1991

Numerical field model simulation of full-scale fire tests in a closed spherical/cylindrical vessel using advanced computer graphics techniques.

McCarthy, Timothy G.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

NUMERICAL FIELD MODEL SIMULATION OF
FULL-SCALE FIRE TESTS IN A CLOSED
SPHERICAL/CYLINDRICAL VESSEL USING
ADVANCED COMPUTER GRAPHICS TECHNIQUES

by

Timothy G. McCarthy

SEPTEMBER 1991

Thesis Advisor: M.D. Kelleher

Unclassified

CURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release: Distribution is unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) ME	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBER PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
NUMERICAL FIELD MODEL SIMULATION OF FULL-SCALE FIRE TESTS IN A CLOSED SPHERICAL/CYLINDRICAL VESSEL USING ADVANCED COMPUTER GRAPHICS TECHNIQUES

12. PERSONAL AUTHORS
TIMOTHY G. McCARTHY

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM ______ TO ______	SEPTEMBER 1991	157

16. SUPPLEMENTARY NOTATION
The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block numbers)
FIELD	GROUP	SUB-GROUP	field model, fire simulation, fire modeling, numerical fire model, fires in enclosed vessels, enclosed fires

19. ABSTRACT (Continue on reverse if necessary and identify by block numbers)

Personnel and equipment casualties, caused by shipboard fires have adversely affected overall readiness of the U.S. Navy for centuries. Understanding the phenomena of fire in enclosed spaces, such as those found on surface ships and submarines, will greatly enhance the Navy's ability to combat or prevent them. This computer model was developed for use in conjunction with Fire-1, an experimental fire chamber test facility at the Naval Research Laboratory in Washington, D.C. It is a three-dimensional finite difference model which includes the phenomena of conduction, turbulence, global pressure correction, surface radiation and strong buoyancy flows. Given specific data on heat release, it predicts velocities, temperatures, pressures, densities and viscosities throughout its geometry. It has been reasonably validated by comparison with experiments in Fire-1. Advanced graphics techniques, such as color contouring and three-dimensional vector field plotting, have been applied to make output data more informative. This model, if easily modified to more specific geometries, may become a useful tool for naval architects in the design of the fire safe ship.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT XX UNCLASSIFIED/UNLIMITED __ SAME AS RPT __ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL M.D. Kelleher	22b. TELEPHONE (Include Area Code) (408) 646-2530 22c. OFFICE SYMBOL ME/Kk

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE Unclassified

S/N 0102-LF-014-6603

Approved for public release: Distribution is unlimited

Numerical Field Model Simulation of Full-Scale Fire Tests in a Closed Spherical/Cylindrical Vessel Using Advanced Computer Graphics Techniques

by

Timothy G. McCarthy
Lieutenant, United States Navy
B.S.M.E., University of Rochester, Rochester, N.Y., 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

SEPTEMBER 1991

ABSTRACT

Personnel and equipment casualties, caused by shipboard fires have adversely affected overall readiness of the U.S. Navy for centuries. Understanding the phenomena of fire in enclosed spaces, such as those found on surface ships and submarines, will greatly enhance the Navy's ability to combat or prevent them. This computer model was developed for use in conjunction with Fire-1, an experimental fire chamber test facility at the Naval Research Laboratory in Washington, D.C. It is a three-dimensional finite difference model which includes the phenomena of conduction, turbulence, global pressure correction, surface radiation and strong buoyancy flows. Given specific data on heat release, it predicts velocities, temperatures, pressures, densities and viscosities throughout its geometry. It has been reasonably validated by comparison with experiments in Fire-1. Advanced graphics techniques, such as color contouring and three-dimensional vector field plotting, have been applied to make output data more informative. This model, if easily modified to more specific geometries, may become a useful tool for naval architects in the design of the fire safe ship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

A	Area
A	Finite Difference Coefficients
ARU_	Source Term Variable
AU_	Source Term Variable
C_	Coefficients for Control Volume
C_M	Coefficients for Control Volume
C_P	Coefficients for Control Volume
COND_1	Coefficients for Control Volume
C_{pm}	Mean Isobaric Heat Capacity
CURV	Curvature Terms
CURVN	Orthogonal Curvature Terms
F_{Ai-Aj}	View Factor for Radiation Emitted by Surface i and Incident Upon Surface j
G	Gravitational Acceleration
G	Mass Flux Rate
G	Term Used in Radiation Model
g	Curvilinear Base Vector
g_i	Scaling Term
g_{ij}	Covariant Metric Tensor
g^{ij}	Contravariant Metric Tensor
H	Mixing Length Parameter
h	Scale Factor
h	Convective Heat Transfer Coefficient
$\mathcal{H}$	Enthalpy
J	Total Heat Flux
K	Adjustable Constant
k	Thermal Conductivity
M	Momentum Flux
m	Rate of Change
n	Normal Direction Toward the Vessel Center
P	Pressure
Pr	Prandtl Number
Pr_t	Turbulent Prandtl Number

q	Heat Flux
q_r	Thermal Radiation Energy
R	Universal Gas Constant
R_	Source Term Variable
RR_	Source Term Variable
Ri	Richardson Number
r	Distance Between Two Surfaces
S_f	Source Term
S_{hs}	Heat Source
S_{mp}	Mass Source Term
T	Temperature
t	Time
u	Velocity
V	Volume
VIS	Local Viscosity
X	Length in X-Direction (In QUICK Scheme)

GREEK LETTERS

β	Angles Formed by Radiation Surface Normals
χ	Term Used in Radiation Model
δ_{ij}	Kronecker Delta
ε	Emissivity
ϕ	Dissipation Function
μ	Dynamic Viscosity
θ	Directions, θ, r, and ϕ or Z
ρ	Fluid Density
σ	Stress
σ	Stefan-Boltzmann Constant
ψ	Term Used in Radiation Model

SUBSCRIPTS

B	Control Volume to the Back
b	Back Control Volume Face
E	Control Volume to the East
EQ	Equilibrium

e	East Control Volume Face
eff	Effective
F	Control Volume to the Front
f	Front Control Volume Face
g	Global
N	Control Volume to the North
n	North Control Volume Face
o	Reference
p	Present Cell
R	Reference
S	Control Volume to the South
s	South Control Volume Face
s	Vessel Wall
W	Control Volume to the West
w	West Control Volume Face
,i	derivative with respect to i
,t	derivative with respect to time

SUPERSCRIPTS

n	Future Value
n-1	Present Value
*	Estimated Value
*	Ventilation Values
′	Correction
^	Prior Value

I. INTRODUCTION

A. BACKGROUND

Annually, the effects of fires on Naval forces are particularly devastating. Ships may be removed from service for repairs which incur costs that may run into the tens of millions of dollars. Personnel casualties, ship down time, equipment repair and replacement all result in a loss of overall readiness of our fleets. The prevention of shipboard fires is of the utmost importance to today's Navy. The understanding of the phenomena of fire, especially in the enclosed spaces found aboard ship, is the first step toward its control and prevention.

The study of fire propagation requires the combined knowledge of fluid dynamics, mass and heat transfer, and combustion. Research into the mechanics of fire and prediction of its behavior will aid engineers in reducing the probability of its ignition and propagation.

There are a number of ways to conduct this research. The most obvious is experimental. But, fires aboard ships are very complex. Often they are in enclosed airtight spaces which allow pressures to build. These spaces may be full of electronic equipment, flammables or toxic substances. Their accessibility may be extremely limited, hampering efforts to combat fires. An experiment that can accurately account for all these complexities becomes very expensive.

At the Naval Research Laboratory in Washington, D.C., the Navy has built Fire-1, a large pressure vessel designed to model fires aboard submarines, or closed compartments and tanks found on surface ships. It allows fires to be studied under the unique conditions experienced in shipboard fires.

Another method for conducting fire research is the use of a computer model. As computers get faster and can allow for large amounts of data storage, researchers are able to thoroughly model fire phenomena and predict future behavior without the continuous expensive full scale testing of Fire-1. Fires may be modeled by the numerical solution of the governing equations. These models are then verified by the existing data from experiments. With an accurate computer model, several options are available. More complex geometries may be incorporated for specific areas of interest. Entire models of ships may someday be developed to show areas of susceptibility in design. Effects of firefighting methods may be accurately predicted. The savings in running computer codes versus full scale testing are considerable.

Also, now that a high speed VAXSTATION 3100 SPX/RJ19 Model 38 workstation may be dedicated to this particular simulation, computing costs may be minimized. The current code requires approximately 1.0 hours of VAXSTATION CPU per second of fire time.

B. COMPUTER MODELING

Field modeling uses difference forms of the conservation equations of mass, momentum, energy and species. These are used to calculate temperature, velocity, pressure, viscosity and density at specific points in the volume of interest. This volume, being the compartment studied, is broken down to finite volume elements. The conservation equations are solved at this level for discrete time steps from a known initial condition. Additional models of physical effects such as radiation, turbulence, and wall conduction are included to increase the simulation's validity. This method requires large amounts of computer memory and high speed processors.

Much research has been done previously and has provided the basis for this thesis. At the University of Notre Dame [Refs. 1 and 2] work has been conducted involving aircraft cabin fires using a two dimensional finite difference field model which predicts velocity, temperature and smoke concentration inside the passenger area of an aircraft. Nicolette *et al.* [Ref. 3] developed a two dimensional model of transient cooling by natural convection. It utilized a fully transient, semi-implicit upwind differencing scheme and global pressure correction that was verified experimentally.

More recent [Refs. 4 through 12] studies have developed numerical solutions for three dimensional rectangular enclosures in which non-linear partial differential equations were solved by finite difference methods. Models for three dimensional cylindrical coordinate buoyant flows [Refs. 13 through 19] have also been developed, and deal mainly with horizontal annuli with differential temperatures specified at inner and outer cylindrical walls. Smutek *et al.* [Ref. 18] studied buoyant flows in horizonal cylinders with differentially heated ends at low Rayleigh numbers ($74 \leq Ra \leq 18700$). Yang *et al.* [Ref. 19] conducted a similar study but with high Rayleigh numbers ($10^4 \leq Ra\ 10^7$).

Studies have also been done on methods for decoupling the pressure terms from the Navier-Stokes Equation. The stream function-vorticity formulation has been used [Refs. 13 through 18] to calculate natural convection in various geometries. There are problems with this method such as instability at high Rayleigh numbers. Yang *et al.* [Ref. 19] address this problem and suggest using a primitive variable formulation when using arbitrary orthogonal coordinates.

Natural convection in spherical annuli was studied by Ozoe [Ref. 20] utilizing velocity-vector formulation. Field models involving prediction of fires in enclosures have been studied by Baum and Rehm [Refs. 21 through 24]. These include time dependent Boussinesq equations to simulate three dimensional buoyant convection and smoke aerosol coagulation. Field models involving three dimensional enclosures and employing the Boussingesq approximation, were studied by Bagnaro *et al.* [Ref. 25] and by Markatos and Pericleous [Ref. 26].

In this thesis, the numerical method developed by Yang *et al.* [Ref. 19] using primitive variable finite difference discretization in generalized orthogonal coordinates is employed. This method can handle complex geometries and has the numerical stability characteristic of primitive variable formulation.

C. FIRE-1 TEST FACILITY

An experimental test facility called Fire-1, has been constructed at the Naval Research Laboratory to study the behavior of fires in enclosed spaces found on submarines and surface ships. Since the computer code presented in this thesis models the geometry of Fire 1, this section contains a brief description of that facility. More information may be obtained from Alexander *et al.* [Ref. 27]. Figure 1.1 shows the basic layout. Fire-1 is a cylindrical pressure vessel with hemispherical endcaps. It is constructed of 3/8-inch ASTM 295 Grade C steel and can withstand internal pressures up to 89.7 psi and temperatures of 450°F. Its total length is 46.6 feet long. The cylinder and endcap radii are both 9.6 feet. Rupture discs are placed at each endcap to prevent failure due to overpressurization.

Figure 1.2 shows the instrumentation layout. An array of chromel-alumel thermocouples with ceramic insulation and stainless steel jackets, are placed near each endcap. Additional thermocouples are placed on the chamber walls, both

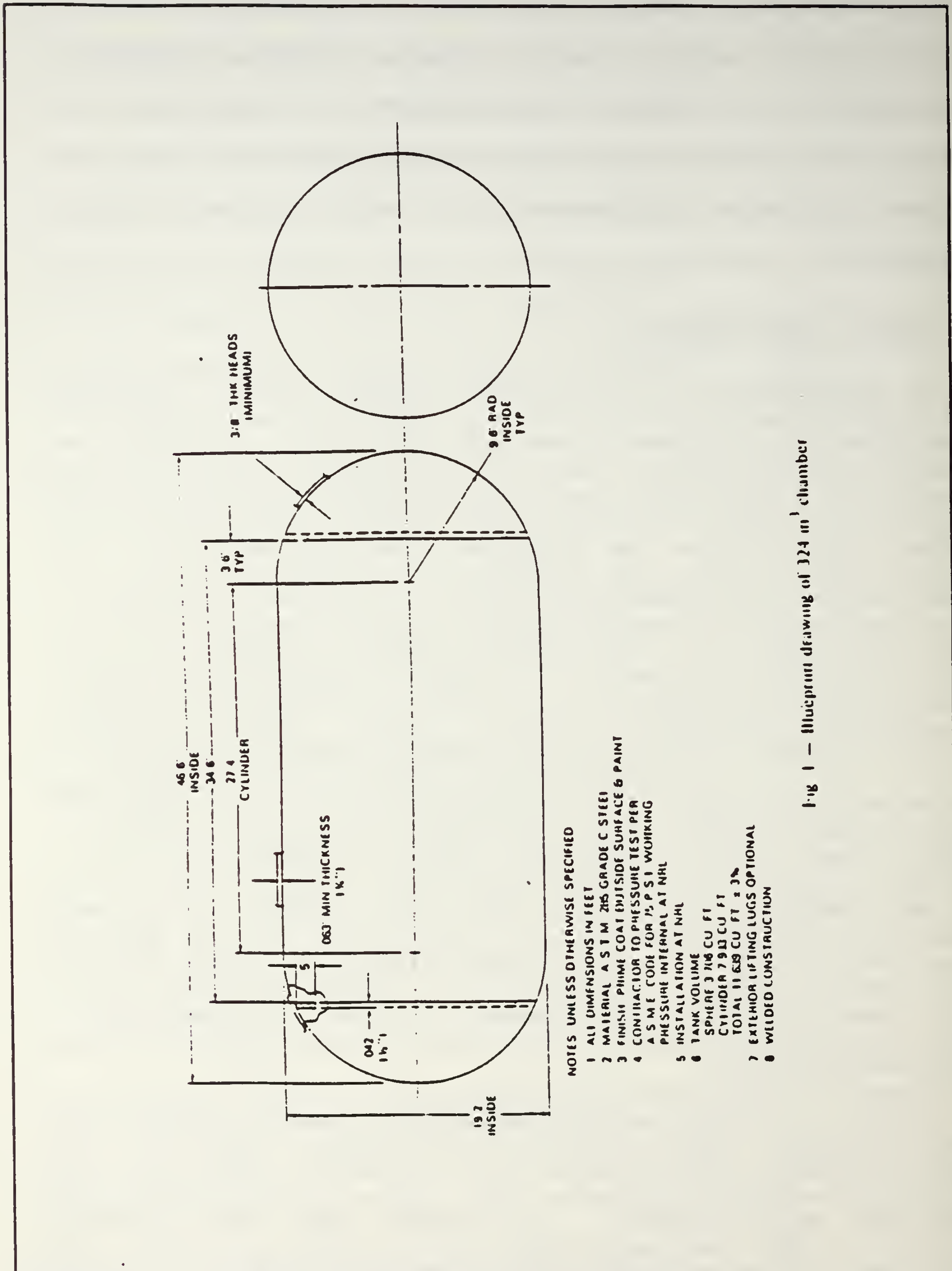

Figure 1.1 Drawing of Fire-1 Test Vessel.

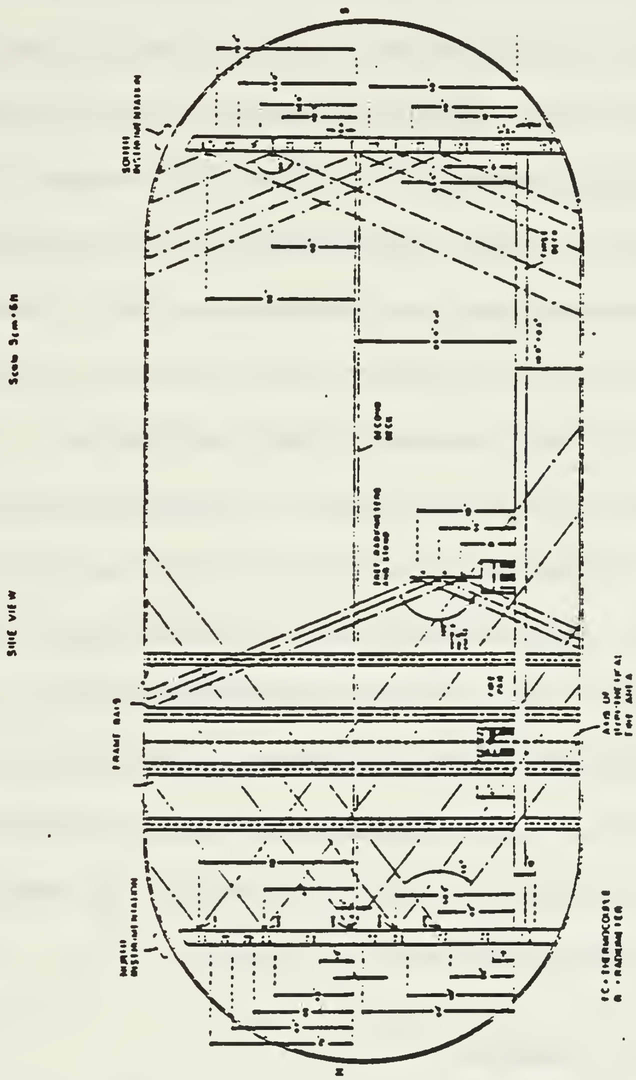

Figure 1.2 Sensor Locations of Fire-1 Test Vessel.

inside and out, to monitor inside and outside wall temperatures. A specific test might call for placement of extra thermocouples or radiometers at various other locations. These are arranged as required by the experiment.

Burn rate data is obtained using round, tapered edge fire pans of various cross sectional areas, and a constant level, liquid fuel supply system. To date, this data has been the least accurate in the experiment. The system and its calibration are described by Alexander *et al.* [Ref. 27]. Smoke concentration can be measured using video cameras, particle analysis and obscuration with laser detectors.

To more completely represent shipboard compartments, the facility has a number of features. First is the installation of two removable decks, one at the midheight, the other at three feet above the bottom. Either grated or solid deck plating is used depending on desired configuration. Second is the installation of a nitrogen pressurization system used as an extinguishing agent. Its performance is being tested for possible use combatting actual fires.

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 alternative to full scale test using Fire-1. With proper modifications, used in conjunction with

Fire-1, it will test effectiveness of damage-control systems and evaluate new ship designs.

In the work by Nies [Ref. 28], the code was based on a rectangular geometry with the volume identical to Fire-1. This was a three dimensional, finite volume model using primitive variables. Turbulence, wall conduction, and a global pressure correction factor were also included. Due to the unreliability of burn rate data, Nies [Ref. 28] devised a scheme for computing a heat release rate by using experimental pressure curves as input.

The actual geometry of Fire-1 was employed by Raycraft [Ref. 29]. Using its spherical/cylindrical coordinate system and detailed formulation of radiation surface view factors, global pressure correction, conduction and turbulence, the code created an extremely viable model for use with Fire-1. There were the continued problems with simulating the heat release data which were partially resolved by numerically fitting experimental burn rate data available.

Houck [Ref. 31] included a model which simulated internal forced circulation. It was compared to data run without circulation and it was concluded that circulation had minimal effects on the overall velocity and temperature profiles.

In this thesis, advanced three dimensional and color graphics techniques are used to present data generated using the previously developed codes. Using the VAXSTATION 3100 SPX and the software CA-DISSPLA [Ref. 31] the data is presented in a more informative fashion. Color graphics are used to present isotherm profiles and three dimensional vector fields will represent velocity profiles.

II. DESCRIPTION OF NUMERICAL MODEL

A. GOVERNING EQUATIONS

The model is based on the system of conservation equations which govern the behavior of fluid flow and heat transfer in gases. These equations are in differential form and are presented in generalized curvilinear coordinates using standard tensor notation. Nies [Ref. 28] based his model on rectangular geometry using Cartesian coordinates. Raycraft [Ref. 29] refined the model to describe the exact geometry of Fire-1 and included surface radiation. Houck [Ref. 30] described the transformation to curvilinear coordinates, used by Yang *et al.* [Ref. 19], in detail and the following forms of the governing equations are obtained.

The equation of continuity is:

$$\rho_t + \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta^i} \{ \sqrt{g} \rho \frac{u^i}{h_i} \} = 0 \tag{2.1}$$

The energy equation becomes:

$$\begin{aligned} (\rho C_{pm} T)_t + \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta^i} \{ \sqrt{g} \rho C_{pm} u^i \frac{T}{h_i} \} \\ = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta^i} \{ \sqrt{g} \frac{k_{eff} T_{,i}}{h_i^2} \} + S_f \end{aligned} \tag{2.2}$$

where the source term, S_f is:

$$S_f = \mu\Phi + P\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta^i}\{\sqrt{g}\frac{u^i}{h_i}\} + S_{hs} \tag{2.3}$$

and the dissipation term is:

$$\begin{aligned}\Phi = 2\{\left(\frac{u^i}{h_i}\right)_{,j}^2\}\delta_{ij} \\ + \{\left(\frac{u^i}{h_i}\right)_{,j}^2(1-\delta_{ij})\}^2 - \frac{2}{3}\{\left(\frac{u^i}{h_i}\right)_{,i}\}^2\end{aligned} \tag{2.4}$$

S_{hs} is the heat source term which is zero everywhere except nodes at the fire's location and δ_{ij} is the Kronecker Delta.

The momentum equation becomes:

$$\begin{aligned}(\rho u^i)_t + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta^i}\{\sqrt{g}\frac{u^i u^j}{h_j}\} \\ = -\frac{P_{,i}}{h_i} + \rho G^i + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta^j}\{\frac{\sqrt{g}\sigma_i^j}{h_j}\} \\ - \frac{1}{h_i h_j}\frac{\partial h_i}{\partial\theta^j}(\rho u^i u^j - \sigma_i^j) + \frac{1}{h_i h_j}\frac{\partial h_j}{\partial\theta^i}(\rho u^j u^i - \sigma_i^j)\end{aligned} \tag{2.5}$$

where the stress tensor is:

$$\begin{aligned}\sigma_i^j = \mu_{eff}\{\frac{h_j}{h_i}\frac{\partial}{\partial\theta^i}\left(\frac{u^j}{h_j}\right) \\ + \frac{h_i}{h_j}\frac{\partial}{\partial\theta^j}\left(\frac{u^i}{h_i}\right) + \frac{\delta_{ij}}{h_i h_j}\frac{\partial q_{ii}}{\partial\theta^m}\left(\sqrt{g}\frac{u^m}{h_m}\right)\}\end{aligned} \tag{2.6}$$

Effective conductivity k_{eff} and dynamic viscosity μ_{eff} include both laminar and turbulent terms. Additional terms found in the momentum equation are due to coriolis and centrifugal effects.

The equations of state remain unchanged through coordinate transformations and are given as:

$$P = \rho R T \tag{2.7}$$

$$\mathcal{H} = C_{pm} \left(T - T_R \right) \tag{2.8}$$

B. INITIAL AND BOUNDARY CONDITIONS

In order to solve this system of differential equations, boundary and initial conditions must be determined and applied.

1. Initial Conditions

The initial conditions for the model are determined from conditions present just prior to ignition in Fire-1. The air inside is totally at rest. The temperature is equal to ambient temperature and is assumed uniform throughout. Therefore, in the model, the entire velocity field is set to zero and the non-dimensional temperature field is set to 1.0 which corresponds to ambient temperature. Pressure and density distributions are at static equilibrium.

2. Boundary Conditions

Since the vessel wall is a solid boundary which is nonporous, the velocities, both normal and tangential to the wall, are zero. Mass flux across the wall is also zero. The temperature of the wall is equal to the temperature of the fluid at the interface. Conservation of energy must also be met at the interface. The following three equations summarize wall boundary conditions:

$$u^{i}_{surf} = 0 \tag{2.9}$$

$$T_{fluid} = T_{solid} \tag{2.10}$$

$$q_r - k_f \frac{\partial T}{\partial n} \Big|_f = -k_s \frac{\partial T}{\partial n} \Big|_{solid} \tag{2.11}$$

where q_r is the heat flux arriving at the solid/fluid interface and n is the normal direction of the surface into the enclosure. There is conduction through the wall and convection from outer surface to ambient temperature.

Due to singularities occurring at r=0 in cylindrical/ spherical coordinates, special care must be taken at the origin. Yang *et al.* [Ref. 19:pp. 167-168] discuss methods for addressing this problem. In this model, two consecutive control volumes are placed at r=0 and continuity is applied.

C. MODELS OF PHYSICAL PHENOMENA

1. Wall Conduction Model

This model calculates heat loss from the vessel through the walls to the environment. It assumes one dimension, unsteady heat flow and constant convective heat transfer coefficient at the wall's exterior. The energy equation is:

$$(\rho_s C_{ps} T)_t = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta^i} (\sqrt{g} k_s T_{,j} g^{ij}) + S \qquad (2.12)$$

2. Turbulence Model

The turbulence model is a simple algebraic method used to predict mean flow quantities for incompressible boundary layer flows. Developed by Nee and Liu [Ref. 33], the model determines the effective viscosity in recirculating buoyant flows with large variations in turbulence levels. The equation, transformed into generalized curvilinear coordinates, is:

$$\frac{\mu_{eff}}{\mu_o} = 1 + \frac{\left(\frac{\ell}{H} \right)^2 \sqrt{\left(\frac{1}{h_j} \frac{\partial u^i}{\partial \theta^j} \right) \left(1 - \delta_i^j \right)}}{2 + \frac{Ri}{Pr_t}} \qquad (2.13)$$

where ℓ/H is a non-dimensional mixing length parameter given as:

$$\frac{\ell}{H} = K \left\{ \frac{\sqrt{u^i u^i}}{\sqrt{\sum_{ij} \left(\frac{1}{h_j} \frac{\partial u^i}{\partial \theta^j} \right)^2}} + \frac{\sqrt{\sum_{ij} \left(\frac{1}{h_j} \frac{\partial u^i}{\partial \theta^j} \right)^2}}{\sqrt{\sum_{ij} \left(\frac{1}{h_i h_j} \frac{\partial^2 u^i}{\partial \theta^i \partial \theta^j} \right)^2}} \right\} \tag{2.14}$$

K is an adjustable constant and the Richardson Number, Ri, is given as:

$$Ri = \frac{H}{u_i^2} \frac{\left(\frac{\partial T}{\partial n} \right) \vec{n} \cdot \vec{g}}{\left[\left(\frac{\partial u^1}{\partial n} \right) \vec{n} \cdot \vec{g} \right]^2 + \left[\left(\frac{\partial u^2}{\partial n} \right) \vec{n} \cdot \vec{g} \right]^2 + \left[\left(\frac{\partial u^3}{\partial n} \right) \vec{n} \cdot \vec{g} \right]^2} \tag{2.15}$$

$\vec{n}$ is a unit vector in the opposite direction of gravity.

Pr_t is the turbulent Prandtl number which is also used to compute the effective conductivity.

$$k_{eff} = \frac{1}{Pr} + \frac{1}{Pr_t} \frac{\mu_{eff}}{\mu_o} \tag{2.16}$$

Pr is the molecular Prandtl number.

3. Surface Radiation Model

Raycraft [Ref. 29, pp. 24-44] describes this model in detail. Summarizing, the radiation model considers only surface radiation. Smoke and gases are considered transparent. Inside the model, walls and flame areas are treated as surfaces. Each surface is considered to be gray and diffuse. Sparrow and Cess [Ref. 34] discuss the net radiosity method upon which this model is based.

Net rate of heat loss per unit area is given as:

$$\frac{Q_i}{A_i} = \sum_{j=1}^{N} G_{ij} \sigma T_j^4 \qquad \textbf{(2.17)}$$

where

$$G_{ij} = \frac{\varepsilon_i}{1 - \varepsilon_i} (\delta_{ij} - \psi_{ij}) \qquad \textbf{(2.18)}$$

$$\psi_{ij} = \chi_{ij}^{-1}$$

$$\chi_{ij} = \frac{\delta_{ij} - (1 - \varepsilon_i) F_{Ai-Aj}}{\varepsilon_i} \qquad \textbf{(2.19)}$$

F_{Ai-Aj} is the view factor of radiation emitted by surface i onto surface j. The general equation is given by

$$F_{Ai-Aj} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\beta_1 \cos\beta_j dA_i dA_j}{\pi r^2} \qquad \textbf{(2.20)}$$

III. FINITE VOLUME CALCULATIONS

A. INTRODUCTION

The numerical model's independent variables are time and three space coordinates. Dependent variables consist of the three dimensional components of velocity, temperature, pressure and density. These six unknowns require six equations for solution. They are the continuity equation (Eq. (2.1)), the three momentum equations (Eq. (2.5)), the energy equation (Eq. (2.2)), and the equations of state (Eq. (2.7) and (2.8)). Doria [Ref. 35] discretized these equations in a method similar to this particular model based on the generalized form presented by Patanker [Ref. 36]. Doria applied the conservation equations in integral form to each control volume creating a set of finite difference equations which would lead to a solution.

Each control volume, or cell, surrounds a nodal point where one value of each property is constant throughout. The center nodal point determines pressure density and temperature. The grid determining velocities are staggered by one-half a cell length. Patanker [Ref. 36:pp. 115-120] describes how this alleviates two problems: the pressure differential between the two adjacent nodes, which ultimately determines the velocity at the node in question, is based on a length which is half as long as in the unstaggered cell (this reduces error by one half); second, stability is gained by this stagger which precludes unrealistic, wavy oscillatory velocity fields, since the difference of adjacent velocities are used to satisfy continuity.

Since primitive variables are used versus the stream function, the pressure term coupling between equations must be handled specially. An iterative procedure estimates pressure and then pressure is corrected to ensure continuity is satisfied for each cell. A local pressure correction is discussed by both Patanker [Ref. 36:pp. 120-128] and Doria [Ref. 35:pp. 26-32]. A global pressure correction is included in the model to handle net energy changes and is described by Nicolette, *et al.* [Ref. 3].

The finite difference equations are solved iteratively. Non-linear problems like fluid flow are difficult to force convergence to final solution. Many schemes have been developed to obtain the flow problem solution. Each method has its problems and instabilities. This model employs the Quadratic Upstream Interpolation for Convective Kinematics, or QUICK, developed by Leonard [Ref. 37]. QUICK estimates values and gradients of transport variables at the faces of the cells. It has the accuracy of central finite difference schemes and the stability of convective diffusion terms found in upwind differencing. Yang [Ref. 12] applied the QUICK scheme to coupled momentum energy and pressure equation solutions for three-dimensional flow in tilted rectangular enclosures.

In this chapter, the governing equations will be applied to the specialized control volumes of the model. They will be put in integral form and discretized according to the QUICK scheme. Pressure correction from iteration will also be applied.

B. CONTROL VOLUME ANALYSIS

At the center of each elemental control volume, or cell, lies the grid point of interest. At this point, the model determines the unknown values of the dependent variables. Denoting this grid point as P (i, j, k) we define its neighbors as: 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), and Back (i, j, k-1). The boundaries around P are designated by lower case letters e, w, n, s f, and b. Typical spherical and cylindrical cells are shown in Figures 3.1 and 3.2 respectively.

Figure 3.3 shows the basic two dimensional cell used to determine pressure, density and temperature. In contrast, Figure 3.4 shows the staggered grid used to determine velocities. The velocity u_i^1 is located on the west face; u_i^2 is located on the south face and u_i^3 is located on the back face (not shown). The superscripts on the velocities designate

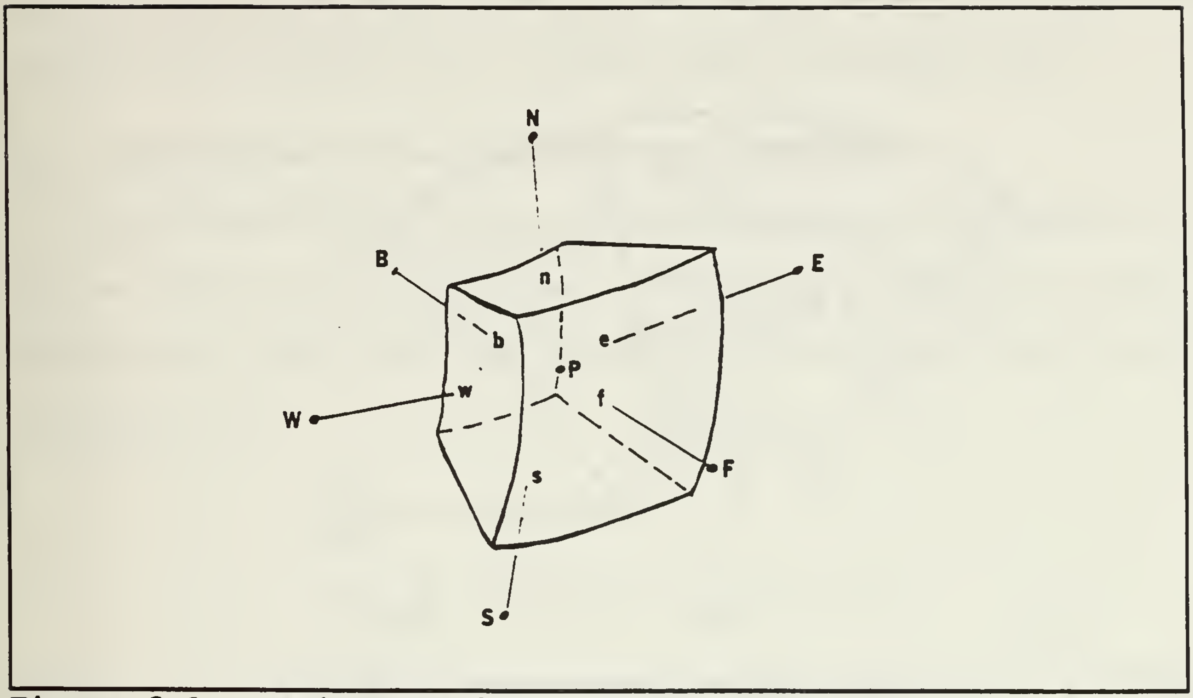

Figure 3.1 Basic Spherical Cell.

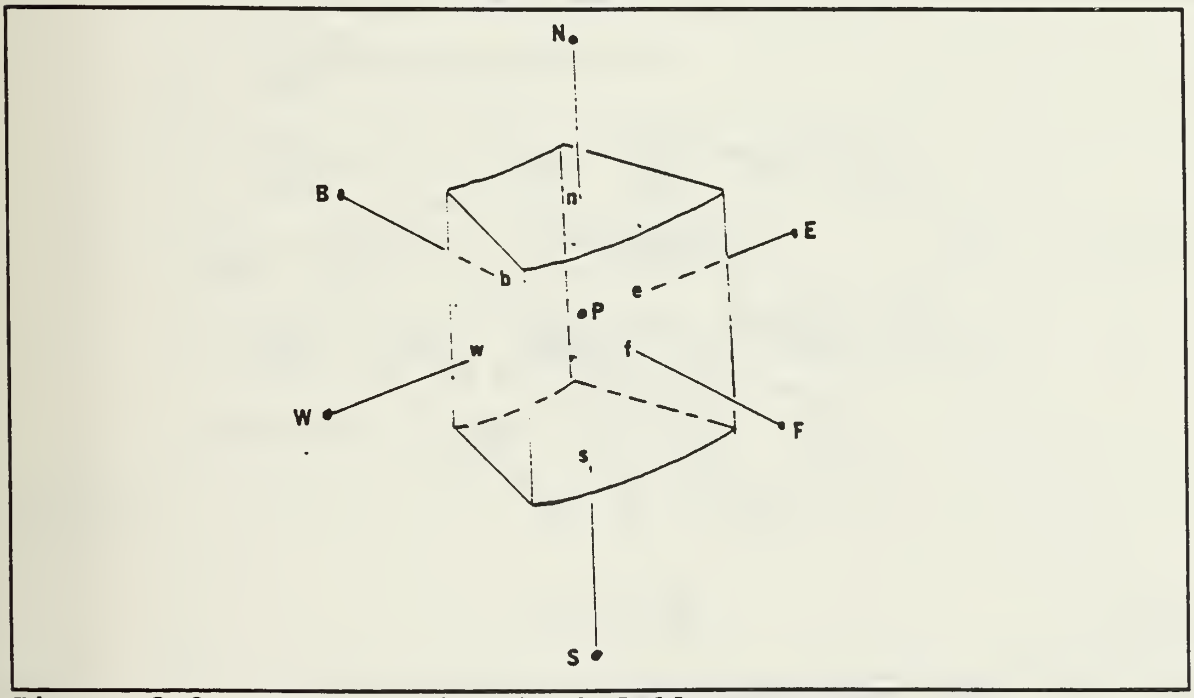

Figure 3.2 Basic Cylindrical Cell.

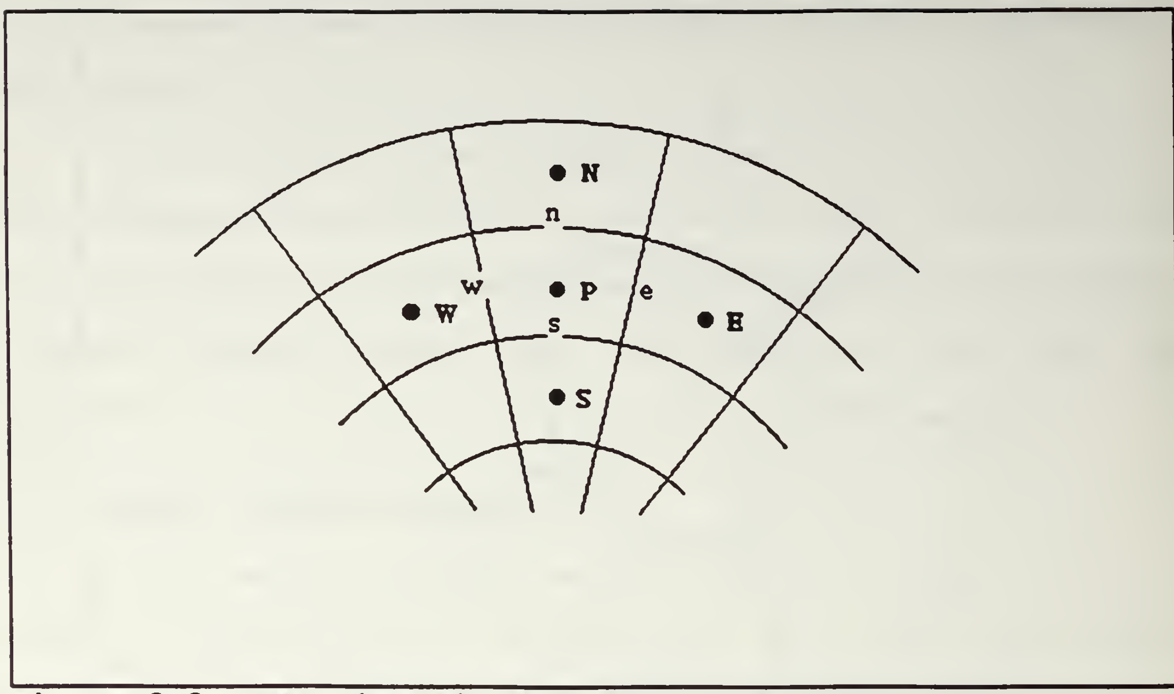

Figure 3.3 Two Dimensional Cell.

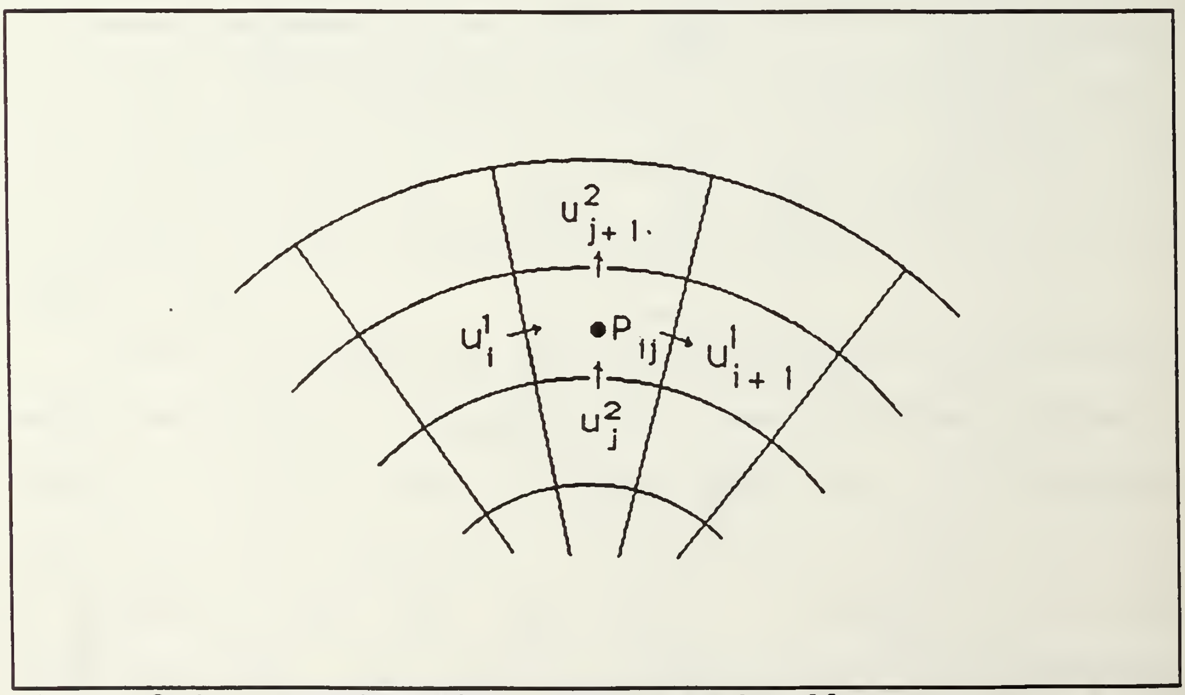

Figure 3.4 Two Dimensional Staggered Cell.

coordinate direction. These velocities are staggered in location by one-half cell length from the primary cell.

C. INTEGRATION OF THE CONSERVATION EQUATIONS

The conservation equations are integrated over each cell volume. From this point, they can be discretized into finite difference equations. The integral form of the continuity equation is:

$$\begin{aligned} &\int \frac{\partial \rho}{\partial t} h_1 h_2 h_3 \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \\ &+ \int \Big[\frac{\partial}{\partial\theta^1} (\rho u^1 h_2 h_3) + \frac{\partial}{\partial\theta^2} (\rho u^2 h_3 h_1) \\ &+ \frac{\partial}{\partial\theta^3} (\rho u^3 h_1 h_2) \Big] \partial\theta^1 \partial\theta^2 \partial\theta^3 \\ &= 0 \end{aligned} \tag{3.1}$$

The energy equation becomes:

$$\begin{aligned} &\int \frac{\partial (\rho C_{pm} T)}{\partial t} h_1 h_2 h_3 \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \\ &+ \int \Big[\frac{\partial}{\partial\theta^1} (\rho C_{pm} u^1 T h_2 h_3) + \frac{\partial}{\partial\theta^2} (\rho C_{pm} u^2 T h_1 h_3) \\ &+ \frac{\partial}{\partial\theta^3} (\rho C_{pm} u^3 T h_1 h_2) \Big] \partial\theta^1 \partial\theta^2 \partial\theta^3 \\ &- \int \Big[\frac{\partial}{\partial\theta^1} (q^1 h_2 h_3) + \frac{\partial}{\partial\theta^2} (q^2 h_1 h_3) + \frac{\partial}{\partial\theta^3} (q^3 h_1 h_2) \Big] \\ &\cdot \partial\theta^1 \partial\theta^2 \partial\theta^3 + \int S h_1 h_2 h_3 \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \end{aligned} \tag{3.2}$$

where:

$$q^i = \frac{-k}{h_i} \frac{\partial T}{\partial \theta^i} \tag{3.3}$$

The momentum equations become:

$$
\begin{aligned}
&\int \frac{\partial}{\partial t} (\rho u^{i}) \, h_1 h_2 h_3 \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \\
&+ \int \frac{\partial}{\partial \theta^{j}} [(\frac{h_1 h_2 h_3}{h_j}) \rho u^{i} u^{j}] \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \\
&= \int \frac{-\partial}{\partial \theta^{i}} (P \frac{h_1 h_2 h_3}{h_i} \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \\
&+ \int \rho G_i h_1 h_2 h_3 \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \qquad (3.4) \\
&+ \int \frac{\partial}{\partial \theta^{j}} (\sigma^{ij} \frac{h_1 h_2 h_3}{h_i h_j}) \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \\
&- \int \frac{h_1 h_2 h_3}{h_i h_j} [\frac{\partial h_i}{\partial \theta^{j}} (\rho u^{j} u^{i} - \sigma^{ij})] \, \partial\theta^1 \partial\theta^2 \partial\theta^3 \\
&+ \int \frac{h_1 h_2 h_3}{h_j h_i} [\frac{\partial h_j}{\partial \theta^{i}} (\rho u^{j} u^{j} - \sigma^{jj})] \, \partial\theta^1 \partial\theta^2 \partial\theta^3
\end{aligned}
$$

D. DISCRETIZATION OF THE CONTINUITY EQUATION

To provide maximum stability and accuracy for the model, three finite differencing schemes are utilized. Forward differencing is used for time dependence, central differencing is used for diffusion terms and the QUICK algorithm is used for the convective terms.

In forward differencing the future value of the time dependent variable is predicted from its previous value plus an additional term derived from the previously known slope m multiplied by the time step Δt. For example the new value for density ρ^n is calculated using the old value ρ^{n-1} plus the extra term:

$$\rho^n = \rho^{n-1} + m\Delta t \tag{3.5}$$

The integrand in the continuity equation (3.1) becomes:

$$\frac{\partial\rho}{\partial t}\,dV = \frac{\rho^n - \rho^{n-1}}{\Delta t}\,h_1 h_2 h_3 \Delta\theta^1 \Delta\theta^2 \Delta\theta^3 = \frac{\rho^n - \rho^{n-1}}{\Delta t}\,\Delta V \tag{3.6}$$

Evaluating the integral, Equation (3.1) becomes:

$$\begin{aligned}&(\rho^n - \rho^{n-1})\,\frac{\Delta V}{\Delta t} + [\,\rho u^1 h_2 h_3 d\theta^2 d\theta^3\,]_e - [\,\rho u^1 h_2 h_3 d\theta^2 d\theta^3\,]_w \\ &+ [\,\rho u^2 h_1 h_3 d\theta^1 d\theta^3\,]_n - [\,\rho u^2 h_1 h_3 d\theta^1 d\theta^3\,]_s \\ &+ [\,\rho u^3 h_1 h_2 d\theta^1 d\theta^2\,]_f - [\,\rho u^3 h_1 h_2 d\theta^1 d\theta^2\,]_b = 0\end{aligned} \tag{3.7}$$

The mass flux, G, must be calculated at each face:

$$G_e = (\rho u^1)_e = u_e^1 \left[\frac{\rho_P\,(h_1\Delta\theta^1)_{i+1} + \rho_E\,(h_1\Delta\theta^1)_i}{(h_1\Delta\theta^1)_{i+1} + (h_1\Delta\theta^1)_i}\right] \tag{3.8}$$

$$G_w = (\rho u^1)_w = u_w^1 \left[\frac{\rho_P\,(h_1\Delta\theta^1)_{i-1} + \rho_W\,(h_1\Delta\theta^1)_i}{(h_1\Delta\theta^1)_{i-1} + (h_1\Delta\theta^1)_i}\right] \tag{3.9}$$

$$G_n = (\rho u^2)_n = u_n^2 \left[\frac{\rho_P\,(h_2\Delta\theta^2)_{j+1} + \rho_N\,(h_2\Delta\theta^2)_j}{(h_2\Delta\theta^2)_{j+1} + (h_2\Delta\theta^2)_j}\right] \tag{3.10}$$

$$G_s = (\rho u^2)_s = u_s^2 \left[\frac{\rho_P\,(h_2\Delta\theta^2)_{j-1} + \rho_S\,(h_2\Delta\theta^2)_j}{(h_2\Delta\theta^2)_{j-1} + (h_2\Delta\theta^2)_j}\right] \tag{3.11}$$

$$G_f = (\rho u^3)_f = u_f^3 \left[\frac{\rho_P (h_3\Delta\theta^3)_{k+1} + \rho_F (h_3\Delta\theta^3)_k}{(h_3\Delta\theta^3)_{k+1} + (h_3\Delta\theta^3)_k} \right] \quad (3.12)$$

$$G_b = (\rho u^3)_b = u_b^3 \left[\frac{\rho_P (h_3\Delta\theta^3)_{k-1} + \rho_B (h_3\Delta\theta^3)_k}{(h_3\Delta\theta^3)_{k-1} + (h_3\Delta\theta^3)_k} \right] \quad (3.13)$$

The areas of the faces of the cell are given as:

$$A_{e,w} = (h_2\Delta\theta^2 \, h_3\Delta\theta^3)_{e,w} \quad (3.14)$$

$$A_{n,s} = (h_1\Delta\theta^1 \, h_3\Delta\theta^3)_{n,s} \quad (3.15)$$

$$A_{f,b} = (h_1\Delta\theta^1 \, h_2\Delta\theta^2)_{f,b} \quad (3.16)$$

In final finite difference form the continuity equation becomes:

$$\frac{(\rho^n - \rho^{n-1})\,\Delta V}{\Delta t} + G_e - G_w + G_n - G_s + G_f - G_b = S_{mp} \quad (3.17)$$

S_{mp} is the mass source term. As this residual approaches zero, the solution approach the exact solution. Iterations occur until S_{mp} reaches a specific, extremely small, cut off value.

E. DISCRETIZATION OF THE ENERGY EQUATION

Integrating over the control volume, the energy equation becomes:

$$\begin{aligned}
&[\,(\rho C_{pm}T)^{n}-(\rho C_{pm}T)^{n-1}\,]\frac{\Delta V}{\Delta t}+G_e\,(C_{pm}T)_e A_e \\
&-G_w\,(C_{pm}T)_w A_w+G_n\,(C_{pm}T)_n A_n-G_s\,(C_{pm}T)_s A_s+ \\
&G_f\,(C_{pm}T)_f A_f-G_b\,(C_{pm}T)_b A_b \\
&=k_e A_e\left(\frac{\partial T}{h_1\partial\theta^1}\right)_e-k_w A_w\left(\frac{\partial T}{h_1\partial\theta^1}\right)_w \\
&+k_n A_n\left(\frac{\partial T}{h_2\partial\theta^2}\right)_n-k_s A_s\left(\frac{\partial T}{h_2\partial\theta^2}\right)_s \\
&+k_f A_f\left(\frac{\partial T}{h_3\partial\theta^3}\right)_f-k_b A_b\left(\frac{\partial T}{h_3\partial\theta^3}\right)_b+S_f\Delta V
\end{aligned} \tag{3.18}$$

where S_f is the source term including dissipation, radiation, pressure work and heat sources. The total heat flux, J, resulting from convection and conduction is:

$$J^1_{e,w}=\left[\,(\rho C_{pm}u^1 T)-k_{eff}\frac{\partial T}{h_1\partial\theta^1}\,\right]_{e,w} \tag{3.19}$$

$$J^2_{n,s}=\left[\,(\rho C_{pm}u^2 T)-k_{eff}\frac{\partial T}{h_2\partial\theta^2}\,\right]_{n,s} \tag{3.20}$$

$$J^3_{f,b}=\left[\,(\rho C_{pm}u^3 T)-k_{eff}\frac{\partial T}{h_3\partial\theta^3}\,\right]_{f,b} \tag{3.21}$$

The final finite difference form of the energy equation becomes:

$$
\begin{aligned}
&[(\rho C_{pm} T)^{n} - (\rho C_{pm} T)^{n-1}] \frac{\Delta V}{\Delta t} + J_e^1 A_e \\
&- J_w^1 A_w + J_n^2 A_n - J_s^2 A_s + J_f^3 A_f - J_b^3 A_b = S_f \Delta V
\end{aligned}
\qquad \textbf{(3.22)}
$$

The term $(\rho u^i C_{pm} T)$ in the flux equations give rise to difficulties since C_{pm}, ρ and T are evaluated at the nodal point instead of the surface of the cell. Thus, fluxes are determined from values of ρ, T, and C_{pm} at P and its neighbors.

The QUICK Scheme is used to determine accurate values of the dependent variables at the control volume surfaces with stable properties. QUICK couples the stability of upwind differencing with the accuracy of central differencing. It is achieved by using a parabolic polynomial interpolation to fit the control volume at three consecutive nodal points. Two nodes are located on either side of the surface and one is located upstream. Yang [Ref. 12:pp. 77-89] discusses QUICK for one, two and three dimensions. Houck [Ref. 30:pp. 37-50] and Raycraft [Ref. 29:pp. 63-74] used the QUICK scheme for the energy equations and that method is repeated here.

Figure 3.5 from Raycraft [Ref. 29:pp. 64] shows the one dimensional scheme for the quadratic interpolation of a non-uniform grid.

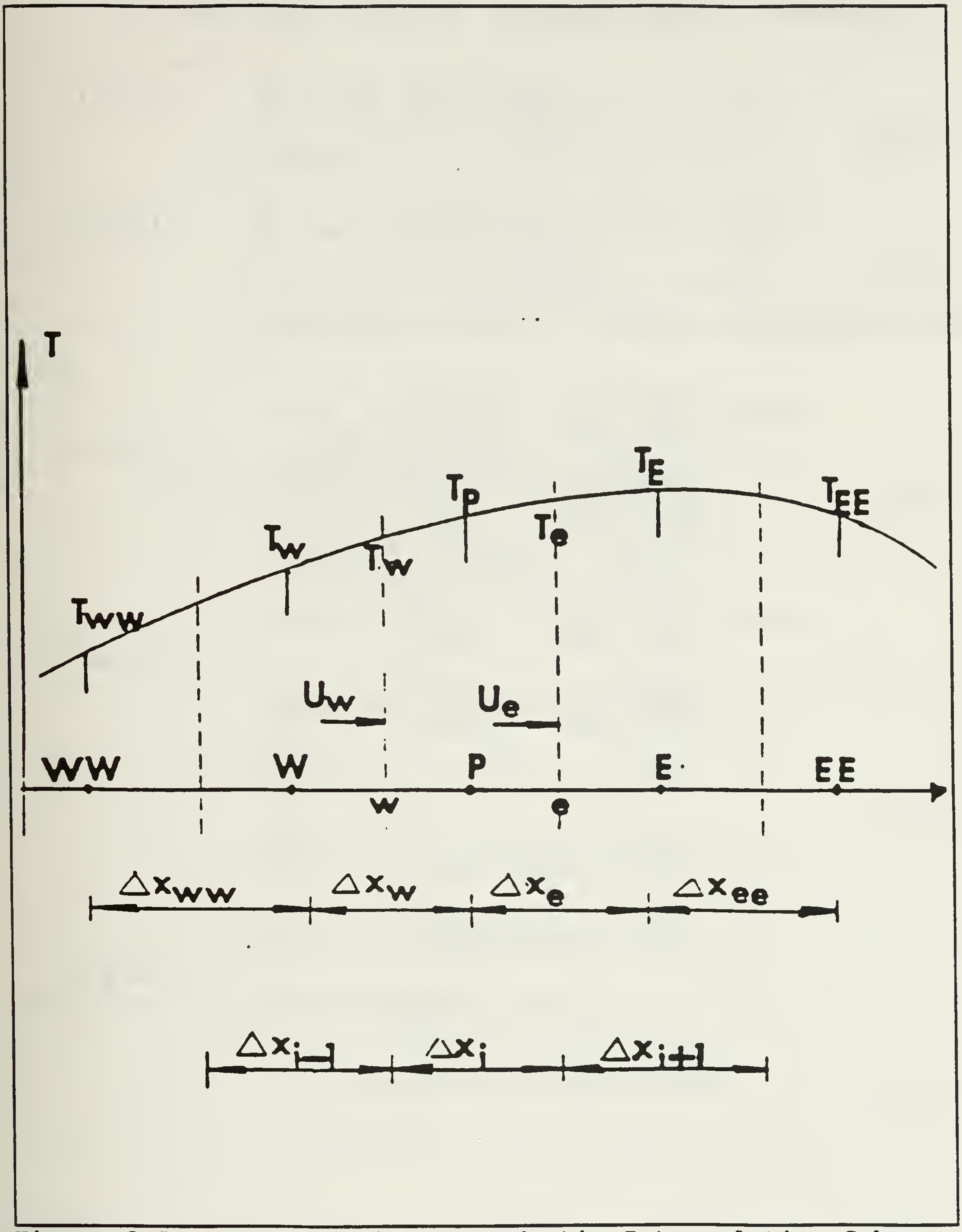

Figure 3.5 One Dimensional Quadratic Interpolation Scheme.

It is given by the equations

$$(\rho C_{pm} u T)_e = G_e C_{pm_e} \left[\left(\frac{T_p + T_E}{2} \right) - \frac{1}{8}\, curv_e \right] \qquad (3.23)$$

$$(\rho C_{pm} v T)_w = G_w C_{pm_w} \left[\left(\frac{T_p + T_w}{2} \right) - \frac{1}{8}\, curv_w \right] \qquad (3.24)$$

where the upstream weighted curvature terms are:

$$\begin{aligned} curv_e &= \frac{\Delta X_e^2}{\Delta X_i} \left[\frac{T_E - T_p}{\Delta X_e} - \frac{T_p - T_w}{\Delta X_w} \right] \quad if \quad G_e > 0 \\ &= \frac{\Delta X_e^2}{\Delta X_{i+1}} \left[\frac{T_{EE} - T_e}{\Delta X_{ee}} - \frac{T_E - T_p}{\Delta X_e} \right] \quad if \quad G_e < 0 \end{aligned} \qquad (3.25)$$

$$\begin{aligned} curv_w &= \frac{\Delta X_w^2}{\Delta X_{i+1}} \left[\frac{T_p - T_W}{\Delta X_w} - \frac{T_W - T_{WW}}{\Delta X_{ww}} \right] \quad if \quad G_w > 0 \\ &= \frac{\Delta X_w^2}{\Delta X_i} \left[\frac{T_E - T_P}{\Delta X_e} - \frac{T_p - T_W}{\Delta X_w} \right] \quad if \quad G_w < 0 \end{aligned} \qquad (3.26)$$

and

$$\begin{aligned} \Delta X_e &= \frac{1}{2} (\Delta X_i + \Delta X_{i+1}) \\ \Delta X_w &= \frac{1}{2} (\Delta X_i + \Delta X_{i-1}) \\ \Delta X_{ee} &= \frac{1}{2} (\Delta X_{i+1} + \Delta X_{i+2}) \\ \Delta X_{ww} &= \frac{1}{2} (\Delta X_{i-1} + \Delta X_{i-2}) \end{aligned} \qquad (3.27)$$

In generalized orthogonal coordinates the convective flux terms become:

$$(\rho C_{pm} u^1 T)_e = G_e C_{pm_e} \left(\frac{T_P + T_E}{2} - \frac{1}{8} curvn_e \right) \tag{3.28}$$

$$(\rho C_{pm} u^2 T)_w = G_w C_{pm_w} \left(\frac{T_P + T_w}{2} - \frac{1}{8} curvn_w \right) \tag{3.29}$$

where

$$\begin{aligned} curvn_e &= \frac{(h_1 \Delta\theta^1)_e^2}{(h_1 \Delta\theta^1)_i} \left[\frac{T_E - T_P}{(h_1 \Delta\theta^1)_e} - \frac{T_P - T_w}{(h_1 \Delta\theta^1)_w} \right] if\ G_e > 0 \\ &= \frac{(h_1 \Delta\theta^1)_e^2}{(h_1 \Delta\theta^1)_{i+1}} \left[\frac{T_{EE} - T_E}{(h_1 \Delta\theta^1)_{ee}} - \frac{T_E - T_P}{(h_1 \Delta\theta^1)_e} \right] if\ G_e < 0 \end{aligned} \tag{3.31}$$

and

$$\begin{aligned} (h_1 \Delta\theta^1)_e &= \frac{1}{2} [(h_1 \Delta\theta^1)_i + (h_1 \Delta\theta^1)_{i+1}] \\ (h_1 \Delta\theta^1)_w &= \frac{1}{2} [(h_1 \Delta\theta^1)_i + (h_1 \Delta\theta^1)_{i-1}] \\ (h_1 \Delta\theta^1)_{ee} &= \frac{1}{2} [(h_1 \Delta\theta^1)_{i+1} + (h_1 \Delta\theta^1)_{i+2}] \\ (h_1 \Delta\theta^1)_{ww} &= \frac{1}{2} [(h_1 \Delta\theta^1)_{i-1} + (h_1 \Delta\theta^1)_{i-2}] \end{aligned} \tag{3.32}$$

Equation (3.22) now becomes:

$$\begin{aligned} &[(\rho C_{pm} T)^n - (\rho C_{pm} T)^{n-1}] \, h_1 \frac{\Delta V}{\Delta t} \\ &= A_E T_E + A_W T_W - A_P T_P + S \, (h_1 \Delta\theta^1) \end{aligned} \tag{3.33}$$

T_{ee} and T_{ww} are included in the source term using a semi-implicit tri-diagonal solution procedure. For a uniform grid, the other coefficients are:

$$A_E = \frac{C_{pm_e}\,(\,-7\,G_e + 3\,|\,G_e\,|\,)}{16} + C_{pm_w}\,(\,-G_w + |\,G_w\,|\,) + \frac{k_e}{h_1 \Delta\theta^1} \qquad \textbf{(3.34)}$$

$$A_W = \frac{C_{pm_w}\,(\,9\,G_w + 3\,|\,G_w\,|\,)}{16} + C_{pm_e}\,(\,G_e + |\,G_e\,|\,) + \frac{k_w}{h_1 \Delta\theta^1} \qquad \textbf{(3.35)}$$

$$A_P = \frac{9}{16}\,(\,G_w C_{pm_w} - G_e C_{pm_e}\,) + 3\,(\,|\,G_w\,|\,C_{pm_w} + G_e\,) + \frac{k_w + k_e}{h_1 \Delta\theta^1} \qquad \textbf{(3.36)}$$

$$S_P = S h_1 \Delta\theta^1 - C_{pm_e}\,(\,|\,G_e\,| - G_e\,)\,T_{EE} - C_{pm_w}\,(\,|\,G_w\,| + G_w\,)\,T_{WW} \qquad \textbf{(3.37)}$$

As mentioned before, Yang [Ref. 12:pp. 82-89] extended the QUICK algorithm to three dimensions. The three dimensional algorithm for generalized orthogonal coordinate system is described below.

As in the one dimensional case, the average temperature of the control volume is determined by interpolation of its neighbors in three directions. For illustration, Figure 3.6 from Raycraft [Ref. 29:pp. 68] shows a simpler uniform rectangular grid. The actual grid is similar except that its cylindrical/spherical geometry is more difficult to show. Yang [Ref. 12] describes how curvature terms are calculated for each of the temperatures and substituted into the convection

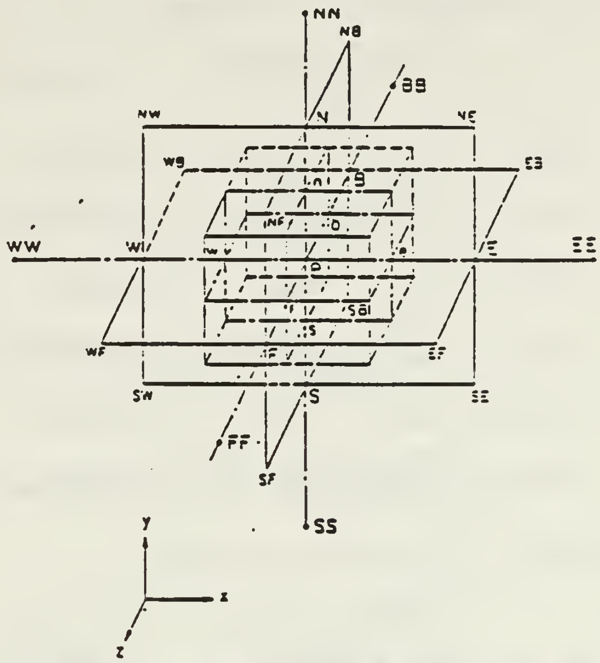

Figure 3.6 Calculation Cells for a Uniform Rectangular Grid.

terms of the energy equation. The new energy equation becomes

$$[\, A_P^T + (\rho\, C_{pm_p})^{n-1} \,] \frac{\Delta V}{\Delta t} T_P$$
$$= A_E^T T_E + A_W^T T_W + A_N^T T_N + A_S^T T_S + A_F^T T_F + A_B^T T_B + S_U^T \tag{3.38}$$

where the additional source term S_u^T is:

$$S_u^T = (\rho\, C_{pm_p})^{n-1} \frac{\Delta V}{\Delta t} - A_{EER} + A_{WWR} + A_{NNR} + A_{SSR} + A_{FFR} + A_{BBR} \tag{3.39}$$

The following terms are part of Equation (3.38). All values are for point (i, j, k) unless specified elsewhere. For = =

example, u^1_{ijk} is designated u_1^1 whereas, $u_{i+1,\,j,\,k}$ is specified u_{i+1}.

$$
\begin{aligned}
CN &= G_n \cdot u^2_{j+1} \cdot (h_3\Delta\theta^3)_n \cdot (h_1\Delta\theta^1)_n \\
CS &= G_s \cdot u^2_j \cdot (h_3\Delta\theta^3)_s \cdot (h_1\Delta\theta^1)_s \\
CE &= G_e \cdot u^1_{i+1} \cdot (h_3\Delta\theta^3)_e \cdot (h_2\Delta\theta^2)_e \\
CW &= G_w \cdot u^1_i \cdot (h_3\Delta\theta^3)_w \cdot (h_2\Delta\theta^2)_w \\
CF &= G_f \cdot u^3_{k+1} \cdot (h_1\Delta\theta^1)_f \cdot (h_2\Delta\theta^2)_f \\
CB &= G_s \cdot u^3_k \cdot (h_1\Delta\theta^1)_b \cdot (h_2\Delta\theta^2)_b
\end{aligned}
\tag{3.40}
$$

Thermal conductivity is expressed as:

$$
\begin{aligned}
k_n &= \left[\frac{(k_j \cdot (h_2\Delta\theta^2)_j)^{-1} + (k_{j+1} \cdot (h_2\Delta\theta^2)_{j+1})^{-1}}{(h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j+1}}\right]^{-1} \\
k_s &= \left[\frac{(k_j \cdot (h_2\Delta\theta^2)_j)^{-1} + (k_{j-1} \cdot (h_2\Delta\theta^2)_{j-1})^{-1}}{(h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j-1}}\right]^{-1} \\
k_e &= \left[\frac{(k_i \cdot (h_1\Delta\theta^1)_i)^{-1} + (k_{i+1} \cdot (h_1\Delta\theta^1)_{i+1})^{-1}}{(h_1\Delta\theta^1)_i + (h_1\Delta\theta^1)_{i+1}}\right]^{-1} \\
k_w &= \left[\frac{(k_i \cdot (h_1\Delta\theta^1)_i)^{-1} + (k_{i-1} \cdot (h_1\Delta\theta^1)_{i_1})^{-1}}{(h_1\Delta\theta^1)_i + (h_1\Delta\theta^1) i_1}\right]^{-1} \\
k_f &= \left[\frac{(k_k \cdot (h_3\Delta\theta^3)_k)^{-1} + (k_{k+1} \cdot (h_3\Delta\theta^3)_{k+1})^{-1}}{(h_3\Delta\theta^3)_k + (h_3\Delta\theta^3)_{k+1}}\right]^{-1} \\
k_b &= \left[\frac{(k_k \cdot (h_3\Delta\theta^3)_k)^{-1} + (k_{k-1} \cdot (h_3\Delta\theta^3)_{k-1})^{-1}}{(h_3\Delta\theta^3)_k + (h_3\Delta\theta^3)_{k-1}}\right]^{-1}
\end{aligned}
\tag{3.41}
$$

$$
\begin{gathered}
CONDN1 = k_n \cdot \left[\frac{h_3 \Delta\theta^3 \cdot h_1 \Delta\theta^1}{h_2 \Delta\theta^2} \right]_n \\
CONDS1 = k_s \cdot \left[\frac{h_3 \Delta\theta^3 \cdot h_1 \Delta\theta^1}{h_2 \Delta\theta^2} \right]_s \\
CONDE1 = k_e \cdot \left[\frac{h_3 \Delta\theta^3 \cdot h_2 \Delta\theta^2}{h_1 \Delta\theta^1} \right]_e \\
CONDW1 = k_w \cdot \left[\frac{h_3 \Delta\theta^3 \cdot h_2 \Delta\theta^2}{h_1 \Delta\theta^1} \right]_w \\
CONDF1 = k_f \cdot \left[\frac{h_1 \Delta\theta^1 \cdot h_2 \Delta\theta^2}{h_3 \Delta\theta^3} \right]_f \\
CONDB1 = k_b \cdot \left[\frac{h_1 \Delta\theta^1 \cdot h_2 \Delta\theta^2}{h_3 \Delta\theta^3} \right]_b
\end{gathered}
\tag{3.42}
$$

$$
\begin{aligned}
CEP &= \frac{|CE| + CE}{16} \cdot \frac{(h_1 \Delta\theta^1)_e}{(h_1 \Delta\theta^1)_i} \\
CEM &= \frac{|CE| - CE}{16} \cdot \frac{(h_1 \Delta\theta^1)_e}{(h_1 \Delta\theta^1)_{i+1}} \\
CWP &= \frac{|CW| + CW}{16} \cdot \frac{(h_1 \Delta\theta^1)_w}{(h_1 \Delta\theta^1)_{i-1}} \\
CWM &= \frac{|CW| - CW}{16} \cdot \frac{(h_1 \Delta\theta^1)_w}{(h_1 \Delta\theta^1)_i} \\
CNP &= \frac{|CN| + CN}{16} \cdot \frac{(h_2 \Delta\theta^2)_n}{(h_2 \Delta\theta^2)_j} \\
CNM &= \frac{|CN| - CN}{16} \cdot \frac{(h_2 \Delta\theta^2)_n}{(h_2 \Delta\theta^2)_{j+1}} \\
CSP &= \frac{|CS| + CS}{16} \cdot \frac{(h_2 \Delta\theta^2)_s}{(h_2 \Delta\theta^2)_{j-1}} \\
CSM &= \frac{|CS| - CS}{16} \cdot \frac{(h_2 \Delta\theta^2)_s}{(h_2 \Delta\theta^2)_j} \\
CFP &= \frac{|CF| + CF}{16} \cdot \frac{(h_3 \Delta\theta^3)_f}{(h_3 \Delta\theta^3)_k} \\
CFM &= \frac{|CF| - CF}{16} \cdot \frac{(h_3 \Delta\theta^3)_f}{(h_3 \Delta\theta^3)_{k+1}} \\
CBP &= \frac{|CB| + CB}{16} \cdot \frac{(h_3 \Delta\theta^3)_b}{(h_3 \Delta\theta^3)_{k-1}} \\
CBM &= \frac{|CB| - CB}{16} \cdot \frac{(h_3 \Delta\theta^3)_b}{(h_3 \Delta\theta^3)_k} .
\end{aligned}
\tag{3.43}
$$

$$A_{EE}^{T} = \frac{-CEM \cdot (h_1 \Delta\theta^1)_e}{(h_1 \Delta\theta^1)_{ee}}$$

$$A_{WW}^{T} = \frac{-CWP \cdot (h_1 \Delta\theta^1)_w}{(h_1 \Delta\theta^1)_{ww}}$$

$$A_{NN}^{T} = \frac{-CNM \cdot (h_2 \Delta\theta^2)_n}{(h_2 \Delta\theta^2)_{nn}}$$

$$A_{SS}^{T} = \frac{-CSP \cdot (h_2 \Delta\theta^2)_s}{(h_2 \Delta\theta^2)_{ss}} \tag{3.44}$$

$$A_{FF}^{T} = \frac{-CFM \cdot (h_3 \Delta\theta^3)_f}{(h_3 \Delta\theta^3)_{ff}}$$

$$A_{BB}^{T} = \frac{-CBP \cdot (h_3 \Delta\theta^3)_b}{(h_3 \Delta\theta^3)_{bb}}$$

Final coefficients for the source term are:

$$A_{EER} = A_{EE}^{T} \cdot T_{i+2} \cdot C_{pm_{i+1}}$$

$$A_{WWR} = A_{WW}^{T} \cdot T_{i-2} \cdot C_{pm_{i-2}}$$

$$A_{NNR} = A_{NN}^{T} \cdot T_{j+2} \cdot C_{pm_{j+2}}$$

$$A_{SSR} = A_{SS}^{T} \cdot T_{j-2} \cdot C_{pm_{j-2}} \tag{3.45}$$

$$A_{FFR} = A_{FF}^{T} \cdot T_{k+2} \cdot C_{pm_{k+2}}$$

$$A_{BBR} = A_{BB}^{T} \cdot T_{k-2} \cdot C_{pm_{k-2}}$$

Intermediate coefficients are:

$$A_{EI} = -\frac{1}{2} \cdot CE + CEP + CEM$$

$$\cdot \left[1 + \frac{(h_1 \Delta\theta^1)_e}{(h_1 \Delta\theta^1)_{ee}} \right] + CWM \cdot \left[\frac{(h_1 \Delta\theta^1)_w}{(h_1 \Delta\theta^1)_e} \right] \tag{3.46}$$

$$A_{WI} = \frac{1}{2} \cdot CW + CWM + CWP$$

$$\cdot \left[1 + \frac{(h_1 \Delta\theta^1)_w}{(h_1 \Delta\theta^1)_{ww}} \right] + CEP \cdot \left[\frac{(h_1 \Delta\theta^1)_e}{(h_1 \Delta\theta^1)_w} \right] \tag{3.47}$$

$$A_{NI} = -\frac{1}{2} \cdot CN + CNP + CNM$$

$$\cdot \left[1 + \frac{(h_2 \Delta\theta^2)_n}{(h_2 \Delta\theta^2)_{nn}} \right] + CEP \cdot \left[\frac{(h_2 \Delta\theta^2)_s}{(h_2 \Delta\theta^2)_n} \right] \tag{3.48}$$

$$A_{SI} = \frac{1}{2} \cdot CS + CSM + CSP$$

$$\cdot \left[1 + \frac{(h_2 \Delta\theta^2)_s}{(h_2 \Delta\theta^2)_{ss}} \right] + CNP \cdot \left[\frac{(h_2 \Delta\theta^2)_n}{(h_2 \Delta\theta^2)_s} \right] \tag{3.49}$$

$$A_{FI} = -\frac{1}{2} \cdot CF + CFP + CFM$$

$$\cdot \left[1 + \frac{(h_3 \Delta\theta^3)_f}{(h_3 \Delta\theta^3)_{ff}} \right] + CBM \cdot \left[\frac{(h_3 \Delta\theta^3)_b}{(h_3 \Delta\theta^3)_f} \right] \tag{3.50}$$

$$A_{BI} = \frac{1}{2} \cdot CB + CBM + CBP$$

$$\cdot \left[1 + \frac{(h_3 \Delta\theta^3)_b}{(h_3 \Delta\theta^3)_{bb}} \right] + CFP \cdot \left[\frac{(h_3 \Delta\theta^3)_f}{(h_3 \Delta\theta^3)_b} \right] \tag{3.51}$$

Final coefficients are:

$$A_E^T = A_{EI} \cdot C_{pm_E} + CONDE1$$
$$A_W^T = A_{WI} \cdot C_{pm_W} + CONDW1$$
$$A_N^T = A_{NI} \cdot C_{pm_N} + CONDN1$$
$$A_S^T = A_{SI} \cdot C_{pm_S} + CONDS1 \qquad (3.52)$$
$$A_F^T = A_{FI} \cdot C_{pm_F} + CONDF1$$
$$A_B^T = A_{BI} \cdot C_{pm_B} + CONDB1$$

and:

$$A_P^T = C_{p_{m,P}} \cdot (A_E^T + A_W^T + A_N^T + A_S^T + A_F^T + A_B^T$$
$$+ A_{EE}^T + A_{WW}^T + A_{NN}^T + A_{SS}^T + A_{FF}^T + A_{BB}^T) + CONDE1 \qquad (3.53)$$
$$+ CONDW1 + CONDN1 + CONDS1 + CONDF1 + CONDB1$$

F. DISCRETIZATION OF THE MOMENTUM EQUATION

The integrated momentum equation is:

$$(\rho u^i)_t V + M_e^{i1} A_e - M_w^{i1} A_w + M_n^{i2} A_n$$
$$- M_s^{i2} A_s + M_f^{i3} A_f - M_b^{i3} A_b = S^i \qquad (3.54)$$

where A_i are the face areas of the staggered cell given by Equations (3.14 - 3.16). M^{ij} is the momentum flux in the θ^{ij} direction due to velocity u^i convection and to diffusion, and is given by:

$$M^{ij} = (\rho u^i u^j - \sigma_i^j) \qquad (3.55)$$

Included in the source term S^i are pressure gradient, body, coriolis and centrifugal forces. The source term for velocity u^i is:

$$S^i = -P_e A_e + P_w A_w + \rho G^1 \Delta V$$
$$- M_p^{12} (A_n - A_s) - M_p^{13} (A_f - A_b) + (M_p^{22} + M_p^{33}) (A_e + A_w) \quad (3.56)$$

Yang, *et al.* [Ref. 19:pp. 11-13] describe a "stress flux formation" as it applies to a curvilinear coordinate system. Stresses are evaluated from previous information and the source is given in the current information. The momentum flux is:

$$M^{ij} = \hat{M}^{ij} + (\sigma_i^j - \sigma_i^j) \quad (3.57)$$

where:

$$\sigma_i^j = \frac{\mu}{\left[h_j \left(\frac{\partial u^i}{\partial \theta^j} \right) \right]} \quad (3.58)$$

$$\hat{M}^{ij} = \rho u^i u^j - \sigma_i^j \quad (3.59)$$

The momentum equation for velocity u^i is now:

$$(\rho u)_t + \hat{M}_e^{11} A_e - \hat{M}_w^{11} A_w + \hat{M}_n^{12} A_n - \hat{M}_s^{12} A_s + \hat{M}_f^{13} A_f + \hat{M}_b^{13} A_b = \hat{S} \quad (3.60)$$

where:

$$\hat{S} = S - (\sigma_1^1 - \sigma_1^1)_e A_e + (\sigma_1^1 - \sigma_1^1)_w A_w - (\sigma_1^2 - \sigma_1^2)_n A_n$$
$$+ (\sigma_1^2 - \sigma_1^2)_s A_s - (\sigma_1^3 - \sigma_1^3)_f A_f - (\sigma_1^3 - \sigma_1^3)_b A_b \quad (3.61)$$

The momentum equation for θ^1 takes a form similar to the energy equation

$$\left(A_p^{u1} + \rho^{n-1}\frac{\Delta V}{\Delta t}\right) u_p^2 = A_e^{u^1} u_e^1 + A_w^{u^1} u_w^1 + A_n^{u^1} u_n^1 + A_s^{u^1} u_s^1 + A_f^{u^1} u_f^1 + A_b^{u^1} u_b^1 + S^{u^1} u^1 \tag{3.62}$$

Again we must obtain final coefficients. Introducing intermediate mass flow rate per unit area:

$$G_{ne} = u_{j+1}^2 \left\{\frac{\left[\rho_{j+1}\,(h_2\Delta\theta^2)_j + \rho_j\,(h_2\Delta\theta^2)_{j+1}\right]}{(h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j+1}}\right\}$$

$$G_{nw} = u_{i-1,\,j+1}^2 \left\{\frac{\left[\rho_{i-1,\,j+1}\,(h_2\Delta\theta^2)_j + \rho_{i-1}\,(h_2\Delta\theta^2)_{j+1}\right]}{(h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j+1}}\right\}$$

$$G_{se} = u^2 \left\{\frac{\left[\rho_{j-1}\,(h_2\Delta\theta^2)_j + \rho_j\,(h_2\Delta\theta^2)_{j-1}\right]}{(h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j+1}}\right\} \tag{3.63a}$$

$$G_{sw} = u_{i-1}^2 \left\{\frac{\left[\rho_{i-1,\,j-1}\,(h_2\Delta\theta^2)_j + \rho_{i-1}\,(h_2\Delta\theta^2)_{j-1}\right]}{(h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j-1}}\right\}$$

$$G_e = u_{i+1}^1 \left\{\frac{\left[\rho_{i+1}\,(h_1\Delta\theta^1)_e + \rho_i\,(h_1\Delta\theta^2)_{ee}\right]}{(h_1\Delta\theta^1)_e + (h_1\Delta\theta^1)_{ee}}\right\}$$

$$G_p = u^1 \{ \frac{\left[\rho_{i-1} \, (h_1 \Delta\theta^1)_e + \rho_i \, (h_1 \Delta\theta^1)_w \right]}{(h_1 \Delta\theta^1)_e + h_1 \Delta\theta^1)_w} \}$$

$$G_w = u^1_{i-1} \{ \frac{\left[\rho_{i-2} \, (h_1 \Delta\theta^1)_w + \rho_{i-1} \, (h_1 \Delta\theta^1)_{ww} \right]}{(h_1 \Delta\theta^1)_w + (h_1 \Delta\theta^1)_{ww}} \}$$

$$G_{fe} = u^3_{k+1} \{ \frac{\left[\rho_{k+1} \, (h_3 \Delta\theta^3)_k + \rho_k \, (h_3 \Delta\theta^3)_{k+1} \right]}{(h_3 \Delta\theta^3)_k + (h_3 \Delta\theta^3)_{k+1}} \}$$

$$G_{fw} = u^3_{i-1,\,k+1} \{ \frac{\left[\rho_{i-1,\,k+1} \, (h_3 \Delta\theta^3)_k + \rho_{i-1} \, (h_3 \Delta\theta^3)_{k+1} \right]}{(h_3 \Delta\theta^3)_k + (h_3 \Delta\theta^3)_{k+1}} \} \tag{3.63b}$$

$$G_{be} = u^3 \{ \frac{\left[\rho_{k-1} \, (h_3 \Delta\theta^3)_k + \rho_k \, (h_3 \Delta\theta^3)_{k-1} \right]}{(h_3 \Delta\theta^3)_k + (h_3 \Delta\theta^3)_{k-1}} \}$$

$$G_{bw} = u^3_{i-1} \{ \frac{\left[\rho_{i-1,\,k-1} \, (h_3 \Delta\theta^3)_k + \rho_{i-1} \, (h_3 \Delta\theta^3)_{k-1} \right]}{(h_3 \Delta\theta^3)_k + (h_3 \Delta\theta^3)_{k-1}} \}$$

Final mass flow rates are:

$$CE = \frac{1}{2} \, (G_e + G_p) \cdot (h_2 \Delta\theta^2)_e \cdot (h_3 \Delta\theta^3)_e$$

$$CW = \frac{1}{2} \, (G_p + G_w) \cdot (h_2 \Delta\theta^2)_w \cdot (h_3 \Delta\theta^3)_w$$

$$CN = (h_1 \Delta\theta^1)_n \cdot (h_3 \Delta\theta^3)_n \cdot \{ \frac{[G_{ne} \, (h_1 \Delta\theta^1)_w + G_{nw} \, (h_1 \Delta\theta^1)_e]}{(h_1 \Delta\theta^1)_w + (h_1 \Delta\theta^1)_e}$$

$$CS = (h_1 \Delta\theta^1)_s \cdot (h_3 \Delta\theta^3)_s \cdot \{ \frac{[G_{se} \, (h_1 \Delta\theta^1)_w + G_{sw} \, (h_1 \Delta\theta^1)_e]}{(h_1 \Delta\theta^1)_w + (h_1 \Delta\theta^1)_e} \tag{3.64}$$

$$CF = (h_1 \Delta\theta^1)_f \cdot (h_2 \Delta\theta^2)_f \cdot \{ \frac{[G_{fe} \, (h_1 \Delta\theta^1)_w + G_{fw} \, (h_1 \Delta\theta^1)_e]}{(h_1 \Delta\theta^1)_w + (h_1 \Delta\theta^1)_e}$$

$$CB = (h_1 \Delta\theta^1)_b \cdot (h_2 \Delta\theta^2)_b \cdot \{ \frac{[G_{be} \, (h_1 \Delta\theta^1)_w + G_{bw} \, (h_1 \Delta\theta^1)_e]}{(h_1 \Delta\theta^1)_w + (h_1 \Delta\theta^1)_e}$$

The local viscosity is:

$$VIS_e = VIS$$

$$VIS_w = VIS_{i-1}$$

$$VIS_n = \frac{(VIS_{j+1} + VIS + VIS_{i-1,\, j+1} + VIS_{i-1})}{4}$$

$$VIS_s = \frac{(VIS_{j-1} + VIS + VIS_{i-1,\, j-1} + VIS_{i-1})}{4} \qquad (3.65)$$

$$VIS_f = \frac{(VIS_{k+1} + VIS + VIS_{i-1,\, k+1} + VIS_{i-1})}{4}$$

$$VIS_b = \frac{(VIS_{k-1} + VIS + VIS_{i-1,\, k-1} + VIS_{i-1})}{4}$$

$$VISN1 = VIS_n \cdot \left[\frac{(h_3 \Delta\theta^3)\ (h_1 \Delta\theta^1)}{(h_2 \Delta\theta^2)} \right]_n$$

$$VISS1 = VIS_s \cdot \left[\frac{(h_3 \Delta\theta^3)\ (h_1 \Delta\theta^1)}{(h_2 \Delta\theta^2)} \right]_s$$

$$VISE1 = VIS_e \cdot \left[\frac{(h_2 \Delta\theta^2)\ (h_3 \Delta\theta^3)}{(h_1 \Delta\theta^1)} \right]_e \qquad (3.66)$$

$$VISW1 = VIS_w \cdot \left[\frac{(h_2 \Delta\theta^2)\ (h_3 \Delta\theta^3)}{(h_1 \Delta\theta^1)} \right]_w$$

$$VISF1 = VIS_f \cdot \left[\frac{(h_1 \Delta\theta^1)\ (h_2 \Delta\theta^2)}{(h_3 \Delta\theta^3)} \right]_f$$

$$VISB1 = VIS_b \cdot \left[\frac{(h_1 \Delta\theta^1)\ (h_2 \Delta\theta^2)}{(h_3 \Delta\theta^3)} \right]_b$$

The momentum equation coefficients are:

$$
\begin{aligned}
A_{EER} &= A^u_{EE} \cdot u^i_{i+1} \\
A_{WWR} &= A^u_{WW} \cdot u^i_{i-2} \\
A_{NNR} &= A^u_{NN} \cdot u^i_{j+2} \\
A_{SSR} &= A^u_{SS} \cdot u^i_{j-2} \\
A_{FFR} &= A^u_{FF} \cdot u^i_{k+2} \\
A_{BBR} &= A^u_{BB} \cdot u^i_{k-2}
\end{aligned}
\tag{3.67}
$$

As with the energy equation, the value of the final coefficients are:

$$
\begin{aligned}
A^u_E &= A_{EI} + VISE1 \\
A^u_W &= A_{WI} + VISW1 \\
A^u_N &= A_{NI} + VISN1 \\
A^u_S &= A_{SI} + VISS1 \\
A^u_F &= A_{FI} + VISF1 \\
A^u_B &= A_{BI} + VISB1
\end{aligned}
\tag{3.68}
$$

and

$$
\begin{aligned}
A^u_P = {} & A^u_E + A^u_W + A^u_N + A^u_S + A^u_F + A^u_B \\
& + A^u_{EE} + A^u_{WW} + A^u_{NN} + A^u_{SS} + A^u_{FF} + A^u_{BB}
\end{aligned}
\tag{3.69}
$$

The final source term is given as

$$
\begin{aligned}
S_u^u = {} & \frac{[\rho\,(h_1\Delta\theta^1)_w + \rho_{i-1}\,(h_1\Delta\theta^1)_e]}{[\,(h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e\,]} \cdot \frac{\Delta V}{\Delta t} \cdot u^1 \\
& + (h_2\Delta\theta^2)_j\,(h_3\Delta\theta^3)_k\,(P_{i-1} - P_i) + A_{EER} + A_{WWR} + A_{NNR} \\
& + A_{SSR} + A_{FFR} + A_{BBR} + RE - RW + RN - RS \\
& + RF - RB + RRY + RRZ - RRX - BUOY \\
& \cdot \{\sin[\,ZC(K)\,] \cdot (\rho - \rho_{eq}) \cdot (h_1\Delta\theta^1)_w \\
& \cdot \cos[\,XC(I)\,]\} + \{(\rho_{i-1} - \rho_{eq_{i-1}})\,(h_1\Delta\theta^1)_e \\
& \cdot \cos[\,XC(I-1)\,]\} / [\,(h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e\,]\,\Delta V
\end{aligned}
\tag{3.70}
$$

where XC and ZC represent the center of the cell. The remainder of the terms are explained below.

$$
\begin{aligned}
RE &= (h_2\Delta\theta^2)_e\,(h_3\Delta\theta^3)_e \cdot \left[\frac{\sigma^{11} - (u^1_{i+1} - u^1_i) \cdot VIS_e}{(h_1\Delta\theta^1)_e}\right] \\
RW &= (h_2\Delta\theta^2)_w\,(h_3\Delta\theta^3)_w \cdot \left[\frac{\sigma^{11}_{i-1} - (u^1 - u^1_{i-1}) \cdot VIS_w}{(h_1\Delta\theta^1)_w}\right] \\
RN &= (h_1\Delta\theta^1)_n\,(h_3\Delta\theta^3)_n \cdot \left[\frac{\sigma^{12}_{j+1} - (u^1_{j+1} - u^1_j) \cdot VIS_n}{(h_2\Delta\theta^2)_n}\right] \\
RS &= (h_1\Delta\theta^1)_s\,(h_3\Delta\theta^3)_s \cdot \left[\frac{\sigma^{12} - (u^1 - u^1_{j-1}) \cdot VIS_s}{(h_2\Delta\theta^2)_s}\right] \\
RF &= (h_1\Delta\theta^1)_f\,(h_2\Delta\theta^2)_f \cdot \left[\frac{\sigma^{13}_{k+1} - (u^1_{k+1} - u^1_k) \cdot VIS_f}{(h_3\Delta\theta^3)_f}\right] \\
RB &= (h_1\Delta\theta^1)_b\,(h_2\Delta\theta^2)_b \cdot \left[\frac{\sigma^{13} - (u^1 - u^1_{k-i}) \cdot VIS_b}{(h_1\Delta\theta^1)_b}\right]
\end{aligned}
\tag{3.71}
$$

$$\overline{\sigma}^{12} = \frac{1}{2}\left(\sigma^{12}_{j+1} + \sigma^{12}_{j}\right)$$

$$\overline{\sigma}^{13} = \frac{1}{2}\left(\sigma^{13}_{k+1} + \sigma^{13}_{k}\right)$$

$$\overline{\sigma}^{22} = \frac{\sigma^{22}\,(h_1\Delta\theta^1)_w + \sigma^{22}_{i-1}\,(h_1\Delta\theta^1)_e}{(h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e} \qquad (3.72)$$

$$\overline{\sigma}^{33} = \frac{\sigma^{33}\,(h_1\Delta\theta^1)_w + \sigma^{33}_{i-1}\,(h_1\Delta\theta^1)_e}{(h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e}$$

$$AU1 = u^1$$

$$AU2 = \left\{\left[\frac{u^2_{j+1}\,(h_2\Delta\theta^2)_j + u^2_j\,(h_2\Delta\theta^2)_j}{2\,(h_2\Delta\theta^2)_j}\right](h_1\Delta\theta^1)_w + \left[\frac{u^2_{i-1,\,j+1}\,(h_2\Delta\theta^2)_j + u^2_{i-1}\,(h_2\Delta\theta^2)_j}{2\,(h_2\Delta\theta^2)_j}\right](h_1\Delta\theta^1)_e\right\} / \left[(h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e\right] \qquad (3.73)$$

$$AU3 = \left\{\left[\frac{u^3_{k+1}\,(h_3\Delta\theta^3)_k + u^3_k\,(h_3\Delta\theta^3)_k}{2\,(h_3\Delta\theta^3)_k}\right](h_1\Delta\theta^1)_w + \frac{\left[u^3_{i-1,\,k+1}\,(h_3\Delta\theta^3)_k + u^3_{i-1}\,(h_3\Delta\theta^3)_k\right]}{2\,(H_3\Delta\theta^3)}(h_1\Delta\theta^1)_e\right\} / \left[(h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e\right]$$

$$AR = \frac{\rho\,(h_1\Delta\theta^1)_w + \rho_{i-1}\,(h_1\Delta\theta^1)_e}{(h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e}$$

$$ARU12 = AR \cdot AU1 \cdot AU2 \qquad (3.74)$$

$$ARU13 = AR \cdot AU1 \cdot AU3$$

$$ARU22 = AR \cdot AU2 \cdot AU2$$

$$ARU33 = AR \cdot AU3 \cdot AU3$$

$$RRY = (\overline{\sigma}^{12} - ARU12)\,(h_3\Delta\theta^3)_k[\,(h_1\Delta\theta^1)_n - (h_1\Delta\theta^1)_s]$$

$$RRZ = (\overline{\sigma}^{13} - ARU13)\,(h_2\Delta\theta^2)_j[\,(h_1\Delta\theta^1)_f - (h_1\Delta\theta^1)_b]$$

$$RRX = (\overline{\sigma}^{22} - AUR22)\,(h_3\Delta\theta^3)_k[\,(h_2\Delta\theta^2)_e - (h_2\Delta\theta^2)_w] + (\overline{\sigma}^{33} - AUR33)\,(h_2\Delta\theta^2)_j[\,(h_3\Delta\theta^3)_e - (h_3\Delta\theta^3)_w] \qquad (3.75)$$

Similarly, momentum equations for the other two directions may be obtained but are omitted for brevity.

G. PRESSURE CORRECTION

In the finite difference scheme, energy and momentum equations are used to solve for temperature and velocities. The equation of state and continuity are used to solve for density and pressure. Doria [Ref. 35] states that pressure is only weakly coupled to the equation of state. Therefore, updated temperatures and pressures determine density in the equation of state and continuity is used to correct pressure across each cell.

As discussed earlier, a disadvantage of using primitive variables is the difficulty in calculating pressure. Two corrections must be applied. First, a global pressure correction accounts for changes in net energy of the closed system. Second, a local pressure correction accounts for pressure changes causing the velocity field.

1. Global Pressure Correction

Nicolette, *et al.* [Ref. 3] developed a correction scheme for a two dimensional square enclosure. Raycraft [Ref.

30] modified it to fit the geometry of Fire-1. In a constant mass and volume system, the overall pressure depends on the addition or removal of energy. In such a system, the sum of all the cells' computed density times its volume is equal to a constant total mass. At any time during a run the mass must equal the total mass at equilibrium. Summing over N cells:

$$\sum \rho_i^n (\Delta V)_i = \sum \rho_{EQ_i} (\Delta V) \tag{3.76}$$

where n is the n^{th} time step and the EQ subscript indicates the equilibrium point. Assuming that air is an ideal gas, its density is a function of temperature and pressure only. The actual values of both consist of the estimate and the global correction:

$$P = P^* + P_g' \tag{3.77}$$

$$T = T^* + T_g' \tag{3.78}$$

where P^* and T^* are the estimates and P_g' and T_g' are the global corrections using the ideal gas law and Equation (3.76). The global pressure correction becomes

$$P_g' = \frac{\sum P_{EQ}\left(\frac{\Delta V}{T_i} - \frac{\Delta V}{T^*}\right) - \sum\left(P^* \frac{\Delta V}{T^*}\right)}{\sum \frac{\Delta V}{T^*}} \tag{3.79}$$

Mass is conserved for each cell when an accurate final pressure is obtained.

2. Local Pressure Correction

Patanker [Ref. 36:pp. 120-126] and Doria [Ref. 36:pp. 26-32] developed a procedure for obtaining the local pressure correction. As in the global correction scheme, a pressure field is estimated from the previous time step. Velocities are calculated according to this pressure distribution and the law of continuity is applied to each cell. If the residual mass term S_{mp} approaches zero, then the estimated pressure field is satisfactory. If not,a local correction is calculated and applied to the original estimate. The new pressure field is used to compute a corrected velocity field and the residual mass S_{mp} is rechecked. The process repeats itself until S_{mp} is an acceptably small value. As in the global correction, the actual local pressure is:

$$P = P^{*} + P' \tag{3.80}$$

where P^{*} is again the estimate, usually the pressure of the preceding iteration, and P' is the local correction. Putting this correction in typical finite difference form:

$$A_P P_P' = A_E P_E' + A_W P_W' + A_N P_N' + A_S P_S' + A_F P_F' + A_B P_B' - S_{mp}\Delta V \tag{3.81}$$

where:

$$A_E = \frac{\rho_e \cdot [\,(h_2 \Delta\theta^2)\,(h_3 \Delta\theta^3)\,]_e^2}{\left[A^{u1}_{p_{i+1}} + \rho_e \dfrac{\Delta V}{\Delta t} \right]}$$

$$A_W = \frac{\rho_w \cdot [\,(h_2 \Delta\theta^2)\,(h_3 \Delta\theta^3)\,]_w^2}{\left[A^{u1}_{p} + \rho_w \dfrac{\Delta V}{\Delta t} \right]}$$

$$A_N = \frac{\rho_n \cdot [\,(h_1 \Delta\theta^1)\,(h_3 \Delta\theta^3)\,]_n^2}{\left[A^{u2}_{p_{j+1}} + \rho_n \dfrac{\Delta V}{\Delta t} \right]}$$

$$A_S = \frac{\rho_s \cdot [\,(h_1 \Delta\theta^1)\,(h_3 \Delta\theta^3)\,]_s^2}{\left[A^{u2}_{p} + \rho_s \dfrac{\Delta V}{\Delta t} \right]} \qquad (3.82)$$

$$A_F = \frac{\rho_f \cdot [\,(h_1 \Delta\theta^1)\,(h_2 \Delta\theta^2)\,]_f^2}{\left[A^{u3}_{p_{k+1}} + \rho_f \dfrac{\Delta V}{\Delta t} \right]}$$

$$A_B = \frac{\rho_b \cdot [\,(h_1 \Delta\theta^1)\,(h_2 \Delta\theta^2)\,]_b^2}{\left[A^{u3}_{p} + \rho_b \dfrac{\Delta V}{\Delta t} \right]}$$

$$A_p = A_E + A_W + A_N + A_S + A_F + A_B$$

Corrected velocities are:

$$u^1 = u^{1*} + u^{1'}$$

$$u^2 = u^{2*} + u^{2'} \qquad (3.83)$$

$$u^3 = u^{3*} + u^{3'}$$

where:

$$u^{1'} = \frac{(P_p + P_w)\ (h_2\Delta\theta^2)\ (h_3\Delta\theta^3)}{\left(A_P^{u1} + \rho_w \frac{\Delta V}{\Delta t}\right)}$$

$$u^{2'} = \frac{(P_p + P_s)\ (h_1\Delta\theta^1)\ (h_3\Delta\theta^3)}{\left(A_P^{u2} + \rho_s \frac{\Delta V}{\Delta t}\right)} \qquad (3.84)$$

$$u^{3'} = \frac{(P_p + P_b)\ (h_1\Delta\theta^1)\ (h_2\Delta\theta^2)}{\left(A_P^{u3} + \rho_b \frac{\Delta V}{\Delta t}\right)}$$

Again S_{mp} is computed using continuity. If the residual mass is within a satisfactory range, the calculation is finished. If not, another iteration takes place.

IV. NUMERICAL PROCESS

A. INTRODUCTION

Temperature, velocity, pressure and density fields are produced by the code. Input parameters are initial conditions, fuel heat release rate, fire location, geometry and material characteristics such as fluid properties, wall properties and the external heat transfer coefficient. These are listed in Table 4.1.

TABLE 4.1 MODEL PARAMETERS

WALL CHARACTERISTICS	
Material	ASTM 285 Grade C Steel
Thickness	3/8 inch
Specific Heat	0.1 BTU/ (lbm•F)
Thermal Conductivity	25 BTU/(hr•ft•F)
Density	487 lbm/ft^3
External Heat Transfer Coefficient	15.0 BTU/(hr•ft^2•F)
FIRE CHARACTERISTICS	
Burn Rate	Function provided in program
Initial Temperature	35.6°C
Initial Pressure	1.0 ATM
Location of Fire	Center of Fire-1 23.1 ft. from each endccap 3.21 ft. from bottom

Figures 4.1 and 4.2 show the spherical/cylindrical grid used by the model. Endcaps are spherical with θ, R, and ϕ

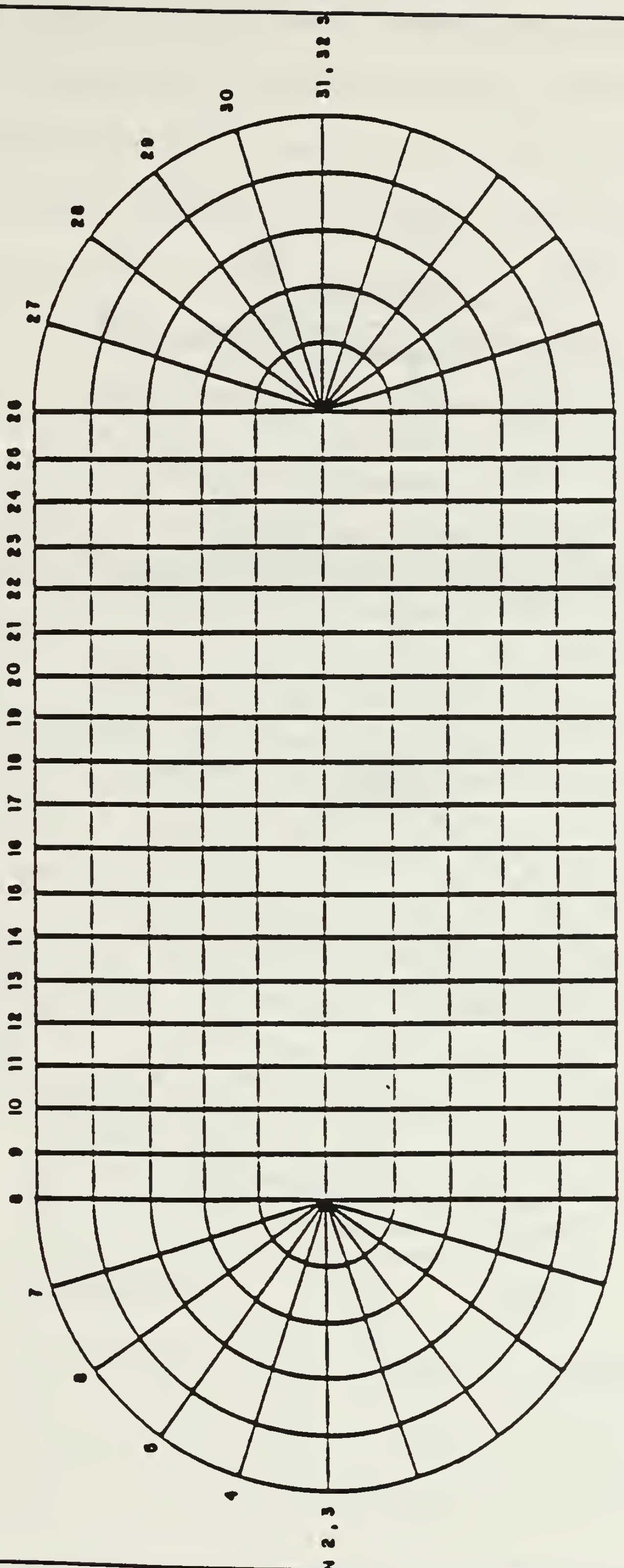

Figure 4.1 Front View of Computer Model (YZ-Plane)

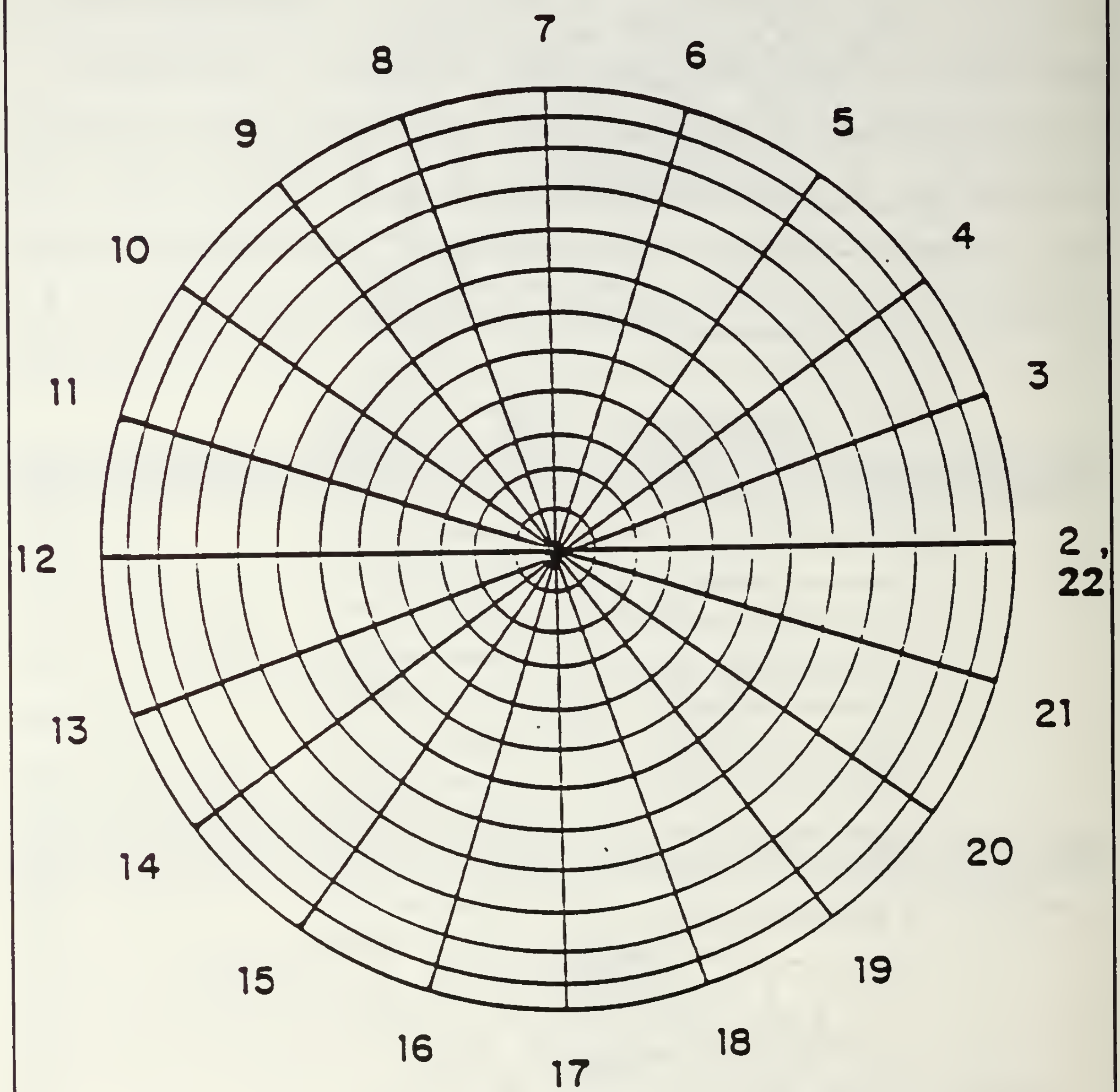

Figure 4.2   Side View of Computer Model (XY-Plane)

directions. The cylindrical midsection have θ, R and Z directions. There are 14 cells in the R direction, one at R=0 for avoiding singularity and one used as the vessel wall. There are 20 cells oriented clockwise in the θ direction. Each endcap has six cells in the φ direction with a cell again at zero to avoid singularity. The midsection has 18 cells in the Z direction (φ is used for simplicity). Table 4.2 gives information on grid parameters.

TABLE 4.2 ADDITIONAL MODEL PARAMETERS

GRID	
Number of interior cells	6,720
Number of wall cells	560
Number of wall radiation zones	560
Number of fire radiation zones	19
Number of cells in R direction	14
Number of cells in θ direction	20
Number of cells in φ direction (per endcap)	6
Number of cells in Z direction (midsection)	18
Time step	0.0288 sec
VAXSTATION 3100 CPU time (1 CPU hour)	0.8-1.0 sec Fire Time

B. SOLUTION PROCESS

The model contains two separate programs. The first authored by Raycraft [Ref. 29] calculates the view factors for surface radiation. It produces a matrix of view factors. It is used only once and its values are stored for use whenever called by the second program.

As described by Nies [Ref. 27], Raycraft [Ref. 29] and Houck [Ref. 30], the main program uses finite difference methods described previously to establish temperature, velocity, pressure and density fields. Initial parameters and the view factors are first read into the program. Geometry of the grid is then calculated and the fields are set to initial conditions. Next, effective viscosity is computed in subroutine CALVIS. Every two time steps, surface radiation flux is recalculated in subroutine RADHT. Subroutines CALT, GLOBE, CALU, CALX, CALW and CALP calculate temperature, the global pressure correction, the velocities and the local pressure correction. Using the corrected velocities, continuity is applied to each cell. If the residual mass RESORM is greater than 10.0 the program is unstable and stops. A smaller time step must be used. If RESORM is greater than a set tolerance level then the program iterates solution by recalculating velocities and pressures. Iterations continue until 1) RESORM is below tolerance levels, solution is reached and the program proceeds to next time step; 2) the maximum number of iterations is reached, or 3) RESORM is greater than 10.0 and the program is stopped.

C. GRAPHICAL ANALYSIS

The use of CA-DISSPLA™ [Ref. 31] has posed some difficult problems. The output from the computer model is in the spherical/cylindrical coordinate system created to simulate Fire-1. This output must be converted to cartesian coordinates in order to be manipulated by CA-DISSPLA™. Even after the conversion is completed the resulting irregularly spaced grid must be interpolated into a regularly spaced grid.

After some experimentation with grid interpolation schemes, a group of CA-DISSPLA™ subroutines are used to create a regularly spaced matrix. These subroutines interpolate values from a set number of neighbors. Care must be taken in choosing a grid size to ensure distortion of the field values does not occur and to ensure that the software will not zero data points with few close neighbors. A relatively course grid has been chosen (50 x 50 x 100) for graphics output. New data points created outside the enclosure have been set to ambient values to minimize distortion at the boundaries.

The VAXSTATION 3100 has proven to be an excellent machine. It has good numerical speed coupled with very sharp graphics capabilities. Future research of this numerical model has been greately enhanced by the incorporation of this workstation.

The following figures are temperature and velocity profiles for times of 30, 60, and 90 seconds. They are two dimensional images of three dimensional phenomena. Each figure, whether temperature or velocity, presents an axial view (XY-plane) of the tank at the midplane and a longitudinal view (YZ-plane), again at the midplane.

Raycraft [Ref. 29] and Houck [Ref. 30] detailed the validation of the code against experimental data of Fire-1. They also discussed such phenomena as the fire plume, pressure effect, temperature stratification, and velocity fields. Much of their analysis will not be repeated here. Instead, the effects of local numerical perturbations will be discussed.

Raycraft [Ref. 29] observed remarkable symmetry in temperature and velocity profiles throughout the entire trial. Houck [Ref. 30] also observed the expected asymmetry, after implementing forced ventilation in two locations. In this thesis, these ventilation equations are not removed. The additive velocities were simply set to zero. As seen in Figures 4.3 to 4.8, a marked asymmetry similar to that observed in Houck [Ref. 30], has developed and is readily observed in both temperature and velocity profiles. This is despite the fact that the effects of ventilation have been supposedly negated. After the millions of calculations done by the computer to provide solutions, terms in the ventilation equations which are set to zero have greatly effected the entire field.

Color graphics have greatly enhanced ability to observe phenomena in the temperature fields. Temperatures can be quickly determined using the color legend, as Figures 4.3, 4.5, and 4.7 show. These figures have been printed on a Tektronics 4693D color printer and exhibit excellent clarity and resolution.

Three dimensional vector field representation of the velocity fields, Figures 4.4, 4.6 and 4.8 can only be represented in two dimensional form for this geometry. Results become confusing if three dimensions are shown.

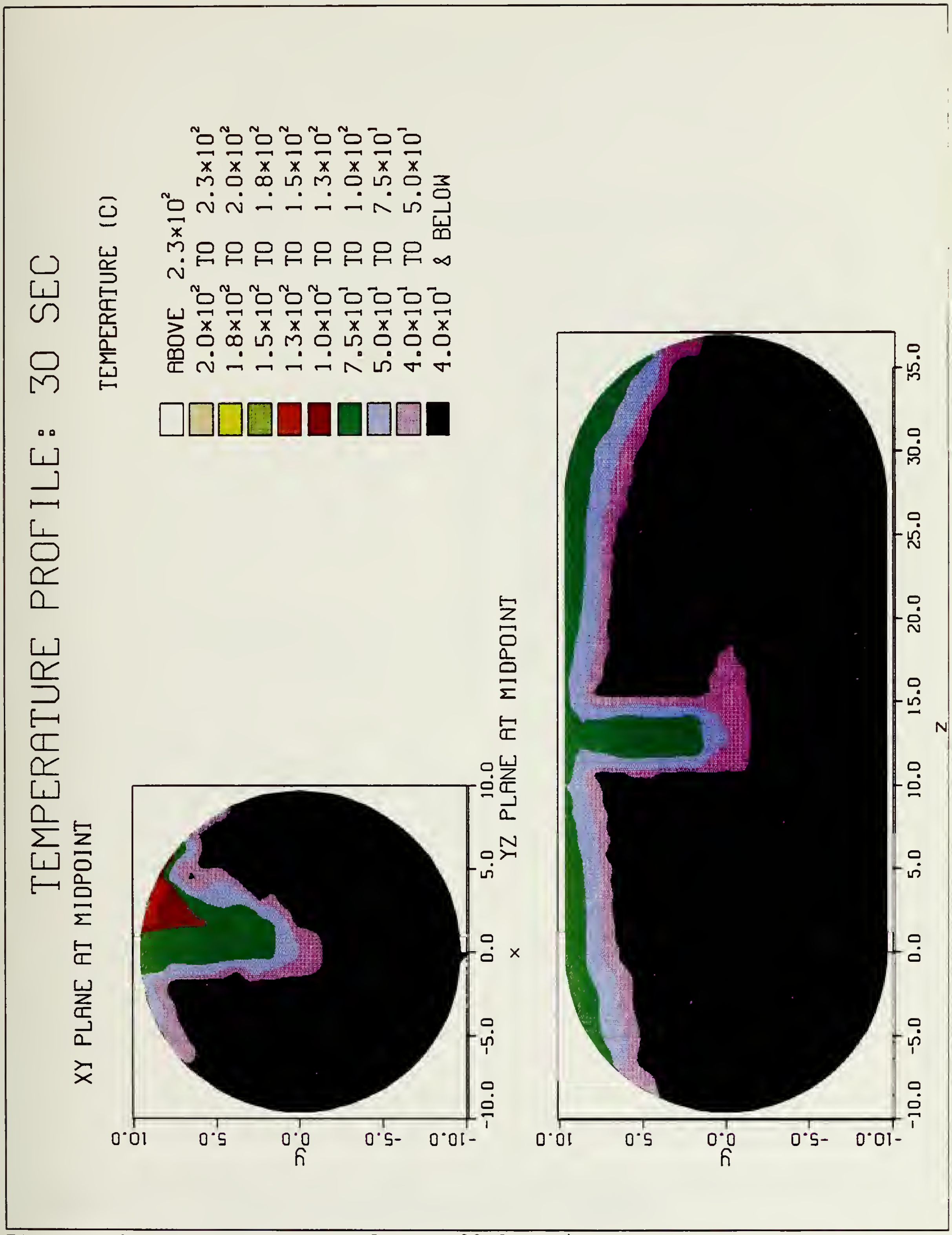

Figure 4.3 Temperature Profiles at 30 Seconds

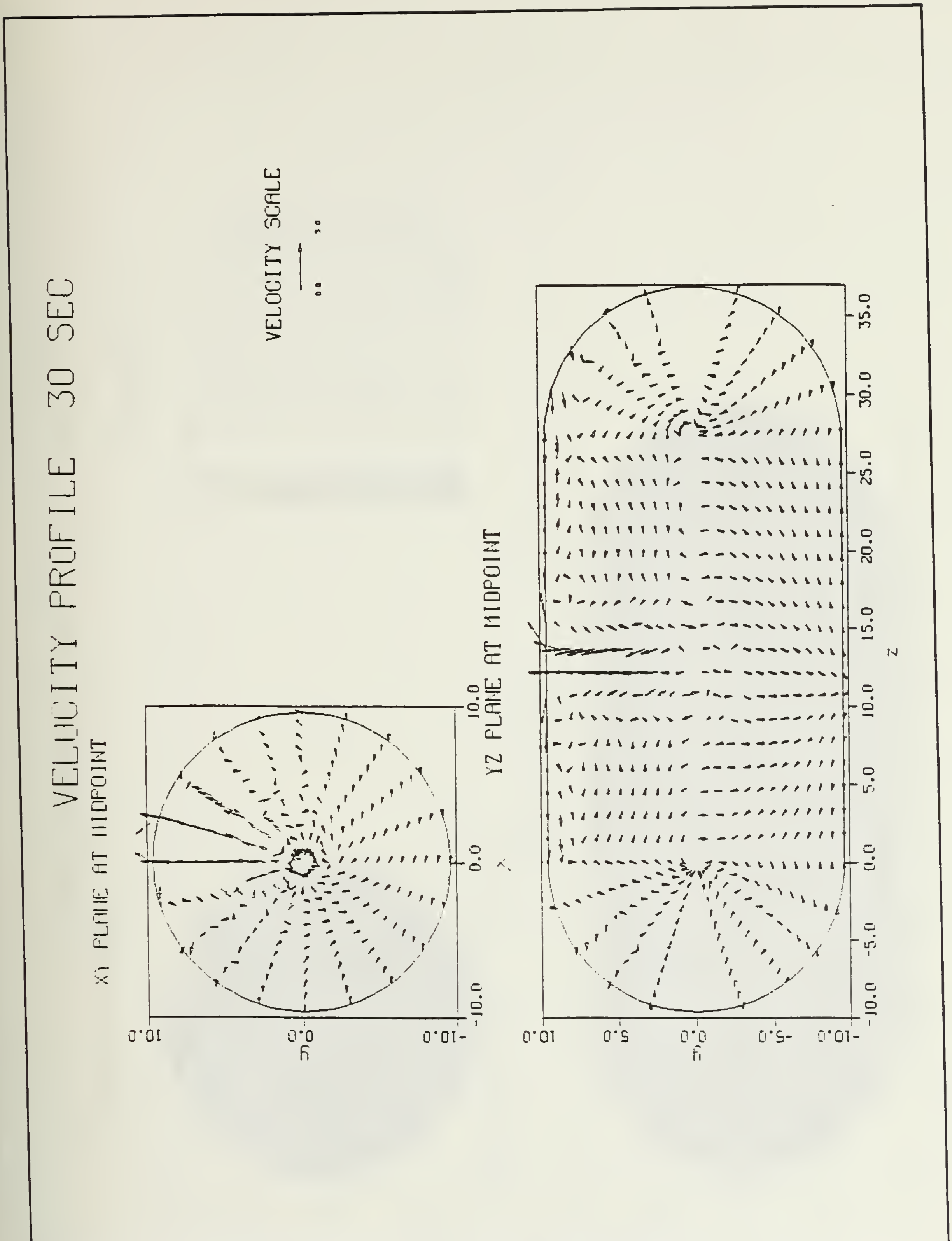

Figure 4.4 Velocity Profile at 30 Seconds

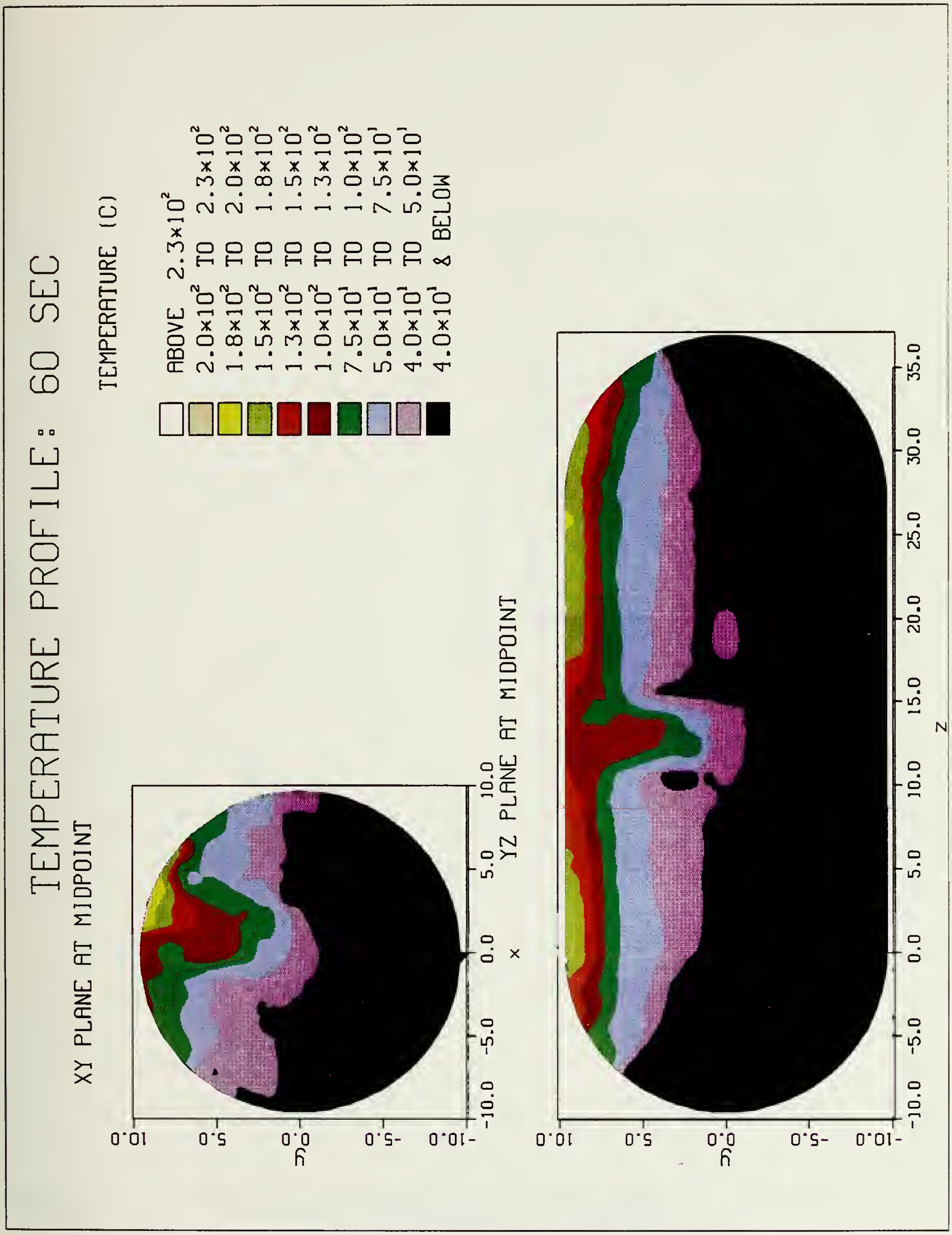

Figure 4.5 Temperature Profiles at 60 Seconds

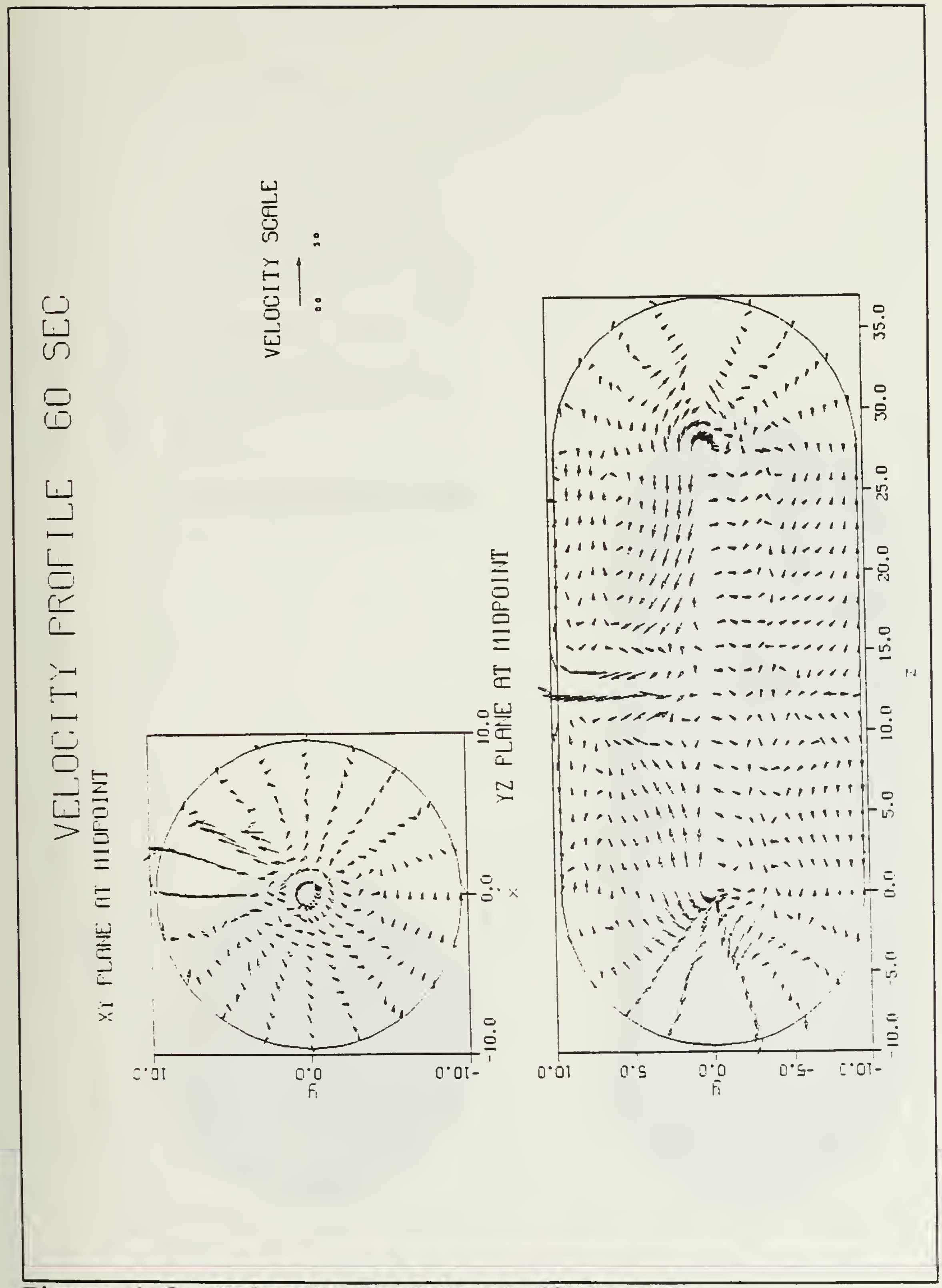

Figure 4.6   Velocity Profile at 60 Seconds

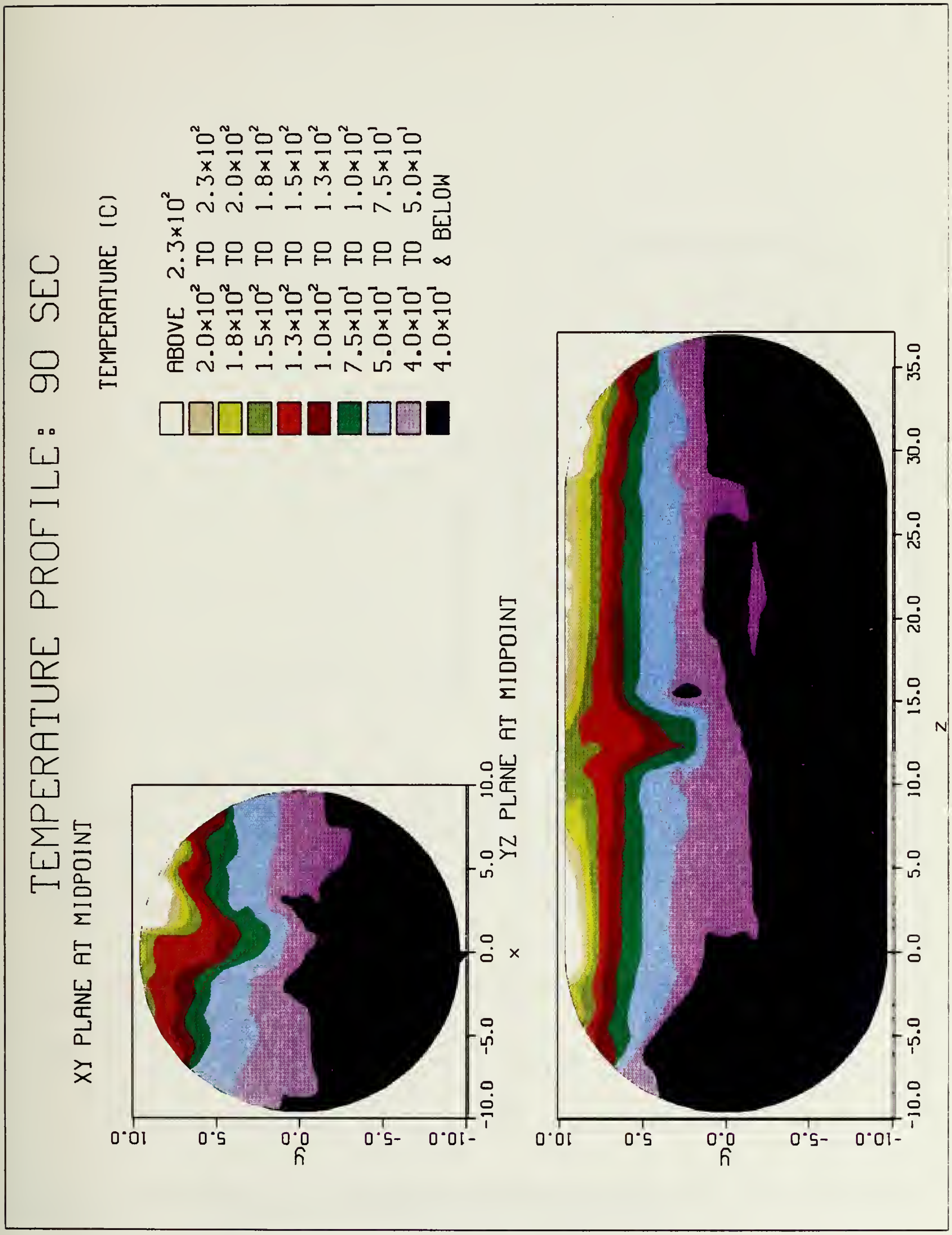

Figure 4.7 Temperature Profiles at 90 Seconds

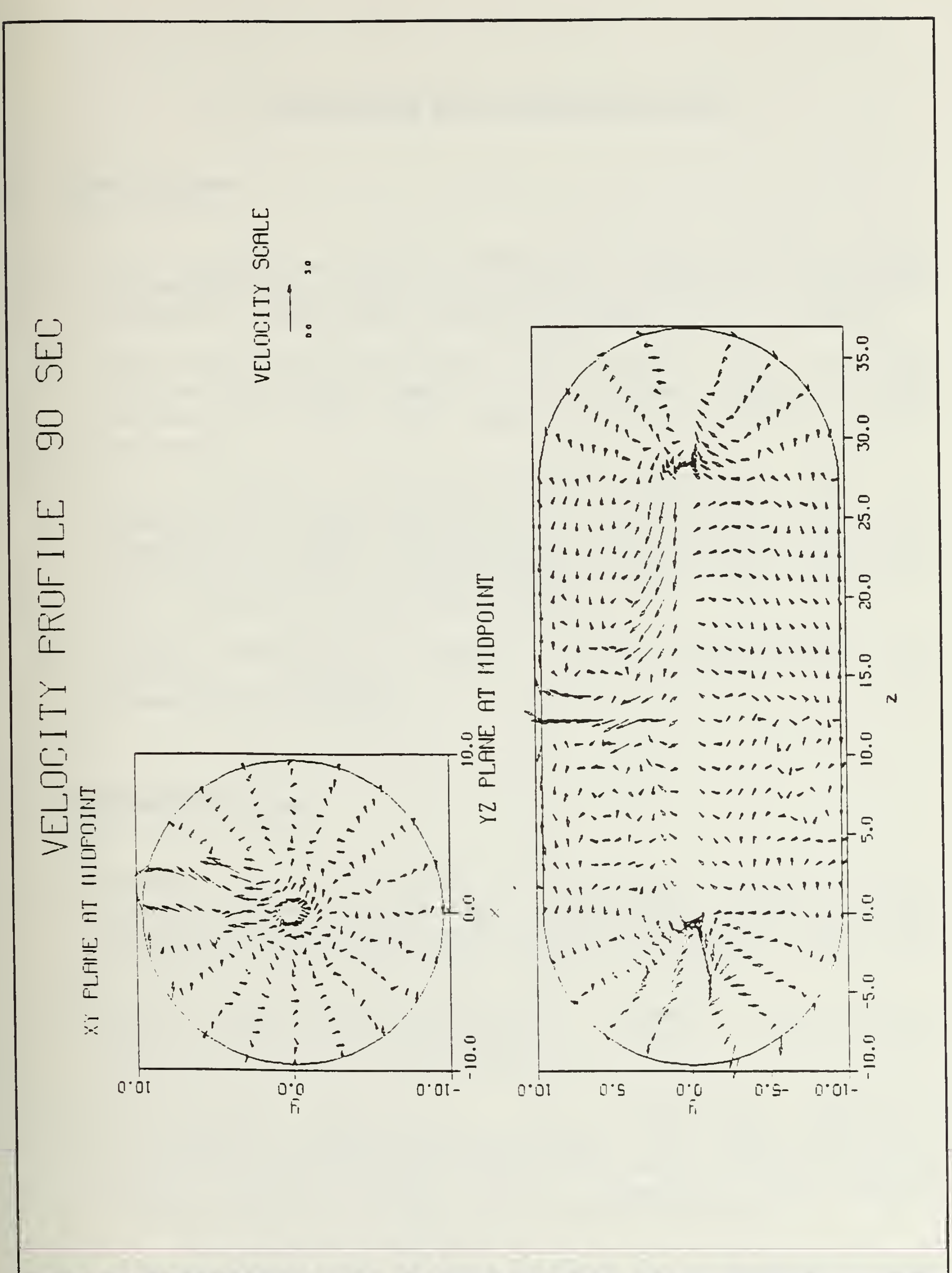

Figure 4.8 Velocity Profile at 90 Seconds

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

1. The acquisition of the VAXSTATION 3100 SPX XRJ19 Model 38 workstation with its blend of numerical speed and graphics clarity has greatly enhanced the research.

2. The ventilation equations incorporated into the model in the previous thesis have a great effect on the entire field even when their output velocities are set to zero.

3. Color graphics have provided an excellent means for presenting temperature profile data. Coupled with the Tektronics 4693 color print, CA-DISSPLA™ Graphics Software provides researchers with an excellent tool for displaying scaler data fields.

4. Three dimensional vector fields are difficult to present, ambiguous, and must be reduced to two dimensional images.

B. RECOMMENDATIONS

1. Removal of the ventilation equations is required to regain symmetry observed in previous research. These equations are effecting the entire field although their additive velocities have been set to zero.

2. More sophisticated physical models need to be formulated and incorporated, such as turbulence, gaseous radiation and combustion.

3. Streakline analysis in three dimensions should be conducted to show the path taken of an individual fluid particle as it leaves the flame area. This method may reveal more of the fluid dynamics than current representations of velocity vector fields.

4. The ultimate goal of this project is to develop a model which can predict behavior of fire in shipboard

situations, for example, changing the geometry to fit machinery spaces and berthing compartments. this will offer designers a valuable tool for the construction of safer ships and submarines.

APPENDIX

```
C     ***********************************************************************
C     **                                                                 **
C     **          THREE-DIMENSIONAL NUMERICAL SIMULATION                 **
C     **        OF A FIRE SPREAD INSIDE A NAVY STORAGE TANK              **
C     **                                                                 **
C     **                    DEVELOPED BY :                               **
C     **               H.Q. YANG AND K.T. YANG                           **
C     **                                                                 **
C     **      DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING           **
C     **                UNIVERSITY OF NOTRE DAME                         **
C     **               NOTRE DAME, INDIANA, 46556                        **
C     **                                                                 **
C     **                      DEC. 1986                                  **
C     **                                                                 **
C     ***********************************************************************
C
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &          ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &        ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &        ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &        ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &        SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &         AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
      DIMENSION VFMXC(579,579),T4WALL(579)
      DATA N,ITLEFT,SORMAX,XTIME,ITMAX/20,400000,0.40,0.0,4/




C *** U0      : REFERENCE VELOCITY (FT/SEC),1 FT/SEC
C *** RHO0    : REFERENCE AIR DENSITY (LBM/FT**3)
C *** H       : REFERENCE LENGTH (FT)
C *** TA      : REFERENCE TEMPERATURE (R)
C *** TINIT   : INITIAL TEMPERATURE (0)
C *** GC      : GRAVITATIONAL CONSTANT
C *** RAITR   : GAS CONSTANT; 53.34
C *** CONST1  : RA*U0**2/GC
C *** CONST3  : INVERSE OF TA
C *** CONST4  : REFERENCE LENGTH (CM)
```

```
C *** CONST6 : REFERENCE VELOCITY (CM/S)
C *** CONSRA : TA**3/(RA*CP*U0*H*H)
C *** NTRWR  : NTREAL/NWRITE*NWRITE
C *** NTRWA  : NTREAL/NWALT*NWALT
C *** HCONV  : HEAT TRANSFER COEFFICIENT ON THE AMBIENT (BTU/H.FT**2K)
C
C
C
C *** RAD,H: RADIUS OF THE CYLINDRICAL AND SPHERICAL SECTIONS
C     CYL  : LENGTH OF THE CYLINDRICAL SECTION OF THE TANK
C *** NI   : TOTAL NUMBER CELLS IN  THETA-DIRECTION
C     NJ   :                        R-DIRECTION
C     NK   :                        Z AND PHI-DIRECTIONS
C     NA   :           FIRST NUMBER  Z-DIRECTION, ALONG THE CYLINDER AXIS
C     NB   :           LAST  NUMBER  Z-DIRECTION, ALONG THE CYLINDER AXIS
C *** HSZ(1,1),HSZ(1,2)  FIRST AND LAST COORDIANTE OF HEAT SOURCE
C                            IN X-DIRECTION  (IN DIMENSIONLESS FORM)
C     HSZ(2,1),HSZ(2,2)  FIRST AND LAST COORDIANTE OF HEAT SOURCE
C                            IN Y-DIRECTION  (IN DIMENSIONLESS FORM)
C     HSZ(3,1),HSZ(3,2)  FIRST AND LAST COORDIANTE OF HEAT SOURCE
C                            IN Z-DIRECTION  (IN DIMENSIONLESS FORM)
C
C *** ICHPB() : STARTING NODAL NUMBER FOR SOLID IN THETA-DIRECTION
C     JCHPB() :                                           R-DIRECTION
C     KCHPB() :                                      Z OR PHI-DIRECTION
C *** NCHPI() : NUMBER OF NODALS  FOR SOLID IN THETA-DIRECTION
C     NCHPJ() :                                    R-DIRECTION
C     NCHPK() :                                  Z,PHI-DIRECTION
C ****************************************************************
      open(21,file='input.dat',status='old')
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C          INPUT DATA                                            &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      write(6,*)'calling input'
      CALL INPUT

C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C          GENERATE GRID SYSTEM                                  &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      write(6,*) 'calling grid'
      CALL GRID

C ################################################################
C ***   READ VIEW FACTOR INVERSE MATRIX                          #
C#################################################################
      open(11,file='view.dat',status='old')
      do 225 iii=1,579
      do 225 jjj=1,579
  225 read(11,*) vfmxc(iii,jjj)
      CLOSE (11)

C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C          INITIALIZE THE WHOLE FIELD                            &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      write(6,*) 'calling init'
      CALL INIT

C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C          START CALCULATION                                     &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

      NT=0
      NTIM=0
      xtime=0.0
  300 CONTINUE

      NT=NT+1
```

```
C ***    NTMAXO HAS THE MEANING AS "NTREAL" WHEN IT IS READ FROM
C        DISK OR TAPE.

      IF(XTIME .GT. TMAX) GO TO 303
      NTREAL=NT+NTMAXO
      TIME=TIME+DTIME
      XTIME=TIME*H/UO
      nxtime=jint(xtime)
      ntwrit=jint(twrite)
      write(6,*) 'time in seconds=',xtime
      write(6,*) 'int time=',nxtime
      write(6,*) 'int time for writing=',ntwrit

C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C     CALCULATE THE TRANSIENT HEAT INPUT                             &
C     NOTE IF 1 IN PARENTHESIS, THE BURN RATE IS CALCULATED          &
C     BY THE PRESSURE CURVE.  IF EQUAL TO TWO, THE BURN RATE         &
C     CURVE IS EITHER GIVEN OR ESTIMATED                             &
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
        write(6,*) 'calling calq'
        CALL CALQ(2)

C ***     START CALCULATION

      ITER=0
      JTERM=0
      JJTERM=0

C     DEFINE THE UPDATED TPD(I,J,K), CPD(I,J,K),RPD(I,J,K)
C     UPD(I,J,K) AND VPD(I,J,K) FOR THE USE OF CALVIS AND SU(I,J,K)

      DO 48 K=1,NKP1
      DO 48 J=1,NJP1
      DO 48 I=1,NIP1
      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
   29 CONTINUE
      JTERM=JTERM+1
  301 CONTINUE


C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C    CALCULATE THE RADIATION HEAT FLUX AT EVERY NRAD TIME STEPS &
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

      NRAD = 2
      IF (MOD(NT,NRAD).NE.0) GOTO 4000
      CALL RADHT(T4WALL,VFMXC)
 4000 CONTINUE

C#######################################################################
C     CALCULATE THE TEMPERATURE                                        #
C#######################################################################
      write(6,*) 'calling calt'
      CALL CALT

C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C     CALCULATE THE SMOKE CONCENTRATION                                &
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      write(6,*) 'calling calc'
      CALL CALC

      DO 2000 J=1,NJP1
```

```
      DO 2000 I=1,NIP1
      DO 2000 K=1,NKP1
      IF(T(I,J,K).LT.TCOOL) T(I,J,K)=TCOOL
 2000 CONTINUE
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     GLOBLE PRESSURE CORRECTION FOR ENCLOSED TANK AIR           %
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       write(6,*) 'calling globe'
       CALL GLOBE

C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     CALCULATE THE TURBULENT VISCOSITY AND CONDUCTIVITY         @
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      write(6,*) 'calling calvis'
      CALL CALVIS

C*****************************************************************
C     CALCULATE THE DENSITY                                      *
C*****************************************************************
      DO 100 J=1,NJP1
      DO 100 I=1,NIP1
      DO 100 K=1,NKP1
      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

CSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
C     CORRECT CONDUCTIVITY OF THE SOLID                          S
CSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
      IF (NCHIP.EQ.0) GOTO 410
      write(6,*) 'calling solcon'
      CALL SOLCON
  410 CONTINUE

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     START PRESSURE CORRECTION ITERATIVE LOOP, IT IS THE MAJOR %
C        PART OF THE ERROR CONTROL ROUTINE                       %
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      ITER=ITER+1

C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     CALCULATE THE VELOCITY U,V,AND W                           @
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      write(6,*) 'calling velocities'
      CALL CALU
c      CALL STRESS
C *** **************
      CALL CALV
c      CALL STRESS
C *** **************
      CALL CALW
      write(6,*) 'wfan(1)=',wfan(1)
c      CALL STRESS
C *** **************

CSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
C     CALCULATE THE PRESSURE AND STRESS                          &
CSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

      write(6,*) 'calling calp'
      CALL CALP
      CALL STRESS

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C      IF SOURCE TERM IS LARGER THAN 10.0, STOP PROGRAM          %
```

```
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      IF (RESORM(ITER).GT.10.0) GOTO 2020


      IF(RESORM(ITER) .LE. SORMAX) GO TO 49
      IF(ITER .EQ. 1) GO TO 302
      ITERM1=ITER-1
      IF(RESORM(ITER) .LE. RESORM(ITERM1)) GO TO 302
      GO TO 304
  302 IF(JTERM .GE. 2) GO TO 37
      SOURCE=RESORM(ITER)
      GO TO 39
   37 IF(RESORM(ITER) .LE. SOURCE) GO TO 38
      GO TO 304
   38 SOURCE=RESORM(ITER)
   39 CONTINUE
c       WRITE(6,95) ITER,RESORM(ITER),SORSUM
   95 FORMAT(53X,'ITER=',I2,2X,'SOURCE=',F9.6,2X,'SORMUP=',F9.6)
      DO 23 K=1,NKP1
      DO 23 J=1,NJP1
      DO 23 I=1,NIP1
      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=0
      IF(ITER .EQ. ITMAX) GO TO 49
      IF(JTERM .EQ. 2) GO TO 35
      IF(ITER .EQ. 4) GO TO 29
   35 CONTINUE
      IF(JTERM .EQ. 3) GO TO 58
      IF(ITER .EQ. 7) GO TO 29
   58 CONTINUE
      JJTERM=0
      GO TO 301
  304 CONTINUE
      JJTERM=JJTERM+1
c       IF(JJTERM .EQ. 1) WRITE(6,95) ITER,RESORM(ITER),SORSUM
      IF(JTERM .EQ. 1) GO TO 41
      IF(JTERM .EQ. 2 .AND. JJTERM .EQ. 1 .AND. ITER .NE. 5) GO TO 41
      GO TO 82
   41 CONTINUE
      DO 40 K=1,NKP1
      DO 40 J=1,NJP1
      DO 40 I=1,NIP1
      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 29
   82 CONTINUE
      DO 43 K=1,NKP1
      DO 43 J=1,NJP1
      DO 43 I=1,NIP1
      T(I,J,K)=TPD(I,J,K)
      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. JJTERM .EQ. 2) GO TO 49
      GO TO 301
   49 CONTINUE

      ITERT=ITERT+ITER
C##################################################################
C     GO TO THE PRESSURE TRACKING SUBROUTINE ,PRINT OUT           #
C     RESULTS IF AT THE RIGHT TIME INTERVAL                       #
C##################################################################
      write(6,*) 'calling ptrack'
      CALL PTRACK
      IF (MOD(ntreal,NWRP).EQ.0) CALL OUT(1)

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C      FIND TEMPERATURES AT THERMOCOUPLE POINTS AND PRINT OUT     %
C       IF AT THE RIGHT TIME INTERVAL                             %
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      if (nthco.eq.0) goto 2422
      CALL TCP
      IF (MOD(NTREAL,NWRP).EQ.0) CALL OUT(2)
 2422 CONTINUE
      IF (MOD(nxtime,ntwrit).EQ.0) CALL OUT(3)
C     IF(NTREAL .EQ. NTREAL/NWRITE*NWRITE) CALL OUT(3)
  305 CONTINUE
      IF((XTIME+DTIME*H/U0) .GE. TMAX) GO TO 277

C *** ************************************************************
C     CALL TLEFT(IT)
C 123 FORMAT(' ITLEFT = ',I10)
C     IT0=IT
C     IF(IT.LT.ITLEFT) CALL OUT(3)
C *** ************************************************************


C ***     RESET THE OLD TIME VALUES  TOD, ROD, UOD, VOD AND POD.

      DO 305 K=1,NKP1
      DO 305 J=1,NJP1
      DO 305 I=1,NIP1
      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 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
C      THIS WRITING IS FOR PLOTTINGS                              ]
C]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
c      IF(NTREAL .NE. NTREAL/NTAPE*NTAPE)GOTO 522
c      IWRITE=10
c      WRITE(9,*)
c     & TIME,NTREAL,T,R,U,V,W,P,CPM,COND,VIS,QRNET,ITERT,QCORRT,PM1,PM2,
c     & H,TA,U0,CONDO,VISO,RHO0,NI,NJ,NK,NIP1,NJP1,NKP1,NIM1,NJM1,NKM1,
c     & XC,YC,ZC,XS,YS,ZS,DXXC,DYYC,DZZC,DXXS,DYYS,DZZS
c      WRITE(6,*) 'THE TIME WHEN THE DATA WAS STORED ON DISK IS:',
c     & XTIME

C *** ************************************************************


  522 CONTINUE
```

```
C *** ***************************************************************
C     CALL TLEFT(IT)
C     IF(IT.LT.ITLEFT) GO TO 166
C *** ***************************************************************
C TIMREM IS USED TO CALCULATE THE CPU TIME REMAINING AT NPS

c     IF (TIMREM(0.).LE.80.) GOTO 166
      do 222 k=1,nkpl
      do 222 i=1,nipl
      do 222 j=1,njpl
      WRITE(9,555) t(i,j,k),u(i,j,k),v(i,j,k),w(i,j,k)

      write(9,555) p(i,j,k),cpm(i,j,k),cond(i,j,k),vis(i,j,k)
 222  continue
      write(9,556) time,qr,qcorrt,pm1,pm2,xxxxx
      write(9,556) h,ta,u0,cond0,vis0,rho0
      write(9,557) ntreal,ni,nj,nk,nip1,njp1,nkp1,nim1,njm1,nkm1,itert
      write(9,556) xc,yc,zc,xs,ys,zs
      write(9,556) dxxc,dyyc,dzzc,dxxs,dyys,dzzs
 555  format(4(3x,e12.4))
 556  format(6(1x,e10.3))
 557  format(11i4)
      REWIND 9
      GO TO 300
  303 CONTINUE
  277 CONTINUE

      WRITE(6,1111)
 1111 FORMAT(2X,'****** THE MAXIMUM TIME HAS BEEN REACHED ******',I8)
c      GO TO 172

C *** ***************************************************************
c  166 IF(NTREAL .NE. NTREAL/NTAPE*NTAPE) then
c234567
      do 223 k=1,nkpl
      do 223 i=1,nipl
      do 223 j=1,njpl
      WRITE(9,555) t(i,j,k),u(i,j,k),v(i,j,k),w(i,j,k)
      write(9,555) p(i,j,k),cpm(i,j,k),cond(i,j,k),vis(i,j,k)
 223  continue
      write(9,556) time,qr,qcorrt,pm1,pm2,xxxxx
      write(9,556) h,ta,u0,cond0,vis0,rho0
      write(9,557) ntreal,ni,nj,nk,nip1,njp1,nkp1,nim1,njm1,nkm1,itert
      write(9,556) xc,yc,zc,xs,ys,zs
      write(9,556) dxxc,dyyc,dzzc,dxxs,dyys,dzzs
      REWIND 9
C *** ***************************************************************

      GOTO 172
 2020 CONTINUE
      WRITE (6,*) '  RESIDUAL MASS IS LARGER THAN 10.0, PROGRAM STOPS'
  172 CONTINUE
      STOP
      END


C ____________________________________________________________________
* *******************************************************************
      SUBROUTINE INPUT
* *******************************************************************
*     THIS SUBROUTINE SETS UP REQUIRED VALUES TO BEGIN THE PROGRAM.
*     VARIABLES ARE:
*             KRUN      =        WHEN EQUAL TO ONE,READ FROM THE
*                                RESTART DISK, ELSE FROM THE JCL
*             NCHIP     =        NUMBER OF SOLID PIECES
*             NWRP      =        NUMBER OF TIME STEPS TO WRITE ON THE
*                                PAPER
```

```
*               NTHCO       =      NUMBER OF THERMOCOUPLES TO PRINT OUT
*               TMAX        =      MAXIMUM TIME ALLOWED (REAL)
*               TWRITE      =      SECONDS IN REAL TIME TO PRINT THE
*                                  P,V,T FIELDS ON PAPER
*               TTAPE       =      TIME INTERVAL TO WRITE ON THE TAPE
*               DTIME       =      TIME STEP (DIMENSIONLESS)
*               HSZ         =      HEAT SOURCE SIZE, USED TO CALCULATE
*                                  THE VOLUME OF THE FIRE CELL
*               ICHPB       =      FIRST SOLID NODE IN THETA DIRECTION
*               JCHPB       =      FIRST SOLID NODE IN R DIRECTION
*               KCHPB       =      FIRST SOLID NODE IN PHI DIRECTION
*               NCHPI       =      NUMBER OF NODES IN THETA DIRECTION
*               NCHPJ       =      NUMBER OF NODES IN R DIRECTION
*               NCHPK       =      NUMBER OF NODES IN PHI DIRECTION
*               CX,CY,CZ    =      THERMOCOUPLE POSITIONS IN THETA,R,PHI
************************************************************************

      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &            ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &            NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &     ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &     ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &     ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &      SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &      AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)


C #1. READ IN DATA TO INDICATE EITHER KRUN=0 OR 1
      READ(21,*) KRUN,NCHIP,NWRP,NTHCO

C #2. READ IN DATA SET 1 - 6 DATA
      READ(21,*) TMAX,TWRITE,TTAPE,DTIME

C #3. READ IN DATA FOR HEAT SOURCE

      READ (21,*)  HSZ(1,1),HSZ(1,2),HSZ(2,1),HSZ(2,2),HSZ(3,1),HSZ(3,2)
      WRITE(6,20) HSZ(1,1),HSZ(1,2),HSZ(2,1),HSZ(2,2),HSZ(3,1),HSZ(3,2)
   20 FORMAT (/,20X,'HEAT SOURCE LOCATION IS IN THE VOLUME (NON-DIME',
     &   'NSIONAL WITH RESPECT TO RADIUS)',
     &      /,5X,'FROM  ',F8.4,'  TO  ',F8.4,'  IN X-DIRECTION',
     &      /,5X,'FROM  ',F8.4,'  TO  ',F8.4,'  IN Y-DIRECTION',
     &      /,5X,'FROM  ',F8.4,'  TO  ',F8.4,'  IN Z-DIRECTION',/)


C #4. READ IN DECK DATA

      IF (NCHIP.EQ.0) GOTO 16
      PRINT *
```

```
      PRINT *,'     THE REGION BOUNDED BY SOLID'
      DO 19 N=1,NCHIP
      READ (21,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
     &            NCHPK(N),TCHP(N),CPS(N),CONS(N),WFAN(N)
      WRITE (6,10) N,ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
     &                 NCHPK(N),TCHP(N),CPS(N),WFAN(N),CONS(N)
   10 FORMAT (2X,'N= ',I2,'  ICHPB= ',I2,' NCHPI= ',I2, '  JCHPB= ',I2,
     &'  NCHPJ=  ',I2, '  KCHPB= ',I2,'  NCHPK= ',I2, ' TCHP= ',F8.5,
     &'  CPS=  ',F8.5,/, '  WFAN =  ',F12.5,'  CONS=  ',F12.5,/)
   19 CONTINUE
   16 CONTINUE
      write(6,*) 'nchip=',nchip

      if(nthco.eq.0) goto 119
C #5.   INPUT THERMOCOUPLE COORDINATE
C       IN TERMS OF X(THETA), Y(RADIUS),Z(PHI)

      PRINT *
      PRINT *, '     THERMOCOUPLE POSITION IN TERMS OF THETA, R, PHI'
      PRINT *
      DO 110 I=1,NTHCO
      READ (21,*) CX(I),CY(I),CZ(I)
      WRITE (6,*) I, CX(I),CY(I),CZ(I)
  110 CONTINUE
  119 continue
      RETURN
      END


C
C ______________________________________________________________________
C *** ******************************************************************
      SUBROUTINE INIT
C *** ******************************************************************
************************************************************************
*     THIS SUBROUTINE INITIALIZES THE FIELD AND CONSTANTS WITH RESPECT *
*     TO INITIAL START OR RESTARTING CAPABILITY.                       *
*     VARIABLES ARE :                                                  *
*           TIME         =        DIMENSIONLESS TIME                   *
*.          U0           =        CHARACTERISTIC VELOCITY (1 FT/SEC)   *
*           H            =        CHARACTERISTIC LENGTH (RADIUS(9.6FT))  *
*           TR           =        TEMP IN DEGREES KELVIN               *
*           TA           =        TEMP IN DEGREES RANKINE              *
*           VISO         =        REFERENCE VISCOSITY (NONDIM)         *
*           VISL         =        MINIMUM VISCOSITY (NONDIM)           *
*           VISMAX       =        MAXIMUM VISCOSITY (NONDIM)           *
*           HR           =        RADIUS IN CM                         *
*           CONDO        =        REFERENCE CONDUCTIVITY               *
*           C0           =        INITIAL SMOKE CONCENTRATION          *
*           NJRA         =        POINT OF RADIATION IN J DIRECTION    *
*                                 LOCATED ON THE INNER SOLID BOUNDARY  *
*           HCONV        =        HEAT TRANSFER COEFFICIENT            *
*           HCOEF        =        DIMENSIONLESS HEAT TRANSFER COEF     *
*           CONST1       =        USED TO NONDIMENSIONALIZE PRESSURE   *
*           RHO0         =        REFERENCE DENSITY                    *
*           GC           =        GRAVITY CONSTANT                     *
*           BUOY         =        BUOYANCY FORCE CONSTANT              *
*           UGRT         =        PERFECT GAS LAW NONDIMENSIONAL CONSTANT*
*           CPO          =        REFERENCE SPECIFIC HEAT              *
*           NWRITE/      =        NONDIMENSIONAL FORMS OF TWRITE AND   *
*           NTAPE                 TTAPE                                *
*     MATRICES OF THE FORM                                             *
*           _OD          =        DIMENSIONLESS PARAMETER AT OLD TIME  *
*           _            =        DIMENSIONLESS PARAMETER              *
*           _PD          =        UPDATED DIMENSIONLESS PARAMETER      *
*     WHERE THE PARAMETERS ARE                                         *
*           U,V,W        =        VELOCITY IN THETA, R , PHI DIRECTION *
*           T,P,C        =        TEMP, PRESSURE, AND SMOKE CONCENTRATION*
*                                                                      *
```

```
*         DU,DV,DZ     =     USED IN PRESSURE CORRECTION SUBROUTINE *
*         PP           =     CORRECTED PRESSURE (P')                *
*         SU           =     SOURCE TERM                            *
*         SP           =     TERM AT P NODAL POINT FOR BOUNDARY     *
*                            CONDITIONS                             *
*         AP           =     COEFICIENT AT NODAL POINT              *
*         AE,AW,AN     =     COEFICIENTS AT PTS EAST,WEST,NORTH,    *
*         AS,AF,AB           SOUTH, FRONT, AND BACK                 *
*         SMP          =     RESIDUAL MASS SUMMATION OF NODAL POINT *
*         SMPP         =     LENGTH SCALE FOR TURBULENCE            *
*         CPM          =     MEAN SPECIFIC HEAT                     *
*         VIS          =     VISCOSITY                              *
*         COND         =     CONDUCTIVITY MATRIX                    *
*         NHSZ         =     WHEN THIS VALUE EQUALS ZERO, THERE IS  *
*                            NO HEAT SOURCE LOCATED AT THE NODE     *
*         NOD          =     IF EQUAL TO ZERO, LIQUID               *
*                            IF EQUAL TO ONE, SOLID                 *
*         _B,_E        =     BEGINNING AND ENDING NODAL POINT FOR   *
*                            THE SOLID IN I,J,K                     *
*         REQ          =     DENSITY AT EQUILIBRIUM                 *
*         NIP1         =     NODAL POINT IN I PLUS 1 (OTHERS SIMILAR)
*         XC,YC,ZC     =     THETA,R,PHI LOCATION OF NODAL POINT OF *
*                            A CENTER CELL                          *
*         DXXC,DYYC    =     LENGTH AROUND THE CENTER CELL          *
*         DZZC                                                      *
*         XS,YS,ZS     =     THETA,R,PHI LOCATION OF NODAL POINT OF *
*                            A STAGGERED CELL                       *
*         DXXS,DYYS    =     LENGTH AROUND THE STAGGERED CELL       *
*         DZZS                                                      *
*         CX,CY,CZ     =     LOCATION OF THERMOCOUPLE IN THETA,R,PHI*
*********************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     & CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &           ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &            NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &        ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &        ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &        ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &        SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &      DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &         AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &      SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
      DATA GRAV/32.17/

C ***     INTRODUCE GIVEN PARAMETERS

      TIME=0.
      TR=TA/1.8
      H=9.6
      VIS0=VIS0/U0/H
```

```
      VISL=VISO
      VISMAX=400.*VISL
      HR=H*30.48
      CONDC=VISO/PRT
      PI=4.*ATAN(1.)
      ALEW = 1.0
      NJRA=15

C  THE HEAT TRANSFER COEFFICIENT IS IN BTU/HR/FT**2/F
      HCONV=15.0
      HCOEF=HCONV/(3600.*CPO*RHOO*UO)
      CC = 0.0


      CONST1=RHOO*UO*UO/(GC*14.696*144.)
      CONST3=1.8/TA
      CONST4=H*30.48
      CONST6=UO*30.48
      NTMAXC=0

      BUOY=GRAV*H/(UO*UO)
      UGRT=UO*UO/(GC*RAIR*TA)
      TCOOL=1.0
      CONSRA=TA*TA*TA/(RHOO*CPO*UO*3600.)*1.714E-9

      WRITE(6,200) TR,CONDO,VISO,CPO,HR,DTIME,HCONV
  200 FORMAT(5X, 'THE REFRENCE TEMPERATURE AND THERMAL PROPERTIES',/,
     &    /,5X, 'T    = ',F10.4,'K,      CONDO = ',E12.6,
     &    /,5X,'VISO = ',E12.6,'    CPO   = ',E12.6,
     &    /,5X,'RADIUS = ',E12.6,'   CM',
     &    /,5X,'DTIME = ',E12.6,
     &    /,5X,'HCONV = ',E12.6,/)

      NWRITE=jint(TWRITE*UO/DTIME/H)
      NTAPE=jint(TTAPE*UO/DTIME/H)
C ***    PRINT OUT INPUT INFORMATION

      WRITE(6,61) (STAR,I=1,90),KRUN,TMAX,TWRITE,TTAPE,NWRP
   61 FORMAT(///,90A1,/,5X,'KRUN  =',I2,/,5X,
     & 'TMAX =',F8.3,' SECONDS',/5X,'TWRITE =',F8.3,
     & ' SECONDS',/,5X,'TTAPE  =',F8.3,' SECONDS',
     & /,5X,' NUMBER INTERVALS OF WRITING ON PAPER  ', I5,/)

C ***    INITIALIZE VARIABLE FIELD

      DO 220 J=1,NJP1
      DO 220 I=1,NIP1
      DO 220 K=1,NKP1
      ROD(I,J,K)=1.
      R(I,J,K)=1.
      RPD(I,J,K)=1.
      UOD(I,J,K)=0.
      U(I,J,K)=0.
      UPD(I,J,K)=0.
      VOD(I,J,K)=0.
      V(I,J,K)=0.
      VPD(I,J,K)=0.
      W(I,J,K)=0.
      WPD(I,J,K)=0.
      WOD(I,J,K)=0.
      POD(I,J,K)=0.
      P(I,J,K)=0.
      PPD(I,J,K)=0.
      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.
      VIS(I,J,K)=VISL
      COND(I,J,K)=CONDO
      CPM(I,J,K)=1.0E0
      TOD(I,J,K)=1.0E0
      T(I,J,K)=TOD(I,J,K)
      TPD(I,J,K)=TOD(I,J,K)
      COD(I,J,K)=C0
      C(I,J,K)=COD(I,J,K)
      CPD(I,J,K)=COD(I,J,K)
      NHSZ(I,J,K)=0
      NOD(I,J,K)=0
  220 CONTINUE



C ***   DETERMINE THE POSITION OF HEAT SOURCE

      DO 300 I=2,NI
      DO 300 J=2,NJ

C CHANGE TO RECTANGULAR COORDINATES
      XX=YC(J)*COS(XC(I))
      YY=YC(J)*SIN(XC(I))

C CHECK TO SEE IF IN HS CONTROL VOLUME, IF SO SET NHSZ=1
      IF (XX.LT.HSZ(1,1).OR.XX.GT.HSZ(1,2)) GOTO 310
      IF (YY.LT.HSZ(2,1).OR.YY.GT.HSZ(2,2)) GOTO 310
      NHSZ(I,J,16)=1
      NHSZ(I,J,17)=1
  315 FORMAT (2X,10(4X,I4,2X,I4))
      GOTO 300
  310 CONTINUE
  300 CONTINUE


C ***     DEFINE THERMAL PROPERTIES OF DECK AND SOLID

      IF (NCHIP.EQ.0) GOTO 410
      DO 402 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 405 I=IB,IE-1
      DO 405 J=JB,JE-1
      DO 405 K=KB,KE-1
      COND(I,J,K)=CONDO*CONS(N)
      CPM(I,J,K)=CP0*CPS(N)
      NOD(I,J,K)=1
  405 CONTINUE
  402 CONTINUE
  410 CONTINUE
```

```
C *** FOR CONTINUING RUN, READ DATA FROM TAPE OR DISK

      IF(KRUN .EQ. 1) GO TO 9997
      GO TO 15
 9997 do 222 k=1,nkp1
      do 222 i=1,nip1
      do 222 j=1,njp1
      read(9,555) t(i,j,k),u(i,j,k),v(i,j,k),w(i,j,k)
      read(9,555) p(i,j,k),cpm(i,j,k),cond(i,j,k),vis(i,j,k)
 222  continue
      read(9,556) time,qr,qcorrt,pm1,pm2,xxxxx
      read(9,556) xxn,xxta,xxu0,xxcond0,xxvis0,xxrho0
      read(9,557) ntreal,ni,nj,nk,nip1,njp1,nkp1,niml,njml,nkml,itert
      read(9,556) xc,yc,zc,xs,ys,zs
      read(9,556) dxxc,dyyc,dzzc,dxxs,dyys,dzzs
 555  format(4(3x,e12.4))
 556  format(6(1x,e10.3))
 557  format(11i4)
      REWIND 9
      WRITE(6,*)NTMAX0
   15 CONTINUE


C ***    DEFINE HEIGHT OF NODE POINTS AND COMPUTE HYDROSTATIC
C        EQUILIBRIUM DENSITY REQ(I,J,K)


      DO 13 K=1,NKP1
      DO 13 I=1,NIP1
      DO 13 J=1,NJP1
      DHY=YC(J)*SIN(XC(I))*SIN(ZC(K))
      HEIGHT(I,J,K)=DHY
 13   CONTINUE
C
      DO 229 J=1,NJP1
      DO 229 I=1,NIP1
      DO 229 K=1,NKP1
      AAAA=-BUOY*UGRT*HEIGHT(I,J,K)
      REQ(I,J,K)=EXP(AAAA)
      IF(KRUN .NE. 0) GO TO 229
      RPD(I,J,K)=REQ(I,J,K)/TPD(I,J,K)
      ROD(I,J,K)=RPD(I,J,K)
      R(I,J,K)=RPD(I,J,K)
  229 CONTINUE

C ***    INITIALIZE U,V,T,R,P FIELD

      DO 210 K=1,NKP1
      DO 210 J=1,NJP1
      DO 210 I=1,NIP1
      T(I,J,K)=TOD(I,J,K)
      C(I,J,K)=COD(I,J,K)
      R(I,J,K)=ROD(I,J,K)
      U(I,J,K)=UOD(I,J,K)
      V(I,J,K)=VOD(I,J,K)
      W(I,J,K)=WOD(I,J,K)
      P(I,J,K)=POD(I,J,K)
  210 CONTINUE

C ***   FOLLOWING IS FOR DETERMINING THE THERMOCOUPLE POSITIONS

      DO 5000 N=1,NTHCO
      DO 5001 I=1,NIP1
      IF (XC(I).LT.CX(N).AND.XC(I-1).GE.CX(N)) GOTO 5002
 5001 CONTINUE
 5002 II=I
```

```
      DO 5003 J=1,NJP1
      IF (YC(J).LT.CY(N).AND.YC(J+1).GE.CY(N)) GOTO 5004
 5003 CONTINUE
 5004 JJ=J

      DO 5005 K=1,NKP1
      IF (ZC(K).LT.CZ(N).AND.ZC(K+1).GE.CZ(N)) GOTO 5006
 5005 CONTINUE
 5006 KK=K
      NTH(N,1)=II
      NTH(N,2)=JJ
      NTH(N,3)=KK
 5000 CONTINUE


      RETURN
      END

C     ______________________________________________________________
C *** **************************************************************
      SUBROUTINE CALVIS

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

      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &           DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &        ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)


C ***    CALCULATE LOCAL SHEAR AND VISCOSITY VIS(I,J,K)
C
C ***    SPECIFY LOCAL TURBULENT LENGTH SCALES  SMPP(I,J,K)

       DO 611 K=2,NK
       KP2=K+2
       KP1=K-1
       KM1=K-1
       KM2=K-2
       DO 611 J=2,NJ
       JP2=J+2
       JP1=J+1
       JM1=J-1
       JM2=J-2
       DO 611 I=2,NI
       IP2=I-2
       IP1=I+1
       IM1=I-1
       IM2=I-2
       IF (I.EQ.2)   IM2=NIM1
       IF (I.EQ.NI) IP2=3
       IF (NOD(I,J,K).EQ.1) GOTO 611
```

```
C        CENTRAL LENGTH OF THE SCALE CONTROL VOLUME

      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)

      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)

      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)

CC    IF (J.EQ.2) DYS=DYS/2.
CC    IF (K.EQ.2) DZB=DZB/2.
      IF (J.NE.NJ) GOTO 101
      JP2=JP1
      DYN=DYN/2.
  101 IF (K.NE.NK) GOTO 102
      KP2=KP1
      DZF=DZF/2.
  102 CONTINUE

C ***     CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T

      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)

      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)

      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)

C ***     CACULATE DV/DX,D2V/DX2,DU/DX,D2U/DX2,DW/DX AND D2W/DX2


      DUDX=(U(IP1,J,K)-U(I,J,K))/DXI
      DUDXW=0.5*(U(IP1,J,K)-U(IM1,J,K))/DXW
      DUDXE=0.5*(U(IP2,J,K)-U(I  ,J,K))/DXE
      D2UDX2=(DUDXE-DUDXW)/DXI


      DVDXW=0.5*(V(I,JP1,K)+V(I,J,K)-V(IM1,JP1,K)-V(IM1,J,K))/DXW
      DVDXE=0.5*(V(IP1,JP1,K)+V(IP1,J,K)-V(I,JP1,K)-V(I,J,K))/DXE
      DVDX=0.5*(DVDXE+DVDXW)
      D2VDX2=   (DVDXE-DVDXW)/DXI


      DWDXW=0.5*(W(I,J,KP1)+W(I,J,K)-W(IM1,J,KP1)-W(IM1,J,K))/DXW
      DWDXE=0.5*(W(IP1,J,KP1)+W(IP1,J,K)-W(I,J,KP1)-W(I,J,K))/DXE
      DWDX=0.5*(DWDXE-DWDXW)
      D2WDX2=   (DWDXE-DWDXW)/DXI


  602 CONTINUE

C ***     CALCULATE DU/DY,D2U/DY2,DV/DY,D2V/DY2,DW/DY AND D2W/DY2


      DVDY=(V(I,JP1,K)-V(I,J,K))/DYJ
      DVDYS=0.5*(V(I,JP1,K)-V(I,JM1,K))/DYS
      DVDYN=0.5*(V(I,JP2,K)-V(I,J  ,K))/DYN
      D2VDY2=(DVDYN-DVDYS)/DYJ
```

```
      DUDYS=0.5*(U(IP1,J,K)+U(I,J,K)-U(IP1,JM1,K)-U(I,JM1,K))/DYS
      DUDYN=0.5*(U(IP1,JP1,K)+U(I,JP1,K)-U(IP1,J,K)-U(I,J,K))/DYN
      DUDY=0.5*(DUDYN+DUDYS)
      D2UDY2=  (DUDYN-DUDYS)/DYJ


      DWDYS=0.5*(W(I,J,KP1)+W(I,J,K)-W(I,JM1,KP1)-W(I,JM1,K))/DYS
      DWDYN=0.5*(W(I,JP1,KP1)+W(I,JP1,K)-W(I,J,KP1)-W(I,J,K))/DYN
      DWDY=0.5*(DWDYN+DWDYS)
      D2WDY2=  (DWDYN-DWDYS)/DYJ

  606 CONTINUE

C ***    CALCULATE DU/DZ,D2U/DZ2,DV/DZ,D2V/DZ2,DW/DZ AND D2W/DZ2


      DWDZ=(W(I,J,KP1)-W(I,J,K))/DZK
      DWDZF=0.5*(W(I,J,KP2)-W(I,J,K  ))/DZF
      DWDZB=0.5*(W(I,J,KP1)-W(I,J,KM1))/DZB
      D2WDZ2=(DWDZF-DWDZB)/DZK


      DVDZB=0.5*(V(I,JP1,K)+V(I,J,K)-V(I,JP1,KM1)-V(I,J,KM1))/DZB
      DVDZF=0.5*(V(I,JP1,KP1)+V(I,J,KP1)-V(I,JP1,K)-V(I,J,K))/DZF
      DVDZ=0.5*(DVDZF+DVDZB)
      D2VDZ2=  (DVDZF-DVDZB)/DZK


      DUDZB=0.5*(U(IP1,J,K)+U(I,J,K)-U(IP1,J,KM1)-W(I,J,KM1))/DZB
      DUDZF=0.5*(U(IP1,J,KP1)+U(I,J,KP1)-U(IP1,J,K)-U(I,J,K))/DZF
      DUDZ=0.5*(DUDZF+DUDZB)
      D2UDZ2=  (DUDZF-DUDZB)/DZK

      DRDX=((R(IP1,J,K)-REQ(IP1,J,K))-(R(IM1,J,K)-REQ(IM1,J,K)))/
     &    (DXE-DXW)
      DRDY=((R(I,JP1,K)-REQ(I,JP1,K))-(R(I,JM1,K)-REQ(I,JM1,K)))/
     &    (DYN-DYS)
      DRDZ=((R(I,J,KP1)-REQ(I,J,KP1))-(R(I,J,KM1)-REQ(I,J,KM1)))/
     &    (DZF-DZB)
      DRDGA=SIN(ZC(K))*(SIN(XC(I))*DRDY+COS(XC(I))*DRDX)
     &     -COS(ZC(K))*DRDZ

C ***    CALCULATE RICHARDSON NUMBER

      STRAIN=DUDY**2+DVDX**2+DWDX**2+DVDZ**2+DWDY**2+DUDZ**2
      DDO2 = SQRT(DUDY*DUDY+DUDX*DUDX+DUDZ*DUDZ+DVDY*DVDY+DVDX*DVDX+
     & DVDZ*DVDZ+DWDX*DWDX+DWDY*DWDY+DWDZ*DWDZ)
      IF(DDO2.EQ.0.)GO TO 600

C ***    CALCULATE TURBULENT LENGTH SCALE SMPP(I,J)

      SMP123=SQRT(((U(IP1,J,K)+U(I,J,K))*0.5)**2+((V(I,JP1,K)-V(I,J,K))*
     &    0.5)**2+((W(I,J,KP1)+W(I,J,K))*0.5)**2)/DDO2
      SMPP12=DDO2 /SQRT(D2UDX2*D2UDX2+D2UDY2*D2UDY2
     & +D2UDZ2*D2UDZ2+D2VDX2*D2VDX2+D2VDY2*D2VDY2+D2VDZ2*D2VDZ2+
     & D2WDZ2*D2WDZ2+D2WDX2*D2WDX2+D2WDY2*D2WDY2)
      SMPP(I,J,K)=CNT*(SMP123+SMPP12)*.5
      RI(I,J,K)=-BUOY*DRDGA/(R(I,J,K)*STRAIN)
      ABRIPR=ABTURB+RI(I,J,K)/PRT
      IF(ABRIPR .LT. 0.) GO TO 600
      IF(ABRIPR .EQ. 0.) GO TO 613
      GO TO 610
  600 VIS(I,J,K)=VISL
      GO TO 611
  613 VIS(I,J,K)=VISMAX
      GO TO 611
  610 VIS(I,J,K)=VISL+R(I,J,K)*SMPP(I,J,K)*SMPP(I,J,K)*SQRT(STRAIN)/
     &         (BTURB*ABRIPR)
```

```
      IF(VIS(I,J,K) .GT. VISMAX) VIS(I,J,K)=VISMAX
  611 CONTINUE


      DO 110 I=1,NIP1
      DO 110 J=1,NJP1
      VIS(I,J,NKP1)=VIS(I,J,NK)
      VIS(I,J,1    )=VIS(I,J,2 )
  110 CONTINUE

      DO 120 J=1,NJP1
      DO 120 K=1,NKP1
      VIS(NIP1,J,K)=VIS(2,J,K)
      VIS(1    ,J,K)=VIS(NI,J,K)
  120 CONTINUE

      DO 130 K=1,NKP1
      DO 130 I=1,NIP1
      VIS(I,NJP1,K)=VIS(I,NJ,K)
      VIS(I,2    ,K)=VIS(I,3 ,K)
      VIS(I,1    ,K)=VIS(I,2 ,K)
  130 CONTINUE

      DO 135 K=1,16
      KK=NKP1-K
      DO 135 I=1,NIP1
      DO 135 J=1,NJP1
      VIS(I,J,KK)=VIS(I,J,K)
  135 CONTINUE

      DO 140 I=1,NIP1
      DO 140 J=1,NJP1
      DO 140 K=1,NKP1
      IF (NOD(I,J,K).EQ.1) GOTO 140
      COND(I,J,K)=VIS(I,J,K)/PRT
  140 CONTINUE

      RETURN
      END



C ______________________________________________________________________
C *** ******************************************************************
      SUBROUTINE CALT
C *** ******************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,UO,H,UGRT,BUOY,
     &  CPO,PRT,CONDO,VISO,RHOO,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
```

```
     &      SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)

C ***     CALCULATE COEFFICIENTS

      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2)  IM2=NIM1
      IF (I.EQ.NI) IP2=3

C       CENTRAL LENGTH OF THE TEMPERTURE CONTROL VOLUME

      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)

      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)

      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)

C ***    SURFACE LENGTH OF THE CONTROL VOLUME

      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)

      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)

      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)

C ***    CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T

      DXEE=XL(IP2,J,K,0,1)
      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)
      DXWW=XL(IM1,J,K,0,1)

      DYNN=YL(I,JP2,K,0,2)
      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)
      DYSS=YL(I,JM1,K,0,2)

      DZFF=ZL(I,J,KP2,0,3)
```

```
      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)
      DZBB=ZL(I,J,KM1,0,3)

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
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW

      GN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      GS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      GE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      GW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
      GF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      GB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)

      CN=GN*V(I,JP1,K)*DZXN
      CS=GS*V(I,J  ,K)*DZXS
      CE=GE*U(IP1,J,K)*DYZE
      CW=GW*U(I  ,J,K)*DYZW
      CF=GF*W(I,J,KP1)*DXYF
      CB=GB*W(I,J,K  )*DXYB


      CONDN=1./((1./COND(I,J,K)*DYJ+1./COND(I,JP1,K)*DYP1)/(DYP1-DYJ))
      CONDS=1./((1./COND(I,J,K)*DYJ+1./COND(I,JM1,K)*DYM1)/(DYM1+DYJ))
      CONDE=1./((1./COND(I,J,K)*DXI+1./COND(IP1,J,K)*DXP1)/(DXP1+DXI))
      CONDW=1./((1./COND(I,J,K)*DXI+1./COND(IM1,J,K)*DXM1)/(DXM1+DXI))
      CONDF=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KP1)*DZP1)/(DZP1+DZK))
      CONDB=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KM1)*DZM1)/(DZM1+DZK))
C
      CONDN1=ZXOYN*CONDN
      CONDS1=ZXOYS*CONDS
      CONDE1=YZOXE*CONDE
      CONDW1=YZOXW*CONDW
      CONDF1=XYOZF*CONDF
      CONDB1=XYOZB*CONDB


      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*DYJ/(DYN*(DYN+DYS ))/8.
      CNM=(ABS(CN)-CN)*DYP1*DYJ/(DYN*(DYN+DYNN))/8.
      CSP=(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*DXI/DXE*CE+CEP+CEM*(1.+DXE/DXEE)-CWM*DXW/DXE
      AW(I,J,K)= .5*DXI/DXW*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW
      AN(I,J,K)=-.5*DYJ/DYN*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN
      AS(I,J,K)= .5*DYJ/DYS*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS
      AF(I,J,K)=-.5*DZK/DZF*CF+CFP+CFM*(1.+DZF/DZFF)-CBM*DZB/DZF
      AB(I,J,K)= .5*DZK/DZB*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB
C
  801 AEE=-CEM*DXE/DXEE
      AEER=AEE*TPD(IP2,J,K)*CPM(IP2,J,K)
  802 CONTINUE

  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*TPD(IM2,J,K)*CPM(IM2,J,K)
  804 CONTINUE

      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*TPD(I,JP2,K)*CPM(I,JP2,K)
  806 CONTINUE

      IF (J.GT.2) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*TPD(I,JM2,K)*CPM(I,JM2,K)
  808 CONTINUE

      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*TPD(I,J,KP2)*CPM(I,J,KP2)
  810 CONTINUE

      IF (K.GT.2) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*TPD(I,J,KM2)*CPM(I,J,KM2)
  812 CONTINUE



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

  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0

  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0

  902 CONTINUE
      IF (NOD(I,JM1,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0
```

```
 903 CONTINUE
     IF (NOD(I,JP1,K).EQ.0) GOTO 904
     ANN=0.0
     ANNR=0.0

 904 CONTINUE
     IF (NOD(I,J,KM1).EQ.0) GOTO 905
     ABB=0.0
     ABBR=0.0

 905 CONTINUE
     IF (NOD(I,J,KP1).EQ.0) GOTO 906
     AFF=0.0
     AFFR=0.0

 906 CONTINUE

C ##############################################################
C ##############################################################

     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)*CPM(I,J,K)
    &          +CONDE1+CONDW1+CONDN1+CONDS1+CONDF1+CONDB1

     AE(I,J,K)=AE(I,J,K)*CPM(IP1,J,K)+CONDE1
     AW(I,J,K)=AW(I,J,K)*CPM(IM1,J,K)+CONDW1
     AN(I,J,K)=AN(I,J,K)*CPM(I,JP1,K)+CONDN1
     AS(I,J,K)=AS(I,J,K)*CPM(I,JM1,K)+CONDS1
     AF(I,J,K)=AF(I,J,K)*CPM(I,J,KP1)+CONDF1
     AB(I,J,K)=AB(I,J,K)*CPM(I,J,KM1)+CONDB1

     SP(I,J,K)=-ROD(I,J,K)*VOLDT*CPM(I,J,K)
     SU(I,J,K)= ROD(I,J,K)*VOLDT*TOD(I,J,K)*CPM(I,J,K)
     SU(I,J,K)=SU(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 ***    RADIUS DIRECTION

     DO 500 I=2,NI
     DO 500 K=2,NK
     SP(I,2,K)=SP(I,2,K)+AS(I,2,K)
CC   SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
CC   SU(I,2,K)=SU(I,2,K)+2.0*AS(I,2,K)*TPD(I,1,K)
     SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
     SU(I,NJ,K)=SU(I,NJ,K)+2.*TPD(I,NJP1,K)*AN(I,NJ,K)
     AS(I,2,K)=0.
     AN(I,NJ,K)=0.
 500 CONTINUE

C ***    CYLIC CONDITIONS

     DO 600 J=2,NJ
     DO 600 K=2,NK
     SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*T(1    ,J,K)
     SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*T(NIP1,J,K)
     AW(2 ,J,K)=0.0
     AE(NI,J,K)=0.0
 600 CONTINUE

C ***       END OF SPHERE

     DO 700 I=2,NI
     DO 700 J=2,NJ
     SP(I,J,2)=SP(I,J,2)-AB(I,J,2)
     SP(I,J,NK)=SP(I,J,NK)-AF(I,J,NK)
     AB(I,J,2)=0.
```

```
      AF(I,J,NK)=0.
 700  CONTINUE

C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS

      DO 300 K=2,NK
      DO 300 J=2,NJ
      DO 300 I=2,NI
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
  300 CONTINUE



C *** VOLUME HEAT SOURCE INPUT

      VOLT=0.0
      DO 113 I=2,NI
      DO 113 J=2,NJ
      DO 113 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 113
      DXI =XL(I  ,J,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DZK =ZL(I,J,K  ,0,0)
      VOL=DXI*DYJ*DZK*H*H*H
      VOLT=VOLT-VOL
  113 CONTINUE

      DO 111 I=2,NI
      DO 111 J=2,NJ
      DO 111 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 111
      DXI =XL(I  ,J,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DZK =ZL(I,J,K  ,0,0)
      QQQ=Q*H/(U0*CP0*RHO0*TA)
      VOL=DXI*DYJ*DZK
      SU(I,J,K)=SU(I,J,K)+VOL*QQQ/VOLT
  111 CONTINUE


C ***   RADIATION INTO THE WALL

C     DO 310 K=3,NKM1
C     DO 310 I=2,NI
C     DXN =XL(I  ,NJRA,K,0,2)
C     DZN =ZL(I,NJRA,K  ,0,2)
C     DZXN=DZN*DXN
C     II=(K-3)*(NI-1)+I-1
C     SU(I,NJRA,K)=SU(I,NJRA,K)-RWALL(II)*DZXN
C 310 CONTINUE

C ***   END OF RADIATION

C ***   SOLVE FOR T
      write(6,*) 'calling trid'
      CALL TRID (2,2,2,NI,NJ,NK,T)

C **** RESET TEMPERATURE AT R=0.0 AND END OF SPHERE

      DO 81 K=1,NKP1
      AVT=0.0
      DO 82 I=2,NI
      AVT=AVT-(T(I,2,K)/NIM1)
   82 CONTINUE
      DO 83 I=1,NIP1
      T(I,1,K)=AVT
```

```
   83 CONTINUE
   81 CONTINUE
C
      DO 74 I=1,NIP1
      DO 74 J=1,NJP1
      T(I,J,1)=T(I,J,2)
      T(I,J,NKP1)=T(I,J,NK)
   74 CONTINUE

C ***    FOR SURFACE HEAT EXCHANGE WITH SURROUNDING

      DO 84 I=2,NI
      DO 84 K=2,NK
      DYJ=YL(I,NJ,K,0,0)
      T(I,NJP1,K)=(2.0*COND(I,NJ,K)*T(I,NJ,K)/DYJ+HCOEF*TINF)/
     &             (HCOEF+2.0*COND(I,NJ,K)/DYJ)
  84  CONTINUE


C ***         FOR CYLIC CONDITION

      DO 80 J=1,NJP1
      DO 80 K=1,NKP1
      T(1,J,K)=T(NI,J,K)
      T(NIP1,J,K)=T(2,J,K)
  80  CONTINUE

      RETURN
      END




C _________________________________________________________________________
C *** *********************************************************************
      SUBROUTINE CALC
C *** *********************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &             NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &        ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &        ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &        SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &         AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR

C ***      CALCULATE COEFFICIENTS

      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
```

```
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2)  IM2=NIM1
      IF (I.EQ.NI) IP2=3

C        CENTRAL LENGTH OF THE SCALE CONTROL VOLUME

      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)

      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)

      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)

C ***    SURFACE LENGTH OF THE CONTROL VOLUME

      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)

      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)

      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)

C ***    CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T

      DXEE=XL(IP2,J,K,0,1)
      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)
      DXWW=XL(IM1,J,K,0,1)

      DYNN=YL(I,JP2,K,0,2)
      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)
      DYSS=YL(I,JM1,K,0,2)

      DZFF=ZL(I,J,KP2,0,3)
      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)
      DZBB=ZL(I,J,KM1,0,3)

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
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW

      GN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      GS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      GE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      GW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
      GF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      GB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)

      CN=GN*V(I,JP1,K)*DZXN
      CS=GS*V(I,J   ,K)*DZXS
      CE=GE*U(IP1,J,K)*DYZE
      CW=GW*U(I   ,J,K)*DYZW
      CF=GF*W(I,J,KP1)*DXYF
      CB=GB*W(I,J,K   )*DXYB


      CONDN=1./((1./COND(I,J,K)*DYJ+1./COND(I,JP1,K)*DYP1)/(DYP1+DYJ))
      CONDS=1./((1./COND(I,J,K)*DYJ+1./COND(I,JM1,K)*DYM1)/(DYM1+DYJ))
      CONDE=1./((1./COND(I,J,K)*DXI+1./COND(IP1,J,K)*DXP1)/(DXP1+DXI))
      CONDW=1./((1./COND(I,J,K)*DXI+1./COND(IM1,J,K)*DXM1)/(DXM1+DXI))
      CONDF=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KP1)*DZP1)/(DZP1+DZK))
      CONDB=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KM1)*DZM1)/(DZM1+DZK))

      CONDN1=ZXOYN*CONDN*ALEW
      CONDS1=ZXOYS*CONDS*ALEW
      CONDE1=YZOXE*CONDE*ALEW
      CONDW1=YZOXW*CONDW*ALEW
      CONDF1=XYOZF*CONDF*ALEW
      CONDB1=XYOZB*CONDB*ALEW


      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*DYJ/(DYN*(DYN+DYS ))/8.
      CNM=(ABS(CN)-CN)*DYP1*DYJ/(DYN*(DYN+DYNN))/8.
      CSP=(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*DXI/DXE*CE+CEP+CEM*(1.-DXE/DXEE)-CWM*DXW/DXE
      AW(I,J,K)= .5*DXI/DXW*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW
      AN(I,J,K)=-.5*DYJ/DYN*CN+CNP+CNM*(1.-DYN/DYNN)+CSM*DYS/DYN
      AS(I,J,K)= .5*DYJ/DYS*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS
      AF(I,J,K)=-.5*DZK/DZF*CF+CFP+CFM*(1.-DZF/DZFF)-CBM*DZB/DZF
      AB(I,J,K)= .5*DZK/DZB*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB


  801 AEE=-CEM*DXE/DXEE
```

```
      AEER=AEE*CPD(IP2,J,K)
  802 CONTINUE

  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*CPD(IM2,J,K)
  804 CONTINUE

      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*CPD(I,JP2,K)
  806 CONTINUE

      IF (J.GT.2) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*CPD(I,JM2,K)
  808 CONTINUE

      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*CPD(I,J,KP2)
  810 CONTINUE

      IF (K.GT.2) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*CPD(I,J,KM2)
  812 CONTINUE



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

  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0

  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0

  902 CONTINUE
      IF (NOD(I,JM1,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0

  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0

  904 CONTINUE
      IF (NOD(I,J,KM1).EQ.0) GOTO 905
      ABB=0.0
```

```
      ABBR=0.0

  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0

  906 CONTINUE

C ##########################################################
C ##########################################################

      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)
     &          +CONDE1+CONDW1+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)= ROD(I,J,K)*VOLDT*TOD(I,J,K)
      SU(I,J,K)=SU(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
C ***    RADIUS DIRECTION

      DO 500 I=2,NI
      DO 500 K=2,NK
CC    SP(I,2,K)=SP(I,2,K)+AS(I,2,K)
      SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
      SU(I,2,K)=SU(I,2,K)+2.0*AS(I,2,K)*CPD(I,1,K)
      SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
      SU(I,NJ,K)=SU(I,NJ,K)+2.*CPD(I,NJP1,K)*AN(I,NJ,K)
      AS(I,2,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE

C ***    CYLIC CONDITIONS

      DO 600 J=2,NJ
      DO 600 K=2,NK
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*C(1    ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*C(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  600 CONTINUE

C ***      END OF SPHERE

      DO 700 I=2,NI
      DO 700 J=2,NJ
      SP(I,J,2)=SP(I,J,2)+AB(I,J,2)
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)
      AB(I,J,2)=0.
      AF(I,J,NK)=0.
  700 CONTINUE



C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS

      DO 300 K=2,NK
```

```
      DO 300 J=2,NJ
      DO 300 I=2,NI
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
  300 CONTINUE



C *** VOLUME MASS SOURCE INPUT

      VOLT=0.0
      DO 113 I=2,NI
      DO 113 J=2,NJ
      DO 113 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 113
      DXI =XL(I  ,J,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DZK =ZL(I,J,K  ,0,0)
      VOL=DXI*DYJ*DZK*H*H*H
      VOLT=VOLT+VOL
  113 CONTINUE

      DO 111 I=2,NI
      DO 111 J=2,NJ
      DO 111 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 111
      DXI =XL(I  ,J,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DZK =ZL(I,J,K  ,0,0)
      QQQ=Q*H/(U0*CP0*RHO0*TA)
      QMS= 1.0
      QMS = QMS*H/(U0*RHO0)
      VOL=DXI*DYJ*DZK
      SU(I,J,K)=SU(I,J,K)+VOL*QMS/VOLT
  111 CONTINUE

C ***   SOLVE FOR C

      CALL TRID (2,2,2,NI,NJM1,NK,C)

C **** RESET CONCENTRATION AT R=0.0 AND END OF SPHERE

      DO 81 K=1,NKP1
      AVT=0.0
      DO 82 I=2,NI
      AVT=AVT+(C(I,2,K)/NIM1)
   82 CONTINUE
      DO 83 I=1,NIP1
      C(I,1,K)=AVT
   83 CONTINUE
   81 CONTINUE

      DO 74 I=1,NIP1
      DO 74 J=1,NJP1
      C(I,J,1)=C(I,J,2)
      C(I,J,NKP1)=C(I,J,NK)
   74 CONTINUE

C ***     FOR SURFACE MASS EXCHANGE WITH SURROUNDING

      DO 84 I=2,NI
      DO 84 K=2,NK
      C(I,NJP1,K)=C(I,NJ,K)
   84 CONTINUE


C ***          FOR CYLIC CONDITION

      DO 80 J=1,NJP1
```

```
      DO 80 K=1,NKP1
      C(1,J,K)=C(NI,J,K)
      C(NIP1,J,K)=C(2,J,K)
   80 CONTINUE

      RETURN
      END


C    ________________________________________________________________________
C    ************************************************************************
      SUBROUTINE CALU
C    ************************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &          ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &        SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &      DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)

C ***    CALCULATE COEFFICIENTS

      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2) IM1=NI
      IF (I.EQ.2) IM2=NIM1
      IF (I.EQ.3) IM2=NI
      IF (I.EQ.NI) IP2=3


C        CENTRAL LENGTH OF THE SCALE CONTROL VOLUME

      DXP1=XL(IP1,J,K,1,0)
      DXI =XL(I  ,J,K,1,0)
```

```
      DXM1=XL(IM1,J,K,1,0)

      DYP1=YL(I,JP1,K,1,0)
      DYJ =YL(I,J  ,K,1,0)
      DYM1=YL(I,JM1,K,1,0)

      DZP1=ZL(I,J,KP1,1,0)
      DZK =ZL(I,J,K  ,1,0)
      DZM1=ZL(I,J,KM1,1,0)

C ***    SURFACE LENGTH OF THE CONTROL VOLUME

      DXN=XL(I,JP1,K,1,2)
      DXS=XL(I,J  ,K,1,2)
      DXF=XL(I,J,KP1,1,3)
      DXB=XL(I,J,K  ,1,3)

      DYF=YL(I,J,KP1,1,3)
      DYB=YL(I,J,K  ,1,3)
      DYE=YL(IP1,J,K,1,1)
      DYW=YL(I  ,J,K,1,1)

      DZE=ZL(IP1,J,K,1,1)
      DZW=ZL(I  ,J,K,1,1)
      DZN=ZL(I,JP1,K,1,2)
      DZS=ZL(I,J  ,K,1,2)

C ***    CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR U

      DXEE=XL(IP2,J,K,1,1)
      DXE =XL(IP1,J,K,1,1)
      DXW =XL(I  ,J,K,1,1)
      DXWW=XL(IM1,J,K,1,1)

      DYNN=YL(I,JP2,K,1,2)
      DYN =YL(I,JP1,K,1,2)
      DYS =YL(I,J  ,K,1,2)
      DYSS=YL(I,JM1,K,1,2)

      DZFF=ZL(I,J,KP2,1,3)
      DZF =ZL(I,J,KP1,1,3)
      DZB =ZL(I,J,K  ,1,3)
      DZBB=ZL(I,J,KM1,1,3)

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
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW


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

```

```
      GNE=SILIN(R(I  ,JP1,K),R(I  ,J,K),DYP1,DYJ)*V(I  ,JP1,K)
      GNW=SILIN(R(IM1,JP1,K),R(IM1,J,K),DYP1,DYJ)*V(IM1,JP1,K)
      GSE=SILIN(R(I  ,JM1,K),R(I  ,J,K),DYM1,DYJ)*V(I  ,J  ,K)
      GSW=SILIN(R(IM1,JM1,K),R(IM1,J,K),DYM1,DYJ)*V(IM1,J  ,K)

      GE =SILIN(R(IP1,J,K),R(I  ,J,K),DXEE,DXE)*U(IP1,J,K)
      GP =SILIN(R(IM1,J,K),R(I  ,J,K),DXW ,DXE)*U(I  ,J,K)
      GW =SILIN(R(IM2,J,K),R(IM1,J,K),DXWW,DXW)*U(IM1,J,K)

      GFE=SILIN(R(I  ,J,KP1),R(I  ,J,K),DZP1,DZK)*W(I  ,J,KP1)
      GFW=SILIN(R(IM1,J,KP1),R(IM1,J,K),DZP1,DZK)*W(IM1,J,KP1)
      GBE=SILIN(R(I  ,J,KM1),R(I  ,J,K),DZM1,DZK)*W(I  ,J,K  )
      GBW=SILIN(R(IM1,J,KM1),R(IM1,J,K),DZM1,DZK)*W(IM1,J,K  )

      CE=0.5*(GE+GP)*DYZE
      CW=0.5*(GP+GW)*DYZW

      CN=SILIN(GNE,GNW,DXE,DXW)*DZXN
      CS=SILIN(GSE,GSW,DXE,DXW)*DZXS

      CF=SILIN(GFE,GFW,DXE,DXW)*DXYF
      CB=SILIN(GBE,GBW,DXE,DXW)*DXYB

      VISE=VIS(I  ,J,K)
      VISW=VIS(IM1,J,K)

       VISN=     (VIS(I  ,JP1,K)+VIS(I  ,J,K)+
     &             VIS(IM1,JP1,K)+VIS(IM1,J,K))/4.0
       VISS=     (VIS(I  ,JM1,K)+VIS(I  ,J,K)+
     &             VIS(IM1,JM1,K)+VIS(IM1,J,K))/4.0

       VISF=     (VIS(I  ,J,KP1)+VIS(I  ,J,K)+
     &             VIS(IM1,J,KP1)+VIS(IM1,J,K))/4.0
       VISB=     (VIS(I  ,J,KM1)+VIS(I  ,J,K)+
     &             VIS(IM1,J,KM1)+VIS(IM1,J,K))/4.0


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


       CEP=(ABS(CE)+CE)*DXE/DXI/16.
       CEM=(ABS(CE)-CE)*DXE/DXP1/16.
       CWP=(ABS(CW)+CW)*DXW/DXM1/16.
       CWM=(ABS(CW)-CW)*DXW/DXI/16.

       CNP=(ABS(CN)+CN)*DYP1*DYJ/(DYN*(DYN+DYS ))/8.
       CNM=(ABS(CN)-CN)*DYP1*DYJ/(DYN*(DYN+DYNN))/8.
       CSP=(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+CEP+CEM*(1.-DXE/DXEE)+CWM*DXW/DXE+VISE1
       AW(I,J,K)= .5*CW+CWM-CWP*(1.-DXW/DXWW)+CEP*DXE/DXW+VISW1


       AN(I,J,K)=-.5*DYJ/DYN*CN+CNP-CNM*(1.-DYN/DYNN)+CSM*DYS/DYN+VISN1
       AS(I,J,K)= .5*DYJ/DYS*CS+CSM+CSP*(1.-DYS/DYSS)+CNP*DYN/DYS+VISS1
       AF(I,J,K)=-.5*DZK/DZF*CF+CFP-CFM*(1.+DZF/DZFF)+CBM*DZB/DZF+VISF1
       AB(I,J,K)= .5*DZK/DZB*CB+CBM+CBP*(1.-DZB/DZBB)+CFP*DZF/DZB+VISB1
```

```
 801 AEE=-CEM*DXE/DXEE
     AEER=AEE*UPD(IP2,J,K)
 802 CONTINUE

 803 AWW=-CWP*DXW/DXWW
     AWWR=AWW*UPD(IM2,J,K)
 804 CONTINUE

     IF (J.LT.NJ) GOTO 805
     ANN=0.
     ANNR=0.
     GOTO 806
 805 ANN=-CNM*DYN/DYNN
     ANNR=ANN*UPD(I,JP2,K)
 806 CONTINUE

     IF (J.GT.2) GOTO 807
     ASS=0.
     ASSR=0.
     GOTO 808
 807 ASS=-CSP*DYS/DYSS
     ASSR=ASS*UPD(I,JM2,K)
 808 CONTINUE

     IF (K.LT.NK) GOTO 809
     AFF=0.
     AFFR=0.
     GOTO 810
 809 AFF=-CFM*DZF/DZFF
     AFFR=AFF*UPD(I,J,KP2)
 810 CONTINUE

     IF (K.GT.2) GOTO 811
     ABB=0.
     ABBR=0.
     GOTO 812
 811 ABB=-CBP*DZB/DZBB
     ABBR=ABB*UPD(I,J,KM2)
 812 CONTINUE


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

 900 CONTINUE
     IF (NOD(IM2,J,K).EQ.0) GOTO 901
     AWW=0.0
     AWWR=0.0

 901 CONTINUE
     IF (NOD(IP1,J,K).EQ.0) GOTO 902
     AEE=0.0
     AEER=0.0

 902 CONTINUE
     IF (NOD(I,JM1,K).EQ.0) GOTO 903
     ASS=0.0
     ASSR=0.0

 903 CONTINUE
     IF (NOD(I,JP1,K).EQ.0) GOTO 904
     ANN=0.0
     ANNR=0.0
 904 CONTINUE
     IF (NOD(I,J,KM1).EQ.0) GOTO 905
```

```
      ABB=0.0
      ABBR=0.0

  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0

  906 CONTINUE
C ######################################################################
C ######################################################################




C ***    SU FROM NORMAL STRESS

      RE=(SIG11(I   ,J,K)-(U(IP1,J,K)-U(I   ,J,K))*VISE/DXE)*DYZE
      RW=(SIG11(IM1,J,K)-(U(I   ,J,K)-U(IM1,J,K))*VISW/DXW)*DYZW
      RN=(SIG12(I,JP1,K)-(U(I,JP1,K)-U(I,J   ,K))*VISN/DYN)*DZXN
      RS=(SIG12(I,J   ,K)-(U(I,J   ,K)-U(I,JM1,K))*VISS/DYS)*DZXS
      RF=(SIG13(I,J,KP1)-(U(I,J,KP1)-U(I,J,K   ))*VISF/DZF)*DXYF
      RB=(SIG13(I,J,K   )-(U(I,J,K   )-U(I,J,KM1))*VISB/DZB)*DXYB

C ***        SU FROM CURVED STRESSES AND ACCELERATIONS

      AVG12=0.5*(SIG12(I,JP1,K)+SIG12(I,J,K))
      AVG13=0.5*(SIG13(I,J,KP1)+SIG13(I,J,K))
      AVG22=SILIN(SIG22(I,J,K),SIG22(IM1,J,K),DXE,DXW)
      AVG33=SILIN(SIG33(I,J,K),SIG33(IM1,J,K),DXE,DXW)

      AU1=U(I,J,K)
      AU2=BILIN(V(I   ,JP1,K),V(I   ,J,K),DYJ,DYJ,
     &          V(IM1,JP1,K),V(IM1,J,K),DYJ,DYJ, DXE,DXW)
      AU3=BILIN(W(I   ,J,KP1),W(I   ,J,K),DZK,DZK,
     &          W(IM1,J,KP1),W(IM1,J,K),DZK,DZK, DXE,DXW)

      AR=SILIN(R(I,J,K),R(IM1,J,K),DXE,DXW)

      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)+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)*DXW+ROD(IM1,J,K)*DXE)/(DXW+DXE)*VOLDT
      SU(I,J,K)= (ROD(I,J,K)*DXW+ROD(IM1,J,K)*DXE)/(DXW+DXE)*VOLDT
     &           *UOD(I,J,K)
      SU(I,J,K)=SU(I,J,K)+DYJ*DZK*(P(IM1,J,K)-P(I,J,K))
     &          -AEER+AWWR+ANNR+ASSR+AFFR+ABBR
     &          -RE-RW+RN-RS+RF-RB+RRY+RRZ-RRX
     &-BUOY*SIN(ZC(K))*((R(I,J,K)-REQ(I,J,K))*DXW*COS(XC(I))+(R(IM1,
     & J,K)-REQ(IM1,J,K))*DXE*COS(XC(IM1)))/(DXW+DXE)*VOL
  100 CONTINUE

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

      DO 500 K=2,NK
      DO 500 I=2,NI
CC    SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
```

```
      SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
      SU(I,2,K)=SU(I,2,K)-2.0*U(I,1,K)*AS(I,2,K)
      SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
      AN(I,NJ,K)=0.
      AS(I,2,K)=0.
  500 CONTINUE

C ***    CYLIC CONDITION

      DO 502 K=2,NK
      DO 502 J=2,NJ
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*U(1    ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*U(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  502 CONTINUE

C ***      FRONT AND BACK WALLS

      DO 600 I=2,NI
      DO 600 J=2,NJ

C ***         SLIP WALLS
      SP(I,J,2)=SP(I,J,2)+AB(I,J,2)
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)

      AF(I,J,NK)=0.
      AB(I,J,2)=0.
 600  CONTINUE



      IF (NCHIP.EQ.0) GOTO 105
C ##############################################################
C ###############################################
C *** MODIFICATION FOR DECK BOUNDARIES

      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE+1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      JBM1=JB-1
      JEP1=JE+1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      KBM1=KB-1
      KEP1=KE+1

      DO 102 J=JB,JE-1
      DO 102 K=KB,KE-1
      AE(IBM1,J,K)=0.0
      AW(IEP1,J,K)=0.0

  102 CONTINUE

      DO 103 I=IB,IE
      DO 103 K=KB,KE-1
      SP(I,JBM1,K)=SP(I,JBM1,K)-AN(I,JBM1,K)
      AN(I,JBM1,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
      SP(I,J,KBM1)=SP(I,J,KBM1)-AF(I,J,KBM1)
      AF(I,J,KBM1)=0.0

      SP(I,J,KE)=SP(I,J,KE)-AB(I,J,KE)
      AB(I,J,KE)=0.0
  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)=-1.0E20
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      SU(I,J,K)=0.
  104 CONTINUE
  101 CONTINUE
  105 CONTINUE

C ##################################################################
C ##################################################################



C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS

      DO 301 K=2,NK
      DO 301 J=2,NJ
      DO 301 I=2,NI
      DYJ=YL(I,J,K,1,0)
      DZK=ZL(I,J,K,1,0)
      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 (2,2,2,NI,NJ,NK,U)

      DO 74 I=2,NIP1
      DO 74 J=2,NJP1
      U(I,J,1)=U(I,J,2)
      U(I,J,NKP1)=U(I,J,NK)
   74 CONTINUE


      DO 79 I=1,NIP1
      DO 79 K=1,NKP1
C     U(I,1,K)=U(I,2,K)
   79 CONTINUE


      IF (NCHIP.EQ.0) GOTO 112
C ########################################################################
C ##############################################
C *** RESET THE VELOCITY INSIDE OF DECK

      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
  112 CONTINUE
C ##################################################
C ##################################################################

      RETURN
      END



C     _______________________________________________________________
C     ****************************************************************
      SUBROUTINE CALV
C     ****************************************************************

      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &          ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &      DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &      SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)


C ***    CALCULATE COEFFICIENTS

      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=3,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
```

```
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2) IM2=NIM1
      IF (I.EQ.NI) IP2=3

C        CENTRAL LENGTH OF THE SCALE CONTROL VOLUME

      DXP1=XL(IP1,J,K,2,0)
      DXI =XL(I  ,J,K,2,0)
      DXM1=XL(IM1,J,K,2,0)

      DYP1=YL(I,JP1,K,2,0)
      DYJ =YL(I,J  ,K,2,0)
      DYM1=YL(I,JM1,K,2,0)

      DZP1=ZL(I,J,KP1,2,0)
      DZK =ZL(I,J,K  ,2,0)
      DZM1=ZL(I,J,KM1,2,0)

C ***     SURFACE LENGTH OF THE CONTROL VOLUME

      DXN=XL(I,JP1,K,2,2)
      DXS=XL(I,J  ,K,2,2)
      DXF=XL(I,J,KP1,2,3)
      DXB=XL(I,J,K  ,2,3)

      DYF=YL(I,J,KP1,2,3)
      DYB=YL(I,J,K  ,2,3)
      DYE=YL(IP1,J,K,2,1)
      DYW=YL(I  ,J,K,2,1)

      DZE=ZL(IP1,J,K,2,1)
      DZW=ZL(I  ,J,K,2,1)
      DZN=ZL(I,JP1,K,2,2)
      DZS=ZL(I,J  ,K,2,2)

C ***     CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME

      DXEE=XL(IP2,J,K,2,1)
      DXE =XL(IP1,J,K,2,1)
      DXW =XL(I  ,J,K,2,1)
      DXWW=XL(IM1,J,K,2,1)

      DYNN=YL(I,JP2,K,2,2)
      DYN =YL(I,JP1,K,2,2)
      DYS =YL(I,J  ,K,2,2)
      DYSS=YL(I,JM1,K,2,2)

      DZFF=ZL(I,J,KP2,2,3)
      DZF =ZL(I,J,KP1,2,3)
      DZB =ZL(I,J,K  ,2,3)
      DZBB=ZL(I,J,KM1,2,3)

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
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW


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


      GEN=SILIN(R(IP1,J  ,K),R(I,J  ,K),DXP1,DXI)*U(IP1,J  ,K)
      GES=SILIN(R(IP1,JM1,K),R(I,JM1,K),DXP1,DXI)*U(IP1,JM1,K)
      GWN=SILIN(R(IM1,J  ,K),R(I,J  ,K),DXM1,DXI)*U(I  ,J  ,K)
      GWS=SILIN(R(IM1,JM1,K),R(I,JM1,K),DXM1,DXI)*U(I  ,JM1,K)

      GN =SILIN(R(I,JP1,K),R(I,J  ,K),DYNN,DYN)*V(I,JP1,K)
      GP =SILIN(R(I,JM1,K),R(I,J  ,K),DYS ,DYN)*V(I,J  ,K)
      GS =SILIN(R(I,JM2,K),R(I,JM1,K),DYSS,DYS)*V(I,JM1,K)

      GFN=SILIN(R(I,J  ,KP1),R(I,J  ,K),DZP1,DZK)*W(I,J  ,KP1)
      GFS=SILIN(R(I,JM1,KP1),R(I,JM1,K),DZP1,DZK)*W(I,JM1,KP1)
      GBN=SILIN(R(I,J  ,KM1),R(I,J  ,K),DZM1,DZK)*W(I,J  ,K  )
      GBS=SILIN(R(I,JM1,KM1),R(I,JM1,K),DZM1,DZK)*W(I,JM1,K  )

      CN=0.5*(GN+GP)*DZXN
      CS=0.5*(GP+GS)*DZXS

      CE=SILIN(GEN,GES,DYN,DYS)*DYZE
      CW=SILIN(GWN,GWS,DYN,DYS)*DYZW

      CF=SILIN(GFN,GFS,DYN,DYS)*DXYF
      CB=SILIN(GBN,GBS,DYN,DYS)*DXYB

      VISN=VIS(I,J  ,K)
      VISS=VIS(I,JM1,K)

      VISE=     (VIS(IP1,J  ,K)+VIS(I,J  ,K)+
     &           VIS(IP1,JM1,K)+VIS(I,JM1,K))/4.0
      VISW=     (VIS(IM1,J  ,K)+VIS(I,J  ,K)+
     &           VIS(IM1,JM1,K)+VIS(I,JM1,K))/4.0

      VISF=     (VIS(I,J  ,KP1)+VIS(I,J  ,K)+
     &           VIS(I,JM1,KP1)+VIS(I,JM1,K))/4.0
      VISB=     (VIS(I,J  ,KM1)+VIS(I,J  ,K)+
     &           VIS(I,JM1,KM1)+VIS(I,JM1,K))/4.0



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

C
      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.
C
      CNP=(ABS(CN)+CN)*DYN/DYJ/16.
      CNM=(ABS(CN)-CN)*DYN/DYP1/16.
      CSP=(ABS(CS)-CS)*DYS/DYM1/16.
      CSM=(ABS(CS)-CS)*DYS/DYJ/16.
C
C
      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.
C
C
      AE(I,J,K)=-.5*DXI/DXE*CE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE+VISE1
      AW(I,J,K)= .5*DXI/DXW*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW+VISW1
C
      AN(I,J,K)=-.5*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN+VISN1
      AS(I,J,K)= .5*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS+VISS1
C
      AF(I,J,K)=-.5*DZK/DZF*CF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF+VISF1
      AB(I,J,K)= .5*DZK/DZB*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB+VISB1
C

  801 AEE=-CEM*DXE/DXEE
      AEER=AEE*VPD(IP2,J,K)
  802 CONTINUE

  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*VPD(IM2,J,K)
  804 CONTINUE

      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*VPD(I,JP2,K)
  806 CONTINUE

      IF (J.GT.3) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*VPD(I,JM2,K)
  808 CONTINUE

      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*VPD(I,J,KP2)
  810 CONTINUE

      IF (K.GT.2) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*VPD(I,J,KM2)
  812 CONTINUE



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

  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0

  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
```

```
      AEE=0.0
      AEER=0.0

  902 CONTINUE
      IF (NOD(I,JM2,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0

  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0

  904 CONTINUE
      IF (NOD(I,J,KM1).EQ.0) GOTO 905
      ABB=0.0
      ABBR=0.0

  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0
  906 CONTINUE

C ##################################################
C ####################################################################


C ***    SU FROM NORMAL STRESS

      RN=(SIG22(I,J   ,K)-(V(I,JP1,K)-V(I,J   ,K))*VISN/DYN)*DZXN
      RS=(SIG22(I,JM1,K)-(V(I,J   ,K)-V(I,JM1,K))*VISS/DYS)*DZXS
      RE=(SIG12(IP1,J,K)-(V(IP1,J,K)-V(I,J   ,K))*VISE/DXE)*DYZE
      RW=(SIG12(I   ,J,K)-(V(I   ,J,K)-V(IM1,J,K))*VISW/DXW)*DYZW
      RF=(SIG23(I,J,KP1)-(V(I,J,KP1)-V(I,J,K   ))*VISF/DZF)*DXYF
      RB=(SIG23(I,J,K   )-(V(I,J,K   )-V(I,J,KM1))*VISB/DZB)*DXYB

C ***         SU FROM CURVED STRESSES AND ACCELERATIONS

      AVG12=0.5*(SIG12(IP1,J,K)+SIG12(I,J,K))
      AVG23=0.5*(SIG23(I,J,KP1)+SIG23(I,J,K))
      AVG11=SILIN(SIG11(I,J,K),SIG11(I,JM1,K),DYN,DYS)
      AVG33=SILIN(SIG33(I,J,K),SIG33(I,JM1,K),DYN,DYS)

      AU2=V(I,J,K)
      AU1=BILIN(U(IP1,J   ,K),U(I,J   ,K),DXI,DXI,
     &          U(IP1,JM1,K),U(I,JM1,K),DXI,DXI, DYN,DYS)
      AU3=BILIN(W(I   ,J,KP1),W(I   ,J,K),DZK,DZK,
     &          W(I,JM1,KP1),W(I,JM1,K),DZK,DZK, DYN,DYS)

      AR=SILIN(R(I,J,K),R(I,JM1,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-DYB)
      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,JM1,K)*DYN)/(DYS+DYN)*VOLDT
      SU(I,J,K)= (ROD(I,J,K)*DYS-ROD(I,JM1,K)*DYN)/(DYS+DYN)*VOLDT
```

```
     &              *VOD(I,J,K)

      SU(I,J,K)=SU(I,J,K)+DZK*DXI*(P(I,JM1,K)-P(I,J,K))
     &          +AEER+AWWR+ANNR+ASSR+AFFR+ABBR
     &          +RE-RW+RN-RS+RF-RB+RRX+RRZ-RRY
     &   -BUOY*((R(I,J,K)-REQ(I,J,K))*DYS+(R(I,JM1,K)
     &       -REQ(I,JM1,K))*DYN)/(DYS+DYN)*VOL*SIN(ZC(K))*SIN(XC(I))
  100 CONTINUE


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

      DO 500 K=2,NK
      DO 500 I=2,NI
CC    SP(I,3,K)=SP(I,3,K)+AS(I,3,K)
      SU(I,3,K)=SU(I,3,K)+AS(I,3,K)*V(I,2,K)
      AS(I,3,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE

C ***    CYLIC CONDITIONS

      DO 502 K=2,NK
      DO 502 J=3,NJ
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*V(1    ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*V(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  502 CONTINUE

C ***    FRONT AND BACK WALL

      DO 600 I=2,NI
      DO 600 J=3,NJ
      JM1=J-1

C ***          SLIP WALLS
      SP(I,J,2)=SP(I,J,2)+AB(I,J,2)
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)

      AF(I,J,NK)=0.
      AB(I,J,2)=0.
  600 CONTINUE



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

      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE+1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      JBM1=JB-1
      JEP1=JE+1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      KBM1=KB-1
      KEP1=KE+1

      DO 102 J=JB,JE
      DO 102 K=KB,KE-1
      SP(IBM1,J,K)=SP(IBM1,J,K)-AE(IBM1,J,K)
```

```
      AE(IBM1,J,K)=0.0

      SP(IE,J,K)=SP(IE,J,K)-AW(IE,J,K)
      AW(IE,J,K)=0.0
  102 CONTINUE

      DO 103 I=IB,IE-1
      DO 103 K=KB,KE-1
      AN(I,JBM1,K)=0.0
      AS(I,JEP1,K)=0.0
  103 CONTINUE

      DO 106 I=IB,IE-1
      DO 106 J=JB,JE
      SP(I,J,KBM1)=SP(I,J,KBM1)-AF(I,J,KBM1)
      AF(I,J,KBM1)=0.0

      SP(I,J,KE)=SP(I,J,KE)-AB(I,J,KE)
      AB(I,J,KE)=0.0
  106 CONTINUE


C ####################################################
C ############################################################
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)=-1.0E20
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      SU(I,J,K)=0.
  104 CONTINUE
  101 CONTINUE
  105 CONTINUE



C ##########################################
C ############################################################
C
C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS

      DO 300 K=2,NK
      DO 300 J=3,NJ
      DO 300 I=2,NI
      DXI=XL(I,J,K,2,0)
      DZK=ZL(I,J,K,2,0)
      DZX=DZK*DXI
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
      DV(I,J,K)=DZX/AP(I,J,K)
  300 CONTINUE


C ***   SOLVE FOR V


      CALL TRID (2,3,2,NI,NJ,NK,V)


      DO 74 I=2,NIP1
      DO 74 J=2,NJP1
      V(I,J,1)=V(I,J,2)
      V(I,J,NKP1)=V(I,J,NK)
   74 CONTINUE
```

```
      DO 79 I=1,NIP1
      DO 79 K=1,NKP1
C     V(I,2,K)=V(I,3,K)
   79 CONTINUE


      IF (NCHIP.EQ.0) GOTO 112
C ##############################################################
C ###############################################
C *** RESET THE VELOCITY INSIDE OF THE DECKS
C
C
      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
      DO 108 K=KB,KE-1
      V(I,J,K)=0.0
  108 CONTINUE
  110 CONTINUE
  112 CONTINUE

C ################################################
C ##############################################################
      RETURN
      END



C      ____________________________________________________________
C      ************************************************************
       SUBROUTINE CALW
C      ************************************************************
       COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &           DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
       COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
       COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
       COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
       COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CPC,PRT,CONDC,VISC,RHOO,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
       COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &            ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
       COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &             NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
       COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
       COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
       COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
       COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &      DU(22,16,32),DV(22,16,32),DW(22,16,32)
       COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &         AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &      SP(22,16,32),SU(22,16,32),RI(22,16,32)
       COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)


C ***         CALCULATE COEFFICIENTS

```

```
      DO 100 K=3,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2) IM2=NIM1
      IF (I.EQ.NI) IP2=3


C        CENTRAL LENGTH OF THE SCALE CONTROL VOLUME

      DXP1=XL(IP1,J,K,3,0)
      DXI =XL(I  ,J,K,3,0)
      DXM1=XL(IM1,J,K,3,0)

      DYP1=YL(I,JP1,K,3,0)
      DYJ =YL(I,J  ,K,3,0)
      DYM1=YL(I,JM1,K,3,0)

      DZP1=ZL(I,J,KP1,3,0)
      DZK =ZL(I,J,K  ,3,0)
      DZM1=ZL(I,J,KM1,3,0)

C ***     SURFACE LENGTH OF THE CONTROL VOLUME

      DXN=XL(I,JP1,K,3,2)
      DXS=XL(I,J  ,K,3,2)
      DXF=XL(I,J,KP1,3,3)
      DXB=XL(I,J,K  ,3,3)

      DYF=YL(I,J,KP1,3,3)
      DYB=YL(I,J,K  ,3,3)
      DYE=YL(IP1,J,K,3,1)
      DYW=YL(I  ,J,K,3,1)

      DZE=ZL(IP1,J,K,3,1)
      DZW=ZL(I  ,J,K,3,1)
      DZN=ZL(I,JP1,K,3,2)
      DZS=ZL(I,J  ,K,3,2)

C ***     CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME

      DXEE=XL(IP2,J,K,3,1)
      DXE =XL(IP1,J,K,3,1)
      DXW =XL(I  ,J,K,3,1)
      DXWW=XL(IM1,J,K,3,1)

      DYNN=YL(I,JP2,K,3,2)
      DYN =YL(I,JP1,K,3,2)
      DYS =YL(I,J  ,K,3,2)
      DYSS=YL(I,JM1,K,3,2)

      DZFF=ZL(I,J,KP2,3,3)
      DZF =ZL(I,J,KP1,3,3)
      DZB =ZL(I,J,K  ,3,3)
      DZBB=ZL(I,J,KM1,3,3)

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
      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,JP1,K  ),R(I,J,K  ),DYP1,DYJ)*V(I,JP1,K  )
      GNB=SILIN(R(I,JP1,KM1),R(I,J,KM1),DYP1,DYJ)*V(I,JP1,KM1)
      GSF=SILIN(R(I,JM1,K  ),R(I,J,K  ),DYM1,DYJ)*V(I,J  ,K  )
      GSB=SILIN(R(I,JM1,KM1),R(I,J,KM1),DYM1,DYJ)*V(I,J  ,KM1)

      GF =SILIN(R(I,J,KP1),R(I,J,K  ),DZFF,DZF)*W(I,J,KP1)
      GP =SILIN(R(I,J,KM1),R(I,J,K  ),DZB ,DZF)*W(I,J,K  )
      GB =SILIN(R(I,J,KM2),R(I,J,KM1),DZBB,DZB)*W(I,J,KM1)

      GEF=SILIN(R(IP1,J,K  ),R(I,J,K  ),DXP1,DXI)*U(IP1,J,K  )
      GEB=SILIN(R(IP1,J,KM1),R(I,J,KM1),DXP1,DXI)*U(IP1,J,KM1)
      GWF=SILIN(R(IM1,J,K  ),R(I,J,K  ),DXM1,DXI)*U(I  ,J,K  )
      GWB=SILIN(R(IM1,J,KM1),R(I,J,KM1),DXM1,DXI)*U(I  ,J,KM1)

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

      CN=SILIN(GNF,GNB,DZF,DZB)*DZXN
      CS=SILIN(GSF,GSB,DZF,DZB)*DZXS

      CE=SILIN(GEF,GEB,DZF,DZB)*DYZE
      CW=SILIN(GWF,GWB,DZF,DZB)*DYZW

      VISF=VIS(I,J,K  )
      VISB=VIS(I,J,KM1)

      VISN=     (VIS(I,JP1,K  )-VIS(I,J,K  )+
     &           VIS(I,JP1,KM1)+VIS(I,J,KM1))/4.0
      VISS=     (VIS(I,JM1,K  )-VIS(I,J,K  )+
     &           VIS(I,JM1,KM1)+VIS(I,J,KM1))/4.0

      VISE=     (VIS(IP1,J,K  )+VIS(I,J,K  )+
     &           VIS(IP1,J,KM1)+VIS(I,J,KM1))/4.0
      VISW=     (VIS(IM1,J,K  )+VIS(I,J,K  )+
     &           VIS(IM1,J,KM1)+VIS(I,J,KM1))/4.0


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

C
```

```
      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.
C
      CNP=(ABS(CN)+CN)*DYP1*DYJ/(DYN*(DYN+DYS ))/8.
      CNM=(ABS(CN)-CN)*DYP1*DYJ/(DYN*(DYN+DYNN))/8.
      CSP=(ABS(CS)+CS)*DYM1*DYJ/(DYS*(DYS+DYSS))/8.
      CSM=(ABS(CS)-CS)*DYM1*DYJ/(DYS*(DYS+DYN ))/8.
C
C
      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.
C
      AE(I,J,K)=-.5*DXI/DXE*CE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE+VISE1
      AW(I,J,K)= .5*DXI/DXW*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW+VISW1
      AN(I,J,K)=-.5*DYJ/DYN*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN+VISN1
      AS(I,J,K)= .5*DYJ/DYS*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS+VISS1
C
      AF(I,J,K)=-.5*CF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF+VISF1
      AB(I,J,K)= .5*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB+VISB1
C

  801 AEE=-CEM*DXE/DXEE
      AEER=AEE*WPD(IP2,J,K)
  802 CONTINUE

  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*WPD(IM2,J,K)
  804 CONTINUE

      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*WPD(I,JP2,K)
  806 CONTINUE

      IF (J.GT.2) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*WPD(I,JM2,K)
  808 CONTINUE

      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*WPD(I,J,KP2)
  810 CONTINUE

      IF (K.GT.3) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*WPD(I,J,KM2)
  812 CONTINUE


C ##################################################################
C ####################################################
```

```
C *** MODIFICATION FOR DECK    BOUNDARIES

  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0

  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0

  902 CONTINUE
      IF (NOD(I,JM1,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0

  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0

  904 CONTINUE
      IF (NOD(I,J,KM2).EQ.0) GOTO 905
      ABB=0.0
      ABBR=0.0

  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0
  906 CONTINUE

C ##################################################
C ######################################################################


C ***    SU FROM NORMAL STRESS

      RF=(SIG33(I,J,K   )-(W(I,J,KP1)-W(I,J,K   ))*VISF/DZF)*DXYF
      RB=(SIG33(I,J,KM1)-(W(I,J,K   )-W(I,J,KM1))*VISB/DZB)*DXYB
      RN=(SIG23(I,JP1,K)-(W(I,JP1,K)-W(I,J   ,K))*VISN/DYN)*DZXN
      RS=(SIG23(I,J   ,K)-(W(I,J   ,K)-W(I,JM1,K))*VISS/DYS)*DZXS
      RE=(SIG13(IP1,J,K)-(W(IP1,J,K)-W(I   ,J,K))*VISE/DXE)*DYZE
      RW=(SIG13(I   ,J,K)-(W(I   ,J,K)-W(IM1,J,K))*VISW/DXW)*DYZW

C ***        SU FROM CURVED STRESSES AND ACCELERATIONS

      AVG23=0.5*(SIG23(I,JP1,K)+SIG23(I,J,K))
      AVG13=0.5*(SIG13(IP1,J,K)+SIG13(I,J,K))
      AVG22=SILIN(SIG22(I,J,K),SIG22(I,J,KM1),DZF,DZB)
      AVG11=SILIN(SIG11(I,J,K),SIG11(I,J,KM1),DZF,DZB)

      AU3=W(I,J,K)
      AU2=BILIN(V(I,JP1,K   ),V(I,J,K   ),DYJ,DYJ,
     &          V(I,JP1,KM1),V(I,J,KM1),DYJ,DYJ, DZF,DZB)
      AU1=BILIN(U(IP1,J,K   ),U(I,J,K   ),DXI,DXI,
     &          U(IP1,J,KM1),U(I,J,KM1),DXI,DXI, DZF,DZB)

      AR=SILIN(R(I,J,K),R(I,J,KM1),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)
```

```
      PRZ=(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,KM1)*DZF)/(DZB+DZF)*VOLDT
      SU(I,J,K)= (ROD(I,J,K)*DZB+ROD(I,J,KM1)*DZF)/(DZB+DZF)*VOLDT
     &              *WOD(I,J,K)
      SU(I,J,K)=SU(I,J,K)+DXI*DYJ*(P(I,J,KM1)-P(I,J,K))
     &          +AEER+AWWR+ANNR+ASSR+AFFR+ABBR
     &          +RE-RW+RN-RS+RF-RB+RRY+RRX-RRZ
     & -BUOY*((R(I,J,K)-REQ(I,J,K))*DZB*COS(ZC(K))+(R(I,J,
     &  KM1)-REQ(I,J,KM1))*DZF*COS(ZC(KM1)))/(DZB+DZF)*VOL*SIN(XC(I))
 100  CONTINUE

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

      DO 500 K=3,NK
      DO 500 I=2,NI
      KM1=K-1
CC    SP(I,2,K)=SP(I,2,K)+AS(I,2,K)
      SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
      SU(I,2,K)=SU(I,2,K)+2.0*W(I,1,K)*AS(I,2,K)
      SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
      AS(I,2,K)=0.
      AN(I,NJ,K)=0.
 500  CONTINUE

C ***   CYLIC CONDITIONS

      DO 502 K=3,NK
      DO 502 J=2,NJ
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*W(1    ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*W(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
 502  CONTINUE

C ***     FRONT AND BACK WALL
      DO 600 I=2,NI
      DO 600 J=2,NJ
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)
      SP(I,J,3 )=SP(I,J,3 )-AB(I,J,3 )
      AF(I,J,NK)=0.
      AB(I,J,3)=0.
  600 CONTINUE


      IF (NCHIP.EQ.0) GOTO 105

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

      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB-NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE-1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      JBM1=JB-1
      JEP1=JE-1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
```

```
      KBM1=KB-1
      KEP1=KE+1


      DO 102 J=JB,JE-1
      DO 102 K=KB,KE
      SP(IBM1,J,K)=SP(IBM1,J,K)-AE(IBM1,J,K)
      SU(IBM1,J,K)=SU(IBM1,J,K)+AE(IBM1,J,K)*WFAN(N)*2.0
      AE(IBM1,J,K)=0.0


      SP(IE,J,K)=SP(IE,J,K)-AW(IE,J,K)
      SU(IE,J,K)=SU(IE,J,K)+AW(IE,J,K)*WFAN(N)*2.0
      AW(IE,J,K)=0.0

  102 CONTINUE

      DO 103 I=IB,IE-1
      DO 103 K=KB,KE
      SP(I,JBM1,K)=SP(I,JBM1,K)-AN(I,JBM1,K)
      SU(I,JBM1,K)=SU(I,JBM1,K)+AN(I,JBM1,K)*WFAN(N)*2.0
      AN(I,JBM1,K)=0.0

      SP(I,JE,K)=SP(I,JE,K)-AS(I,JE,K)
      SU(I,JE,K)=SU(I,JE,K)+AS(I,JE,K)*WFAN(N)*2.0
      AS(I,JE,K)=0.0
  103 CONTINUE

      DO 106 I=IB,IE-1
      DO 106 J=JB,JE-1
      SU(I,J,KBM1)=SU(I,J,KBM1)+AF(I,J,KBM1)*WFAN(N)
      SU(I,J,KEP1)=SU(I,J,KEP1)+AB(I,J,KEP1)*WFAN(N)
      AF(I,J,KBM1)=0.0
      AB(I,J,KEP1)=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
  105 CONTINUE

C ########################################
C ##########################################################


C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS


      DO 301 K=3,NK
      DO 301 J=2,NJ
      DO 301 I=2,NI
      DXI=XL(I,J,K,3,0)
      DYJ=YL(I,J,K,3,0)
      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  (2,2,3,NI,NJ,NK,W)

C
      DO 76 I=1,NI
      DO 76 J=1,NJ
      W(I,J,2)=W(I,J,3)
      W(I,J,NKP1)=W(I,J,NK)
   76 CONTINUE


      IF (NCHIP.EQ.0) GOTO 112
C ###############################################################
C ###############################################################
C *** RESET THE VELOCITY INSIDE OF THE DECKS

      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
  112 CONTINUE

      RETURN
      END


C     ---------------------------------------------------------------
C     ***************************************************************
      SUBROUTINE CALP
C     ***************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &      DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,UO,H,UGRT,BUOY,
     &  CPO,PRT,CONDO,VISO,RHOO,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &              NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     & ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &        ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &        ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &        SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
```

```
C ***    CALCULATE COEFFICIENTS

      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.NI) IP1=2


C        CENTRAL LENGTH OF THE SCALE CONTROL VOLUME

      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)

      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)

      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)

C ***    SURFACE LENGTH OF THE CONTROL VOLUME

      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)

      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)

      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)


C ***    DEFINE 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

      RN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      RS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      RE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      RW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
```

```
      RF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      RB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)

C ***    DU ON VERTICAL WALLS AND DV ON HORIZENTAL WALLS ARE ZERO

      AN(I,J,K)=RN*DZXN*DV(I,JP1,K)
      AS(I,J,K)=RS*DZXS*DV(I,J,K)
      AE(I,J,K)=RE*DYZE*DU(IP1,J,K)
      AW(I,J,K)=RW*DYZW*DU(I,J,K)
      AF(I,J,K)=RF*DXYF*DW(I,J,KP1)
      AB(I,J,K)=RB*DXYB*DW(I,J,K)

      CN=RN*V(I,JP1,K)*DZXN
      CS=RS*V(I,J  ,K)*DZXS
      CE=RE*U(IP1,J,K)*DYZE
      CW=RW*U(I  ,J,K)*DYZW
      CF=RF*W(I,J,KP1)*DXYF
      CB=RB*W(I,J,K  )*DXYB

      SMP(I,J,K)=-(R(I,J,K)-ROD(I,J,K))*VOL/DTIME-CE+CW-CN+CS-CF+CB
C     SMP(I,J,K)=-CE+CW-CN+CS-CF+CB
      SU(I,J,K)=SMP(I,J,K)
      SP(I,J,K)=0.
  100 CONTINUE

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

      DO 500 K=2,NK
      DO 500 I=2,NI
      AS(I,2,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE

C ***    LEFT WALL AND RIGHT WALL

      DO 501 K=2,NK
      DO 501 J=2,NJ
C     AW(2,J,K)=0.
C     AE(NI,J,K)=0.
  501 CONTINUE

C ***     FRONT AND BACK WALL

      DO 502 I=2,NI
      DO 502 J=2,NJ
      AB(I,J,2)=0.0
      AF(I,J,NK)=0.0
 502  CONTINUE



      IF (NCHIP.EQ.0) GOTO 105

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

      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB-NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE-1
      JB=JCHPB(N)
      JE=JB-NCHPJ(N)-1
      JBM1=JB-1
```

```
      JEP1=JE-1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      KBM1=KB-1
      KEP1=KE+1

      DO 102 J=JB,JE-1
      DO 102 K=KB,KE-1
      AE(IBM1,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,JBM1,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,KBM1)=0.0
      AB(I,J,KE)=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-1
      SP(I,J,K)=-1.0E20
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      SU(I,J,K)=0.
  104 CONTINUE
  101 CONTINUE
  105 CONTINUE


C ################################################
C ###############################################################



C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS

      DO 300 J=2,NJ
      DO 300 I=2,NI
      DO 300 K=2,NK
      AP(I,J,K)=AN(I,J,K)+AS(I,J,K)+AE(I,J,K)+AW(I,J,K)-SP(I,J,K)
     &     -AF(I,J,K)+AB(I,J,K)
  300 CONTINUE

C ***    SOLUTION OF FINITE DIFFERENCE EQUATION

      CALL TRID  (2,2,2,NI,NJ,NK,PP)

C *** THIS IS FOR CKECKING


      DO 161 I=1,NIP1
C     WRITE (6,*) I
 949  FORMAT ( '  AW ')
C     WRITE (6,949)
C     WRITE (6,999) ((AW(I,J,K),K=1,NKP1),J=1,NJP1)
 161  CONTINUE
```

```
      DO 160 I=1,NIP1
C     WRITE (6,*) I
 948  FORMAT ( '  AE ')
C     WRITE (6,948)
C     WRITE (6,999) ((AE(I,J,K),K=1,NKP1),J=1,NJP1)
 160  CONTINUE
      DO 170 I=1,NIP1
C     WRITE (6,*) I
 958  FORMAT ( '  AB ')
C     WRITE (6,958)
C     WRITE (6,999) ((AB(I,J,K),K=1,NKP1),J=1,NJP1)
 170  CONTINUE
      DO 180 I=1,NIP1
C     WRITE (6,*) I
 968  FORMAT ( '  AF ')
C     WRITE (6,968)
C     WRITE (6,999) ((AF(I,J,K),K=1,NKP1),J=1,NJP1)
 180  CONTINUE
C     WRITE (6,999) ((SU(I,5,K),K=1,NKP1),I=1,NIP1)
      DO 190 I=1,NIP1
C     WRITE (6,*) I
 978  FORMAT ( '  SU ')
C     WRITE (6,978)
C     WRITE (6,999) ((SU(I,J,K),K=1,NKP1),J=1,NJP1)
 190  CONTINUE
      DO 191 I=1,NIP1
C     WRITE (6,*) I
C     WRITE (6,988)
 988  FORMAT ( '  PP ')
C     WRITE (6,999) ((PP(I,J,K),J=1,NJP1),K=7,7)
 191  CONTINUE
 999  FORMAT (12E10.3)



C ***     CORRECT VELOCITIES AND PRESSURE
C
C ***     CORRECTION FOR VELOCITY U

      DO 600 I=2,NI
      IM1=I-1
      IF (I.EQ.2) IM1=NI
      DO 600 J=2,NJ
      DO 600 K=2,NK
      U(I,J,K)=U(I,J,K)+DU(I,J,K)*(PP(IM1,J,K)-PP(I,J,K))
 600  CONTINUE

C ***     CORRECTION FOR VELOCITY V

      DO 603 J=3,NJ
      JM1=J-1
      DO 603 K=2,NK
      DO 603 I=2,NI
      V(I,J,K)=V(I,J,K)+DV(I,J,K)*(PP(I,JM1,K)-PP(I,J,K))
 603  CONTINUE

C ***    CORRECTION OF VELOCITY W

      DO 604 K=3,NK
      KM1=K-1
      DO 604 I=2,NI
      DO 604 J=2,NJ
      W(I,J,K)=W(I,J,K)+DW(I,J,K)*(PP(I,J,KM1)-PP(I,J,K))
 604  CONTINUE


C ***     CORRECTION FOR PRESSURE P
```

```
      DO 606 J=2,NJ
      DO 606 I=1,NIP1
      DO 606 K=1,NK
      P(I,J,K)=P(I,J,K)+PP(I,J,K)
      PP(I,J,K)=0.
  606 CONTINUE

C *** THIS IS FOR R=0.0 CASE

      DO 75 I=1,NIP1
      DO 75 K=1,NKP1
C     U(I,1,K)=U(I,2,K)
C     W(I,1,K)=W(I,2,K)
C     V(I,2,K)=V(I,3,K)
  75  CONTINUE


C *** MODIFICATION FOR R=0.0
C
      DO 55 K=2,NK
      VY=0.0
      VX=0.0
      VZ=0.0
      DO 50 I=2,NI
      VY=VY+U(I,2,K)*COS(XS(I))
      VX=VX-U(I,2,K)*SIN(XS(I))
   50 CONTINUE

      DO 51 I=2,NI
      VY=VY+V(I,3,K)*SIN(XC(I))
      VX=VX+V(I,3,K)*COS(XC(I))
      VZ=VZ+W(I,2,K)
   51 CONTINUE


C *** FIND THE VELOCITIES AT R=0.0

      DO 52 I=1,NIP1
      U(I,1,K)=(-VX*SIN(XS(I))+VY*COS(XS(I)))/NIM1
      V(I,2,K)= (VX*COS(XC(I))+VY*SIN(XC(I)))/NIM1
      W(I,1,K)=VZ/NIM1
   52 CONTINUE
   55 CONTINUE



C ***  THIS IS FOR THE CYLINDER ONLY (CYLIC CONDITION)

      DO 76 J=1,NJP1
      DO 76 K=1,NKP1
      U(1,J,K)=U(NI,J,K)
      U(NIP1,J,K)=U(2,J,K)
      V(1,J,K)=V(NI,J,K)
      V(NIP1,J,K)=V(2,J,K)
      W(1,J,K)=W(NI,J,K)
      W(NIP1,J,K)=W(2,J,K)
  76  CONTINUE

C ***   THIS FOR SPHERE ONLY

      DO 77 I=1,NIP1
      DO 77 J=1,NJP1
      U(I,J,1)=U(I,J,2)
      V(I,J,1)=V(I,J,2)
      W(I,J,2)=W(I,J,3)
      U(I,J,NKP1)=U(I,J,NK)
      V(I,J,NKP1)=V(I,J,NK)
      W(I,J,NKP1)=W(I,J,NK)
```

```
   77 CONTINUE


      IF (NCHIP.EQ.0) GOTO 116
C ##############################################################
C ##########################################
C *** RESET THE VELOCITY INSIDE OF DECK

      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-1
      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
  116 CONTINUE
C ##########################################
C ################################################################

C ***     RECALCULATE THE ERROR SOURCE AFTER CORRECTIONS OF U, V, P

       SORSUM=0.
       RESORM(ITER)=0.
       DO 700 J=2,NJ
       JP1=J+1
       JM1=J-1
       DO 700 I=2,NI
       IP1=I+1
       IM1=I-1
       DO 700 K=2,NK
       KP1=K+1
       KM1=K-1


C         CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME

       DXP1=XL(IP1,J,K,0,0)
       DXI =XL(I  ,J,K,0,0)
       DXM1=XL(IM1,J,K,0,0)

       DYP1=YL(I,JP1,K,0,0)
       DYJ =YL(I,J  ,K,0,0)
       DYM1=YL(I,JM1,K,0,0)

       DZP1=ZL(I,J,KP1,0,0)
       DZK =ZL(I,J,K  ,0,0)
       DZM1=ZL(I,J,KM1,0,0)
```

```
C ***    SURFACE LENGTH OF THE CONTROL VOLUME

      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)

      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)

      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)


C ***   DEFINE 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



      RN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      RS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      RE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      RW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
      RF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      RB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)

      CN=RN*V(I,JP1,K)*DZXN
      CS=RS*V(I,J  ,K)*DZXS
      CE=RE*U(IP1,J,K)*DYZE
      CW=RW*U(I  ,J,K)*DYZW
      CF=RF*W(I,J,KP1)*DXYF
      CB=RB*W(I,J,K  )*DXYB
C     SMP(I,J,K)=-CE-CW-CN+CS-CF-CB
      SMP(I,J,K)=-(R(I,J,K)-ROD(I,J,K))*VOL/DTIME-CE-CW-CN-CS-CF+CB

C ***    SORSUM IS ACTUAL MASS INCREASE OR DECREASE FROM CONTINUITY
C        EQUATUON , THIS WILL COMPARE TO SOURCE

      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))
  700 CONTINUE
      RETURN
      END



C **********************************************************************
      SUBROUTINE TRID(IST,JST,KST,ISP,JSP,KSP,PHI)
C **********************************************************************
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
```

```
     &    ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &         AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &      SP(22,16,32),SU(22,16,32),RI(22,16,32)
      DIMENSION A(99),B(99),C(99),PHI(22,16,32)

C     GOTO 405
      ISTM1=IST-1
      A(ISTM1)=0.
      C(ISTM1)=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-10) A(I)=0.0
      IF (ABS(B(I)).LE.1.0E-10) B(I)=0.0
      IF (ABS(C(I)).LE.1.0E-10) C(I)=0.0
      IF (ABS(TERM).LE.1.0E-10) 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)
      ISTA=IST+1
      DO 102 II=ISTA,ISP
      I=IST+ISP-II
      IP1=I+1
      PHI(I,J,K)=A(I)*PHI(IP1,J,K)+C(I)
  102 CONTINUE
  100 CONTINUE

      DO 2000 J=JST,JSP
      DO 2000 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2000 CONTINUE


      JSTM1=JST-1
      A(JSTM1)=0.
      C(JSTM1)=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)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,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(J)*A(J-1))
      IF (ABS(A(J)).LE.1.0E-10) A(J)=0.0
      IF (ABS(B(J)).LE.1.0E-10) B(J)=0.0
      IF (ABS(C(J)).LE.1.0E-10) C(J)=0.0
      IF (ABS(TERM).LE.1.0E-10) TERM=0.0
      A(J)=A(J)*TERM
      C(J)=(C(J)+B(J)*C(J-1))*TERM
  201 CONTINUE
      PHI(I,JSP,K)=C(JSP)
      JSTA=JST+1
      DO 202 JJ=JSTA,JSP
      J=JST+JSP-JJ
      JP1=J+1
      PHI(I,J,K)=A(J)*PHI(I,JP1,K)+C(J)
  202 CONTINUE
  200 CONTINUE
```

```
      DO 2001 J=JST,JSP
      DO 2001 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2001 CONTINUE


      KSTM1=KST-1
      A(KSTM1)=0.
      C(KSTM1)=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)*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)-B(K)*A(K-1))
      IF (ABS(A(K)).LE.1.0E-10) A(K)=0.0
      IF (ABS(B(K)).LE.1.0E-10) B(K)=0.0
      IF (ABS(C(K)).LE.1.0E-10) C(K)=0.0
      IF (ABS(TERM).LE.1.0E-10) TERM=0.0
      A(K)=A(K)*TERM
      C(K)=(C(K)+B(K)*C(K-1))*TERM
  301 CONTINUE
      PHI(I,J,KSP)=C(KSP)
      KSTA=KST+1
      DO 302 KK=KSTA,KSP
      K=KST+KSP-KK
      KP1=K+1
      PHI(I,J,K)=A(K)*PHI(I,J,KP1)+C(K)
  302 CONTINUE
  300 CONTINUE

      DO 2002 J=JST,JSP
      DO 2002 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2002 CONTINUE


      GOTO 700

 4405 CONTINUE
  405 KSP1=KSP+1
      B(KSP1)=0.
      C(KSP1)=0.
      DO 600 II=IST,ISP
      I=IST+ISP-II
      DO 600 JJ=JST,JSP
      J=JST+JSP-JJ
      DO 601 KK=KST,KSP
      K=KSP+KST-KK
      KP1=K+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-10) A(K)=0.0
      IF (ABS(B(K)).LE.1.0E-10) B(K)=0.0
      IF (ABS(C(K)).LE.1.0E-10) C(K)=0.0
  601 CONTINUE
      PHI(I,J,KST)=C(KST)
      KSTP1=KST+1
      DO 602 K=KSTP1,KSP
```

```
      PHI(I,J,K)=B(K)*PHI(I,J,K-1)+C(K)
  602 CONTINUE
  600 CONTINUE

      DO 2003 J=JST,JSP
      DO 2003 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2003 CONTINUE


      JSP1=JSP+1
      B(JSP1)=0.
      C(JSP1)=0.
      DO 500 KK=KST,KSP
      K=KST+KSP-KK
      DO 500 II=IST,ISP
      I=IST+ISP-II
      DO 501 JJ=JST,JSP
      J=JSP+JST-JJ
      JP1=J+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,K+1)+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-10) A(J)=0.0
      IF (ABS(B(J)).LE.1.0E-10) B(J)=0.0
      IF (ABS(C(J)).LE.1.0E-10) C(J)=0.0
  501 CONTINUE
      PHI(I,JST,K)=C(JST)
      JSTP1=JST+1
      DO 502 J=JSTP1,JSP
      PHI(I,J,K)=B(J)*PHI(I,J-1,K)+C(J)
  502 CONTINUE
  500 CONTINUE

      DO 2004 J=JST,JSP
      DO 2004 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2004 CONTINUE


      ISP1=ISP+1
      B(ISP1)=0.
      C(ISP1)=0.
      DO 400 JJ=JST,JSP
      J=JST+JSP-JJ
      DO 400 KK=KST,KSP
      K=KST+KSP-KK
      DO 401 II=IST,ISP
      I=ISP+IST-II
      IP1=I+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,K+1)+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-10) A(I)=0.0
      IF (ABS(B(I)).LE.1.0E-10) B(I)=0.0
      IF (ABS(C(I)).LE.1.0E-10) C(I)=0.0
  401 CONTINUE
      PHI(IST,J,K)=C(IST)
```

```
      ISTP1=IST+1
      DO 402 I=ISTP1,ISP
      PHI(I,J,K)=B(I)*PHI(I-1,J,K)+C(I)
  402 CONTINUE
  400 CONTINUE

      DO 2005 J=JST,JSP
      DO 2005 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2005 CONTINUE


  700 CONTINUE
      RETURN
      END

C     ***************************************************************
      BLOCK DATA
C     ***************************************************************

      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      DATA NIP2,NIP1,NI,NIM1/23,22,21,20/
      DATA NJP2,NJP1,NJ,NJM1/17,16,15,14/
      DATA NKP2,NKP1,NK,NKM1/33,32,31,30/
      DATA NAP1,NA,NAM1,NBP1,NB,NBM1/9,8,7,27,26,25/
      DATA U0,TA,PRT,RHO0,CP0,VIS0,NTMAX0/
     &      1.0,555.86,1.0,0.0714,0.24,1.56E-4,0/
      DATA TINF,CNT,ABTURB,BTURB/1.0,0.2,2.0,1.0/
      DATA GC,RAIR/32.17,53.34/
      DATA QCORRT,PM1/1.0,0.9/
      END



C   ******************************************************************
      SUBROUTINE GRID
C   ******************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP

C ***   GENERATION OF GRID

      PI=4.*ATAN(1.)
      DX=1.0/FLOAT(NIM1)
C     DY=1./FLOAT(NJM1-2)
      DY=1./FLOAT(NJM1-1)
      DZ=PI/FLOAT(NKM1-NB+NA-2)


      DO 19 I=1,NIP2
      XS(I)=(I-2)*DX*2.0*PI
   19 CONTINUE

C     XS(1)=-DX*2.0*PI
C     XS(2)=0.0
C     XS(3)=0.01*2.0*PI
C     DO 19 I=4,13
C     XS(I)=(I-3)*DX*2.0*PI
C 19  CONTINUE
```

```
C
C     XS(14)=XS(13)
C     XS(13)=XS(14)-0.01*2.0*PI
C     DO 18 I=15,NIP1
C     XS(I)=XS(14)+(I-14)*DX*2.0*PI
C  18 CONTINUE
C     XS(NIP2)=XS(NIP1)+XS(3)


      YS(1)=0.000
      YS(2)=0.025
C     YS(3)=0.05
      DO 3 J=3,NJ
      YS(J)=(J-2)*DY
    3 CONTINUE
      YS(NJP1)=YS(NJ)
      YS(NJ  )=YS(NJP1)-3./8./12./9.6
      YS(NJP2)=YS(NJP1)+3./8./12./9.6

CC    DO 3 J=4,NJP2
CC    YS(J)=(J-3)*DY
CC 3  CONTINUE
      DO 4 I=1,NIP1
      IP1=I+1
      DXXC(I)=XS(IP1)-XS(I)
    4 CONTINUE

      DXXC(NIP2)=DXXC(NIP1)
      DO 5 I=2,NIP2
      IM1=I-1
      DXXS(I)=.5*(DXXC(I)+DXXC(IM1))
    5 CONTINUE
      DXXS(1)=DXXS(2)

      DO 7 J=1,NJP1
      JP1=J+1
      DYYC(J)=YS(JP1)-YS(J)
    7 CONTINUE

      DYYC(NJP2)=DYYC(NJP1)
      DO 8 J=2,NJP2
      JM1=J-1
      DYYS(J)=.5*(DYYC(J)+DYYC(JM1))
    8 CONTINUE
      DYYS(1)=DYYS(2)

      DO 20 I=1,NIP2
      XC(I)=XS(I)+DXXC(I)/2.0
   20 CONTINUE

      DO 21 J=1,NJP2
      YC(J)=YS(J)+DYYC(J)/2.0
   21 CONTINUE


      DO 9 K=4,NA
      ZS(K)=(K-3)*DZ
    9 CONTINUE

      DO 30 K=NBP1,NK
      ZS(K)=ZS(NA)+(K-NB)*DZ
   30 CONTINUE

      DO 31 K=NAP1,NB
      ZS(K)=PI/2.
   31 CONTINUE

      ZS(1)=0.0
```

```
      ZS(2)=0.05
      ZS(3)=0.10
C     ZS(NKP1)=ZS(NKM1)
C     ZS(NK)=ZS(NKP1)-0.05
C     ZS(NKM1)=ZS(NKP1)-0.10
C     ZS(NKP2)=ZS(NKP1)+0.05

      ZS(NKP2)=ZS(NK)
      ZS(NKP1)=ZS(NKP2)-0.05
      ZS(NK)=ZS(NKP2)-0.10


      DO 10 K=1,NKP1
      IF (K.GE.NA.AND.K.LT.NB) GOTO 10
      KP1=K+1
      DZZC(K)=ZS(KP1)-ZS(K)
   10 CONTINUE

      DO 32 K=NA,NBM1
      DZZC(K)=2.854/(NB-NA)
   32 CONTINUE

      DZZC(NKP2)=DZZC(NKP1)

      DO 11 K=2,NKP2
C     IF (K.EQ.NA.OR.K.EQ.NB) GOTO 11
      KM1=K-1
      DZZS(K)=.5*(DZZC(K)+DZZC(KM1))
   11 CONTINUE

      DZZS(1)=DZZS(2)
      DO 22 K=1,NKP2
      IF (K.GE.NA.AND.K.LT.NB) GOTO 22
      ZC(K)=ZS(K)+DZZC(K)/2.0
   22 CONTINUE

      DO 33 K=NA,NBM1
      ZC(K)=PI/2.
   33 CONTINUE

      IF (YS(1).LT.0.0) YS(1)=0.0
      IF (YC(1).LT.0.0) YC(1)=0.0
      PRINT *
      PRINT *,'      INPUT COORDINATE OF THE TANK IN THE ORDER OF '
      PRINT *,'     I         XS        YS        ZS        XC        YC',
     &    '        ZC        DXXS       DYYS       DZZS       DXXC        '
     & ,  'DYYC        DZZC'
      DO 12 I=1,NKP2
c     WRITE(6,102) I,XS(I),YS(I),ZS(I),XC(I),YC(I),ZC(I),
c    &           DXXS(I),DYYS(I),DZZS(I),DXXC(I),DYYC(I),DZZC(I)
c 102 FORMAT(2X,I4,12(2X,F8.5))
   12 CONTINUE

      RETURN
      END



C     ***********************
      FUNCTION XL(I,J,K,M,N)
C     ***********************
C***************************************************************
C     WHEN M OR N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
```

```
C     WHEN M =  N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     WHOLE CELL                                *
C***************************************************************

      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      X1=XC(I)
      X2=YC(J)
      X3=ZC(K)
      DXL=DXXC(I)
      IF(M.EQ.N) GOTO 100

      IF(M.EQ.1.OR.N.EQ.1) X1=XS(I)
      IF(M.EQ.1.OR.N.EQ.1) DXL=DXXS(I)
      IF(M.EQ.2.OR.N.EQ.2) X2=YS(J)
      IF(M.EQ.3.OR.N.EQ.3) X3=ZS(K)
      GOTO 1000
  100 IF(M.EQ.1) X1=XC(I-1)
      IF(M.EQ.1) DXL=DXXC(I-1)
      IF(M.EQ.2) X2=YC(J-1)
      IF(M.EQ.3) X3=ZC(K-1)
 1000 CONTINUE
      XL=X2*SIN(X3)*DXL
      RETURN
      END


C     ***********************
      FUNCTION YL(I,J,K,M,N)
C     ***********************
C***************************************************************
C     WHEN M OR N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M =  N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     WHOLE CELL                                *
C***************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      X1=XC(I)
      X2=YC(J)
      X3=ZC(K)
      DYL=DYYC(J)
      IF(M.EQ.N) GOTO 100

      IF(M.EQ.2.OR.N.EQ.2) X2=YS(J)
      IF(M.EQ.2.OR.N.EQ.2) DYL=DYYS(J)
      IF(M.EQ.1.OR.N.EQ.1) X1=XS(I)
      IF(M.EQ.3.OR.N.EQ.3) X3=ZS(K)
      GOTO 1000
  100 IF(M.EQ.2) X2=YC(J-1)
      IF(M.EQ.2) DYL=DYYC(J-1)
      IF(M.EQ.1) X1=XC(I-1)
      IF(M.EQ.3) X3=ZC(K-1)
 1000 CONTINUE
      YL=1.00*DYL
      RETURN
```

```
      END


C     ************************
      FUNCTION ZL(I,J,K,M,N)
C     ************************
C*********************************************************************
C     WHEN M OR N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                   HALF CELL (STAGGERED CELL)                   *
C     WHEN M OR N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                   HALF CELL (STAGGERED CELL)                   *
C     WHEN M OR N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                   HALF CELL (STAGGERED CELL)                   *
C     WHEN M =  N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                   WHOLE CELL                                   *
C     WHEN M =  N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                   WHOLE CELL                                   *
C     WHEN M =  N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                   WHOLE CELL                                   *
C*********************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      X1=XC(I)
      X2=YC(J)
      X3=ZC(K)
      DZL=DZZC(K)
      IF(M.EQ.N) GOTO 100

      IF(M.EQ.2.OR.N.EQ.2) X2=YS(J)
      IF(M.EQ.1.OR.N.EQ.1) X1=XS(I)
      IF(M.EQ.3.OR.N.EQ.3) GOTO 200
      GOTO 1000

  200 CONTINUE
      IF (K.EQ.NA.OR.K.EQ.NB) GOTO 2000
      X3=ZS(K)
      DZL=DZZS(K)
      GOTO 1000

  100 IF(M.EQ.3) X3=ZC(K-1)
      IF(M.EQ.3) DZL=DZZC(K-1)
      IF(M.EQ.2) X2=YC(J-1)
      IF(M.EQ.1) X1=XC(I-1)
 1000 CONTINUE
      ZL=X2*DZL
      GOTO 300
 2000 CONTINUE
      DZL1=DZZC(K-1)
      DZL2=DZZC(K)
      IF (K.EQ.NB) DZL1=DZZC(K)
      IF (K.EQ.NB) DZL2=DZZC(K-1)
      ZL=(X2*DZL1+DZL2)/2.
  300 CONTINUE
      RETURN
      END


C     ***************************
      FUNCTION SILIN(V1,V2,D1,D2)
C     ***************************
C     IF (D1.EQ.0.0.AND.D2.EQ.0.0) D1=0.1
C     IF (D1.EQ.0.0.AND.D2.EQ.0.0) D2=0.1
      SILIN=(V1*D2-V2*D1)/(D1+D2)
      RETURN
      END
```

```
C       ************************************************
        FUNCTION BILIN(V1,V2,D1,D2,V3,V4,D3,D4,D5,D6)
C       ************************************************
        V12=(V1*D2+V2*D1)/(D1+D2)
        V34=(V3*D4+V4*D3)/(D3-D4)
        BILIN=(V12*D6+V34*D5)/(D5+D6)
        END


C       ******************
        SUBROUTINE STRESS
C       ******************
        COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &           DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
        COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
        COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
        COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &           ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
        COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
        COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
        COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)


        DO 100 K=2,NK
        KP2=K+2
        KP1=K+1
        KM1=K-1
        KM2=K-2
        DO 100 J=2,NJ
        JP2=J+2
        JP1=J+1
        JM1=J-1
        JM2=J-2
        DO 100 I=2,NI
        IP2=I+2
        IP1=I+1
        IM1=I-1
        IM2=I-2

C         CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME

        DXP1=XL(IP1,J,K,0,0)
        DXI =XL(I  ,J,K,0,0)
        DXM1=XL(IM1,J,K,0,0)

        DYP1=YL(I,JP1,K,0,0)
        DYJ =YL(I,J  ,K,0,0)
        DYM1=YL(I,JM1,K,0,0)

        DZP1=ZL(I,J,KP1,0,0)
        DZK =ZL(I,J,K  ,0,0)
        DZM1=ZL(I,J,KM1,0,0)

C ***     SURFACE LENGTH OF THE CONTROL VOLUME

        DXN=XL(I,JP1,K,0,2)
        DXS=XL(I,J  ,K,0,2)
        DXF=XL(I,J,KP1,0,3)
        DXB=XL(I,J,K  ,0,3)

        DYF=YL(I,J,KP1,0,3)
        DYB=YL(I,J,K  ,0,3)
        DYE=YL(IP1,J,K,0,1)
```

```
      DYW=YL(I  ,J,K,0,1)

      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)

C ***    CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T

      DXEE=XL(IP2,J,K,0,1)
      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)
      DXWW=XL(IM1,J,K,0,1)

      DYNN=YL(I,JP2,K,0,2)
      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)
      DYSS=YL(I,JM1,K,0,2)

      DZFF=ZL(I,J,KP2,0,3)
      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)
      DZBB=ZL(I,J,KM1,0,3)

      UBAR=0.5*(U(IP1,J,K)+U(I,J,K))
      VBAR=0.5*(V(I,JP1,K)+V(I,J,K))
      WBAR=0.5*(W(I,J,KP1)+W(I,J,K))

      DXY=DXI*DYJ
      DYZ=DYJ*DZK
      DZX=DZK*DXI

      SIG11(I,J,K)=2.*VIS(I,J,K)*((U(IP1,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,JP1,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,KP1)-W(I,J,K))/DZK
     &              +UBAR*(DZE-DZW)/DZX
     &              +VBAR*(DZN-DZS)/DYZ)
  100 CONTINUE

      DO 200 K=2,NKP1
      KP2=K-2
      KP1=K-1
      KM1=K-1
      KM2=K-2
      DO 200 J=2,NJP1
      JP2=J-2
      JP1=J-1
      JM1=J-1
      JM2=J-2
      DO 200 I=2,NIP1
      IP2=I-2
      IP1=I-1
      IM1=I-1
      IM2=I-2


C ****    FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CONTROL
C  ,      VOLUME FOR SIG12

C      IF (J.EQ.2) GOTO 300
      DXN=XL(I,J  ,K,1,0)
      DXS=XL(I,JM1,K,1,0)
```

```
      DYE=YL(I  ,J,K,2,0)
      DYW=YL(IM1,J,K,2,0)
      DXI=XL(I  ,J,K,1,2)
      DYJ=YL(I  ,J,K,2,1)

      DYN=YL(I,J  ,K,1,0)
      DYS=YL(I,JM1,K,1,0)
      DXE=XL(I  ,J,K,2,0)
      DXW=XL(IM1,J,K,2,0)

      UBAR=SILIN(U(I,J,K),U(I,JM1,K),DYN,DYS)
      VBAR=SILIN(V(I,J,K),V(IM1,J,K),DXE,DXW)

      VIS12=BILIN(VIS(I  ,J,K),VIS(I  ,JM1,K),DYN,DYS,
     &            VIS(IM1,J,K),VIS(IM1,JM1,K),DYN,DYS, DXE,DXW)

      SIG12(I,J,K)=                 VIS12*((V(I,J,K)-V(IM1,J,K))/DXI
     &                              -VBAR*(DYE-DYW)/(DXI*DYJ))
      SIG12(I,J,K)=SIG12(I,J,K)+VIS12*((U(I,J,K)-U(I,JM1,K))/DYJ
     &                              -UBAR*(DXN-DXS)/(DXI*DYJ))
  300 CONTINUE

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

      DXF=XL(I,J,K  ,1,0)
      DXB=XL(I,J,KM1,1,0)
      DZE=ZL(I  ,J,K,3,0)
      DZW=ZL(IM1,J,K,3,0)
      DXI=XL(I  ,J,K,1,3)
      DZK=ZL(I  ,J,K,3,1)

      DZF=ZL(I,J,K  ,1,0)
      DZB=ZL(I,J,KM1,1,0)
      DXE=XL(I  ,J,K,3,0)
      DXW=XL(IM1,J,K,3,0)

      IF (DZF.EQ.0.0.OR.DZB.EQ.0.0.OR.DZE.EQ.0.0.OR.DZW.EQ.0.0)
     &  WRITE (6,*) I,J,K, DZF,DZB,DZE,DZW
      UBAR=SILIN(U(I,J,K),U(I,J,KM1),DZF,DZB)
      WBAR=SILIN(W(I,J,K),W(IM1,J,K),DXE,DXW)

      VIS13=BILIN(VIS(I  ,J,K),VIS(I  ,J,KM1),DZF,DZB,
     &            VIS(IM1,J,K),VIS(IM1,J,KM1),DZF,DZB, DXE,DXW)

      SIG13(I,J,K)=                 VIS13*((W(I,J,K)-W(IM1,J,K))/DXI
     &                              -WBAR*(DZE-DZW)/(DXI*DZK))
      SIG13(I,J,K)=SIG13(I,J,K)+VIS13*((U(I,J,K)-U(I,J,KM1))/DZK
     &                              -UBAR*(DXF-DXB)/(DXI*DZK))


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

      DZN=ZL(I,J  ,K,3,0)
      DZS=ZL(I,JM1,K,3,0)
      DYF=YL(I,J,K  ,2,0)
      DYB=YL(I,J,KM1,2,0)
      DZK=ZL(I,J,K,3,2)
      DYJ=YL(I,J,K,2,3)

      DYN=YL(I,J  ,K,3,0)
      DYS=YL(I,JM1,K,3,0)
      DZF=ZL(I,J,K  ,2,0)
      DZB=ZL(I,J,KM1,2,0)

      WBAR=SILIN(W(I,J,K),W(I,JM1,K),DYN,DYS)
      VBAR=SILIN(V(I,J,K),V(I,J,KM1),DZF,DZB)
```

```
      VIS23=BILIN(VIS(I  ,J,K),VIS(I,JM1,K  ),DYN,DYS,
     &           VIS(I,J,KM1),VIS(I,JM1,KM1),DYN,DYS, DZF,DZB)

      SIG23(I,J,K)=               VIS23*((V(I,J,K)-V(I,J,KM1))/DZK
     &                            -VBAR*(DYF-DYB)/(DZK*DYJ))
      SIG23(I,J,K)=SIG23(I,J,K)+VIS23*((W(I,J,K)-W(I,JM1,K))/DYJ
     &                            -WBAR*(DZN-DZS)/(DZK*DYJ))

  200 CONTINUE
      DO 110 I=1,NIP1
      DO 110 J=1,NJP1
C     WRITE (6,998) I,J,SIG11(I,J,5),SIG12(I,J,5),SIG13(I,J,5),
C    &                 SIG22(I,J,5),SIG23(I,J,5),SIG33(I,J,5)
  998 FORMAT (2X,I4,1X,I4,6(1X,E11.4))
  110 CONTINUE
      RETURN
      END



C     ______________________________________________________________________
***   **********************************************************************
      SUBROUTINE CALQ(LL)
***   **********************************************************************
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
      COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     & CP0,PRT,COND0,VISC,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &        SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR

C *** IN MANY OF THE FOLLOWING LINES A TEMPORARY CORRECTION FOR
C  *      ADJUSTING QQ TO AGREE WITH THE PRESSURE HAS BEEN APPLIED.

      XTIME=TIME*H/U0

      VOLT=0.0
      DO 113 I=2,NI
      DO 113 J=2,NJ
      DO 113 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 113
      DXI =XL(I  ,J,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DZK =ZL(I,J,K  ,0,0)
      VOL=DXI*DYJ*DZK*H*H*H
      VOLT=VOLT+VOL
  113 CONTINUE

      QRVOL=0.
      DO 70 I=561,579
      QRVOL=QRVOL+RWALL(I)*1./12.*0.2*PI
   70 CONTINUE
C
      QR=QRVOL/VOLT*U0*CP0*RHO0*TA/H

        IF (XTIME.LT.23.1) THEN
          PCURVE=9.789522E-5*XTIME**2-2.388310E-6*XTIME**3-
     &        REQ(10,9,16)
          DPDT  =9.789522E-5*XTIME*2-2.388310E-6*XTIME**2*3
        ELSE
      PCURVE=0.0052+.81264E-3*XTIME-.22604E-5*XTIME**2-.27262E-8*XTIME**
```

```
     &       3-.115621E-11*XTIME**4+REQ(10,9,16)
      DPDT=.81264E-3-.22604E-5*XTIME*2+.27262E-8*XTIME**
     &       2*3.0-.115621E-11*XTIME**3*4
        ENDIF
      IF ( LL .EQ. 1) THEN
      QQ=1.0E8*DPDT
      Q=QQ*3.4134/60./60.
   65 CONTINUE
      Q=Q*QCORRT-QR

      ELSE
C     THIS USES A CURVE FIT THROUGH THE BURNRATE DATA GIVEN BY NRL
         QCORRT=0.0
         QCORR=0.0
         ITEST = 0
         BURNR1= 5.4576748 +0.18815346*XTIME-.20153996E-03*XTIME**2
         BURNR2= -1.3116787 + .33158595*XTIME-.7342952E-03*XTIME**2
     &          +.50945510E-06*XTIME**3
         IF (XTIME .LT. 100) THEN
           BURNR= BURNR2 + 1.3117-.013117*XTIME
         ELSE
           BURNR = BURNR2
         ENDIF
         IF(XTIME .LE. 300) GO TO 60
         IF(BURNR2 .LT. BURNR1) THEN
             BURNR = (BURNR1 + BURNR2) / 2
             GO TO 60
         ELSE
             IF ( XTIME .LT. 600.0) GO TO 60
             IF (ITEST .EQ. 0) THEN
                BURNR3 = BURNR2
                ITEST = 1
             ENDIF
             BURNR = BURNR3
          ENDIF
   60     Q = BURNR*2.2046*9612./3600.-QR
CC    THIS GIVES Q IN BTU/SEC

       ENDIF
       Q=59.313+0.7195*XTIME-0.1139E-2*XTIME**2-0.3367E-5*XTIME**3
       Q=Q*3412/3600
       RETURN
       END



C     ------------------------------------------------------------------
***   ******************************************************************
       SUBROUTINE RADHT(T4WALL,VFMXC)
***   ******************************************************************
       COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
       COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,UO,H,UGRT,BUOY,
     &  CP0,PRT,CONDO,VISO,RHOO,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
       COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &        ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
       COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
       COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR


       DIMENSION VFMXC(579,579),T4WALL(579)
       DO 4010 K=3,NKM1
       DO 4010 I=2,NI
       II=(K-3)*(NI-1)+I-1
       T4WALL(II)=CONSRA*T(I,NJRA,K)*T(I,NJRA,K)*T(I,NJRA,K)*T(I,NJRA,K)
 4010 CONTINUE
```

```
C RADIATION FROM THE FIRE TO THE WALL

      DO 4011 J=3,9
      JJ=561+9-J
      AVT=0.25*(T(16,J,16)+T(17,J,16)+T(16,J,17)+T(17,J,17))
      T4WALL(JJ)=CONSRA*AVT*AVT*AVT*AVT
 4011 CONTINUE
C
      DO 4012 J=3,14
      JJ=568+J-3
      AVT=0.25*(T(6,J,16)+T(7,J,16)+T(6,J,17)+T(7,J,17))
      T4WALL(JJ)=CONSRA*AVT*AVT*AVT*AVT
 4012 CONTINUE
C
      DO 4020 I=1,579
      RWALL(I)=0.0
      DO 4020 J=1,579
      RWALL(I)=RWALL(I)+VFMXC(I,J)*T4WALL(J)
 4020 CONTINUE
      RETURN
      END



C     ________________________________________________________________
***   ****************************************************************
      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                             *
*               PCORR       =     PRESSURE CORRECTION                  *
************************************************************************
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &        ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &        ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)

      SUMT=0.
      SUMPT=0.
      SUMPET=0.
      DO 370 I=2,NI
      DO 370 J=2,NJ
      DO 370 K=2,NK
      IF (NOD(I,J,K).EQ.1) GOTO 370
      DXI=XL(I,J,K,0,0,0)
      DYJ=YL(I,J,K,0,0,0)
      DZK=ZL(I,J,K,0,0,0)
      VOL=DXI*DYJ*DZK
      SUMT=SUMT+1./T(I,J,K)*VOL
      SUMPT=SUMPT+P(I,J,K)/T(I,J,K)*VOL
      SUMPET=SUMPET+REQ(I,J,K)*(1./1.0-1./T(I,J,K))*VOL
  370 CONTINUE
      SUMPET=SUMPET/UGRT
      PCORR=(SUMPET-SUMPT)/SUMT
      PCORRN=PCORR
```

```
      DO 371 I=1,NIP1
      DO 371 J=1,NJP1
      DO 371 K=1,NKP1
      P(I,J,K)=P(I,J,K)+PCORRN
  371 CONTINUE

      RETURN
      END



C     ________________________________________________________________
***   ****************************************************************
      SUBROUTINE SOLCON
***   ****************************************************************
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &            NCHPK(10),TCHP(10),CPS(10),CONS(10),WFAN(10)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)

      DO 402 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 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)=CPS(N)
      NOD(I,J,K)=1
      IF (J.EQ.NJ) COND(I,NJP1,K)=COND(I,NJ,K)
      IF (I.EQ.2)  COND(1,J,K)=COND(2,J,K)
      IF (I.EQ.NI) COND(NIP1,J,K)=COND(NI,J,K)
      IF (I.EQ.2.AND.J.EQ.NJ)  COND(1,J+1,K)=COND(2,J,K)
      IF (I.EQ.NI.AND.J.EQ.NJ) COND(NIP1,J+1,K)=COND(NI,J,K)
      IF (J.EQ.NJ) CPM(I,NJP1,K)=CPM(I,NJ,K)
      IF (I.EQ.2)  CPM(1,J,K)=CPM(2,J,K)
      IF (I.EQ.NI) CPM(NIP1,J,K)=CPM(NI,J,K)
      IF (I.EQ.2.AND.J.EQ.NJ)  CPM(1,J+1,K)=CPM(2,J,K)
      IF (I.EQ.NI.AND.J.EQ.NJ) CPM(NIP1,J+1,K)=CPM(NI,J,K)
  405 CONTINUE
  402 CONTINUE
       RETURN
       END


C     ________________________________________________________________
***   ****************************************************************
       SUBROUTINE PTRACK
***   ****************************************************************
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &      DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
```

```
CC ** THE FOLLOWING PRESSURE TEST IS A TEMPORARY MEASURE TO MODIFY THE
CC    HEAT INPUT TO FORCE THE CALCULATED PRESSURE TO AGREE WITH THE
CC    EXPERIMENTAL PRESSURE.  IT WILL BE USED UNTIL ACCURATE HEAT INPUT
CC ** IS RECEIVED.
CC
      PSOUTH=P(10,9,16)*CONST1+REQ(10,9,16)
      PERROR=(PCURVE-PSOUTH)/PCURVE
      QCORR=1.0+PERROR-(PSOUTH-PM1)/PCURVE
      QCORR=1.0+PERROR-(PSOUTH-PM1)/PCURVE+(PSOUTH-PM1)/(PCURVE-PCURM1)*
     &     (PCURVE-PM1)/PCURVE
      QCORRT=QCORRT*QCORR
      PCURM1=PCURVE
      PM1=PSOUTH
C
      RETURN
      END



C     ____________________________________________________________________
***   ********************************************************************
      SUBROUTINE TCP
***   ********************************************************************

**************************************************************************
*     THIS SUBROUTINE CALCULATES THE TEMPERATURE AT THE TERMOCOUPLE      *
*     POSITIONS.                                                         *
**************************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     & CP0,PRT,COND0,VISO,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)


      DO 5100 N=1,NTHCO
      II=NTH(N,1)
      JJ=NTH(N,2)
      KK=NTH(N,3)
      VOL=ABS((XC(II+1)-XC(II))*(YC(JJ+1)-YC(JJ))*(ZC(KK-1)-ZC(KK)))
      TCOUP(N)=0.
      DO 5101 I=II,II+1
      III=II+II+1-I
      DO 5101 J=JJ,JJ+1
      JJJ=JJ+JJ+1-J
      DO 5101 K=KK,KK+1
      KKK=KK+KK+1-K
      WVOL=ABS((XC(I)-CX(N))*(YC(J)-CY(N))*(ZC(K)-CZ(N)))/VOL
      TCOUP(N)=TCOUP(N)+WVOL*T(III,JJJ,KKK)
 5101 CONTINUE
      TCOUP(N)=TCOUP(N)*TR-273.18

 5100 CONTINUE

      RETURN
      END



C     ____________________________________________________________________
***   ********************************************************************
      SUBROUTINE OUT(NN)
***   ********************************************************************
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
```

```
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VISO,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(579)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
      XTIME=TIME*H/U0
      nnn=jnint(xtime)
      nnx=nnn+1
      IF( NN .EQ. 1) THEN
C
      QRR=60.*60./3.412/1000.*QR
      WRITE(6,500) XTIME,NTREAL,TIME,ITER,RESORM(ITER),SORSUM,QRR
  500 FORMAT(1X, 'TIME=',F7.3,' S',1X,'NTREAL=',I9,1X,
     & 'TIME=',F7.2,'<0>',1X,'ITER=',I2,1X,'SOURCE=',
     & F9.6,1X,'SORSUM=',F9.6,1X,' QR(KW) = ',F10.4)
C
      QKW = ((60.*60.)/(3.412*1000.))* Q
      PRINT *
      PRINT *, '  PCURVE           PSOUTH              PERROR          Q
     &CRR          QCORRT            Q(KW) '
      PRINT *, PCURVE,PSOUTH,PERROR,QCORR,QCORRT,QKW
      PRINT *
C
      ELSE IF( NN .EQ. 2 ) THEN
      PRINT *
      PRINT *,'     TEMPERATURES  AT THERMOCOUPLE POSITION IN (C)'
      WRITE (6,*) (TCOUP(N),N=1,NTHCO)
      PRINT *
      PRINT *

      ELSE
c       write(nnn,*)'  time=',xtime,'seconds'
c       write(nnn,*)'  node#              u          v          w'
c       write(nnx,*)'  time=',xtime,'seconds'
c       write(nnx,*)'  node*          temperature    pressure'
      DO 502 L=1,nkp1
      K=L
      DO 502 M=1,NIP1
      I=M
      WRITE(6,504) I,K
  504 FORMAT(/,2X,'I=',I2,5X,'K=',I2,/,10X,' T NOD',3X,'R(GM/C.C.)',2X,
     & 'U(CM/SEC)',2X,'V(CM/SEC)',2X,'W(CM/SEC)','P (ATM)',5X,'SMP',5X,
     & 'VIS(SEC/CM-CM)',3X,'COND(SEC/CM-CM)',' XSMP',/)
  513 DO 503 J=1,NJP1
C     XTEMP=T(I,J,K)/CONST3-273.16
      XTEMP=T(I,J,K)
C     XR=R(I,J,K)*RHO0/2.2048 *1000.*(0.0328)**3
      XR=R(I,J,K)
C     XU=U(I,J,K)*CONST6
C     XV=V(I,J,K)*CONST6
C     XW=W(I,J,K)*CONST6
C     XP=(P(I,J,K)*CONST1-REQ(I,J,K)*PINT)
      XP=P(I,J,K)
      XU=U(I,J,K)
      XV=V(I,J,K)
      XW=W(I,J,K-1)
CC    XVIS=VIS(I,J,K)*RHO0*CP0*H*U0*1.48814
```

```
CC    XCOND=COND(I,J,K)*RHOO*CP0*H*U0*1.48814
      XVIS=VIS(I,J,K)/VISO
      XCOND=COND(I,J,K)/VISO
      XSMP=RI(I,J,K)
      DDYY=1./FLOAT(NJM1-2)
      PE =SQRT(U(I,J,K)**2+V(I,J,K)**2+W(I,J,K)**2)*DDYY/COND(I,J,K)
      WRITE(nnn,555) i,j,k,xu,xv,xw
 555  format('node(',3i3,')',3e12.4)
      write(nnx,556) i,j,k,xtemp,xp
 556  format('node(',3i3,')',2(2x,e12.4))
  503 CONTINUE
  502 CONTINUE
      WRITE(6,*) 'THE TIME WHEN THE DATA WAS STORED ON DISK IS:',
     & XTIME
      close(nnn)
      close(nnx)
      ENDIF
      RETURN
      END
```

LIST OF REFERENCES

1. Yang, K.T., J.R. Lloyd, A.M. Kanury, and K. Satoh, "Modeling of Turbulent Buoyant Flows in Aircraft Cabins," *Combustion Science and Technology,* vol. 39, pp. 107-118, 1984.

2. Kou, H.S., K.T. Yang, and J.R. Lloyd, "Turbulent Buoyant Flow and Pressure Variations Around an Aircraft Fuselage in a Cross Wind Near the Ground," *Fire Safety Science--Proceedings of the First International Symposium,* pp. 173-184, 1986.

3. Nicolette, V.F., K.T. Yang, and J.R. Lloyd, "Transient Cooling by Natural Convection in a Two-Dimensional Square Enclosure," *Intl. J. Heat Transfer,* vol. 28, no. 9, pp. 1721-1732, 1985.

4. Aziz, K., and J.D. Hellums, "Numerical Solution of the Three-Dimensional Equation of Motion for Laminar Natural Convection," *Physics of Fluid,* vol. 10, pp. 314-324, 1967.

5. Mallinson, G.D., and G. De Vahl Davis, "Three-Dimensional Numerical Analysis of Transient Natural Convection in a Box a Numerical Study," *J. Fluid Mech.,* vol. 83, pp. 1-31, 1977.

6. Chan, A.M.C., and S. Banerjee, "Three-Dimensional Numerical Analysis of Transient Natural Convection in Rectangular Enclosure," *Int. J. Heat Mass Transfer,* vol. 26, pp. 1427-1438, 1983.

7. Ozeo, H., K.. Fujii, N. Lior, and S.W. Churchill, "Long Rolls Generated by Natural Convection in an Inclined, Rectangular Enclosure," *Int. J. Heat Mass Transfer,* vol. 26, pp. 1427-1438, 1983.

8. Morrison, G.L., and V.G. Tran, "Laminar Flow Structure in Vertical Free Convection Cavities," *Int. J. Heat Mass Transfer,* vol. 21, pp. 203-213, 1978.

9. Yang, H.Q., K.T. Yang, and J.R. Lloyd, "Flow Transition in Laminar Flow in a Three-Dimensional Tilted Rectangular Enclosure," *Heat Transfer,* vol. 4, pp. 1495-1500, 1986.

10. Yang, H.Q., K.T. Yang, and J.R. Lloyd, "Laminar Natural Convection Flow Transition in Tilted Three-Dimensional Longitudinal Rectangular Enclosures," *Int. J. Heat Mass Transfer,* vol. 30, no. 8., pp. 1637,1644, 1987.

11. Yang, H.Q., K.T. Yang, and J.R. Lloyd, "Three Dimensional Buoyant Bimodal Flow Transition in Tilted Enclosures," *Int. J. Heat and Fluid Flow,* 1988.

12. Yang, H.Q., *Laminar Buoyant Flow Transitions in Three-Dimensional Tilted Rectangular Enclosures,* Ph.D. Thesis, University of Notre Dame, South Bend, IN, 1987.

13. Ozeo, H., T. Shibata, and S.W. Churchill, "Natural Convection in an Inclined Circular Cylinder Annulus Heated and Cooled in its End Plates," *Int. J. Heat Mass Transfer,* vol. 24, pp. 727-737, 1981.

14. Takata, Y., K. Fukuda, S. Hasegawa, H. Shimomura, and K. Sanokawa, "Three-Dimensional Natural Convection in a Porous Medium Enclosed in a Vertical Curved Annulus," *Numerical Heat Transfer,* pp. 29-39, 1983.

15. Takata, H., K. Iwashige, K. Fukuda, and S. Hasegawa, "Three-Dimensional Natural Convection in an Inclined Cylindrical Annulus, *Int. J. Heat Mass Transfer,* vol. 27, pp. 747-754, 1984.

16. Rao, Y., Y. Miki, K. Fukuda, and Y. Takata, "Flow Patterns of Natural Convection in Horizontal Cylindrical Annuli," *Int. J. Heat Mass Transfer,* vol. 28, pp. 705-714, 1985.

17. Fusegi, T., and B. Farouk, "A Three-Dimensional Study of Natural Convection in the Annulus Between Horizontal Concentric Cylinders," *Proc. 8th Int. Heat Transfer Conf.,* San Francisco, CA, vol. 4, pp. 1575-1580, 1986.

18. Smutek, C., P. Bontoux, B. Roux, G.H. Schiroky, A.C. Hurford, F. Rosenberger, and G. De Vahl Davis, "Three Dimensional Convection in Horizontal Cylinders: Numerical Solutions and Comparison with Experimental and Analytical Results," *Numerical Heat Transfer,* vol. 8, pp. 613-631, 1985.

19. Yang, H.Q., K.T. Yang, and J.R. Lloyd, "A Numerical Study of Three-Dimensional Laminar Natural Convection in a Horizontal Cylinder with Differentially Heated End Walls at High Rayleigh Numbers," *Proceedings of the Symposium on Heat and Mass Transfer Honoring Professor B.T. Chao,* University of Illinois, Urbana, IL, pp. 153-195, 1987.

20. Ozeo, H., K. Fujii, T. Shibata and S.W. Churchill, "Three-Dimensional Numerical Analysis of Natural Convection in a Spherical Annulus," *Numerical Heat Transfer,* vol. 8, pp. 383-406, 1985.

21. Baum, H.R., and R.G. Rehm, "Calculations of Three Dimensional Buoyant Plumes in Enclosures," *Combustion Science and Technology,* vol. 40, pp. 55-77.

22. Baum, H.R., and R.G. Rehm, "Natural Computation of Large Scale Fire-Induced Flows," paper presented at the Eighth Int. Conf. on Numerical Methods in Fluid Dynamics, Aachen, West Germany, 28 June-2 July 1982.

23. Baum, H.R. and R.G. Rehm, "Computation of Fire Induced Flow and Smoke Coagulation," Nineteenth Symposium, Int. of Combustion, The Combustion Institute, Pittsburgh, PA, pp. 921-931, 1982.

24. Rehm, R.G., and H.R. Baum, "The Equations of Motion for Thermally Driven, Buoyant Flows," *J. of Research of the National Bureau of Standards,* vol. 83, no. 3, pp. 297-308, May-June 1978.

25. Bagnaro, M., M. Laouisset, and F.C. Lockwood, "Field Model Prediction of Some Room Fires: Steady and Transient," *Fire Dynamics and Heat Transfer,* ASME HTD, vol. 25, ASME, New York,NY, pp. 107-114, 1983.

26. Markatos, N.C., and K.a. Pericleous, "An Investigation of Three Dimensional Fires in Enclosures," *Fire Dynamics and Heat Transfer,* ASME HTD, vol. 25, ASME, New York, NY, pp. 115-124, 1983.

27. Alexander, J.I., H.J. St. Aubin, J.P. Stone, T.T. Street, and F.W. Williams, "Large-Scale Pressurizable Fire Test Facility -- Fire-1," NRL Report 864, Naval Research Laboratory, Washington, D.C., December 1982.

28. Nies, G.F., *Numerical Field Model Simulation of Full Scale Tests in a Closed Vessel,* Master's and Mechanical Engineer's Thesis, Naval Postgraduate School, Monterey, CA, December 1986.

29. Raycraft, J.K., *Numerical Field Model Simulation of Full Scale Fire Tests in a Closed Spherical/Cylindrical Vessel,* Master's and Mechanical Engineer's Thesis, Naval Postgraduate School, Monterey, CA, December 1987.

30. Houck, R.R., *Numerical Field Model Simulation of Full Scale Fire Tests in a Closed Spherical/Cylindrical Vessel with Internal Ventilation,* Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1988.

31. CA-DISSPLA™ Computer Associates International Inc., Garden City, NY, 1988.

32. Eringen, A.C., *Mechanics of Continua,* John Wiley & Sons, Inc., New York, NY, 1967.

33. Department of Aerospace and Mechanical Engineering, University of Notre Dame, South Bend, IN, Technical Report TR-79002-78-2, "An Algebraic Turbulence Model for Buoyant Recirculating Flow," by V.W. Nee and V.K. Liu, 1978.

34. Sparrow, E.M., and R.D. Cess, *Radiation Heat Transfer,* Hemisphere Publishing Corporation, Washington, D.C., 1978.

35. Department of Aerospace and Mechanical Engineering, University of Notre Dame, South Bend, IN, Technical Report, TR-37191-74-4, "A Numerical Model for the Prediction of Two Dimensional Unsteady Flows of Multicomponent Gases with Strong Buoyancy Effect and Recirculation," by Michael L. Doria, November 1974.

36. Patanker, S.V., *Numerical Heat Transfer and Fluid Flow,* Hemisphere Publishing Company, New York, NY, 1980.

37. Leonard, S.P., "A Convectively Stable, Third-Order Accurate Finite-Difference Method for Steady Two-Dimensional Flow and Heat Transfer," *Numerical Properties and Methodologies in Heat Transfer,* ed. T.M. Shih, Hemisphere Publishing Corp., Washington, D.C., pp. 211-226, 1983.

INITIAL DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Dr. Patricia A. Tatum Code 6183 Naval Research Laboratory Washington, D.C. 20375	2
4.	Professor K.T. Yang Department of Aerospace and Mechanical Engineering University of Notre Dame Notre Dame, IN 46556	2
5.	Professor M.D. Kelleher, Code ME/Kk Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 939443-5004	2
6.	Chairman, Department of Mechanical Engineering Code ME Naval Postgraduate School Monterey, CA 93943-5004	1
7.	Naval Engineering Curricular Officer, Code 34 Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
8.	Mr. James Gagorik Code 211, ONT Balston Towers #1 800 N. Quincy Arlington, VA 22217	2
9.	LT Timothy McCarthy FSC 473 Box 8-28 FPO AP 96349-1400	1

Thesis
M16652 McCarthy
c.1
Numerical field model simulation of full-scale fire tests in a closed spherical/cylindrical vessel using advanced computer graphics techniques.