZXS
RS=(SIG23(I,J,K)-(W(I+1,J,K)-W(I,J,K))*VISS/DYS)*DZXS
RE=(SIG13(I+1,J,K)-(W(I+1,J,K)-W(I,J,K))*VISE/DXE)*DYZE
RW=(SIG13(I,J,K)-(W(I,J,K)-W(I-1,J,K))*VISW/DXW)*DYZW

C *** SU FROM CURVED STRESSES AND ACCELERATIONS
AVG23= 0.5*(SIG23(I+1,J,K)+SIG23(I,J,K))
AVG13= 0.5*(SIG13(I+1,J,K)+SIG13(I,J,K))
AVG22=SILIN(SIG22(I,J,K),SIG22(I,J,K-1),DZF,DZB)
AVG11=SILIN(SIG11(I,J,K),SIG11(I,J,K-1),DZF,DZB)
AU3=W(I,J,K)
AU2=BILIN(V(I,J+1,K),V(I,J,K),DYJ,DYJ,
& V(I,J+,K-1),V(I,J,K-1),DYJ,DYJ,DZF,DZB)
AU1=BILIN(U(I+1,J,K),U(I,J,K),DXI,DXI,
& U(I+1,J,K-1),U(I,J,K-1),DXI,DXI,DZF,DZB)
AR=SILIN(R(I,J,K),R(I,J,K-1),DZF,DZB)
ARU23 = AR*AU2 + AU3
ARU13 = AR*AU1 + AU3
ARU22 = AR*AU2 + AU2
ARU11 = AR*AU1 + AU1

RRY = (AVG23 - ARU23)*DXI*(DZN - DZS)
RRX = (AVG13 - ARU13)*DYJ*(DZE - DZW)
RRZ = (AVG22 - ARU22)*DXI*(DYF - DYB) + (AVG11 - ARU11)*DYJ*(DXF - DXB)

AP(I, J, K) = AE(I, J, K) + AW(I, J, K) + AN(I, J, K) + AS(I, J, K)
   & + AF(I, J, K) + AB(I, J, K) + AEE + AWW + ANN + ASS + AFF + ABB
SP(I, J, K) = -(ROD(I, J, K) + DZB + ROD(I, J, K - 1) + DZF)*VOLDT/(DZB + DZF)
SU(I, J, K) = -SP(I, J, K)*WOD(I, J, K) + DXI*DYJ*(P(IJK1) - P(IJK)) +
   & AEER + AWWR + ANNR + ASSR + AFFR + ABBR + RE - RW + RN - RS + RF - RB + RRY +
   & RR - R - RY(R1JK - EQIJK)DB((,) -)
   & REQ(I, J, K - 1)*DZF)*VOL/(DZB + DZF)

100 CONTINUE

C *** TAKE CARE OF B.C. THRU AN, AS, AE, AW, AP AND SU

C *** Y DIRECTION
DO 500 K=3, NK + 3
DO 500 J=2, NJ + 3
   SP(I, 2 , K) = SP(I, 2 , K) - AS(I, 2, K)
   SP(I, NJ + 3, K) = SP(I, NJ + 3, K) - AN(I, NJ + 3, K)
   AS(I, 2, K) = 0.
   AN(I, NJ + 3, K) = 0.
500 CONTINUE

C *** X DIRECTION
DO 502 K=3, NK + 3
DO 502 J=2, NJ + 3
   SP(2 , J, K) = SP(2 , J, K) - AW(2 , J, K)
   SP(NI + 3, J, K) = SP(NI + 3, J, K) - AE(NI + 3, J, K)
   AW(2 , J, K) = 0.0
   AE(NI + 3, J, K) = 0.0
502 CONTINUE

C *** Z DIRECTION
DO 600 I=2, NI + 3
DO 600 J=2, NJ + 3
   AF(I, J, NK + 3) = 0.
   AB(I, J, 3 ) = 0.
600 CONTINUE

C *** MODIFICATION FOR DECK BOUNDARIES
IF (NCHIP.EQ.0) GOTO 201
DO 101 N=1, NCHIP
   IB = ICHPB(N)
   IE = IB + NCHIP(N) - 1
   JB = JCHPB(N)
   JE = JB + NCHIP(N) - 1
   KB = KCHPB(N)
   KE = KB + NCHIP(N) - 1
114
DO 102 J=JB,JE-1
DO 102 K=KB,KE
   SP(IB-1,J,K) = SP(IB-1,J,K) - AE(IB-1,J,K)
   SP(IE,J,K) = SP(IE,J,K) - AW(IE,J,K)
   SU(IB-1,J,K) = SU(IB-1,J,K) + AE(IB-1,J,K)*WFAN(N)*2.0
   SU(IE,J,K) = SU(IE,J,K) + AW(IE,J,K)*WFAN(N)*2.0
   AE(IB-1,J,K) = 0.0
   AW(IE,J,K) = 0.0
102   CONTINUE

DO 103 I=IB,IE-1
DO 103 K=KB,KE
   SP(I,JB-1,K) = SP(I,JB-1,K) - AN(I,JB-1,K)
   SP(I,JE,K) = SP(I,JE,K) - AS(I,JE,K)
   SU(I,JB-1,K) = SU(I,JB-1,K) + AN(I,JB-1,K)*WFAN(N)*2.0
   SU(I,JE,K) = SU(I,JE,K) + AS(I,JE,K)*WFAN(N)*2.0
   AN(I,JB-1,K) = 0.0
   AS(I,JE,K) = 0.0
103   CONTINUE

DO 106 I=IB,IE-1
DO 106 J=JB,JE-1
   SU(I,J,KB-1) = SU(I,J,KB-1) + AF(I,J,KB-1)*WFAN(N)
   SU(I,J,KE+1) = SU(I,J,KE+1) + AB(I,J,KE+1)*WFAN(N)
   AF(I,J,KB-1) = 0.0
   AB(I,J,KE+1) = 0.0
106   CONTINUE

C *** FOR THE CELLS INSIDE OF THE DECKS
DO 104 I=IB,IE-1
DO 104 J=JB,JE-1
DO 104 K=KB,KE
   SP(I,J,K) = -1.0E2
   AW(I,J,K) = 0.
   AE(I,J,K) = 0.
   AS(I,J,K) = 0.
   AN(I,J,K) = 0.
   AB(I,J,K) = 0.
   AF(I,J,K) = 0.
   SU(I,J,K) = 1.0E2*WFAN(N)
104   CONTINUE
101   CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
201   DO 301 K=3,NK+3
   DO 301 J=2,NJ+3
   DO 301 I=2,NI+3
      DXI = DXXC(I)
      DYJ = DYYC(J)
      DXY = DXI*DYJ
      AP(I,J,K) = AP(I,J,K) - SP(I,J,K)
      DW(I,J,K) = DXY/AP(I,J,K)
301   CONTINUE

C *** SOLVE FOR W
   CALL TRID (3,3,4,NI+2,NJ+2,NK+2,W)
C *** RESET THE VELOCITY INSIDE OF THE DECKS
IF (NCHIP.EQ.0) GOTO 111
DO 110 N=1,NCHIP
   IB=ICHPB(N)
   IE=IB+NCHPI(N)-1
   JB=JCHPB(N)
   JE=JB+NCHPJ(N)-1
   KB=KCHPB(N)
   KE=KB+NCHPK(N)-1
   DO 108 I=IB,IE-1
   DO 108 J=JB,JE-1
   DO 108 K=KB,KE
      W(I,J,K)=WFAN(N)
108   CONTINUE
110   CONTINUE
111   RETURN
END

***********************************************************************
***********************************************************************
SUBROUTINE GLOBE
***********************************************************************
* THIS SUBROUTINE CALCULATES THE GLOBAL PRESSURE CORRECTION, WHEREBY THE
* PRESSURE MATRIX IS UPDATED.
* VARIABLES USED ARE:
* SUMT = SUM OF TEMPERATURES
* SUMPT = SUM OF PRESSURE OVER TEMPERATURE
* SUMPET = SUM OF EQUILIBRIUM PRESSURE OVER TEMP
* UGRT = CONSTANT (FROM SUBROUTINE INIT)
* PCORR = PRESSURE CORRECTION
***********************************************************************
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
 & DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL16/UO,UGRT,BUOY,CPO,PRT,COND0,VISO,RHOO,
 & TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
 & U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
 & SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
 & DV(25,25,15),DW(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
 & CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
SUMT=0.
SUMPT=0.
SUMPET=0.
DO 370 I=3,NI+2
DO 370 J=3,NJ+2
DO 370 K=3,NK+2
   IF (NOD(I,J,K).NE.1) THEN
DIRIX = DXXC(I)
DYJ = DYYC(J)
DZK = DZZC(K)
VOL = DXI*DYJ*DZK
SUMT = SUM+VOL/T(I,J,K)
SUMPT = SUMPT+P(I,J,K)*VOL/T(I,J,K)
SUMPET = SUMPET+REQ(I,J,K)*VOL*(1./T(I,J,K))

ENDIF
CONTINUE
SUMPET = SUMPET/UGRT
PCORR = (SUMPET-SUMPT)/SUMT

DO 371 I=1,NI+4
DO 371 J=1,NJ+4
DO 371 K=1,NK+4
P(I,J,K) = P(I,J,K)+PCORR
371 CONTINUE

RETURN
END

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

SUBROUTINE GRID

*NONDIMENSIONAL VARIABLES:
* GRID SIZES:
* DX = X DIRECTION
* DY = Y DIRECTION
* DZ = Z DIRECTION
*
* CENTRAL CELLS:
* XC() = X COORDINATE
* YC() = Y COORDINATE
* ZC() = Z COORDINATE
* DXXC() = X LENGTH
* DYYC() = Y LENGTH
* DZZC() = Z LENGTH
*
* STAGGERED CELLS:
* XS() = X COORDINATE
* YS() = Y COORDINATE
* ZS() = Z COORDINATE
* DXXS() = X LENGTH
* DYYS() = Y LENGTH
* DZZS() = Z LENGTH

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

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),Y(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWAL
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP

117
**GENERATION OF THE GRIDS**

\[ DX = \frac{X}{(DFLOAT(NI)*H)} \]
\[ DY = \frac{Y}{(DFLOAT(NJ)*H)} \]
\[ DZ = \frac{H}{(DFLOAT(NK)*H)} \]

**CALCULATE XS,YS,ZS (COORDINATES OF STAGGERED CV'S)**

```
DO 10 I=3,NI+3
   XS(I)=(I-3)*DX
10 CONTINUE
XS(2)=XS(3)-TWAL/(H*12.)
XS(1)=XS(2)-TWAL/(H*12.)
XS(NI+4)=XS(NI+3)+TWAL/(H*12.)
XS(NI+5)=XS(NI+4)+TWAL/(H*12.)
```

```
DO 12 J=3,NJ+3
   YS(J)=(J-3)*DY
12 CONTINUE
YS(2)=YS(3)-TWAL/(H*12.)
YS(1)=YS(2)-TWAL/(H*12.)
YS(NJ+4)=YS(NJ+3)+TWAL/(H*12.)
YS(NJ+5)=YS(NJ+4)+TWAL/(H*12.)
```

```
DO 14 K=3,NK+3
   ZS(K)=(K-3)*DZ
14 CONTINUE
ZS(2)=ZS(3)-TFLR/(H*12.)
ZS(1)=ZS(2)-TFLR/(H*12.)
ZS(NK+4)=ZS(NK+3)+TFLR/(H*12.)
ZS(NK+5)=ZS(NK+4)+TFLR/(H*12.)
```

**CALCULATE DXXC,DYYC AND DZZC (DIMENSIONS OF CENTERED CV'S)**

```
DO 20 I=1,NI+4
   DXXC(I)=XS(I+1)-XS(I)
20 CONTINUE
DXXC(NI+5)>DXXC(NI+4)
```

```
DO 22 J=1,NJ+4
   DYYC(J)=YS(J+1)-YS(J)
22 CONTINUE
DYYC(NJ+5)=DYYC(NJ+4)
```

```
DO 24 K=1,NK+4
   DZZC(K)=ZS(K+1)-ZS(K)
24 CONTINUE
DZZC(NK+5)=DZZC(NK+4)
```

**CALCULATE DXXS,DYYS,DZZS (DIMENSIONS OF STAGGERED CV'S)**

```
DO 30 I=2,NI+5
   DXXS(I)=(DXXC(I)+DXXC(I-1))/2.0
30 CONTINUE
DXXS(1)=DXXS(2)
```

```
DO 32 J=2,NJ+5
   DYYS(J)=(DYYC(J)+DYYC(J-1))/2.0
32 CONTINUE
DYYS(1)=DYYS(2)
```
DO 34 K=2,NK+5
   DZZS(K)=(DZZC(K)+DZZC(K-1))/2.0
34 CONTINUE
   DZZS(1)=DZZS(2)

C *** CALCULATE XC,YC,ZC (LOCATION OF CENTER CELLS)
DO 40 I=1,NI+5
   XC(I)=XS(I)+DXXC(I)/2.0
40 CONTINUE

DO 42 J=1,NJ+5
   YC(J)=YS(J)+DYYC(J)/2.0
42 CONTINUE

DO 44 K=1,NK+5
   ZC(K)=ZS(K)+DZZC(K)/2.0
44 CONTINUE
RETURN
END

************************************************************
************************************************************
SUBROUTINE INIT
************************************************************
*THIS SUBROUTINE INITIALIZES THE FIELD AND CONSTANTS WITH RESPECT
*TO INITIAL START OR RESTARTING CAPABILITY.
*
*VARIABLES ARE :
* ALEW = LEWIS NUMBER (USED IN SMOKE CONCENTRATION CALCULATIONS)
* BUOY = BUOYANCY FORCE CONSTANT
* CO = INITIAL SMOKE CONCENTRATION
* CONDO = REFERENCE CONDUCTIVITY
* CONSR = NONDIMENSIONAL RADIATION CONSTANT
* CPO = REFERENCE SPECIFIC HEAT
* F = INITIAL MASS OF FUEL (LBM)
* FR = MASS OF FUEL REMAINING (LBM)
* GC = GRAVITY CONSTANT
* H = CHARACTERISTIC LENGTH; HEIGHT OF CHAMBER=10.FT
* HOEF = DIMENSIONLESS HEAT TRANSFER COEF
* HCONV = HEAT TRANSFER COEFFICIENT IN BTU/(HR*FT**2*DEGREES)
* HR = HEIGHT IN CM
* NTAPE = NONDIMENSIONAL FORMS OF TTAPE
* NWRITE = NONDIMENSIONAL FORMS OF TWRITE
* RHOO = REFERENCE DENSITY
* TA = TEMP IN DEGREES RANKINE
* TIME = DIMENSIONLESS TIME
* TR = TEMP IN DEGREES KELVIN
* U0 = CHARACTERISTIC VELOCITY (1 FT/SEC)
* UGRT = PERFECT GAS LAW NONDIMENSIONAL CONSTANT
* VISO = REFERENCE VISOSITY (NONDIM)
* VISL = MINIMUM VISOSITY (NONDIM)
* VISMAX = MAXIMUM VISOSITY (NONDIM)
*
*MATRICES OF THE FORM
* _OD  = DIMENSIONLESS PARAMETER AT PREVIOUS TIME STEP
* _PD  = DIMENSIONLESS PARAMETER AT CURRENT TIME STEP
* _P  = DIMENSIONLESS PARAMETER AT NEXT TIME STEP

*WHERE THE PARAMETERS ARE
* AP  = COEFFICIENT AT NODE P
* AE,AW,AN = COEFFICIENTS AT PTS EAST,WEST,NORTH,
* AS,AF,AB = SOUTH, FRONT, AND BACK
* CPM  = MEAN SPECIFIC HEAT
* COND( ) = CONDUCTIVITY MATRIX
* CX,CY,CZ = LOCATION OF THERMOCOUPLE IN X,Y,Z
* DU,DV,DW = USED IN PRESSURE CORRECTION SUBROUTINE
* DXXC,DYYC = LENGTH AROUND THE CENTER CELL
* DZZC
* DXXS,DYYS = LENGTH AROUND THE STAGGERED CELL
* DZZS
* NOD  = IF EQUAL TO ZERO, LIQUID; IF EQUAL TO ONE, SOLID
* PP  = CORRECTED PRESSURE (P')
* REQ  = DENSITY AT EQUILIBRIUM
* SMP  = RESIDUAL MASS SUMMATION OF NODAL POINT
* SMP  = RESIDUAL MASS SUMMATION OF NODAL POINT
* SP  = BOUNDARY CONDITION TERM AT NODE P
* SU  = SOURCE TERM
* T,P,C = TEMP, PRESSURE, AND SMOKE CONCENTRATION
* U,V,W = VELOCITY COMPONENTS IN X,Y,X DIRECTIONS
* VIS  = VISCOSITY
* B,E  = BEGINNING AND ENDING NODAL POINT FOR
* THE SOLID IN I,J,K
* XC,YC,ZC = X,Y,Z LOCATION OF CENTER CELL NODAL POINT
* XS,YS,ZS = X,Y,Z LOCATION OF STAGGERED CELL NODAL POINT

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

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
&   DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWAL
COMMON/BL3/FR,F,HSTART
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL12/NWRITE,NATAPE,NTMAX0,NREAL,TIME,SORSUM,ITER
COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
COMMON/BL16/UO,UGRT,BUOY,CP0,PR,COND0,VIS0,RHO0,
&   TA,DTEMP,TWRITE,TAPE,TMAX,GC,RAIR,NT
COMMON/BL20/SI11(25,25,15),SI12(25,25,15),SI122(25,25,15),
&   SIG13(25,25,15),SIG133(25,25,15),SIG1333(25,25,15),
COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
&   JCHPB(20),NCHPBJ(20),NCHPK(20)
COMMON/BL23/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
&   COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
&   WOD(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
&   U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
&   CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
&   WPD(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
& SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
& DV(25,25,15),DW(25,25,15)
COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
& AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
& SU(25,25,15),RI(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
& CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
COMMON/BL38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO
COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN
& WAIR,WWAL,WINS,WERR,WWFAN

C *** INITIALIZE GIVEN PARAMETERS
C0=0.0
F=200.0
HCONV=15.0
NBLOR=13
PI=4.*ATAN(1.)
TCOOL=1.0

C *** NONDIMENSIONALIZE THE REFERENCE VISCOSITY
VISO=VISO/(U0*H)
VISL=VISO

C *** SET MAXIMUM VISCOSITY
VISMAX=400.*VISL

C *** NONDIMENSIONALIZE THE HEAT TRANSFER COEFFICIENT
HCOEF=HCONV/(3600.*CP0*RHOO*U0)

CONDO=VISO/PRT
FR=F
BUOY=GC*H/(U0**2)
UGRT=U0**2/(GC*RAIR*TA)
CONSRA=1.714E-9*TA**3/(RHOO*CP0*U0*3600.)

C *** FOR ENERGY DISTRIBUTION
QSIN=0.
QSWER=0.
QSWAL=0.
QSAIR=0.
QSFAN=0.

C *** INITIALIZE CONDUCTION HEAT FLUX TO EACH WALL
QSCONF=0.
QSCONB=0.
QSCONE=0.
QSCONW=0.
QSCONN=0.
QSCONS=0.

C *** INITIALIZE RADIATION HEAT FLUX TO EACH WALL
QSRADF=0.
QSRADB=0.
QSRADE=0.
QSRADW=0.
QSRADN=0.
QSRADS=0.

NWRITE=TWRITE*U0/(DTIME*H)
NTAPE=TTAPE*U0/(DTIME*H)

C *** INITIALIZE VARIABLE FIELDS
DO 220 J=1,NJ+4
DO 220 I=1,NI+4
DO 220 K=1,NK+4

IF(KRUN.LE.0) THEN
  UOD(I,J,K) =0.
  VOD(I,J,K) =0.
  WOD(I,J,K) =0.
  POD(I,J,K) =0.
  TOD(I,J,K) =TA/TA
  COD(I,J,K) =CO
ENDIF

U(I,J,K) =UOD(I,J,K)
UPD(I,J,K) =UOD(I,J,K)
V(I,J,K) =VOD(I,J,K)
VPD(I,J,K) =VOD(I,J,K)
W(I,J,K) =WOD(I,J,K)
WPD(I,J,K) =WOD(I,J,K)
P(I,J,K) =FOD(I,J,K)
PPD(I,J,K) =POD(I,J,K)
T(I,J,K) =TOD(I,J,K)
TPD(I,J,K) =TOD(I,J,K)
C(I,J,K) =COD(I,J,K)
CPD(I,J,K) =COD(I,J,K)

DU(I,J,K) =0.
DV(I,J,K) =0.
DW(I,J,K) =0.
SU(I,J,K) =0.
SP(I,J,K) =0.
PP(I,J,K) =0.
AP(I,J,K) =0.
AW(I,J,K) =0.
AE(I,J,K) =0.
AN(I,J,K) =0.
AS(I,J,K) =0.
AF(I,J,K) =0.
AB(I,J,K) =0.
SMP(I,J,K) = 0.
SMPP(I,J,K) = 0.

SIG11(I,J,K) = 0.
SIG12(I,J,K) = 0.
SIG13(I,J,K) = 0.
SIG22(I,J,K) = 0.
SIG23(I,J,K) = 0.
SIG33(I,J,K) = 0.

VIS(I,J,K) = VISL
COND(I,J,K) = CONDO
CPM(I,J,K) = 1.0E0
NOD(I,J,K) = 0

CONTINUE

C *** DEFINE THERMAL PROPERTIES OF DECK AND SOLID
IF (NCHIP.NE.0) CALL SOLCON

C *** DEFINE HEIGHT OF NODE POINTS AND COMPUTE HYDROSTATIC EQUILIBRIUM DENSITY Req(I,J,K)
15 DO 229 J=1,NJ+4
   DO 229 I=1,NI+4
   DO 229 K=1,NK+4
      HEIGHT(I,J,K) = ZC(K)
      REQ(I,J,K) = EXP(-BUOY*UGRT*HEIGHT(I,J,K))
      IF (KRUN.LE.0) THEN
         ROD(I,J,K) = REQ(I,J,K)/TPD(I,J,K)
      ENDIF
      R(I,J,K) = ROD(I,J,K)
      RPD(I,J,K) = ROD(I,J,K)
229 CONTINUE

C *** FOLLOWING IS FOR DETERMINING THE THERMOCOUPLE POSITIONS
DO 5000 N=1,NTHCO
   DO 5001 I=1,NI+4
      IF (XC(I).LT.CX(N).AND.XC(I+1).GE.CX(N)) GOTO 5002
   5001 CONTINUE

   II=I
   DO 5003 J=1,NJ+4
      IF (YC(J).LT.CY(N).AND.YC(J+1).GE.CY(N)) GOTO 5004
   5003 CONTINUE

   JJ=J
   DO 5005 K=1,NK+4
      IF (ZC(K).LT.CZ(N).AND.ZC(K+1).GE.CZ(N)) GOTO 5006
   5005 CONTINUE

   KK=K
   NTH(N,1)=II
   NTH(N,2)=JJ
   NTH(N,3)=KK

5000 CONTINUE

RETURN
SUBROUTINE INPUT(NSTOP)

* THIS SUBROUTINE SETS UP REQUIRED VALUES TO BEGIN THE PROGRAM.

* VARIABLES ARE:
*  KRUN = RESTART INDICATOR
*  NCHIP = NUMBER OF INTERNAL SOLID PIECES
*  NMS = NUMBER OF MASS SOURCES
*  NWRP = NUMBER OF TIME STEPS BETWEEN WRITES TO OUTPUT FILE
*  NTHCO = NUMBER OF THERMOCOUPLES TO PRINT OUT
*  TMAX = NONDIMENSIONAL MAXIMUM TIME ALLOWED
*  XMAX = MAXIMUM TIME ALLOWED (SECONDS)
*  TWRITE = TIME BETWEEN FIELD VARIABLE OUTPUT (SECONDS)
*  TTAP = TIME INTERVAL BETWEEN PLOTS (SECONDS)
*  DTIME = NONDIMENSIONAL TIME STEP
*  XDTIME = TIME STEP (SECONDS)
*  HSTART = FIRE START TIME (SECONDS)
*  NHSZ(1,1) = STARTING NODE OF HEAT SOURCE, X-DIR
*  NHSZ(1,2) = ENDING NODE OF HEAT SOURCE, X-DIR
*  NHSZ(2,1) = Y-DIR
*  NHSZ(2,2) = Y-DIR
*  NHSZ(3,1) = Z-DIR
*  NHSZ(3,2) = Z-DIR
*  ICHPB = FIRST NODE OF INTERNAL SOLID IN X DIR
*  JCHPB = Y DIR
*  JCHPB = Z DIR
*  NCHPB = NUMBER OF INTERNAL SOLID NODES IN X DIR
*  JCHPB = Y DIR
*  KCHPB = Z DIR
*  NCHPK = NUMBER OF INTERNAL SOLID NODES IN Y DIR
*  JCHPK = Y DIR
*  NCHPK = Z DIR
*  IMSB = FIRST MASS SOURCE NODE IN X DIR
*  JMSB = Y DIR
*  KMSB = Z DIR
*  NMSB = NUMBER OF MASS SOURCE NODES IN X DIR
*  JMSB = Y DIR
*  KMSB = Z DIR
*  RMS = DIMENSIONLESS MASS SOURCE
*  (= CFM/(60.*H**2*UO*NMSI*NMSJ*NMSK))
*  CX, CY, CZ = THERMOCOUPLE POSITIONS IN X, Y, Z

* DATA FILES USED IN THIS PROGRAM:
*  FILE # 10 = FIRE.DAT : INITIAL SET-UP DATA
*  11 = FIRE1.CONT : RESTART/CONTINUATION DATA
*

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWAL
COMMON/BL3/F,FR,HSTART
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL12/NWRITE,NTAPE,TMAXO,NTREAL,TIME,SORSUM,ITER
COMMON/BL16/U0,UGRT,BUOY,CPO,PRT,CONDO,VISO,RHOO,  
& TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT  
COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),  
& JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)  
COMMON/BL23/RMS(20),NMS,IMSB(20),NMSI(20),JMSB(20),NMSJ(20),  
& KMSB(20),NMSK(20)  
COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),  
& COD(25,25,15),UOD(25,25,15),VOD(25,25,15),  
& WOD(25,25,15)  
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),  
& CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)  
COMMON/BL38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO  

CHARACTER ANS*1  
LOGICAL L1,L2  
NSTOP=0  
KRUN=0  

***** CHECK FOR INPUT DATA FILE  
INQUIRE (FILE='/FIRE DATA B1',EXIST=L1)  
IF (L1) THEN  

C *** READ IN DATA FROM EXISTING DATA FILE  
OPEN(10,FILE='/FIRE DATA B1',STATUS='OLD')  
REWIND(10)  
READ(10,*) X,Y,H,TFLR,TWAL,TA  
READ(10,*) NI,NJ,NK  
READ(10,*) NCHIP,NMS,NWRP,NTHCO  
READ(10,*) TMAX,DTIME,TTAPE,TWRITE,HSTART  
READ(10,*) NHSZ(1,1),NHSZ(1,2),NHSZ(2,1),NHSZ(2,2),NHSZ(3,1),  
& NHSZ(3,2)  
IF (NCHIP.LE.0) GOTO 33  
DO 32 N=1,NCHIP  
    READ(10,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),  
& NCHPK(N),CPS(N),CONS(N),WFAN(N)  
32 CONTINUE  
33 IF (NMS.LE.0) GOTO 37  
DO 36 N=1,NMS  
    READ(10,*) IMSB(N),NMSI(N),JMSB(N),NMSJ(N),KMSB(N),  
& KMSK(N),RMS(N)  
36 CONTINUE  
37 DO 38 I=1,NTHCO  
    READ (10,*) CX(I),CY(I),CZ(I)  
38 CONTINUE  
REWIND(10)  
CLOSE(10)  
ELSE  

C *** STOP PROGRAM IF INPUT DATA NOT AVAILABLE  
NSTOP=9999  
GOTO 999  
ENDIF  

***** CHECK FOR CONTINUATION FILE  
INQUIRE (FILE='/CONTINUE DATA B4',EXIST=L2)  
IF (L2) THEN  

125
C *** READ IN DATA FROM OLD CONTINUATION FILE
OPEN(11,FILE='/CONTINUE DATA B4',STATUS='OLD',
&
  FORM='UNFORMATTED')
  KRUN=1
  REWIND(11)
  READ(11) TIME,NTMAXO,FR,TOD,ROD,UOD,VOD,WOD,POD,COD
  REWIND(11)
  IF(TIME.GE.TMAX) TMAX=TIME+TMAX
ELSE
C *** CREATE NEW CONTINUATION FILE
OPEN(11,FILE='/CONTINUE DATA B4',STATUS='NEW',
 &
  FORM='UNFORMATTED')
  KRUN=0
ENDIF
999 RETURN
END

******************************************************************************
******************************************************************************
SUBROUTINE OUT(NN)
******************************************************************************
*THIS SUBROUTINE GENERATES OUTPUT.
*
* NN = 1 SELECTED VALUES ARE PRINTED. INCLUDING TIME, ERROR,
* PRESSURE, HEAT GENERATION
* NN = 2 TEMPERATURE AT THE THERMOCOUPLES
* NN = 3 FILED VALUES ARE PRINTED
* NN = 4 ENERGY DISTRIBUTION
******************************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TLR,TLW
COMMON/BL7/NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
COMMON/BL16/UO,UGRT,BUOY,CPO,PRT,COND0,VI0,RH00,
  T,DTM,DTWIRE,NTAPE,TMAX,GC,RAIR,NT
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
  U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
  SMP(25,25,15),PP(25,25,15),DU(25,25,15),
  DV(25,25,15),DW(25,25,15)
COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
  AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
  SU(25,25,15),RI(25,25,15)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
  CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
COMMON/BL38/TCOUP(30),GX(30),CY(30),CZ(30),NTH(30,3),NTHCO
COMMON/BL39/ALEW,CONSR, QWIN, QSWEQ, QSWAL, QSAIR, QSFAN
COMMON/BL43/QSCONF, QSCONB, QSCONE, QSCONW, QSCONN, QSCONS, QSRADF,
  QSRADB, QSRADQ, QSRADW, QSRADN, QSRADWN, WAIR, WWAL, WINS,
  WERR, WWFN

126
**REFERENCE TEMPERATURE IN DEGREES K**
\[ T_R = \frac{TA}{1.8} \]

**REFERENCE VELOCITY IN CM/SEC**
\[ U_R = U_0 \times 30.48 \]

**REFERENCE LENGTH IN CM**
\[ H_R = H \times 30.48 \]

\[ XTIME = TIME \times H/U_0 \]

**IF (NN.EQ.1) THEN**
\[ QRR = 60 \times 2^2 \times QR/3412. \]
\[ QKW = 60 \times 2^2 \times Q/3412. \]

WRITE(12,500) XTIME,NTREAL,ITER,RESORM(ITER),SORSUM,QKW
ELSE IF (NN.EQ.2) THEN
  WRITE (12,*) ' TEMPERATURES AT THERMOCOUPLE POSITION IN (C):',
  (TCOUP(N)*TR-273.16,N=1,NTHCO)
&
ELSE IF (NN.EQ.3) THEN
  WRITE (12,'(1X,A,F10.6)') 'TIME =',XTIME
  DO 501 J=3,NJ+3,NJ
    DO 502 K=2,NK+4
      WRITE (12,504) J,K
      513 DO 503 I=1,NI+4
        IF (T(I,J,K).LT.TCOOL) T(I,J,K)=TCOOL
        XTEMP=T(I,J,K)*TR-273.16
        XR =1000.*(0.0328)**3*R(I,J,K)*RH00/2.2048
        XU =U(I,J,K)*UR
        XV =V(I,J,K)*UR
        XW =W(I,J,K)*UR
        XP =P(I,J,K)*RH00*U0**2/(GC*14.696*144.)+REQ(I,J,K)
        XVIS =VIS(I,J,K)*HR:*UR
        XCOND=COND(I,J,K)*HR*UR
      WRITE(12,511)1,XTEMP,XU,XV,XW
      503 CONTINUE
    502 CONTINUE
  501 CONTINUE
ELSE

C *** CALCULATE THE PERCENTAGE AND PRINT OUT THE RESULTS
QTCON=QSCONF+QSCONB+QSCONW+QSCONE+QSCONS+QSCONN+QSCONB+QSCONE
QTRAD=QSRADF+QSRADB+QSRADS+QSRADN+QSRADW+QSRAD
C *** WATT PERCENTAGE
  IF (WINS.EQ.0.) WINS=1.0E-5
  PAIR=100.*WAIR /WINS
  PWAL=100.*WWAL /WINS
  PFAN=100.*WFAN/WINS
  PWER=100.*WERR /WINS
C *** ENERGY PERCENTAGE
  IF (QSIN.EQ.0.0) QSIN=1.0E-3
  PSAIR=100.*QSAIR/QSIN
  PSWAL=100.*QSWAL/QSIN
  PSFAN=100.*QSFAN/QSIN
  PWER=100.*QWER/QSIN
  WRITE(12,1084)XTIME,WINS,WAIR,WFAN,WWAL,WERR,QTCON,QTRAD
  & WRITE(12,2728)QSRADN,QSRADQ,QSRADS,QSRADD,QSRADF,QSRAD
  & WRITE(12,1088)PAIR,PWER,PWAL,PFAN,PWER,PSWAL
  WRITE(12,1091)QSIN,QSWAR,QSAIR,QSFAN,QSWAL
ENDIF

RETURN
END
SUBROUTINE RADHT (NN)

* NN = CONTROL PARAMETER FOR HEAT FLUX CALCULATIONS
* WHERE NN=1 :CALCULATE HEAT FLUX FROM FIRE TO WALLS
* NN=2 :CALCULATE HEAT FLUX FROM FIRE TO BLOCKS
* NTHS = TOTAL NUMBER OF CV'S CONTAINING HEAT SOURCES
* NFX,NFY,NFZ = CV NUMBER OF HEAT SOURCE
* FX,FY,FZ = COORDINATES OF HEAT SOURCE

**IMPLICIT DOUBLE PRECISION (A-H,O-Z)**
**COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),**
& DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40),DZZS(40)
**COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,RQ**
**COMMON/BL7/NI,NJ,NK, KRUN,NBLOR,NWRP**
**COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),**
& JCHPB(20),NCHPK(20),KCHPB(20),NCHIP(20)
**COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),**
& U(25,25,15),V(25,25,15),W(25,25,15)
**COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),**
& AS(25,25,15),AF(25,25,15),SP(25,25,15),RI(25,25,15)
**COMMON/BL37/VS(25,25,15),CON(25,25,15),RESOR(40),**
& CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
**COMMON/BL39/ALEW,CONSRA,QSIN,QSVAL,QSAIR,QSFAN**
**COMMON/BL40/VFHSW(5,25,15),VFHSE(5,25,15),VFHSS(5,15,25),**
**COMMON/BL41/VFHSBW(5,8,34,34),VFHSBE(5,8,34,34),VFHSBS(5,8,34,34),**
& VFHSBB(5,8,34,34),VFHSSF(5,8,34,34)

NTHS=NHSZ(3,2)-NHSZ(3,1)+1
DO 500 N=1,NTHS
  NFX=NHSZ(1,1)
  NFY=NHSZ(2,1)
  NFZ=NHSZ(3,1)-1+N

C *** AREA OF THE FIRE ELEMENTS
  DYHS=DYYC(NFY)
  DZHS=DZZC(NFZ)
  DAHS=PI*DYHS*DZHS
  EMIS=0.6

C *** NN=1: CALCULATE RADIATION HEAT FLUX FROM FIRE TO WEST AND
C *** EAST SURFACES OF THE ENCLOSURE
  IF (NN.EQ.1) THEN
    DO 100 J=3,NJ+2
    DO 100 K=3,NK+2
      SU(2,J,K) =SU(2,J,K)+CONSRA*EMIS*DAHS*VFHSW(N,J,K)*
      (T(NFX,NEY,NFZ)**4-T(2,J,K)**4)
      SU(NI+3,J,K)=SU(NI+3,J,K)+CONSRA*EMIS*DAHS*VFHSE(N,J,K)*
      (T(NFX,NEY,NFZ)**4-T(NI+3,J,K)**4)
    100 CONTINUE

C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO NORTH AND
C *** SOUTH SURFACES OF THE ENCLOSURE

129
DO 200 I=3,NI+2
  DO 200 K=3,NK+2
  SU(I,2,K) =SU(I,2,K)+CONSRA*EMIS*DAHS*VFHSS(N,K,I)*
             (T(NFX,NFY,NFZ)**4-T(I,2,K)**4)
  & SU(I,NJ+3,K)=SU(I,NJ+3,K)+CONSRA*EMIS*DAHS*VFHSN(N,K,I)*
             (T(NFX,NFY,NFZ)**4-T(I,NJ+3,K)**4)
  200 CONTINUE

C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO BACK AND
C *** FRONT SURFACES OF THE ENCLOSURE
DO 300 I=3,NI+2
  DO 300 J=3,NJ+2
  SU(I,J,2) =SU(I,J,2)+CONSRA*EMIS*DAHS*VFHSB(N,I,J)*
             (T(NFX,NFY,NFZ)**4-T(I,J,2)**4)
  & SU(I,J,NK+3)=SU(I,J,NK+3)+CONSRA*EMIS*DAHS*VFHSF(N,I,J)*
             (T(NFX,NFY,NFZ)**4-T(I,J,NK+3)**4)
  300 CONTINUE
ENDIF
C *** NN=2: CALCULATE RADIATION HEAT FLUX FROM FIRE TO WEST AND
C *** EAST SURFACES OF BLOCK M
IF (NN.EQ.2) THEN
  IF (NCHIP.LT.NBLOR) THEN
    DO 900 M=1,NCHIP-NBLOR+1
      IB=ICHPB(M+NBLOR-1)
      IE=IB+NCHPI(M+NBLOR-1)-1
      JB=JCHPB(M+NBLOR-1)
      JE=JB+NCHPJ(M+NBLOR-1)-1
      KB=KCHPB(M+NBLOR-1)
      KE=KB+NCHPK(M+NBLOR-1)-1
  C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO WEST AND
  C *** EAST SURFACES OF THE BLOCK
    DO 400 J=JB,JE-1
      DO 400 K=KB,KE-1
      SU(IB,J,K) =SU(IB,J,K)+CONSRA*EMIS*DAHS*
                   VFHSBW(N,M,J,K)*(T(NFX,NFY,NFZ)**4-
                                      T(IB,J,K)**4)
      & SU(IE-1,J,K)=SU(IE-1,J,K)+CONSRA*EMIS*DAHS*
                   VFHSBE(N,M,J,K)*(T(NFX,NFY,NFZ)**4-
                                      T(IE-1,J,K)**4)
      400 CONTINUE
  C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO NORTH AND
  C *** SOUTH SURFACES OF THE BLOCK
    DO 600 I=IB,IE-1
      DO 600 K=KB,KE-1
      SU(I,JB,K) =SU(I,JB,K)+CONSRA*EMIS*DAHS*
                   VFHSBS(N,H,K,I)*(T(NFX,NFY,NFZ)**4-
                                      T(I,JB,K)**4)
      & SU(I,JE-1,K)=SU(I,JE-1,K)+CONSRA*EMIS*DAHS*
                   VFHSBN(N,H,K,I)*(T(NFX,NFY,NFZ)**4-
                                      T(I,JE-1,K)**4)
      600 CONTINUE

130
C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO BACK AND
C *** FRONT SURFACES OF BLOCK
DO 700 I=IB,IE-1
DO 700 J=JB,JE-1
SU(I,J,KB) =SU(I,J,KB)+CONSRA*EMIS*DAHS*
& VFHSBB(N,M,I,J)*(T(NFX,NFY,NFZ)**4-
& T(I,J,KB)**4)
SU(I,J,KE-1)=SU(I,J,KE-1)+CONSRA*EMIS*DAHS*
& VFHSBF(N,M,I,J)*(T(NFX,NFY,NFZ)**4-
& T(I,J,KE-1)**4)
700 CONTINUE
900 CONTINUE
ENDIF
ENDIF
500 CONTINUE
RETURN
END

*******************************************************************************
*******************************************************************************
SUBROUTINE SOLCON
*******************************************************************************
* THIS SUBROUTINE RESETS THE CONDUCTIVITY OF THE SOLID. IN CALVIS *
* THE VISCOSITY ARE CALCULATED AT EVERY CELL INCLUDING THOSE *
* CONTAINING SOLID ONES. *
*******************************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL16/UO,UGRT,BUOY,CP0,PRT,COND0,VISO,RHOO,
& TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
COMMON/BL22/CPSC20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHIP(20),
& JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
& CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

DO 402 N=1,NCHIP
IB=ICHPB(N)
IE=IB+NCHIP(N)-1
JB=JCHPB(N)
JE=JB+NCHIPJ(N)-1
KB=KCHPB(N)
KE=KB+NCHPK(N)-1
DO 405 I=IB,IE-1
DO 405 J=JB,JE-1
DO 405 K=KB,KE-1
COND(I,J,K)=COND0*CONS(N)
CPM(I,J,K)=CP0(N)
NOD(I,J,K)=1

C *** SET VALUE AT CORNER OR BOUNDARIES FOR BOUNDARY CONDITIONS
IF (I.EQ.2) THEN
COND(1,J,K)=COND(2,J,K)
CPM (1,J,K)=CPM (2,J,K)
ELSEIF (I.EQ.NI+3) THEN

131
COND(NI+4,J,K)=COND(NI+3,J,K)
CPM(NI+4,J,K)=CPM(NI+3,J,K)
ENDIF

IF (J.EQ.2) THEN
  COND(I,1,K)=COND(I,2,K)
  CPM(I,1,K)=CPM(I,2,K)
ELSEIF (J.EQ.NJ+3) THEN
  COND(I,NJ+4,K)=COND(I,NJ+3,K)
  CPM(I,NJ+4,K)=CPM(I,NJ+3,K)
ENDIF

IF (K.EQ.2) THEN
  COND(I,J,1)=COND(I,J,2)
  CPM(I,J,1)=CPM(I,J,2)
ELSEIF (K.EQ.NK+3) THEN
  COND(I,J,NK+4)=COND(I,J,NK+3)
  CPM(I,J,NK+4)=CPM(I,J,NK+3)
ENDIF

405 CONTINUE
402 CONTINUE
RETURN
END

****************************************************************************
******************************************************************************
** SUBROUTINE STRESS **
******************************************************************************
*THIS SUBROUTINE CALCULATES THE SHEAR STRESSES*

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
&DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
&S1G13(25,25,15),SIG23(25,25,15),SIG33(25,25,15),
&U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
&CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

DO 100 K=2,NK+3
DO 100 J=2,NJ+3
DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
DXP1=DXXC(I+1)
DXI=DXXC(I)
DXM1=DXXC(I-1)

DYP1=DYYC(J+1)
DYM1=DYYC(J-1)

DZP1=DZZC(K+1)

132
\[ DZK = DZZC(K) \]
\[ DZM1 = DZZC(K-1) \]

C *** SURFACE LENGTH OF THE CONTROL VOLUME
\[ DXN = DXXC(I) \]
\[ DXS = DXXC(I) \]
\[ DXF = DXXC(I) \]
\[ DXB = DXXC(I) \]
\[ DYF = DYYC(J) \]
\[ DYB = DYYC(J) \]
\[ DYE = DYYC(J) \]
\[ DW = DYYC(J) \]
\[ DZE = DZZC(K) \]
\[ DZW = DZZC(K) \]
\[ DZN = DZZC(K) \]
\[ DZS = DZZC(K) \]

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR TEMPERATURE
\[ DXEE = DXXS(I+2) \]
\[ DXE = DXXS(I+1) \]
\[ DXW = DXXS(I) \]
\[ DXWW = DXXS(I-I) \]
\[ DYNN = DYYS(J+2) \]
\[ DYN = DYYS(J+1) \]
\[ DYS = DYYS(J) \]
\[ DYS = DYYS(J-1) \]
\[ DZFF = DZZS(K+2) \]
\[ DZF = DZZS(K+1) \]
\[ DZB = DZZS(K) \]
\[ DZBB = DZZS(K-1) \]

C *** CALCULATE THE AVERAGE VELOCITY IN THE CENTER OF THE CVOLUME
\[ UBAR = (U(I+1,J,K) + U(I,J,K))/2. \]
\[ VBAR = (V(I,J+1,K) + V(I,J,K))/2. \]
\[ WBAR = (W(I,J,K+1) + W(I,J,K))/2. \]

C *** CROSS-SECTIONAL AREA OF THE CV AT IT'S CENTER
\[ DXY = DXI*DYJ \]
\[ DYZ = DJY*DKZ \]
\[ DXZ = DZK*DXI \]

C *** THE NOMRAL STRESSES
\[ SIG11(I,J,K) = 2.*VIS(I,J,K)*((U(I+1,J,K) - U(I,J,K))/DXI + \]
\[ VBAR*(DXN-DXS)/DXY+WBAR*(DXF-DXB)/DZX) \]
\[ SIG22(I,J,K) = 2.*VIS(I,J,K)*((V(I,J+1,K) - V(I,J,K))/DYJ + \]
\[ WBAR*(DYF-DYB)/DYZ+UBAR*(DYE-DYW)/DXY \]
\[ SIG33(I,J,K) = 2.*VIS(I,J,K)*((W(I,J,K+1) - W(I,J,K))/DZK + \]
\[ UBAR*(DZE-DZW)/DZX+VBAR*(DZN-DZS)/DYZ) \]

100 CONTINUE

C *** FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CV FOR SIG12
DO 200 K=2,NK+4
DO 200 J=2,NJ+4
DO 200 I=2,NI+4

C *** CALCULATE THE LENGTH AT VARIOUS POSITIONS
DXN=DXXS(I)
DXS=DXXS(I)
DXI=DXXS(I)
DXE=DXXC(I)
DXW=DXXC(I-1)
DYE=DYYS(J)
DYW=DYYS(J)
DYJ=DYYS(J)
DYN=DYYC(J)
DYS=DYYC(J-1)

C *** THE AVERAGE VELOCITY IN THE CONTROL VOLUME
UBAR=SILIN(U(I,J,K),U(I-1,J,K),DYN,DYS)
VBAR=SILIN(V(I,J,K),V(I-1,J,K),DXE,DXW)

C *** AVERAGE VISCOSITY
VIS12=BILIN(VIS(I,J,K),VIS(I,J-1,K),DYN,DYS,
& VIS(I-1,J,K),VIS(I-1,J-1,K),DYN,DYS,DXE,DXW)

C *** SHEAR STRESS SIG12
SIGA=((V(I,J,K)-V(I-1,J,K))-UBAR*(DYE-DYW)/DYJ)/DXI
SIGB=((U(I,J,K)-U(I,J-1,K))-UBAR*(DXN-DXS)/DXI)/DYJ
SIG12(I,J,K)=VIS12*(SIGA+SIGB)

C *** FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CV FOR SIG13

C *** CALCULATE THE LENGTH AT VARIOUS POSITIONS
DXF=DXXS(I)
DXB=DXXS(I)
DXI=DXXS(I)
DXE=DXXC(I)
DXW=DXXC(I-1)
DZE=DZZS(K)
DZW=DZZS(K)
DZK=DZZS(K)
DFZ=DZZC(K)
DZB=DZZC(K-1)

C *** THE AVERAGE VELOCITY IN THE CONTROL VOLUME
UBAR=SILIN(U(I,J,K),U(I,J,K-1),DFZ,DZB)
WBAR=SILIN(W(I,J,K),W(I-1,J,K),DXE,DXW)

C *** AVERAGE VISCOSITY
VIS13=BILIN(VIS(I,J,K),VIS(I,J,K-1),DFZ,DZB,
& VIS(I-1,J,K),VIS(I-1,J,K-1),DFZ,DZB,DXE,DXW)
C *** SHEAR STRESS SIG13
SIGA=((W(I,J,K)-W(I-1,J,K))/DXI-WBAR*(DZE-DZW)/DZK)/DXI
SIGB=((U(I,J,K)-U(I,J,K-1))/DZK-UBAR*(DXF-DXB)/DXI)/DZK
SIG13(I,J,K)=VIS13*(SIGA+SIGB)

C *** FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CV FOR SIG23

C *** LENGTH AT VARIOUS POSITIONS
DYF=DYYC(J)
DYB=DYYC(J)
DYJ=DYYC(J)
DYN=DYYC(J)
DYS=DYYC(J-1)
DZN=DZZS(K)
DZS=DZZS(K)
DZK=DZZS(K)
DZF=DZZC(K)
DZB=DZZC(K-1)

C *** THE AVERAGE VELOCITY IN THE CONTROL VOLUME
WBAR=SILIN(W(I,J,K),W(I,J-1,K),DYN,DYS)
VBAR=SILIN(V(I,J,K),V(I,J,K-1),DZF,DZB)

C *** AVERAGE VISCOSITY
VIS23=BILIN(VIS(I,J,K),VIS(I,J-1,K),DYN,DYS,
&VIS(I,J,K-1),VIS(I,J-1,K-1),DYN,DYS,DZF,DZB)
SIGA=((V(I,J,K)-V(I,J,K-1))/DYJ-VBAR*(DYF-DYB)/DYJ)/DZK
SIGB=((W(I,J,K)-W(I,J-1,K))/DZK-WBAR*(DZN-DZS)/DZK)/DYJ
SIG23(I,J,K)=VIS23*(SIGA+SIGB)

200 CONTINUE
RETURN
END

************************************************************************************
************************************************************************************
************************************************************************************
SUBROUTINE TCP
************************************************************************************
*THIS SUBROUTINE CALCULATES THE NONDIMENSIONAL TEMPERATURE AT THE
* THERMOCOUPLE POSITIONS.
************************************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZG(40),XS(40),YS(40),ZS(40),DXXC(40),
& DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
COMMON/B1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/B3/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
& U(25,25,15),V(25,25,15),W(25,25,15)
COMMON/B38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO

C *** CALCULATE SIZE OF CONTROL VOLUME CONTAINING THE THERMOCOUPLES
DO 5100 N=1,NTHCO
VOL = \text{ABS}((XC(NTH(N,1)+1)-XC(NTH(N,1)))*(YC(NTH(N,2)+1)-YC(NTH(N,2)))*(ZC(NTH(N,3)+1)-ZC(NTH(N,3))))

TCOUP(N) = 0.

DO 5101 I = NTH(N,1), NTH(N,1)+1
II = 2*NTH(N,1)+1-I
DO 5102 J = NTH(N,2), NTH(N,2)+1
JJ = 2*NTH(N,2)+1-J
DO 5103 K = NTH(N,3), NTH(N,3)+1
KK = 2*NTH(N,3)+1-K

C *** CORRECT TEMPERATURES FOR THERMOCOUPLES NOT LOCATED ON NODES
TVOL = \text{ABS}((XC(I)-CX(N))*(YC(J)-CY(N))*(ZC(K)-CZ(N)))
WVOL = TVOL/VOL
TCOUP(N) = TCOUP(N) + WVOL*T(IJ,JK)
5103 CONTINUE
5102 CONTINUE
5101 CONTINUE
IF (TCOUP(N) .LT. TCOOL) TCOUP(N) = TCOOL
5100 CONTINUE
RETURN
END

**********************************************************************************************
**********************************************************************************************
SUBROUTINE TRID(IST,JST,KST,ISP,JSP,KSP,PHI)
**********************************************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
& AS(25,25,15),AF(25,25,15),AB(25,25,15),
& SP(25,25,15),RI(25,25,15)
DIMENSION A(99),B(99),C(99),PHI(25,25,15)

C *** FORWARD SWEEP IN THE X DIRECTION (FROM IST TO ISP)
A(IST-1) = 0.
C(IST-1) = 0.
DO 100 J = JST, JSP
DO 100 K = KST, KSP
DO 101 I = IST, ISP
A(I) = AE(I,J,K)
B(I) = AW(I,J,K)
C(I) = AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+
& AF(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
TERM = 1./(AP(I,J,K)-B(I)*A(I-1))
IF (ABS(A(I)).LE.1.0E-38) A(I) = 0.0
IF (ABS(B(I)).LE.1.0E-38) B(I) = 0.0
IF (ABS(C(I)).LE.1.0E-38) C(I) = 0.0
IF (ABS(TERM)).LE.1.0E-38 TERM = 0.0
A(I) = A(I)*TERM
C(I) = (C(I)+B(I)*C(I-1))*TERM
101 CONTINUE
PHI(ISP,J,K) = C(ISP)
DO 102 I = ISP-1, IST, -1
\[ \text{PHI}(I,J,K) = A(I) \times \text{PHI}(I+1,J,K) + C(I) \]

102 CONTINUE
100 CONTINUE

C *** FORWARD SWEEP IN THE Y DIRECTION (FROM JST TO JSP)
A(JST-1)=0.
C(JST-1)=0.
DO 200 K=KST,KSP
DO 200 I=IST,ISP
   DO 201 J=JST,JSP
      A(J) = AN(I,J,K)
      B(J) = AS(I,J,K)
      C(J) = AE(I,J,K) \times \text{PHI}(I+1,J,K) + AW(I,J,K) \times \text{PHI}(I-1,J,K) + 
      AN(I,J,K) \times \text{PHI}(I+1,J,K) + AU(I,J,K) \times \text{PHI}(I,J,K) + SU(I,J,K) 
      & \text{TERM} = 1.0 / (A(I,J,K) + B(J) \times A(I,J,K)) 
      IF (ABS(A(J)) \leq 1.0E-38) A(J)=0.0
      IF (ABS(B(J)) \leq 1.0E-38) B(J)=0.0
      IF (ABS(C(J)) \leq 1.0E-38) C(J)=0.0
      IF (ABS(\text{TERM}) \leq 1.0E-38) \text{TERM}=0.0
      A(J) = A(J) \times \text{TERM}
      C(J) = (C(J)+B(J) \times C(J-1)) \times \text{TERM}
   201 CONTINUE
   PHI(I,J,JSP,K)=C(JSP)
   DO 202 J=JSP-1,JST,-1
      \text{PHI}(I,J,K) = A(J) \times \text{PHI}(I,J,K+1) + C(J)
   202 CONTINUE
200 CONTINUE

C *** FORWARD SWEEP IN THE Z DIRECTION (FROM KST TO KSP)
A(KST-1)=0.
C(KST-1)=0.
DO 300 I=IST,ISP
DO 300 J=JST,JSP
   DO 301 K=KST,KSP
      A(K) = AF(I,J,K)
      B(K) = AB(I,J,K)
      C(K) = AE(I,J,K) \times \text{PHI}(I+1,J,K) + AW(I,J,K) \times \text{PHI}(I-1,J,K) + 
      AN(I,J,K) \times \text{PHI}(I+1,J,K) + AS(I,J,K) \times \text{PHI}(I,J,K) + SU(I,J,K) 
      & \text{TERM} = 1.0 / (A(I,J,K) + B(K) \times A(K-1)) 
      IF (ABS(A(K)) \leq 1.0E-38) A(K)=0.0
      IF (ABS(B(K)) \leq 1.0E-38) B(K)=0.0
      IF (ABS(C(K)) \leq 1.0E-38) C(K)=0.0
      IF (ABS(\text{TERM}) \leq 1.0E-38) \text{TERM}=0.0
      A(K) = A(K) \times \text{TERM}
      C(K) = (C(K)+B(K) \times C(K-1)) \times \text{TERM}
   301 CONTINUE
   PHI(I,J,KSP,K)=C(KSP)
   DO 302 K=KSP-1,KST,-1
      \text{PHI}(I,J,K) = A(K) \times \text{PHI}(I,J,K+1) + C(K)
   302 CONTINUE
300 CONTINUE

C *** REVERSE SWEEP IN X DIRECTION (FROM ISP TO IST)
B(KSP+1)=0.
C(KSP+1)=0.
DO 600 I=ISP,IST,-1

137
DO 600 J=JSP, JST, -1
DO 601 K=KSP, KST, -1
A(K)=AF(I, J, K)
B(K)=AB(I, J, K)
C(K)=AE(I, J, K)*PHI(I+1, J, K)+AW(I, J, K)*PHI(I-1, J, K)+
&
AN(I, J, K)*PHI(I, J+1, K)+AS(I, J, K)*PHI(I, J-1, K)+SU(I, J, K)
TERM=1./((AP(I, J, K)-A(K)*B(K+1))
B(K)=B(K)*TERM
C(K)=(C(K)+A(K)*C(K+1))*TERM
IF (ABS(A(K)).LE.1.0E-38) A(K)=0.0
IF (ABS(C(K)).LE.1.0E-38) C(K)=0.0
601 CONTINUE
PHI(I, J, KST)=C(KST)
DO 602 K=KST+1, KSP
PHI(I, J, K)=B(K)*PHI(I, J, K-1)+C(K)
602 CONTINUE
600 CONTINUE
C *** REVERSE SWEEP IN THE Y DIRECTION (FROM JSP TO JST)
B(JSP+1)=0.
C(JSP+1)=0.
DO 500 K=KSP, KST, -1
DO 500 I=ISP, IST, -1
DO 301 J=JSP, JST, -1
A(J)=AN(I, J, K)
B(J)=AS(I, J, K)
C(J)=AE(I, J, K)*PHI(I+1, J, K)+AW(I, J, K)*PHI(I-1, J, K)+
&
AF(I, J, K)*PHI(I, J+1, K)+AB(I, J, K)*PHI(I, J, K-1)+SU(I, J, K)
TERM=1./((AP(I, J, K)-A(J)*B(J+1))
B(J)=B(J)*TERM
C(J)=(C(J)+A(J)*C(J+1))*TERM
IF (ABS(A(J)).LE.1.0E-38) A(J)=0.0
IF (ABS(B(J)).LE.1.0E-38) B(J)=0.0
IF (ABS(C(J)).LE.1.0E-38) C(J)=0.0
501 CONTINUE
PHI(I, JST, K)=C(JST)
DO 502 J=JST+1, JSP
PHI(I, J, K)=B(J)*PHI(I, J-1, K)+C(J)
502 CONTINUE
500 CONTINUE
C *** REVERSE SWEEP IN THE Z DIRECTION (FROM KSP TO KST)
B(ISP+1)=0.
C(ISP+1)=0.
DO 400 J=JSP, JST, -1
DO 400 K=KSP, KST, -1
DO 301 I=ISP, IST, -1
A(I)=AE(I, J, K)
B(I)=AW(I, J, K)
C(I)=AN(I, J, K)*PHI(I, J+1, K)+AS(I, J, K)*PHI(I, J-1, K)+
&
AF(I, J, K)*PHI(I, J+1, K)+AB(I, J, K)*PHI(I, J, K-1)+SU(I, J, K)
TERM=1./((AP(I, J, K)-A(I)*B(I+1))
B(I)=B(I)*TERM
C(I)=(C(I)+A(I)*C(I+1))*TERM
IF (ABS(A(I)).LE.1.0E-38) A(I)=0.0
138
IF (ABS(B(I)).LE.1.0E-38) B(I)=0.0
IF (ABS(C(I)).LE.1.0E-38) C(I)=0.0

401 CONTINUE

PHI(IST,J,K)=C(IST)
DO 402 I=IST+1,ISP
   PHI(I,J,K)=B(I)*PHI(I-1,J,K)+C(I)
402 CONTINUE
400 CONTINUE

RETURN
END

***********************************************************************
***SUBROUTINE VIEW
***********************************************************************

*NTHS = TOTAL NUMBER OF NODES CONTAINING HEAT SOURCE
*NFX,NFY,NFZ = NUMBER OF NODES IN HEAT SOURCE PER DIRECTION
*FX,FY,FZ = STARTING COORDINATES OF HEAT SOURCE
*DXS,DYS,DZS = LENGTH IN EACH DIRECTION ON THE SOUTH SURFACE
*SXS,SY,SZ = COORDINATES OF THE SURFACE ELEMENT (USED TO
*CALCULATE THE DISTANCE TO HEAT SOURCE)
*VFHSE,VFHSE = VIEW FACTOR FROM THE HEAT SOURCE TO WEST, EAST
*SURFACES OF THE ENCLOSURE
*RSQW,RSQE = SQUARE OF DISTANCE FROM THE HEAT SOURCE TO THE
*WEST, EAST SURFACE ELEMENT.
*
*NODES 2 THRU NI,2 THRU NJ AND 2 THRU NK ARE CONTAINED
*INSIDE THE WALL, RADIATION EFFECTS INVOLVE ONLY NODES
*3 THRU NI 1, 3 THRU NJM1, AND 3 THRU NKM1
*
*VFHSBW(N,M,I,J),
*VFHSBE(N,M,I,J) = VIEW FACTOR FROM FIRE NODE N TO ELEMENT (I,J) OF
*WEST, EAST SURFACES OF CV M.
***********************************************************************

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40) ,YC(40) ,ZC(40) ,XS(40) ,YS(40) ,ZS(40) ,DXXC(40),
&DYYC(40) ,DZZC(40) ,DXXS(40) ,DYS(40) ,DZZS(40)
COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
COMMON/BL2/X,Y,H,TFLR,TWAL
COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
COMMON/BL22/CPS(20) ,CONS(20) ,WFAN(20) ,NCHIP,ICHPB(20) ,NCHPI(20),
&JCHPB(20) ,NCHPJJ(20) ,KCHPB(20) ,NCHPK(20)
COMMON/BL37/VIS(25,25,15) ,COND(25,25,15) ,RESORM(40),
&CPM(25,25,15) ,NHSZ(3,2) ,NOD(25,25,15)
COMMON/BL40/VFHSW(5,25,15) ,VFHSE(5,25,15) ,VFHSS(5,15,25),
COMMON/BL41/VFHSBW(5,8,34,34) ,VFHSBE(5,8,34,34) ,VFHSBS(5,8,34,34),
&VFHSBN(5,8,34,34) ,VFHSBB(5,8,34,34) ,VFHSBF(5,8,34,34)

NTHS=NHSZ(3,2)-NHSZ(3,1)+1
DO 500 N=1,NTHS
   SUMM=0.
   NFX=NHSZ(1,1)
500 CONTINUE
NFY = NHSZ(2, 1)
NFZ = NHSZ(3, 1) - 1 + N
FX = XS(NFX + 1)
FY = YS(NFY + 1)
FZ = ZS(NFZ + 1)

C *** VIEW FACTOR FROM FIRE TO WEST & EAST SURFACES OF ENCLOSURE
DO 100 J = 3, NJ + 2
DO 100 K = 3, NK + 2
     DYW = DYYC(J)
     DYE = DYYC(J)
     DZW = DZZC(K)
     DZE = DZZC(K)
     SXW = XS(3)
     SXE = XS(NI + 3)
     SYW = YC(J)
     SYE = YC(J)
     SZW = ZC(K)
     SZE = ZC(K)
     DYZW = DYW * DZW
     DYZE = DYE * DZE
     RSQW = (FX - SXW)**2 + (FY - SYW)**2 + (FZ - SZW)**2
     RSQE = (FX - SXE)**2 + (FY - SYE)**2 + (FZ - SZE)**2
     VFHSW(N, J, K) = SQRT((FX - SXW)**2/RSQW)*DYZW/(4.0*PI*RSQW)
     VFHSE(N, J, K) = SQRT((FX - SXE)**2/RSQE)*DYZE/(4.0*PI*RSQE)
     SUMM = SUMM + VFHSW(N, J, K) + VFHSE(N, J, K)
100 CONTINUE

C *** VIEW FACTOR FROM FIRE TO NORTH & SOUTH SURFACES OF ENCLOSURE
DO 200 I = 3, NI + 2
DO 200 K = 3, NK + 2
     DXS = DXXC(I)
     DXN = DXXC(I)
     DZS = DZZC(K)
     DZN = DZZC(K)
     SXN = XC(I)
     SXS = XC(I)
     SYN = YS(NJ + 3)
     SYS = YS(3)
     SZN = ZC(K)
     SZS = ZC(K)
     DZXS = DXS * DZS
     DZXN = DXN * DZN
     RSQS = (FX - SXS)**2 + (FY - SYS)**2 + (FZ - SZS)**2
     RSQN = (FX - SXN)**2 + (FY - SYN)**2 + (FZ - SZN)**2
     VFHSS(N, K, I) = SQRT((FY - SYS)**2/RSQS)*DZXS/(4.0*PI*RSQS)
     VFHSN(N, K, I) = SQRT((FY - SYN)**2/RSQN)*DZXN/(4.0*PI*RSQN)
     SUMM = SUMM + VFHSS(N, K, I) + VFHSN(N, K, I)

140
200 CONTINUE
C *** VIEW FACTOR FROM FIRE TO FRONT & BACK SURFACES OF ENCLOSURE
DO 300 I=3,NI+2
DO 300 J=3,NJ+2
  DXF=DXXC(I)
  DXB=DXXC(I)
  DYF=DYYC(J)
  DYB=DYYC(J)
  SXF=XC(I)
  SXB=XC(I)
  SYF=YC(J)
  SYB=YC(J)
  SZF=ZS(NK+3)
  SZB=ZS(3)
  DXYB=DXB*DYB
  DXYF=DXF*DYF
  RSQB=(FX-SXB)**2+(FY-SYB)**2+(FZ-SZB)**2
  RSQF=(FX-SXF)**2+(FY-SYF)**2+(FZ-SZF)**2
VHSB(N,I,J)=SQRT((FZ-SZB)**2/RSQB)*DXYB/(4.0*PI*RSQB)
VHSF(N,I,J)=SQRT((FZ-SZF)**2/RSQF)*DXYF/(4.0*PI*RSQF)
  SUMM=SUMM+VHSB(N,I,J)+VHSF(N,I,J)
300 CONTINUE
C *** MODIFY VIEW FACTORS SO THEIR SUMMATION EQUALS UNITY.
DO 150 J=3,NJ+2
DO 150 K=3,NK+2
  VFHSW(N,J,K)=VHSW(N,J,K)/SUMM
  VFHSE(N,J,K)=VFHSE(N,J,K)/SUMM
150 CONTINUE
DO 250 K=3,NK+2
DO 250 I=3,NI+2
  VFHSS(N,K,I)=VHS(N,K,I)/SUMM
  VFHSN(N,K,I)=VFHSN(N,K,I)/SUMM
250 CONTINUE
DO 350 I=3,NI+2
DO 350 J=3,NJ+2
  VHSB(N,I,J)=VHSB(N,I,J)/SUMM
  VHSF(N,I,J)=VHSF(N,I,J)/SUMM
350 CONTINUE
IF (NCHIP.LT.NBLOR) GOTO 500
C *** CALCULATE VIEW FACTORS FROM FIRE TO INTERNAL SOLID BLOCKS
C *** (BLOCKS ARE ONLY THOSE NOT INCLUDED IN THE WALL)
C *** VIEW FACTOR FROM THE FIRE TO WEST & EAST SURFACES OF BLOCK M
DO 900 M=1,NCHIP-NBLOR+1
  IB =ICHPB(M+NBLOR-1)
  IE =IB+NCHPI(M+NBLOR-1)-1
  JB =JCHPB(M+NBLOR-1)
  JE =JB+NCHPJ(M+NBLOR-1)-1
900 CONTINUE
KB = KCHPB(M + NBLOR - 1)
KE = KB + NCHPK(M + NBLOR - 1) - 1

SXW = XS(IB)
SXE = XS(IE)

DO 400 JJ = JB, JE - 1
DO 400 KK = KB, KE - 1
DYW = DYYC(JJ)
DYE = DYYC(JJ)
DZW = DZZC(KK)
DZE = DZZC(KK)
SYW = YC(JJ)
SYE = YC(JJ)
SZW = ZC(KK)
SZE = ZC(KK)

DYZW = DYW * DZW
DYZE = DYE * DZE
RSQW = (FX - SXW)**2 + (FY - SYW)**2 + (FZ - SZW)**2
RSQE = (FX - SXE)**2 + (FY - SYE)**2 + (FZ - SZE)**2
VFHSBW(N, M, JJ, KK) = SQRT((FX - SXW)**2 / RSQW) * DYZW / (4.0 * PI * RSQW)
VFHSBE(N, M, JJ, KK) = SQRT((FX - SXE)**2 / RSQW) * DYZE / (4.0 * PI * RSQE)

C *** MODIFY VIEW FACTORS DUE TO INTRODUCTION OF INTERNAL BLOCKS
IF (SXE.LT.FX) VFHSBW(N, M, JJ, KK) = 0.
IF (SXW.GT.FX) VFHSBE(N, M, JJ, KK) = 0.
C *** THE FIRE CAN'T SEE THE WEST AND EAST SURFACES OF THE BLOCK
IF (SXW.LE.FX.AND.SXE.GE.FX) THEN
VFHSBW(N, M, JJ, KK) = 0.
VFHSBE(N, M, JJ, KK) = 0.
ENDIF

400 CONTINUE

C *** CHECK TO SEE IF ANY ELEMENT ON THE WALL IS SHADDED BY A SOLID BLOCK.
C *** CHECK WEST AND EAST WALLS OF ENCLOSURE
DO 410 J = 3, NJ + 2
DO 410 K = 3, NK + 2

C *** THE BLOCK IS ON THE WEST SIDE OF THE FIRE
IF (SXE.LT.FX) THEN
NVIW = NVIWX(FX, FY, FZ, XS(3), YC(J), ZC(K), SXE, IB, JB, KB, IE, JE, KE)
IF (NVIW.EQ.1) VFHSW(N, J, K) = 0.
ENDIF

C *** THE BLOCK IS ON THE EAST SIDE OF THE FIRE
IF (SXW.GT.FX) THEN
NVIW = NVIWX(FX, FY, FZ, XS(NI + 3), YC(J), ZC(K), SXW, IB, JB, KB, IE, JE, KE)
IF (NVIW.EQ.1) VFHSE(N, J, K) = 0.
ENDIF

142
CONTINUE

C *** CHECK SOUTH AND NORTH WALLS OF THE ENCLOSURE
DO 420 K=3,NK+2
DO 420 I=3,NI+2

C *** THE BLOCK IS ON THE WEST SIDE OF THE FIRE
IF (SXE.LT.FX.AND.XC(I).LT.FX) THEN
   NVIW=NVIWX(FX,FY,FZ,XC(I),YS(K),ZC(K),SXE,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
   NVIW=NVIWX(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SXE,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
ENDIF

C *** THE BLOCK IS ON THE EAST SIDE OF THE FIRE
IF (SXW.GT.FX.AND.XC(I).GT.FX) THEN
   NVIW=NVIWX(FX,FY,FZ,XC(I),YS(K),ZS(K),SXW,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
   NVIW=NVIWX(FX,FY,FZ,XC(I),YS(NK+3),ZS(K),SXW,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
ENDIF

CONTINUE

C *** CHECK BACK AND FRONT WALLS OF ENCLOSURE
DO 430 I=3,NI+2
DO 430 J=3,NJ+2

C *** THE BLOCK IS ON THE WEST SIDE OF THE FIRE
IF (SXE.LT.FX.AND.XC(I).LT.FX) THEN
   NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(K),SXE,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
   NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SXE,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSF(NI,J)=0.
ENDIF

C *** THE BLOCK IS ON THE EAST SIDE OF THE FIRE
IF (SXW.GT.FX.AND.XC(I).GT.FX) THEN
   NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(3),SXW,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
   NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SXW,
   & IB,JB,KB,IE,JE,KE)
   IF (NVIW.EQ.1) VFHSF(NI,J)=0.
ENDIF

CONTINUE

C *** VIEW FACTOR FROM FIRE TO NORTH & SOUTH SURFACES OF BLOCK M
DO 600 II=IB,IE-1
DO 600 KK=KB,KE-1
   DXN=DXXC(II)
DXS=DXXC(II)
DZN=DZZC(KK)
DZS=DZZC(KK)

SXN=XC(II)
SXZ=XC(II)
SYN=YS(JE)
SYS=YS(JB)
SZN=ZC(KK)
SZS=ZC(KK)

DZXN=DXN*DZN
DZXN=DXS*DZS

RSQS=((FX-SXS)**2+(FY-SYS)**2+(FZ-SZS)**2
RSQN=((FX-SXN)**2+(FY-SYN)**2+(FZ-SZN)**2
VFHSBS(N,M,KK,II)=SQRT((FY-SYS)**2/RSQS)*DZXS/C4.O*PI*RSQS
VFHSBN(N,M,KK,II)=SQRT((FY-SYN)**2/RSQN)*DZXN/(4.O*PI*RSQN)

C *** MODIFY VIEW FACTORS DUE TO INTRODUCTION OF INTERNAL BLOCKS
IF (SYN.LT.FY) VFHSBS(N,M,KK,II)=0.
IF (SYS.GT.FY) VFHSBN(N,M,KK,II)=0.
IF (SYS.LE.FY .AND. SYN.GE.FY) THEN
  VFHSBS(N,M,KK,II)=0.
  VFHSBN(N,M,KK,II)=0.
ENDIF

C *** CHECK IF ANY ELEMENT ON WALL IS SHADIED BY INTERNAL SOLID BLOCKS
C *** CHECK WEST AND EAST WALLS OF THE ENCLOSURE
DO 610 J=3,NJ+2
DO 610 K=3,NK+2

C *** THE BLOCK IS ON THE SOUTH SIDE OF THE FIRE
IF (SYN.LT.FY .AND. YC(J).LT.FY) THEN
  NVIW=NVIWY(FX,FY,FZ,XS(3),YC(J),ZC(K),SYN,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
  NVIW=NVIWY(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SYN,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
ENDIF

C *** THE BLOCK IS ON THE NORTH SIDE OF THE FIRE
IF (SYS.GT.FY .AND. YC(J).GT.FY) THEN
  NVIW=NVIWY(FX,FY,FZ,XS(3),YC(J),ZC(K),SYS,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
  NVIW=NVIWY(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SYS,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
ENDIF

610 CONTINUE

C *** CHECK THE SOUTH AND NORTH WALLS OF THE ENCLOSURE
DO 620 K=3,NK+2
DO 620 I=3,NI+2
C *** THE BLOCK IS ON THE SOUTH SIDE OF THE FIRE
IF (SYN.LT.FY) THEN
  NVIW=NVIWY(FX,FY,FZ,XC(I),YS(3),ZC(K),SYN,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
ENDIF
C *** THE BLOCK IS ON THE NORTH SIDE OF THE FIRE
IF (SYS.GT.FY) THEN
  NVIW=NVIWY(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SYS,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
ENDIF
620 CONTINUE
C *** THE BACK AND FRONT WALLS OF THE ENCLOSURE
DO 630 I=3,NI+2
DO 630 J=3,NJ+2
C *** THE BLOCK IS ON THE SOUTH SIDE OF THE FIRE
IF (SYN.LT.FY.AND.YC(J).LT.FY) THEN
  NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(3),SYN,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.-1) VFHSB(N,I,J)=0.
  NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SYN,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
ENDIF
C *** THE BLOCK IS ON THE NORTH SIDE OF THE FIRE
IF (SYS.GT.FY.AND.YC(J).GT.FY) THEN
  NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(3),SYS,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.-1) VFHSB(N,I,J)=0.
  NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SYS,
  & IB,JB,KB,IE,JE,KE)
  IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
ENDIF
630 CONTINUE
C *** CHECK VIEW FACTORS FROM FIRE TO BACK & FRONT SURFACES OF BLOCK M
DO 700 II=IB,IE-1
DO 700 JJ=JB,JE-1
  DXF=DXXC(II)
  DXB=DXXC(II)
  DYF=DYYC(JJ)
  DYB=DYYC(JJ)
  DXYB=DXB*DYB
  DXYF=DXF*DYF
  SXF=XC(II)
  SXB=XC(II)
MODIFY VIEW FACTORS DUE TO INTRODUCTION OF INTERNAL SOLID BLOCKS

IF (SZF.LT.FZ) VFHSBB(N,M,II,JJ)=0.
IF (SZB.GT.FZ) VFHSBF(N,M,II,JJ)=0.
IF (SZB.LE.FZ.AND.SZF.GE.FZ) THEN
VFHSBB(N,M,II,JJ)=0.
VFHSBF(N,M,II,JJ)=0.
ENDIF

CHECK IF ANY ELEMENT ON THE WALL IS SHADED BY SOLID BLOCK.

THE WEST AND EAST WALLS OF THE ENCLOSURE
DO 710 J=3,NJ+2
DO 710 K=3,NK+2

THE BLOCK IS ON THE BACK SIDE OF THE FIRE
IF (SZF.LT.FZ.AND.ZC(K).LT.FZ) THEN
NVIW=NVIWZ(FX,FY,FZ,XS(3),YC(J),ZC(K),SZF, IB,JB,KB,IE,JE,KE)
IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
NVIW=NVIWZ(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SZF,
IB,JB,KB,IE,JE,KE)
IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
ENDIF

THE BLOCK IS ON THE FRONT SIDE OF THE FIRE
IF (SZB.GT.FZ.AND.ZC(K).LT.FZ) THEN
NVIW=NVIWZ(FX,FY,FZ,XS(3),YC(J),ZC(K),SZB,
IB,JB,KB,IE,JE,KE)
IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
NVIW=NVIWZ(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SZB,
IB,JB,KB,IE,JE,KE)
IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
ENDIF

CHECK THE SOUTH AND NORTH WALLS OF THE ENCLOSURE
DO 720 K=3,NK+2
DO 720 I=3,NI+2

THE BLOCK IS ON THE BACK SIDE OF THE FIRE
IF (SZF.LT.FZ.AND.ZC(K).LT.FZ) THEN
NVIW=NVIWZ(FX,FY,FZ,XC(I),YS(3),ZC(K),SZF,
IB,JB,KB,IE,JE,KE)
IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
NVIW=NVIWZ(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SZF,
& IB,JB,KB,IE,JE,KE
  IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
ENDIF

C *** THE BLOCK IS ON THE FRONT SIDE OF THE FIRE
   IF (SZB.GT.FZ.AND.ZC(K).GT.FZ)
     NVIW=NVIWZ(FX,FY,FZ,XC(I),YS(3),ZC(K),SZB,
& IB,JB,KB,IE,JE,KE)
     IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
   ENDIF

720 CONTINUE

C *** CHECK THE BACK AND FRONT WALLS OF THE ENCLOSURE
   DO 730 I=3,NI+2
   DO 730 J=3,NJ+2

C *** THE BLOCK IS ON THE BACK SIDE OF THE FIRE
   IF (SZF.LT.FZ)
     NVIW=NVIWZ(FX,FY,FZ,XC(I),YC(J),YS(3),SZF,
& IB,JB,KB,IE,JE,KE)
     IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
   ENDIF

C *** THE BLOCK IS ON THE FRONT SIDE OF THE FIRE
   IF (SZB.GT.FZ)
     NVIW=NVIWZ(FX,FY,FZ,XC(I),YC(J),YS(NJ+3),ZC(K),SZB,
& IB,JB,KB,IE,JE,KE)
     IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
   ENDIF

730 CONTINUE

900 CONTINUE

500 CONTINUE

RETURN
END

******************************************************************************
******************************************************************************
FUNCTION BILIN(V1,V2,D1,D2,V3,V4,D3,D4,D5,D6)
******************************************************************************
******************************************************************************
* BI-LINEAR INTERPOLATION

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
V12=(V1*D2+V2*D1)/(D1+D2)
V34=(V3*D4+V4*D3)/(D3+D4)
BILIN=(V12*D6+V34*D5)/(D5+D6)

RETURN
END
INTEGER FUNCTION NVIX(FX,FY,FZ,X1,Y1,Z1,X3,IB,JB,KB,IE,JE,KE)
*USED ONLY IN SUBROUTINE VIEW
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
&
  DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40)

NVIWX=0
TPARA=(X3-X1)/(FX-X1)
Y3=(FY-Y1)*TPARA+Y1
Z3=(FZ-Z1)*TPARA+Z1
IF (Y3.LE.YS(JE).AND.Y3.GE.YS(JB)) THEN
  IF (Z3.LE.ZS(KE).AND.Z3.GE.ZS(KB)) NVIWX=1
ENDIF
RETURN
END

INTEGER FUNCTION NVIWY(FX,FY,FZ,X1,Y1,Z1,Y3,IB,JB,KB,IE,JE,KE)
*USED ONLY IN SUBROUTINE VIEW
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
&
  DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40)

NVIWY=0
TPARA=(Y3-Y1)/(FY-Y1)
X3=(FX-X1)*TPARA+X1
Z3=(FZ-Z1)*TPARA+Z1
IF (X3.LE.XS(IE).AND.X3.GE.XS(IB)) THEN
  IF (Z3.LE.ZS(KE).AND.Z3.GE.ZS(KB)) NVIWY=1
ENDIF
RETURN
END

INTEGER FUNCTION NVIWZ(FX,FY,FZ,X1,Y1,Z1,Z3,IB,JB,KB,IE,JE,KE)
*USED ONLY IN SUBROUTINE VIEW
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
&
  DYYC(40),DZZC(40),DXXS(40),DYS(40),DZZS(40)

NVIWZ=0
TPARA=(Z3-Z1)/(FZ-Z1)
Y3=(FY-Y1)*TPARA+Y1
X3=(FX-X1)*TPARA+X1
IF (Y3.LE.YS(JE).AND.Y3.GE.YS(JB)) THEN
  IF (X3.LE.XS(IE).AND.X3.GE.XS(IB)) NVIWZ=1
**SINGLE LINEAR INTERPOLATION**

FUNCTION SILIN(V1,V2,D1,D2)

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

SILIN=(V1*D2+V2*D1)/(D1+D2)

RETURN

END
APPENDIX C. PROGRAM ISOTHERM

This program generates two dimensional isotherm plots using the National Center for Atmospheric Research (NCAR) Graphics software, from three dimensional temperature field data produced by program FIRE.

PROGRAM ISOTHERM

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

This program uses NCAR GRAPHICS to calculate and plot isotherms in an arbitrary two dimensional section with uniform grid spacing from a non-uniform three dimensional grid using linear interpolation routines. Isotherms are plotted using 14 equal size color ranges based on the maximum temperature in the 3-D grid.

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

Variables used in this program. Arrays names are followed by left and right parenthesis (). (NOTE: some variables used in the plotting routines have been specifically omitted from this list)

* bth : Non-dimensional compartment breadth (X-direction)
* cx : Interpolation parameter used to generate 2-D grid
* cy : Interpolation parameter used to generate 2-D grid
* cl : Interpolation parameter used to locate desired 3-D section
* c2 : Interpolation parameter used to locate desired 3-D section
* c11 : Interpolation parameter used to generate 2-D grid
* c12 : Interpolation parameter used to generate 2-D grid
* c21 : Interpolation parameter used to generate 2-D grid
* c22 : Interpolation parameter used to generate 2-D grid
* dx : Non-dimensional 3-D grid size inside compartment, X dir.
  (redefined in SUBROUTINE SCTN as non-dimensional uniform grid spacing in the X direction for the 2-D plot)
* dy : Non-dimensional 3-D grid size inside compartment, Y dir.
  (redefined in SUBROUTINE SCTN as non-dimensional uniform grid spacing in the Y direction for the 2-D plot)
* dz : Non-dimensional 3-D grid size inside compartment, Z dir.
* h : Non-dimensional Z length of compartment to model
* iplot : Number of grids in X direction of 2-D grid
* iscn : If >0 hold X constant, else =0
* ixx : Number of X grid points in "xxs" array
* jplot : Number of grids in Y direction of 2-D grid
* jscn : If >0 hold Y constant, else =0
* jyy : Number of Y grid points in "yys" array
* kscn : If >0 hold Z constant, else =0
* ncrmax : Maximum number of contours
* ni : Number of cells across compartment, X dir.
* nj : Number of cells across compartment, Y dir.
* nk : Number of cells across compartment, Z dir.
* nstop : Control variable to terminate program

150
* nview : Input value of desired plot (from SUBROUTINE SECTION)
* t() : Imported array of temperature values
* ta : Ambient absolute temperature prior to fire ignition
* temp() : Temperatures of desired section of 3-D grid
* tflr : Non-dimensional thickness of compartment floor/ceiling
* time : Dimensional time (in seconds) since fire ignition
* tmax : Maximum temperature in "temp" array (redefined in SUBROUTINE SECTION as maximum temperature in interpolated 2-D array "tt")
* tt() : Section temperatures interpolated to 2-D grid
* twal : Non-dimensional thickness of compartment walls
* wth : Non-dimensional compartment width (Y-direction)
* x : Non-dimensional X length of compartment to model
* xfrnl : Starting 2-D, X coordinate of section to zoom in on
* xfrn2 : Ending 2-D, X coordinate of section to zoom in on
* xs() : Non-dimensional grid locations
* xscn : X location where Y-Z plane to be taken
* xss() : Non-dimensional staggered grid locations
* xtime : Non-dimensional time since fire ignition
* xx : Non-dimensional length of 2-D X axis
* xxs() : 3-D grid points to be used as X coordinate in 2-D plot
* y : Non-dimensional Y length of compartment to model
* yfrnl : Starting 2-D, Y coordinate of section to zoom in on
* yfrn2 : Ending 2-D, Y coordinate of section to zoom in on
* yscn : Y location where X-Z plane to be taken
* yy : Non-dimensional length of 2-D Y axis
* yys() : 3-D grid points to be used as Y coordinate in 2-D plot
* zscn : Z location where X-Y plane to be taken

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

C *** INPUT DATA FROM PROGRAM "FIRE"
call INPUT

C *** GENERATE GRID
call GRID

C *** DEFINE DESIRED SECTION
  5 call SECTION(nstop)

C *** CHECK IF FINISHED WITH PROGRAM
if (nstop.ne.0) goto 9999

C *** LOCATE DESIRED SECTION TO PLOT
call INTERPUL

C *** INTERPOLATE TO A 2-D PLOT
call SCTN

C *** DEVELOP PLOT USING NCAR GRAPHICS
call PLOT

C *** LOOP BACK FOR ANOTHER SECTION
  goto 5

C *** FINISHED WITH PROGRAM
9999 end

*******************************************************************************
The variables IPLOT and JPLOT must be changed in order to vary the number of grids in the uniform 2-D plot. NCRMAX must be changed to vary the maximum number of colors allowed by the NCAR plots (NOTE: NCRMAX must include one value for the default foreground color).

```
implicit real*8 (a-h,o-z)
common/ugrd/iplot,jplot,ncrmax
common/sctl/iscn,jscn,kscn,nview

data iplot,jplot,ncrmax/24,24,15/
data nview/0/
end
```

```
SUBROUTINE SECTION(nstop)

implicit real*8 (a-h,o-z)
common/fctn/xfrn1,xfrn2,yfrn1,yfrn2
common/sctl/iscn,jscn,kscn,nview
common/sct2/xscn,yscn,zscn
common/bl2/x,y,h,tflr,twal,ta

99 print *
print *
prient *, 'Enter For:
print *, '0 Finished with program'
print *, '1 Plan View (X-Y Plane)'
print *, '2 X-Z Profile'
print *, '3 Y-Z Profile'
if(nview.le.0) goto 97
print *, '4 Enlargement of portion of previous plot'
97 print *
prient *, 'Enter the desired isotherm plot:'
read (*,*) nview
nstop=0
xfrn1=0.
xfrn2=1.
yfrn1=0.
yfrn2=1.
c *** END PROGRAM
if(nview.eq.0) then
  nstop=9999
C *** PLOT AN X-Y PLANE (a.k.a. Z section)
elseif(nview.eq.1) then
  iscn=0
  jscn=0
  kscn=1
```
xscn=0.
yscn=0.

10 print *
7 format (1x,a/1x,a,f5.2,a)
print 7,
& 'Enter height of desired view plane in feet above the floor',
& '(Floor = 0.0; Ceiling = ',h,'):'
read (*,*) zscn
if(zscn.lt.0.0.or.zscn.gt.h) then
  print *
  print 5,'Input value must be between 0.0 and ',h
  format (1x,a,f4.1)
  print *,'Try again.'
go to 10
endif

C *** PLOT AN X-Z PLANE (a.k.a. Y Section)
elseif(nview.eq.2) then
  iscn=0
  jscn=1
  kscn=0
  xscn=0.
  zscn=0.
20 print *
print 7,
& 'Enter Y distance (in ft) of desired view plane from'
& 'the left wall (Left Wall = 0.0; Right Wall = ',y,'):'
read (*,*) yscn
if(yscn.lt.0.0.or.yscn.gt.y) then
  print *
  print 5,'Input value must be between 0.0 and ',y
  print *,'Try again.'
go to 20
endif

C *** PLOT A Y-Z PLANE (a.k.a. X Section)
elseif(nview.eq.3) then
  iscn=1
  jscn=0
  kscn=0
  yscn=0.
  zscn=0.
30 print *
print 7,
& 'Enter X distance (in ft) of desired view plane from '
& 'the left wall (Left Wall = 0.0; Right Wall = ',x,'):'
read (*,*) xscn
if(xscn.lt.0.0.or.xscn.gt.x) then
  print *
  print 5,'Input value must be between 0.0 and ',x
  print *,'Try again.'
go to 30
endif

c *** ZOOM IN ON A PORTION OF THE LAST PLOT
elseif (nview.eq.4) then
40  print *, 'Enter the X & Y coordinates (expressed as a decimal'
40  print *, 'percentage of the whole plot, i.e. 25%=0.25) for the '
40  print *, 'lower left corner of the area to be enlarged.'
40  read (*,*) xfrnl,yfrnl
40  if(xfrnl.lt.0..or.xfrnl.gt.1..or.
40 &   yfrnl.lt.0..or.yfrnl.gt.1.) then
40    print *, 'All coordinates must be between 0.00 and 1.00!'
40    print *, 'Please correct and reenter.'
40    goto 40
40  endif

45  print *, 'Enter the X & Y coordinates (expressed as a decimal'
45  print *, 'percentage of the whole plot, i.e. 25%=0.25) for the '
45  print *, 'upper right corner of the area to be enlarged.'
45  read (*,*) xfrn2,yfrn2
45  if(xfrn2.lt.0..or.xfrn2.gt.1..or.
45 &   yfrn2.lt.0..or.yfrn2.gt.1.) then
45    print *, 'All coordinates must be between 0.00 and 1.00!'
45    print *, 'Please correct and reenter.'
45    goto 45
45  endif
45  if(xfrn2.lt.xfrnl.or.yfrn2.lt.yfrnl) then
45    print ('(ix,a,2(f4.2,a))'),
45    & 'These coordinates must be greater than ',xfrnl,
45    & 'and ',yfrnl, ' respectively.'
45    print *, 'Please correct and reenter.'
45    goto 45
45  endif

C *** ERROR ON ENTRY OF NVIEW
else
  print *, 'Incorrect response! Please enter 0 - 4'
  goto 99
endif
return
end

******************************************************************************************
SUBROUTINE INPUT
******************************************************************************************
*
* Input data from existing datafiles.
*
******************************************************************************************
implicit real*8 (a-h,o-z)
common/bl1/dx,dy,dz
common/bl2/x,y,h,tfir,twal,ta
common/bl7/ni,nj,nk
common/data/t(25,25,15),temp(24,24),tt(24,24),xtime,ttmax

C *** READ IN DATA FROM EXISTING DATA FILE
open(10,file='fire.dat',status='old')
read(10,*) x,y,h,tf1r,twal,ta
read(10,*) ni,nj,nk
rewind 10
close (10)

open (unit=9, file='plot.data', status='unknown')
read(9,1001) xtime,t
1001 format(4(f17.7))
rewind 9
close (9)

C *** DIMENSIONALIZE TEMPERATURES IN DEGRESS CELSIUS
do 77 i=1,ni+4
   do 77 j=1,nj+4
      do 77 k=1,nk+4
         t(i,j,k)=ta*t(i,j,k)/1.8-273.16
      77 continue
   return
end

***************************************************************************************
SUBROUTINE GRID
***************************************************************************************

implicit real*8 (a-h,o-z)
common/r4/xss(40), yss(40), zss(40), xs(40), ys(40), zs(40)
common/b11/dx, dy, dz
common/b12/x, y, h, tf1r, twal, ta
common/b17/ni, nj, nk

C *** GENERATION OF THE GRIDS
dx=x/(float(ni)*h)
dy=y/(float(nj)*h)
dz=h/(float(nk)*h)

C *** CALCULATE XSS,YSS,ZSS (COORDINATES OF STAGGERED CV'S)
do 10 i=3,ni+3
   xss(i)=(i-3)*dx
10 continue
xss(2)=xss(3)-twal/(h*12.)
xss(1)=xss(2)-twal/(h*12.)
xss(ni+4)=xss(ni+3)+twal/(h*12.)
xss(ni+5)=xss(ni+4)+twal/(h*12.)

   do 12 j=3,nj+3
      yss(j)=(j-3)*dy
12 continue
yss(2)=yss(3)-twal/(h*12.)
yss(1)=yss(2)-twal/(h*12.)
yss(nj+4)=yss(nj+3)+twal/(h*12.)
yss(nj+5)=yss(nj+4)+twal/(h*12.)

   do 14 k=3,nk+3
      zss(k)=(k-3)*dz
14 continue
zss(2)=zss(3)-tf1r/(h*12.)
**CONVERT TO CENTERED GRID FOR USE BY PROGRAM**

```fortran
do 20 i=1,ni+4
   xs(i)=(xss(i)+xss(i+1))/2
20 continue

do 22 j=1,nj+4
   ys(j)=(yss(j)+yss(j+1))/2
22 continue

do 24 k=1,nk+4
   zs(k)=(zss(k)+zss(k+1))/2
24 continue

return
end
```

**SUBROUTINE INTERPOL**

```fortran
C *** CONVERT TO CENTERED GRID FOR USE BY PROGRAM
do 20 i=1,ni+4
   xs(i)=(xss(i)+xss(i+1))/2
20 continue

do 22 j=1,nj+4
   ys(j)=(yss(j)+yss(j+1))/2
22 continue

do 24 k=1,nk+4
   zs(k)=(zss(k)+zss(k+1))/2
24 continue

return
end
```

**NONDIMENSIONALIZE NECESSARY VARIABLES**

```fortran
if(nview.ne.4) then
   xscn = xscn / h
   yscn = yscn / h
   zscn = zscn / h
endif
wth = y / h
bth = x / h
```
\begin{verbatim}
igi = i
endif
continue
cl = (xs(igi) - xs0) / (xs0 - xs(igi-1))
c2 = 1. - cl

** INTERPOLATION DONE HERE

do 20 j=1,nj+4
   do 20 k=1,nk+4
      temp(j,k) = cl * t(igi-1,j,k) + c2 * t(igi,j,k)
   continue
ixx = nj+4
jyy = nk+4
xx = wth
yy = 1.
do 22 i=1,i xx
   xxs(i) = ys(i)
continue
do 24 j=1,jyy
   yys(j) = zs(j)
continue
endif

** Y SECTION
if (jscn .ne. 0) then
   do 30 j=2,nj+4
      if (ys(j) .ge. yscn .and. ys(j-1) .lt. yscn) then
         jgi = j
      endif
   continue
c1 = (ys(jgi) - yscn) / (ys(jgi) - ys(jgi-1))
c2 = 1. - c1

** INTERPOLATION DONE HERE

do 40 i=1,ni+4
   do 40 k=1,nk+4
      temp(i,k) = cl * t(i,jgi-1,k) + c2 * t(i,jgi,k)
   continue
i xx = ni+4
jyy = nk+4
xx = bth
yy = 1.
do 42 i=1,i xx
   xxs(i) = xs(i)
continue
do 44 j=1,jyy
   yys(j) = zs(j)
continue
endif

** Z SECTION
if (kscn .ne. 0) then
   do 50 k=2,nk+4
      if (zs(k) .ge. zscn .and. zs(k-1) .lt. zscn) then
         kgi = k
      endif
      ...
\end{verbatim}
**INTERPOLATION DONE HERE**

```fortran
C  
do 60 i=l,ni+4
   do 60 j=l,nj+4
      temp(i,j) = cl * t(i,j,kgi-l) + c2 * t(i,j,kgi)
60 continue
ixx = ni+4
jyy = nj+4
xx = bth
yy = wth
do 62 i=l,ixx
   xxs(i) = xs(i)
62 continue
do 64 j=l,jyy
   yys(j) = ys(j)
64 continue
endif
```

**DETERMINE MAXIMUM TEMPERATURE ON SECTION**

```fortran
C  
tmax = 0.
do 70 i=l,ixx
   do 70 j=l,jyy
      if (temp(i,j) .gt. tmax) tmax = temp(i,j)
70 continue
print *, 'THE MAXIMUM TEMPERATURE IS', tmax
return
end
```

---

**DETERMINE LIMITS OF INTERPOLATION**

```fortran
C  
xlmt1 = xfrn1 * xx
xlmt2 = xfrn2 * xx
ylmt1 = yfrn1 * yy
ylmt2 = yfrn2 * yy
```

---

```
SUBROUTINE SCTN

*** In this subroutine the non-uniform grid is interpolated onto a uniform grid for plotting purposes. The routine has the option of "blowing up" or "zooming in on" a certain portion for detailed viewing.
```

---

```fortran
implicit real*8 (a-h,o-z)
common/data/t(25,25,15),temp(24,24),tt(24,24),xtime,ttmax
common/mt1/ixx,jyy
common/mt2/xx,yy
common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
common/ngs/xss(40),yss(40)
common/ugrd/iplot,jplot,ncrmax
common/fctn/xfrn1,xfrn2,yfrn1,yfrn2
```

---

```
C  
```
dx = (xmt2 - xmt1) / float(iplot - 1)
dy = (ymt2 - ymt1) / float(jplot - 1)

do 20 i=1,iplot
   do 20 j=1,jplot

   C *** LOCATE SECTION AND DEVELOP INTERPOLATION PARAMETERS
   xtmp = xmt1 + dx * float(i - 1)
   ytmp = ymt1 + dy * float(j - 1)
   do 10 ii=2,ixx
      if (xxs(ii) .gt. xtmp .and. xxs(ii-1) .lt. xtmp) then
         iil = ii
      endif
   10 continue
   cx = (xxs(iil) - xtmp) / (xxs(iil) - xxs(iil-1))
   do 15 jj=2,jyy
      if (yys(jj) .gt. ytmp .and. yys(jj-1) .lt. ytmp) then
         jjl = ii
      endif
   15 continue
   cy = (yys(jjl) - ytmp) / (yys(jjl) - yys(jjl-1))
   c11 = cx * cy
   c12 = cx * (1. - cy)
   c21 = (1. - cx) * cy
   c22 = (1. - cx) * (1. - cy)

   C *** INTERPOLATION DONE HERE
   tt(i,j) = c11 * temp(iil-1,jjl-1) + c12 * temp(iil-1,jjl) 
   &  + c21 * temp(iil,jjl-1) + c22 * temp(iil,jjl)

   20 continue

   c *** DETERMINE MAXIMUM INTERPOLATED TEMPERATURE
   ttmax = 0.
   do 30 i=1,iplot
      do 30 j=1,jplot
         if (tt(i,j) .gt. ttmax) ttmax = tt(i,j)
      30 continue
   print *, 'THE MAXIMUM INTERPOLATED TEMPERATURE IS', ttmax
   return
end

**************************************************************************************************
SUBROUTINE PLOT
**************************************************************************************************
*
* This subroutine plots the isotherms using the CONRAN routine of the NCAR Graphics package.
*
**************************************************************************************************

common/lmt2/xx,yy
common/data/t(25,25,15),temp(24,24),tt(24,24),xtime,ttmax
common/ugrd/iplot,jplot,ncmax
common/sctl/isctn,jscn,kscn,nview
common/sct2/xscn,yscn,zscn
common/fctn/xfrnl,xfrn2,yfrnl,yfrn2

159
common/b12/x,y,h,tflr,twal,ta
real*8 t,temp,tt,xtime,ttmax,xx,yy,xfrn1,xfrn2,yfrn1,yfrn2
real*8 xscn,yscn,zscn,x,y,h,tflr,twal,ta

dimension zdat(24,24),rwrk(1000),iwrk(1000),iama(120000)
dimension iasf(13)
dimension xcra(1000),ycra(1000)
dimension iaia(10),igia(10)
dimension lind(14)
character*10 llbs(15)
character*5 sec, elev
character*45 title
character*4 xll,x12,yll,y12

C *** DEFINE EXTERNAL SUBROUTINE TO SET COLORS
external COLRAM
C *** SET PARAMETERS REQUIRED BY NCAR GRAPHICS ROUTINES
data iasf/13*1/
data lind/2,3,4,5,6,7,8,9,10,11,12,13,14,15/

**** CONVER TO SINGLE PRECISION (NCAR WON'T WORK IN DOUBLE PRECISION) ****
C *** FIRE TIME WHEN DATA WAS TAKEN
time=xtime*h/1.0

C *** SECTION TO PLOT
if(iscn.gt.0) then
   scn=xscn*h
   xzoom1=xfrn1*y
   xzoom2=xfrn2*y
   yzoom1=yfrn1*h
   yzoom2=yfrn2*h
elseif(jscn.gt.0) then
   scn=yscn*h
   xzoom1=xfrn1*x
   xzoom2=xfrn2*x
   yzoom1=yfrn1*h
   yzoom2=yfrn2*h
else
   scn=zscn*h
   xzoom1=xfrn1*x
   xzoom2=xfrn2*x
   yzoom1=yfrn1*y
   yzoom2=yfrn2*y
endif

C *** SET VERTICAL DIMENSION OF OUTPUT DISPLAY
C   if(iscn.gt.0.or.jscn.gt.0) then
C      ndim=14
C   else
C      ndim=24
C   endif

C *** DETERMINE MAX AND MIN TEMPS FOR ENTIRE 3-D GRID
cmax = -2.0e20
cmin = 2.0e20
do 10 i=1,24
  do 10 j=1,24
    do 10 k=1,14
      if(t(i,j,k).gt.cmax) cmax = t(i,j,k)
      if(t(i,j,k).lt.cmin) cmin = t(i,j,k)
    10 continue
  20 continue

c *** CALCULATE THE SPACINGS BETWEEN CONTOURS
dc=(cmax-cmin)/real(ncrmax-1)

***** RUN NCAR GRAPHICS PACKAGE *****
C *** START NCAR GRAPHICS
  call GOPKS (6,0)
  call GOPWK (1,2,1)
  call GACWK (1)

C *** TURN OFF CLIPPING SO WORDS WILL PLOT
  call GSCLIP (0)
  call GSASF (iasf)
  call GSFAIS (1)

C *** DEFINE COLORS
  call DFCLRS

C *** DEFINE VIEWPORT AND PLOT CONTOURS
  call CPSETR ('VPS - VIEWPORT SHAPE',0)
  if(kscn.gt.0) then
    call CPSETR ('VPB - VIEWPORT BOTTOM',.15)
    call CPSETR ('VPT - VIEWPORT TOP' ,.95)
  else
    call CPSETR ('VPB - VIEWPORT BOTTOM',.25)
    call CPSETR ('VPT - VIEWPORT TOP' ,.65)
  endif
  call CPSETR ('VPL - VIEWPORT LEFT',.1)
  call CPSETR ('VPR - VIEWPORT RIGHT',.9)
  call CPSETI ('NOF - NUMERIC OMISSION FLAGS',0)
  call CPSETI ('CLS - CONTOUR LEVEL SELECTOR',ncrmax)
  call CPSETR ('CLS - CONTOUR INTERVAL SPECIFIER',dc)
  call CPSETI ('LLP - LINE LABEL POSITIONING',0)
  call CPSETR ('CMN - CONTOUR MINIMUM',cmin)
  call CPSETR ('CMX - CONTOUR MAXIMUM',cmax)
  call CPRECT (zdat,24,24,ndim,rwrk,1000,iwrk,1000)
  call ARINAM (iama,120000)
  call CPCLAM (zdat,rwrk,iwrk,iama)
  call ARSCAM (iama,xcra,ycra,1000,iaia,igia,10,COLRAM)
  call GSPCLI (0)
  call CPCLDR (zdat,rwrk,iwrk)
  call GSPCLI (1)

C *** CONVERT REAL VARIABLES TO CHARACTER VARIABLES FOR PLOTTING
C *** AND SET COLORS FOR PLOTTING
call CPGETI ('NCL - NUMBER OF CONTOUR LEVELS', ncl)
do 102 i=1, ncl
call CPSETI ('PAI - PARAMETER ARRAY INDEX',i)
call CPSETI ('AIA - AREA IDENTIFIER ABOVE',i)
call CPSETI ('AIB - AREA IDENTIFIER BELOW',i-1)
call CPGETR ('CLV - CONTOUR LEVEL VALUES', zlib)
call CPSETR ('ZDV - Z DATA VALUE', zlib)
   call CPGETC ('ZDV - Z DATA VALUE', IIbs(i))
102 continue

call CPSETR ('ZDV - Z DATA VALUE', time)
call CPGETC ('ZDV - Z DATA VALUE', sec)
call CPSETR ('ZDV - Z DATA VALUE', elev)
call CPSETR ('ZDV - Z DATA VALUE', xzoom1)
call CPGETC ('ZDV - Z DATA VALUE', xl1)
call CPSETR ('ZDV - Z DATA VALUE', xzoom2)
call CPGETC ('ZDV - Z DATA VALUE', xI2)
call CPSETR ('ZDV - Z DATA VALUE', yzoom1)
call CPGETC ('ZDV - Z DATA VALUE', y11)
call CPSETR ('ZDV - Z DATA VALUE', yzoom2)
call CPGETC ('ZDV - Z DATA VALUE', y12)
c
***** CONSTRUCT LABEL BAR
   call LBSETI ('CBL - COLOR OF BOX LINES', 0)
   if (kscn.gt.0) then
      call LBLBAR (0, .05, .95, .05, .10, 14, 1., 1., 1., ind0, 11bs, 15, 1)
   else
      call LBLBAR (0, .05, .95, .15, .20, 14, 1., 1., 1., ind0, 11bs, 15, 1)
   endif

***** LABEL AXIS AND TITLE PLOT *****
c
***** X SECTION
   if (iscn.gt.0) then
      title='Y-Z PROFILE (X = '//elev//' FT.) AT '//sec//' SEC.'
      call PLCHHQ(3., 25., .015, 0, -1.)
      call PLCHHQ(9., -6., 'TEMPERATURE (CELSIUS)', 01, 0., -1.)
      call PLCHHQ(10., 0., 'BREADTH (Y-DIR)', 01, 0., -1.)
      call PLCHHQ(1., 0., 11., 01, 0., -1.)
      call PLCHHQ(23., 0., x12, 01, 0., -1.)
      call PLCHHQ(.5, 11.0, 'HEIGHT (Z-DIR)', 01, 90., 0.)
      call PLCHHQ(.5, 1.5, y11, 01, 90., 0.)
      call PLCHHQ(.5, 23., y12, 01, 90., 0.)
   endif

   elseif (jscn.gt.0) then
      title='X-Z PROFILE (Y = '//elev//' FT.) AT '//sec//' SEC.'
      call PLCHHQ(3., 25., .015, 0, -1.)
      call PLCHHQ(9., -6., 'TEMPERATURE (CELSIUS)', 01, 0., -1.)
      call PLCHHQ(10., 0., 'DEPTH (X-DIR)', 01, 0., -1.)
      call PLCHHQ(1., 0., x11, 01, 0., -1.)
call PLCHHQ(23.,0.,x12.,01.,0.,-1.)
call PLCHHQ(.5,11.0,'HEIGHT (Z-DIR)',.01,90.,0.)
call PLCHHQ(.5,1.5,y11.,01.,90.,0.)
call PLCHHQ(.5,23.,y12.,01.,90.,0.)

C *** Z SECTION
elseif(kscn.gt.0) then
title='PLAN VIEW (Z = '/elev//' FT.) AT '/sec//' SEC.'
call PLCHHQ(3.5,24.5,title,.015,0.,-1.)
call PLCHHQ(8.,'-2.5,' TEMPERATURE (CELSIUS)',.01,0.,-1.)
call PLCHHQ(10.,5., 'DEPTH (X-DIR)',.01,0.,-1.)
call PLCHHQ(1.,5.,x11.,01.,0.,-1.)
call PLCHHQ(23.,5.,x12.,01.,0.,-1.)
call PLCHHQ(.5,9.0,'BREADTH (Y-DIR)',.01,90.,-1.)
call PLCHHQ(.5,1.5,y11.,01.,90.,0.)
call PLCHHQ(.5,23.5,y12.,01.,90.,0.)
endif

C *** DRAW BOUNDARY AROUND VIEWPORT
call BNDARY

C *** ADVANCE FRAME FOR NEXT PLOT
call FRAME

C *** FINISHED WITH NCAR GRAPHICS
call GCLRKW (1,1)
call GDAWK (1)
call GCLWK (1)
call GCLKS

return
end

*******************************************************************************
SUBROUTINE DFCLRS
*******************************************************************************
*
* Define colors using Red-Green-Blue (RGB) triples.
*
*******************************************************************************

dimension rgbv(3,15)
data rgbv/ 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,
& 0.00,0.00,1.00,0.00,0.00,0.00,0.50,0.75,0.00,0.75,0.50,
& 0.00,1.00,0.00,0.65,1.00,0.00,1.00,1.00,0.00,
& 1.00,0.75,0.00,1.00,0.50,0.00,1.00,0.00,0.00,
& 1.00,0.10,0.40,1.00,0.40,0.40,0.70,1.00,0.70,1.00/

c *** DEFINE DEFAULT BACKGROUND COLOR AS WHITE
call GSCR (1,0,1.,1.,1.,1.)

c
C *** DEFINE REMAINING COLORS
C *** (NOTE: i=1 is default foreground color and is set to black)
do 101 i=1,15
   call GSCR (1,i,rgbv(1,i),rgbv(2,i),rgbv(3,i))
101 continue
return
end

*********************************************************A--*--
SUBROUTINE COLRAM (xcra, ycra, ncra, iaia, igia, naia)
*********************************************************A---
* *
* This subroutine is used by NCAR GRAPHICS (call DEFCLRS) to assign *
* the colors to the contour levels. It must be declared EXTERNAL *
* before any calls are made to any NCAR subroutines. *
*********************************************************A--
* *
* dimension xcra(*), ycra(*), iaia(*), igia(*)
* ifll = 1
* do 101 i=1, naia
* 101 if (iaia(i) .lt. 0) ifll = 0
* continue
* if (ifll .ne. 0) then
* 102 ifll = 0
* do 102 i=1, naia
* 102 if (igia(i) .eq. 3) ifll = iaia(i)
* continue
* if (ifll .gt. 0 .and. ifll .lt. 15) then
* call GSFACI (ifll)
* call GFA (ncra-1, xcra, ycra)
* endif
* return
* end

*********************************************************A--
SUBROUTINE BNDARY
*********************************************************A--
* *
* This subroutine defines the edge of the plot frame and draws a line *
* at its location. *
*********************************************************A--
* *
call PLOTIF (0., 0., 0)
call PLOTIF (1., 0., 1)
call PLOTIF (1., 1., 1)
call PLOTIF (0., 1., 1)
call PLOTIF (0., 0., 1)
call PLOTIF (0., 0., 2)

return
end
APPENDIX D. PROGRAM VELOCITY

This program plots two dimensional velocity profiles, using the National Center for Atmospheric Research (NCAR) Graphics software, from the three dimensional velocity field generated by the program FIRE.

PROGRAM VELOCITY

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

This program uses NCAR GRAPHICS to calculate and plot velocity vectors in an arbitrary two dimensional section with uniform grid spacing from a non-uniform three dimensional grid using linear interpolation routines.

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

Variables:

nstop : Control variable to end program
xscn : X location where Y-Z plane to be taken
yscn : Y location where X-Z plane to be taken
zscn : Z location where X-Y plane to be taken
iscn : If >0 hold X constant, else =0
jscn : If >0 hold Y constant, else =0
kscn : If >0 hold Z constant, else =0
xfrn1 : Starting 2-D, X coordinate of section to zoom in on
xfrn2 : Ending 2-D, X coordinate of section to zoom in on
yfrn1 : Starting 2-D, Y coordinate of section to zoom in on
yfrn2 : Ending 2-D, Y coordinate of section to zoom in on
t() : Dummy variable necessary due to format of input datafile
u() : Non-dimensional input value of X component of velocity
v() : Non-dimensional input value of Y component of velocity
w() : Non-dimensional input value of Z component of velocity
iplot : Number of grids in X direction of 2-D grid
jplot : Number of grids in Y direction of 2-D grid
x : Non-dimensional X length of compartment to model
y : Non-dimensional Y length of compartment to model
h : Non-dimensional Z length of compartment to model
tflr : Non-dimensional thickness of compartment floor/ceiling
twal : Non-dimensional thickness of compartment walls
ta : Non-dimensional ambient temperature
dx : Non-dimensional 3-D grid size inside compartment, X dir.
dy : Non-dimensional 3-D grid size inside compartment, Y dir.
dz : Non-dimensional 3-D grid size inside compartment, Z dir.
ni : Number of cells across compartment, X dir.
nj : Number of cells across compartment, Y dir.
nk : Number of cells across compartment, Z dir.
nview : Input value of desired plot (from SUBROUTINE SECTION)

C *** READ IN DATA GENERATED BY PROGRAM "FIRE"
CALL INPUT

C *** GENERATE GRID
    CALL GRID

C *** DEFINE DESIRED SECTION
    5 CALL SECTION(NSTOP)

C *** CHECK IF FINISHED WITH PROGRAM
    IF (NSTOP.GT.0) GOTO 9999

C *** LOCATE DESIRED SECTION
    CALL INTERPOL

C *** INTERPOLATE TO A 2-D PLOT
    CALL SCTN

C *** GENERATE VELOCITY VECTOR PLOT USING NCAR GRAPHICS
    CALL PLOT

C *** LOOP BACK FOR ANOTHER SECTION
    GOTO 5

C *** FINISHED WITH PROGRAM
7999 END

*****************************************************************************
BLOCK DATA
*****************************************************************************

* The variables IPLOT and JPLOT define the number of uniformly spaced *
* grids to be used in generating the 2-D plot and must be changed if *
* a different grid spacing is desired.
*****************************************************************************

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/UGRD/IPLOT,JPLOT
COMMON/SCT1/ISCN,JSCN,KSCN,NVIEW

DATA IPLOT,JPLOT/24,24/
DATA NVIEW/0/

END

*****************************************************************************
SUBROUTINE INPUT
*****************************************************************************

* Input data from existing datafiles.
*****************************************************************************

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/B11/DX,DY,DZ
COMMON/B12/T,X,Y,
COMMON/B17/NI,NJ,NK
COMMON/DATA/T(25,25,15),U(25,25,15),V(25,25,15),W(25,25,15),
C *** READ IN DATA FROM EXISTING DATA FILE
open(10, file='fire.dat', status='old')
read(10,*) x,y,h,tflr,twal,ta
read(10,*) ni,nj,nk
rewind 10
close (10)

open (unit=9, file='plot.data', status='unknown')
read(9,1000) xtime,t,u,v,w
1000 format(4(f17.7))

C *** DIMENSIONALIZE VELOCITIES TO CM/SEC
  do 10 i=1,ni+4
    do 10 j=1,nj+4
      do 10 k=1,nk+4
        u(i,j,k)=u(i,j,k)*30.25
        v(i,j,k)=v(i,j,k)*30.25
        w(i,j,k)=w(i,j,k)*30.25
    10 continue
  return
end

SUBROUTINE GRID

implicit real*8 (a-h,o-z)
common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
common/b11/dx,dy,dz
common/b12/x,y,h,tflr,twal,ta,xtime
common/b17/ni,nj,nk

C *** GENERATION OF THE GRIDS
  dx=x/(float(ni)*h)
  dy=y/(float(nj)*h)
  dz=h/(float(nk)*h)

C *** CALCULATE XSS,YSS,ZSS (COORDINATES OF STAGGERED CV'S)
  do 10 i=3,ni+3
    xss(i)=(i-3)*dx
  10 continue
  xss(2)=xss(3)-twal/(h*12.)
  xss(1)=xss(2)-twal/(h*12.)
  xss(ni+4)=xss(ni+3)+twal/(h*12.)
  xss(ni+5)=xss(ni+4)+twal/(h*12.)

  do 12 j=3,nj+3
    yss(j)=(j-3)*dy
  12 continue
  yss(2)=yss(3)-twal/(h*12.)
  yss(1)=yss(2)-twal/(h*12.)
  yss(nj+4)=yss(nj+3)+twal/(h*12.)
  yss(nj+5)=yss(nj+4)+twal/(h*12.)
do 14 k=3,nk+3
    zss(k)=(k-3)*dz
14 continue
    
zss(2)=zss(3)-tflr/(h*12.)
zss(1)=zss(2)-tflr/(h*12.)
zss(nk+4)=zss(nk+3)+tflr/(h*12.)
zss(nk+5)=zss(nk+4)+tflr/(h*12.)

C *** CONVERT TO CENTERED GRID FOR USE BY PROGRAM
do 20 i1=1,ni+4
    xs(i)=(xss(i)+xss(i+1))/2
20 continue
    
do 22. j1,nj+4
    ys(j)=(yss(j)+yss(j+1))/2
22 continue
    
do 24 k1,nk+4
    zs(k)=(zss(k)+zss(k+1))/2
24 continue
    
return
end

********************************************************************
SUBROUTINE SECTION(nstop)
********************************************************************

implicit real*8 (a-h,o-z)
common/fctn/xfrnl,xfrn2,yfrnl,yfrn2
    common/sctl/iscn,jscn,kscn,nview
    common/sct2/xscn,yscn,zscn
    common/bl2/x,y,h,tflr,twal,ta,xtime

99 print *
   print *
   print *,'ENISR FOR:'
   print *,'----- -----
   print *,' 0 Finished with program'
   print *,' 1 Plan View (X-Y plane)'
   print *,' 2 Elevation (X-Z plane)'
   print *,' 3 Elevation (Y-Z plane)'
   if(nview.le.0) goto 97
   print *,' 4 Enlargement of portion of previous plot'
97 print *
   print *,'Enter your desired vector field plot:'
   read (*,*), nview
   nstop=0
   xfrnl=0.
   xfrn2=1.
   yfrnl=0.
   yfrn2=1.

C *** END PROGRAM
if (nview.eq.0) then
    nstop=9999

   168
C *** PLOT AN X-Y PLANE (aka Z SECTION)
elseif (nview.eq.1) then
  iscn=0
  jscn=0
  kscn=1
  xscn=0.
  yscn=0.
10  print *
    format (1x,a/1x,a-f5.2,a)
    print 7,
    & 'Enter desired height of desired plot above the floor',
    & '(Floor = 0.0; Ceiling = ',h,')'
    read (*,*) zscn
    if(zscn.lt.0..or.zscn.gt.h) then
      print *
      print 5,'Input value must be between 0.0 and ',h
      print *,'Try again.'
      go to 10
    endif
C *** PLOT AN X-Z PLANE (aka Y SECTION)
elseif (nview.eq.2) then
  iscn=0
  jscn=1
  kscn=0
  xscn=0.
  zscn=0.
20  print *
    print 7,
    & 'Enter Y distance (in ft) of desired plot from the left wall',
    & '(Left Wall = 0.0; Right Wall = ',y,')'
    read (*,*) yscn
    if(yscn.lt.0..or.yscn.gt.y) then
      print *
      print 5,'Input value must be between 0.0 and ',y
      print *,'Try again.'
      go to 20
    endif
C *** PLOT A Y-Z PLANE (aka X SECTION)
elseif (nview.eq.3) then
  iscn=1
  jscn=0
  kscn=0
  zscn=0.
  yscn=0.
30  print *
    print 7,
    & 'Enter X distance (in ft) of desired plot from the right wall',
    & '(Left Wall = 0.0; Right Wall = ',x,')'
    read (*,*) xscn
    if(xscn.lt.0..or.xscn.gt.x) then
      print *
      print 5,'Input value must be between 0.0 and ',x
print *, 'Try again.'
go to 30
endif

C *** ZOOM IN ON A PORTION OF THE LAST PLOT
elseif (nview.eq.4) then
40 print *
print *, 'Enter the X & Y coordinates (expressed as a decimal'
print *, 'percentage of the whole plot, i.e. 25%=0.25) for the '
print *, 'lower left corner of the area to be enlarged.'
read (*,*) xfrnl,yfrnl
if(xfrnl.lt.0..or.xfrnl.gt.1..or.
& yfrnl.lt.0..or.yfrnl.gt.1..) then
print *, 'All coordinates must be between 0.00 and 1.00!'
print *, 'Please correct and reenter.'
goto 40
endif
45 print *
print *, 'Enter the X & Y coordinates (expressed as a decimal'
print *, 'percentage of the whole plot, i.e. 25%=0.25) for the '
print *, 'upper right corner of the area to be enlarged.'
read (*,*) xfrn2,yfrn2
if(xfrn2.lt.0..or.xfrn2.gt.1..or.
& yfrn2.lt.0..or.yfrn2.gt.1..) then
print *, 'All coordinates must be between 0.00 and 1.00!'
print *, 'Please correct and reenter.'
goto 45
endif
if(xfrn2.gt.xfrnl.or.yfrn2.gt.yfrnl) then
print ('(lx,a,2(f4.2,a))'),
& 'These coordinates must be greater than ',xfrnl,
& and ',yfrnl, respectively.'
print *, 'Please correct and reenter.'
goto 45
endif

C *** ERROR ON ENTRY OF NVIEW
else
print *
print *, 'Incorrect response! Please enter 0 - 4'
goto 99
endif
return
end

****************************************************************************
SUBROUTINE INTERPOL
*****************************************************************************
* * This subroutine interpolates to the desired section. It is *
* necessary since any arbitrary section would fall in between two *
* grid points and must be linearly interpolated. *
*****************************************************************************
C *** NON-DIMENSIONALIZE REQUIRED VARIABLES
if(nview.ne.4) then
    xscn = xscn / h
    yscn = yscn / h
    zscn = zscn / h
endif
wth = y / h
bth = x / h

***** LOCATE SECTION OF INTEREST *****

C *** X SECTION
if (iscn .ne. 0) then
    do 10 i=2,ni+4
        if (xs(i) .ge. xscn .and. xs(i-1) .lt. xscn) then
            igi = i
        endif
    10 continue
    c1 = (xs(igi) - xscn) / (xs(igi) - xs(igi-1))
    c2 = 1. - c1

C *** INTERPOLATION DONE HERE
    do 20 j=2,nj+4
        do 20 k=2,nk+4
            vv(j,k) = c1 * w(igi-1,j,k) + c2 * w(igi,j,k)
            uu(j,k) = c1 * v(igi-1,j,k) + c2 * v(igi,j,k)
        20 continue
    ixx = nj+4
    jyy = nk+4
    xx = wth
    yy = 1.
    do 22 i=1,ixx
        xxs(i) = ys(i)
        xl(i) = yss(i)
    22 continue
    do 24 j=1,jyy
        yys(j) = zs(j)
        yl(j) = zss(j)
    24 continue
endif

C *** Y SECTION
if (jscn .ne. 0) then
    do 30 j=2,nj+4
        if (ys(j) .ge. yscn .and. ys(j-1) .lt. yscn) then
            jgi = j
        endif
    30 continue
    c1 = (ys(jgi) - yscn) / (ys(jgi) - ys(jgi-1))
    c2 = 1. - c1
    C *** INTERPOLATION DONE HERE

    do 40 i=2,ni+4
        do 40 k=2,nk+4
            uu(i,k) = c1 * u(i,jgi-1,k) + c2 * u(i,jgi,k)
            vv(i,k) = c1 * v(i,jgi-1,k) + c2 * v(i,jgi,k)
        40 continue
    end if
    C *** Z SECTION

    if (kscn .ne. 0) then
        do 50 k=2,nk+4
            if (zs(k) .ge. zscn .and. zs(k-1) .lt. zscn) then
                kgi = k
            endif
        50 continue
        c1 = (zs(kgi) - zscn) / (zs(kgi) - zs(kgi-1))
        c2 = 1. - c1
        C *** INTERPOLATION DONE HERE

        do 60 i=2,ni+4
            do 60 j=2,nj+4
                uu(i,j) = c1 * u(i,jgi-1,k) + c2 * u(i,jgi,k)
                vv(i,j) = c1 * v(i,jgi-1,k) + c2 * v(i,jgi,k)
            60 continue
        end if
    endif
**DETERMINE MAXIMUM VELOCITY COMPONENTS**

\[
x_c = 0.
\]
\[
y_c = 0.
\]

\[
do 70 i=1,ixx
\]
\[
do 70 j=1,jyy
\]
\[
\text{if (abs(uu(i,j)) .gt. } cx \text{ then } cx = \text{abs(uu(i,j))}
\]
\[
\text{if (abs(vv(i,j)) .gt. } cy \text{ then } cy = \text{abs(vv(i,j))}
\]
\[
70 \text{ continue}
\]

\[
\text{print *,'THE MAXIMUM X-DIRECTION VELOCITY IS',cx}
\]
\[
\text{print *,'THE MAXIMUM Y-DIRECTION VELOCITY IS',cy}
\]

**RETURN**

**SUBROUTINE SCTN**

\[
\text{implicit real*8 (a-h,o-z)}
\]
\[
\text{common/data/t(25,25,15),u(25,25,15),v(25,25,15),w(25,25,15),}
\]
\[
\text{& uu(24,24),vv(24,24),uul(24,24),vvl(24,24)}
\]
\[
\text{common/lmt1/ixx,jyy}
\]
\[
\text{common/lmt2/xx,yy}
\]
\[
\text{common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)}
\]
\[
\text{common/ngs/xss(40),yss(40),x1(40),y1(40)}
\]
\[
\text{common/ugrd/iplot,jplot}
\]
\[
\text{common/fctn/xfrnl,xfrn2,yfrnl,yfrn2}
\]

**DETERMINE LIMITS OF INTERPOLATION**

\[
x_{lmt1} = xfrn1 * xx
\]
\[
x_{lmt2} = xfrn2 * xx
\]
\[
y_{lmt1} = yfrn1 * yy
\]
\[
y_{lmt2} = yfrn2 * yy
\]

\[
dx = (x_{lmt2} - x_{lmt1}) / \text{float(iplot - 1)}
\]
\[
dy = (y_{lmt2} - y_{lmt1}) / \text{float(jplot - 1)}
\]

**LOCATE SECTION AND DEVELOP INTERPOLATION PARAMETERS**

\[
do 20 i=1,iplot
\]
\[
do 20 j=1,jplot
\]
\[
\text{xtmp = x_{lmt1} + dx * float(i - 1)}
\]
\[
ytmp = y_{lmt1} + dy * float(j - 1)
\]
\[
do 10 ii=2,ixx
\]
\[
\text{if (xss(ii) .ge. xtmp .and. xss(ii-1) .lt. xtmp) then}
\]
\[
ii1 = ii
\]
\[
\text{endif}
\]
\[
\text{if (x1(ii) .ge. xtmp .and. x1(ii-1) .lt. xtmp) then}
\]

173
ii2 = ii
endif
continue

10
continue

15
continue

174
SUBROUTINE PLOT
**************************************************************************
 *                                                                 *
 * This subroutine plots the velocity vectors using the VELVCT routine *
 * of the NCAR Graphics package.                                       *
 *
**************************************************************************

common/data/t(25,25,15),u(25,25,15),v(25,25,15),w(25,25,15),
 & uu(24,24),vv(24,24),uul(24,24),vvl(24,24)
common/1mt2/xx,yy
common/ugrd/iplot,jplot
common/sctl/iscn,jscn,kscp,nview
common/sct2/xscn,yscn,zscn
common/bl2/x,y,h,tflr,twal,ta,xtime
common/fctn/xfrn1,xfrn2,yfrn1,yfrn2

external TAG

real*8 t,u,v,w,uu,vv,uul,vvl,xx,yy
real*8 xscn,yscn,zscn,x,y,h,tflr,twal,ta,xtime
real*8 xfrn1,xfrn2,yfrn1,yfrn2
character*5 elev,sec
character*45 title
character*4 xll,xl2,yll,yl2
dimension varx(24,24),vary(24,24),spv(2)

C *** SET PARAMETERS REQUIRED FOR NCAR GRAPHICS ROUTINES
data spv/2=0./

****** CONVERT TO SINGLE PRECISION (NCAR WON'T WORK IN DOUBLE PRECISION) ****.

C *** FIRE TIME WHEN DATA WAS TAKEN
time=xtime*h/1.0

C *** SECTION TO PLOT
if(iscn.gt.0) then
  scn=xscn*h
  xzoom1=xfrn1+y
  xzoom2=xfrn2+y
  yzoom1=yfrn1*h
  yzoom2=yfrn2*h
elseif(jscn.gt.0) then
  scn=yscn*h
  xzoom1=xfrn1+x
  xzoom2=xfrn2+x
  yzoom1=yfrn1+h
  yzoom2=yfrn2+h
else
  scn=zscn*h
  xzoom1=xfrn1+x
  xzoom2=xfrn2+x
  yzoom1=yfrn1+y
  yzoom2=yfrn2+y
endif

175
C *** CONVERT REAL NUMBERS TO CHARACTERS FOR USE IN TITLE
   call TAG(scn,elev)
   call TAG(time,sec)
   call TAG(xzoom1,xl1)
   call TAG(xzoom2,xl2)
   call TAG(yzoom1,yl1)
   call TAG(yzoom2,yl2)

C *** DEFINE DIMENSIONS OF PLOT AREA
   xlft = 0.125
   xrgt = xlft + 0.75*xx/2.
   ybot = 0.20
   ytop = ybot + 0.75*yy/2.
   x1 = 24.
   if (kscn.gt.0) then
      y1 = 24.
   else
      y1 = 14.
   endif

C *** VELOCITY COMPONENTS
   do 20 i=1,iplot
      do 20 j=l,jplot
         varx(i,j) = uul(i,j)
         vary(i,j) = vvl(i,j)
   20 continue

C *** START WITH NCAR GRAPHICS
   call GOPKS (6,0)
   call GOPWK (1,2,1)
   call GACWK (1)

C *** TURN OFF CLIPPING SO CHARACTERS ARE PRINTED OUTSIDE PLOT
   call GSCLIP(0)

C *** DEFINE BOUNDARIES AND SET PERIMETER FOR VECTOR PLOT
   call SET (xlft,xrgt,ybot,ytop,l.,xl,l.,yl,l)
   call PERIM (1,0,1,0)

C  PLOT VELOCITY VECTORS
   call VELVCT (varx,iplot,vary,iplot,iplot,jplot,0.,0.,=1,O,0,spv)

C *** LABEL AXIS AND TITLE PLOT *****
   c *** X SECTION
      if(iscn.gt.0) then
         title='Y-Z ELEVATION (X = //elev//' FT.) AT //sec//' SEC.'
         call PLCHHQ((xlft+xrgt)/2.+0.75,ybot-2.5,title,.015,0.,-1.)
         call PLCHHQ(10.,5.,'BREADTH (Y-DIR)',.01,0.,-1.)
         call PLCHHQ(1.,5.,x11,.01,0.,-1.)
         call PLCHHQ(23.,5.,x12,.01,0.,-1.)
         call PLCHHQ(.5,7.0,'HEIGHT (Z-DIR)',.01,90.,0.)
         call PLCHHQ(.5,1.5,yl1,.01,90.,0.)
         call PLCHHQ(.5,13.5,yl2,.01,90.,0.)
      endif

176
C *** Y SECTION
elseif(jscn.gt.0) then
  title='X-Z ELEVATION (Y = //elev//' FT.) AT //sec// SEC.'
call PLCHHQ((xlft+xrgt)/2.+0.75,ybot-2.5,title,.015,0..-1.)
call PLCHHQ(10.5,'DEPTH (X-DIR)',01,0.,-1.)
call PLCHHQ(1.,5,x11,.01,0.,-1.)
call PLCHHQ(23.5,x12,.01,0.,-1.)
call PLCHHQ(.5,7.0,'HEIGHT (Z-DIR)',.01,90.,0.)
call PLCHHQ(.5,1.5,y11,.01,90.,0.)
call PLCHHQ(.5,13.5,y12,.01,90.,0.)
  endif
C *** Z SECTION
elseif(kscn.gt.0) then
  title='PLAN VIEW (Z = //elev//' FT.) AT //sec// SEC.'
call PLCHHQ((xlft+xrgt)/2.+1.5,ybot-2.5,title,.015,0.,-1.)
call PLCHHQ(10.5,'DEPTH (X-DIR)',.01,0.,-1.)
call PLCHHQ(1.,5,x11,.01,0.,-1.)
call PLCHHQ(23.5,x12,.01,0.,-1.)
call PLCHHQ(.5,9.0,'BREADTH (Y-DIR)',.01,90.,-1.)
call PLCHHQ(.5,1.5,y11,.01,90.,0.)
call PLCHHQ(.5,23.5,y12,.01,90.,0.)
endif
C *** FINISHED WITH NCAR GRAPHICS
call GCLRWK (1,1)
call GDAWK (1)
call GCLWK (1)
call GCLKS
return
end

**************************************************************************
SUBROUTINE TAG(scn,elev)
**************************************************************************
* This subroutine converts scn to a character value for use in the *
* title of the plot. *
**************************************************************************

character*1 e(0:9)
character*5 elev

data e/ '0', '1', '2', '3', '4',
    '5', '6', '7', '8', '9' /
if(scn.ge.100..and.scn.lt.1000.) then
  il=int(scn/100.)
  i2=int((scn-real(il)*100.)/10.)
  i3=int(scn-(real(il)*100.+real(i2)*10))
  i4=int((scn*10.-int(scn*10.))*10.)
elev=e(il)//e(i2)//e(i3)//'.//'//e(i4)
elev=e ila 100...and.scn.11.100.) then
  i2=int(scn/10.)
  i2=int(scn-real(i1)*10.)
177
i3=int((scn-int(scn))*10.)
i4=nint((scn*10.-int(scn*10.))*10.)
elev=e(i1)//e(i2)//'.'//e(i3)//e(i4)
elseif(scn.lt.10) then
  i1=int(scn)
i2=int((scn-int(scn))*10.)
i3=int((scn*10.-int(scn*10.))*10.)
i4=nint((scn*100.-int(scn*100.))*10.)
elev=e(i1)//'.'//e(i2)//e(i3)//e(i4)
endif

return
end
LIST OF REFERENCES


180


# INITIAL DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No.</th>
<th>Copies</th>
<th>Initial Distribution List</th>
</tr>
</thead>
</table>
| 1.  | 2      | Defense Technical Information Center  
|     |        | Cameron Station  
|     |        | Alexandria, VA 22304-6145 |
| 2.  | 2      | Library, Code 52  
|     |        | Naval Postgraduate School  
|     |        | Monterey, CA 93943-5002 |
| 3.  | 1      | Chairman, Code ME  
|     |        | Department of Mechanical Engineering  
|     |        | Naval Postgraduate School  
|     |        | Monterey, CA 93943-5000 |
| 4.  | 1      | Naval Engineering Curricular Office  
|     |        | Code 34  
|     |        | Naval Postgraduate School  
|     |        | Monterey, CA 93943-5000 |
| 5.  | 2      | Professor M.D. Kelleher, Code ME kk  
|     |        | Department of Mechanical Engineering  
|     |        | Naval Postgraduate School  
|     |        | Monterey, CA 93943-5000 |
| 6.  | 1      | Dr. Patricia Tatum  
|     |        | Code 6183  
|     |        | Naval Research Laboratory  
|     |        | Washington, DC 20375 |
| 7.  | 1      | Mr. James Gagorik  
|     |        | Code 211, ONT  
|     |        | Balston Towers #1  
|     |        | 800 N. Quincy  
|     |        | Arlington, VA 22217 |
| 8.  | 1      | Commandant (G-MTH-2)  
|     |        | United States Coast Guard  
|     |        | 2100 2nd St. SW  
|     |        | Washington, DC 20593 |
| 9.  | 1      | Coast Guard Law Library  
|     |        | United States Coast Guard  
|     |        | 2100 2nd St. SW  
|     |        | Washington, DC 20593 |

183
10. Professor K.T. Yang  
Department of Aerospace and Mechanical Engineering  
University of Notre Dame  
Notre Dame, IN 46556

11. Mr. Kent Farmer  
Code 38903, Bldg 31618  
Naval Air Warfare Center  
China Lake, CA 93555

12. LT Kenneth J. Thorkildsen  
U.S. Coast Guard Marine Safety Center  
400 7th Street, SW  
Washington, DC 20590-0001