NISTIR 89-4211

A NUMERICAL METHOD FOR CALCULATING INDOOR AIRFLOWS USING A TURBULENCE MODEL

T. Kurabuchi
Guest Worker

Department of Architecture
Faculty of Engineering
University of Tokyo

J. B. Fang
Richard A. Grot

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
Lee Mercer, Deputy Under Secretary for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

NISTIR 89-4211

A NUMERICAL METHOD FOR CALCULATING INDOOR AIRFLOWS USING A TURBULENCE MODEL

T. Kurabuchi
Guest Worker

Department of Architecture
Faculty of Engineering
University of Tokyo

J. B. Fang
Richard A. Grot

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

January 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
Lee Mercer, Deputy Under Secretary for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond G. Kammer, Acting Director

Abstract

This report describes a numerical method based on a finite difference technique for simulating indoor airflows in a building using a κ-ε turbulence method. The model treats three dimensional non-isothermal turbulent flows using the Boussinesq approximation for buoyancy. It solves the resulting nonlinear system of momentum, energy and turbulence equations by an explicit time marching technique to obtain a solution for either a steady state or transient flow. An upwind/central combination scheme with arbitrary specification for the switching parameter is used to approximate the convective terms. This switching parameter can be specified at each point in the flow regime allowing for different strategies in different flow regions. The switching technique includes both the central and hybrid schemes found in the literature. A pressure relaxation method is used to satisfy the Poisson equation for continuity. The model handles a variety of flow, pressure, temperature and heat flux boundary conditions including prescribed inflows, outflows by either prescribing the flow or pressure, wall boundary conditions together with heat flux and temperature and/or heat transfer coefficients specified on the boundary. Volumetric heat sources are also included. The model has the ability of handling an arbitrary number of obstacles in the flow region. This permits the modeling of the effect of furniture and partitions on the flow field and also provides a means for modeling multi-room airflows. The predicted airflows can be used in a companion computer model for predicting the three-dimensional dispersion of contaminants in a building. The computer code for this model exists both in a vectorized version for the Cyber 205 supercomputer and in a non-vectorized version which has been successfully run on a Sun 3/260 workstation with a floating point processor board (based on a Weitek 1167) under a UNIX operating system on a Compaq 386/25 computer equipped with either an Intel 80387 or a Weitek 3167 coprocessor under an extended DOS operating system. The relative performance of these systems for the examples considered in this report are 1 second per iteration for the Cyber 205, 9 seconds for the Compaq 386/25 with a Weitek 3167 coprocessor, 30 seconds for the Compaq 386 with a 387 coprocessor and 90 seconds for the Sun 3/260 under UNIX. Isothermal simulations seem to converge in approximately 10,000 iterations and non-isothermal simulations in approximately 30,000 iterations. Several ideal and practical applications of the model are presented and the results of the simulations are compared with existing experimental data contained in the literature. [1]

1 Key Words: Airflow, building ventilation, contaminat dispersal, finite difference method, indoor air quality, mathematical modeling, room convection, temperature, turbulence, velocity.

Table of Contents

1. Introduction

Although it is well known that the indoor environment is strongly affected by airflow, especially when the space is conditioned, the mechanisms driving indoor airflow are not well understood. In Japan, advanced computer simulation techniques to predict room air movement have been available for several years and have been used for evaluating both thermal comfort and indoor air quality. To date, such computer simulation techniques have not widely been used in the United States. However, increasing demand for efficient clean rooms needed for high-tech factories and for the optimal design of air conditioning systems for office buildings is stimulating extensive research on the micro modeling of indoor airflow distribution.

Conventional methods, such as single or multi-zonal methods for predicting room air movement or simple calculations based on free jet decay experiments, are not useful room air movement predictions because of their inherent inability to extend the applicable range of calculation needed for a more detailed prediction. Alternative approaches based on more fundamental fluid theory are required.

This report describes one possible approach, a numerical flow calculation method, based on the κ-ε turbulence model, and discusses its reliability and accuracy with respect to several experimental observations.

A vectorized three dimensional computer simulation program called "EXACT", which has been recently installed in the CYBER 205 computer system of the National Institute of Standards and Technology is discussed. Non-vectorized versions of this program have also been implemented on a Sun 3/260 workstation and on a Compaq 386/25 computer using a DOS extender.

Although the prediction method given here requires large amounts of computation even with advanced and highly efficient scientific computers, the authors believe that, with the advances in both super and workstation computer technology and the theoretical development in turbulence theory, it has a fair prospect to become a practical prediction method in the near future.

1.1 Indoor Airflow and Turbulence

Although most available flow calculation methods employ the Navier-Stokes equation as the basic starting point, different characteristic flow regimes require markedly different numerical calculation strategies.

Consider the requirements for the numerical calculations of low speed oil flow in a pipeline and the airflow around a car which is moving at 60mph. In the first case, a laminar flow calculation method would be appropriate; yet,it would not be valid in the second case since turbulence induced by the movement of the car must be taken into consideration. This is in spite of the fact that both flows are described adequately by the same fundamental equations. The first task in the development of a model for predicting room air movement is to determine which regime, turbulent or non-turbulent, the indoor airflow belongs to. Generally, in developing a numerical calculation method for airflow, a representative Reynold's number *Re* is useful in deciding whether the flow to be considered is laminar or turbulent. The representative Reynold's number is defined as

$$Re = \frac{U_0 L_0}{\nu} \tag{1.1}$$

where

U_0	is a representative velocity scale (m/s)
L_0	is a representative length scale (m)
ν	is the kinematic viscosity (m^2/s) (for air $\nu = 1.5 \times 10^{-5} m^2/s$

For the discharge airflow related to air-conditioning devices, the following values have been determined experimentally as a critical representaitve Reynolds number.

• plane jets

Re_{crit} :where U_0 is taken as jet exit velocity and L_0 slit width respectively.

This means, for instance, that the jet from 1.5 cm slit type air inlet becomes turbulent if its exit velocity is larger than 3 or 5 cm/s.

• round jets

Re_{crit} :where U_0 is taken as jet exit velocity and L_0 as the diameter of the round nozzle respectively. This means, for instance, that the jet from a nozzle with a 15 cm diameter becomes turbulent if its exit velocity is larger than $3 \sim 20 cm/s$.

These experimental data show that airflow around the supply registers in an actual air-conditioned building is always turbulent under normal conditions.

For flows near the wall, experimental data on forced convection over the flat plate may be useful.

• flow over a flat plate

Re_{crit} :where a wind tunnel type flow is assumed and U_0 taken as velocity outside the wall boundary layer and and L_0 as the distance from the leading edge of the flat plate respectively. This means that wall boundary layer does not become completely turbulent unless the length scale is greater than $4.5 \sim 6m$ when the velocity is of the order of 1 m/s.

Though it is not clear that the actual near wall flow in an air conditioned room actually corresponds to this type of flow, this suggests that there is a strong possibility that turbulent and non-turbulent flow portions coexist in the same room space. In spite of this possibility, the currently available numerical method is limited to single phase problems either laminar or turbulent flows.

It is necessary, therefore, first to assume which condition is dominant in the overall condition of the room flow field. In the case of air-conditioned rooms using forced air systems, due to the existence of the jet produced by the room air handling system, the fully turbulent assumption is probably justified as a first appoximation since the main interest in room airflow simulation is the accurate prediction of the location of the stream of incoming air and resultant recirculating regions produced by this stream of air.

In order to improve the current full turbulence model and to extend its range of applications, more fundamental research for weak turbulent flow is required.

1.2 History of Numerical Prediction Method of Indoor Air Distribution in Japan

The first attempt at the numerical calculation of indoor airflow in Japan was made by Terai[6] in 1959 for the case of two-dimensional buoyant convection flow. Since then, many Japanese researchers[6-19] have made extensive efforts to develop a numerical prediction method which could be used as a practical tool for indoor environmental design. Given below is brief description of the history of these activities in Japan.

The first practical and important contributions were made by Tsuchiya[7], Kaizuka[8] and Yamazaki[9] independently for the two-dimensional, isothermal laminar flows. These research efforts were followed by those of Nomura, Matsuo, Kaizuka, Sakamoto and Endo[10] for three-dimensional laminar flows using the Marker and Cell (MAC) method. They also made qualitative comparisons of their calculation results with flow visualization experiments. Although their work was limited to laminar flow conditions, the numerical algorithm used by these authors was virtually the same as the current standard numerical method used in Japan today.

The first Japanese research to introduce the turbulence concept into the calculation of room airflows was the work by Yoshikawa and Yamaguchi[11] for two-dimensional high Reynolds number flows using an one equation model approach. After this research, Nomura, Matsuo, Kaizuka, Sakamoto and Kamata[12,13] calculated three-dimensional forced and natural convection flows using the κ-ε turbulence model and the MAC method. The predictions using this model were compared to corresponding model experiments.

After their work, Kato, Sato[14], Kunihira[15], Murakami[16], Kamata, Kurabuchi[17,18] and Matsuo[19] attempted comparative studies to verify the reliability and accuracy of the numerical method, and proposed extensions and improvements to the current method employing advanced boundary condition techniques or different numerical schemes. Recently, Murakami, Mochida and Hibi[16] have made extensive calculations using the large eddy simulation (LES) method and clarified its potential applicability to this field.

Nowadays, general interest is being focused on buoyant flow predictions, and the research groups mentioned above are trying to make original contribut
ions to this area.

1.3 Numerical Modeling of Air Movement in Buildings

Both experimental and numerical modeling were performed recently by Chen, et al.[48] to study air movement, temperature field and contaminant distribution in a ventilated room under different ventilation rates and heating loads. They have found that numerical predictions generally agree with the measured results and theat both the temperature and ventilation efficiencies increase with an increase in ventilation rate. Davidson and Olsson[49] carried out numerical simulations of the local age distribution and local purging rate in buoyantly ventilated rooms to investigate ventilation efficiency and contaminant dispersal in buildings. Very recently, Awbi[50] solved numerically the two and three dimensional, steady-state conservation of mass, momentum and energy equations, along with a two-equation turbulence model to predict the air velocity and temperature distributions in a ventilated room and found reasonably good correlation between the predicted results and experimental data.

2. Basic Equations for Turbulent Flows

As seen in the previous chapter, indoor airflow including jet type air inlet is expected to contain a turbulent portion, and a numerical prediction method must account for turbulence. Although the basic Navier-Stokes equations are capable of describing turbulent motion as well as laminar motion, their direct application for turbulent stimulation requires unrealistically large computational time and memory. The development of turbulence models and prediction methods that require moderate computational effort is desirable. So far, two series of prediction methods are considered realistic as far as engineering problems are concerned; one is called the Large Eddy Simulation (LES) method[20,21] and the other is called the field method [22].

The LES method is based on spatially filtered Navier-Stokes equations over coarse computational grids. The non-linear interaction between the large eddies and the subgrid eddies, whose size is less than grid size, is simulated using a subgrid eddy viscosity model. Although this subgrid model can be made theoretically acceptable and applicable to general problems, the LES method still requires large computer memory and long CPU time, even with the current vector processing supercomputers (partly due to the large amount of transient computations required for the method).

In contrast to the LES method, the field method employs averaged Navier-Stokes equations whose derivation requires rather uncertain assumptions to arrive at its basic equations. However, partly because it requires only a moderate computational effort, there have been many practical applications of this method. Moreover, the field model has proven to be capable of predicting a wide variety of practical problems with better accuracy than expected. In this chapter, the turbulence model of the field method, the κ-ε turbulence model [23] employed in the "Explicit Time Marching Algroithm for Continuous Thermal Fluid Flow" (EXACT) program is described.

2.1 The Navier-Stokes and Reynolds Equations

If general indoor flows and the related pollutant dispersion and diffusion occurring in actual rooms are to be considered, appropriate equations must be selected which are applicable to different indoor geometries, to buoyancy induced by temperature gradient and include the effect of turbulent flows. Under ordinary indoor conditions, the temperature differences and representative air speeds in a room are relatively small. In this case the Boussinesq approximation to the general compressible flow equations is applicable. The following simplified equations are taken as the basic equations for describing general indoor flows.

Momentum Equation

$$\frac{\partial U_i}{\partial t}+\frac{\partial U_i U_j}{\partial x_j}=-\frac{1}{\rho}\frac{\partial P}{\partial x_i}+\frac{\partial}{\partial x_j}\left\{\nu\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)\right\}-\beta g_i\Theta \qquad (2-1)$$

Conservation of Mass

$$\frac{\partial U_j}{\partial x_j}=0 \qquad (2-2)$$

Conservation of Energy

$$\frac{\partial \Theta}{\partial t}+\frac{\partial \Theta U_j}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\kappa\frac{\partial \Theta}{\partial x_j}\right)+H \qquad (2-3)$$

Conservation of Contaminant Species

$$\frac{\partial C}{\partial t}+\frac{\partial C U_j}{\partial x_j}=\frac{\partial}{\partial xj}\left(D\frac{\partial C}{\partial x_j}\right)+S \qquad (2-4)$$

where

C	is the instantaneous concentration for passive contaminant
D	is the molecular diffusion coefficient for passive contaminant
g_i	is the gravitational acceleration in x_i direction
U_i	is the instantaneous velocity component in x_i direction
P	is the instantaneous static pressure difference
t	is the time
x_i	are the cartesian coordinates
β	is the volumetric coefficient of expansion
κ	is the thermal diffusivity
Θ	is the instantaneous temperature difference
ρ	is the density of air
ν	is the kinematic viscosity of air
H	is the volume heat source generation rate
S	is the volume contaminant generation source

Since turbulent flow is characterized by random eddies of various sizes and it is the energy transfer process from the large scale motion to the small scale which is critical, the correct simulation of turbulent flow requires the numerical grid size at least comparable to the size of the smallest possible eddy. The eddy size may be estimated from the dimensional analysis to be of the order of the Kolmogorov length scale[24] $(\nu^3/\varepsilon)^{\frac{1}{4}}$ (typically $10^{-2} \sim 10^{-3}$ m) where ε is the dissipation rate of turbulence energy. Simulating numerically indoor flows for typical rooms on this small scale is too much for even the most advanced vector processing computers. On the other hand, most engineering problems are not concerned with the analysis of turbulent structure itself, but only the general distributions of the mean quantities and their statistical characteristics.

An averaging process is used on the original equations (2-1) to (2-4) to eliminate the unnecessary flow details. Assuming that the instantaneous quantity consists of a mean part and the fluctuating part, such as $U_i = u_i + u'_i$ the conservation equations for instantaneous parameters are averaged to produce equations for the mean parameters. The averaging procedure can be either a temporal or an ensemble average. Equations (2-1) to (2-4) are replaced by the following:

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{1}{\rho}\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left\{\nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\overline{u'_i u'_j}\right\}-\beta g_i \theta \qquad (2-5)$$

$$\frac{\partial u_j}{\partial x_j}=0 \qquad (2-6)$$

$$\frac{\partial \theta}{\partial t}+\frac{\partial \theta u_j}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\kappa\frac{\partial \theta}{\partial x_j}-\overline{\theta' u'_j}\right)+h \qquad (2-7)$$

$$\frac{\partial c}{\partial t}+\frac{\partial c u_j}{\partial x_j}=\frac{\partial}{\partial x j}\left(D\frac{\partial c}{\partial x_j}-\overline{c' u'_j}\right)+s \qquad (2-8)$$

where

c	is the mean volumetric concentration of passive contaminant
u_i	is themean velocity component in x_i direction
p	is the mean static pressure difference
θ	is the mean temperature difference
h	is the mean volume heat generation rate
s	is the mean volume contaminant source generation rate

and where ' indicates the fluctuating part of the quantity.

Since the spatial and temporal variations of mean parameters are expected to be much less than variations of the actual quantities, they can be numerically represented with relatively coarse grids leading to systems of equations that can be handled by the current computers.

Unlike the original equations, however, mean parameter equations contain unknown correlation terms for the fluctuating components (the apparent stress tensor appearing in equation (2-5) $-\nu u'_i u'_j$ is called the Reynolds stress tensor) resulting from the fluctuations of the non-linear convection terms. In order to close the equation system, these terms, which represent apparent stresses or fluxes due to turbulent motion, have to be approximated.

2.2 Turbulence Models

There are several approaches which have been proposed in order to approximate these unknown terms. They may be classified according to their modeling strategy as follows.

2.2.1 Indirect Methods for Evaluating the Unknown Turbulent Stresses and Fluxes

In this category of approximation methods, the unknown stress and flux terms are assumed proportional to the local gradient of mean variables and the isotropic turbulent viscosity or diffusivity (ν_t). The problem is then to evaluate spatial variation of ν_t. The most common indirect methods are the following:

Algebraic Approach [25]

The turbulent viscosity ν_t is given as a product of local gradient of mean velocity and an appropriate length scale of turbulence, typically, the distance from the wall. This approach was first introduced by Prandtl to predict the velocity profile near a wall. In a room, however, it is doubtful that the turbulent velocity scale can always be expressed only through the local gradient of mean velocity. Moreover, the algebraic expression for the turbulence length scale seems to restrict the applicable range of this method.

One Equation Model Approach [2]

Following Prandtl and Kormogorov, ν_t is assumed to be proportional to the square root of turbulence kinetic energy (k) and the length scale of energy containing eddies (l), where the spatial variation of k is solved using a model transport equation for k, while l is given by an algebraic expression. Although this is an improvement on the algebraic model, the problem inherent in the empirical assumption for the length scale is the same as the algebraic model.

The Two Equation Model Approach [25]

Again, the Prandtl-Kormogorov approach is employed in a similar manner to the one equation model. An additional transport equation concerning the value of $z = k^m l^n$ is solved together with the k transport equation, and ν_t is given as follows:

$$\nu_t \sim k^{\frac{1}{2}-\frac{m}{n}} z^{\frac{1}{n}} \qquad (2-9)$$

Different choices of m and n in equation (2-9) produce different types of two-equation models, such as $k-k\,l$ model [26] (m=1,n=1), k-ε model (m=3/2,n=1), k-W model [27] (m=1,n=2).

2.2.2 The Direct Method for Evaluating The Unknown Turbulent Stresses and Fluxes [28 29]

In the direct method the turbulent viscosity/diffusivity concept is abandoned and the turbulent stresses and fluxes are determined directly from the model transport equations themselves. Although this method seems to enjoy general applicability, it requires many partial differential equations derived from numerous assumptions. It has not been clear to date whether the increased complexity of the model and computational resources required by the method are justified by the obtained improvement in accuracy.

2.3 The k - ε Turbulence Model

Among these available turbulence models, the two equation model with the choice of the ε transport equation has been most widely validated and shown to have sufficient accuracy for most engineering problems. Since the k-ε turbulence model is also employed in the "EXACT" code, a brief description of its derivation form the basic equations is given below.

In the k-ε turbulence model, the Reynolds stress tensor is approximated by using the eddy (turbulent) viscosity concept given in equation (2-10):

$$-\overline{u'_i u'_j} = \nu_t\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)-\frac{2}{3}k\delta_{ij} \qquad (2-10)$$

where δ_{ij} is Kronecker's delta (= 1 for i = j, and = 0 for i ≠ j).

The second term in equation (2-10) is added in order that both sides are equal to -2k when a summation of normal stress is taken.

Similarly, a turbulent scalar flux term is approximated by assuming gradient transport and constant Prandtl number. This leads to equation (2-11):

$$-\overline{\phi' u'_j} = \frac{\nu_t}{\sigma}\left(\frac{\partial \phi}{\partial x_j}\right) \qquad (2-11)$$

where φ is the mean scalar dependent variable and σ is the turbulent Prandtl number for φ.

These approximations are derived from an analogy to the molecular transport process.

By using equation (2-10), the equation for mean momentum transport (Reynolds equation) can be approximated as follows:

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{1}{\rho}\frac{\partial \Pi}{\partial x_i}+\frac{\partial}{\partial x_j}\left\{(\nu+\nu_t)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\right\}-\beta g_i\theta \qquad (2-12)$$

where the mean static pressure p is replaced by the total pressure: $\Pi = p + 2\rho\, k/3$ which includes the turbulent pressure.

The energy equation (2-7) and contaminant transport equation (2-8) are approximated as follows:

$$\frac{\partial \theta}{\partial t}+\frac{\partial \theta u_j}{\partial x_j}=\frac{\partial}{\partial x_j}\left\{\left(\kappa+\frac{\nu_t}{\sigma_\theta}\right)\frac{\partial \theta}{\partial x_j}\right\}+h \qquad (2-13)$$

$$\frac{\partial c}{\partial t}+\frac{\partial c u_j}{\partial x_j}=\frac{\partial}{\partial x_j}\left\{\left(D+\frac{\nu_t}{\sigma_c}\right)\frac{\partial c}{\partial x_j}\right\}+s \qquad (2-14)$$

The remaining unknown parameter ν_t is given as a product of the square root of turbulence kinetic energy (k) and the length scale of energy containing eddies (l) as:

$$\nu_t = k^{\frac{1}{2}} l \qquad (2-15)$$

In the k-ε turbulence model, the model transport equation for dissipation rate of turbulence energy (ε) is employed to obtain the additional parameter and a dimensional analysis leads to the following relationship among k, ε and l for high Reynolds number flow.

$$\varepsilon = C_D k^{\frac{3}{2}}/l \qquad (2-16)$$

where C_D is an empirical constant.

If l in equation (2-15) is eliminated by using equation (2-16), an approximate form of ν_t is given as follows:

$$\nu_t = C_D \frac{k^2}{\varepsilon} \qquad (2-17)$$

An exact form of the transport equation of turbulence kinetic energy is derived by manipulating Navier-Stokes equation (2-1) and given as follows:

$$\underbrace{\frac{Dk}{Dt}}_{I} = \underbrace{-\frac{\partial}{\partial x_j}\left(\overline{u'_j\left(\frac{u'_k u'_k}{2}+\frac{p'}{\rho}\right)}\right)}_{II} \underbrace{-\ \overline{u'_j u'_i}\frac{\partial u_i}{\partial x_j}}_{III} \underbrace{+\ \nu\frac{\partial}{\partial x_j}\overline{u'_i\left(\frac{\partial u'_i}{\partial x_j}+\frac{\partial u'_j}{\partial x_i}\right)}}_{IV}$$

$$\underbrace{-\ \nu\overline{\left(\frac{\partial u'_i}{\partial x_j}+\frac{\partial u'_j}{\partial x_i}\right)\frac{\partial u'_i}{\partial x_j}}}_{V} \underbrace{-\ \beta g_i \overline{u'_i \theta'}}_{VI} \qquad (2-18)$$

where $\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u_i \frac{\partial}{\partial u_i}$ denotes the substantial derivative operator.

The terms in (2-18) represented by Roman numerals have the following physical meaning.

I	: rate of change and advective transport
II	: diffusive transport due to turbulence
III	: shear stress production $= P$
IV	: molecular diffusion
V	: viscous dissipation rate
VI	: buoyancy production/destruction $= G$

An alternative form of the exact transport equation for k can be derived from transformation of IV and V terms in equation (2-18), and the resultant equation has a more convenient form for modeling:

$$\frac{Dk}{Dt} = -\frac{\partial}{\partial x_j}\left(\overline{-u'_j\left(\frac{u'_k u'_k}{2} + \frac{p'}{\rho}\right)} + \nu \frac{\partial k}{\partial x_j}\right) + P + G - \nu \overline{\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_i}{\partial x_j}} \qquad (2-19)$$

In equation (2-19), the turbulent and molecular diffusion effects are put in simplified form, and only the isotropic part of viscous dissipation rate appears explicitly.

The first term in equation (2-19) represents the turbulent transport of k by diffusion. This is the internal term and it vanishes when integrated over the flow volume. This implies that this term exerts no contribution to production nor to the destruction of k, but merely transports it from one place to another. The second part of this term represents molecular diffusion.

In the k-ε turbulence model, the first part of this term is modeled in accordance with the gradient-diffusion assumption as follows:

$$\overline{-u'_j\left(\frac{u'_k u'_k}{2} + \frac{p'}{\rho}\right)} \sim \frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_j} \qquad (2-20)$$

where σ_k is the turbulent Prandtl number for turbulence energy.

The P , G terms in equation (2-9) are approximated using equations (2-10) and (2-11) as follows:

$$P = -\overline{u'_i u'_j}\frac{\partial u_i}{\partial x_j} \sim \nu_t\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_i}{\partial x_j} \qquad (2-21)$$

$$G = -\beta g_i \overline{u'_i \theta'} \sim \beta g_i \frac{\nu_t}{\sigma_\theta}\frac{\partial \theta}{\partial x_i} \qquad (2-22)$$

The last term in equation (2-19) represents the isotropic part of viscous dissipation rate. The local structure of turbulence tends to be isotropic in the high Reynolds number flow and this part exerts a dominant contribution to total dissipation. The non-isotropic contribution is thus assumed negligible:

$$\nu \overline{\frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}} \sim \varepsilon \qquad (2-23)$$

Using these approximations, the model equation for k is given as follows.

$$\frac{Dk}{Dt}=\frac{\partial}{\partial x_j}\left\{\left(\nu+\frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right\}+\nu_t\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_i}{\partial x_j}+\beta g_i\frac{\nu_t}{\sigma_\theta}\frac{\partial \theta}{\partial x_i}-\varepsilon \qquad (2-24)$$

The exact equation for the dissipation rate of turbulence kinetic energy (ε) is also derived by manipulating Navier-Stokes (2-1), resulting in:

$$\underbrace{\frac{D\varepsilon}{Dt}}_{I}=\frac{\partial}{\partial x_j}\{\underbrace{-\nu\overline{u'_j\frac{\partial u'_i}{\partial x_m}\frac{\partial u'_i}{\partial x_m}}}_{II}\ \underbrace{\frac{2\nu}{\rho}\overline{\frac{\partial p'}{\partial x_m}\frac{\partial u'_j}{\partial x_m}}}_{III}+\underbrace{\nu\frac{\partial \varepsilon}{\partial x_j}}_{IV}\}$$

$$\underbrace{-2\nu\frac{\partial u_i}{\partial x_j}\left(\overline{\frac{\partial u'_i}{\partial x_m}\frac{\partial u'_j}{\partial x_m}}+\overline{\frac{\partial u'_m}{\partial x_i}\frac{\partial u'_m}{\partial x_j}}\right)}_{V}-\underbrace{2\nu\frac{\partial^2 u_i}{\partial x_i\partial x_j}\overline{u'_k\frac{\partial u'_i}{\partial x_m}}}_{VI}$$

$$\underbrace{-2\nu\overline{\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_i}{\partial x_m}\frac{\partial u'_j}{\partial x_m}}}_{VII}\ \underbrace{-2\nu^2\overline{\frac{\partial^2 u'_i}{\partial x_j\partial x_m}\frac{\partial^2 u'_i}{\partial x_j\partial x_m}}}_{VIII} \qquad (2-25)$$

where isothermal conditions ared assumed.

The terms (II, III) due to turbulent diffusion are model as in the *k*-equation using the gradient diffusion assumption .

$$-\nu\overline{u'_j\frac{\partial u'_i}{\partial x_m}\frac{\partial u'_i}{\partial x_m}}-2\frac{\nu}{\rho}\overline{\frac{\partial p'}{\partial x_m}\frac{\partial u'_j}{\partial x_m}} \sim \frac{\nu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_j} \qquad (2-26)$$

Term IV does not have to be modeled. Terms V and VI are production terms due to the mean motion. These terms are omitted because they tend to be less important when the Reynolds number is large and local structure of turbulence becomes nearly isotropic.

Terms VI and VII represent the generation terms due to vortex stretching and destruction due to viscous action, respectively. They cannot be modeled separately. The reason is, as given by Rodi[30], that these terms increase with increasing Reynolds number while their differences are independent of the Reynolds number. The generation-destruction terms in ε-equation must be treated simultaneously and their modeled forms must not contain any term depending on the Reynolds number. The most widely used approximation for these terms is given as follows:

$$-2\nu\overline{\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_i}{\partial x_m}\frac{\partial u'_j}{\partial x_m}}-2\nu^2\overline{\frac{\partial^2 u'_i}{\partial x_j\partial x_m}\frac{\partial^2 u'_i}{\partial x_j\partial x_m}} \sim \frac{\varepsilon}{k}(C_1P-C_2\varepsilon) \qquad (2-27)$$

where C_1, C_2 are empirical constants.

Although the most straightforward way to include the buoyancy effect in this term is to replace P in equation (2-27) by $P + G$, several numerical experiments have shown this method to be ineffective in some cases. To attempt to correct this difficiency, another empirical constant C_3 is introduced for the buoyancy generation term resulting in the following equation for ε:

$$\frac{D\varepsilon}{Dt}=\frac{\partial}{\partial x_j}\left\{\left(\nu+\frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial x_j}\right\}+\frac{\varepsilon}{k}\left\{C_1\nu_t\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_i}{\partial x_j}-C_2\varepsilon+C_3\beta g_i\frac{\nu_t}{\sigma_\theta}\frac{\partial\theta}{\partial x_i}\right\} \qquad (2-28)$$

2.3.1 Estimation of the Empirical Constants

The k-ε turbulence model contains several empirical constants which must be determined in such a manner that the model can describe very basic turbulent flows properly.

First, the model should replicate the turbulence near the wall. Within a flow of this kind, convection and diffusion of turbulence energy are negligible and the boundary layer approximation can be applied. With this assumption, the k-equation (2-24) may be simplified as follows:

$$\nu_t\left(\frac{\partial u}{\partial y}\right)^2=\varepsilon=C_D\frac{k^{\frac{3}{2}}}{l} \qquad (2-29)$$

where u is the streamwise component of mean velocity and y is the coordinate axis in the direction lateral to the stream line. Equation (2-16) is also used to express the dissipation rate of turbulence energy. Multiplying ν_t on both sides of equation (2-29) and the use of equation (2-15) yields the following equation:

$$\nu_t\left(\frac{\partial u}{\partial y}\right)=C_D{}^{\frac{1}{2}}k \qquad (2-30)$$

The left hand side of equation (2-30) represents apparent shear stress (divided by density of fluid) due to turbulence. Meanwhile, Reynolds equation (2-5) under the condition of two-dimensional, isothermal and steady state can be written for the streamwise component as follows:

$$\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial x}+\frac{\partial}{\partial x}\left\{2\nu\frac{\partial u}{\partial x}-\overline{u'^2}\right\}+\frac{\partial}{\partial y}\left\{\nu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)-\overline{u'v'}\right\} \qquad (2-31)$$

where

- x : cartesian coordinate in the streamwise direction
- y : cartesian coordinate in the lateral direction
- u : time-averaged velocity component in x direction
- v : time-averaged velocity component in y direction

Here, if zero pressure gradient in the streamwise direction is assumed and the boundary layer approximation is applied to equation (2-31), the non-vanishing term has to satisfy the following relation:

$$0=\frac{\partial}{\partial y}\left\{\nu\left(\frac{\partial u}{\partial y}\right)-\overline{u'v'}\right\} \qquad (2-32)$$

This means that shear stress does not change and takes a constant value of skin friction (τ^*) on the wall within the boundary layer. The left hand side of equation (2-30) is, therefore, independent of y and equal to u^{*2} (u^*:friction velocity, $u^*=(\tau^*/\rho)^{\frac{1}{2}}$) except where y is close to zero and the kinematic viscosity effect is not negligible. Hence, the turbulence energy, right hand side of equation (2-30), is also constant within the boundary layer. Finally, the equation for C_D is given as follows:

$$C_D = \left(\frac{u^{*2}}{k} \right)^2 \qquad (2-33)$$

Experimental results suggest that C_D should take the value of 0.09. Other useful relations may be derived from the mixing-length hypothesis $\frac{\partial u}{\partial y} = \frac{u^*}{\kappa y}$, κ(Karman's constant)~0.4 as follows:

$$\nu_t = C_D^{\frac{1}{4}} \kappa\, y\, k^{\frac{1}{2}} \qquad (2-34)$$

$$\varepsilon = \frac{C_D^{\frac{3}{4}} k^{\frac{3}{2}}}{\kappa\, y} \qquad (2-35)$$

If the ε-equation (2-28) is also applied to the same problem, its simplified form is given as follows, neglecting convection terms. (The diffusion term cannot be neglected in this case.)

$$0 = \frac{\partial}{\partial y}\left\{ \left(\frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial y} \right\} + \frac{\varepsilon}{k}\left\{ C_1 \nu_t \left(\frac{\partial u}{\partial y} \right)^2 - C_2 \varepsilon \right\} \qquad (2-36)$$

By using equations (2-33), (2-34), (2-35) and the mixing-length hypothesis, equation (2-36) may be transformed into the following in order to show the relationship among various empirical constants appearing in equation (2-36).

$$C_1 = C_2 - \frac{\kappa^2}{\sigma_\varepsilon C_D^{\frac{1}{2}}} \qquad (2-37)$$

Next to be considered is the decay of turbulence behind a grill. As the production and diffusion terms in k and ε-equations vanish due to the negligibly small mean strain rate in this process, the convection and destruction terms are nearly in balance as follows.

$$u \frac{\partial k}{\partial x} = -\varepsilon \qquad (2-38)$$

$$u \frac{\partial \varepsilon}{\partial x} = -C_2 \frac{\varepsilon^2}{k} \qquad (2-39)$$

where x is the streamwise coordinate and u is the velocity component in x-direction.

Equation (2-38) shows that turbulence kinetic energy decreases monotonically in the streamwise direction and its variation may be approximated using the following equation.

$$k(x) = \frac{k_0 x_0^n}{(x + x_0)^n} \tag{2-40}$$

By substituting equation (2-40) and (2-38) and using equation (2-39), the following equation is obtained for C_2.

$$C_2 = 1 + \frac{1}{n} \tag{2-41}$$

Experimental data show that n is close to unity and a value of 1.92 is commonly used for C_2.

Prandtl numbers for turbulence energy and its dissipation rate, σ_k and σ_ε should be of the order of unity, and 1.0 and 1.3 are chosen respectively. From this, C_1 takes the value of 1.44 from equation (2-37). The only remaining constant C_3 is believed to take a value ranging from 0 to C_1.

Rodi suggested[31], however, that C_3 should depend on the amount of thermal stratification and should be a function of the Richardson number Rf (Rf is defined as minus the ratio of buoyancy production of k to stress production, $-G/P$)

Since the final conclusion has not been obtained for this problem so far, C_3 should be tuned depending on each specific condition so as to give the best result.

Finally, the complete k-ε turbulence model together with the recommended empirical constants are summarized as follows.

Conservation equations for mean dependent variables

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \Pi}{\partial x_i} + \frac{\partial}{\partial x_j}\left\{(\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right\} - \beta g_i \theta \tag{2-42}$$

where $\Pi = p + 2\rho k/3$

$$\frac{\partial \theta}{\partial t} + \frac{\partial \theta u_j}{\partial x_j} = \frac{\partial}{\partial x_j}\left\{\left(\kappa + \frac{\nu_t}{\sigma_\theta}\right)\frac{\partial \theta}{\partial x_j}\right\} + h(x_i, t) \tag{2-43}$$

$$\frac{\partial c}{\partial t} + \frac{\partial c u_j}{\partial x_j} = \frac{\partial}{\partial x j}\left\{\left(D + \frac{\nu_t}{\sigma_c}\right)\frac{\partial c}{\partial x_j}\right\} + s(x_i, t) \tag{2-44}$$

Eddy viscosity

$$\nu_t = C_D \frac{k^2}{\varepsilon} \tag{2-45}$$

Conservation equation for turbulence kinetic energy and its dissipation rate

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_j}\left\{\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right\} + \nu_t\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_i}{\partial x_j} + \beta g_i \frac{\nu_t}{\sigma_\theta}\frac{\partial \theta}{\partial x_i} - \varepsilon \tag{2-46}$$

$$\frac{D\varepsilon}{Dt}=\frac{\partial}{\partial x_j}\left\{\left(\nu+\frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right\}+\frac{\varepsilon}{k}\left\{C_1\nu_t\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_i}{\partial x_j}-C_2\varepsilon+C_3\beta g_i\frac{\nu_t}{\sigma_\theta}\frac{\partial \theta}{\partial x_i}\right\} \qquad (2-47)$$

Empirical constants

$C_D = 0.09$ $\qquad$ $\sigma_k = 1.00$

$C_1 = 1.44$ $\qquad$ $\sigma_\varepsilon = 1.30$

$C_2 = 1.92$ $\qquad$ $\sigma_\theta = \sigma_c \sim 0.5, 0.7$

$C_3 \sim 0, C_1$

3. A Numerical Method for Solving System of Partial Differential Equations

The partial differential equations mentioned above are all non-linear and coupled with each other with the exception of transport equations for passive contaminants, which may be solved independently if the flow distribution is given. Only a numerical approach is possible to obtain solutions of these equations for most practical problems encountered in indoor airflow analysis. Moreover, although the handling of free surface boundary is required in some field of research, it is not necessary for the indoor airflow problems. Extensive effort has been made to date to develop appropriate algorithms to solve the Navier-Stokes equations and convection-diffusion transport equations numerically together with the treatment of free surface boundary condition. Among these, the method called the "Marker and Cell" method (MAC method)[32] developed at the Los Alamos Scientific Laboratory in 1965 has stood out due to its efficiency and potential applicability to general fluid flow phenomena. Although the original MAC method was developed for solving transient two-dimensional (2-D) liquid flow, including free surface boundary, many researchers have worked on this method to extend its applicability and generalization. The MAC method has become one of the standard solution techniques to solve 2-D as well as three-dimensional (3-D) fluid flow problems. If the problem is restricted only to the 2-D cases, the vorticity-stream function (χ-ψ)[33] approach can be used. In this case the continuity equation for the fluid is automatically satisfied and the pressure can be eliminated as a dependent variables.

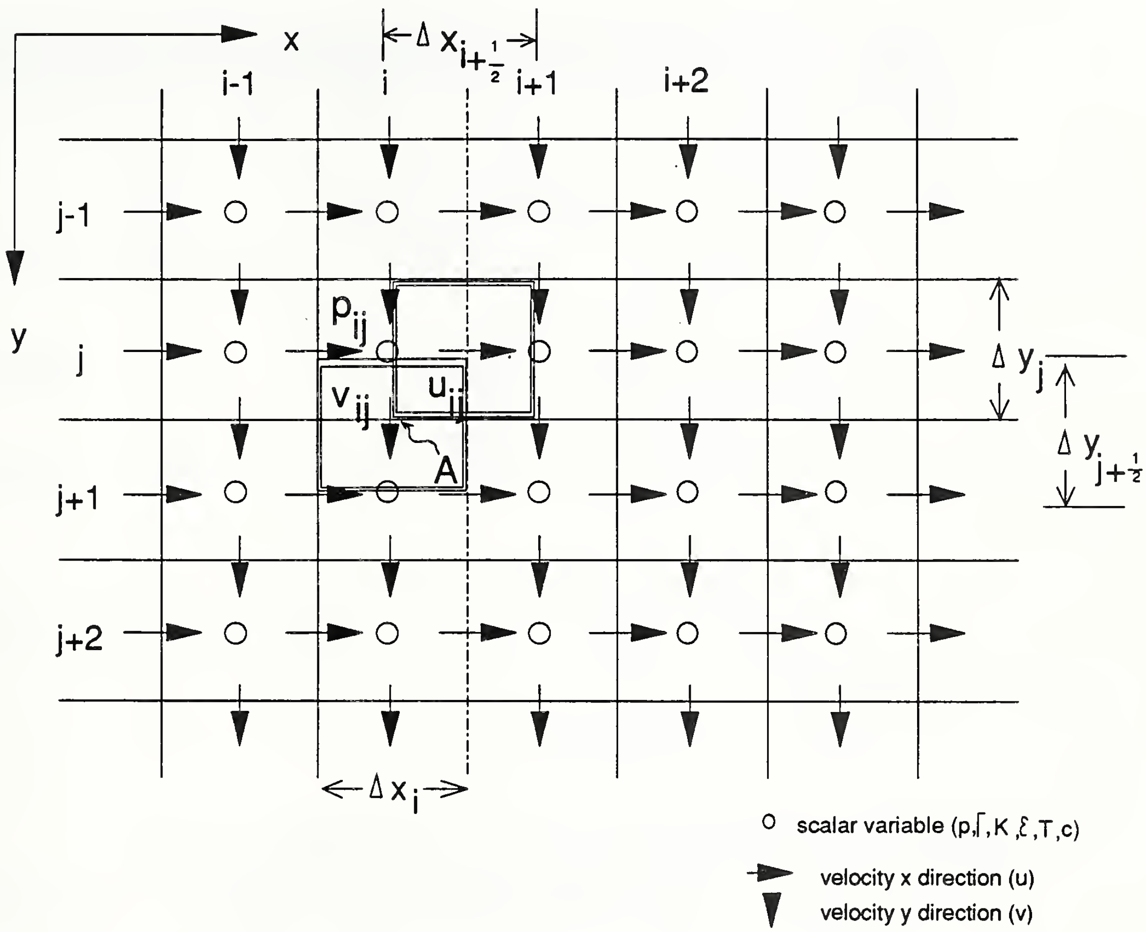
Although the (χ-ψ) method is advantageous for 2-D problems, its extension to 3-D problems is not an easy task. It also causes additional difficulty in the treatment of boundary conditions. Aside from this argument, 2-D and 3-D cases are not basically different from each other in the MAC method and the treatment of boundary conditions is straightforward since primitive variables are used.

In this chapter, the numerical method, which in principle is based on the MAC method, employed in the "EXACT" code is described.

3.1 Momentum Equation

In the MAC method, the flow domain to be considered is subdivided into rectangular grids called the "cell" and the velocity components u, v, pressure p and diffusivity/viscosity ν are defined at different locations shown in figure 3-1. (Here, for ecomony of notation, only the 2-D case is used to explain the numerical solution scheme; the extention to 3-D case is straightforward). In other words, velocity components are defined on the cell boundary normal to its cell face, while pressure is defined in the center of the cell. This mesh layout is called the "staggered mesh system".

Figure 3.1 Staggered Mesh System



The following conventions are used for this discretized system.

- each cell is numbered using a combination of two integer values, i and j for x and y direction respectively.
- i and j are also used as grid coordinate system:
  (i, j) indicates the center of cell i, j and an additional 1/2 the location on the cell boundary, for instance, (i,j + 1/2) specifies the point A in Figure 3-1.

- each dependent variable which is defined at the cell center (i,j) is expressed with subscript $_{i,j}$ such as $p_{i,j}$.
- velocity components defined at the cell boundary are also expressed with integer subscript: $u_{i,j}$ and $v_{i,j}$ are the velocity components defined at (i+1/2,j) and (i,j+1/2) respectively.
- Δx_i and Δy_j are the cell dimensions of the (i,j) cell in the x and y directions respectively. An additional subscript 1/2 indicates the arithematic mean of the cell interval in each direction (the distance between adjacent cell centers) $\Delta x_{i+1/2} = 0.5(\Delta x_i + \Delta x_{i+1})$

It must be emphasized here that $u_{i,j}$,$v_{i,j}$ and $p_{i,j}$ are all defined at different locations (the use of an integer subscript is merely for the convenience of coding).

The momentum transport equations for 2-D incompressible flow can be writtem in the following form:

$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}=-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left\{2\Gamma\frac{\partial u}{\partial x}\right\}+\frac{\partial}{\partial y}\left\{\Gamma\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right\}+Bx \tag{3-1}$$

$$\frac{\partial v}{\partial t}+\frac{\partial uv}{\partial x}+\frac{\partial v^2}{\partial y}=-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left\{\Gamma\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right\}+\frac{\partial}{\partial y}\left\{2\Gamma\frac{\partial v}{\partial y}\right\}+By \tag{3-2}$$

where

$$p=\frac{\Pi}{\rho} \quad , \quad \Gamma=\nu+\nu_t.$$

The equation of continuity is:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y} \tag{3-3}$$

For the convenience of discretization, equation 3-1, 2 are rearranged as follows:

$$\frac{\partial u}{\partial t}=-\frac{\partial}{\partial x}\left(u^2-\Gamma\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(uv-\Gamma\frac{\partial u}{\partial y}\right)-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left(\Gamma\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial v}{\partial x}\right)+Bx \tag{3-4}$$

$$\frac{\partial v}{\partial t}=-\frac{\partial}{\partial x}\left(vu-\Gamma\frac{\partial v}{\partial x}\right)-\frac{\partial}{\partial y}\left(v^2-\Gamma\frac{\partial v}{\partial y}\right)-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left(\Gamma\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial v}{\partial y}\right)+By \tag{3-5}$$

If Γ does not have any spatial variation as in the isothermal laminar case, the 4th and 5th terms in the right hand side of equations (3-4) and (3-5) vanish using the continuity equation. In order to treat the general case, these terms are left in this discussion. Each term in equation (3-4) may be interpreted as follows:

On the left hand side, the first term is rate of change of momentum. On the right hand side, the first and second terms are partial derivatives of momentum flux due to convective and diffusive transport in x and y directions. The third term is the net force due to the pressure gradient, the fourth and fifth terms are the net viscous shear stresses on on the cell and the sixth term is the gravity body force.

An approximate numerical form for each term of equation (3-4) is given below using a finite difference technique and staggered grid system with respect to $u_{i,j}$ with its surrounding rectangular gridspace (control volume) defined with (i,j-1/2) and (i+1,j+1/2) as the coordinates of the diagonally opposite corners. (The control volume for $v_{i,j}$ is defined as the volume with (i-1/2,j) and (i+1/2,j+1) as coordinates of the diagonally opposite corners.)

$$-\frac{\partial}{\partial x}\left(u^2-\Gamma\frac{\partial u}{\partial x}\right) \sim \frac{\{XFLUX_{i,j}-XFLUX_{i+1,j}\}}{\Delta x_{i+\frac{1}{2}}} \qquad (3-6)$$

$$-\frac{\partial}{\partial y}\left(uv-\Gamma\frac{\partial u}{\partial y}\right) \sim \frac{\left\{YFLUX_{i+\frac{1}{2},j-\frac{1}{2}}-YFLUX_{i+\frac{1}{2},j+\frac{1}{2}}\right\}}{\Delta y_j} \qquad (3-7)$$

$$-\frac{\partial p}{\partial x} \sim \frac{(p_{i,j}-p_{i+1,j})}{\Delta x_{i+\frac{1}{2},j}} \qquad (3-8)$$

$$\frac{\partial}{\partial x}\left(\Gamma\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial v}{\partial x}\right)+Bx \sim \frac{\left\{\Gamma_{i+1,j}\frac{u_{i+1,j}-u_{i,j}}{\Delta_{i+1}}-\Gamma_{i,j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x_i}\right\}}{\Delta x_{i+\frac{1}{2}}}$$

$$+\frac{\left\{\Gamma_{i+\frac{1}{2},j+\frac{1}{2}}\frac{v_{i+1,j}-v_{i,j}}{\Delta x_{i+\frac{1}{2}}}-\Gamma_{i+\frac{1}{2},j-\frac{1}{2}}\frac{v_{i+1,j-1}-v_{i,j-1}}{\Delta x_{i+\frac{1}{2}}}\right\}}{\Delta y_j}+Bx_{i,j} \qquad (3-9)$$

where $\Gamma_{i+1/2,j+1/2}$ is, for instance, interpolated value of Γ at (i+1/2,j+1/2) using

$\Gamma_{i,j}$, $\Gamma_{i+1,j}$, $\Gamma_{i,j+1}$ and $\Gamma_{i+1,j+1}$.

The terms XFLUX and YFLUX in equations (3-6) and (3-7) are momentum fluxes due to convection and diffusion across the control volume surfaces in x and y directions respectively. In the original MAC method, these terms were approximated by the central difference method as follows:

$$XFLUX_{i,j}=\left(\frac{u_{i,j}+u_{i-1,j}}{2}\right)\left(\frac{u_{i,j}+u_{i-1,j}}{2}\right)-\Gamma_{i,j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x_i} \qquad (3-10)$$

$$YFLUX_{i+\frac{1}{2},j+\frac{1}{2}} = \left(\frac{u_{i,j}+u_{i,j+1}}{2}\right)\left(\frac{v_{i,j}+v_{i+1,j}}{2}\right) - \Gamma_{i+\frac{1}{2},j+\frac{1}{2}} \frac{u_{i,j+1}-u_{i,j}}{\Delta y_{j+\frac{1}{2}}} \qquad (3-11)$$

With a uniform grid spacing, this formulation is second order in accuracy. It also makes the entire solution unstable or oscillating when convection is dominant (the cell Peclet number $Pe = u\Delta x/\Gamma$ becomes of the order of ten or more). The easiest remedy for this difficulty is to replace the central difference used in the convective term by the upwind difference scheme. The upwind form of equation 3-11 can be written as follows:

$$YFLUX_{i+\frac{1}{2},j+\frac{1}{2}} = u_{i,j}\left(\frac{v_{i,j}+v_{i+1,j}}{2}\right) - \Gamma_{i+\frac{1}{2},j+\frac{1}{2}} \frac{u_{i,j+1}-u_{i,j}}{\Delta y_{j+\frac{1}{2}}}$$

$$\text{when} \quad \left(\frac{v_{i,j}+v_{i+1,j}}{2}\right) \geq 0 \qquad (3-12)$$

$$YFLUX_{i+\frac{1}{2},j+\frac{1}{2}} = u_{i,j+1}\left(\frac{v_{i,j}+v_{i+1,j}}{2}\right) - \Gamma_{i+\frac{1}{2},j+\frac{1}{2}} \frac{u_{i,j+1}-u_{i,j}}{\Delta y_{j+\frac{1}{2}}}$$

$$\text{when} \quad \left(\frac{v_{i,j}+v_{i+1,j}}{2}\right) < 0 \qquad (3-13)$$

Although this treatment makes the numerical solution unconditionally stable, it also introduces an additional false diffusion/viscosity, which is proportional to $|\,(v_{i,j}+v_{i+1,j})\Delta y_{j+1/2}\,|$.

The "EXACT" code, therefore, employs a central/upwind combination scheme and switches from one scheme to another according to the local Peclet number in order to minimize the disadvantage in each scheme. The basic idea of this scheme originates from the well-known "Hybrid" scheme [34]. The flux $YFLUX_{i+1/2,j+1/2}$, for instance, is approximated as follows:

$$YFLUX_{i+\frac{1}{2},j+\frac{1}{2}} = \left(\frac{v_{i,j}+v_{i+1,j}}{2}\right)\left\{\left(\frac{u_{i,j}+u_{i,j+1}}{2}\right) + \left(\frac{u_{i,j}-u_{i,j+1}}{2}\right)\right\}$$

$$+ (u_{i,j}-u_{i,j+1}) \ll 0, \frac{\Gamma_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{j+\frac{1}{2}}} - \frac{v_{i,j}+v_{i+1,j}}{2C}, -\frac{v_{i,j}+v_{i+1,j}}{2C} \gg \qquad (3-14)$$

where $\ll \ldots \gg$ stands for the largest quantity in the enclosed list and C is a switching parameter: $C \leq 2$.

It is an easy task to prove that this formulation becomes identical with the central difference scheme (equation 3-11) when the local grid Peclet number $(Pe = (v_{i,j}+v_{i+1,j})\Delta y_{j+1/2}/2\Gamma_{i+1/2,j+1/2})$ satisfies the condition $-C \leq Pe \leq C$. Outside this range, this method reduces to a upwind interpolated scheme without the diffusion/viscosity term. The upwind interpolation factor is chosen so that the absolute value of the effective grid Peclet number, which takes account of an additional false diffusivity/viscosity due to truncation error, does not exceed C. Since the range and magnitude of the switching parameter can be specified arbitrarily, a different value of C can be used in different portions of the flow domain.

With the above approximation, equations (3-4) and (3-5) are finally approximated as follows:

$$\frac{\partial}{\partial t}(u_{i,j}) = \frac{p_{i,j} - p_{i+1,j}}{\Delta x_{i+\frac{1}{2}}} + (fu)_{i,j} \qquad (3-15)$$

$$\frac{\partial}{\partial t}(v_{i,j}) = \frac{p_{i,j} - p_{i,j+1}}{\Delta y_{j+\frac{1}{2}}} + (fv)_{i,j} \qquad (3-16)$$

where fu and fv are the approximate forms of the right hand side of equations (3-4) and (3-5) except for the pressure term.

In the original MAC method, the remaining partial derivative with respect to time is approximated by a simple explicit scheme. If the superscript n is used for the dependent variable $t = n\Delta t$, equations (3-15) and (3-16) become:

$${}^{n+1}u_{i,j} = {}^{n}u_{i,j} + \Delta t\left(\frac{{}^{n}p_{i,j} - {}^{n}p_{i+1,j}}{\Delta x_{i+\frac{1}{2}}}\right) + \Delta t\, {}^{n}(fu)_{i,j} \qquad (3-17)$$

$${}^{n+1}v_{i,j} = {}^{n}v_{i,j} + \Delta t\left(\frac{{}^{n}p_{i,j} - {}^{n}p_{ij+1}}{\Delta y_{j+\frac{1}{2}}}\right) + \Delta t\, {}^{n}(fv)_{i,j} \qquad (3-18)$$

This formulation allows the calculation of the velocity components at $t = (n + 1)\Delta t$ given the velocity and pressure field at $t = n\Delta t$. The velocity field at $t = (n + 1)\Delta t$, however, does not generally satisfy the continuity condition, unless the pressure field at $t = n\Delta t$ is choosen correctly. The velocity at $t = (n + 1)\Delta t$ and pressure field at $t = n\Delta t$ must, therefore, be updated simultaneously so that the continuity equation is satisfied. Finite difference forms of equation (3-3) in terms of the updated velocity components are:

$$\frac{{}^{n+1}u_{i,j} - {}^{n+1}u_{i-1,j}}{\Delta x_i} + \frac{{}^{n+1}v_{i,j} - {}^{n+1}v_{i,j-1}}{\Delta y_j} = 0 \qquad (3-19)$$

where the pressure ${}^{n}p_{i,j}$ is at $t = n\Delta t$ is choosen so that the continuity condition (3-19) is satisfied $t = (n + 1)\Delta t$.

In the original MAC method, a Poisson equation for pressure was solved so that the velocity field satisfies the continuity condition at $t = (n + 1)\Delta t$. The numerical Poisson equation is derived by substituting equations (3-17) and (3-18) into equation (3-19). The set of resultant linear algebraic equations for the pressure is then solved iteratively.

In contrast , the "EXACT" code employs the "pressure relaxation technique", which corrects the pressure and velocity components simultaneously so that the continuity error becomes minimal. Assume that tentative velocity components and pressure at each step of the iteration are expressed as ${}^{n+1}u_{i,j}^{\alpha}$, ${}^{n+1}v_{i,j}^{\alpha}$ and ${}^{n}p_{i,j}^{\alpha}$.

To start the iteration, set $\alpha = 0$ and the initial pressure iteration to the pressure of the last time step:

$$ {}^{n}p_{i,j}^{0} = {}^{n-1}p_{i,j} \qquad (3-20) $$

Then update velocity field using:

$$ {}^{n+1}u_{i,j}^{\alpha} = {}^{n}u_{i,j} + \Delta t\left(\frac{{}^{n}p_{i,j}^{\alpha} - {}^{n}p_{i+1,j}^{\alpha}}{\Delta x_{i+\frac{1}{2}}}\right) + \Delta t\, {}^{n}(fu)_{i,j} \qquad (3-21) $$

$$ {}^{n+1}v_{i,j}^{\alpha} = {}^{n}v_{i,j} + \Delta t\left(\frac{{}^{n}p_{i,j}^{\alpha} - {}^{n}p_{ij+1}^{\alpha}}{\Delta y_{j+\frac{1}{2}}}\right) + \Delta t\, {}^{n}(fv)_{i,j} \qquad (3-22) $$

The error in the continuity equation for this velocity field is then given by

$$ {}^{n+1}D_{i,j}^{\alpha} = \frac{{}^{n+1}u_{i,j}^{\alpha} - {}^{n+1}u_{i-1,j}^{\alpha}}{\Delta x_i} + \frac{{}^{n+1}v_{i,j}^{\alpha} - {}^{n+1}v_{i,j-1}^{\alpha}}{\Delta y_j} \qquad (3-23) $$

To reduce this error, the pressure is updated using

$$ {}^{n}p_{i,j}^{\alpha+1} = {}^{n}p_{i,j}^{\alpha} + {}^{n}\delta p_{i,j}^{\alpha} \qquad (3-24) $$

where

$$ {}^{n}\delta p_{i,j}^{\alpha} = -B\frac{{}^{n+1}D_{i,j}^{\alpha}}{L_{i,j}^{2}} \qquad (3-25) $$

where $1 \le B < 2$ to insure convergence.

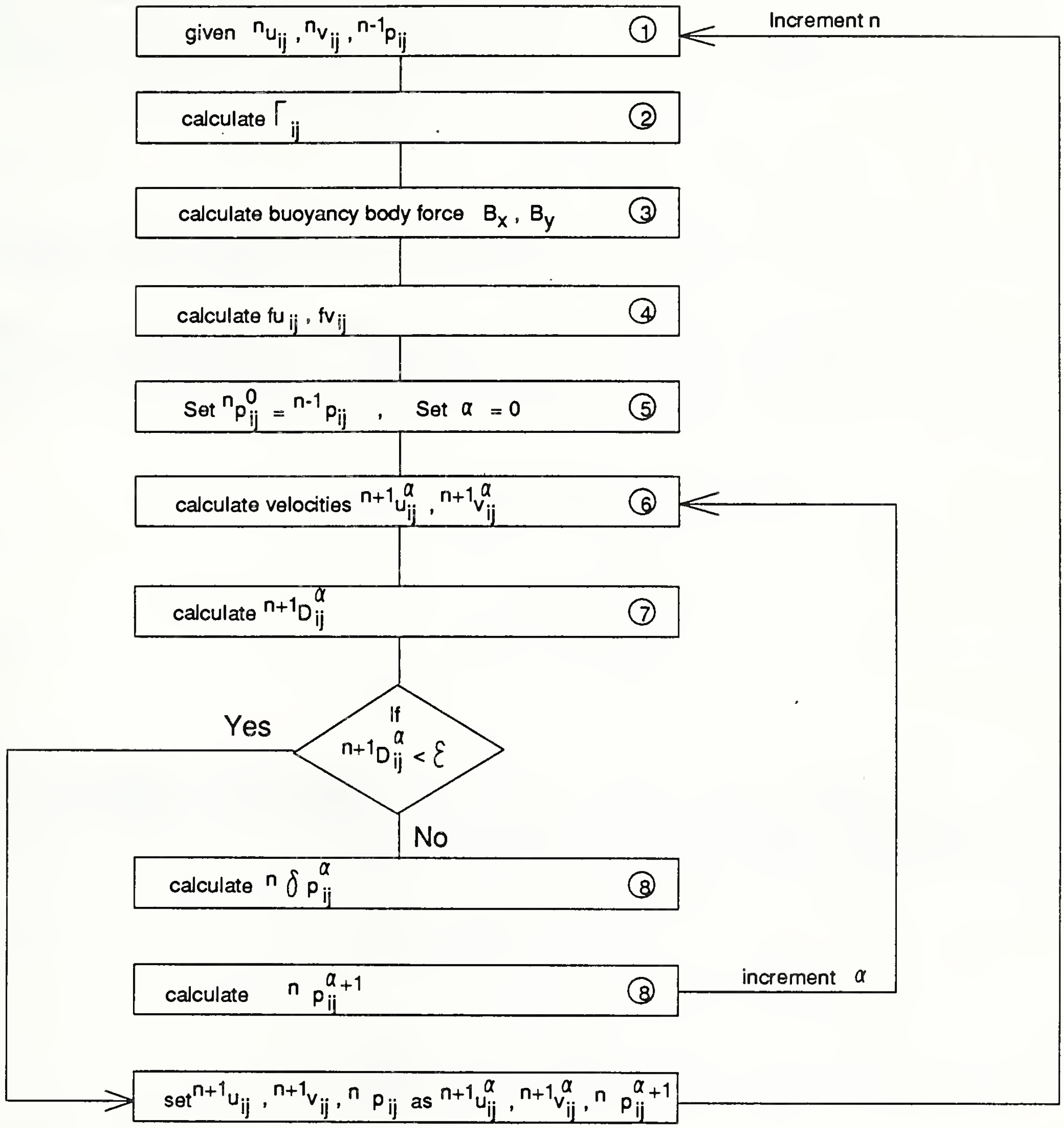
In equations (3-25)

$$ \frac{1}{L_{ij}^{2}} = \left\{ \frac{\frac{1}{\Delta x_{i+\frac{1}{2}}} + \frac{1}{\Delta x_{i-\frac{1}{2}}}}{\Delta x_i} + \frac{\frac{1}{\Delta y_{j+\frac{1}{2}}} + \frac{1}{\Delta y_{j-\frac{1}{2}}}}{\Delta y_j} \right\} $$

This process is repeated until the error in the continuity is reduced to an acceptable limit.

The solution procedure for the momentum equation is shown in fiqure 3-2.

Fiqure 3.2 Schematic of Solution Process for Momentum Equation



3.2 Scalar Transport Equations

In turbulent and/or buoyant flow conditions, additional transport equations, such the energy equation and k-ε equations in the two equation model, have to be solved in order to obtain spatial distribution of the effective diffusivity/viscosity or body force due to buoyancy. These equations are scalar transport equations and are similar in nature. The solution technique for the general scalar transport equation is adopted for them all. For simplicity the 2-D case is considered.

$$\frac{\partial \phi}{\partial t} = -\frac{\partial}{\partial x}\left(u\phi - \Gamma\frac{\partial \phi}{\partial x}\right) - \frac{\partial}{\partial y}\left(v\phi - \Gamma\frac{\partial \phi}{\partial y}\right) + S\phi + s \qquad (3-26)$$

Equation (3-26)becomes, for instance, the transport equation for turbulence kinetic energy in isothermal condition if $\phi = k, \Gamma = \nu + \nu_t/\sigma_k, S\phi = \nu_t S - \varepsilon$.

In the numerical solution method used in the "EXACT" code, the scalar valuable φ is defined at (i, j) point in the staggered grid system and expressed by $\phi_{i,j}$. For each term on the right hand side of (3-26) the following approximations for $\phi_{i,j}$ in the control volume: (i-1/2,j-1/2), (i+1/2,J+1/2).

$$-\frac{\partial}{\partial x}\left(u\phi - \Gamma\frac{\partial \phi}{\partial x}\right) \sim \frac{\left\{XFLUX_{i-\frac{1}{2},j} - XFLUX_{i+\frac{1}{2},j}\right\}}{\Delta x_i} \qquad (3-27)$$

$$-\frac{\partial}{\partial y}\left(v\phi - \Gamma\frac{\partial \phi}{\partial y}\right) \sim \frac{\left\{YFLUX_{i,j-\frac{1}{2}} - YFLUX_{i,j+\frac{1}{2}}\right\}}{\Delta y_j} \qquad (3-28)$$

$$S\phi \sim S\phi_{i,j} \qquad (3-29)$$

The terms *XFLUX* and *YFLUX* in equations (3-27) and (3-28) are the convection-diffusion fluxes of the faces of the control volume. They can be approximated in a similar manner as the corrsponding term in the momentum equation.

$$XFLUX_{i+\frac{1}{2},j} = u_{i,j}\left\{\phi_{i,j} + (\phi_{i,j} - \phi_{i+1,j}) \ll 0, \frac{\Gamma_{i+\frac{1}{2},j}}{\Delta x_{i+\frac{1}{2}}} - \frac{u_{i,j}}{2}, -u_{i,j} \gg\right\} \qquad (3-30)$$

$$YFLUX_{i,j+\frac{1}{2}} = v_{i,j}\left\{\phi_{i,j} + (\phi_{i,j} - \phi_{i,j+1}) \ll 0, \frac{\Gamma_{i,j+\frac{1}{2}}}{\Delta y_{j+\frac{1}{2}}} - \frac{v_{i,j}}{2}, -v_{i,j} \gg\right\} \qquad (3-31)$$

where the switching parameter C is fixed at 2 in order to guarantee that these fluxes are always positive for k, ε or the contaminant concentration c.

Equation (3-26) can be written in the form given in equation (3-32)

$$\frac{\partial}{\partial t}(\phi_{i,j}) = (f\phi)_{i,j}$$

where

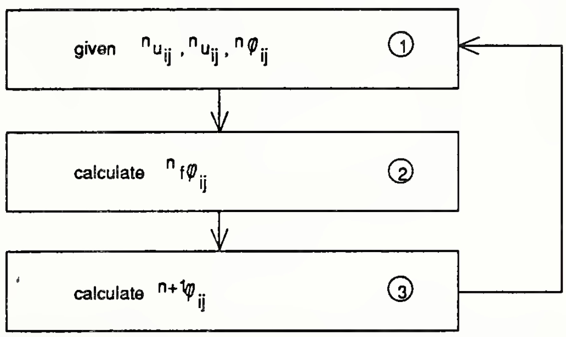
$$(f\phi)_{i,j} = \frac{\left\{XFLUX_{i-\frac{1}{2},j} - XFLUX_{i+\frac{1}{2},j}\right\}}{\Delta x_i} + \frac{\left\{YFLUX_{i,j-\frac{1}{2}} - YFLUx_{i,j+\frac{1}{2}}\right\}}{\Delta y_j} + S\phi_{i,j} \qquad (3-32)$$

A simple explicit formulation allows the calculation of φ at $t = n\Delta t$ in equation (3-39).

$$^{n+1}\phi_{i,j} = {}^{n}\phi_{i,j} + \Delta t\, {}^{n}(f\phi)_{i,j} \qquad (3-33)$$

The solution procedure for scalar transport equation can be summarized as follows.

Figure 3.3 Schematic of Solution for Scalar Equations

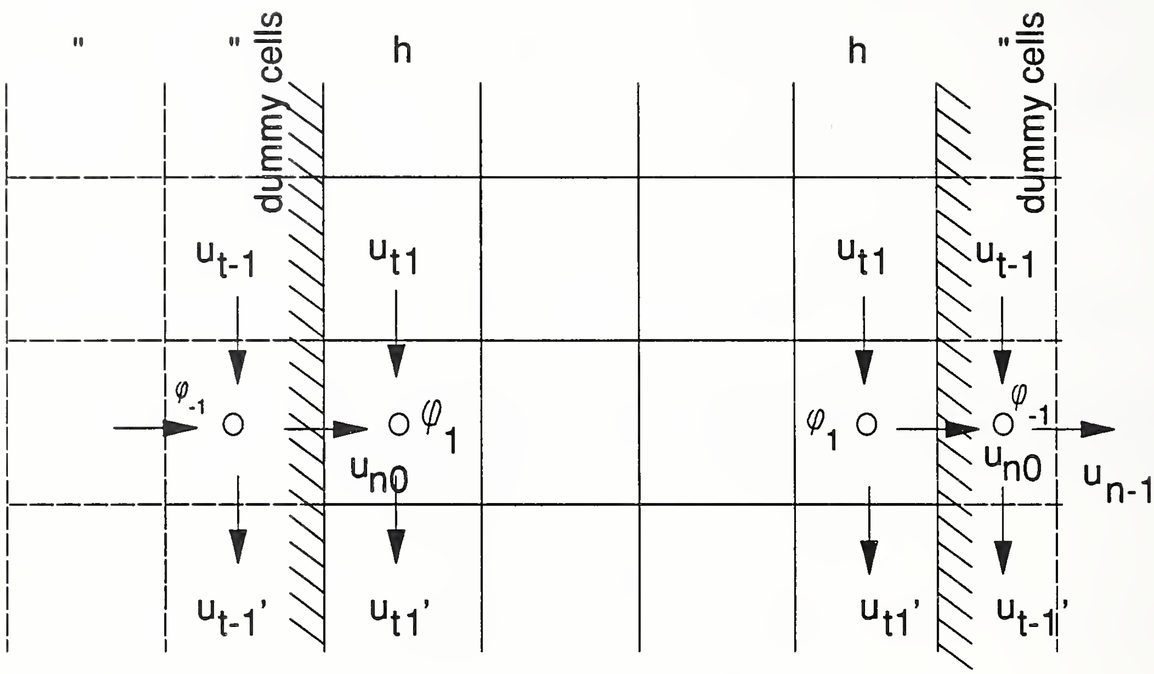


3.3 Boundary Conditions

The k-ε turbulence model consists of a system of elliptic partial differential equations. The dependent variables in them require appropriate boundary conditions. These are usually expressed as conditions on the primitive variables except for the pressure. The pressure boundary condition is required only when the velocity component normal to the boundary is unknown and has to be determined through calculation.

In the staggered grid approach, the boundary location is designed so as to coincide with the end face of a real cell, and one or two cells with the same cell interval as that of terminal real cell are added just outside it to handle the boundary conditions. The velocity component normal to the boundary is, therefore, located just on the boundary, while the lateral component and other scalar variables are defined at a half cell interval detached from the boundary as shown in Figure 3-4.

Figure 3-4 Mesh Layout Near Boundary



By using symbols and subscripts shown in figure 3-4, typical boundary conditions required in the indoor airflow analysis are described as follows.

- **Inflow condition**

This is used to specify the inflow information where there is mass transfer from outside to inside the calculation domain.

u_{n0}	specified
u_{t-1}, u_{t-1}'	specified (usually = 0)
$k_{-1}, \varepsilon_{-1}, \theta_{-1}$	specified

Since the inflow values of k and ε are not usually known except when measurements are undertaken, they are determined with reference to available data for similar conditions. Recommended values are given as follows from Kato's experiments [35]

- for a straight duct end or for a nozzle type jet

$$k = 0.01\text{\textasciitilde}0.03U_0^2 \quad , \quad l = 0.05\text{\textasciitilde}0.25L_0$$

- for the "anemo type" radial diffuser

$$k = 0.05 \sim 0.15 U_0^2 \quad , \quad l = 0.01 \sim 0.02 L_0$$

where

U_0 is the area averaged inflow velocity

L_0 is the representative length of intake.

The value of ε is calculated from equation (2-16).

• **outflow condition**

Although this is the opposite condition of inflow case, most outflow values are not given a priori. Hence an approximate condition is imposed using engineering judgment:

Case 1 Known Outflow

u_{n0}	specified
u_{t-1}, u_{t-1}'	specified (usually = 0) or set $u_{t-1} = u_{t1}, u_{t-1}' = u_{t1}'$ assuming $\frac{\partial u_t}{\partial y} = 0$
$k_{-1}, \varepsilon_{-1}, \theta_{-1}$	$k_{-1} = k_1$ assuming $\frac{\partial k}{\partial y} = 0$ $\varepsilon_{-1} = \varepsilon_1$ assuming $\frac{\partial \varepsilon}{\partial y} = 0$ $\theta_{-1} = \theta_1$ assuming $\frac{\partial \theta}{\partial y} = 0$

Case 2 Unknown outflow

u_{n0}	not specified
u_{t-1}, u_{t-1}'	specified (usually = 0) or set $u_{t-1} = u_{t1}, u_{t-1}' = u_{t1}'$ assuming $\frac{\partial u_t}{\partial y} = 0$
u_{n-2}	adjusted so that the continuity condition at the -1 cell is satisfied using updated v_{no}, v_{t-1} and v_{t-1}'
$k_{-1}, \varepsilon_{-1}, \theta_{-1}$	$k_{-1} = k_1$ assuming $\frac{\partial k}{\partial y} = 0$ $\varepsilon_{-1} = \varepsilon_1$ assuming $\frac{\partial \varepsilon}{\partial y} = 0$ $\theta_{-1} = \theta_1$ assuming $\frac{\partial \theta}{\partial y} = 0$
p_{-1}	specified (usually=0)

The last pressure boundary condition is useful when a significant variation of outflow velocity is expected, and yet it is not known prior to the calculation.

• **symmetry condition**

This condition is used to reduce unknown variables when the flow domain contains a plane of symmetry.

u_{n0}	specified as $u_{n0} = 0$.
u_{t-1}, u_{t-1}'	set $u_{t-1} = u_{t1}, u_{t-1}' = u_{t1}'$ because $\frac{\partial u_t}{\partial y} = 0$
$k_{-1}, \varepsilon_{-1}, \theta_{-1}$	$k_{-1} = k_1$ because $\frac{\partial k}{\partial y} = 0$
	$\varepsilon_{-1} = \varepsilon_1$ because $\frac{\partial \varepsilon}{\partial y} = 0$
	$\theta_{-1} = \theta_1$ because $\frac{\partial \theta}{\partial y} = 0$

• **wall boundary condition 1 (logarithmic law type)**

Since the whole flow domain, except for the intake and outlet, is surrounded by walls (floor and ceiling) as far as the indoor airflow is concerned, numerical wall boundary conditions are very important for obtaining accurate results. The most straightforward boundary condition is to set all velocity component on the wall to zero following the theoretical requirement that there is no slippage at the wall. In spite of plausibility of this method, its direct application requires an excessively a fine grid layout in the vicinity of the wall. This is because near the wall the flow is less turbulent and diffuse with the gradient of each dependent variable becoming much steeper in this region than in flow domain away from the wall.

In addition, low turbulence is inconsistent with the fundamental assumptions of the k-ε turbulence model. A more approximate treatment is needed for both theoretical and economical reasons. The most commonly used wall boundary condition is the so called logarithmic wall condition. It assumes that the flow has a zero pressure gradient in the cells adjacent to the wall. This leads to an equation for the flow near the wall of the form:

$$\frac{u_t}{u*} = \frac{1}{\kappa}\ln\frac{yu*}{\nu} + C \qquad (3-34)$$

where

C	empirical constant (for a smooth wall ~5.5)
u	velocity component parallel to wall
u^*	the friction velocity = $(\tau/\rho)^{\frac{1}{2}}$
y	the distance from the wall
κ	an empirical constant (Karman constant ~0.4)

If equation (3-34) is applied at y = h/2 in figure 3-4, the following implicit formulation is obtained for friction velocity u^*

$$\frac{u_{t1}}{u*} = \frac{1}{\kappa}\ln\frac{hu*}{2\nu} + C \qquad (3-35)$$

u^* can be calculated using any available iteration procedure such as the Newton method. If the partial derivative with respect to y is taken in equation (3-34), the following relation is obtained assuming constant a viscous stress layer:

$$u*^2 = \frac{\tau}{\rho} = \kappa u u* \frac{\partial u_t}{\partial y} \tag{3-36}$$

This is consistent with the Prandtl's mixing length hypothesis and ν_t is evaluated by

$$\nu_t = \kappa y u* \tag{3-37}$$

k_1, is calculated using equation (2-33), resulting the following equation:

$$\kappa_1 = C_D^{-\frac{1}{2}} u*^2 \tag{3-38}$$

ε_1 can be obtained from equations (3-47), (3-38_ and (2-17) as follows:

$$\varepsilon_1 = \frac{C_D^{\frac{3}{4}} \kappa_1^{\frac{3}{2}}}{\frac{\kappa h}{2}} \tag{3-39}$$

Finally, the whole procedure is summarized as follows:

set $u_{n0} = 0$
obtain u^* by solving (3-35)
determine k_l and ε_1 using equations (3-38) and (3-39)
determin u_{t-1} using

$$u*^2 = (\nu + \nu_t) \frac{u_{tl} - u_{t-1}}{h} \tag{3-40}$$

where equation (3-40) is a finite difference form of equation (3-36) in which kinematic viscosity has been added.

• **boundary condition 2 (power law type)**

Logarithmic wall boundary condition may be feasible as far as the relation given in equation 3-40 is correct. This is, however, merely an approximation of actual near wall flow and requires an iterative procedure, which may be disagreeable for practical calculation. An alternative wall boundary condition may be derived from a crude assumption of the velocity profile in the vicinity of wall. Experimental data show that the following power law representation is a good approximation for constant stress layer:

$$u_t = u_\delta \left(\frac{y}{\delta} \right)^n \tag{3-41}$$

If equation 3-41 is applied at $y=h/2$ in figure 3-4, the following equation is given:

$$u_{tl} = u_\delta \left(\frac{y}{2\delta} \right)^n \tag{3-42}$$

Equation 3-41 is transformed by dividing both sides by equation 3-42 as follows:

$$\frac{u_t}{u_{tl}} = \left(\frac{2y}{h}\right)^n \qquad (3-43)$$

By taking partial derivative of equation (3-44) with respect to y, one obtains:

$$\frac{\partial u_t}{\partial y} = u_{t1}\frac{2n}{h}\left(\frac{2y}{h}\right)^{n-1} \qquad (3-44)$$

The gradient of u_t at y=h/2 can be calculated using equation 3-44 as follows:

$$\frac{\partial u_t}{\partial y} = u_{tl}\frac{2n}{h} \qquad y = \frac{h}{2} \qquad (3-45)$$

The one sided finite difference form of the left hand side of equation 3-45 between y=h/2 and y=-h/2 yields the following equation:

$$u_{t1} - u_{t-1} = 2nu_{t1} \qquad (3-46)$$

The physical meaning of equation 3-46 is the extrapolation of u_t to the wall boundary and it can be used to set the value of u_{t-1}.
The boundary condition for k may be derived from equation (2-30) as follows:

$$k_{-1} = k_1 \text{assuming} \quad \frac{\partial k}{\partial y} = 0$$

The boundary condition for ε_1 is given by equation (3-39) as in the logarithmic law condition. The parameter n in these equations may be choosen as constant ($n = 1/6 \sim 1/8$ for fully developed wall boundary layer) or as a function of turbulence information near wall such as ν_{t1} when a laminar flow region exist near the wall.

• **wall boundary condition 3 (for energy equation)**

The wall boundary condition for energy equation is very important to the entire accuracy of a buoyant flow calculation because it exerts a significant effect on the total heat loss/gain at the wall and also on the buoyancy driving force in momentum equation. Although the heat flux specified wall boundary condition may be set without difficulty, the temperature specificaton condition must be transformed to the flux type similar to the estimation of wall shear stress or velocity gradient in momentum equation. This is nothing more than the estimation of the local heat transport coefficient. The most straightforward and reasonable way to accomplish this would be to impose a similar empirical information as equation (3-35) or (3-45) for temperature. It has been shown, however, that a direct application of this approach causes significant underestimation of heat fluxes. An ad hoc treatment has always been needed to obtain acceptable results. The following tentative procedure is adapted in the "EXACT" code. Wall heat flux can be related to the temperature gradient at y=h/2 in figure 3-4 assuming a constant flux layer as follows:

$$q_w = \frac{H_w}{C_p\rho} = \left(\kappa + \frac{\nu_t}{\sigma_\theta}\right)\frac{\partial\theta}{\partial y} \qquad (3-47)$$

where

C_ρ	specific heat
q_w	wall thermal flux
H_w	wall heat flux
ρ	density
κ	thermal diffusivity

One sided finite difference approximation of the right hand side derivative between y=h/2 to y=0 yields equation (3-48) assuming constant ν_t:

$$q_w = \frac{H_w}{C_p \rho} = \left(\kappa + \frac{\nu_t}{\sigma_\theta} \right) \frac{2}{h} (\theta_1 - \theta_0) \tag{3-48}$$

where

θ_0 is the given wall temperature

The right hand term in the bracket in equation (3-48) represents local thermal transfer coefficient (α) and it is expected to vary little depending on h (cell interval) as long as a sufficiently large value for h is taken. This is because θ_1 varies little if it is defined at nearly boundary layer thickness. If a reasonable value of α is assumed from available experimental data the following equation may be used to calculate approximate thermal flux:

$$q_w = \frac{H_w}{C_p \rho} = \alpha(\theta_1 - \theta_0) \tag{3-49}$$

An approximated form of equation (3-47) using the one sided finte differencing scheme between y=h/2 and y=-h/2 is given as follows, again assuming constant ν_t:

$$q_w = \frac{H_w}{C_p \rho} = \left(\kappa + \frac{\nu_t}{\sigma_\theta} \right) \frac{(\theta_1 - \theta_{-1})}{h} \tag{3-50}$$

Elimination of the left hand side of equation (3-49), (3-51) yields the following equation:

$$\alpha(\theta_1 - \theta_0) = \left(\kappa + \frac{\nu_t}{\sigma_\theta} \right) \frac{(\theta_1 - \theta_{-1})}{h} \tag{3-51}$$

Equation (3-51) is used to determine an appropriate value of θ_{-1} when the boundary temperature is given. Otherwise equation (3-50) can used to obtain an appropriate value of θ_{-1} when the wall flux is given.

4. Experimental Validation for Isothermal Flows

In this chapter, several examples of numerical results using the "EXACT" code are presented for which experimental data are available for isothermal conditions. The numerical time iteration is carried out on the Cyber 205 vector processor until the temporal solution reaches steady-state, which takes 10 to 90 minutes of CPU time depending on the problem. Convergence is evaluated by using the volumetric root mean square residual(r) of each partial differential equation as follows.

For k, ε and the continuity equation

$$r_k, \quad r_\varepsilon, \quad r_c \leq 10^{-5}$$

For the momentum equations

$$r_m \leq 6 \times 10^{-5}$$

In each case, r.m.s. residuals deceased monotonically and no numerical instability nor oscillation was observed. After a converged solution was obtained, it was stored in a file on the Cyber 855 and appropriate two-dimenional or three-dimensional graphical presentation of the results was made using post-processing graphic routines on "TEMPLATE".

4.1 Sakamoto's geometry

The first geometry to be considered is a 2 x 2 x 2 m cubic box with the same size of rectangular inlet and outlet located in the middle of the ceiling and at the bottom corner of a vertical wall as shown in figure 4-1. This geometry was employed by Sakamoto et al[36] both for experiment and numerical calculation. A full scale model room with an air inlet and outlet was constructed of wooden particle boards. Mean velocity and some statistical quantity distribution were measured with an ultrasonic anemometer. The Reynolds number, based on the area average inlet/outlet velocity U_0, rectangular inlet/outlet nozzle size parameter L_0 and kinematic viscosity of air ν is approximately $1.2 \cdot 10^5$. Corresponding to this condition, the numerical prediction was performed with 18x18x9 uniform mesh layout for a symmetric half of the flow domain. (2x2 meshes were used for inlet and outlet).

The converged solution was obtained using "EXACT" code without special modification. Detailed information is given as follows:

time step	$0.1\ L_0/U_0$
inflow streamwise velocity	U_0 (uniform)
inflow lateral velocity	0 at upwind cells
outflow streamwise velocity	U_0 (uniform)
outflow lateral velocity	0 at upwind cells
inflow turbulence energy	$k = 0.005\ U_0^2$ $\varepsilon = 0.00032\ U_0^2/L_0$
inflow length scale	$0.1\ L_0$
wall boundary condition	1/7 th power profile
switching parameter for convection term	2 for whole domain

convergent criteria $r_{k,} r_{\varepsilon,} r_c < 10^{-5}$
after 10 min CPU time $r_m \sim 2 \cdot 10^{-5}$

The calculation result and Sakamoto's experiment are compared in figure 4-2 and 4-3 on the plan of symmetry. Figure 4-3 shows trajectories of passive markers from arbitrary starting points based on the measured mean velocity field. The numerical result shows that the inflow goes straight downward, diverges horizontally (radially) after impinging on the floor and the left side portion of the horizontal flow deflects upward up to a certain distance along the left side vertical wall. It may be interesting to note that a strong recirculating zone appears only in the left side of the flow domain with its center at the near bottom corner.

On the other hand, the right side of the flow domain is comprised of a downward flow and is not recirculating. These characteristics can be clearly observed in the experiment and the general feature of the mean flow corresponds fairly well. The main discrepancy between calculation and experiment seems to occur just below the inlet and near the left wall. The experiment shows that the initial inflow jet diffuses more rapidly than the calculation and no upward flow in the vicinity of the left wall is observed in the experiment.

Sakamoto et al suggested that the anemometer probe they used in the experiment has a 10 cm span and may not be suitable for velocity measurement where steep velocity gradient existed within a zone comparable to the probe span. Since the steep velocity gradient usually appeared in the vicinity of air inlet and wall, this discrepancy might have been originated from inaccuracy of the measured data. Predicted mean flow structure is given in a perspective view of figure 4-4, where velocity vectors on the symmetric half plane in the vicinity of the ceiling, flow,and two side walls and those on the plane of symmetry are plotted. The flow direction near the vertical walls is generally upward except the downward flow over the exhaust outlet. The near floor flow shows radial dispersion of the main jet and the near ceiling flow is induced toward the inlet. This flow structure is consistent with the measured mean flow distribution in other planes.

Figure 4.1 Sakamoto's Geometery and Experimental Apparatus
(dimensions in mm)

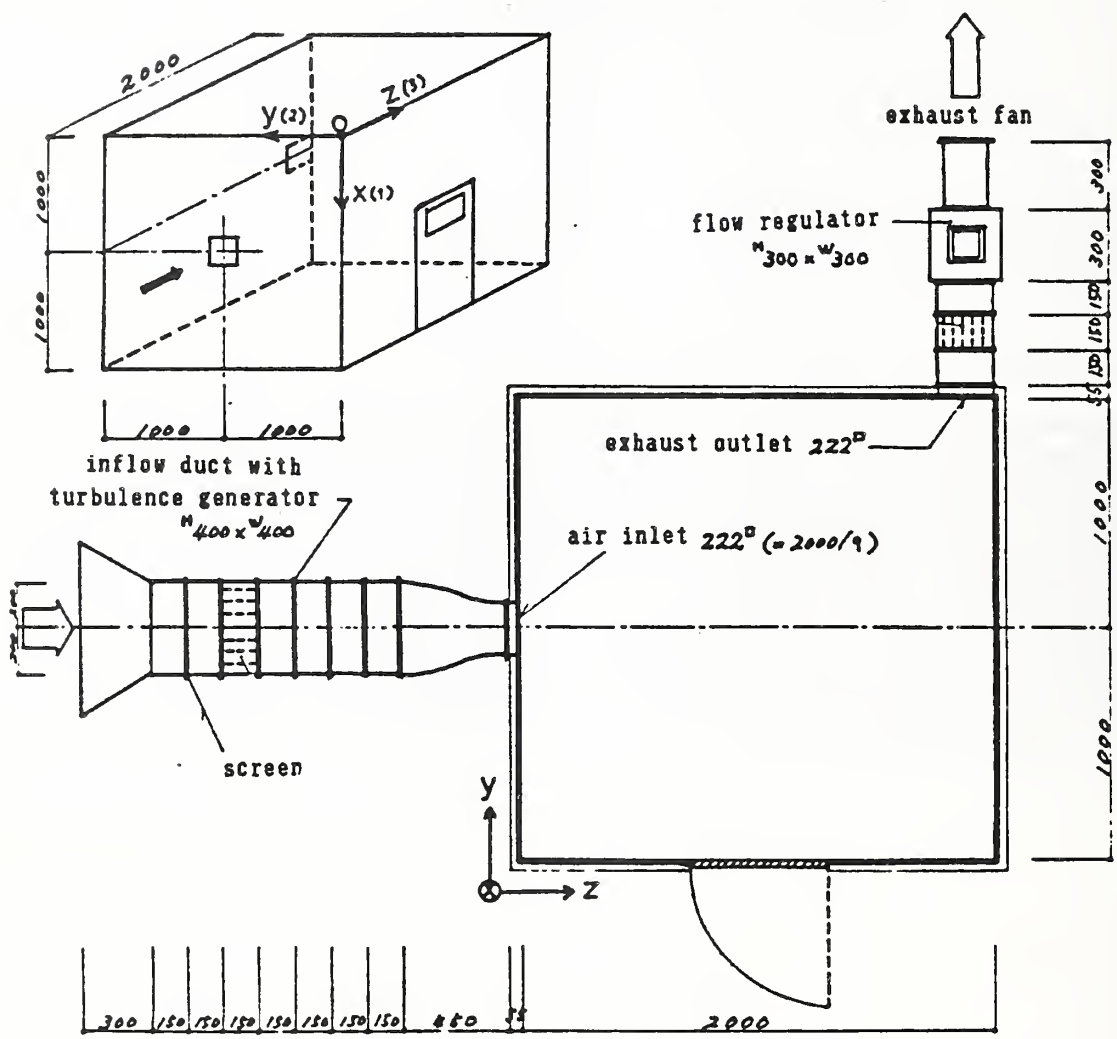


Figure 4.2 Predicted Time Average Velocity Vectors on Plane of Symmetry

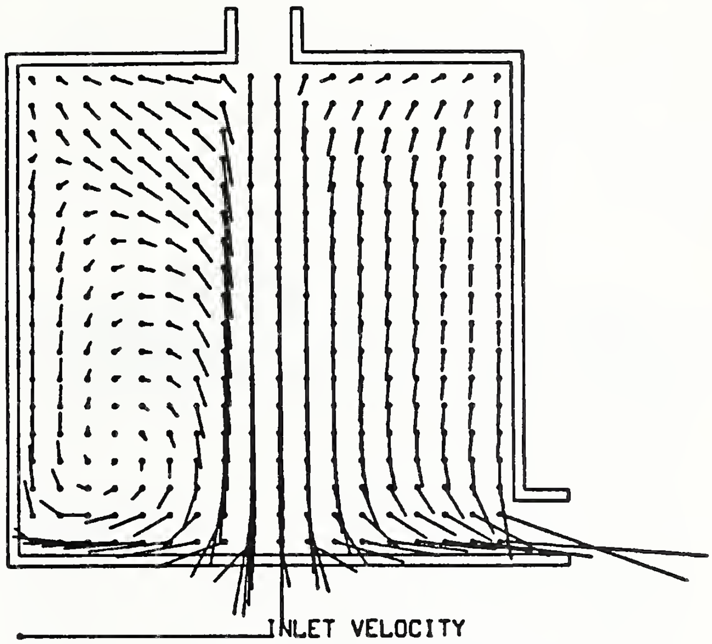


Figure 4.3 Trajectories of Passive Markers Based on Experimental Data

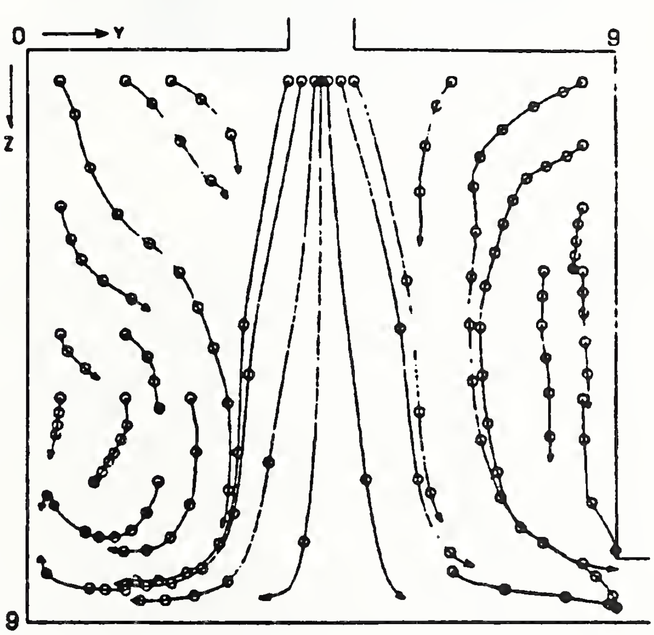
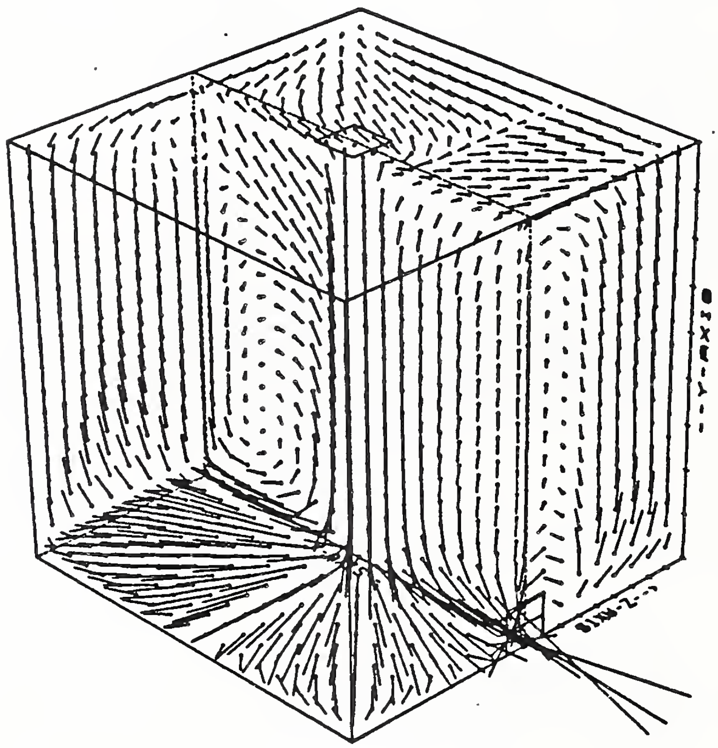


Figure 4.4 Perspective View of Predicted Mean Flow Structure



4.2 Kato's geometry

The next model is also a cubic box of 1500 x 1500 x 1500 mm dimensions with a rectangular air inlet and exhaust outlet of the same size located on the same wall as shown in figure 4-5. Numerical and experimental studies were conducted using this geometry by Kato et al [37]. An experimental study was carried out with a full-scale model constructed of clear acrylic boards with an air circulation system. They made detailed measurements for mean velocity as well as its fluctuating component using the tandem special hot wire anemometer, which can measure both instantaneous velocity and direction. The anemometer sensor they used is less than 5 mm in size and expected to measure velocity with a fair accuracy even where a steep gradient exists.

The Reynolds number of the inflow condition was maintained approximately at 60.000 and the flow was expected to be fully turbulent. The numerical prediction was carried out using a 20x20x10 uniform mesh layout (2x2 meshes for inlet/outlet) for the symmetric half flow domain and the following condition was imposed to obtain converged solutions.

time step	$0.1\ L_0/U_0$
inflow streamwise velocity	U_0 (uniform)
inflow lateral velocity	0 at upwind cells
outflow streamwise velocity	U_0 (uniform)
outflow lateral velocity	0 at upwind cells
inflow turbulence energy	$k = 0.005\ U_0^2$ $\varepsilon = 0.00032\ U_0^2/L_0$
inflow length scale	$0.1\ L_0$
wall boundary condition	1/7 th power profile
switching parameter for convection term	2 for whole domain
convergent criteria after 30 min CPU time	$r_k, r_\varepsilon, r_c < 10^{-5}$ $r_m \sim 2 \cdot 10^{-5}$

Predicted and measured time-averaged velocity vectors on the plane of symmetry are presented in figures 4-6 and 4-7 respectively. Predicted velocity vectors show that inflow becomes a wall jet near the ceiling, gradually induces surrounding low velocity fluid and deflects downward after impinging onto the opposite wall. Most of the main flow is a low speed diagonal return flow after reaching the floor and a recirculating flow structure appears in the whole flow domain with its center near the bottom corner of the opposite wall. The agreement with the corresponding experiment is excellent not only for the general flow structure but also for the flow velocity and direction at each point. The predicted and measured distribution of turbulence kinetic energy are also compared in figures 4-8 and 4-9. The agreement between these is not so good as that of time-averaged velocity distribution. Relatively high turbulence kinetic energy regions occur in the vicinity of the ceiling and the opposite wall in both prediction and experiment, and their orders of magnitude seem to correspond fairly well. Predicted mean flow structure is presented in Figure 4- 10 in a similar manner as the previous case. Small secondary recirculating regions appear near the opposite corner of the ceiling as well as at the top corner of the front wall, and these subtle flow structures are also observed during the measurement at corresponding locations.

Figure 4.5 Kato's Geometry and Experimental Apparatus
(dimensions in mm)

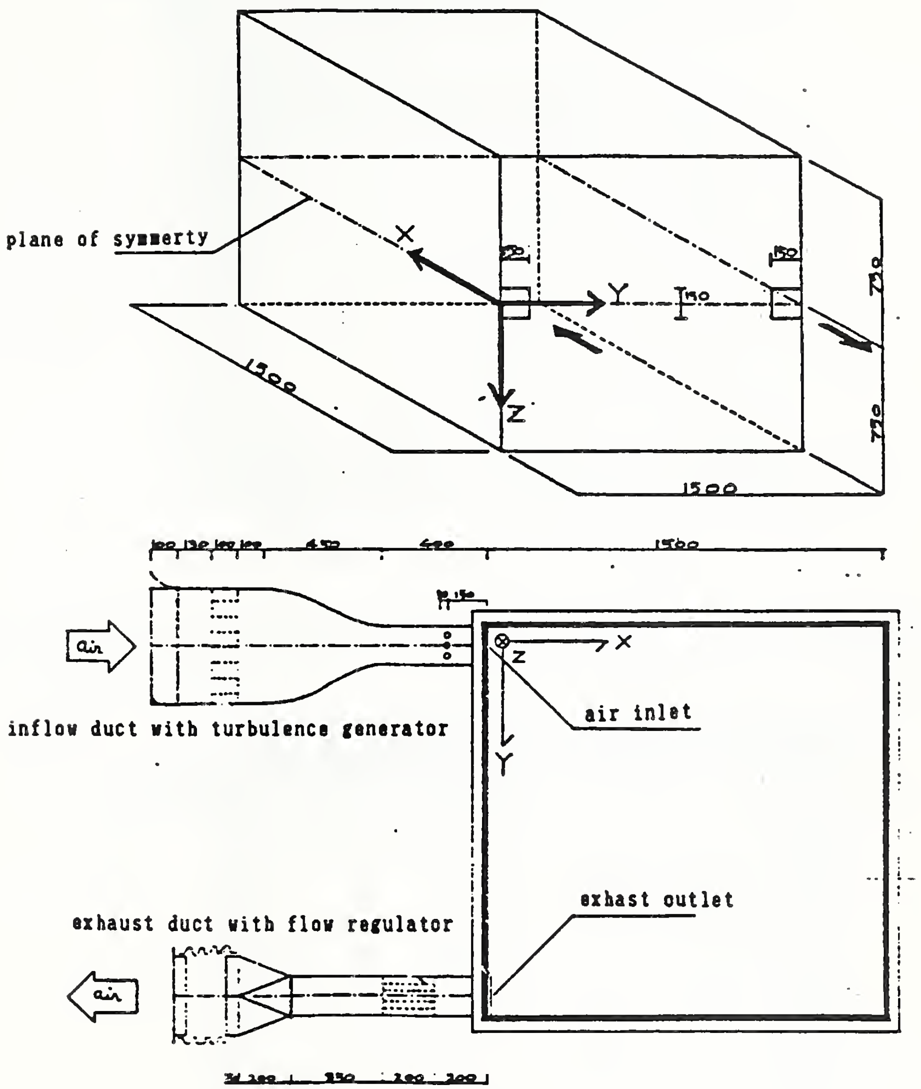


Figure 4.6 Predicted Average Velocity Vectors on Plane of Symmetry

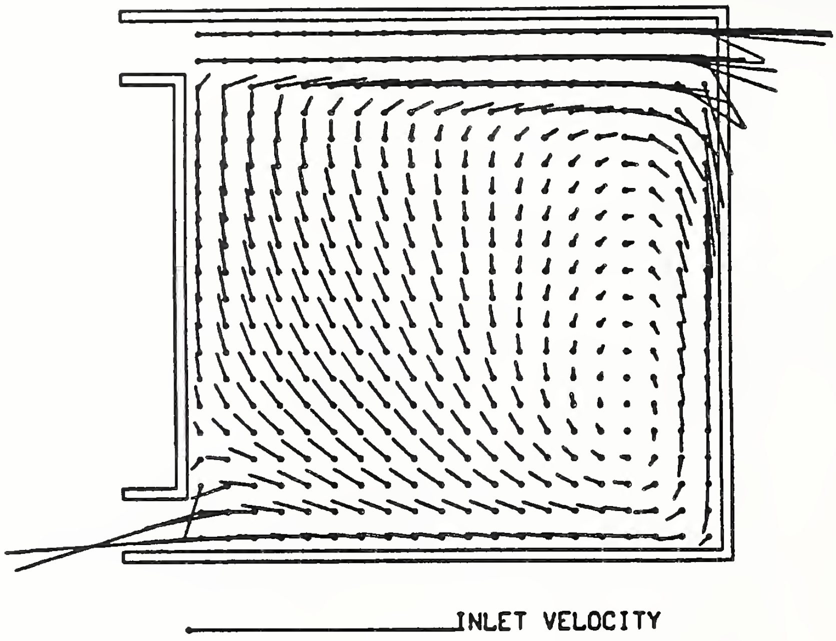


Figure 4.7 Measured Average Velocity Vectors on Plane of Symmetry

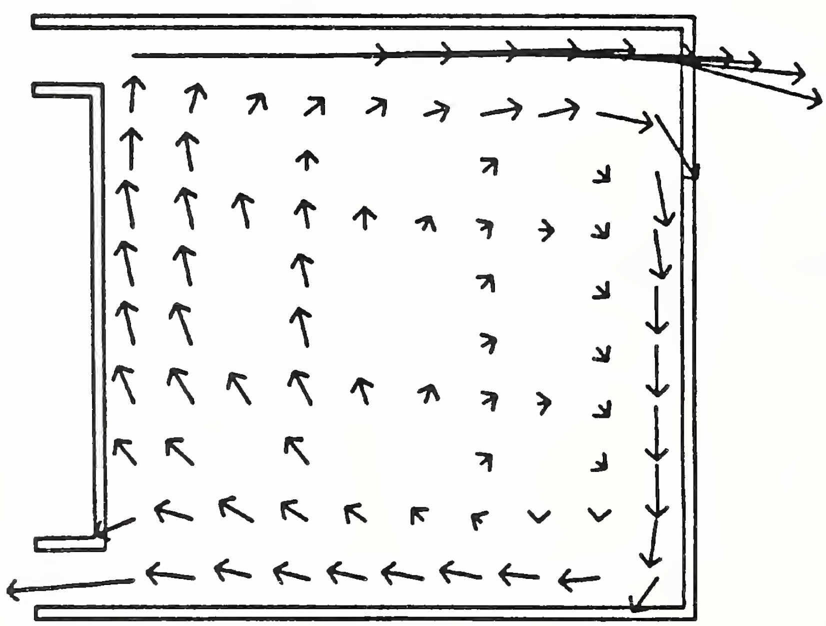


Figure 4.8 Predicted Distribution of Turbulent Kinetic Energy on Plane of Symmetry

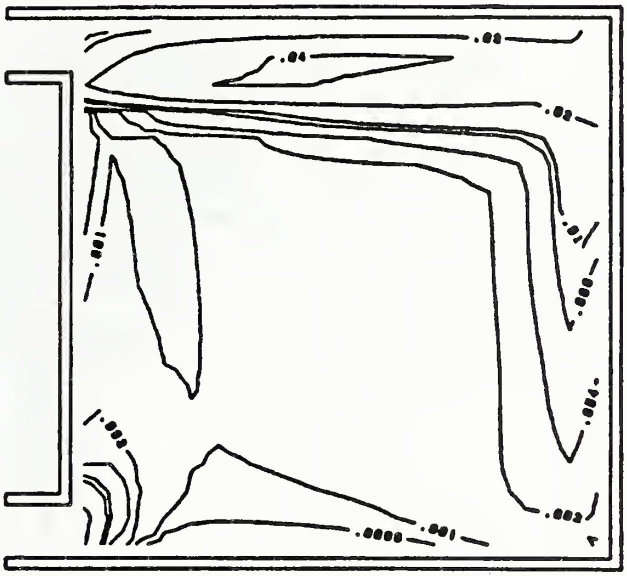


Figure 4.9 Measured Distribution of Turbulent Kinetic Energy on Plane of Symmetry

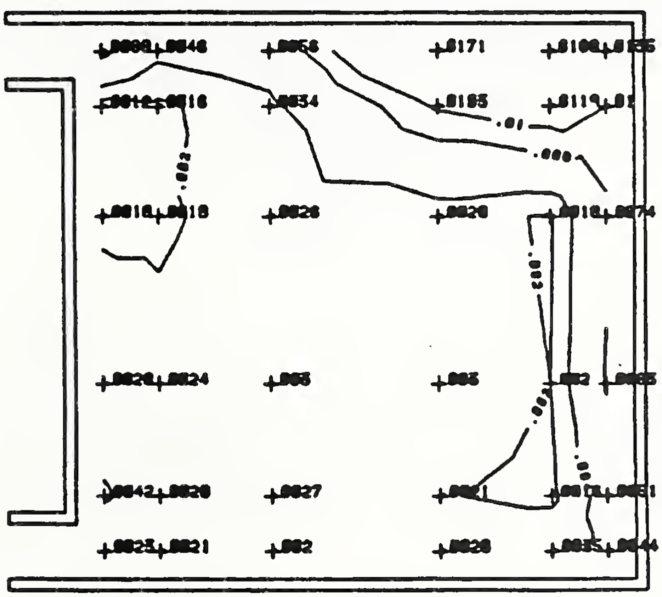
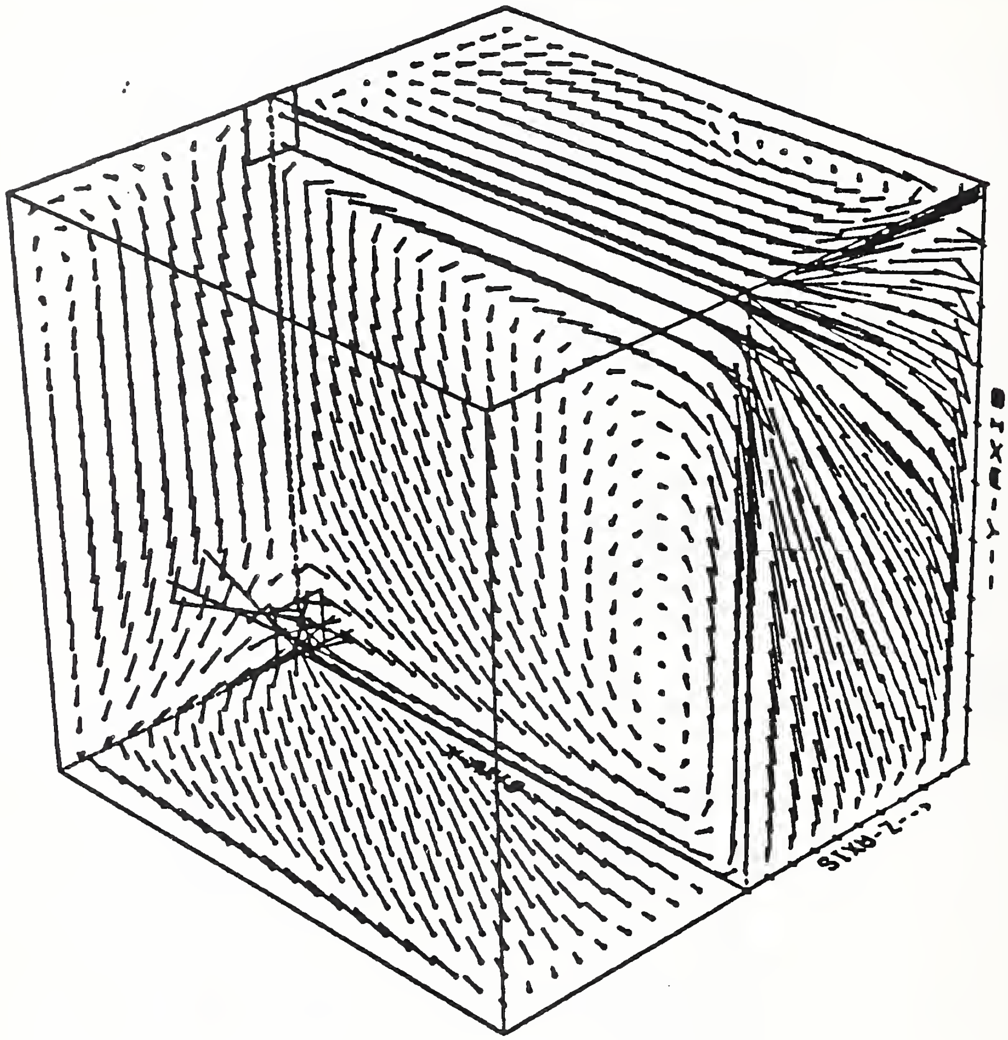


Figure 4.10 Prespective View of Predicted Mean Flow Structure



4.3 Baron's geometry

The next geometry shown in figure 4-11 was employed by Baron et al[38] for both experimental and numerical studies. Since they used the Large Eddy Simulation (LES) method as the turbulence model, comparison between different numerical approaches is also possible in this case. Their experimental apparatus consists of a 300 x 300 x 300 mm cubic test cell with water recirculating system, and velocity profiles in representative horizontal and vertical planes were measured using a Laser doppler velometer. Water inlet and outlet are relatively larger in this case than the previous geometries, and the Reynolds number based on the same definition is approximately 36,000. They used a 20x20x40 uniform mesh layout (dense in vertical direction) and a primitive version of the LES method was employed for numerical calculation. Since their geometry contained no plane of symmetry, present numerical study by the "EXACT" code was carried out for the entire flow domain with a 20x20x20 uniform grid layout (6x6 meshes for inlet/outlet). which is coarser than theirs.

Detailed numerical data are given below.

time step	$0.1\ L_0/U_0$
inflow streamwise velocity	U_0 (uniform)
inflow lateral velocity	0 at upwind cells
outflow streamwise velocity	not specified
outflow pressure	0 at downflow cells
outflow lateral velocity	$\partial u_t/\partial n = 0$(free slip)
inflow turbulence energy	$k = 0.005\ U_0^2$ $\varepsilon = 0.00016\ U_0^2/L_0$
inflow length scale	$0.1\ L_0$
wall boundary condition	1/7 th power profile
switching parameter for convection term	2 for whole domain
convergent criteria after 45 min CPU time	$r_k, r_\varepsilon, r_c, r_m < 10^{-5}$

the pressure outflow boundary condition is employed to allow outflow velocity variation.

"EXACT" predictions for the time-averaged velocity vectors on representative vertical and horizontal planes are given in figures 4-12, 4-13, respectively. Inflow deflects slightly upward toward the ceiling and a small recirculating zone is observed near the top corner of the opposite wall, while the flow below the jet region is generally upward and a reversed flow appears near the floor as shown in figure 4-12. Meanwhile, a horizontal recirculation is observed on both sides of the jet region in figure 4-13.

Figures 4-14 and 4-15 show comparison among "EXACT" prediction, prediction with the LES method and measurement concerning a representative vertical plane. These figures show that both "EXACT" and the LES results seem to coincide fairly well with the experiment results as far as the gross features of mean flow are concerned. A marked difference is observed, however, if more detailed comparison is made among them. The LES calculation significantly underpredicts the initial vertical diffusion of the inflow jet; the "EXACT" result shows clearly better agreement with the experimental result in this region. Moreover, the reversed flow appearing in the experiment near the floor region is not predicted in the LES method. Although this reversed flow near the floor is observed in the "EXACT" numerical result, the calculated velocity gradient is less steep than the experiment and the 0 velocity point appears at a much higher position in the first two measurement loctions. On the other hand, agreement of the "EXACT" result with the experiment in the initial horizontal diffusion of inflow jet is not so good as that of the LES method, although the former is better than the latter near the side wall reversed flow region as shown in figures 4-16 and 17. The general flow pattern calculated from the "EXACT" method is given is figure 4-18.

It may be concluded that the present k-ε model and the LES method have a similar degree of accuracy at least for this specific problem, although the source and tendency of discrepancy is different between them, and the use of the LES method does not necessarily guarantee a better result than the application of the k-ε model. The choice of the k-ε model, therefore, seems to be preferable due to its relatively small demand of computational effort for more practical and complicated problems.

Figure 4.11 Baron's Geometry

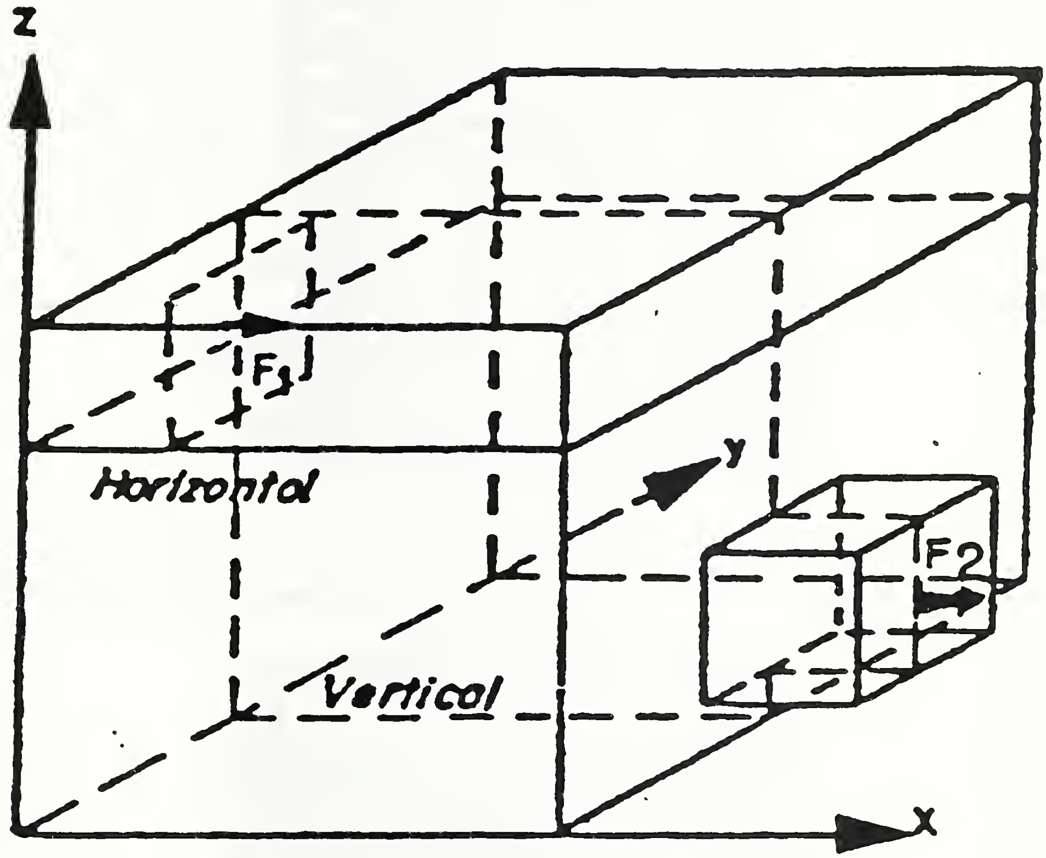


Figure 4.12 Predicted Average Velocity Vectors on a Representative Vertical Plane

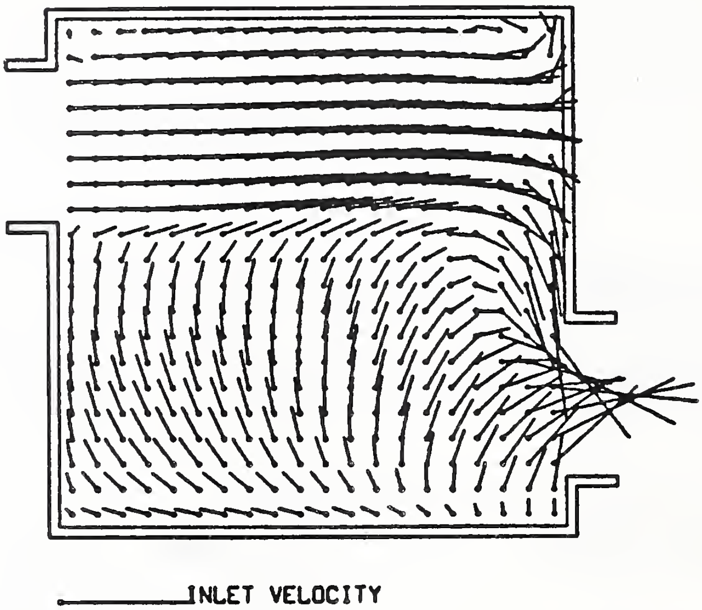


Figure 4.13 Predicted Average Velocity Vectors on Representative Horizontal Plane

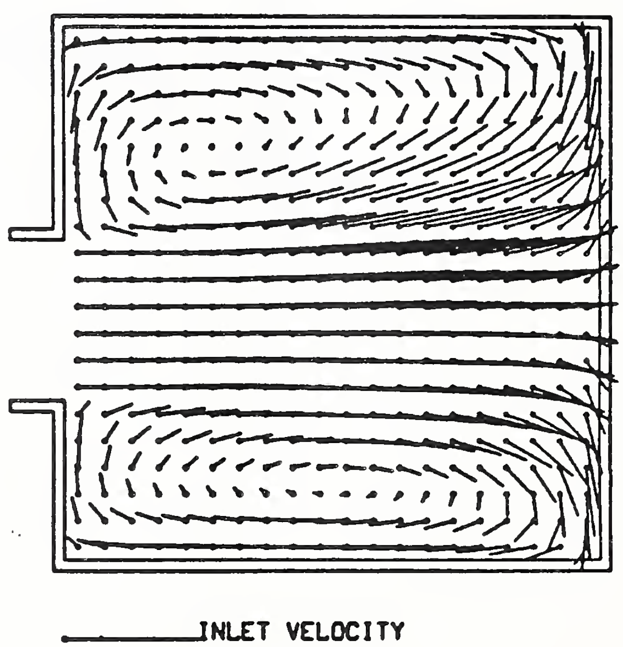


Figure 4.14 Predicted Average Velocity Vectors on Representaitve Vertical Plane

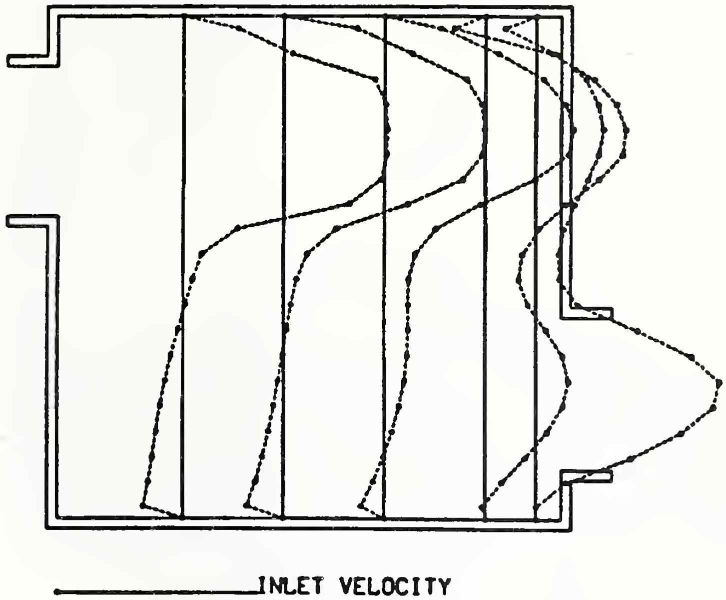


Figure 4.15 Predicted and Measured Average Velocity Profile on Representative Vertical Plane
Solid line : measured
Dashed line :predicted with LES

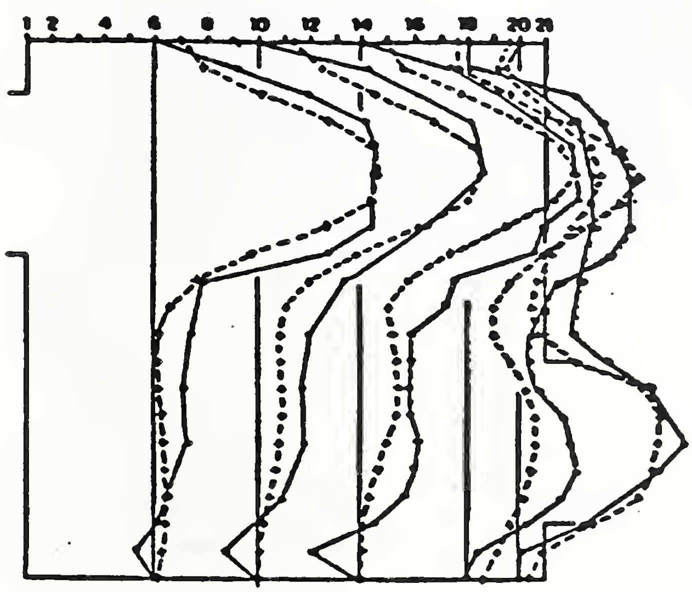


Figure 4.16 Predicted Average Velocity Vectors on Representative Horizontal Plane

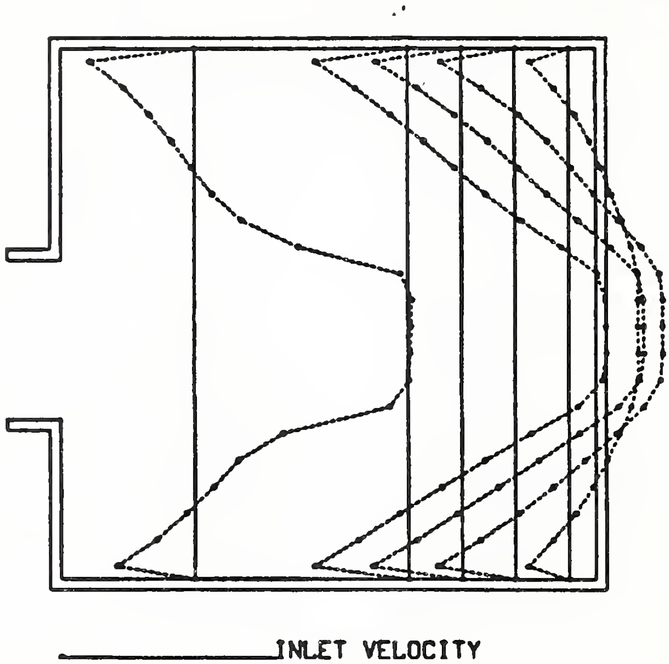


Figure 4.17 Predicted and Measured Average VelocityProfile on Representative Horizontal Plane
Solid line : measured
Dashed line :predicted with LES

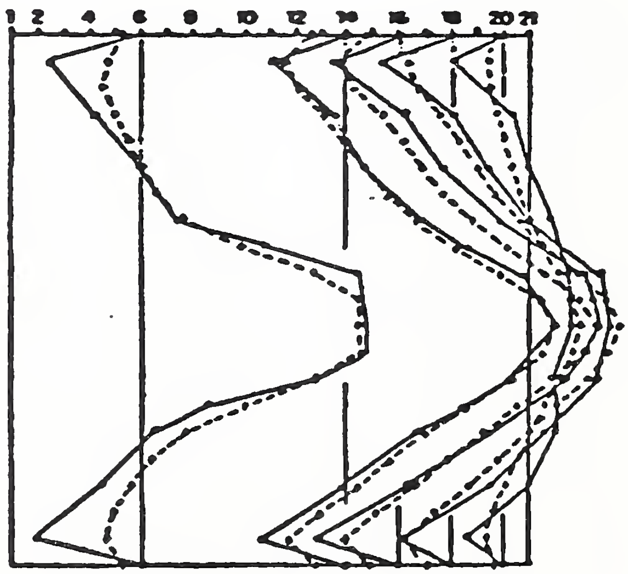
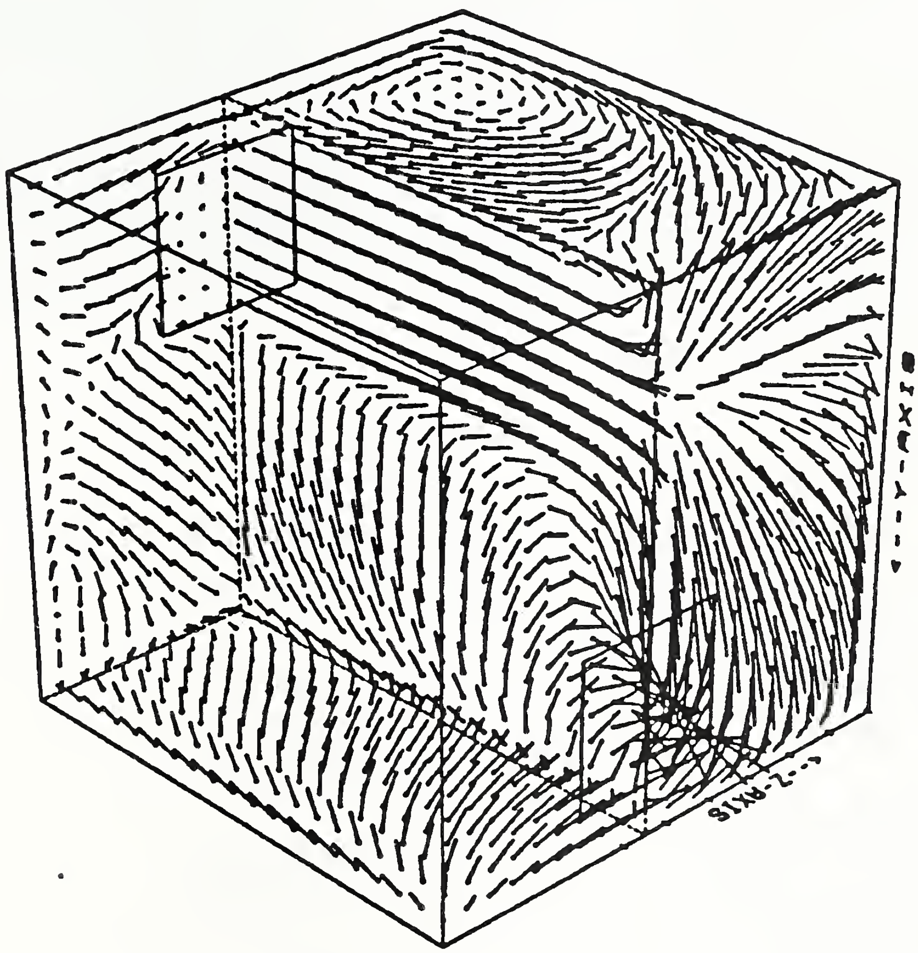


Figure 4.18 Perspective View of Predicted Mean Flow Structure



4.4 Application to A Practical Problem

Through the previous studies on elementary cubic geometry, it has been made clear that the "EXACT" numerical method is capable of predicting the three-dimensional mean flow structure as well as some statistical quantities with fair accuracy. This section gives the results of its application to more complicated and realistic geometry. The geometry to be considered is a typical conventional clean room with two air inlets in the ceiling and four exhaust outlets near the bottom corner of vertical walls. A box shaped work bench is located in the middle of the room as shown in figure 4-19. This geometry was employed by Murakami et al.[39] and they made both numerical and experimental studies. The mean velocity distribution as well as contaminant concentration distributions were measured on representative planes of a full scale clean room laboratory. Flow velocity and direction were measured separately with a thermister type non-dimensional anemometer and by a flow visualization technique, which was found to be less than 50 cm/s in most parts of the flow domain. Their results on velocity distribution may, therefore, be less accurate than those in previous sections. Concentration distribution was also measured using a nearly passive contaminant source (0.31 μm polystyrene standard particles) located on the workbench and a particle counter as a detector. The Reynolds number of the inflow condition was approximately 38,000 and most of the flow domain is expected to be turbulent. Numerical prediction corresponding to the test condition was conducted with a 21x23x16 nonuniform mesh layout including one inlet and outlet, where fine grid spacing is employed in the vicinity of wall, inlet, outlet and workbench.

The following input data were used for "EXACT" calculation.

time step	$0.0175\ L_0/U_0$
inflow streamwise velocity	U_0 (uniform)
inflow lateral velocity	$\partial u_t/\partial n = 0$(free slip)
outflow streamwise velocity	not specified
outflow lateral velocity	
outflow pressure	0 at downflow cells
inflow turbulence energy	$k = 0.005\ U_0^2$ $\varepsilon = 0.00032\ U_0^2/L_0$
inflow length scale	$0.1\ L_0$
wall boundary condition	1/7 th power profile
switching parameter for convection term	2 for whole domain
convergent criteria after 90 min CPU time	$r_k, r_\varepsilon, r_c < 10^{-5}$ $r_m \sim 3 \cdot 10^{-5}$

The pressure outflow boundary condition is also employed to allow outflow velocity variation.

The "EXACT" prediction for the time-averaged velocity vectors and corresponding experimental result are compared in figure 4-20 on the plane of symmetry. The numerical result shows that the inflow consists of a downward flow, diverging horizontally (radially) after impinging the floor, and creating small secondary recirculations in the vicinity of vertical side wall and vertical wall of the work bench. Although the downward flow from the air inlet and recirculation near the side wall are also observed in the experiment, coarse velocity measurement locations does not permit detailed comparison. While the flow direction is generally downward in the prediction, the main discrepancy between prediction and experiment occurs in the region just over the workbench, where the flow direction is rather unstable and its velocity is quite small in the experiment. Numerical and experimental concentration distributions are also compared in figure 4-21 on the same representative plane. As can be seen from this figure, the peak concentration zone occurs over the workbench in both prediction and experiment. The calculated high concentration zone is small, suggesting an overestimation of flow velocity over the work bench. Aside from this discrepancy, the calculated concentration distribution pattern, as well as its magnitude, seems to correspond well with the experiment within an acceptable range of accuracy. General features of the mean flow also seems to be predicted with fair accuracy. Finally the three dimentional mean flow structure in given in figure 4-22 for the symmetric half flow domain.

Figure 4.19 Hypothetical Clean Room (Dimensions in mm)

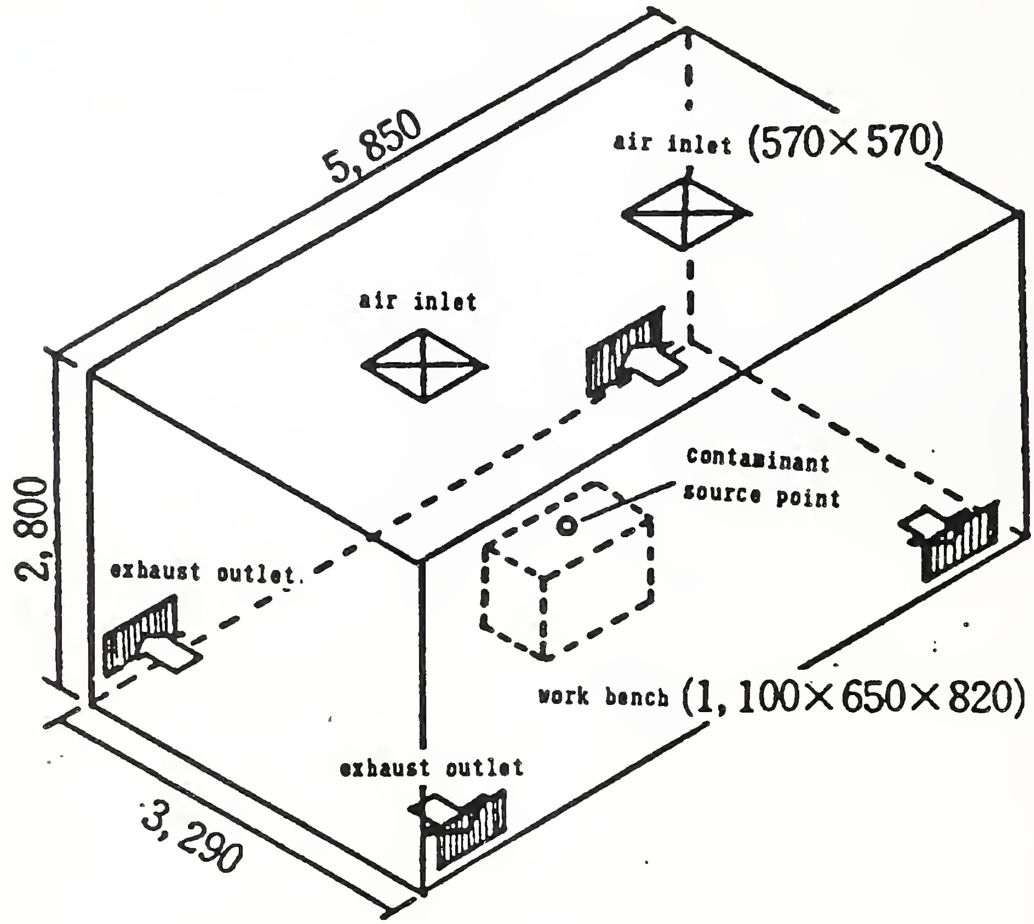


Figure 4.20 Predicted and Measured Velocity Vectors on Plane of Symmetry
left view: predicted
right view: measured.

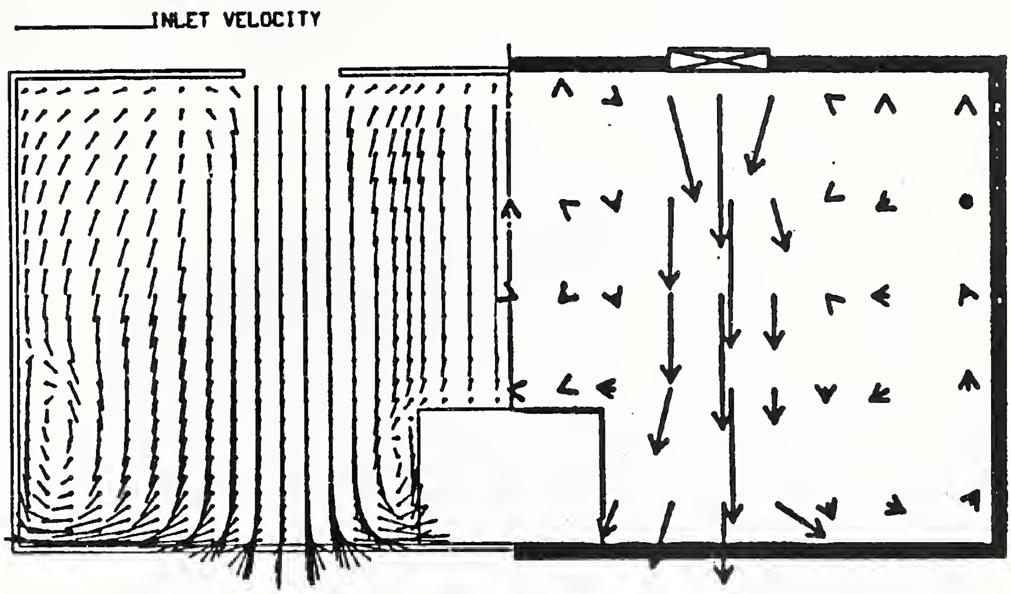


Figure 4.21 Predicted and Measured Concentration Contour Lines on Plane of Symmetry
left view: predicted
right view: measured

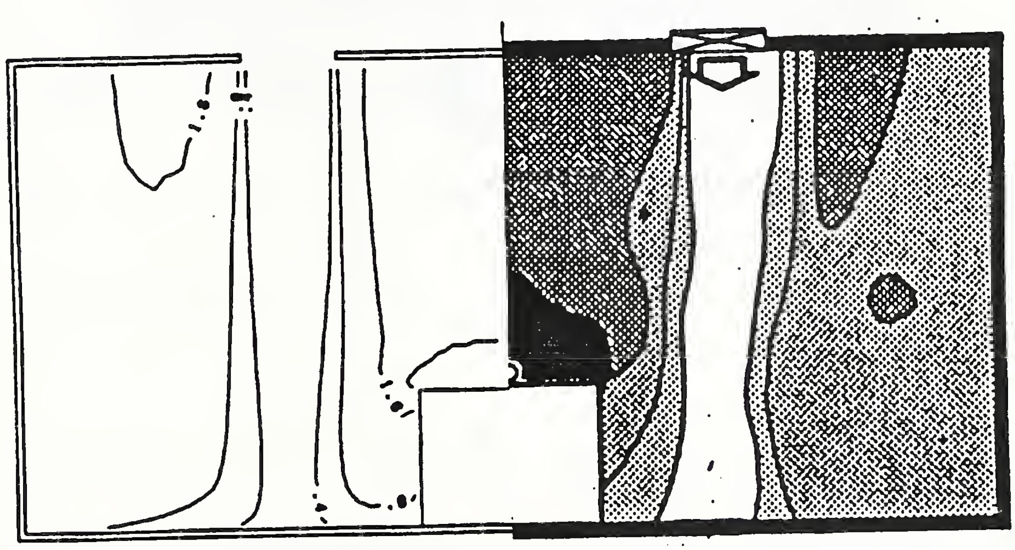
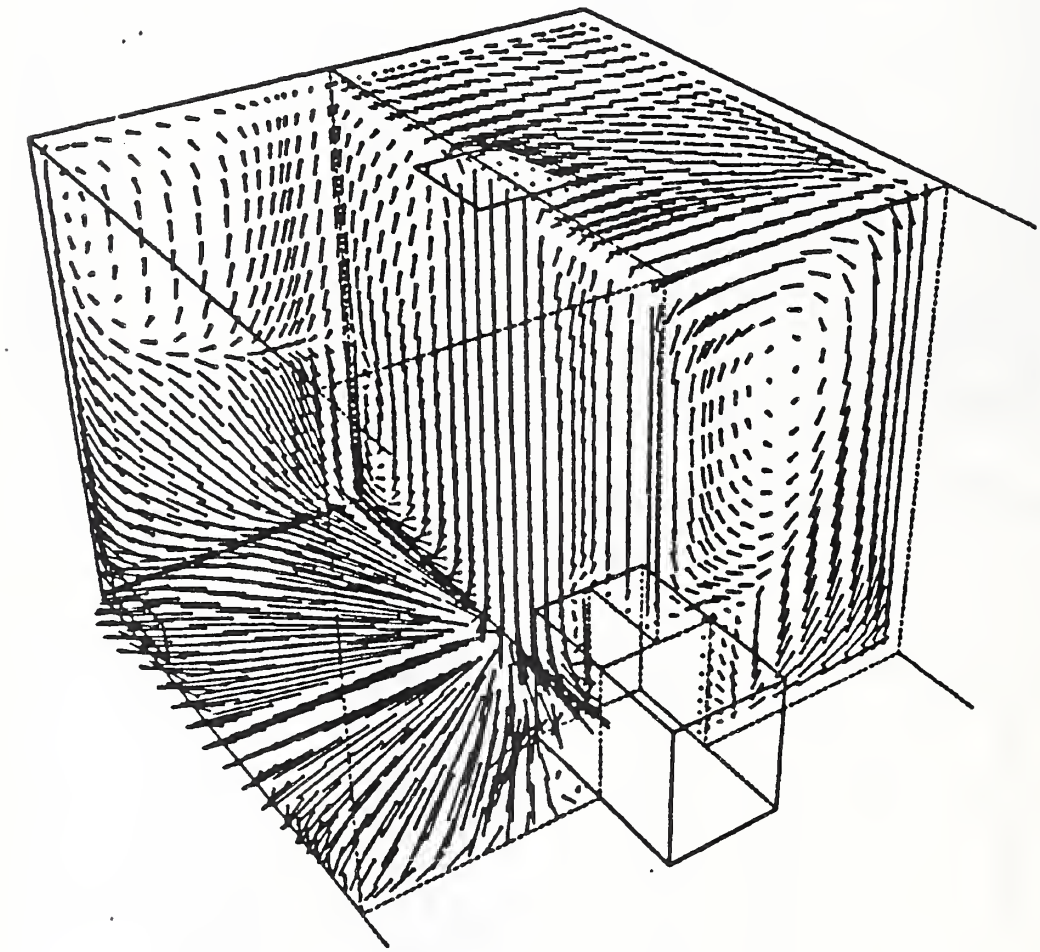


Figure 4.22 Perspective View of Predicted Flow Structure.



5. Buoyancy Affected Flows

In the preceeding chapter, some typical examples of the use of the developed computer code for predicting indoor air movement under isothermal flow conditions were presented. For most room air conditioning and ventilation problems of importance, the airflow is generally non-isothermal and under a buoyancy influenced situation. In this section, the finite difference formulation developed previously will be applied to the problem of the flow of cold air entering a ventilated room subject to known heat input rates and diverse supply air velocities.

It is convenient to solve a system of governing equations in non-dimensional form since the flow variables will then fall between certain prescribed limits such as 0 and 1 and the characteristic parameters such as Reynolds number and Archimedes number can be varied independently.

Introducing the follwing dimensionless variables:

$$\hat{x}_j = x_j/L_0 \quad , \quad \hat{t} = tU_0/L_0 \quad , \quad \hat{u}_j = u_j/U_0 \quad , \hat{\rho} = \rho/\rho_0$$

$$\hat{p} = p/\rho_0 U_o^2 \quad , \quad \hat{\theta} = \theta/\theta_0 \quad , \quad \hat{k} = k/U_0^2 \quad , \quad \hat{\varepsilon} = \varepsilon L_0/U_0^3$$

where L_0 is the width of the inlet opening, U_0 is the inlet air velocity, ρ_0 is the average density of air, θ_0 is the temperature difference between the supply air and the average room air, into the basic equations describing conservation of mass (2-6) and momentum (2-42), the energy equation (2-43) and the two transport equations for turbulence kinetic energy (2-46) and its dissipation rate (2-47) results in the following non-dimensional form of these equations:

$$\frac{\partial \hat{u}_j}{\partial \hat{x}_j} = 0 \qquad (5-1)$$

$$\frac{\partial \hat{u}_i}{\partial \hat{t}} + \frac{\partial \hat{u}_i \hat{u}_j}{\partial \hat{x}_j} = -\frac{1}{\rho}\frac{\partial P}{\partial \hat{x}_i} + \frac{\partial}{\partial \hat{x}_j}\left\{\hat{\nu}_{eff}\left(\frac{\partial \hat{u}_i}{\partial \hat{x}_j} + \frac{\partial \hat{u}_j}{\partial \hat{x}_i}\right)\right\} - Ar_i\hat{\theta} \qquad (5-2)$$

$$\frac{\partial \hat{\theta}}{\partial \hat{t}} + \frac{\partial \hat{\theta}\hat{u}_j}{\partial \hat{x}_j} = \frac{\partial}{\partial \hat{x}_j}\left(\alpha_{eff}\frac{\partial \hat{\theta}}{\partial \hat{x}_j}\right) + S_\theta \qquad (5-3)$$

$$\frac{\partial \hat{\kappa}}{\partial \hat{t}} + \frac{\partial \kappa \hat{u}_j}{\partial \hat{x}_j} = \frac{\partial}{\partial \hat{x}_j}\left(\Gamma_\kappa \frac{\partial \hat{\kappa}}{\partial \hat{x}_j}\right) + \hat{\nu}_t S + G - \hat{\varepsilon} \qquad (5-4)$$

$$\frac{\partial \hat{\varepsilon}}{\partial \hat{t}} + \frac{\partial \hat{\varepsilon}\hat{u}_j}{\partial \hat{x}_j} = \frac{\partial}{\partial \hat{x}_j}\left(\Gamma_\varepsilon \frac{\partial \hat{\varepsilon}}{\partial \hat{x}_j}\right) + \frac{\hat{\varepsilon}}{\hat{\kappa}}(c_1\hat{\nu}_t S - c_2\hat{\varepsilon} + c_3 G) \qquad (5-5)$$

where

$\nu_t = C_D \frac{\kappa^2}{\varepsilon}$ eddy viscosity

$\hat{\nu}_t = \frac{\nu_t}{U_0 L_0}$ the dimensionless eddy viscosity

$$\nu_{eff} = \frac{1}{Re} + \hat{\nu}_t$$ effective eddy viscosity

$$\alpha_{eff} = \frac{1}{RePr} + \frac{\hat{\nu}_t}{\sigma_\theta}$$ effective thermal diffusivity

$$S = \left(\frac{\partial \hat{u}_i}{\partial \hat{x}_j} + \frac{\partial \hat{u}_j}{\partial \hat{x}_i}\right)\frac{\partial \hat{u}_i}{\partial \hat{x}_j}$$

$$G = \beta g_j \frac{\hat{\nu}_t}{\sigma_\theta}\frac{\partial \hat{\theta}}{\partial \hat{x}_j}$$

$$\Gamma_\kappa = \frac{1}{Re} + \frac{\hat{\nu}_t}{\sigma_\kappa}$$

$$\Gamma_\varepsilon = \frac{1}{Re} + \frac{\hat{\nu}_t}{\sigma_\varepsilon}$$

$Re = L_0 U_0 / \nu$ the Reynolds number

$Pr = \nu/\alpha$ the Prandtl number

$Ar_i = \beta g_i L_0 \theta_0 / U_0^2$

$S_\theta - = hL_0/\rho C_p U_0 \theta_0$ the dimensionless heat generation rate

where h is the volumetric rate of heat generation and α is the thermal diffusivity of air.

Using the following values for the empirical constants appearing in equations (5-1) to (5-5)

$c_D = 0.09,\quad c_1 = 1.44,\quad c_2 = 1.92,\quad c_3 = 1.0$

$\sigma_\kappa = 1.0,\quad \sigma_\varepsilon = 1.3,\quad \sigma_\theta = 0.9$

and following the MAC method, iterative solving and numerical time integration of these equations were made to obtain the converged solution.

The test room modeled numerically was studied extensively in references 44 through 47 and had overall dimensions of 6.10 m wide x 3.66 m long x 2.74 m high, simulating an interior room of a multistory office building. The energy input to the room was composed of electrical and lighting loads. The electrical heat loads were generated by finstrip heaters located around the center of the room floor and having a total heat output of 22.07 W/m^2 over a 3.66 x 1.52 m actual floor area, and a concentrated load composed of a 0.97 x 0.91 x 0.31 m angle iron framework installed 0.20 m from the south wall, and having a heat output rate of 40.98 W/m^2 of floor area. The ceiling lighting consisted of eight flourescent recessed double-tubed fixtures evenly distributed 0.31 m from the east and west walls and 0.61 m from the end walls of the short dimension and having a total power input of approximately 760 W. A sidewall grille measuring 0.61 m wide by 0.15 m high was located in the center of the 3.66 m long north wall, with its horizontal center line 0.15 m below the ceiling. A 0.76 m x 0.42 m high return air grill was situated directly beneath the supply air grille, 0.71 m above the floor. Local air temperatures and velocities within the occupied zone of the test room were measured at 216 locations using anemometers and thermocouples. The air supply grilles had two rows of 19 mm wide adjustable vanes and both sets of vanes were straight for all tests performed. For all tests, the room was maintained at an average temperature of 23.33 ± 0.39 C. Under a total heating load of 63.05 W/m^2 of floor area plus a lighting load of 34.09 W/m^2 of floor area, the flow rates of air supply to the room varied from 10.97 to 91.44 $m^3/h - m^2$ of floor area.

Mathematical modeling was performed to simulate six air distribution tests on the sidewall grille installed in a ventilated room. A numerical grid with the symmetrical half-portion of the room being subdivided nonuniformly into 29 x 27 x 20 rectangular parallelepiped cells was used with six different airflow rates including 10.97, 18.29, 36.58, 54.86, 73.15 and 91.44 $m^3/h\text{-}m^2$ of floor area. In order to deal with boundary conditions, one or two dummy cells with the same cell intervals as that of the terminal real cell were added outside of the boundary. The temperature difference between the supply air and the average room air for each flow rate was calculated based on an overall energy balance for the whole room. The calculated values of temperature difference between the room air and the supply air, for different rates of air inflow are tabulated in Table 5-1 along with the Reynolds number and the Archimedes number.

The incoming air was colder than the bulk air in the room to compensate for the input of the room heating loads, simulating a cooling situation. Numerical modeling corresponding to the test conditions was made with a nonuniform mesh layout for a symmetric half portion of the flow domain including the concentrated and uniform heating loads and lighting loads, and an inlet and an outlet opening. Fine grid spacing was used in the vicinity of the walls, the heating loads, and the inlet and outlet openings.

The sidewall grille used for air distribution was assumed to be a 75% free area air diffuser and 83% of the lighting energy was assumed to transmit downward into the room by convection and radiation. The boundary conditions for the velocity and turbulence properties included zero gradients in the exit plane and logarithmic wall functions to describe the near-wall or solid surface regions. All surfaces were assumed to be adiabatic except a portion of the floor and the ceiling around the room center and the top face of the concentrated load, where constant heat input rates were prescribed. Heat was supplied at a rate of 88.3 W/m^2 from the uniform load at the center of the room floor, and at 3279 W/m^2 from the concentrated load in the vicinity of wall opposite to the inlet and at 26.7 W/m^2 from the fluorescent lights at the center of the ceiling. The heat inputs into the flow domain were assumed to be transmitted to the surrounding fluid cells immediately adjacent to the heated surfaces by adding corresponding heat source terms in the energy equation. As illustrated in table 5-1, the inlet temperatures for the six air inflow rates varying from 10.97 to 91.44 $m^3/h\text{-}m^2$ were respectively 17.3, 10.4, 5-2, 3.5, 2.6 and 2.1 °C lower than the average temperature of the room of 23.3 °C.

Figures 5-1 to 5-6 show the calculated velocity distributions in the vertical center plane of the test room for the six different air inflow rates. As shown in figure 5-1, the cold air coming from the inlet travelled down towards the floor due to the downward directed buoyancy force. A portion of flow circulated around the lower left corner of the room and exited through the return air grille. The remaining portion proceeded along the floor surface, turned upward after impinging onto the concentrated load, and after being accelerated by merging with the hot gas stream rising from the concentrated load, spread radially along the ceiling and entrained into the main flows from the inlet. Two recirculating zones were observed, one situated in the top right corner of the room and the other one near the floor in the vicinity of the concentrated load. The experimental air distribution patterns on the vertical center plane of the room obtained with smoke filaments for different flow rates[46,46] are given in figure 5-7 for comparison with the predictions. As shown in figures 5-1 and 5-7.a, the predicted general flow structure agrees quite well with the corresponding experimental observations. In figure 5-2, the inflows following the ceiling spread radially toward the floor and the wall opposite to the entrance because of the increased inlet velocity and decreased downward directed buoyancy effects. The flows are then curved along the wall and floor, and entrained into the jet stream or depart from the test room. It can be seen that a secondary recirculation is created in the vicinity of the ceiling and the concentrated heat source. It is interesting to note that the recirculation in this case has moved toward the ceiling in comparison to figure 5-1 where the recirculation is along the floor. There is good agreement between prediction and experimental observations of flow patterns by comparison of figures 5-2 and 5-7.b.

Figures 5-3 through 5-6 show a turbulent buoyant wall jet issuing from the inlet grille, spreading along the ceiling and turning downward and horizontally toward the outlet after impinging onto the opposite wall. A recirculating flow structure appeared in the whole flow domain with its center located near the concentrated load. The predicted flow patterns shown in figures 5-3 to 5-6 are generally consistent with the corresponding experimental observations illustrated in figure 5-7. Some examples of calculated isotherms in the middle section of the vented room are shown in figures 5-8 through 5-10 where air supply rates per unit floor area are 18.29, 54.86 and 73.15 $m^3/h\text{-}m^2$, respectively. The temperature field is generally dependent upon the inlet air velocity, the temperature difference between the supply air and the room air, and the distribution and output rate of the heat sources. As shown in figure 5-8, the air temperature increases with increasing distance away from the inlet grille due to continuous entrainment of warm air into the cold jet stream discharging from the inlet and flowing toward the floor. The air temperature exhibits a sharp gradient close to the heat source and its distribution generally agrees with the corresponding flow field illustrated in figure 5-1. As shown in figures 5-9 and 5-10, the isothermal lines in the upper portion of the room are nearly horizontal. These diminished temperature gradients and the increased thermal stratification in the upper region are attributed to greater inlet velocities and better mixing of the cold supply air with warm room air.

Prediction of buoyancy-affected airflows emerging from an air diffuser in a ventilated room has been demonstrated over a wide range of air supply rates with constant heating loads using the numerical technique presented in this paper. The three-dimensional distributions of air velocity and temperature in an air conditioned room are calculated, and the calculated velocity distributions are generally in reasonably good agreement with experimental observations obtained with smoke filaments.

Table 5-1. The Numerical Values of Temperature Difference between the Room Air and the Supply Air and Nondimensional Parameters Used in Numerical Calculations

Parameters Used in Numerical Simulations				
Inlet Flow Rate (m^3/h-m^2)	Inlet Velocity (m/s)	Temperature Difference (°C)	Reynolds Number	Archimedes Number
10.97	0.975	17.26	38860	0.365920
18.29	1.626	10.36	64430	0.079040
36.58	3.251	5-180	128860	0.009879
54.86	4.877	3.452	193290	0.002927
73.15	6.502	2.589	257710	0.001235
91.44	8.128	2.071	322150	0.000632

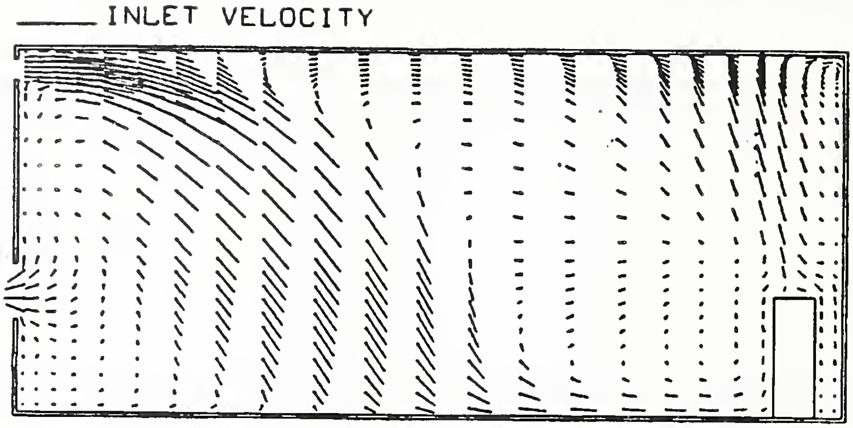


Figure 5-1. Distribution of Calculated Velocity Vectors in the Center Plane for Inflow Rate of 10.97 $m^3/h\text{-}m^2$

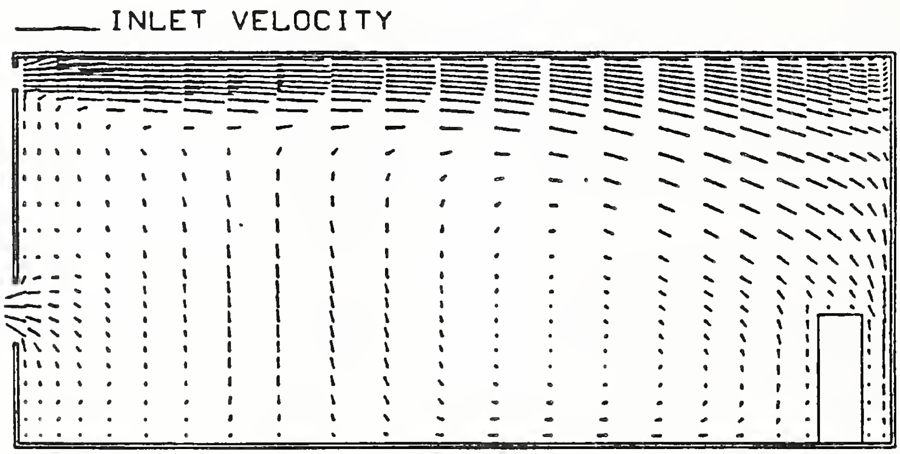


Figure 5-2. Distribution of Calculated Velocity Vectors in the Center Plane for Inflow Rate of 18.29 $m^3/h\text{-}m^2$

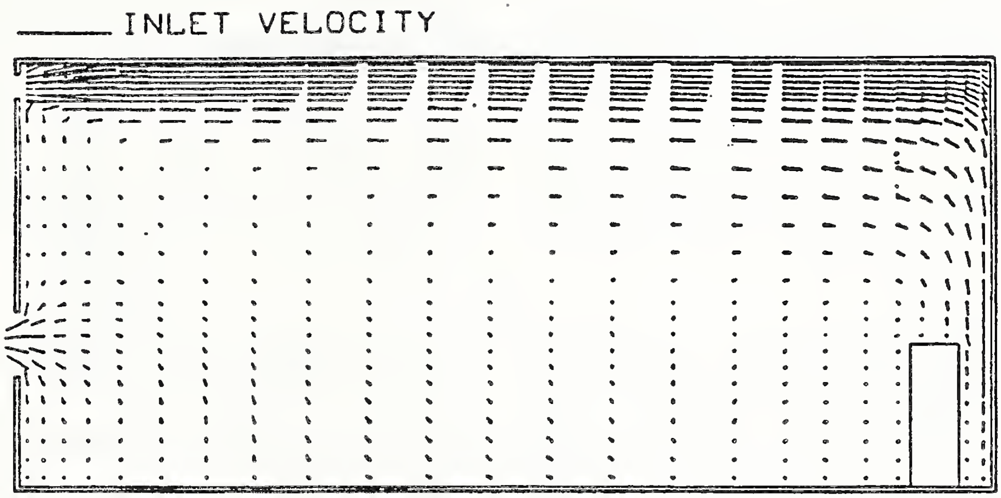


Figure 5-3. Distribution of Calculated Velocity Vectors in the Center Plane for Inflow Rate of 36.58 $m^3/h\text{-}m^2$

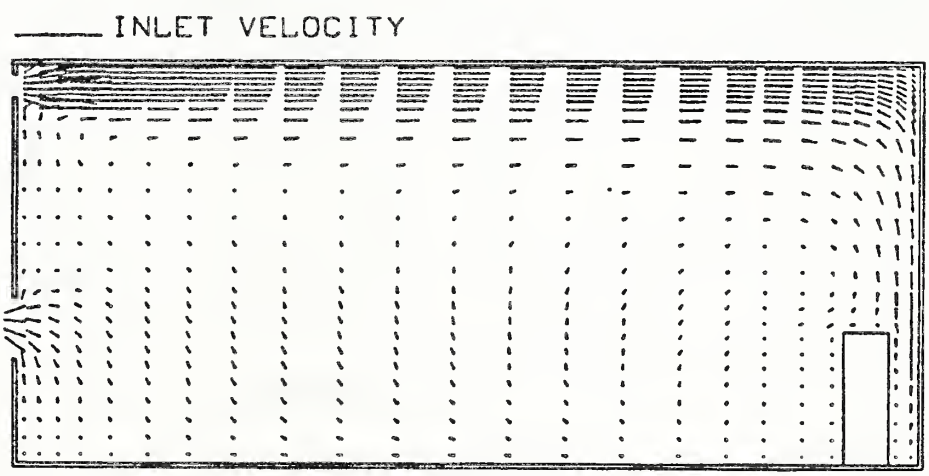


Figure 5-4. Distribution of Calculated Velocity Vectors in the Center Plane for Inflow Rate of 54.86 $m^3/h\text{-}m^2$

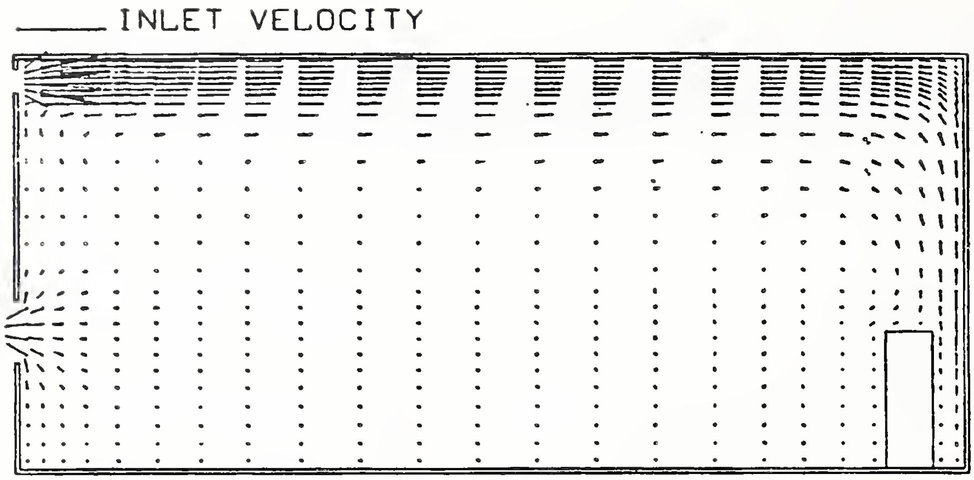


Figure 5-5- Distribution of Calculated Velocity Vectors in the Center Plane for Inflow Rate of 73.15 $m^3/h\text{-}m^2$

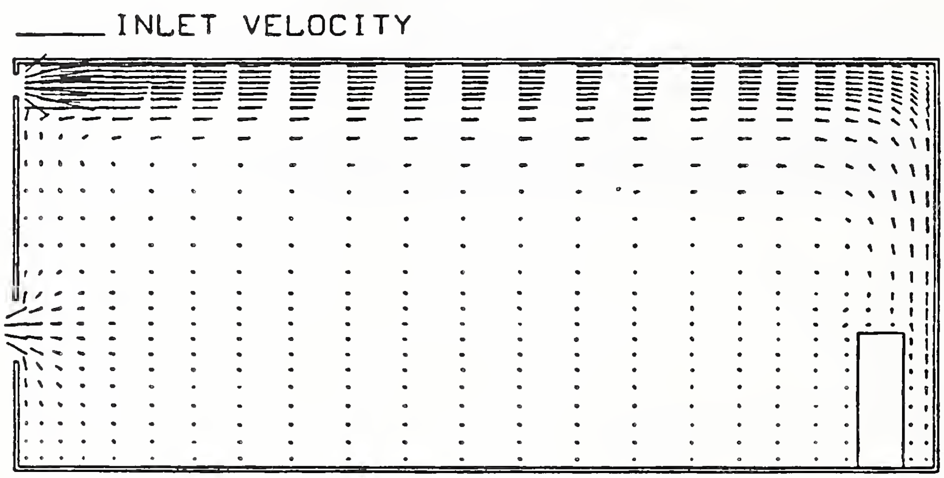
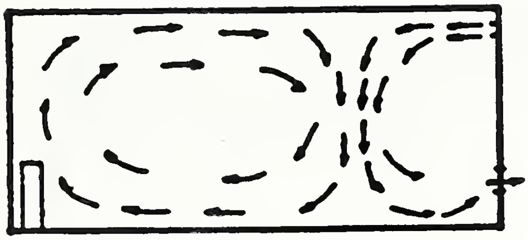
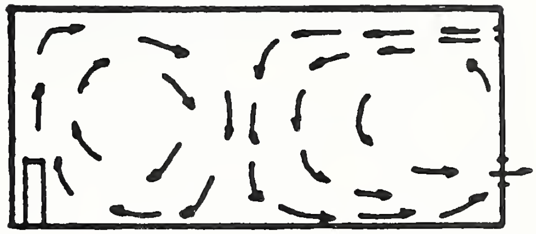


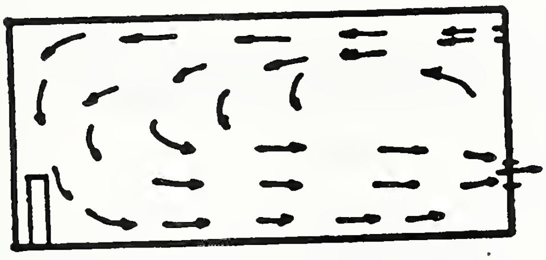
Figure 5-6. Distribution of Calculated Velocity Vectors in the Center Plane for Inflow Rate of 91.44 $m^3/h\text{-}m^2$



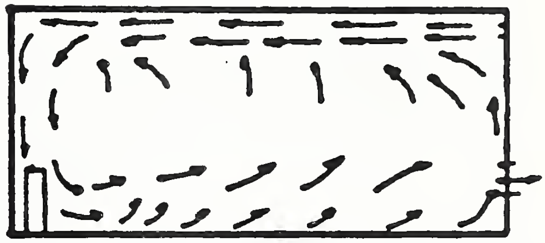
a. Inflow rate = 10.97 $m^3/h\text{-}m^2$



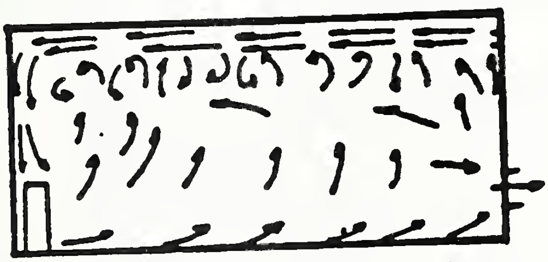
b. Inflow rate = 18.29 $m^3/h\text{-}m^2$



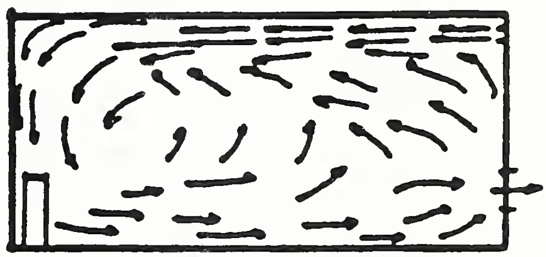
c. Inflow rate = 36.58 $m^3/h\text{-}m^2$



d. Inflow rate = 54.86 $m^3/h\text{-}m^2$



e. Inflow rate = 73.15 $m^3/h\text{-}m^2$



f. Inflow rate = 91.44 $m^3/h\text{-}m^2$

Figure 5-7. Measured Airflow Patterns for Different Airflow Rates

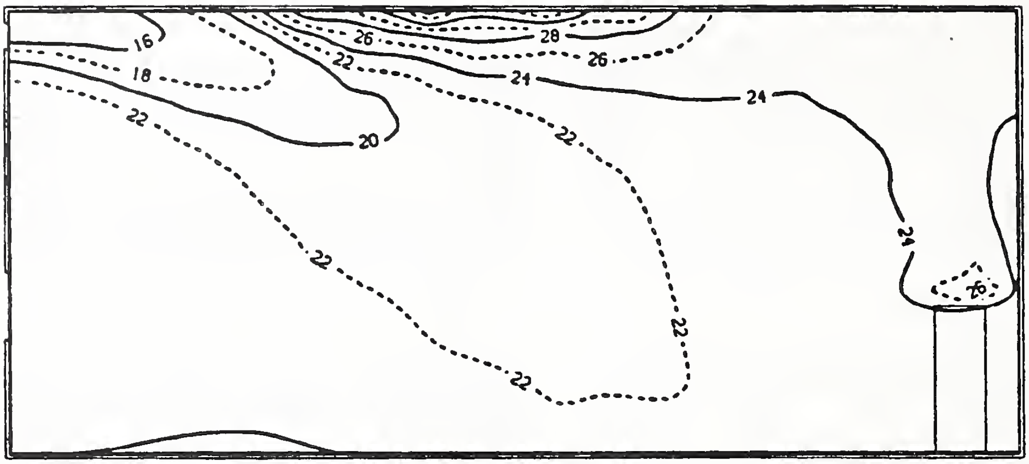


Figure 5-8. Calculated Isotherms in the Center Plane for Inflow Rate of 18.29 m³/h-m²

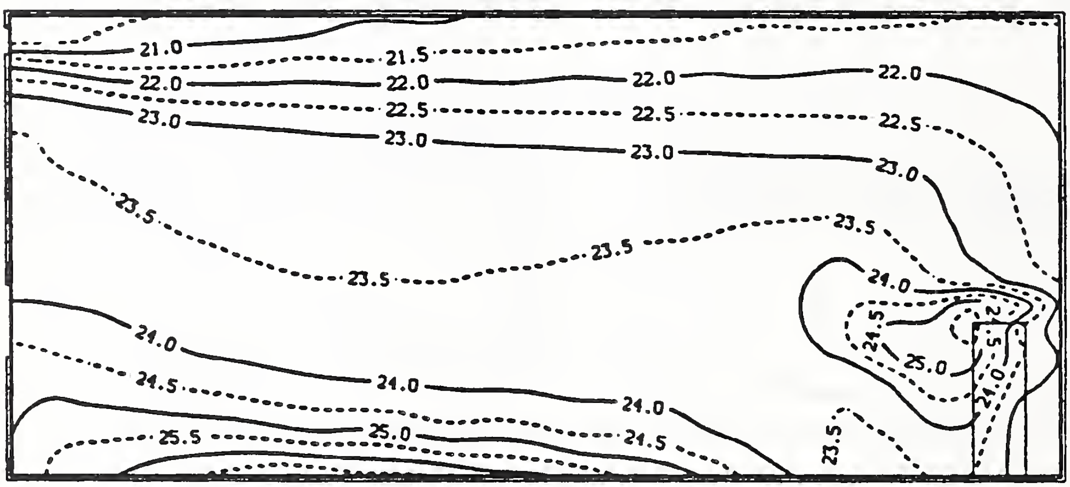


Figure 5-9. Calculated Isotherms in the Center Plane for Inflow Rate of 54.86 m³/h-m²

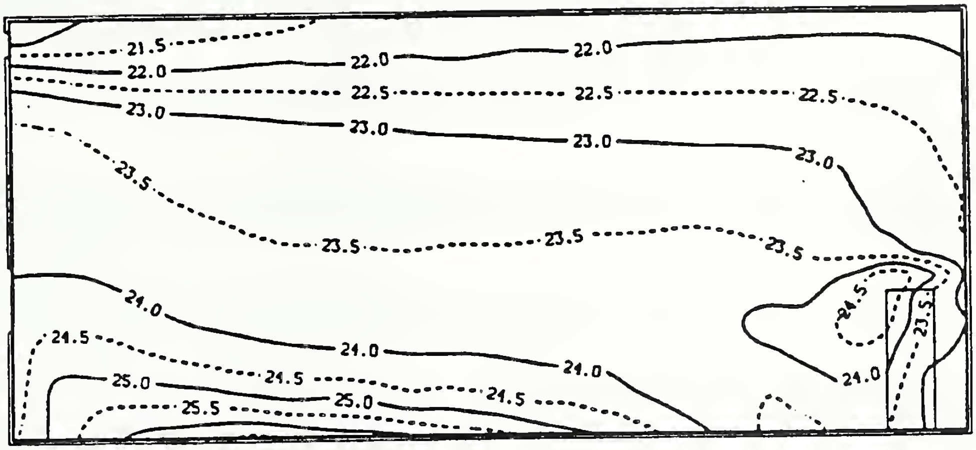


Figure 5-10. Calculated Isotherms in the Center Plane for Inflow Rate of 73.15 $m^3/h-m^2$

6. Conclusions and Future Research Needs

As briefly demonstrated in the above examples, the "EXACT" numerical code is powerful and is able to predict room air motion and contaminant distribution with a satisfactory accuracy and within an acceptable range of computational effort. Nevertheless, it is important to recognize the following limitations inherent in the present calculation method.

- The calculation domain including air inlet and outlet must be a rectangle.
- The reference Reynolds number must be large enough so that the viscous effect can be negligible except near the wall region.
- The temperature gradient in the flow regime must be small enough that buoyancy exerts little influence on the turbulence structure.

The first is caused by the use of standard finite-difference approximations based on the rectangular coordinate system. This may be a serious limitation for the application of the numerical method to spaces enclosed by curved surfaces such as air domes. This difficulty is expected to be removed, however, by using the recently developed approximation techniques such as finite-element method or grid generation method which have been employed extensively in the field of numerical aerodynamics. The other two limitations are much more serious than the first one because they originate from the inadequacies of the basic turbulence model itself.

The k - ε turbulence model consists of a set of equations governing the transport process of turbulent kinetic energy and its dissipation rate. These equations for k and ε together with the eddy viscosity formulation are not the exact forms derived from the Navier Stokes equation but they are model equations derived from several assumptions as discussed in the previous chapters.

Hence, there is a fair possibility that the numerical results become unrealistic when Reynolds number is not large enough so that eddy viscosity is as small as molecular viscosity or when the buoyancy effect is so large that the gravitational force exerts a strong directional influence on turbulence.

So far, several authors have proposed modified versions of the k-ε turbulence model or different approaches, which can account for these additional effects. An interesting attempt to extend the applicability of the k- e turbulence model for high Reynolds number was carried out by Jones and Launder[40]). Their modification was mainly to make the empirical constants appearing in the conventional k-ε turbulence model to be the functions of the local turbulence Reynolds number by a semi-empirical approach so that the molecular viscosity effect in the very near wall region can be accounted for.

Although their modified equations have been applied to the two-dimensional low Reynolds number wall flows with relative success, they are not likely to be applicable to the general recirculating low Reynolds number flows.

For strongly buoyant flows, several researchers proposed different schemes called the second-order closure model [41, 42] which is much more complicated than the k-ε turbulence model. Recently, Rodi[43] and associates attempted to simplify the model equations proposed by Launder and associates and clarified the applicability of their simplified version to various buoyancy influenced flows. At the present, the study for the buoyant flow prediction has not been conducted as extensively as for isothermal flow in the field of building physics. This is partially due to the difficulty of experimental verification but mainly due to the uncertainty of the basic model equations. In order to develop effective modifications, it is an urgent task to determine the applicable range of the present method by providing careful comparative studies with experimental observations for fundamental buoyant flows.

References

1. Andrade, E.N., "The velocity distribution in a liquid-into-liquid jet. The plane jet", Proc. Phys. Soc. vol. 51, 1939.
2. Sato, M., Sakao, F., "An experimental investigation of the stability of a two-dimensional jet at low Reynolds number", J. Fluid Mech., vol. 20, 1964.
3. Reynolds, A.J., "Observations of a liquid-into-liquid jet", J. Fluid Mech., vol. 14, 1962.
4. Wille, R., "Beitrage zur Phanomenologie der Freistrablen", Z. Fluswissenschafter, 6, 1963 (in German).
5. Cebeci, T., Smith, A.M.O.., "Analysis of turbulent boundary layers", Academic Press, 1974.
6. Terai, T., "Indoor thermal convection", Paper of A.I.J. (Architectural Institution of Japan), vol. 63, 1959 (in Japanese).
7. Tsuchiya, T., "Application of meteorological numerical calculation method to indoor air-flows", Proc. annual meeting of A.I.J., 1973 (in Japanese).
8. Nomura, T., Kaizuka, M. "Numerical calculation method of air distribution, part 1 4", Proc. annual meeting of A.I.J., 1972 1973 (in Japanese).
9. Yamazaki, H., Urano, Y., Nashida, M. Watanabe, T., "Comparisons of numerical calculation oand visualization experiment of two-dimensional flows", Paper of A.I.J., vol. 240, 1976 (in Japanese).
10. Nomura, T., Matsuo, Y., Kaizuka, M., Sakamoto, Y., Endo, K., "Numerical study of room air distribution, part 1, 2", Paper of A.I.J., vol. 231, 232, 1975 (in Japanese).

11. Yoshikawa, A., Yamaguchi, K., "Numerical analysis of indoor airflows", Japanese society of heating-cooling air conditioning and sanitary engineering (J.S.H.A.S.E.) vol. 48, 1974 (in Japanese).

12. Nomura, T., Matsuo, Y., Kaizuka, M., Sakamoto, Y., Endo, K., "Numerical study of room air distribution, part 3", Paper of A.I.J., vol. 238, 1975 (in Japanese).

13. Sakamoto, Y., Matsuo, Y., Nomura, T., Kamata, M., "Numerical prediction of three dimensional thermal convection by means of two equation turbulence model", Proc annual meeting of A.I.J., 1978 (in Japanese).

14. Nomura, T., Murakami, S., Kato, S., Sato, M., "Correspondence of the three-dimensional numerical analysis of turbulence flow to model experiment", Paper of A.I.J., vol. 298, 1980 (in Japanese).

15. Kamata, M., Kunihira, H., "Numerical simulation of buoyancy influenced indoor airflows", Proc. annual meeting of A.I.J., 1981 (in Japanese).

16. Murakami, S., Kato, S., Mochida, A., et al., "Numerical simulation of turbulent flows", Mon. J. Institute of Industrial Sci., Univ. Tokyo, vol. 38 No. 1 (in Japanese).

17. Kurabuchi, T., Kato, S., Kamata, M., "Numerical study on three- dimensional flow with inflow and outflow boundary condition given in static pressure", J. Fac. Eng., Univ. Tokyo A., No. 20, 1982 (in Japanese).

18. Kurabuchi, T., Kamata, M., "Study of finite-difference approximation of convection and diffusion terms", Proc. annual meeting of A.I.J., 1983 (in Japanese).

19. Matsuo, Y., "A new method for solving steady state Navier-Stokes equations", Proc. annual meeting of A.I.J., 1986 (in Japanese).

20. Smagorinsky, J., Manabe, S., Holloway, J.L. Jr. "Numerical results from nine-level general circulation model of the atmosphere", Mon. Weath Rev., vol. 93, 1965-

21. Deardorff, J.W., "A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers", J. Fluid Mech., vol. 41, 1970.

22. edited by Tani, I., "History of fluid mechanics-Turbulence", Maruzen, 1980 (in Japanese).

23. Launder, B.E., Spalding, D.B., "Mathematical model of turbulence", Academic press, London, 1972.

24. Tennekes, H., Lumley, J.L., "A first course in turbulence", MIT press, 1972.

25. Markatos, N.C., "The mathematical modelling of turbulent flows", Appl. Math. Modelling, vol. 10, 1986.

26. Rodi, W., Spalding, D.B., "A two-parameter model of turbulence and its application to free jets", Warme u. stoffubertragung 3, 1970 (in German).

27. Spalding, D.B., "The prediction of two-dimensional, steady turbulent flows", Imperial College Heat Transfer sec. rep., EF/TN/A/16, 1969.

28. Donaldson, C. duP., "A computer study of an analytical model of boundary layer transition", A.I.A.A.J. 7, 1969.

29. Launder, B.E., Reece, G.J., Rodi, W., "Progress in the development of a Reynolds-stress turbulence closure", J. Fluid Mech., 1975-

30. Rodi, W., "On the equation governing the rate of turbulent energy dissipation", Imperial college. dep. eng. rept. TM/TN/A/14, 1971.

31. edited by Rodi, W., "Turbulent buoyant jets and plumes", Pergamon press, 1982.

32. Harlow, F.H., Welch, J.E. "Numerical calculation of time dependent viscous incompressible flow of fluid with free surface", Phys. Fluids, vol. 8, 1965

33. Roache, P.J., "Computational fluid dynamics", Hermola publisher, 1978.

34. Patankar, S.V., "Numerical heat transfer and fluid flow", Hamisphere publishing corp., 1980.

35. Nomura, T., Kato, S., Sato, M., "Study on the turbulence model boundary condition for numerical indoor air distribution analysis", Proc. annual meeting of A.I.J., 1978.

36. Sakamoto, Y., Matsuo, Y., "Numerical predictions of three-dimensional flow in a ventilated room using turbulence models", Appl. Math.Modeling, fol. 4, No. 1, 1980.

37. Same as 15.

38. Baron, F. Benque, J.P., Coeffe, Y., "Turbulent flow induced by a jet in a cavity-measurements and 3D numerical simulation, Turbulent shear flow 3, Springer, 1982.

39. Murakami, S., Kato, S., Nagano, S., "Study on diffusion of airborne particles in clean room part 9", Mon. J. Institute of Industrial Sci., Univ. Tokyo, vol. 38 No. 1 (in Japanese).

40. Jones, W.P., Launder, B.E. "The prediction of laminarization with a two-equation model of turbulence", Int. J. Heat Mass Transfer, vol. 15, 1972.

41. Lumley, J.L., "A model for Computation of Stratified Turbulent Flows", Int. symp. on stratified flows, Novosibirsk USSR, 1972.

42. Launder, B.E., "On the effects of a gravitational fields on the turbulent transport of heat and momentum", J. Fluid Mech., vol. 67, 1975-

43. Hossain, M.S., Rodi, W., "Influence of buoyancy on the turbulent intensities in horizontal and vertical jets", in Spalding, D.B., and Afgan, N. (Eds.), Heat transfer and turbulent buoyant convection, vol. A. Hemisphere Publishing Corp., 1977.

44. Miller, P. L. and R. G. Nevins, 'An Analysis of the Performance of Room Air Distribution Systems', ASHRAE Transactions, 78, Part I, 191-198, (1972).

45. Miller, P. L., 'Room Air Distribution Performance of Four Selected Outlets', ASHRAE Transactions, 77, Part II, 194-204, (1971).

46. Miller, P. L. and R. T. Nash, 'A Further Analysis of Room Air Distribution Performance', ASHRAE Transactions, 77, Part II, 205-212, (1971).

47. Nevins, R. G., 'Air Diffusion Dynamics--Theory, Design and Application', Business News Publishing Co., Birmingham, Michigan, 1976.

48. Chen, Qingyang, van der Kooi, J. and Meyers, A., "Measurements and Computations of Ventialtion Efficiency and Temperature Efficiency in a Ventilated Room", Energy and Buildings, 12,85-99,(1988).

49. Davidson, L. and Olsson, E., "Calculation of Age and Local Purging Flow Rate in Rooms", Building and Environment, 22, 11-127,(1987).

50. Awbi, H.B., "Application of Computational Fluid Dynamics in Room Ventilation", Building and Environment, 24, 73-84, (1989).

Appendix A. "EXACT" user's guide

A.1. General Scope of the "EXACT" code

The procedure which includes most of the theoretical and numerical processes described in chapters 2 and 3 is currently available by the use of the "EXACT" code. "EXACT", which stands for Explicit Arogorithm for Continuous Turbulent fluid flow, is a complete program for predicting indoor air distribution, including turbulent/thermal effect, and was developed by the first author based on the previous research conducted mainly at the University of Tokyo and adding his recent experience. Since the original "EXACT" code was intended to be processed on the HITACS 810-20 computer system at the University of Tokyo, a drastic alteration was made in order to achieve the effective use of vector/parallel processors characteristic of Cyber 205-2p system at the National Institute of Standards and Technology. Hence, the user is ready to enjoy peak performance of a supercomputer without special modification or knowledge about program vectorization. The code was further modified by the third author to reduce the data storage requirements by about 60%. A non-vectorized version of the latest code has been successfully compiled and run on a Sun 3/260 workstation, and a Compaq 386/25 microcomputer. Since detailed descriptions about theoretical and numerical stragegy employed in the "EXACT" code has already been given, only a summary of important features is given as follows:

<u>turbulence effect:</u>

buoyancy extended k - ε turbulence model

<u>applicable flows:</u>

three-dimensional continuous non-isothermal/buoyant turbulent flow
(Bonssinesq approximation for buoyant flow)

<u>iterative procedure:</u>

explicit time marching technique to obtain converged solution (periodic or time-dependent problems can be handled by adding user defined subroutines when the boundary condition are time dependent)

<u>convective term approximation:</u>

upwind/central combination scheme with arbitrary specification for switching parameter and location, which includes central and hybrid schemes.

<u>solution procedure for Poisson equation:</u>

pressure relaxation method

<u>boundary condition:</u>

Symmetric, inflow, outflow (velocity or pressure type), wall boundary conditions together with heat flux and temperature/heat transfer coefficient specified boundary condition (user defined subroutine may be necessary when special boundary condition is considered)

The "EXACT" code consists of one main routine (driver program) and 13 subroutines. A listing of the program is given in appendix A.5.

A.2 Determination of Memory Size and Mesh Layout Subdivision of Flow Domain

Flow domain to be considered should be carefully subdivided into cells so that their faces coincide with the physical wall boundary, inflow/outflow boundaries and internal obstacles. Both uniform and non-uniform mesh layout are available in the "EXACT" code; however, it should be noted that an abrupt change of cell intervals sometimes reduces numerical accuracy due to truncation error. Choice of axis and its direction are arbitrary under isothermal conditions, however, positive x- direction must be aligned to the direction of the gravitational force for buoyancy influenced problems. In order to handle boundary conditions, three dummy cells are attached outside the real flow domain in each axis, two dummy cells for negative direction and one dummy cell for positive direction. Therefore, a problem which uses 21X23X16 cells for real flow domain requires 24X26X19 cells including dummy cells. Since cells are numbered from one to the cell division number in each axis, the real flow region is enclosed by cells from (three) to (cell division number - 1).

<u>Source program modification</u>

In order to minimize inactive computer memories and for maximum efficiency of the vector processor, "EXACT" code is designed for a fixed number of unknown variables so that three dimensional arrays are accessed continuously within DO loops. Array size must be adjusted depending on the problem size by replacing numbers in the PARAMETER statement with required cell division numbers. If, for instance, NX x NY x NZ cells including three dummy cells are required, PARAMETER statement of <u>all subroutines</u> should be written as

```
PARAMETER(L=NX,M=NY,N=NZ)
```

where NX, NY and NZ are the number of cells in the x, y and z directions, respectively. The program must be recompiled and assembled for each problem. If the compiler used supports the INCLUDE directive, this statement can be put in an include file and changed only once.

A.3 Input data format

All input information except memory size is stored as a formatted permanent file on Cyber 855 or for a UNIX or DOS system in any permanent file to which input can be redirected (for example EXACT3 < DataFile). An example , which is used for problem presented in Chapter 5, is given as follows. This permanent file is made local on Cyber 205 as soon as the batch job stream is submitted.

Sample Input File for EXACT3

```
1  AIR DISTRIBUTION PERFORMANCE FOR 4.0 CFM WITH 75 % FREE AREA DIFFUSER
2      1999000,              20,           288000,           2
3      257710,               0.71,         -0.001235,        1.0,         0.0010,
4      0.09,                 1.44,         1.92,             1.00,        1.0,
5      0.0625,               0.0625,       0.0625,           0.0625,      0.0625,
       0.0625,               0.0625,       0.1250,           0.1250,      0.29774,
       0.29774,              0.29774,      0.29774,          0.17188,     0.17188,
       0.17188,              0.17188,      0.20833,          0.20833,     0.20833,
       0.18485
       0.16667,              0.16667,      0.16667,          0.16667,     0.250,
       0.41667,              0.500,        0.500,            0.6250,      0.6250,
       0.6250,               0.6250,       0.6250,           0.6250,      0.41667,
       0.250,                0.250,        0.250,            0.16667,     0.16667,
       0.125,                0.125,        0.125,            0.125,       0.125,
       0.125,                0.125,        0.1875,           0.1875,      0.29167,
       0.29167,              0.16667,      0.16667,          0.125,       0.125,
6      6,      6,      6
7      41
8      1,   4,   8,   2,   2,   3,   6,       0.0
       2,   5,   8,   2,   2,   3,   6,       1.0
       3,   5,   8,   2,   2,   2,   6,       0.0
       4,   5,   8,   2,   2,   3,   6,   0.21140
       5,   5,   8,   2,   2,   3,   6,  0.022194
       6,   5,   8,   2,   2,   3,   6,       1.0
       8,  19,  22,   2,   3,   3,   7,       0.0
       1,  18,  22,   2,   2,   3,   7,       0.0
       2,  19,  22,   2,   2,   3,   7,  -0.29090
       3,  19,  22,   2,   2,   2,   7,       0.0
       6,  19,  22,   2,   2,   3,   7,       0.0
      10,  28,  29,   3,  22,   3,  19,   0.14286
      10,  20,  21,  23,  24,   3,   8,   0.14286
      10,  28,  29,  23,  24,   9,  19,   0.14286
      10,  28,  29,  25,  26,   3,  19,   0.14286
      10,   3,   2,   3,  26,   3,  19,   0.14286
      11,   3,   4,   3,   2,   3,  19,   0.14286
      11,   5,   8,   3,   2,   7,  19,   0.14286
      11,   9,  18,   3,   2,   3,  19,   0.14286
      11,  19,  22,   3,   2,   8,  19,   0.14286
      11,  23,  28,   3,   2,   3,  19,   0.14286
      11,   3,  28,  26,  27,   3,  19,   0.14286
      11,  21,  28,  22,  23,   3,   8,   0.14286
      11,  21,  28,  25,  24,   3,   8,   0.14286
      12,   3,  28,   3,  26,  19,  20,   0.14286
      12,  21,  28,  23,  24,   9,   8,   0.14286
      13,  28,  29,   3,  22,   3,  19,       0.0
      13,  20,  21,  23,  24,   3,   8,       0.0
      13,  28,  29,  23,  24,   9,  19,       0.0
      13,  28,  29,  25,  26,   3,  19,       0.0
      13,   3,   2,   3,  26,   3,  19,       0.0
      14,   3,   4,   3,   2,   3,  19,       0.0
      14,   5,   8,   3,   2,   7,  19,       0.0
      14,   9,  18,   3,   2,   3,  19,       0.0
      14,  19,  22,   3,   2,   8,  19,       0.0
      14,  23,  28,   3,   2,   3,  19,       0.0
      14,   3,  28,  26,  27,   3,  19,       0.0
      14,  21,  28,  22,  23,   3,   8,       0.0
      14,  21,  28,  25,  24,   3,   8,       0.0
      15,   3,  28,   3,  26,  19,  20,       0.0
      15,  21,  28,  23,  24,   9,   8,       0.0
9      1
.      21,              28,              23,              24,              3,
```

10	3				
	20, 28, 3,	20, 28, 3,	23, 10, 8,	24, 18, 21,	3, 3, 7,
11	1				
	1,	29,	1,	27,	1,

Input information is written with following format and order.

1 Comment line

format: 20A4

example:

AIR DISTRIBUTION PERFORMANCE FOR 4.0 CFM WITH 75 % FREE AREA DIFFUSER

*this comment is printed on top of LP image listing and the listing of each dependent variable.

2 calculation condition 1

stored at: KSTM, NSTM, MAXT, INIC

format: free

example:

```
 1999000,        20,      288000,          2
```

*KSTM maximum time iteration steps
If accumulated time iteration counter KSTP exceeds KSTM, time iteration is terminated and subroutine OUTPUT is processed.

*NSTM maximum pressure iteration steps
If pressure iteration counter NSTP exceeds NSTM, pressure iteration is unconditionally terminated and next time iteration begins. (recommended value betwen 10 and 30)

*MAXT maximum CPU time in seconds.

Accumulated CPU time is measured once each time iteration and if it exceeds MAXT, time iteration is terminated and subroutine OUTPUT is processed.

Job termination is, therefore, controlled by either KSTM or MAXT unless divergence occurs.

*INIC Input/output unformatted file controller.

For INIC $\geq$ 2, initial flow field is assumed to be stored as an unformatted permanent file on Cyber 855, otherwise, calculation is started from 0 flow field which is generated internally by the subroutine ARAIN.

For INIC $\leq$ 2, final flow field is stored as a new unformatted permanent file on Cyber 855, otherwise, it is lost as soon as the job is terminated.

Recommended value
1 to generate initial flow field using small MAXT or KSTM value.
2 to resume interrupted calculation)

3 <u>calculation condition 2</u>

stored at: RE, PR, AR, BETA, EPS, DT, ENMX
format: free
example:

```
    257710,      0.71,-0.001235,      1.0,    0.0010,     0.010,      10.0
```

*RE	Reynolds number calculated from (see below)
*PR	Prandtl number calculated from (see below)
*AR	Archimedes number calculated from "
*BETA	acceleration coefficient for pressure relaxation iteration (recommended value 1.4)
*EPS	maximum continuity error criteria (recommended value $5 \times 10^{-3} \sim 5 \times 10^{-4}$) too large an EPS sometimes causes instability of calculation.
*DT	non-dimensional time step DT is chosen from the stability criteria for both convection and diffusion terms.
*ENMX	divergence criteria. If volumetric mean energy exceeds ENNX, calculation is unconditionally terminated and subroutine OUTPUT is processed. ENMX is used to detect an abnormal calculation condition or its sympton. (recommended value 10.0)

*normalization of dependent variables

symbol	meaning	example
L_0	length scale	1 m or inlet width
U_0	velocity scale	1 m/s or inlet velocityu
ν	kinemateic viscosity	1.5×10^{-5} m^2/s for normal air
T_0	reference temperature	average interior temperature in °K
ΔT_0	temperature difference scale	1 K or largest possible temperature differnce
β	volumetric expansion coefficient	$1/T^0$ for normal air
α	thermal diffusivity	2.1×10^{-5} m^2/s for normal air
ρ	density	1.2 kg/m^3 for normal air

C_p	specific heat	10.05 kJ/kg· (2.4 kCal/kg·K) for normal air
g	gravitational acceleration	9.8 m/s^2

All dependent variables must be properly normalized prior to calculation using the following independent dimensional units.

symbol	meaning	unit
t_0	time	L_0/U_0
k_0	turbulence energy	U_0^2
ε_0	dissipation rate of turbulence energy	U_0^2/L_0
ν_{t0}	eddy viscosity	$U_0 L_0$
α_0	thermal transfer coefficient	U_0
q_0	thermal flux	$U_0 \Delta T_0$

Dimensional standard units for other dependent variables are automatically chosen in combination with above dimensional units.

4 <u>empirical constants for turbulence model</u>
stored at: ZCD, ZC1, ZC2, ZC3, ZSQ, ZSE, ZSC,
format: free
example:

```
0.09,     1.44,     1.92,     1.00,     1.0,     1.3,     0.9
```

*ZCD, ZC1, ZC2, ZSQ, ZSE, ZSC are equivalent to C_D, C_1, C_2, C_3, σ_k, σ_ε and σ_θ used in equations (2-43), (2-46) and (2-47).

5 <u>cell interval data</u>

stored at arrays: DX,DY, DZ
format: free
example

```
  0.0625,   0.0625,   0.0625,   0.0625,   0.0625,   0.0625,   0.0625,
  0.0625,   0.0625,   0.1250,   0.1250,  0.29774,  0.29774,  0.29774,
 0.29774,  0.29774,  0.29774,  0.17188,  0.17188,  0.17188,  0.17188,
 0.17188,  0.17188,  0.20833,  0.20833,  0.20833,  0.18485,  0.18485,
 0.18485
 0.16667,  0.16667,  0.16667,  0.16667,    0.250,    0.250,  0.41667,
 0.41667,    0.500,    0.500,   0.6250,   0.6250,   0.6250,   0.6250,
  0.6250,   0.6250,   0.6250,   0.6250,  0.41667,  0.41667,  0.41667,
   0.250,    0.250,    0.250,  0.16667,  0.16667,  0.16667
   0.125,    0.125,    0.125,    0.125,    0.125,    0.125,    0.125,
   0.125,    0.125,   0.1875,   0.1875,  0.29167,  0.29167,  0.29167,
 0.29167,  0.16667,  0.16667,    0.125,    0.125,    0.125
```

underlined values are intervals for dummy and terminal real cells.

*non-dimensional cell intervals are written from first cell to terminal cell from X, Y to Z direction.
There must be L, M, and N elements in X,Y and Z direction in correspondence with the values declared at the aforementioned PARAMETER statement. Intervals of the three dummy cells must take the same values as those of the nearest real cell, i.e.

DX(1)=DX(2)=DX(3), DX(L)=DX(L-1)
DY(1)=DY(2)=DY(3), DY(M)=DY(M-1)
DZ(1)=DZ(2)=DZ(3), DZ(N)=DZ(N-1)

6 <u>image listing pointer</u>
stored at array: ICON
format: free
example:

```
        6,          6,          6
```

*Spatial variations of all dependent variables is given as a LP image listing for Y-Z Z-X, X-Y plane at locations where this pointer specifies.
For instance, (14, 2, 2) means that LP listing is given for Y-Z plane at I=14, Z-X plane at J=3 and X-Y plane at K=3.

7 <u>number of boundary conditions</u>
stored at: NIN
format: free
example:

```
       41
```

*NIN is the number of given boundary conditions following this line.

8 <u>boundary condition data</u>
stored at arrays: IIN, UIN
format: free

```
   1,   4,   8,   2,   2,   3,   6,       0.0
   2,   5,   8,   2,   2,   3,   6,       1.0
   3,   5,   8,   2,   2,   2,   6,       0.0
   4,   5,   8,   2,   2,   3,   6,   0.21140
   5,   5,   8,   2,   2,   3,   6,  0.022194
   6,   5,   8,   2,   2,   3,   6,       1.0
   8,  19,  22,   2,   3,   3,   7,       0.0
   1,  18,  22,   2,   2,   3,   7,       0.0
   2,  19,  22,   2,   2,   3,   7,  -0.29090
   3,  19,  22,   2,   2,   2,   7,       0.0
   6,  19,  22,   2,   2,   3,   7,       0.0
  10,  28,  29,   3,  22,   3,  19,   0.14286
  10,  20,  21,  23,  24,   3,   8,   0.14286
  10,  28,  29,  23,  24,   9,  19,   0.14286
  10,  28,  29,  25,  26,   3,  19,   0.14286
  10,   3,   2,   3,  26,   3,  19,   0.14286
  11,   3,   4,   3,   2,   3,  19,   0.14286
  11,   5,   8,   3,   2,   7,  19,   0.14286
  11,   9,  18,   3,   2,   3,  19,   0.14286
```

11,	19,	22,	3,	2,	8,	19,	0.14286
11,	23,	28,	3,	2,	3,	19,	0.14286
11,	3,	28,	26,	27,	3,	19,	0.14286
11,	21,	28,	22,	23,	3,	8,	0.14286
11,	21,	28,	25,	24,	3,	8,	0.14286
12,	3,	28,	3,	26,	19,	20,	0.14286
12,	21,	28,	23,	24,	9,	8,	0.14286
13,	28,	29,	3,	22,	3,	19,	0.0
13,	20,	21,	23,	24,	3,	8,	0.0
13,	28,	29,	23,	24,	9,	19,	0.0
13,	28,	29,	25,	26,	3,	19,	0.0
13,	3,	2,	3,	26,	3,	19,	0.0
14,	3,	4,	3,	2,	3,	19,	0.0
14,	5,	8,	3,	2,	7,	19,	0.0
14,	9,	18,	3,	2,	3,	19,	0.0
14,	19,	22,	3,	2,	8,	19,	0.0
14,	23,	28,	3,	2,	3,	19,	0.0
14,	3,	28,	26,	27,	3,	19,	0.0
14,	21,	28,	22,	23,	3,	8,	0.0
14,	21,	28,	25,	24,	3,	8,	0.0
15,	3,	28,	3,	26,	19,	20,	0.0
15,	21,	28,	23,	24,	9,	8,	0.0

*Default boundary condition employed in the "EXACT" code is the symmetric wall condition described in the next subsection.

*Since boundary condition data are referenced in regular sequence, earlier condition including default boundary condition may be overwritten by the latter one.
*Boundary condition data consist of boundary condition number, boundary location and boundary parameter.
General format of boundary condition data is given as follows.

Where III is the boundary condition number I1, I2, J1, J2, K1, K2 are the boundary location in X, Y and Z-direction in cell number, SEV is the boundary parameter.

*Boundary condition data for III-1 6 are used to impose specified boundary value to each dependent variable with the location I=I1 to I2, J=J1 to J2, K=K1 to K2 in cell number

III	function	effect
1	specify velocity component in X direction	U(I,J,K) = SERV I=I1 to I2, J=J1 to J2, K= K1 to K2
2	specify velocity component in Y direction	V(I,J,K) = SERV I=I1 to I2, J=J1 to J2, K= K1 to K2
3	specify velocity component in Z direction	W(I,J,K) = SERV I=I1 to I2, J=J1 to J2, K= K1 to K2
4	specify turbulence energy	Q(I,J,K) = SERV I=I1 to I2, J=J1 to J2, K= K1 to K2
5	specify dissipation rate of turbulence energy	E(I,J,K) = SERV I=I1 to I2, J=J1 to J2, K= K1 to K2
6	specify temperature difference	C(I,J,K) = SERV I=I1 to I2, J=J1 to J2, K= K1 to K2

*Boundary conditions for III = 7 to 9 are prepared for pressure boundary condition. Pressure boundary is located on the cell surface between dummy cell and terminal real cell. Since velocity component normal to the pressure boundary is determined from calculation, work array to conserve velocity component are released prior to calculation to allow temporal velocity variation. For II=7, I1 is the dummy cell number where pressure is defined outside the real flow domain and I2 specifies the location where work array to conserve U value on the pressure boundary is released. Hence I1 =I2 or I1 + 1 = I2 are permitted in this case.

III	function	effect
7	specify pressure boundary normal to X axis	P(I,J,K) = SERV I=I1, J=J1 to J2, K= K1 to K2 SX(I,J,K) = 1.0
8	specify pressure boundary normal to Y axis	P(I,J,K) = SERV I=I1 to I2, J=J1, K= K1 to K2 SY(I,J,K) = 1.0
9	specify pressure boundary normal to Z axis	P(I,J,K) = SERV I=I1 to I2, J=J1 to J2, K= K1 SZ(I,J,K) = 1.0

where SX, SY, SZ are work arrays to conserve U,V, W.

*Boundary conditions for III=10 to 12 are prepared to set power law type wall boundary condition in which equation 3-42 is assumed. SEV is in this case equivalent to n in equation 3-42 and a value of 1/7 is commonly used for the fully developed turbulent boundary layer. Wall boundary is located between terminal real cell and dummy cell.

III	function	boundary location
10	specify wall boundary normal to X axis	between I=I1 and I2 for J= J1 to J2, K= K1 to K2 I1 is a terminal real cell and I2 is an adjacent dummy cell numbrt
11	specify wall boundary normal to Y axis	between J=J1 and J2 for I= I1 to I2, K= K1 to K2 J1 is a terminal real cell and J2 is an adjacent dummy cell number
12	specify wall boundary normal to Z axis	between K=K1 and K2 for I= I1 to I2, J= J1 to J2 I1 is a terminal real cell and I2 is an adjacent dummy cell number

*Boundary conditions for III= 13 to 15 are preapred to set thermal flux across the wall boundary appearing in equation 3-47. Thermal flux, SEV, takes positive value when its direction is from dummy cell to real cell.

III	function	boundary location
13	specify wall boundary normal to X axis	same as III = 10

14	specify wall boundary normal to Y axis	same as III = 11
15	specify wall boundary normal to Z axis	same as III = 12

*Boundary conditions for III= 16 to 18 are used to set local thermal transfer coefficient defined in equation 3-48.
Temperature must be specified using III=6 boundary condition at the wall adjacent dummy cell before this line appears.
Non-dimensional thermal transfer coefficient is stored at SEV.

III	function	boundary location
16	specify wall boundary normal to X axis	same as III= 10
17	specify wall boundary normal to Y axis	smae as III= 11
18	specify wall boundary normal to Z axis	same as III= 12

* Boundary conditions for III= 19 to 21 are used to specify the wall temperature defined in equatins 3-49

III	function	boundary location
19	specify wall temperatue in X plane	same as III= 10
20	specify wall temperature in Y plane	smae as III= 11
21	specify wall temperatue in Z plane	same as III= 12

* Boundary conditions for III= 22 to 24 are used to specify the K slippage condition per equation (3-51)

III	function	boundary location
22	specify k slippage normal to X axis	same as III= 10
23	specify k slippage normal to Y axis	smae as III= 11
24	specify k slippage normal to Z axis	same as III= 12

9 obstacle location data

stored at :NOBS/ IOB
format :free
example

```
1
```

```
       21,        28,        23,        24,         3,         8
```

*When there are obstacles in the flow domain, such as furniture, dependent variables within and on the surface of obstacles are made unchanged in the time iteration step by setting the obstacle location data. (wall boundary condition, velocity component normal to the obstacle wall, thermal flux across the obstacles have to be imposed by setting boundary condition data if necessary)
The obstacle location data are written as follows:

NOBS/ I1, I2, J1, J2, K1, K2
where NOBS is the number of the obstacles and following location data,

If NOBS = 0 the whole flow domain is assumed to be filled with fluid and there must not be location data. Cells within the range of I=I1 to I2, J=J1 to J2, K=K1 to K2 in cell number are treated as dummy cells.

10 <u>Heat Generation Rate Data</u>

stored at NSOR
format free
example

```
        3
```

stored at I1,I2,J1,J2,K1,K2,SERV
format: free
example

```
       20,        20,        23,        24,         3,         8, -1.26057
       28,        28,        10,        18,         3,        11, -0.034417
        3,         3,         8,        21,         7,        16, -0.056347
```

NSOR is the number of heat generation sources. If there are none, NSOR = 0.

The each volume heat generation rate is set:
H(I,J,K) = SERV in the region defined by
for I= I1 to I2, J= J1 to J2 and K= K1 to K2

<u>11 upwind interpolation data</u>

stored at NCR
format :free
example

```
        1
```

stored at ICR, CR
format: free
example

```
        1,        29,         1,        27,         1,        20,       2.0
```

*The range and magnitude of the switching parameter for the upwind/central combination scheme described in 3-1 are controlled by upwind interpolation data. The upwind interpolation data are written as follows.

NCR is the number upwind interpolation data and the following location and switching parameter I1, I2, J1, J2, K1, K2, CR is read NCR times. If NCR = 0 all convection terms in the momentum equation are approximated using central difference scheme, and there must not be location and switching parameter. Convection terms in the momentum equation within the range of I= I1 to I2, J= J1 to J2, K= K1 to K2 in cell number are upwinded with the factor of CR.

A.4 Batch job stream

Since computation on Cyber 205 is conducted only through a batch job on the Cyber 855, the batch job stream is prepared and submitted on the Cyber 855. An example of the batch job stream and its format will be given briefly.

line	function
1...4	job control cards and user identification, user account information is assumed to be stored on Cyber 855 as a permanent file names USER.
5	job category and limit information JCAT, TL, LP parameters may be replaced according to the volume of calculation.
6...11	initial flow field is assumed to be stored as an unformatted permanent file named AAAAA on Cyber 855. AAAAA is accessed from Cyber 205 and made a temporary file named UNIT1. UNIT 1 is converged to a 64bit local file, UNIT2 by a permanent load module C6064, which is a copy of public file, prepared for data convergion from Cyber 855 to Cyber 205 Finally, UNIT2 is renamed UNIT22 and prepared for calculation.
12	compile of EXACT3 code by FTN200 compiler
13	input data is assumed to be stored as a formatted permanent file named INDATA INDATA is accessed from Cyber 205 and made a local file named UNIT5
14...16	load and go
17...23	final calculated flow field, UNIT21, is renamed UNIT1, converted to a 60 bit data by *c6460, then sent and saved as a new unformatted permanent file named BBBB
24	LP image listing, UNIT6, is copied to a local file OUTPUT. OUTPUT information together with program source listing, load map, and dayfile may be accessed from the Cyber 855 using QGET command
25...28	specification of the program and input data name.

Sample Batch File for Cyber 205

```
/JOB
/NOSEQ
FLOW,ST205
/READ,USER
RESOURCE,JCAT=P3,TL=2720,LP=20.
MFLONK,UNIT1,ST=CS2,DD=UU,JCS="USER,LISHI,TAKASHI."
"ATTACH,TMFK02."
REQUEST,UNIT2/300,RT=W.
ATTACH,C6064.
C6064.
SWITCH,UNIT2,UNIT22.
FTN200,OPT=1,SC=1,UNSAFE=1.
COPY,NEWDUMP,DUMP.
COPY,INPUT,UNIT%.
LOAD(GRLP=*VEL).
COPY,NEWDUMP,DUMP.
GO.
RETURN,UNIT1.
SWITCH,UNIT21,UNIT1.
REQUEST,UNIT2/300,TR=W.
ATTACH,C6460.
C6460.
MFLINK,UNIT2,ST=CS2,DD=UU,JCS="USER,LISHI,TAKASHI.",
"DEFINE,MM!K03."
COPY,UNIT^,OUTPUT.
FILES,PRIVATE=*.
/EOR
/READ,EXACT#
/EOR
/READ,SEAIRIK
```

This batch job strean, named SUBFILE is submitted using following command

SUBMIT,SUBFILE,E.

On UNIX and extended DOS 386-computer no batch job is required. One must only compile and link the source code. Create the data file describing the problem and run the compiled code as
EXACT3 < DataFile
where DataFile is the name of the file describing the problem. Output will be to the screen and also to a file name EXACT3.DAT. If this file is not renamed it will be over written.

A.5 Listing of EXACT3

The following is a listing of EXACT3 for the Compaq 386/25 compiler with SVS FORTRAN-386. Only the time function ISCONDS must be rewritten for other compiler (since FORTRAN does not have a standard time function).

```
C
C                        ****************************
C                        *  EXACT3 (R. K-E MODEL)  *
C                        ****************************
C-------------------------------------------------------
--------------------
C                   EXPLICIT TIME-MARCHING ARGORITHM FOR
C                   CONTINUOUS THERMAL FLUID FLOW
C
C          ----------CODED BY T.KURABUTI,1987
C          CODE REVISED BY R. GROT 1988 TO REDUCE STORAGE
REQUIREMENTS
C-------------------------------------------------------
--------------------
C                   INPUT DATA FORMAT
C--------1---------2---------3---------4---------5-----
----6---------7--
C               TITLE
C     KSTM      NSTM      MAXT      INIC
C        RE           PR          AR          BETA         EPS
DT      ENMX
C        CD           C1          C2          C3           SQ
SE        SC
C       DX   (7F10.0)
C       DY   (7F10.0)
C       DZ   (7F10.0)
C       I1           J1          K1
C      NIN
C      IIN  (7I5    )          UIN   (F10.0)
C     NOBS
C        I1           I2          J1          J2           K1
K2
C     NSOR
C        I1           I2          J1          J2           K1
K2
C      NCR
C        I1           I2          J1          J2           K1
K2        CR
C--------1---------2---------3---------4---------5-----
----6---------7--
C
C
C*****MAIN
C
      PROGRAM MAIN
C
C*****DECLARATION OF COMMON
C
C
      PARAMETER (L=29,M=27,N=20)
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
C
      COMMON /SCAL/ ICON(3)
C
      DIMENSION R1(L+1),R2(M+1),R3(N+1)
C
      OPEN (22,FILE='UNIT22',FORM='UNFORMATTED')
      OPEN (10,FILE='MEXACT3.OUT',FORM='FORMATTED')
C
C*****ARRAY INITIALIZATION
C
      CALL ARAIN
C
C*****DATA INITIALIZATION
C
      CALL DATAIN
C
      IF ( INIC .LE. 1 ) GO TO 100
C
      READ(22) U,V,W,P,Q,E,C,R1,R2,R3,KSTP
C
  100 CONTINUE
C
C*****SOLVER
```

```
C
      CALL SOLVER
C
      IF ( INIC .EQ. 0 ) GO TO 200
      IF ( INIC .EQ. 3 ) GO TO 200
C
      OPEN (21,FILE='UNIT21',FORM='UNFORMATTED')
      WRITE(21) U,V,W,P,Q,E,C,DX,DY,DZ,KSTP
      CLOSE(21)
C
  200 CONTINUE
C
C*****JOB TERMINATION
C
      CALL OUTPUT
C
      STOP
      END

      SUBROUTINE DATAIN
C
C*****DECRALATION OF COMMON
C
C
      PARAMETER (L=29,M=27,N=20)
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      COMMON /SCAL/ ICON(3)
C
      DIMENSION ICR(10,6),CR(10)
C
C*****OBSTACLE SIGN INITIALIZATION
C
      DO 250 K=3,N-1
      DO 250 J=3,M-1
      DO 250 I=3,L-2
  250 SX (I,J,K) = 1
C
      DO 260 K=3,N-1
      DO 260 J=3,M-2
      DO 260 I=3,L-1
  260 SY (I,J,K) = 1
C
      DO 270 K=3,N-2
      DO 270 J=3,M-1
      DO 270 I=3,L-1
  270 SZ (I,J,K) = 1
C
C*****DATA INPUT
C
      READ(5,100)   TITLE
      WRITE(6,600)   TITLE
      WRITE(10,600)   TITLE
      READ(5,*)  KSTM,NSTM,MAXT,INIC
      WRITE(6,610)  KSTM,NSTM,MAXT,INIC
      WRITE(10,610)  KSTM,NSTM,MAXT,INIC
      READ(5,*)  RE  ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
      WRITE(6,620)  RE  ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
      WRITE(10,620)  RE  ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
      READ(5,*)  ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC
      WRITE(6,630)  ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC
      WRITE(10,630)  ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC
      READ(5,*)   (DX(I),I=1,L)
      WRITE(6,641)   (DX(I),I=1,L)
      WRITE(10,641)   (DX(I),I=1,L)
      READ(5,*)   (DY(J),J=1,M)
      WRITE(6,642)   (DY(J),J=1,M)
      WRITE(10,642)   (DY(J),J=1,M)
      READ(5,*)   (DZ(K),K=1,N)
      WRITE(6,643)   (DZ(K),K=1,N)
      WRITE(10,643)   (DZ(K),K=1,N)
      READ(5,*)   (ICON(I),I=1,3)
      WRITE(6,650)   (ICON(I),I=1,3)
      WRITE(10,650)   (ICON(I),I=1,3)
```

```
C
C*****BOUNDARY CONDITION INPUT
C
      READ(5,*)  NIN
      DO 10 I=1,NIN
   10 READ(5,*)  (IIN(I,J),J=1,7),UIN(I)
C
C
      WRITE(6,660)
      WRITE(10,660)
      DO 40 I=1,NIN
      WRITE(6,661)  (IIN(I,J),J=1,7),UIN(I)
      WRITE(10,661)  (IIN(I,J),J=1,7),UIN(I)
   40 CONTINUE
C
C*****OBSTACLE BOUNDARY CONDITION
C
      READ(5,*)  NOBS
C
      IF ( NOBS .EQ. 0 ) GO TO 25
C
      DO 20 III=1,NOBS
      READ(5,*)(IOB(III,J),J=1,6)
C
      I1 = IOB(III,1)
      I2 = IOB(III,2)
      J1 = IOB(III,3)
      J2 = IOB(III,4)
      K1 = IOB(III,5)
      K2 = IOB(III,6)
C
      DO 21 K=K1  ,K2
      DO 21 J=J1  ,J2
      DO 21 I=I1-1,I2
   21 SX (I,J,K) = 0
C
      DO 22 K=K1  ,K2
      DO 22 J=J1-1,J2
      DO 22 I=I1  ,I2
   22 SY (I,J,K) = 0
C
      DO 23 K=K1-1,K2
      DO 23 J=J1  ,J2
      DO 23 I=I1  ,I2
   23 SZ (I,J,K) = 0
C
   20 CONTINUE
C
   25 CONTINUE
C
      IF ( NOBS .EQ. 0 ) GO TO 55
      WRITE(6,670)
      WRITE(10,670)
      DO 50 I=1,NOBS
      WRITE(6,671)  (IOB(I,J),J=1,6)
      WRITE(10,671)  (IOB(I,J),J=1,6)
   50 CONTINUE
   55 CONTINUE
C
C
C*****THERMAL HEAT SOURCE DATA
C
      READ(5,*) NSOR
      IF(NSOR .EQ. 0) WRITE(*,690)
  690 FORMAT(1H ,10X,'**** NO HEAT SOURCES ****',/)
      IF(NSOR .NE. 0) THEN
      WRITE(*,691) NSOR
  691 FORMAT(1H ,10X,'**** HEAT SOURCES *****',/,10X,
     1'NUMBER = ',I4,/)
      WRITE(*,692)
   692  FORMAT(1H1,'....I1....I2....J1....J2....K1....K2
SOURCE')
      DO 255 II=1,NSOR
      READ(5,*) (ISOR(II,J),J=1,6),SOR(II)
      WRITE(*,681) (ISOR(II,J),J=1,6),SOR(II)
  255 CONTINUE
      ENDIF
C
C
C*****CRITICAL MESH REYNOLDS NUMBER INPUT
C
      READ(5,*)  NCR
C
      IF ( NCR  .EQ. 0 ) GO TO 35
C
      DO 30 III=1,NCR
      READ(5,*)(ICR(III,J),J=1,6),CR(III)
C
      I1 = ICR(III,1)
      I2 = ICR(III,2)
      J1 = ICR(III,3)
      J2 = ICR(III,4)
      K1 = ICR(III,5)
      K2 = ICR(III,6)
C
      DO 31 K=K1,K2
      DO 31 J=J1,J2
      DO 31 I=I1,I2
   31 CMR(I,J,K) = CR (III)
C
   30 CONTINUE
C
   35 CONTINUE
C
C
      IF ( NCR .EQ. 0 ) GO TO 65
      WRITE(6,680)
      WRITE(10,680)
      DO 60 I=1,NCR
      WRITE(6,681)  (ICR(I,J),J=1,6),CR(I)
      WRITE(10,681)  (ICR(I,J),J=1,6),CR(I)
   60 CONTINUE
   65 CONTINUE
C
C
C
C*****SET HX(I),HY(J),XD(I),YD(J),XIP(I),YIP(J)
C
      DO 200 I=1,L
      HX (I) = 1.0    /  DX(I)
      XD (I) = 2.0    /( DX(I) + DX(I+1) )
      XIP(I) = DX(I+1)/( DX(I) + DX(I+1) )
  200 CONTINUE
C
      DO 201 J=1,M
      HY (J) = 1.0    /  DY(J)
      YD (J) = 2.0    /( DY(J) + DY(J+1) )
      YIP(J) = DY(J+1)/( DY(J) + DY(J+1) )
  201 CONTINUE
C
      DO 202 K=1,N
      HZ (K) = 1.0    /  DZ(K)
      ZD (K) = 2.0    /( DZ(K) + DZ(K+1) )
      ZIP(K) = DZ(K+1)/( DZ(K) + DZ(K+1) )
  202 CONTINUE
C
C*****PROTECTOR SET FOR BOUNDARY CONDITION
C
      DO 210 III=1,NIN
C
      I1 = IIN(III,2)
      I2 = IIN(III,3)
      J1 = IIN(III,4)
      J2 = IIN(III,5)
      K1 = IIN(III,6)
      K2 = IIN(III,7)
C
      IF ( IIN(III,1) .EQ. 1 ) THEN
      DO 205 K=K1,K2
      DO 205 J=J1,J2
      DO 205 I=I1,I2
  205 SX(I,J,K) = 0
      ENDIF
C
      IF ( IIN(III,1) .EQ. 2 ) THEN
      DO 206 K=K1,K2
      DO 206 J=J1,J2
      DO 206 I=I1,I2
  206 SY(I,J,K) = 0
      ENDIF
C
      IF ( IIN(III,1) .EQ. 3 ) THEN
      DO 207 K=K1,K2
      DO 207 J=J1,J2
      DO 207 I=I1,I2
  207 SZ(I,J,K) = 0
      END IF
C
      IF ( IIN(III,1) .EQ. 7 ) THEN
      DO 211 K=K1,K2
      DO 211 J=J1,J2
  211 SX(I2,J,K) = 1
      END IF
C
      IF ( IIN(III,1) .EQ. 8 ) THEN
      DO 212 K=K1,K2
      DO 212 I=I1,I2
  212 SY(I,J2,K) = 1
      END IF
C
      IF ( IIN(III,1) .EQ. 9 ) THEN
      DO 213 J=J1,J2
      DO 213 I=I1,I2
  213 SZ(I,J,K2) = 1
      END IF
C
  210 CONTINUE
C
C*****INITIAL VALUE FOR SCALAR QUANTITY ( FOR INIC = 0,1
)
C
      QINI = 0.1
      EINI = ZCD * QINI ** 2 * RE
      HRE = 1.0 / RE
```

```
C
      DO 300 K=1,N
      DO 300 J=1,M
      DO 300 I=1,L
C
      D(I,J,K) = HRE
      Q(I,J,K) = QINI
      E(I,J,K) = EINI
C
  300 CONTINUE
C
C*****INITIAL CALCULATION 1
C
      VOLM = (L-3) * (M-3) * (N-3)
      VOLM = 1.0 / VOLM
C
      HZSQ = 1.0 / ZSQ
      HZSE = 1.0 / ZSE
      HZSC = 1.0 / ZSC
      PE = RE * PR
      HPE = 1.0 / PE
C
C*****INITIAL CALCULATION 2
C
      DO 400 K=3,N-1
      DO 400 J=3,M-1
      DO 400 I=3,L-1
C
     DIVE = DT * ( ( XD(I)*SX(I,J,K) + XD(I-1)*SX(I-1,J,K))
* HX(I)
     1    +       ( YD(J)*SY(I,J,K) + YD(J-1)*SY(I,J-1,K))
* HY(J)
     2    +       ( ZD(K)*SZ(I,J,K) + ZD(K-1)*SZ(I,J,K-1))
* HZ(K) )
C
      BE(I,J,K) = 0.0
C
      IF ( DIVE .NE. 0.0 ) BE(I,J,K) = -BETA / DIVE
C
  400 CONTINUE
C
C*****FORMAT STATEMENT
C
C*****READ
C
  100 FORMAT (20A4    )
  110 FORMAT ( 4I10   )
  120 FORMAT ( 7F10.0 )
  130 FORMAT ( 7F10.0 )
  140 FORMAT ( 3I10   )
  150 FORMAT (  I10   )
  160 FORMAT ( 7I 5   ,F10.0 )
  170 FORMAT ( 6I10   )
  180 FORMAT ( 6I10   ,F10.0 )
C
C*****WRITE
C
  600 FORMAT (1H1////1H ,20X,20A4,10X////)
C
  610 FORMAT (
      1          1H ,2X,'KSTM;.......TIME ITERATION MAX
NO.=',I15/
      2         ,1H ,2X,'NSTM;...PRESSURE ITERATION MAX
NO.=',I15/
   3    ,1H ,2X,'MAXT;.........CPU TIME MAX (SEC.)=',I15/
   4    ,1H ,2X,'INIC;....INITIAL FILE DATA CONTROL=',I15/
)
C
  620 FORMAT (
       1         1H ,2X,'RE   ;......LAMINAR REYNOLDS
NUMBER=',F15.7/
        2        ,1H ,2X,'PR   ;......LAMINAR PRANDTL
NUMBER=',F15.7/
        3        ,1H ,2X,'AR   ;...........ARCHIMEDES
NUMBER=',F15.7/
        4        ,1H ,2X,'BETA;..ACCERATE RELAXIZATION
CONST=',F15.7/
        5        ,1H ,2X,'EPS ;...DIVERGENCE TOLERANCE
LIMIT=',F15.7/
         6        ,1H ,2X,'DT    ;................TIME
DIVISION=',F15.7/
        7        ,1H ,2X,'ENMX;....CONVERGENCE JUDGING
CONST=',F15.7/   )
C
  630 FORMAT (
             1                       1H   ,2X,'ZCD   ;
',F15.7/
             2                      ,1H   ,2X,'ZC1   ;
',F15.7/
             3                      ,1H   ,2X,'ZC2   ;
',F15.7/
             4                      ,1H   ,2X,'ZC3   ;
',F15.7/
             5                      ,1H   ,2X,'ZSQ   ;
',F15.7/
             6                      ,1H   ,2X,'ZSE   ;
',F15.7/
             7                      ,1H   ,2X,'ZSC   ;
```

```
',F15.7/ )
C
  641 FORMAT (1H ,'****DX'/,6(10F6.3/) )
  642 FORMAT (1H ,'****DY'/,6(10F6.3/) )
  643 FORMAT (1H ,'****DZ'/,6(10F6.3/) )
C
  650 FORMAT (1H ,/,10X,'*****DATA OUTPUT POINT*****',
//
     1            ,10X,'(',I10,',',I10,',',I10,')', / )
C
  660 FORMAT (1H ,/,10X,'*****BOUNDARY CONDITION*****',
//
     1        '...NOB....I1,...I2,...J1'
     2       ,'....J2....K1,...K2.........SPV',/ )
  661 FORMAT (1H ,7I6,F12.5)
C
 670 FORMAT (1H ,/,10X,'*****OBSTACLE CONDITION*****',//
     1        '....I1...,I2,...J1'
     2       ,'....J2....K1....K2',/ )
  671 FORMAT (1H ,6I6)
C
      680   FORMAT   (1H   ,/,10X,'*****CRITICAL   REYNOLDS
NO.*****'//
     1        '....I1....I2....J1....J2'
     2       ,'....K1....K2.........CMR'  /)
  681 FORMAT (1H ,6I6,F12.5)
C
      RETURN
      END


      SUBROUTINE ARAIN
C
C*****DECLARATION OF COMMON
C
      PARAMETER (L=29,M=27,N=20)
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      COMMON /SCAL/ ICON(3)
C
C
C*****ARRAY INITIALIZATION
C
      DO 100 K=1,N
      DO 100 J=1,M
      DO 100 I=1,L
C
      U   (I,J,K) = 0.0
      V   (I,J,K) = 0.0
      W   (I,J,K) = 0.0
      UU  (I,J,K) = 0.0
      VV  (I,J,K) = 0.0
      WW  (I,J,K) = 0.0
      P   (I,J,K) = 0.0
      D   (I,J,K) = 0.0
      S   (I,J,K) = 0.0
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
      Q   (I,J,K) = 0.0
      E   (I,J,K) = 0.0
      C   (I,J,K) = 0.0
      SX  (I,J,K) = 0
      SY  (I,J,K) = 0
      SZ  (I,J,K) = 0
      BE  (I,J,K) = 0.0
C
  100 CONTINUE
C
      DO 130 J=1,60
      DO 130 I=1,10
      DA(I,J) = 0.0
  130 CONTINUE
C
```

```
      DO 150 K=1,N
      DO 150 J=1,M
      DO 150 I=1,L
      CMR (I,J,K) = 100000.0
  150 CONTINUE
C
      DO 152 I=1,L+1
  152 DX(I) = 0.0
C
      DO 153 J=1,M+1
  153 DY(J) = 0.0
C
      DO 154 K=1,N+1
  154 DZ(K) = 0.0
C
C*****PROTECTER INITIALIZE
C
      RETURN
C
      END

C*****ROOM AIR DISTRIBUTION SOLVER
C
      SUBROUTINE SOLVER
C
      PARAMETER (L=29,M=27,N=20)
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      COMMON /SCAL/ ICON(3)
C
     DIMENSION  UY(L,M,N),UZ(L,M,N),VZ(L,M,N),VX(L,M,N),
    1WX(L,M,N),WY(L,M,N),QQ(L,M,N), EE(L,M,N), CC(L,M,N),
     2XY(L,M,N),YZ(L,M,N),ZX(L,M,N)
C
                                                        EQUIVALENCE
(AFX(1,1,1),UY(1,1,1),XY(1,1,1),VZ(1,1,1),YZ(1,1,1)
     1,UZ(1,1,1),ZX(1,1,1))
C
                                                        EQUIVALENCE
(DFX(1,1,1),VX(1,1,1),WY(1,1,1),WX(1,1,1))
C
      EQUIVALENCE (UU(1,1,1),QQ(1,1,1))
C
      EQUIVALENCE (VV(1,1,1),EE(1,1,1))
C
      EQUIVALENCE (WW(1,1,1),CC(1,1,1))
C
C*****BCON RESET
C
      WRITE(6,3000)
      WRITE(10,3000)
 3000 FORMAT(1H ,"    Starting Solution of Navier Stokes
Equations",/)
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
      U(I,J,K) = U(I,J,K) * SX(I,J,K)
      V(I,J,K) = V(I,J,K) * SY(I,J,K)
      W(I,J,K) = W(I,J,K) * SZ(I,J,K)
 1000 CONTINUE
C
      CALL BCONSUB
      WRITE(6,3020)
      WRITE(10,3020)
 3020 FORMAT(1H ,"Set Initial Boundary Conditions")
      IKSTP = KSTP
      KSTP = KSTP-1
C
      IF ( INIC .EQ. 1 ) THEN
      IKSTP = 1
      KSTP = 0
      ENDIF
      NSTP = 0
      KTIM = 0
      QAVE = 0.0
      EAVE = 0.0
      CAVE = 0.0
      DAVE = 0.0
      PAVE = 0.0
      ENAV = 0.0
      DMAX = 0.0
C
      JTIME1 = ISECND(1)
      WRITE(6,3022) JTIME1
      WRITE(10,3022) JTIME1
 3022 FORMAT(1H ,"Initial Time is ",I15," seconds")
C
C*****WORK VECTOR INITIALIZATION
C
      DO 1010 I=1,L
      DO 1010 J=1,M
      DO 1010 K=1,N
      DFX(I,J,K) = 0.0
      AFX(I,J,K) = 0.0
        S(I,J,K) = 0.0
 1010 CONTINUE
C
C*****DO LOOP RETURN
      WRITE(6,3021)
      WRITE(10,3021)
 3021 FORMAT(1H ,"Starting Solution Loop")
C
  200 CONTINUE
C
      KSTP = KSTP + 1
C
C*****TERMINATION CRITERIA
C
     IF ( (KSTP.GT.KSTM) .OR. (KTIM.GT.MAXT) ) GO TO 2000
  203 CONTINUE
C
      DO 1020 I=1,L
      DO 1020 J=1,M
      DO 1020 K=1,N
      S(I,J,K) = 0.0
 1020 CONTINUE
C
C*****SET S
      DO 10 K=2,N-1
      DO 10 J=2,M-1
      DO 10 I=2,L-1
C
      UX = ( U(I,J,K) - U(I-1,J,K) ) * HX(I)
      VY = ( V(I,J,K) - V(I,J-1,K) ) * HY(J)
      WZ = ( W(I,J,K) - W(I,J,K-1) ) * HZ(K)
C
      S (I,J,K) = 2.0 * ( UX**2 + VY**2 + WZ**2 )
   10 CONTINUE
C*****CALCULATION OF XY
C
      DO 1030 I=1,L
      DO 1030 J=1,M
      DO 1030 K=1,N
      DFX(I,J,K) = 0.0
      AFX(I,J,K) = 0.0
 1030 CONTINUE
      DO 15 K=1,N-1
      DO 15 J=1,M-1
      DO 15 I=1,L-1
C
      UY(I,J,K) = ( U(I,J+1,K) - U(I,J,K) ) * YD(J)
      VX(I,J,K) = ( V(I+1,J,K) - V(I,J,K) ) * XD(I)
C
   15 CONTINUE
C
      DO 1040 I=1,L
      DO 1040 J=1,M
      DO 1040 K=1,N
      XY(I,J,K) = UY(I,J,K) + VX(I,J,K)
 1040 CONTINUE
C
C*****ADD XY TO S
C
      DO 20 K=2,N-1
      DO 20 J=2,M-1
      DO 20 I=2,L-1
C
C
      S (I,J,K) = S(I,J,K)
    1 + ( XY(I,J,K)+XY(I-1,J,K)+XY(I,J-1,K)+XY(I-1,J-1,K)
)**2 / 16.0
   20 CONTINUE
```

```
C
C*****CALCULATION OF YZ,
C
      DO 25 K=1,N-1
      DO 25 J=1,M-1
      DO 25 I=1,L-1
C
      VZ(I,J,K) = ( V(I,J,K+1) - V(I,J,K) ) * ZD(K)
      WY(I,J,K) = ( W(I,J+1,K) - W(I,J,K) ) * YD(J)
C
   25 CONTINUE
C
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      YZ(I,J,K) = VZ(I,J,K) + WY(I,J,K)
 1050 CONTINUE
C
C*****ADD YZ TO S
C
      DO 26 K=2,N-1
      DO 26 J=2,M-1
      DO 26 I=2,L-1
C
      S (I,J,K) = S(I,J.K)
   2 + ( YZ(I,J,K)+YZ(I,J-1,K)+YZ(I,J,K-1)+YZ(I,J-1,K-1)
)**2 / 16.0
C
   26 CONTINUE
C
C*****CALCULATION OF ZX
C
      DO 27 K=1,N-1
      DO 27 J=1,M-1
      DO 27 I=1,L-1
C
      UZ(I,J,K) = ( U(I,J,K+1) - U(I,J,K) ) * ZD(K)
      WX(I,J,K) = ( W(I+1,J,K) - W(I,J,K) ) * XD(I)
C
   27 CONTINUE
C
      DO 1060 I=1,L
      DO 1060 J=1,M
      DO 1060 K=1,N
      ZX(I,J,K) = WX(I,J,K) + UZ(I,J,K)
 1060 CONTINUE
C
C*****ADD ZX TO S
C
      DO 28 K=2,N-1
      DO 28 J=2,M-1
      DO 28 I=2,L-1
C
      S (I,J,K) = S(I,J,K)
   3 + ( ZX(I,J,K)+ZX(I,J,K-1)+ZX(I-1,J,K)+ZX(I-1,J,K-1)
)**2 / 16.0
C
   28 CONTINUE
C
      IF(KSTP.EQ.1) WRITE(6.3001)
      IF(KSTP.EQ.1) WRITE(10,3001)
 3001 FORMAT(1H ,"The matrix S is formed ")
C
C*****SET D
C
      DO 1070 I=1,L
      DO 1070 J=1,M
      DO 1070 K=1,N
      IF(E(I,J,K).LE.0.0) THEN
      D(I,J,K) = 0.0
C
      ELSE
      D(I,J,K) = ZCD*Q(I,J,K)*Q(I,J,K)/E(I,J,K)
C
      END IF
 1070 CONTINUE
      IF(KSTP.EQ.1) WRITE(6,3002)
      IF(KSTP.EQ.1) WRITE(10,3002)
 3002 FORMAT(1H ,"The diffusivity has been calulated")
C
C*****SCALAR VARIABLE UPDATE***************************
C
      DO 1080 I=1,L
      DO 1080 J=1,M
      DO 1080 K=1,N
      QQ(I,J,K) = 0.0
      EE(I,J,K) = 0.0
      CC(I,J,K) = 0.0
 1080 CONTINUE
C
      CALL SOLVEX
      IF(KSTP.EQ.1) WRITE(6,3003)
      IF(KSTP.EQ.1) WRITE(10,3003)
 3003 FORMAT(1H ,"Finished Scalar X Fluxes")
      CALL SOLVEY
      IF(KSTP.EQ.1) WRITE(6,3004)
      IF(KSTP.EQ.1) WRITE(10,3004)
 3004 FORMAT(1H ,"Finished Scalar Y Fluxes")
      CALL SOLVEZ
      IF(KSTP.EQ.1) WRITE(6,3005)
      IF(KSTP.EQ.1) WRITE(10,3005)
 3005 FORMAT(1H ,"Finished Scalar Z Fluxes")
C
      DO 1090 I=1,L
      DO 1090 J=1,M
      DO 1090 K=1,N
      DFX(I,J,K) = 0.0
 1090 CONTINUE
C
C*****DETERMINE UPDATE OF Q,E,C
C
      DO 55 K=3,N-1
      DO 55 J=3,M-1
      DO 55 I=3,L-1
C
C*****CALCULATION OF TEMPERATURE DIFFERENCE
C
      DFX(I,J,K) = ( ( C(I,J,K)-C(I-1,J,K) ) * XD(I-1) *
SX(I-1,J,K)
     1          +    ( C(I+1,J,K)-C(I,J,K) ) * XD(I   ) *
SX(I   ,J,K)
     2              ) * 2.0 / ( SX(I-1,J,K) + 1.0 ) / (
SX(I,J.K) + 1.0 )
C
   55 CONTINUE
C
C*****CALCULATION OF DQDT & DEDT
C
      DO 1100 I=3,L-1
      DO 1100 J=3,M-1
      DO 1100 K=3,N-1
      IF(Q(I,J,K).GT.0.0) THEN
C
      QQ(I,J,K) = QQ(I,J,K)
     1           + D (I,J,K) * S (I,J,K)
     2           - E (I,J,K)
     3           + D (I,J,K) * DFX(I,J,K) * AR * HZSC
C
      EE(I,J,K) = EE(I,J,K)
     1        + (   D (I,J,K) * S (I,J,K)   * ZC1
     2           - E (I,J,K) * ZC2
     3           + D (I,J,K) * DFX(I,J,K) * AR * HZSC *
ZC3
     4         ) * E (I,J,K) / Q  (I,J,K)
C
      END IF
 1100 CONTINUE
C
C THERMAL SOURCE TERMS
C
      IF(NSOR.NE.0) THEN
      DO 1300 III = 1,NSOR
      I1 = ISOR(III,1)
      I2 = ISOR(III,2)
      J1 = ISOR(III,3)
      J2 = ISOR(III,4)
      K1 = ISOR(III,5)
      K2 = ISOR(III,6)
      DO 1300 I=I1,I2
      DO 1300 J=J1,J2
      DO 1300 K=K1,K2
      CC(I,J,K) = CC(I,J,K) + SOR(III)
 1300 CONTINUE
      END IF
      IF(KSTP.EQ.1) WRITE(6,3006)
      IF(KSTP.EQ.1) WRITE(10,3006)
 3006 FORMAT(1H ,"Checking Obstacles")
C**** OBSTACTLES CHECKED
C
      IF(NOBS.NE.0) THEN
      DO 300 III=1,NOBS
      I1 = IOB(III,1)
      I2 = IOB(III,2)
      J1 = IOB(III,3)
      J2 = IOB(III,4)
      K1 = IOB(III,5)
      K2 = IOB(III,6)
      DO 300 I=I1,I2
      DO 300 J=J1,J2
      DO 300 K=K1,K2
      CC(I,J,K) = 0.0
      QQ(I,J,K) = 0.0
      EE(I,J,K) = 0.0
  300 CONTINUE
      END IF
C
C RESTRAINTED BOUNDARY CONDITIONS CHECKED
C
      IF(KSTP.EQ.1) WRITE(6,3007)
      IF(KSTP.EQ.1) WRITE(10,3007)
 3007 FORMAT(1H ,"Checking Boundary Conditions")
      IF(NIN.NE.0) THEN
      DO 310 III=1,NIN
      I1 = IIN(III,2)
      I2 = IIN(III,3)
```

```
      J1 = IIN(III,4)
      J2 = IIN(III,5)
      K1 = IIN(III,6)
      K2 = IIN(III,7)
      IF (IIN(III,1).EQ.10) THEN
      DO 320 J=J1,J2
      DO 320 K=K1,K2
      EE(I1,J,K) = 0.0
  320 CONTINUE
      END IF
      IF(IIN(III,1) .EQ.11) THEN
      DO 321 I=I1,I2
      DO 321 K=K1,K2
      EE(I,J1,K) = 0.0
  321 CONTINUE
      END IF
      IF (IIN(III,1) .EQ.12) THEN
      DO 322 I=I1,I2
      DO 322 J=J1,J2
      EE(I,J,K1) = 0.0
  322 CONTINUE
      END IF
  310 CONTINUE
      END IF
C
C*****SAVE TEMPERATURE IN S FOR BOUYANCY FORCE
C
      DO 1110 I=1,L
      DO 1110 J=1,M
      DO 1110 K=1,N
      S(I,J,K) = C(I,J,K)
 1110 CONTINUE
C
C*****UPDATE OF Q,E,C
C
      DO 1120 I=1,L
      DO 1120 J=1,M
      DO 1120 K=1,N
      Q(I,J,K) = Q(I,J,K) + DT * QQ(I,J,K)
      E(I,J,K) = E(I,J,K) + DT * EE(I,J,K)
      C(I,J,K) = C(I,J,K) + DT * CC(I,J,K)
 1120 CONTINUE
C
      IF(KSTP.EQ.1) WRITE(6,3008)
      IF(KSTP.EQ.1) WRITE(10,3008)
 3008 FORMAT(1H ,"Updated Q, E & C")
C*****CHECK FOR POSITIVE VALUES OF Q & E
C
      DO 1130 I=1,L
      DO 1130 J=1,M
      DO 1130 K=1,N
      IF(Q(I,J,K).LE.0.0) THEN
      Q(I,J,K) = 0.0
      E(I,J,K) = 0.0
      END IF
C
      IF(E(I,J,K).LE.0.0) THEN
      Q(I,J,K) = 0.0
      E(I,J,K) = 0.0
      END IF
 1130 CONTINUE
C
C CALCULATE AVERAGE CHANGES OF Q, E, & C
C
      QAVE = 0.0
      EAVE = 0.0
      CAVE = 0.0
C
      DO 110 K=3,N-1
      DO 110 J=3,M-1
      DO 110 I=3,L-1
      QAVE = QAVE + QQ(I,J,K)**2
      EAVE = EAVE + EE(I,J,K)**2
      CAVE = CAVE + CC(I,J,K)**2
C
  110 CONTINUE
C
C SOLVE MOMENTUM EQUATIONS
      IF(KSTP.EQ.1) WRITE(6,3009)
      IF(KSTP.EQ.1) WRITE(10,3009)
 3009 FORMAT(1H ,"Averages of Q, E & C calculated")
C
      CALL MOMENTX
      IF(KSTP.EQ.1) WRITE(6,3010)
      IF(KSTP.EQ.1) WRITE(10,3010)
 3010 FORMAT(1H ,"Momentum in X direction solved")
      CALL MOMENTY
      IF(KSTP.EQ.1) WRITE(6,3011)
      IF(KSTP.EQ.1) WRITE(10,3011)
 3011 FORMAT(1H ,"Momentum in Y direction solved")
      CALL MOMENTZ
      IF(KSTP.EQ.1) WRITE(6,3012)
      IF(KSTP.EQ.1) WRITE(10,3012)
 3012 FORMAT(1H ,"Momentum in Z direction solved")
C
      DO 1140 I=1,L
      DO 1140 J=1,M
      DO 1140 K=1,N
      UU(I,J,K)=UU(I,J,K)*SX(I,J,K)
      VV(I,J,K)=VV(I,J,K)*SY(I,J,K)
      WW(I,J,K)=WW(I,J,K)*SZ(I,J,K)
C
      IF(BE(I,J,K).EQ.0.0) THEN
C
      UU(I,J,K) = 0.0
      VV(I,J,K) = 0.0
      WW(I,J,K) = 0.0
C
      END IF
C
C UPDATE U, V & W
C
      U (I,J,K) = U(I,J,K) + DT * UU(I,J,K)
      V (I,J,K) = V(I,J,K) + DT * VV(I,J,K)
      W (I,J,K) = W(I,J,K) + DT * WW(I,J,K)
 1140 CONTINUE
C
      IF(KSTP.EQ.1) WRITE(6,3013)
      IF(KSTP.EQ.1) WRITE(10,3013)
 3013 FORMAT(1H ,"Velocities updated")
C*****PRESSURE RELAXATION***************************
C
      NSTP = 0
C
   82 CONTINUE
C
      NSTP = NSTP + 1
      DMAX = 0.0
C
C*****CALCULATION OF DIVERGENCE
C
      DO 85 K=3,N-1
      DO 85 J=3,M-1
      DO 85 I=3,L-1
C
      S (I,J,K) = ( U(I,J,K) - U(I-1,J,K) ) * HX(I)
     1          + ( V(I,J,K) - V(I,J-1,K) ) * HY(J)
     2          + ( W(I,J,K) - W(I,J,K-1) ) * HZ(K)
C
C*****CORRECTION OF U,V & P
C
      DP = BE(I,J,K) * S (I,J,K)
      P (I,J,K) = P (I,J,K) + DP
C
     U (I  ,J,K) = U (I  ,J,K) + DT * DP * XD(I  ) * SX(I
,J,K)
       U (I-1,J,K) = U (I-1,J,K) - DT * DP * XD(I-1) *
SX(I-1,J,K)
       V (I,  J,K) = V (I,  J,K) + DT * DP * YD(J  ) *
SY(I,  J,K)
       V (I,J-1,K) = V (I,J-1,K) - DT * DP * YD(J-1) *
SY(I,J-1,K)
       W (I,J,K  ) = W (I,J,K  ) + DT * DP * ZD(K  ) *
SZ(I,J,K  )
       W (I,J,K-1) = W (I,J,K-1) - DT * DP * ZD(K-1) *
SZ(I,J,K-1)
C
   85 CONTINUE
C
      DO 1150 I=1,L
      DO 1150 J=1,M
      DO 1150 K=1,N
      IF(BE(I,J,K).EQ.0.0) S(I,J,K)=0.0
 1150 CONTINUE
C
C*****DETERMINE MAXIMUM DIVERGENCE
C
      DO 88 K=3,N-1
      DO 88 J=3,M-1
      DO 88 I=3,L-1
C
      IF( ABS(S(I,J,K)).GT.DMAX ) DMAX = ABS( S(I,J,K) )
C
   88 CONTINUE
C
C*****CHECK FOR CONVERGENCE OR MAX ITERATIONS OF PRESSURE
RELAXATION
C
C*****DIVERGENCE TORELANCE LIMIT
C
      IF ( DMAX .LE. EPS ) GO TO 90
C
      IF ( NSTP .LE. NSTM ) GO TO 82
C
   90 CONTINUE
C
C*****SET BOUNDARY CONDITION
C
      CALL BCONSUB
C
C*****CALCULATE  AVERAGE CHANGES
C
      DAVE = 0.0
```

```
      PAVE = 0.0
      ENAV = 0.0
C
      DO 100 K=3,N-1
      DO 100 J=3,M-1
      DO 100 I=3,L-1
C
      DAVE = DAVE + UU(I,J,K)**2*.5 + VV(I,J,K)**2*.5 +
WW(I,J,K)**2*.5
      PAVE = PAVE + P (I,J,K)
      ENAV = ENAV + ( ( U(I-1,J,K) + U(I,J,K) ) ** 2
     1             +   ( V(I,J-1,K) + V(I,J,K) ) ** 2
     2             +   ( W(I,J,K-1) + W(I,J,K) ) ** 2 ) *
0.125
C
  100 CONTINUE
C
C     QAVE = QAVE * VOLM
C     EAVE = EAVE * VOLM
C     CAVE = CAVE * VOLM
C     DAVE = DAVE * VOLM
      QAVE = SQRT( QAVE * VOLM )
      EAVE = SQRT( EAVE * VOLM )
      CAVE = SQRT( CAVE * VOLM )
      DAVE = SQRT( DAVE * VOLM )
      PAVE = PAVE * VOLM
      ENAV = ENAV * VOLM
C
C
    IF ( KSTP.EQ.1 .OR. MOD(KSTP,20) .EQ. 0 ) WRITE(6,205)
     IF ( KSTP.EQ.IKSTP .OR. KSTP .EQ. (IKSTP + 20) .OR.
     1MOD(KSTP,200) .EQ. 0 )  WRITE(10,205)
      IF (MOD(KSTP,100) .EQ. 0) THEN
      OPEN (21,FILE='UNIT21',FORM='UNFORMATTED')
      WRITE(21) U,V,W,P,Q,E,C,DX,DY,DZ,KSTP
      CLOSE(21)
      END IF
C
C
  205 FORMAT(1H /  '.KSTP.NSTP.KTIM'
     1            ,'....QAVE.......EAVE.....CAVE'
    2            ,'.....DAVE.....PAVE.....ENAV.....DMAX')
C
      KTIM = ISECND(1)-JTIME1
          WRITE(6,206)  KSTP,NSTP,KTIM,QAVE,EAVE,CAVE,DA-
VE,PAVE,ENAV,DMAX
     IF(KSTP .LE. (IKSTP + 19) .OR. MOD(KSTP,10) .EQ. 0)
THEN
         WRITE(10,206)  KSTP,NSTP,KTIM,QAVE,EAVE,CAVE,DA-
VE,PAVE,ENAV,DMAX
      ENDIF
  206 FORMAT(I5,I3,I6,F9.5,F11.5,5F9.5)
      IF ( ENAV .GT. ENMX ) GO TO 2000
C
      GO TO 200
C
 2000 CONTINUE
C
      RETURN
      END


      SUBROUTINE SOLVEX
C
      PARAMETER (L=29,M=27,N=20)
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE   ,HRE  ,PE   ,HPE  ,PR   ,AR   ,BETA,EPS ,DT   ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD  ,ZC1  ,ZC2  ,ZC3  ,ZSQ  ,ZSE  ,ZSC  ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)     ,DY(M+1)     ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      DIMENSION QQ(L,M,N),EE(L,M,N),CC(L,M,N)
C
      EQUIVALENCE (UU(1,1,1),QQ(1,1,1))
C
      EQUIVALENCE (VV(1,1,1),EE(1,1,1))
C
      EQUIVALENCE (WW(1,1,1),CC(1,1,1))
C
```

```
C*****SCALAR VARIABLE UPDATE*************************
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
 1000 CONTINUE
C
C*****EDDY DIFFUSIVITY INTERPOLATION
C
      DO 40 K=2,N-1
      DO 40 J=2,M-1
      DO 40 I=2,L-1
C
       DFX(I,J,K) = D(I,J,K) * XIP(I) + D(I+1,J,K) * (
1.0-XIP(I) )
C
   40 CONTINUE
C
C*****CALCULATION OF QQ
C
      DO 45 K=1,N
      DO 45 J=1,M
      DO 45 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSQ + HRE ) * XD(I)
C
   45 CONTINUE
C
      DO 1010 I=1,L
      DO 1010 J=1,M
      DO 1010 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * U(I,J,K)
C
    IF(AFX(I,J,K).LE.(-U(I,J,K))) AFX(I,J,K) = -U(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1010 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 50 K=2,N-1
      DO 50 J=2,M-1
      DO 50 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( Q(I,J,K) - Q(I+1,J,K)
)
C
   50 CONTINUE
C
      DO 1020 I=1,L
      DO 1020 J=1,M
      DO 1020 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+Q(I,J,K)*U(I,J,K)
 1020 CONTINUE
C
C
C*****SET QQ
C
      DO 55 K=3,N-1
      DO 55 J=3,M-1
      DO 55 I=3,L-1
      QQ(I,J,K) = QQ(I,J,K) +( AFX(I-1,J,K) - AFX(I,J,K)
) * HX(I)
   55 CONTINUE
C
      DO 1030 I=1,L
      DO 1030 J=1,M
      DO 1030 K=1,N
      AFX(I,J,K) = 0.0
 1030 CONTINUE
C
C***** CALCULATE EE
C
      DO 145 K=1,N
      DO 145 J=1,M
      DO 145 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSE + HRE ) * XD(I)
C
  145 CONTINUE
C
      DO 1040 I=1,L
      DO 1040 J=1,M
      DO 1040 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * U(I,J,K)
C
    IF(AFX(I,J,K).LE.(-U(I,J,K))) AFX(I,J,K) = -U(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1040 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 150 K=2,N-1
      DO 150 J=2,M-1
      DO 150 I=2,L-1
```

```
C
      AFX(I,J,K) = AFX(I,J,K) * ( E(I,J,K) - E(I+1,J,K)
)
C
  150 CONTINUE
C
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+E(I,J,K)*U(I,J,K)
 1050 CONTINUE
C
C*****SET EE
C
      DO 155 K=3,N-1
      DO 155 J=3,M-1
      DO 155 I=3,L-1
C
     EE(I,J,K) = EE(I,J,K) + ( AFX(I-1,J,K) - AFX(I,J,K)
) * HX(I)
C
  155 CONTINUE
C
C CALCULATION OF CC
C
      DO 245 K=1,N
      DO 245 J=1,M
      DO 245 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSC + HPE ) * XD(I)
C
  245 CONTINUE
C
      DO 1060 I=1,L
      DO 1060 J=1,M
      DO 1060 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * U(I,J,K)
C
   IF(AFX(I,J,K).LE.(-U(I,J,K))) AFX(I,J,K) = -U(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1060 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 250 K=2,N-1
      DO 250 J=2,M-1
      DO 250 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( C(I,J,K) - C(I+1,J,K)
)
C
  250 CONTINUE
C
      DO 1070 I=1,L
      DO 1070 J=1,M
      DO 1070 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+C(I,J,K)*U(I,J,K)
 1070 CONTINUE
C
C*****SET CC
C
      DO 255 K=3,N-1
      DO 255 J=3,M-1
      DO 255 I=3,L-1
C
     CC(I,J,K) = CC(I,J,K) + ( AFX(I-1,J,K) - AFX(I,J,K)
) * HX(I)
C
  255 CONTINUE
C
      RETURN
      END


      SUBROUTINE SOLVEY
C
      PARAMETER (L=29,M=27,N=20)
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      DIMENSION QQ(L,M,N),EE(L,M,N),CC(L,M,N)
C
      EQUIVALENCE (UU(1,1,1),QQ(1,1,1))
C
      EQUIVALENCE (VV(1,1,1),EE(1,1,1))
C
      EQUIVALENCE (WW(1,1,1),CC(1,1,1))
C
C*****INITIALIZE FLUX ARRAY********************
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
 1000 CONTINUE
C
C*****EDDY DIFFUSIVITY INTERPOLATION
C
      DO 40 K=2,N-1
      DO 40 J=2,M-1
      DO 40 I=2,L-1
C
       DFX(I,J,K) = D(I,J,K) * YIP(J) + D(I,J+1,K) * (
1.0-YIP(J) )
C
   40 CONTINUE
C
C*****CALCULATION OF QQ
C
    . DO 45 K=1,N
      DO 45 J=1,M
      DO 45 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSQ + HRE ) * YD(J)
C
   45 CONTINUE
C
      DO 1010 I=1,L
      DO 1010 J=1,M
      DO 1010 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * V(I,J,K)
C
   IF(AFX(I,J,K).LE.(-V(I,J,K))) AFX(I,J,K) = -V(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1010 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 50 K=2,N-1
      DO 50 J=2,M-1
      DO 50 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( Q(I,J,K) - Q(I,J+1,K)
)
C
   50 CONTINUE
C
      DO 1020 I=1,L
      DO 1020 J=1,M
      DO 1020 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+Q(I,J,K)*V(I,J,K)
 1020 CONTINUE
C
C
C*****SET QQ
C
      DO 55 K=3,N-1
      DO 55 J=3,M-1
      DO 55 I=3,L-1
     QQ(I,J,K) = QQ(I,J,K) +( AFX(I,J-1,K) - AFX(I,J,K)
) * HY(J)
   55 CONTINUE
C
      DO 1030 I=1,L
      DO 1030 J=1,M
      DO 1030 K=1,N
      AFX(I,J,K) = 0.0
 1030 CONTINUE
C
C***** CALCULATE EE
C
      DO 145 K=1,N
      DO 145 J=1,M
      DO 145 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSE + HRE ) * YD(J)
C
  145 CONTINUE
C
      DO 1040 I=1,L
      DO 1040 J=1,M
```

```
      DO 1040 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * V(I,J,K)
C
   IF(AFX(I,J,K).LE.(-V(I,J,K))) AFX(I,J,K) = -V(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1040 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 150 K=2,N-1
      DO 150 J=2,M-1
      DO 150 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( E(I,J,K) - E(I,J+1,K)
)
C
  150 CONTINUE
C
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+E(I,J,K)*V(I,J,K)
 1050 CONTINUE
C
C*****SET EE
C
      DO 155 K=3,N-1
      DO 155 J=3,M-1
      DO 155 I=3,L-1
C
     EE(I,J,K) = EE(I,J,K) + ( AFX(I,J-1,K) - AFX(I,J,K)
) * HY(J)
C
  155 CONTINUE
C
C CALCULATION OF CC
C
      DO 245 K=1,N
      DO 245 J=1,M
      DO 245 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSC + HPE ) * YD(J)
C
  245 CONTINUE
C
      DO 1060 I=1,L
      DO 1060 J=1,M
      DO 1060 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * V(I,J,K)
C
   IF(AFX(I,J,K).LE.(-V(I,J,K))) AFX(I,J,K) = -V(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1060 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 250 K=2,N-1
      DO 250 J=2,M-1
      DO 250 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( C(I,J,K) - C(I,J+1,K)
)
C
  250 CONTINUE
C
      DO 1070 I=1,L
      DO 1070 J=1,M
      DO 1070 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+C(I,J,K)*V(I,J,K)
 1070 CONTINUE
C
C*****SET CC
C
      DO 255 K=3,N-1
      DO 255 J=3,M-1
      DO 255 I=3,L-1
C
     CC(I,J,K) = CC(I,J,K) + ( AFX(I,J-1,K) - AFX(I,J,K)
) * HY(J)
C
  255 CONTINUE
C
      RETURN
      END


      SUBROUTINE SOLVEZ
C
      PARAMETER (L=29,M=27,N=20)
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE   ,HRE ,PE   ,HPE ,PR   ,AR   ,BETA,EPS ,DT   ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
```

```
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
   2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      DIMENSION QQ(L,M,N),EE(L,M,N),CC(L,M,N)
C
      EQUIVALENCE (UU(1,1,1),QQ(1,1,1))
C
      EQUIVALENCE (VV(1,1,1),EE(1,1,1))
C
      EQUIVALENCE (WW(1,1,1),CC(1,1,1))
C
C*****INITIALIZE FLUX ARRAY**********************
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
 1000 CONTINUE
C
C*****EDDY DIFFUSIVITY INTERPOLATION
C
      DO 40 K=2,N-1
      DO 40 J=2,M-1
      DO 40 I=2,L-1
C
       DFX(I,J,K) = D(I,J,K) * ZIP(K) + D(I,J,K+1) * (
1.0-ZIP(K) )
C
   40 CONTINUE
C
C*****CALCULATION OF QQ
C
      DO 45 K=1,N
      DO 45 J=1,M
      DO 45 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSQ + HRE ) * ZD(K)
C
   45 CONTINUE
C
      DO 1010 I=1,L
      DO 1010 J=1,M
      DO 1010 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * W(I,J,K)
C
   IF(AFX(I,J,K).LE.(-W(I,J,K))) AFX(I,J,K) = -W(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1010 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 50 K=2,N-1
      DO 50 J=2,M-1
      DO 50 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( Q(I,J,K) - Q(I,J,K+1)
)
C
   50 CONTINUE
C
      DO 1020 I=1,L
      DO 1020 J=1,M
      DO 1020 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+Q(I,J,K)*W(I,J,K)
 1020 CONTINUE
C
C
C*****SET QQ
C
      DO 55 K=3,N-1
      DO 55 J=3,M-1
      DO 55 I=3,L-1
C
     QQ(I,J,K) = QQ(I,J,K) +( AFX(I,J,K-1) - AFX(I,J,K)
) * HZ(K)
```

```
   55 CONTINUE
C
      DO 1030 I=1,L
      DO 1030 J=1,M
      DO 1030 K=1,N
      AFX(I,J,K) = 0.0
 1030 CONTINUE
C
C***** CALCULATE EE
C
      DO 145 K=1,N
      DO 145 J=1,M
      DO 145 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSE + HRE ) * ZD(K)
C
  145 CONTINUE
C
      DO 1040 I=1,L
      DO 1040 J=1,M
      DO 1040 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * W(I,J,K)
C
   IF(AFX(I,J,K).LE.(-W(I,J,K))) AFX(I,J,K) = -W(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1040 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 150 K=2,N-1
      DO 150 J=2,M-1
      DO 150 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( E(I,J,K) - E(I,J,K+1)
)
C
  150 CONTINUE
C
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+E(I,J,K)*W(I,J,K)
 1050 CONTINUE
C
C*****SET EE
C
      DO 155 K=3,N-1
      DO 155 J=3,M-1
      DO 155 I=3,L-1
C
     EE(I,J,K) = EE(I,J,K) + ( AFX(I,J,K-1) - AFX(I,J,K)
) * HZ(K)
C
  155 CONTINUE
C
C CALCULATION OF CC
C
      DO 245 K=1,N
      DO 245 J=1,M
      DO 245 I=1,L
C
      AFX(I,J,K) = ( DFX(I,J,K) * HZSC + HPE ) * ZD(K)
C
  245 CONTINUE
C
      DO 1060 I=1,L
      DO 1060 J=1,M
      DO 1060 K=1,N
      AFX(I,J,K) = AFX(I,J,K) - 0.5 * W(I,J,K)
C
   IF(AFX(I,J,K).LE.(-W(I,J,K))) AFX(I,J,K) = -W(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K)=0.0
 1060 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 250 K=2,N-1
      DO 250 J=2,M-1
      DO 250 I=2,L-1
C
      AFX(I,J,K) = AFX(I,J,K) * ( C(I,J,K) - C(I,J,K+1)
)
C
  250 CONTINUE
C
      DO 1070 I=1,L
      DO 1070 J=1,M
      DO 1070 K=1,N
      AFX(I,J,K) = AFX(I,J,K)+C(I,J,K)*W(I,J,K)
 1070 CONTINUE
C
C*****SET CC
C
      DO 255 K=3,N-1
      DO 255 J=3,M-1
      DO 255 I=3,L-1
C
     CC(I,J,K) = CC(I,J,K) + ( AFX(I,J,K-1) - AFX(I,J,K)
) * HZ(K)
C
  255 CONTINUE
C
      RETURN
      END


C
      SUBROUTINE MOMENTX
C
      PARAMETER (L=29,M=27,N=20)
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
     DIMENSION AFY(L,M,N),AFZ(L,M,N),XY(L,M,N),ZX(L,M,N)
     1,VU(L,M,N),WU(L,M,N)
C
      EQUIVALENCE (AFX(1,1,1),AFY(1,1,1),AFZ(1,1,1))
C
      EQUIVALENCE (DFX(1,1,1),XY(1,1,1),ZX(1,1,1))
C
      EQUIVALENCE (S(1,1,1),VU(1,1,1),WU(1,1,1))
C
C*****MOMENTUM VARIABLE UPDATE*********************
C
C*****INITIALIZATION OF WORK ARRAYS ON X FACES
C
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
      UU(I,J,K) = 0.0
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
 1000 CONTINUE
C
      DO 60 K=2,N
      DO 60 J=2,M
      DO 60 I=2,L
C
      UU(I,J,K) = (U(I,J,K) + U(I-1,J,K)) * 0.5
C
      AFX(I,J,K) = ( D(I,J,K) + HRE ) * HX(I)
C
   60 CONTINUE
C
      DO 1010 I=1,L
      DO 1010 J=1,M
      DO 1010 K=1,N
      AFX(I,J,K)=AFX(I,J,K)-UU(I,J,K)/CMR(I,J,K)
C
      IF(AFX(I,J,K).LE.(-2.0*UU(I,J,K)/CMR(I,J,K)))
     1      AFX(I,J,K) =   -2.0*UU(I,J,K)/CMR(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K) = 0.0
 1010 CONTINUE
C
      DO 65 K=2,N
      DO 65 J=2,M
      DO 65 I=2,L
C
      DU = U(I-1,J,K) - U(I,J,K)
      GRADU = -DU*HX(I)
       AFX(I,J,K)=UU(I,J,K)*(UU(I,J,K)+DU/CMR(I,J,K)) +
AFX(I,J,K)*DU
     2  - GRADU*( HRE + D(I,J,K))
C
   65 CONTINUE
C
      DO 1020 I=1,L
      DO 1020 J=1,M
      DO 1020 K=1,N
      UU(I,J,K) = 0.0
 1020 CONTINUE
```

```
C
      DO 66 K=2,N-1
      DO 66 J=2,M-1
      DO 66 I=2,L-1
      DU = (AFX(I,J,K) - AFX(I+1,J,K)) * XD(I)
     1      -AR * (S(I,J,K) * XIP(I) + S(I+1,J,K) * (1.0
- XIP(I)))
       UU(I,J,K) = DU +(P(I,J,K) - P(I+1,J,K))*XD(I)
   66 CONTINUE
C
      DO 1030 I=1,L
      DO 1030 J=1,M
      DO 1030 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
      S(I,J,K) = 0.0
 1030 CONTINUE
C
C*****LATERAL FLUX ON Y FACE
C
      DO 70 K=1,N-1
      DO 70 J=1,M-1
      DO 70 I=1,L-1
C
      R1 = D(I,J,K) * YIP(J) + D(I,J+1,K) * (1.0-YIP(J))
    R2 = D(I+1,J,K) * YIP(J) + D(I+1,J+1,K) * (1.0-YIP(J))
C
      XY(I,J,K) = R1 * XIP(I) + R2 * (1.0-XIP(I)) + HRE
C
      VU(I,J,K) = (V(I,J,K) + V(I+1,J,K)) * 0.5
      AFY(I,J,K) = XY(I,J,K) * YD(J)
C
   70 CONTINUE
C
      DO 1040 I=1,L
      DO 1040 J=1,M
      DO 1040 K=1,N
      AFY(I,J,K)=AFY(I,J,K)-VU(I,J,K)/CMR(I,J,K)
C
      IF(AFY(I,J,K).LE.(-2.0*VU(I,J,K)/CMR(I,J,K)))
     1      AFY(I,J,K)  =  -2.0*VU(I,J,K)/CMR(I,J,K)
C
      IF(AFY(I,J,K).LE.0.0) AFY(I,J,K) = 0.0
 1040 CONTINUE
C
      DO 75 K=1,N-1
      DO 75 J=1,M-1
      DO 75 I=1,L-1
C
      UV = (U(I,J,K) + U(I,J+1,K)) * 0.5
      GRADV = (V(I+1,J,K) - V(I,J,K))*XD(I)
      DU = U(I,J,K) - U(I,J+1,K)
      AFY(I,J,K)=VU(I,J,K)*(UV+DU/CMR(I,J,K))
     1 + AFY(I,J,K)*DU - GRADV*XY(I,J,K)
C
   75 CONTINUE
C
      DO 80 K=2,N-1
      DO 80 J=2,M-1
      DO 80 I=2,L-1
C
      DU = ( AFY(I,J-1,K) - AFY(I,J,K) ) * HY(J)
C
      UU(I,J,K) = UU(I,J,K) + DU
C
   80 CONTINUE
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
      S(I,J,K) = 0.0
 1050 CONTINUE
C
C*****LATERAL FLUX ON Z FACE
C
      DO 90 K=1,N-1
      DO 90 J=1,M-1
      DO 90 I=1,L-1
C
      R1 = D(I,J,K) * XIP(I) + D(I+1,J,K) * (1.0-XIP(I))
    R2 = D(I,J,K+1) * XIP(I) + D(I+1,J,K+1) * (1.0-XIP(I))
C
      ZX(I,J,K) = R1 * ZIP(K) + R2 * (1.0-ZIP(K)) + HRE
C
      WU(I,J,K) = (W(I,J,K) + W(I+1,J,K)) * 0.5
      AFZ(I,J,K) = ZX(I,J,K) * ZD(K)
C
   90 CONTINUE
C
      DO 1060 I=1,L
      DO 1060 J=1,M
      DO 1060 K=1,N
      AFZ(I,J,K)=AFZ(I,J,K)-WU(I,J,K)/CMR(I,J,K)
C
      IF(AFZ(I,J,K).LE.(-2.0*WU(I,J,K)/CMR(I,J,K)))
     1      AFZ(I,J,K)  =  -2.0*WU(I,J,K)/CMR(I,J,K)
C
      IF(AFZ(I,J,K).LE.0.0) AFZ(I,J,K) = 0.0
 1060 CONTINUE
C
      DO 95 K=1,N-1
      DO 95 J=1,M-1
      DO 95 I=1,L-1
C
      UW = (U(I,J,K) + U(I,J,K+1)) * 0.5
      GRADW = (W(I+1,J,K) - W(I,J,K))*XD(I)
      DU = U(I,J,K) - U(I,J,K+1)
      AFZ(I,J,K)=WU(I,J,K)*(UW+DU/CMR(I,J,K))
     1 + AFZ(I,J,K)*DU - GRADW*ZX(I,J,K)
C
   95 CONTINUE
C
C
      DO 100 K=2,N-1
      DO 100 J=2,M-1
      DO 100 I=2,L-1
C
C
      DU = ( AFZ(I,J,K-1) - AFZ(I,J,K) ) * HZ(K)
C
      UU(I,J,K) = UU(I,J,K) + DU
C
  100 CONTINUE
C
      RETURN
      END

C
      SUBROUTINE MOMENTY
C
      PARAMETER (L=29,M=27,N=20)
C
C       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE   ,HRE ,PE   ,HPE ,PR   ,AR   ,BETA,EPS ,DT   ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
   2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
    DIMENSION AFY(L,M,N),AFZ(L,M,N),XY(L,M,N),YZ(L,M,N)
     1,UV(L,M,N),WV(L,M,N)
C
      EQUIVALENCE (AFX(1,1,1),AFY(1,1,1),AFZ(1,1,1))
C
      EQUIVALENCE(DFX(1,1,1),XY(1,1,1),YZ(1,1,1))
C
      EQUIVALENCE(S(1,1,1),UV(1,1,1),WV(1,1,1))
C
C*****Y MOMENTUM VARIABLE UPDATE********************
C
C*****INITIALIZATION OF WORK ARRAYS FOR Y FACES
C
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
      VV(I,J,K) = 0.0
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
 1000 CONTINUE
C
      DO 60 K=2,N
      DO 60 J=2,M
      DO 60 I=2,L
C
      VV(I,J,K) = (V(I,J,K) + V(I,J-1,K)) * 0.5
C
      AFY(I,J,K) = ( D(I,J,K) + HRE ) * HY(J)
C
   60 CONTINUE
C
      DO 1010 I=1,L
      DO 1010 J=1,M
      DO 1010 K=1,N
      AFY(I,J,K)=AFY(I,J,K)-VV(I,J,K)/CMR(I,J,K)
C
      IF(AFY(I,J,K).LE.(-2.0*VV(I,J,K)/CMR(I,J,K)))
```

```
     1       AFY(I,J,K) =   -2.0*VV(I,J,K)/CMR(I,J,K)
C
      IF(AFY(I,J,K).LE.0.0) AFY(I,J,K) = 0.0
 1010 CONTINUE
C
      DO 65 K=2,N
      DO 65 J=2,M
      DO 65 I=2,L
C
      DV = V(I,J-1,K) - V(I,J,K)
      GRADV = -DV*HY(J)
       AFY(I,J,K)=VV(I,J,K)*(VV(I,J,K)+DV/CMR(I,J,K)) +
AFX(I,J,K)*DV
     2  - GRADV*( HRE + D(I,J,K))
C
   65 CONTINUE
C
      DO 1020 I=1,L
      DO 1020 J=1,M
      DO 1020 K=1,N
      VV(I,J,K) = 0.0
 1020 CONTINUE
C
      DO 66 K=2,N-1
      DO 66 J=2,M-1
      DO 66 I=2,L-1
      DV = (AFY(I,J,K) - AFY(I,J+1,K)) * YD(J)
      VV(I,J,K) = DV +(P(I,J,K) - P(I,J+1,K))*YD(J)
   66 CONTINUE
C
      DO 1030 I=1,L
      DO 1030 J=1,M
      DO 1030 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
      S(I,J,K) = 0.0
 1030 CONTINUE
C
C*****LATERAL FLUX ON X FACE
C
      DO 70 K=1,N-1
      DO 70 J=1,M-1
      DO 70 I=1,L-1
C
     R1 = D(I,J,K) * YIP(J) + D(I,J+1,K) * (1.0-YIP(J))
    R2 = D(I+1,J,K) * YIP(J) + D(I+1,J+1,K) * (1.0-YIP(J))
C
      XY(I,J,K) = R1 * XIP(I) + R2 * (1.0-XIP(I)) + HRE
C
      UV(I,J,K) = (U(I,J,K) + U(I,J+1,K)) * 0.5
      AFX(I,J,K) = XY(I,J,K) * XD(I)
C
   70 CONTINUE
C
      DO 1040 I=1,L
      DO 1040 J=1,M
      DO 1040 K=1,N
      AFX(I,J,K)=AFX(I,J,K)-UV(I,J,K)/CMR(I,J,K)
C
      IF(AFX(I,J,K).LE.(-2.0*UV(I,J,K)/CMR(I,J,K)))
     1       AFX(I,J,K)  =   -2.0*UV(I,J,K)/CMR(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K) = 0.0
 1040 CONTINUE
C
      DO 75 K=1,N-1
      DO 75 J=1,M-1
      DO 75 I=1,L-1
C
      VU = (V(I,J,K) + V(I+1,J,K)) * 0.5
      GRADU = (U(I,J+1,K) - U(I,J,K))*YD(J)
      DV = V(I,J,K) - V(I+1,J,K)
      AFX(I,J,K)=UV(I,J,K)*(VU+DV/CMR(I,J,K))
     1 + AFX(I,J,K)*DV - GRADU*XY(I,J,K)
C
   75 CONTINUE
C
      DO 80 K=2,N-1
      DO 80 J=2,M-1
      DO 80 I=2,L-1
C
      DV = ( AFX(I-1,J,K) - AFX(I,J,K) ) * HX(I)
C
      VV(I,J,K) = VV(I,J,K) + DV
C
   80 CONTINUE
C
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
      S(I,J,K) = 0.0
 1050 CONTINUE
C
C*****LATERAL FLUX ON Z FACE
C
      DO 90 K=1,N-1
      DO 90 J=1,M-1
      DO 90 I=1,L-1
C
     R1 = D(I,J,K) * ZIP(K) + D(I,J,K+1) * (1.0-ZIP(K))
    R2 = D(I,J+1,K) * ZIP(K) + D(I,J+1,K+1) * (1.0-ZIP(K))
C
      YZ(I,J,K) = R1 * YIP(J) + R2 * (1.0-YIP(J)) + HRE
C
      WV(I,J,K) = (W(I,J,K) + W(I,J+1,K)) * 0.5
      AFZ(I,J,K) = YZ(I,J,K) * ZD(K)
C
   90 CONTINUE
C
      DO 1060 I=1,L
      DO 1060 J=1,M
      DO 1060 K=1,N
      AFZ(I,J,K)=AFZ(I,J,K)-WV(I,J,K)/CMR(I,J,K)
C
      IF(AFZ(I,J,K).LE.(-2.0*WV(I,J,K)/CMR(I,J,K)))
     1       AFZ(I,J,K)  =   -2.0*WV(I,J,K)/CMR(I,J,K)
C
      IF(AFZ(I,J,K).LE.0.0) AFZ(I,J,K) = 0.0
 1060 CONTINUE
C
      DO 95 K=1,N-1
      DO 95 J=1,M-1
      DO 95 I=1,L-1
C
      VW = (V(I,J,K) + V(I,J,K+1)) * 0.5
      GRADW = (W(I,J+1,K) - W(I,J,K))*YD(J)
      DV = V(I,J,K) - V(I,J,K+1)
      AFZ(I,J,K)=WV(I,J,K)*(VW+DV/CMR(I,J,K))
     1 + AFZ(I,J,K)*DV - GRADW*YZ(I,J,K)
C
   95 CONTINUE
C
C

      DO 100 K=2,N-1
      DO 100 J=2,M-1
      DO 100 I=2,L-1
C
C
      DV = ( AFZ(I,J,K-1) - AFZ(I,J,K) ) * HZ(K)
C
      VV(I,J,K) = VV(I,J,K) + DV
C
  100 CONTINUE
C
      RETURN
      END


C
      SUBROUTINE MOMENTZ
C
      PARAMETER (L=29,M=27,N=20)
C
C       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
   2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
    DIMENSION AFY(L,M,N),AFZ(L,M,N),YZ(L,M,N),ZX(L,M,N)
     1,VW(L,M,N),UW(L,M,N)
C
      EQUIVALENCE (AFX(1,1,1),AFY(1,1,1),AFZ(1,1,1))
C
      EQUIVALENCE(DFX(1,1,1),YZ(1,1,1),ZX(1,1,1))
C
      EQUIVALENCE(S(1,1,1),VW(1,1,1),UW(1,1,1))
C
C*****MOMENTUM VARIABLE UPDATE************************
C
C*****INITIALIZATION OF WORK ARRAYS FOR Z FACE
C
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
```

```
      WW(I,J,K) = 0.0
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
 1000 CONTINUE
C
      DO 60 K=2,N
      DO 60 J=2,M
      DO 60 I=2,L
C
      WW(I,J,K) = (W(I,J,K) + W(I,J,K-1)) * 0.5
C
      AFZ(I,J,K) = ( D(I,J,K) + HRE ) * HZ(K)
C
   60 CONTINUE
C
      DO 1010 I=1,L
      DO 1010 J=1,M
      DO 1010 K=1,N
      AFZ(I,J,K)=AFZ(I,J,K)-WW(I,J,K)/CMR(I,J,K)
C
      IF(AFZ(I,J,K).LE.(-2.0*WW(I,J,K)/CMR(I,J,K)))
     1       AFZ(I,J,K) =    -2.0*WW(I,J,K)/CMR(I,J,K)
C
      IF(AFZ(I,J,K).LE.0.0) AFZ(I,J,K) = 0.0
 1010 CONTINUE
C
      DO 65 K=2,N
      DO 65 J=2,M
      DO 65 I=2,L
C
      DW = W(I,J,K-1) - W(I,J,K)
      GRADW = -DW*HZ(K)
        AFZ(I,J,K)=WW(I,J,K)*(WW(I,J,K)+DW/CMR(I,J,K)) +
AFX(I,J,K)*DW
     2  - GRADW*( HRE + D(I,J,K))
C
   65 CONTINUE
C
      DO 1020 I=1,L
      DO 1020 J=1,M
      DO 1020 K=1,N
      WW(I,J,K) = 0.0
 1020 CONTINUE
C
      DO 66 K=2,N-1
      DO 66 J=2,M-1
      DO 66 I=2,L-1
      DW = (AFZ(I,J,K) - AFZ(I,J,K+1)) * ZD(K)
      WW(I,J,K) = DW +(P(I,J,K) - P(I,J,K+1))*ZD(K)
   66 CONTINUE
C
      DO 1030 I=1,L
      DO 1030 J=1,M
      DO 1030 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
      S(I,J,K) = 0.0
 1030 CONTINUE
C
C*****LATERAL FLUX ON Y FACE
C
      DO 70 K=1,N-1
      DO 70 J=1,M-1
      DO 70 I=1,L-1
C
      R1 = D(I,J,K) * ZIP(K) + D(I,J,K+1) * (1.0-ZIP(K))
    R2 = D(I,J+1,K) * ZIP(K) + D(I,J+1,K+1) * (1.0-ZIP(K))
C
      YZ(I,J,K) = R1 * YIP(J) + R2 * (1.0-YIP(J)) + HRE
C
      VW(I,J,K) = (V(I,J,K) + V(I,J,K+1)) * 0.5
      AFY(I,J,K) = YZ(I,J,K) * YD(J)
C
   70 CONTINUE
C
C*****AMAX JUDGEMENT FOR CYBER 205
C
      DO 1040 I=1,L
      DO 1040 J=1,M
      DO 1040 K=1,N
      AFY(I,J,K)=AFY(I,J,K)-VW(I,J,K)/CMR(I,J,K)
C
      IF(AFY(I,J,K).LE.(-2.0*VW(I,J,K)/CMR(I,J,K)))
     1       AFY(I,J,K)   =   -2.0*VW(I,J,K)/CMR(I,J,K)
C
      IF(AFY(I,J,K).LE.0.0) AFY(I,J,K) = 0.0
 1040 CONTINUE
C
      DO 75 K=1,N-1
      DO 75 J=1,M-1
      DO 75 I=1,L-1
C
      WV = (W(I,J,K) + W(I,J+1,K)) * 0.5
      GRADV = (V(I,J,K+1) - V(I,J,K))*ZD(K)
      DW = W(I,J,K) - W(I,J+1,K)
      AFY(I,J,K)=VW(I,J,K)*(WV+DW/CMR(I,J,K))
     1 + AFY(I,J,K)*DW - GRADV*YZ(I,J,K)
C
   75 CONTINUE
C
      DO 80 K=2,N-1
      DO 80 J=2,M-1
      DO 80 I=2,L-1
C
      DW = ( AFY(I,J-1,K) - AFY(I,J,K) ) * HY(J)
C
      WW(I,J,K) = WW(I,J,K) + DW
C
   80 CONTINUE
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      AFX(I,J,K) = 0.0
      DFX(I,J,K) = 0.0
      S(I,J,K) = 0.0
 1050 CONTINUE
C
C*****LATERAL FLUX ON X FACE
C
      DO 90 K=1,N-1
      DO 90 J=1,M-1
      DO 90 I=1,L-1
C
      R1 = D(I,J,K) * XIP(I) + D(I+1,J,K) * (1.0-XIP(I))
    R2 = D(I,J,K+1) * XIP(I) + D(I+1,J,K+1) * (1.0-XIP(I))
C
      ZX(I,J,K) = R1 * ZIP(K) + R2 * (1.0-ZIP(K)) + HRE
C
      UW(I,J,K) = (U(I,J,K) + U(I,J,K+1)) * 0.5
      AFX(I,J,K) = ZX(I,J,K) * XD(I)
C
   90 CONTINUE
C
      DO 1060 I=1,L
      DO 1060 J=1,M
      DO 1060 K=1,N
      AFX(I,J,K)=AFX(I,J,K)-UW(I,J,K)/CMR(I,J,K)
C
      IF(AFX(I,J,K).LE.(-2.0*UW(I,J,K)/CMR(I,J,K)))
     1       AFX(I,J,K)   =   -2.0*UW(I,J,K)/CMR(I,J,K)
C
      IF(AFX(I,J,K).LE.0.0) AFX(I,J,K) = 0.0
 1060 CONTINUE
C
      DO 95 K=1,N-1
      DO 95 J=1,M-1
      DO 95 I=1,L-1
C
      WU = (W(I,J,K) + W(I+1,J,K)) * 0.5
      GRADU = (U(I,J,K+1) - U(I,J,K))*ZD(K)
      DW = W(I,J,K) - W(I+1,J,K)
      AFX(I,J,K)=UW(I,J,K)*(WU+DW/CMR(I,J,K))
     1 + AFZ(I,J,K)*DW - GRADU*ZX(I,J,K)
C
   95 CONTINUE
C
C
      DO 100 K=2,N-1
      DO 100 J=2,M-1
      DO 100 I=2,L-1
C
C
      DW = ( AFX(I-1,J,K) - AFX(I,J,K) ) * HX(I)
C
      WW(I,J,K) = WW(I,J,K) + DW
C
  100 CONTINUE
C
      RETURN
      END


C
      SUBROUTINE BCONSUB
C
C
      PARAMETER (L=29,M=27,N=20)
C
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE   ,HRE ,PE   ,HPE ,PR   ,AR   ,BETA,EPS ,DT   ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)     ,DY(M+1)     ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
```

```
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      COMMON /SCAL/ ICON(3)
C
C*****SET FREE SLIP WALL 1,2
C
      DO 10 K=2,N-1
      DO 10 J=2,M-1
C
      V (2,J,K) = V (3  ,J,K)
      V (L,J,K) = V (L-1,J,K)
C
      W (2,J,K) = W (3  ,J,K)
      W (L,J,K) = W (L-1,J,K)
C
      Q (2,J,K) = Q (3  ,J,K)
      Q (L,J,K) = Q (L-1,J,K)
C
      E (2,J,K) = E (3  ,J,K)
      E (L,J,K) = E (L-1,J,K)
C
      C (2,J,K) = C (3  ,J,K)
      C (L,J,K) = C (L-1,J,K)
C
      U (1,J,K) = U (2  ,J,K)
     1                        + ( V (2,J,K) - V (2,J-1,K) )
* HY(J) / HX(2)
     2                        + ( W (2,J,K) - W (2,J,K-1) )
* HZ(K) / HX(2)
      U (L,J,K) = U (L-1,J,K)
     1                        - ( V (L,J,K) - V (L,J-1,K) )
* HY(J) / HX(L)
     2                        - ( W (L,J,K) - W (L,J,K-1) )
* HZ(K) / HX(L)
C
   10 CONTINUE
C
C*****SET FREE SLIP WALL 3,4
C
      DO 20 K=2,N-1
      DO 20 I=2,L-1
C
      W (I,2,K) = W (I,  3,K)
      W (I,M,K) = W (I,M-1,K)
C
      U (I,2,K) = U (I,  3,K)
      U (I,M,K) = U (I,M-1,K)
C
      Q (I,2,K) = Q (I,  3,K)
      Q (I,M,K) = Q (I,M-1,K)
C
      E (I,2,K) = E (I,  3,K)
      E (I,M,K) = E (I,M-1,K)
C
      C (I,2,K) = C (I,  3,K)
      C (I,M,K) = C (I,M-1,K)
C
      V (I,1,K) = V (I,  2,K)
     1                        + ( W (I,2,K) - W (I,2,K-1) )
* HZ(K) / HY(2)
     2                        + ( U (I,2,K) - U (I-1,2,K) )
* HX(I) / HY(2)
      V (I,M,K) = V (I,M-1,K)
     1                        - ( W (I,M,K) - W (I,M,K-1) )
* HZ(K) / HY(M)
     2                        - ( U (I,M,K) - U (I-1,M,K) )
* HX(I) / HY(M)
C
   20 CONTINUE
C
C*****SET FREE SLIP WALL 5,6
C
      DO 30 J=2,M-1
      DO 30 I=2,L-1
C
      U (I,J,2) = U (I,J,  3)
      U (I,J,N) = U (I,J,N-1)
C
      V (I,J,2) = V (I,J,  3)
      V (I,J,N) = V (I,J,N-1)
C
      Q (I,J,2) = Q (I,J,  3)
      Q (I,J,N) = Q (I,J,N-1)
C
      E (I,J,2) = E (I,J,  3)
      E (I,J,N) = E (I,J,N-1)
C
      C (I,J,2) = C (I,J,  3)
      C (I,J,N) = C (I,J,N-1)
C
      W (I,J,1) = W (I,J,  2)
     1                        + ( U (I,J,2) - U (I-1,J,2) )
* HX(I) / HZ(2)
     2                        + ( V (I,J,2) - V (I,J-1,2) )
* HY(J) / HZ(2)
      W (I,J,N) = W (I,J,N-1)
     1                        - ( U (I,J,N) - U (I-1,J,N) )
* HX(I) / HZ(N)
     2                        - ( V (I,J,N) - V (I,J-1,N) )
* HY(J) / HZ(N)
C
   30 CONTINUE
C
C*****BOUNDARY CONDITION SET
C
      DO 100 III=1,NIN
C
      I1 = IIN(III,2)
      I2 = IIN(III,3)
      J1 = IIN(III,4)
      J2 = IIN(III,5)
      K1 = IIN(III,6)
      K2 = IIN(III,7)
C
      SEV = UIN(III)
C
                                                     GO        TO
(110,120,130,140,150,160,170,180,190,200,210,220
     1        ,230,240,250,260,270,280,290,300,310,320,33
0,340),IIN(III,1)
C
C*****U VALUE
C
  110 DO 111 K=K1,K2
      DO 111 J=J1,J2
      DO 111 I=I1,I2
  111 U (I,J,K) = SEV
      GO TO 100
C
C*****V VALUE
C
  120 DO 121 K=K1,K2
      DO 121 J=J1,J2
      DO 121 I=I1,I2
  121 V (I,J,K) = SEV
      GO TO 100
C
C*****W VALUE
C
  130 DO 131 K=K1,K2
      DO 131 J=J1,J2
      DO 131 I=I1,I2
  131 W (I,J,K) = SEV
      GO TO 100
C
C*****Q VALUE
C
  140 DO 141 K=K1,K2
      DO 141 J=J1,J2
      DO 141 I=I1,I2
  141 Q (I,J,K) = SEV
      GO TO 100
C
C*****E VALUE
C
  150 DO 151 K=K1,K2
      DO 151 J=J1,J2
      DO 151 I=I1,I2
  151 E (I,J,K) = SEV
      GO TO 100
C
C*****C VALUE
C
  160 DO 161 K=K1,K2
      DO 161 J=J1,J2
      DO 161 I=I1,I2
  161 C (I,J,K) = SEV
      GO TO 100
C
C*****P VALUE ( X DIRECTION )
C
  170 DO 171 K=K1,K2
      DO 171 J=J1,J2
  171 P (I1,J,K) = SEV
      GO TO 100
C
C*****P VALUE ( Y DIRECTION )
C
  180 DO 181 K=K1,K2
      DO 181 I=I1,I2
  181 P (I,J1,K) = SEV
      GO TO 100
C
C*****P VALUE ( Z DIRECTION )
C
  190 DO 191 J=J1,J2
      DO 191 I=I1,I2
  191 P (I,J,K1) = SEV
      GO TO 100
C
C*****SET WALL LAW
```

```
C
C*****X DIRECTION
C
  200 CONTINUE
C
      DO 201 K=K1,K2
      DO 201 J=J1,J2-1
  201 V (I2,J,K) = V (I1,J,K) * ( 1.0-2.0*SEV )
C
      DO 202 K=K1,K2-1
      DO 202 J=J1,J2
  202 W (I2,J,K) = W (I1,J,K) * ( 1.0-2.0*SEV )
C
      DD = DX(I1)
C
      DO 203 K=K1,K2
      DO 203 J=J1,J2
      E (I1,J,K) = ZCD**.75 * ABS(Q(I1,J,K))**1.5 * 2.0
/ ( 0.4*DD )
      E (I2,J,K) = E(I1,J,K)
  203 CONTINUE
C
      GO TO 100
C
C*****Y DIRECTION
C
  210 CONTINUE
C
      DO 211 K=K1,K2-1
      DO 211 I=I1,I2
  211 W (I,J2,K) = W (I,J1,K) * ( 1.0-2.0*SEV )
C
      DO 212 K=K1,K2
      DO 212 I=I1,I2-1
  212 U (I,J2,K) = U (I,J1,K) * ( 1.0-2.0*SEV )
C
      DD = DY(J1)
C
      DO 213 K=K1,K2
      DO 213 I=I1,I2
      E (I,J1,K) = ZCD**.75 * ABS(Q(I,J1,K))**1.5 * 2.0
/ ( 0.4*DD )
      E (I,J2,K) = E(I,J1,K)
  213 CONTINUE
C
      GO TO 100
C
C*****Z DIRECTION
C
  220 CONTINUE
C
      DO 221 J=J1,J2
      DO 221 I=I1,I2-1
  221 U (I,J,K2) = U (I,J,K1) * ( 1.0-2.0*SEV )
C
      DO 222 J=J1,J2-1
      DO 222 I=I1,I2
  222 V (I,J,K2) = V (I,J,K1) * ( 1.0-2.0*SEV )
C
      DD = DZ(K1)
C
      DO 223 J=J1,J2
      DO 223 I=I1,I2
      E (I,J,K1) = ZCD**.75 * ABS(Q(I,J,K1))**1.5 * 2.0
/ ( 0.4*DD )
      E (I,J,K2) = E(I,J,K1)
  223 CONTINUE
C
      GO TO 100
C
C*****HEAT FLUX SPECIFIED ( X DIRECTION )
C
  230 DO 231 K=K1,K2
      DO 231 J=J1,J2
C
      IF( E(I1,J,K) .LE. 0.0 )THEN
      RAM1 = HPE
      ELSE
      RAM1 = ZCD * HZSC * Q(I1,J,K)**2 / E(I1,J,K) + HPE
      END IF
      RAM  =    RAM1                  * HX(I1)
      C(I2,J,K) = SEV / RAM + C(I1,J,K)
C
  231 CONTINUE
C
      GO TO 100
C
C*****HEAT FLUX SPECIFIED ( Y DIRECTION )
C
  240 DO 241 K=K1,K2
      DO 241 I=I1,I2
C
      IF( E(I,J1,K) .LE. 0.0 )THEN
      RAM1 = HPE
      ELSE
      RAM1 = ZCD * HZSC * Q(I,J1,K)**2 / E(I,J1,K) + HPE
      END IF
      RAM  =    RAM1                  * HY(J1)
```

```
      C(I,J2,K) = SEV / RAM + C(I,J1,K)
C
  241 CONTINUE
C
      GO TO 100
C
C*****HEAT FLUX SPECIFIED ( Z DIRECTION )
C
  250 DO 251 J=J1,J2
      DO 251 I=I1,I2
C
      IF( E(I,J,K1) .LE. 0.0 )THEN
      RAM1 = HPE
      ELSE
      RAM1 = ZCD * HZSC * Q(I,J,K1)**2 / E(I,J,K1) + HPE
      END IF
      RAM  =    RAM1                  * HZ(K1)
      C(I,J,K2) = SEV / RAM + C(I,J,K1)
C
  251 CONTINUE
C
      GO TO 100
C
C*****HEAT TRANSFER COEFFICIENT SPECIFIED ( X DIRECTION
)
C
  260 DO 261 K=K1,K2
      DO 261 J=J1,J2
C
      IF( E(I1,J,K) .LE. 0.0 )THEN
      RAM1 = HPE
      ELSE
      RAM1 = ZCD * HZSC * Q(I1,J,K)**2 / E(I1,J,K) + HPE
      END IF
      RAM  =    RAM1                  * HX(I1)
      C(I2,J,K) = SEV / RAM * ( C(I2,J,K) - C(I1,J,K) )
+ C(I1,J,K)
C
  261 CONTINUE
C
      GO TO 100
C
C*****HEAT TRANSFER COEFFICIENT SPECIFIED ( Y DIRECTION
)
C
  270 DO 271 K=K1,K2
      DO 271 I=I1,I2
C
      IF( E(I,J1,K) .LE. 0.0 )THEN
      RAM1 = HPE
      ELSE
      RAM1 = ZCD * HZSC * Q(I,J1,K)**2 / E(I,J1,K) + HPE
      END IF
      RAM  =    RAM1                  * HY(J1)
      C(I,J2,K) = SEV / RAM * ( C(I,J2,K) - C(I,J1,K) )
+ C(I,J1,K)
C
  271 CONTINUE
C
      GO TO 100
C
C*****HEAT TRANSFER COEFFICIENT SPECIFIED ( Z DIRECTION
)
C
  280 DO 281 J=J1,J2
      DO 281 I=I1,I2
C
      IF( E(I,J,K1) .LE. 0.0 )THEN
      RAM1 = HPE
      ELSE
      RAM1 = ZCD * HZSC * Q(I,J,K1)**2 / E(I,J,K1) + HPE
      END IF
      RAM  =    RAM1                  * HY(J1)
      C(I,J,K2) = SEV / RAM * ( C(I,J,K2) - C(I,J,K1) )
+ C(I,J,K1)
C
  281 CONTINUE
C
      GO TO 100
C
C*****WALL TEMPERATURE SPECIFIED ( X DIRECTION )
C
  290 DO 291 K=K1,K2
      DO 291 J=J1,J2
C
  291 C(I2,J,K) = 2.0 * SEV - C(I1,J,K)
C
      GO TO 100
C
C*****WALL TEMPERATURE SPECIFIED ( Y DIRECTION )
C
  300 DO 301 K=K1,K2
      DO 301 I=I1,I2
C
  301 C(I,J2,K) = 2.0 * SEV - C(I,J1,K)
C
      GO TO 100
C
```

```
C*****WALL TEMPERATURE SPECIFIED ( Z DIRECTION )
C
  310 DO 311 J=J1,J2
      DO 311 I=I1,I2
C
  311 C(I,J,K2) = 2.0 * SEV - C(I,J,K1)
C
      GO TO 100
C
C*****Q SLIP VALUE ( X DIRECTION )
C
  320 DO 321 K=K1,K2
      DO 321 J=J1,J2
C
  321 Q(I2,J,K) = Q(I1,J,K) * ( 1.0-2.0*SEV )
C
      GO TO 100

C
C*****Q SLIP VALUE ( Y DIRECTION )
C
  330 DO 331 K=K1,K2
      DO 331 I=I1,I2
C
  331 Q(I,J2,K) = Q(I,J1,K) * ( 1.0-2.0*SEV )
C
      GO TO 100
C
C*****Q SLIP VALUE ( Z DIRECTION )
C
  340 DO 341 J=J1,J2
      DO 341 I=I1,I2
C
  341 Q(I,J,K2) = Q(I,J,K1) * ( 1.0-2.0*SEV )
C
      GO TO 100
C
C*****
C
  100 CONTINUE
C
      CALL BSET
C
      RETURN
C
      END


C
C DUMMY ROUTINE
C
      SUBROUTINE BSET
C
      RETURN
      END

C*****OUTPUT*******************************************
C
      SUBROUTINE OUTPUT
C
      PARAMETER (L=29,M=27,N=20)
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*4 THEMA,PLAN,TERM
C
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE  ,HRE ,PE  ,HPE ,PR  ,AR  ,BETA,EPS ,DT  ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZC1 ,ZC2 ,ZC3 ,ZSQ ,ZSE ,ZSC ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      COMMON /SCAL/ ICON(3)
C
      DIMENSION  AA(L,M,N)
      EQUIVALENCE (AFX(1,1,1),AA(1,1,1))
```

```
      NUM = 1
C
    1 CONTINUE
      GO TO ( 10,20,30,40,50,60,70),NUM
C
   10 CONTINUE
      DO 11 K=1,N
      DO 11 J=1,M
      DO 11 I=1,L
   11 AA(I,J,K) = U (I,J,K)
      THEMA = '   U'
      GO TO 200
C
   20 CONTINUE
      DO 21 K=1,N
      DO 21 J=1,M
      DO 21 I=1,L
   21 AA(I,J,K) = V (I,J,K)
      THEMA = '   V'
      GO TO 200
C
   30 CONTINUE
      DO 31 K=1,N
      DO 31 J=1,M
      DO 31 I=1,L
   31 AA(I,J,K) = W (I,J,K)
      THEMA = '   W'
      GO TO 200
C
   40 CONTINUE
      DO 41 K=1,N
      DO 41 J=1,M
      DO 41 I=1,L
   41 AA(I,J,K) = P (I,J,K)
      THEMA = '   P'
      GO TO 200
C
   50 CONTINUE
      DO 51 K=1,N
      DO 51 J=1,M
      DO 51 I=1,L
   51 AA(I,J,K) = Q (I,J,K)
      THEMA = '   Q'
      GO TO 200
C
   60 CONTINUE
      DO 61 K=1,N
      DO 61 J=1,M
      DO 61 I=1,L
   61 AA(I,J,K) = E (I,J,K)
      THEMA = '   E'
      GO TO 200
C
   70 CONTINUE
      DO 71 K=1,N
      DO 71 J=1,M
      DO 71 I=1,L
   71 AA(I,J,K) = C (I,J,K)
      THEMA = '   C'
      GO TO 200
C
C*****OUTPUT
C
  200 CONTINUE
C
C*****Y-Z PLANE
C
      PLAN = ' Y-Z'
      TERM = ' I ='
      IC = ICON(1)
      INDEX = 1
C
      CALL PRINT1 (THEMA,PLAN,TERM,IC,INDEX)
C
C*****Z-X PLANE
C
      PLAN = ' Z-X'
      TERM = ' J ='
      IC = ICON(2)
      INDEX = 2
C
      CALL PRINT1 (THEMA,PLAN,TERM,IC,INDEX)
C
C*****X-Y PLANE
C
      PLAN = ' X-Y'
      TERM = ' K ='
      IC = ICON(3)
      INDEX = 3
C
      CALL PRINT1 (THEMA,PLAN,TERM,IC,INDEX)
C
      NUM = NUM + 1
      IF ( NUM .LE. 7 ) GO TO 1
C
      RETURN
      END
```

```
C
C*****PRINT************************************
C
      SUBROUTINE PRINT1(THEMA,PLAN,TERM,IC,INDEX)
C
      PARAMETER (L=29,M=27,N=20)
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*4 THEMA,PLAN,TERM
C
C     INTEGER SX,SY,SZ
      COMMON /CONTR/
     1 TITLE(20)
     2,RE   ,HRE ,PE   ,HPE ,PR   ,AR   ,BETA,EPS ,DT   ,ENMX
     3,KSTM,NSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD  ,ZC1  ,ZC2  ,ZC3  ,ZSQ  ,ZSE  ,ZSC  ,HZSQ,HZSE,HZSC
     5,DA(10,60)
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1)
    2,CMR(L,M,N),NIN ,IIN(100,7),UIN(100),NOBS,IOB(100,6)
     3,NSOR,ISOR(100,6),SOR(100)
C
      COMMON
     2 U(L,M,N),V(L,M,N),W(L,M,N),UU(L,M,N),VV(L,M,N),W
W(L,M,N)
     3,P(L,M,N),D(L,M,N),S(L,M,N),AFX(L,M,N),DFX(L,M,N)
     5,Q(L,M,N),E(L,M,N),C(L,M,N)
     6,SX(L,M,N),SY(L,M,N),SZ(L,M,N),BE(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
      COMMON /SCAL/ ICON(3)
C
      DIMENSION  AA(L,M,N)
      EQUIVALENCE (AFX(1,1,1),AA(1,1,1))
C
C*****OUTPUT TITLE
C
      WRITE(10,600) TITLE
      WRITE(10,601) THEMA,PLAN,TERM,IC,KSTP
      GO TO ( 10,20,30 ),INDEX
C
C*****Y-Z PLANE
C
   10 CONTINUE
      KA = 1
  100 CONTINUE
      KK = 5 * ( KA-1 ) + 1
      KE = 5 * KA
C
      IF ( (KE-N) .GE. 0 ) GO TO 103
C
      WRITE(10,602) (K,K=KK,KE)
      WRITE(10,101) (J,(AA(IC,J,K),K=KK,KE),J=1,M)
      WRITE(10,102)
      KA = KA + 1
      GO TO 100
C
  103 CONTINUE
      WRITE(10,602) (K,K=KK,N)
      DO 104 J=1,M
      WRITE(10,101) J,(AA(IC,J,K),K=KK,N)
  104 CONTINUE
C
      RETURN
C
C*****Z-X PLANE
C
   20 CONTINUE
      IA = 1
  200 CONTINUE
      II = 5 * ( IA-1 ) + 1
      IE = 5 * IA
C
      IF ( (IE-L) .GE. 0 ) GO TO 203
C
      WRITE(10,602) (I,I=II,IE)
      WRITE(10,101) (K,(AA(I,IC,K),I=II,IE),K=1,N)
      WRITE(10,102)
      IA = IA + 1
      GO TO 200
C
  203 CONTINUE
      WRITE(10,602) (I,I=II,L)
      DO 204 K=1,N
      WRITE(10,101) K,(AA(I,IC,K),I=II,L)
  204 CONTINUE
C
      RETURN
C
C*****X-Y PLANE
C
   30 CONTINUE
      JA = 1
  300 CONTINUE
      JJ = 5 * ( JA-1 ) + 1
      JE = 5 * JA
C
      IF ( (JE-M) .GE. 0 ) GO TO 303
C
      WRITE(10,602) (J,J=JJ,JE)
      WRITE(10,101) (I,(AA(I,J,IC),J=JJ,JE),I=1,L)
      WRITE(10,102)
      JA = JA + 1
      GO TO 300
C
  303 CONTINUE
      WRITE(10,602) (J,J=JJ,M)
      DO 304 I=1,L
      WRITE(10,101) I,(AA(I,J,IC),J=JJ,M)
  304 CONTINUE
C
      RETURN
C
C*****FORMAT STATEMENT
C
  101 FORMAT (1H ,I5,5F12.7)
  102 FORMAT (1H ///          )
C
  600 FORMAT (1H ///1H ,20A4)
  601 FORMAT (1H ,A4,'-DISTRIBUTION ON ',A4,' PLAN',A4,I4
     1                              ,'    KSTP=',I10///)
  602 FORMAT (1H ,5I12)
C
      END
      FUNCTION ISECND (I)
      SAVE
      DIMENSION MDAYS(12)
                                                            DATA
MDAYS/0,31,59,90,120,151,181,212,243,273,303,334/
      IF (I.EQ.0) THEN
         CALL GETTIM (IHR,MIN,ISEC,I110)
         ISECND = 0.0
         RETURN
      ELSE
         CALL GETTIM (IH,MIN1,IS,I10)
         CALL GETDAT(IYEAR,IMONTH,IDAY)
             ISECND  =  IS+60*(MIN1+60*IH)+(IDAY+MDAYS(I-
MONTH))*24*3600
          IF(ISECND.LT.0.0) ISCOND = ISCOND + 24*60*60
          RETURN
      ENDIF
      END
```

A.6 Sampe Output

An sample output is given below for the problem pressented in Chapter 5. The physical meaning of the terms are :

KSTP	time iteration step number
NSTP	pressure iteration step number
KTIM	CPU time in seconds
QAVE	volumetric root mean square residual of k-equation
EAVE	volumetric root mean square residual of ε-equation
CAVE	volumetric root mean square residual of temperature-equation
DAVE	volumetric root mean square residual of momentum equation
PAVE	volumetric mean pressure
ENAV	volumetric mean energy
DMAX	maximum continuity error

Convergence is judged by monitoring the temporal variation of these parmaeters

After the convergent status listing, spatial variatin at the final time is given per the liting pointer.

On a UNIX or DOS computer, the flow field is save every 100 iterations to UNIT21. The problem can be terminated by a Cntrl Break sequence and restarted by copying UNIT21 to UNIT22. (Save UNIT22 before this copying if you have disk space to burn and need it for some reason).

```
AIR DISTRIBUTION PERFORMANCE FOR 4.0 CFM WITH 75 % FREE AREA DIFFUSER


   KSTM;.......TIME ITERATION MAX NO.=        1999000
   NSTM;...PRESSURE ITERATION MAX NO.=             20
   MAXT;.........CPU TIME MAX  (SEC.)=         288000
   INIC;....INITIAL FILE DATA CONTROL=              1
   RE  ;......LAMINAR REYNOLDS NUMBER= 257710.0000000
   PR  ;......LAMINAR PRANDTL  NUMBER=       .7100000
   AR  ;............ARCHIMEDES NUMBER=      -.0012350
   BETA;..ACCERATE RELAXIZATION CONST=      1.0000000
   EPS ;...DIVERGENCE TOLERANCE LIMIT=       .0010000
   DT  ;................TIME DIVISION=       .0100000
   ENMX;....CONVERGENCE JUDGING CONST=     10.0000000
   ZCD ;                                     .0900000
   ZC1 ;                                    1.4400000
   ZC2 ;                                    1.9200000
   ZC3 ;                                    1.0000000
   ZSQ ;                                    1.0000000
   ZSE ;                                    1.3000000
   ZSC ;                                     .9000000
 ****DX
  .063  .063  .063  .063  .063  .063  .063  .063  .063  .125
  .125  .298  .298  .298  .298  .298  .298  .172  .172  .172
  .172  .172  .172  .208  .208  .208  .185  .185  .185
 ****DY
  .167  .167  .167  .167  .250  .250  .417  .417  .500  .500
  .625  .625  .625  .625  .625  .625  .625  .625  .417  .417
```

```
 .417  .250  .250  .250  .167  .167  .167
****DZ
 .125  .125  .125  .125  .125  .125  .125  .125  .125  .188
 .188  .292  .292  .292  .292  .167  .167  .125  .125  .125
```

*****DATA OUTPUT POINT*****

(6, 6, 6)

*****BOUNDARY CONDITION*****

...NOB.	...I1.	...I2.	...J1.	...J2.	...K1.	...K2.	........SPV
1	4	8	2	2	3	6	.00000
2	5	8	2	2	3	6	1.00000
3	5	8	2	2	2	6	.00000
4	5	8	2	2	3	6	.21140
5	5	8	2	2	3	6	.02219
6	5	8	2	2	3	6	1.00000
8	19	22	2	3	3	7	.00000
1	18	22	2	2	3	7	.00000
2	19	22	2	2	3	7	-.29090
3	19	22	2	2	2	7	.00000
6	19	22	2	2	3	7	.00000
10	28	29	3	22	3	19	.14286
10	20	21	23	24	3	8	.14286
10	28	29	23	24	9	19	.14286
10	28	29	25	26	3	19	.14286
10	3	2	3	26	3	19	.14286
11	3	4	3	2	3	19	.14286
11	5	8	3	2	7	19	.14286
11	9	18	3	2	3	19	.14286
11	19	22	3	2	8	19	.14286
11	23	28	3	2	3	19	.14286
11	3	28	26	27	3	19	.14286
11	21	28	22	23	3	8	.14286
11	21	28	25	24	3	8	.14286
12	3	28	3	26	19	20	.14286
12	21	28	23	24	9	8	.14286
13	28	29	3	22	3	19	.00000
13	20	21	23	24	3	8	.00000
13	28	29	23	24	9	19	.00000
13	28	29	25	26	3	19	.00000
13	3	2	3	26	3	19	.00000
14	3	4	3	2	3	19	.00000
14	5	8	3	2	7	19	.00000
14	9	18	3	2	3	19	.00000
14	19	22	3	2	8	19	.00000
14	23	28	3	2	3	19	.00000
14	3	28	26	27	3	19	.00000
14	21	28	22	23	3	8	.00000
14	21	28	25	24	3	8	.00000
15	3	28	3	26	19	20	.00000
15	21	28	23	24	9	8	.00000

*****OBSTACLE CONDITION*****

....I1.	...I2.	...J1.	...J2.	...K1.	...K2
21	28	23	24	3	8

*****CRITICAL REYNOLDS NO.*****

....I1.	...I2.	...J1.	...J2.	...K1.	...K2.	........CMR
1	29	1	27	1	20	2.00000

```
   Starting Solution of Navier Stokes Equations

Set Initial Boundary Conditions
Initial Time is        22349639 seconds
Starting Solution Loop
The matrix S is formed
The diffusivity has been calulated
Finished Scalar X Fluxes
Finished Scalar Y Fluxes
Finished Scalar Z Fluxes
```

```
Checking Obstacles
Checking Boundary Conditions
Updated Q, E & C
Averages of Q, E & C calculated
Momentum in X direction solved
Momentum in Y direction solved
Momentum in Z direction solved
Velocities updated

.KSTP.NSTP.KTIM....QAVE.......EAVE.....CAVE.....DAVE.....PAVE.....ENAV.....DMAX
    1 21     17205.03020***********   .26084   .12598   .00529   .00064  1.70416
    2 21     33   .11876     .01939   .24591  1.86575   .01448   .00092   .37954
    3 21     50   .12725     .01946   .23696   .83422   .01417   .00108   .47056
    4 21     66   .14544     .02261   .22641   .48167   .00989   .00116   .43783
    5 21     83   .16436     .03035   .21528   .30238   .00527   .00117   .33681
    6 21    100   .17365     .03873   .20561   .21257   .00234   .00117   .23681
    7 21    116   .17402     .04529   .19629   .17335   .00140   .00115   .15429

SIMILAR OUTPUT FOR REST OF RUN

.KSTP.NSTP.KTIM....QAVE.......EAVE.....CAVE.....DAVE.....PAVE.....ENAV.....DMAX
95400  1     12   .00304     .00281   .00035   .00005   .00102   .01106   .00000
95401  1     22   .00040     .00025   .00035   .00007   .00102   .01106   .00065
95402  1     32   .00034     .00022   .00035   .00006   .00102   .01106   .00051
95403  1     42   .00029     .00019   .00035   .00005   .00102   .01106   .00030
95404  1     52   .00025     .00018   .00035   .00005   .00102   .01106   .00041
95405  1     62   .00022     .00016   .00035   .00005   .00102   .01106   .00026
95406  1     72   .00019     .00015   .00035   .00005   .00102   .01106   .00009
95407  1     82   .00017     .00014   .00035   .00005   .00102   .01106   .00007
95408  1     92   .00015     .00013   .00035   .00005   .00102   .01106   .00005
95409  1    102   .00014     .00012   .00035   .00005   .00102   .01106   .00005
95410  1    112   .00012     .00011   .00035   .00005   .00102   .01106   .00004
95411  1    122   .00011     .00010   .00035   .00005   .00102   .01106   .00004
95412  1    132   .00010     .00010   .00035   .00005   .00102   .01106   .00003
95413  1    142   .00009     .00009   .00035   .00005   .00102   .01106   .00003
95414  1    152   .00008     .00008   .00035   .00005   .00102   .01106   .00003
95415  1    162   .00008     .00008   .00035   .00005   .00102   .01106   .00003
95416  1    172   .00007     .00007   .00035   .00005   .00102   .01106   .00002
95417  1    182   .00006     .00007   .00035   .00005   .00102   .01106   .00002
95418  1    192   .00006     .00007   .00035   .00005   .00102   .01106   .00002
95419  1    202   .00005     .00006   .00035   .00005   .00102   .01106   .00002

.KSTP.NSTP.KTIM....QAVE.......EAVE.....CAVE.....DAVE.....PAVE.....ENAV.....DMAX
95420  1    212   .00005     .00006   .00035   .00005   .00102   .01106   .00002
95430  1    312   .00003     .00004   .00035   .00005   .00102   .01106   .00001

 AIR DISTRIBUTION PERFORMANCE FOR 4.0 CFM WITH 75 % FREE AREA DIFFUSER
    U-DISTRIBUTION ON  Y-Z PLAN I =   6     KSTP=      95436
```

	1	2	3	4
1	.0000000	.0000000	.0000000	.0000000
2	.0000000	-.0377082	.0000000	.0000000
3	.0000000	-.0377082	-.0377082	-.0350278
4	.0000000	-.0465862	-.0465862	-.0439187
5	.0000000	-.0248950	-.0248950	-.0241430
6	.0000000	-.0120372	-.0120372	-.0121700
7	.0000000	-.0042385	-.0042385	-.0045329
8	.0000000	-.0020490	-.0020490	-.0022703
9	.0000000	-.0016675	-.0016675	-.0017640
10	.0000000	-.0008836	-.0008836	-.0009005
11	.0000000	-.0008800	-.0008800	-.0007864
12	.0000000	-.0004160	-.0004160	-.0002405
13	.0000000	.0000699	.0000699	.0003121
14	.0000000	.0004613	.0004613	.0007533
15	.0000000	.0007440	.0007440	.0010655
16	.0000000	.0009741	.0009741	.0013044
17	.0000000	.0012447	.0012447	.0015627

18	.0000000	.0016338	.0016338	.0019175
19	.0000000	.0030029	.0030029	.0032152
20	.0000000	.0052749	.0052749	.0053874
21	.0000000	.0097016	.0097016	.0096809
22	.0000000	.0161370	.0161370	.0159706
23	.0000000	.0259591	.0259591	.0256499
24	.0000000	.0414892	.0414892	.0410592
25	.0000000	.0604879	.0604879	.0599995
26	.0000000	.0698387	.0698387	.0693406
27	.0000000	.0698387	.0498844	.0495286

	6	7	8	9
1	.0000000	.0000000	.0000000	.0000000
2	.0000000	-.0393853	-.0637311	-.0401753
3	-.0245346	-.0551399	-.0892242	-.0562459
4	-.0378026	-.0534110	-.0625486	-.0430626
5	-.0267209	-.0355210	-.0367339	-.0252890
6	-.0169687	-.0225887	-.0224592	-.0153645
7	-.0079478	-.0102730	-.0087487	-.0035122
8	-.0043679	-.0053353	-.0037557	.0003382
9	-.0026373	-.0026147	-.0008050	.0027980
10	-.0010775	-.0005971	.0012312	.0044513
11	-.0002110	.0007682	.0028125	.0059031
12	.0008196	.0020826	.0042299	.0071837
13	.0017258	.0031825	.0053962	.0082142
14	.0024030	.0039652	.0061684	.0087658
15	.0028276	.0043950	.0064729	.0087156
16	.0030626	.0045409	.0063854	.0081733
17	.0032064	.0045105	.0060361	.0073339
18	.0033374	.0043904	.0055335	.0063495
19	.0042511	.0049662	.0056770	.0060300
20	.0059023	.0061839	.0063725	.0062132
21	.0095149	.0092611	.0088313	.0081019
22	.0150420	.0141836	.0130242	.0115443
23	.0239632	.0224984	.0205484	.0181153
24	.0387190	.0367096	.0339944	.0307105
25	.0572293	.0545944	.0511900	.0471226
26	.0660153	.0631779	.0595933	.0553454
27	.0471534	.0451267	.0425663	.0395321

	11	12	13	14
1	.0000000	.0000000	.0000000	.0000000
2	-.0260435	-.0224750	-.0199974	-.0181848
3	-.0364612	-.0314653	-.0279966	-.0254590
4	-.0297660	-.0262754	-.0238537	-.0222110
5	-.0211116	-.0189681	-.0171322	-.0162099
6	-.0136742	-.0144790	-.0134715	-.0129946
7	.0010562	-.0057046	-.0072731	-.0078250
8	.0064331	-.0002699	-.0034730	-.0048783
9	.0094734	.0038978	-.0001369	-.0023128
10	.0109130	.0061625	.0021521	-.0004258
11	.0116471	.0073066	.0038951	.0012885
12	.0117415	.0073349	.0046673	.0023201
13	.0112110	.0066456	.0048718	.0035413
14	.0100956	.0057623	.0047162	.0042750
15	.0085848	.0047427	.0038126	.0029637
16	.0069381	.0036193	.0024581	.0005392
17	.0053817	.0025446	.0011415	-.0012153
18	.0041085	.0016686	-.0000410	-.0021425
19	.0035437	.0015516	-.0000116	-.0014717
20	.0033549	.0016696	.0005814	-.0002378
21	.0044061	.0023308	.0016724	.0015413
22	.0060022	.0035121	.0028110	.0030482
23	.0107872	.0073510	.0058224	.0055340
24	.0214509	.0167060	.0138116	.0125402
25	.0355210	.0291609	.0248333	.0220781
26	.0422746	.0341934	.0280216	.0236157
27	.0301959	.0244237	.0200153	.0168682

	16	17	18	19
1	.0000000	.0000000	.0000000	.0000000
2	-.0165876	-.0167902	-.0174928	-.0125987
3	-.0232228	-.0235064	-.0244901	-.0176383
4	-.0218904	-.0230909	-.0242375	-.0184035
5	-.0181522	-.0201583	-.0217025	-.0175901
6	-.0160853	-.0183273	-.0199082	-.0171033
7	-.0123243	-.0145300	-.0162169	-.0148544
8	-.0100705	-.0121004	-.0137682	-.0132023
9	-.0078026	-.0095516	-.0111212	-.0110079
10	-.0059177	-.0073731	-.0087869	-.0088463
11	-.0037183	-.0048389	-.0059967	-.0059878
12	-.0018536	-.0028023	-.0039828	-.0043651
13	.0042161	.0059216	.0085380	.0116575
14	.0053066	.0054078	.0070759	.0089459
15	-.0013491	.0001183	.0026906	.0052289
16	-.0021199	.0004753	.0035959	.0058337
17	-.0007674	.0021602	.0052376	.0067883
18	.0005566	.0037432	.0068490	.0077464
19	.0020534	.0053045	.0085868	.0089171
20	.0031630	.0065368	.0102890	.0103000
21	.0045949	.0081643	.0126005	.0123141
22	.0069406	.0110353	.0160834	.0153452
23	.0087439	.0135132	.0188705	.0181980
24	.0141438	.0177914	.0220195	.0213177
25	.0218979	.0237792	.0254927	.0241146
26	.0221965	.0234468	.0246224	.0244043
27	.0158545	.0167476	.0175873	.0174315

AIR DISTRIBUTION PERFORMANCE FOR 4.0 CFM WITH 75 % FREE AREA DIFFUSER
U-DISTRIBUTION ON Z-X PLAN J = 6 KSTP= 95436

	1	2	3	4
1	.0000000	.0000000	.0000000	.0000000
2	.0000000	.0000000	-.0143519	-.0189411
3	.0143518	.0000000	-.0143519	-.0189411
4	.0139656	.0000000	-.0139657	-.0185999
5	.0141733	.0000000	-.0141734	-.0191325
6	.0169746	.0000000	-.0169747	-.0226847
7	.0172213	.0000000	-.0172213	-.0246470
8	.0095349	.0000000	-.0095349	-.0167507
9	.0006130	.0000000	-.0006130	-.0053793
10	-.0013917	.0000000	.0013917	-.0002593
11	.0033576	.0000000	-.0033576	-.0067365
12	.0038058	.0000000	-.0038058	-.0074607
13	.0035006	.0000000	-.0035006	-.0069066
14	.0033552	.0000000	-.0033552	-.0066313
15	.0036080	.0000000	-.0036080	-.0070847
16	.0046765	.0000000	-.0046765	-.0087982
17	.0056995	.0000000	-.0056995	-.0104180
18	.0073451	.0000000	-.0073451	-.0118308
19	.0020610	.0000000	-.0020610	-.0071204
20	.0000000	.0000000	-.0014721	-.0050859

	6	7	8	9
1	.0000000	.0000000	.0000000	.0000000
2	-.0120372	-.0047630	.0023117	.0051375
3	-.0120372	-.0047630	.0023117	.0051375
4	-.0121700	-.0052184	.0015532	.0041970
5	-.0133087	-.0067412	-.0003866	.0019947
6	-.0169687	-.0105542	-.0046391	-.0027094
7	-.0225887	-.0179371	-.0141078	-.0137871
8	-.0224592	-.0223625	-.0222611	-.0239644
9	-.0153645	-.0191911	-.0227698	-.0269146
10	-.0086966	-.0143135	-.0202919	-.0261743
11	-.0136742	-.0172778	-.0209608	-.0245591
12	-.0144790	-.0178476	-.0210509	-.0240471
13	-.0134715	-.0166370	-.0196965	-.0226228

14	-.0129946	-.0160980	-.0191283	-.0220636
15	-.0137529	-.0169653	-.0200852	-.0230994
16	-.0160853	-.0194225	-.0226109	-.0256702
17	-.0183273	-.0218525	-.0251715	-.0283179
18	-.0199082	-.0236401	-.0271297	-.0303517
19	-.0171033	-.0215231	-.0255427	-.0291806
20	-.0122165	-.0153735	-.0182447	-.0208431

	11	12	13	14
1	.0000000	.0000000	.0000000	.0000000
2	-.0430232	-.0676554	-.0620071	-.0582557
3	-.0430232	-.0676554	-.0620071	-.0582557
4	-.0430970	-.0662431	-.0612462	-.0575953
5	-.0436360	-.0636396	-.0598391	-.0564032
6	-.0452980	-.0604394	-.0579815	-.0549010
7	-.0478742	-.0570141	-.0558599	-.0532921
8	-.0493298	-.0534254	-.0536418	-.0517405
9	-.0462343	-.0502044	-.0515229	-.0503724
10	-.0406483	-.0467014	-.0491549	-.0489710
11	-.0359153	-.0436505	-.0471309	-.0480229
12	-.0337168	-.0414247	-.0456781	-.0477307
13	-.0326763	-.0407759	-.0456367	-.0485812
14	-.0324829	-.0413402	-.0468901	-.0503614
15	-.0337677	-.0430844	-.0495197	-.0535081
16	-.0363246	-.0458313	-.0525078	-.0568586
17	-.0389054	-.0487159	-.0556288	-.0601535
18	-.0411802	-.0513815	-.0586059	-.0633899
19	-.0412297	-.0521806	-.0599983	-.0653032
20	-.0294495	-.0372716	-.0428556	-.0466448

	16	17	18	19
1	.0000000	.0000000	.0000000	.0000000
2	-.0493617	-.0468528	-.0506302	-.0586866
3	-.0493617	-.0468528	-.0506302	-.0586866
4	-.0483445	-.0452245	-.0483374	-.0555754
5	-.0467976	-.0429649	-.0452591	-.0514649
6	-.0452741	-.0410919	-.0426844	-.0479619
7	-.0441806	-.0400958	-.0412677	-.0456113
8	-.0436704	-.0400289	-.0409938	-.0444634
9	-.0437303	-.0407694	-.0415434	-.0442163
10	-.0443841	-.0423755	-.0429024	-.0446254
11	-.0458282	-.0447681	-.0450112	-.0457849
12	-.0480768	-.0477011	-.0475900	-.0474768
13	-.0506147	-.0504006	-.0498635	-.0490515
14	-.0530226	-.0525912	-.0516393	-.0502672
15	-.0561365	-.0552556	-.0538500	-.0519352
16	-.0598234	-.0586303	-.0568874	-.0545585
17	-.0633443	-.0622321	-.0603653	-.0577506
18	-.0670155	-.0660723	-.0643082	-.0617878
19	-.0699569	-.0695920	-.0682784	-.0661509
20	-.0499688	-.0497082	-.0487699	-.0472503

	21	22	23	24
1	.0000000	.0000000	.0000000	.0000000
2	-.0767805	-.0807695	-.0783228	-.0692598
3	-.0767805	-.0807695	-.0783228	-.0692598
4	-.0723167	-.0758911	-.0732261	-.0641063
5	-.0665526	-.0697794	-.0671630	-.0584657
6	-.0612425	-.0641043	-.0616782	-.0536591
7	-.0568445	-.0591846	-.0569485	-.0496656
8	-.0533945	-.0550453	-.0529241	-.0463195
9	-.0508372	-.0517373	-.0495989	-.0435272
10	-.0486759	-.0486998	-.0464117	-.0407859
11	-.0471100	-.0462381	-.0436657	-.0382760
12	-.0461025	-.0443560	-.0414491	-.0361796
13	-.0457402	-.0431012	-.0397690	-.0344305

14	-.0457102	-.0424360	-.0385978	-.0330392
15	-.0462220	-.0423204	-.0378543	-.0318723
16	-.0478530	-.0434069	-.0383007	-.0315731
17	-.0501886	-.0452078	-.0394632	-.0319304
18	-.0539036	-.0484036	-.0419872	-.0334129
19	-.0590533	-.0537877	-.0470343	-.0371121
20	-.0421806	-.0384195	-.0335957	-.0265084

	26	27	28	29
1	.0000000	.0000000	.0000000	.0000000
2	-.0389216	-.0211550	.0000000	.0161963
3	-.0389216	-.0211550	.0000000	.0211550
4	-.0344700	-.0179067	.0000000	.0179067
5	-.0307137	-.0156498	.0000000	.0156498
6	-.0280947	-.0143614	.0000000	.0143614
7	-.0261493	-.0134493	.0000000	.0134493
8	-.0245984	-.0127067	.0000000	.0127067
9	-.0233050	-.0120862	.0000000	.0120862
10	-.0220860	-.0115610	.0000000	.0115610
11	-.0208452	-.0109524	.0000000	.0109524
12	-.0199020	-.0105461	.0000000	.0105461
13	-.0189083	-.0101347	.0000000	.0101347
14	-.0178994	-.0096108	.0000000	.0096108
15	-.0167708	-.0089472	.0000000	.0089472
16	-.0158262	-.0079965	.0000000	.0079965
17	-.0155819	-.0078870	.0000000	.0078870
18	-.0159512	-.0081197	.0000000	.0081197
19	-.0170482	-.0091029	.0000000	.0091029
20	-.0121772	-.0065020	.0000000	.0000000

AIR DISTRIBUTION PERFORMANCE FOR 4.0 CFM WITH 75 % FREE AREA DIFFUSER
U-DISTRIBUTION ON X-Y PLAN K = 6 KSTP= 95436

	1	2	3	4
1	.0000000	.2623130	.0413388	.0682832
2	.0000000	.0000000	.0000000	.0000000
3	.0000000	-.0295275	-.0413389	-.0682832
4	.0000000	.0000000	-.1863700	-.0923136
5	.0000000	.0000000	-.0893838	-.0675452
6	.0000000	.0000000	-.0245346	-.0378026
7	.0000000	.0000000	.0365729	-.0086253
8	.0000000	.0000000	.1219318	.0152867
9	.0000000	-.0351019	-.0491430	-.0044251
10	.0000000	-.0904742	-.1266649	-.0768795
11	.0000000	-.0800154	-.1120225	-.0926885
12	.0000000	-.0637690	-.0892773	-.0814231
13	.0000000	-.0569287	-.0797009	-.0737525
14	.0000000	-.0517023	-.0723838	-.0666816
15	.0000000	-.0451473	-.0632068	-.0573831
16	.0000000	-.0330946	-.0463328	-.0420997
17	.0000000	-.0011071	-.0015500	-.0154425
18	.0000000	.0000000	.0884159	-.0023577
19	.0000000	.0000000	-.0051592	-.0279787
20	.0000000	.0000000	-.0634966	-.0609481
21	.0000000	.0000000	-.1231590	-.0950417
22	.0000000	.0000000	-.2201474	-.1250677
23	.0000000	-.0999543	-.1399371	-.1090776
24	.0000000	-.0674200	-.0943888	-.0839412
25	.0000000	-.0471658	-.0660326	-.0612670
26	.0000000	-.0300123	-.0420175	-.0399037
27	.0000000	-.0149410	-.0209175	-.0203981
28	.0000000	.0000000	.0000000	.0000000
29	.0000000	-.0030302	.0209175	.0203981

	6	7	8	9
1	.0169746	.0062654	.0034033	.0018757
2	.0000000	.0000000	.0000000	.0000000
3	-.0169747	-.0062655	-.0034033	-.0018757
4	-.0226847	-.0088192	-.0048181	-.0027221
5	-.0216966	-.0091154	-.0050037	-.0029045
6	-.0169687	-.0079478	-.0043679	-.0026373
7	-.0105542	-.0060373	-.0033300	-.0021535
8	-.0046391	-.0042321	-.0023977	-.0017228
9	-.0027094	-.0036557	-.0021552	-.0016235
10	-.0188261	-.0090256	-.0050731	-.0030244
11	-.0452980	-.0212590	-.0122917	-.0067926
12	-.0604394	-.0479988	-.0325932	-.0199239
13	-.0579815	-.0484130	-.0382537	-.0299496
14	-.0549010	-.0479031	-.0396694	-.0324758
15	-.0506275	-.0466791	-.0405060	-.0340801
16	-.0452741	-.0454214	-.0411546	-.0354510
17	-.0410919	-.0451599	-.0418553	-.0364824
18	-.0426844	-.0464080	-.0423121	-.0368234
19	-.0479619	-.0484474	-.0427712	-.0369257
20	-.0550212	-.0505755	-.0428940	-.0366279
21	-.0612425	-.0518395	-.0423601	-.0357740
22	-.0641043	-.0513618	-.0408655	-.0342133
23	-.0616782	-.0485998	-.0381776	-.0318219
24	-.0536591	-.0425034	-.0332716	-.0276421
25	-.0420129	-.0336599	-.0264224	-.0219452
26	-.0280947	-.0226892	-.0178863	-.0148572
27	-.0143614	-.0116528	-.0092759	-.0077782
28	.0000000	.0000000	.0000000	.0000000
29	.0143614	.0116528	.0092759	.0077782

	11	12	13	14
1	.0005358	.0001328	-.0001890	-.0004303
2	.0000000	.0000000	.0000000	.0000000
3	-.0005358	-.0001328	.0001890	.0004303
4	-.0006596	.0000330	.0006053	.0010341
5	-.0005185	.0003770	.0011407	.0017123
6	-.0002110	.0008196	.0017258	.0024030
7	.0001620	.0012859	.0023010	.0030570
8	.0004986	.0017060	.0028151	.0036361
9	.0007073	.0020204	.0032276	.0041118
10	.0005619	.0022320	.0036889	.0047176
11	-.0005798	.0016971	.0035362	.0047891
12	-.0061647	-.0017631	.0013698	.0033456
13	-.0132152	-.0066573	-.0020812	.0006973
14	-.0185706	-.0114144	-.0057099	-.0023842
15	-.0214553	-.0147495	-.0091768	-.0058515
16	-.0234342	-.0173847	-.0123001	-.0094541
17	-.0250438	-.0195999	-.0154380	-.0131359
18	-.0257588	-.0207091	-.0170536	-.0151127
19	-.0262188	-.0216071	-.0184264	-.0167961
20	-.0263573	-.0222080	-.0194502	-.0180155
21	-.0260775	-.0224065	-.0200201	-.0187112
22	-.0252682	-.0220928	-.0200493	-.0188541
23	-.0238060	-.0211240	-.0194022	-.0183308
24	-.0209676	-.0188650	-.0174932	-.0165722
25	-.0168522	-.0152915	-.0142214	-.0134619
26	-.0114865	-.0104264	-.0096257	-.0090030
27	-.0059559	-.0053345	-.0048330	-.0044243
28	.0000000	.0000000	.0000000	.0000000
29	.0059559	.0053345	.0048330	.0044243

	16	17	18	19
1	-.0007094	-.0007936	-.0008691	-.0011439
2	.0000000	.0000000	.0000000	.0000000
3	.0007094	.0007936	.0008691	.0011439
4	.0015009	.0016258	.0017307	.0022378
5	.0023007	.0024423	.0025618	.0032796

6	.0030626	.0032064	.0033374	.0042511
7	.0037539	.0038927	.0040395	.0051371
8	.0043530	.0044852	.0046566	.0059271
9	.0048466	.0049752	.0051824	.0066144
10	.0055051	.0056418	.0059619	.0077023
11	.0057255	.0059043	.0063787	.0083698
12	.0047960	.0051502	.0061737	.0088280
13	.0026867	.0031820	.0046870	.0078003
14	-.0001864	.0003219	.0021556	.0055847
15	-.0037302	-.0032715	-.0012088	.0025256
16	-.0077625	-.0072525	-.0050203	-.0007740
17	-.0117589	-.0110287	-.0088254	-.0035618
18	-.0135814	-.0126120	-.0108199	-.0048595
19	-.0149137	-.0135688	-.0124143	-.0059112
20	-.0156920	-.0139415	-.0133302	-.0069690
21	-.0160745	-.0139083	-.0121489	-.0043359
22	-.0160672	-.0135093	-.0108362	-.0029365
23	-.0156388	-.0130527	-.0097324	-.0021038
24	-.0143619	-.0122953	-.0089689	-.0017612
25	-.0118398	-.0105478	-.0081094	-.0024978
26	-.0078844	-.0071213	-.0054106	-.0005081
27	-.0038974	-.0036419	-.0028785	-.0003211
28	.0000000	.0000000	.0000000	.0000000
29	.0038974	.0036419	.0028785	.0003211

	21	22	23	24
1	-.0025895	-.0040887	-.0065256	-.0107627
2	.0000000	.0000000	.0000000	.0000000
3	.0025895	.0040887	.0065256	.0107627
4	.0050083	.0079448	.0127086	.0208133
5	.0073153	.0115873	.0185025	.0301114
6	.0095149	.0150420	.0239632	.0387190
7	.0115972	.0183004	.0290885	.0467969
8	.0135495	.0213436	.0338561	.0543211
9	.0153580	.0241436	.0382224	.0612376
10	.0186195	.0292075	.0460637	.0737017
11	.0211797	.0330090	.0519222	.0840029
12	.0263722	.0415949	.0652931	.1033870
13	.0288792	.0462282	.0722531	.1163251
14	.0289096	.0473097	.0753003	.1230175
15	.0268660	.0460967	.0762747	.1239657
16	.0238451	.0446482	.0742648	.1194536
17	.0233297	.0422119	.0661820	.1055225
18	.0250556	.0410944	.0569059	.0876882
19	.0306541	.0457375	.0282287	.0643205
20	.0354897	.0519016	.0000000	.0000000
21	.0327376	.0285513	.0203936	.0650249
22	.0264564	.0144236	.0103025	.0518695
23	.0188709	.0071698	.0051213	.0402276
24	.0108533	.0044634	.0031881	.0268177
25	.0070673	.0046229	.0033021	.0143454
26	.0071276	.0047500	.0033928	.0040101
27	.0051603	.0031460	.0022471	-.0009860
28	.0000000	.0000000	.0000000	.0000000
29	-.0051603	-.0031460	.0023519	-.0111387

	26	27
1	-.0125194	.0000000
2	.0000000	.0000000
3	.0125194	.0089423
4	.0294811	.0210578
5	.0477565	.0341115
6	.0660153	.0471534
7	.0837382	.0598125
8	.1008150	.0720101
9	.1174010	.0838572
10	.1459392	.1042414
11	.1729971	.1235683

12	.2045285	.1460906
13	.2198451	.1570310
14	.2256579	.1611829
15	.2249453	.1606739
16	.2184400	.1560273
17	.2061886	.1472764
18	.1975775	.1411257
19	.1865053	.1332170
20	.1866410	.1333140
21	.1763016	.1259287
22	.1614625	.1153294
23	.1438662	.1027608
24	.1201745	.0858382
25	.0940292	.0671632
26	.0648474	.0463192
27	.0346386	.0247417
28	.0000000	.0000000
29	-.0346387	.0000000

SIMILAR OUTPUT FOR V, W, P, TEMPERATURE, K and ε

Appendix B. User Guide oft CONTAM3: 3-Dimensional Contaminant Dispersal Code

B.1 General Descripton

The theory described in Chapter 3 for scalar variables has been implemented in a computer program call CONTAM3. Thus code is run after a satisfactory flow field has been obtained using EXACT3. The general steps to run CONTAM3 are:

1. Rename the output file UNIT21 of EXACT3 as FLOW.DAT
2. Prepared an program description file, for example DataFile.
3. Run the program: CONTAM3 <DataFile

The program CONTAM3 will produced a binary outfile UNIT24. If it is desired to continue the calculation, UNIT24 should be copied to or renamed UNIT23 and the initial control parameter INIC (see below) set to 2.

B.2 Description of Input File of CONTAM#

The following is the input file for the problem presented in Chapter 4.4 :

```
1  | Test for CONTAM3 - Practical Application
2  |   75500    36000     2    10
3  |   0.010
4  |   1.0E-5    0.7
5  |    14        3       3
6  |    17
7  |   1     12    15   2   3   3   5          0.0
   |   3      3    11   3   2   3  18          0.0
   |   3     16    23   3   2   3  18          0.0
   |   3     12    15   3   2   6  18          0.0
   |   2      3     2   3  25   3  18          0.0
   |   3      3    19  25  26   3  18          0.0
   |   3     20    23  25  26   7  18          0.0
   |   3      3    11   3   2   3  18          0.0
   |   3     12    15   3   2   6  18          0.0
   |   3     16    23   3   2   3  18          0.0
   |   4      3     4   3  25  18  19          0.0
   |   4      5     8   3  20  18  19          0.0
   |   4      5     8  21  23  18  19          0.0
   |   4      9    23   3  25  18  19          0.0
   |   2     19    20  17  25   3   6          0.0
   |   4     20    23  17  25   7   6          0.0
   |   3     20    23  16  17   3   6          0.0
8  |      1
9  |  20     23  17  25   3   6
10 |      1
11 |  23     23    25  25   7   7       0.9542
```

Input information is written with the following format and order.

1 <u>Comment line</u>

format: A80

example:

```
.........1.........2.........3.........4.........5.........6.........7
Test for CONTAM3 - Practical Application
```

*this comment is printed on top of LP image listing and the listing of each dependent variable.

2 <u>calculation condition 1</u>

stored at: KSTM, MAXT, INIC, NPRINT

format: free

example:

```
  75500    36000     2    10
```

*KSTM maximum time iteration steps

If accumulated time iteration counter KSTP exceeds KSTM, time iteration is terminated and subroutine OUTPUT is processed.

*MAXT maximum CPU time in seconds.

Accumulated CPU time is measured once each time iteration and if it exceeds MAXT, time iteration is terminated and subroutine OUTPUT is processed.

Job termination is, therefore, controlled by either KSTM or MAXT except divergence occurs.

*INIC Input/output unformatted file controller.

For INIC $\geq$ 2, initial flow field is assumed to be stored as an unformatted permanent file on Cyber 855, otherwise, calculation is started from 0 flow field which is generated internally by the subroutine ARAIN.

For INIC $\leq$ 2, final flow field is stored as a new unformatted permanent file on Cyber 855, otherwise, it is lost as soon as the job is terminated.

Recommended value

1 to generate initial flow field using small MAXT or KSTM value.

2 to resume interrupted calculation

NPRINT Print control parameter. Data will be printed to file every NPRINT iterations. This is used to keep the output file size low.

3 Time Step

DT non-dimensional time step

DT is chosen from the stabillity criteria for both convection and diffusion terms.

stored at: DT
format: free
example:

```
0.010
```

4 calculation condition 2

stored at: HPE, ZSC
format: free
example:

```
1.0E-5    0.7
```

*HPE Non-dimensional diffusion coefficincnt

*ZSC constant C3 used in equation (2-47)

5 image listing pointer
stored at array: ICON
format: free
example:

```
6,    6,    6
```

*Spatial variations of all dependent variables is given as a LP image listing for Y-Z Z-X, X-Y plane at locations where this pointer specifies.
For instance, (14, 2, 2) means that LP listing is given for Y-Z plane at I=14, Z-X plane at J=3 and X-Y plane at K=3.

6 number of obstacles
stored at: NIN
format: free
example:

```
    1
```

*NIN is the number of given boundary conditions following this line.

7 obstacle data
stored at arrays: IIN
format: free

```
 20     23   17   25    3    6
```

8 number of boundary conditons
stored at: NIN
format: free
example:

```
    17
```

*NIN is the number of given boundary conditions following this line.

9 boundardy condition data
stored at arrays: IIN, SERV
format: free

```
  3       3    11    3    2    3   18          0.0
  3      16    23    3    2    3   18          0.0
  3      12    15    3    2    6   18          0.0
  2       3     2    3   25    3   18          0.0
  3       3    19   25   26    3   18          0.0
  3      20    23   25   26    7   18          0.0
  3       3    11    3    2    3   18          0.0
  3      12    15    3    2    6   18          0.0
  3      16    23    3    2    3   18          0.0
  4       3     4    3   25   18   19          0.0
  4       5     8    3   20   18   19          0.0
  4       5     8   21   23   18   19          0.0
  4       9    23    3   25   18   19          0.0
  2      19    20   17   25    3    6          0.0
  4      20    23   17   25    7    6          0.0
  3      20    23   16   17    3    6          0.0
```

*Default boundary condition employed in the "CONTAM3" code is the symmetric wall condition described in the next subsection.

*Since boundary condition data are referenced in regular sequence, earlier conditions including default boundary condition may be overwritten by the latter one.
*Boundary condition data consist of boundary condition number, boundary location and boundary parameter.
General format of boundary condition data is given as follows.

Where III is the boundary condition number I1, I2, J1, J2, K1, K2 are the boundary location in X, Y and Z-direction in cell number, SEV is the boundary parameter.

*Boundary condition data for III= 1 to 6 are used to impose discret boundary condition to each dependent variable with the location I=I1 to I2, J=J1 to J2, K=K1 to K2 in cell number

III	function	effect
1	specify concentration	C(I,J,K) = SERV I=I1 to I2, J J1 to J2, K= K1 to K2
2	specify wall boundary normal to X axis	between I=I1 and I2 for J= J1 to J2, K= K1 to K2 I1 is a terminal real cell and I2 is an adjacent dummy cell numbrt
3	specify wall boundary normal to Y axis	between J=J1 and J2 for I= I1 to I2, K= K1 to K2 J1 is a terminal real cell and J2 is an adjacent dummy cell number
4	specify wall boundary normal to Z axis	between K=K1 and K2 for I= I1 to I2, J= J1 to J2 I1 is a terminal real cell and I2 is an adjacent dummy cell number

*Boundary conditions for III= 5 to 7 are prepared to set contaminant flux across the wall boundary appearing in equation 3-47. Thermal flux, SEV, takes positive value when its direction is from dummy cell to real cell.

III	function	boundary location
5	specify wall boundary normal to X axis	same as III = 2
6	specify wall boundary normal to Y axis	same as III = 3
7	specify wall boundary normal to Z axis	same as III = 4

*Boundary conditions for III= 8 to 10 are used to set local flux transfer coefficients defined in equation 3-48.
<u>Concentration must be specified using III=1 boundary condition at the wall adjacent dummy cell before this line appears</u>.
Non-dimensional flux transfer coefficient is stored at SEV.

III	function	boundary location
8	specify wall boundary normal to X axis	same as III= 2
9	specify wall boundary normal to Y axis	same as III= 3
10	specify wall boundary normal to Z axis	same as III= 4

10 number of generation rate sources
stored at: NSOURCES
format: free
example:

```
    1
```

*NSOURCES is the number of given generation rates following this line.

11 generation rate source data
stored at arrays: IIN, S
format: free

```
 20     23  17  25   3   6
```

The each generation source rate is set:
H(I,J,K) = SERV for I= I1 to I2, J= J1 to J2 and K= K1 to K2

B.3 Listing of Contaminant Dispersal Code

```
C
C                 ***************************************
C                   *  CONTAM3 3D CONTAMINANT DISPERSAL
*
C                 ***************************************
C------------------------------------------------------
--------------------
C                   EXPLICIT TIME-MARCHING ARGORITHM FOR
C                   3 DIMENSIONAL CONTAMINANT DISPERSAL
C
C          ----------THEORY BY T.KURABUTI,1987
C           CODE REVISED BY R. GROT 1988 TO REDUCE STORAGE
REQUIREMENTS
C------------------------------------------------------
--------------------
C                   INPUT DATA FORMAT
C--------1---------2---------3---------4---------5-----
----6---------7--
C               TITLE
C     KSTM       MAXT       INIC
C       DT
C      HPE          SC
C       I1         J1         K1
C      NIN
C       III          I1          I2          J1          J2
K1      K2    SEV
C     NOBS
C        I1          I2          J1          J2          K1
K2
C    NSOURCES
C        I1          I2          J1          J2          K1
K2        S
C--------1---------2---------3---------4---------5-----
----6---------7--
C
C
C*****MAIN
C
      PROGRAM MAIN
C
C*****DECLARATION OF COMMON
C
C
      PARAMETER (L=24,M=26,N=19)
C
      COMMON /CONTR/
     1 TITLE(20)
     2,HPE    ,DT
     3,KSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZSC ,HZSC
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1),NOBS,IOB(100,6)
            2,NIN    ,IIN(100,7),UIN(100),NSOURCES,ISOUR-
CE(100,6),S(100)
C
      COMMON
     2 U  (L,M,N),V  (L,M,N),W  (L,M,N)
     3,D  (L,M,N),CFX(L,M,N)
     5,C  (L,M,N),CC(L,M,N)
     7,XD  (L),YD  (M),ZD  (N)
     8,HX  (L),HY  (M),HZ  (N)
     9,XIP(L),YIP(M),ZIP(N)
C
C
      COMMON /SCAL/ ICON(3),NPRINT
C
C
      OPEN (10,FILE='CONTAM3.OUT' )
C
C*****ARRAY INITIALIZATION
C
      CALL ARAIN
C
C*****DATA INITIALIZATION
C
      CALL DATAIN
C
      IF ( INIC .LE. 1 ) GO TO 100
C
      OPEN (24,FILE='UNIT24',FORM='UNFORMATTED')
      READ(24) C,KSTP
      CLOSE(24)
C
  100 CONTINUE
C
C*****SOLVE CONTAMINANT DISPERSAL EQUATIONS
C
      CALL SOLVEC
C
      IF ( INIC .EQ. 0 ) GO TO 200
      IF ( INIC .EQ. 3 ) GO TO 200
C
      OPEN (23,FILE='UNIT23',FORM='UNFORMATTED')
      WRITE(23) C,KSTP
      CLOSE(23)
C
  200 CONTINUE
C
C*****JOB TERMINATION
C
      CALL OUTPUT
C
      STOP
      END
      SUBROUTINE DATAIN
C
C*****DECRALATION OF COMMON
C
C
C
      PARAMETER (L=24,M=26,N=19)
C
      COMMON /CONTR/
     1 TITLE(20)
     2,HPE    ,DT
     3,KSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZSC ,HZSC
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1),NOBS,IOB(100,6)
            2,NIN    ,IIN(100,7),UIN(100),NSOURCES,ISOUR-
```

```
CE(100,6),S(100)
C
      COMMON
     2 U   (L,M,N),V   (L,M,N),W   (L,M,N)
     3,D   (L,M,N),CFX(L,M,N)
     5,C   (L,M,N),CC(L,M,N)
     7,XD  (L),YD  (M),ZD  (N)
     8,HX  (L),HY  (M),HZ  (N)
     9,XIP(L),YIP(M),ZIP(N)
C
C
      COMMON /SCAL/ ICON(3),NPRINT
C
C*****DATA INPUT
C
      READ(5,100)   TITLE
      WRITE(6,600)   TITLE
      WRITE(10,600)   TITLE
      READ(5,*)  KSTM,MAXT,INIC,NPRINT
      IF(NPRINT.LE.0) NPRINT = 0
      WRITE(6,610)   KSTM,MAXT,INIC ,NPRINT
      WRITE(10,610)   KSTM,MAXT,INIC ,NPRINT
      READ(5,*)   DT
      WRITE(6,620)   DT
      WRITE(10,620)   DT
      READ(5,*)   HPE, ZSC
      WRITE(6,630)   HPE,ZSC
      WRITE(10,630)   HPE,ZSC
      READ(5,*)    (ICON(I),I=1,3)
      WRITE(6,650)    (ICON(I),I=1,3)
      WRITE(10,650)    (ICON(I),I=1,3)
C
C
C*****BOUNDARY CONDITION INPUT
C
      READ(5,*)   NIN
      DO 10 I=1,NIN
   10 READ(5,*)    (IIN(I,J),J=1,7),UIN(I)
C
      WRITE(6,660)
      WRITE(10,660)
      DO 40 I=1,NIN
      WRITE(6,661)    (IIN(I,J),J=1,7),UIN(I)
      WRITE(10,661)    (IIN(I,J),J=1,7),UIN(I)
   40 CONTINUE
C
C
C*****OBSTACLE BOUNDARY CONDITION
C
      READ(5,*)   NOBS
C
      IF ( NOBS .NE. 0 ) THEN
C
      DO 20 III=1,NOBS
      READ(5,*) (IOB(III,J),J=1,6)
C
   20 CONTINUE
      END IF
      IF ( NOBS .EQ. 0 ) GO TO 55
      WRITE(6,670)
      WRITE(10,670)
      DO 50 I=1,NOBS
      WRITE(6,671)    (IOB(I,J),J=1,6)
      WRITE(10,671)    (IOB(I,J),J=1,6)
   50 CONTINUE
   55 CONTINUE
C
C   CONTAMINANT SOURCES
C
      READ(5,*)   NSOURCES
      IF(NSOURCES.NE.0) THEN
      DO 25 I=1,NSOURCES
   25 READ(5,*)    (ISOURCE(I,J),J=1,6),S(I)
      END IF
C
C
C*****INITIAL CALCULATION 1
C
      VOLM = (L-3) * (M-3) * (N-3)
      VOLM = 1.0 / VOLM
C
      HZSC = 1.0 / ZSC
C
C*****OUTPUT OF INITIAL DATA
C
C
C
      WRITE(6,680)
      WRITE(10,680)
      IF(NSOURCES.NE.0) THEN
C
      DO 60 I=1,NSOURCES
      WRITE(10,681)    (ISOURCE(I,J),J=1,6),S(I)
   60 WRITE(6,681)    (ISOURCE(I,J),J=1,6),S(I)
      END IF
C
C*****FORMAT STATEMENT
C
C*****READ
C
  100 FORMAT (20A4     )
  110 FORMAT ( 4I10    )
  130 FORMAT ( 6I5,F10.0)
  120 FORMAT ( 7F10.0 )
  140 FORMAT ( 3I10    )
  150 FORMAT (  I10    )
  160 FORMAT ( 7I 5    ,F10.0 )
  170 FORMAT ( 6I10    )
  180 FORMAT ( 6I10    ,F10.0 )
C
C*****WRITE
C
  600 FORMAT (1H ////1H ,20X,20A4,10X////)
C
  610 FORMAT (
     1          1H ,10X,'KSTM:.......TIME ITERATION MAX
NO.=',I15/
     3             ,1H  ,10X,'MAXT:.........CPU  TIME  MAX
(SEC.)=',I15/
     4             ,1H  ,10X,'INIC:....INITIAL FILE DATA
CONTROL=',I15/
     5          ,1H ,10X,'NPRINT:............PRINT CONTROL
=',I15 )
C
  620 FORMAT (
     6          1H  ,10X,'DT    :...............TIME
DIVISION=',F15.7/)
  630 FORMAT (
     1                          1H   ,10X,'HPE   :
',F15.7/
     7                         ,1H   ,10X,'ZSC   :
',F15.7/ )
C
C
  650 FORMAT (1H ,10X,'*****DATA OUTPUT POINT*****'    /
     1                 ,10X,'(',I10,',',I10,',',I10,')'
// )
C
  660  FORMAT  (1H  ,10X,'*****BOUNDARY  CONDITION*****'
/
     1         '...NOB....I1....I2....J1'
     2        ,'....J2....K1....K2.........SPV' )
  661 FORMAT (1H ,7I6,F12.5)
C
  670 FORMAT (1H ,10X,'*****OBSTACLE CONDITION*****'/
     1         '....I1....I2....J1'
     2        ,'....J2....K1....K2' )
  671 FORMAT (1H ,6I6)
C
  680 FORMAT (1H ,10X,'*****SOURCES*****'/
     1         '....I1....I2....J1....J2'
     2        ,'....K1....K2.........S'   )
  681 FORMAT (1H ,6I6,1PE12.5)
C
      RETURN
C
      END
      SUBROUTINE ARAIN
C
C*****DECLARATION OF COMMON
C
C
      PARAMETER (L=24,M=26,N=19)
C
      COMMON /CONTR/
     1 TITLE(20)
     2,HPE   ,DT
     3,KSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZSC ,HZSC
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1),NOBS,IOB(100,6)
            2,NIN    ,IIN(100,7),UIN(100),NSOURCES,ISOUR-
CE(100,6),S(100)
C
      COMMON
     2 U   (L,M,N),V   (L,M,N),W   (L,M,N)
     3,D   (L,M,N),CFX(L,M,N)
     5,C   (L,M,N),CC(L,M,N)
     7,XD  (L),YD  (M),ZD  (N)
     8,HX  (L),HY  (M),HZ  (N)
     9,XIP(L),YIP(M),ZIP(N)
C
C
      COMMON /SCAL/ ICON(3),NPRINT
C
      OPEN (22,FILE='flow.dat',FORM='UNFORMATTED')
      READ(22) U,V,W,CC,D,CC,CFX,DX,DY,DZ,KK
      CLOSE(22)
C
C*****ARRAY INITIALIZATION
C
      DO 100 K=1,N
      DO 100 J=1,M
      DO 100 I=1,L
C
```

```
      CFX(I,J,K) = 0.0
      C(I,J,K) = 0.0
      IF(CC(I,J,K).NE.0.0) THEN
      D(I,J,K) = ZSC * D(I,J,K)**2/CC(I,J,K)
      ELSE
      D(I,J,K) = 0.0
      END IF
      CC(I,J,K) = 0.0
C
  100 CONTINUE
      DO 110 I=1,L
      HX(I) = 1.0/DX(I)
      XD(I)=2.0/(DX(I)+DX(I+1))
      XIP(I)=DX(I+1)/(DX(I)+DX(I+1))
  110 CONTINUE
      DO 120 J=1,M
      HY(J) = 1.0/DY(J)
      YD(J)=2.0/(DY(J)+DY(J+1))
      YIP(J)=DY(J+1)/(DY(J)+DY(J+1))
  120 CONTINUE
      DO 130 K=1,N
      HZ(K)=1.0/DZ(K)
      ZD(K)=2.0/(DZ(K)+DZ(K+1))
      ZIP(K)=DZ(K)/(DZ(K)+DZ(K+1))
  130 CONTINUE
C
C
      RETURN
C
      END
C
C*****ROOM AIR CONTAMINANT DISPERSAL SOLVEC
C
      SUBROUTINE SOLVEC
C
C
      PARAMETER (L=24,M=26,N=19)
C
      INTEGER TIME
      DIMENSION DFX(L,M,N), DFY(L,M,N),DFZ(L,M,N)
C
      COMMON /CONTR/
     1 TITLE(20)
     2,HPE   ,DT
     3,KSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD ,ZSC ,HZSC
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1),NOBS,IOB(100,6)
            2,NIN   ,IIN(100,7),UIN(100),NSOURCES,ISOUR-
CE(100,6),S(100)
C
      COMMON
     2 U  (L,M,N),V  (L,M,N),W  (L,M,N)
     3,D  (L,M,N),CFX(L,M,N)
     5,C  (L,M,N),CC(L,M,N)
     7,XD (L),YD (M),ZD (N)
     8,HX (L),HY (M),HZ (N)
     9,XIP(L),YIP(M),ZIP(N)
C
C
      COMMON /SCAL/ ICON(3),NPRINT
C
C
C
      CALL BCONSUB
C
      IF ( INIC .EQ. 1 ) KSTP = 1
      IKSTP = KSTP
      KSTP = KSTP
      NSTP = 0
      KTIM = 0
      CAVE = 0.0
      ENAV = 0.0
C
      JTIME1 = TIME()
      WRITE(6,3000) JTIME1
      WRITE(10,3000) JTIME1
 3000 FORMAT(1H ,"Initial Time is ",I15," seconds")
C
C*****WORK VECTOR INITIALIZATION
C
      DO 1000 I=1,L
      DO 1000 J=1,M
      DO 1000 K=1,N
      DFX(I,J,K) = 0.0
      DFY(I,J,K) = 0.0
      DFZ(I,J,K) = 0.0
      CC(I,J,K) = 0.0
 1000 CONTINUE
C
C*****EDDY DIFFUSIVITY INTERPOLATION
C
      DO 40 K=2,N-1
      DO 40 J=2,M-1
      DO 40 I=2,L-1
C
       DFX(I,J,K) = D(I,J,K) * XIP(I) + D(I+1,J,K) * (
1.0-XIP(I) )
       DFY(I,J,K) = D(I,J,K) * YIP(J) + D(I,J+1,K) * (
1.0-YIP(J) )
       DFZ(I,J,K) = D(I,J,K) * ZIP(K) + D(I,J,K+1) * (
1.0-ZIP(K) )
C
   40 CONTINUE
C
      DO 45 K=1,N
      DO 45 J=1,M
      DO 45 I=1,L
C
      DFX(I,J,K) = ( DFX(I,J,K) * HZSC + HPE ) * XD(I)
      DFY(I,J,K) = ( DFY(I,J,K) * HZSC + HPE ) * YD(J)
      DFZ(I,J,K) = ( DFZ(I,J,K) * HZSC + HPE ) * ZD(K)
C
   45 CONTINUE
C
      DO 1045 I=1,L
      DO 1045 J=1,M
      DO 1045 K=1,N
      DFX(I,J,K) = DFX(I,J,K) - 0.5 * U(I,J,K)
C
   IF(DFX(I,J,K).LE.(-U(I,J,K))) DFX(I,J,K) = -U(I,J,K)
C
      IF(DFX(I,J,K).LE.0.0) DFX(I,J,K)=0.0
C
      DFY(I,J,K) = DFY(I,J,K) - 0.5 * V(I,J,K)
C
   IF(DFY(I,J,K).LE.(-V(I,J,K))) DFY(I,J,K) = -V(I,J,K)
C
      IF(DFY(I,J,K).LE.0.0) DFY(I,J,K)=0.0
C
      DFZ(I,J,K) = DFZ(I,J,K) - 0.5 * W(I,J,K)
C
   IF(DFZ(I,J,K).LE.(-W(I,J,K))) DFZ(I,J,K) = -W(I,J,K)
C
      IF(DFZ(I,J,K).LE.0.0) DFZ(I,J,K)=0.0
C
 1045 CONTINUE
      WRITE(6,3001)
 3001 FORMAT(1H ,"Starting Solution Loop")
C
  200 CONTINUE
C
C
C*****TERMINATION CRITERIA
C
     IF ( (KSTP.GT.KSTM) .OR. (KTIM.GT.MAXT) ) GO TO 2000
C
C
      DO 1050 I=1,L
      DO 1050 J=1,M
      DO 1050 K=1,N
      CFX(I,J,K) = 0.0
      CC(I,J,K) = 0.0
 1050 CONTINUE
C
C*****CALCULATION OF CONVECTION-DIFFUSION FLUX
C
      DO 50 K=2,N-1
      DO 50 J=2,M-1
      DO 50 I=2,L-1
C
      CFX(I,J,K) = DFX(I,J,K) * ( C(I,J,K) - C(I+1,J,K)
)
C
   50 CONTINUE
C
      DO 1055 I=1,L
      DO 1055 J=1,M
      DO 1055 K=1,N
      CFX(I,J,K) = CFX(I,J,K)+C(I,J,K)*U(I,J,K)
 1055 CONTINUE
C
C*****SET C
C
      DO 55 K=3,N-1
      DO 55 J=3,M-1
      DO 55 I=3,L-1
C
     CC(I,J,K) = CC(I,J,K) + ( CFX(I-1,J,K) - CFX(I,J,K)
) * HX(I)
C
   55 CONTINUE
C
C*****CALCULATION    OF    CONVECTION-DIFFUSION    FLUX
Y-DIRECTION
C
C
      DO 1150 I=1,L
      DO 1150 J=1,M
      DO 1150 K=1,N
       CFX(I,J,K) = 0.0
 1150 CONTINUE
      DO 150 K=2,N-1
      DO 150 J=2,M-1
      DO 150 I=2,L-1
```

```
C
      CFX(I,J,K) = DFY(I,J,K) * ( C(I,J,K) - C(I,J+1,K)
)
C
  150 CONTINUE
C
      DO 1155 I=1,L
      DO 1155 J=1,M
      DO 1155 K=1,N
      CFX(I,J,K) = CFX(I,J,K)+C(I,J,K)*V(I,J,K)
 1155 CONTINUE
C
C*****SET C
C
      DO 155 K=3,N-1
      DO 155 J=3,M-1
      DO 155 I=3,L-1
C
     CC(I,J,K) = CC(I,J,K) + ( CFX(I,J-1,K) - CFX(I,J,K)
) * HY(J)
C
  155 CONTINUE
C
C*****CALCULATION    OF    CONVECTION-DIFFUSION    FLUX
Z-DIRECTION
C
      DO 2150 I=1,L
      DO 2150 J=1,M
      DO 2150 K=1,N
       CFX(I,J,K) = 0.0
 2150 CONTINUE
C
      DO 250 K=2,N-1
      DO 250 J=2,M-1
      DO 250 I=2,L-1
C
      CFX(I,J,K) = DFZ(I,J,K) * ( C(I,J,K) - C(I,J,K+1)
)
C
  250 CONTINUE
C
      DO 1250 I=1,L
      DO 1250 J=1,M
      DO 1250 K=1,N
      CFX(I,J,K) = CFX(I,J,K)+C(I,J,K)*W(I,J,K)
 1250 CONTINUE
C
C*****SET C
C
      DO 255 K=3,N-1
      DO 255 J=3,M-1
      DO 255 I=3,L-1
C
     CC(I,J,K) = CC(I,J,K) + ( CFX(I,J,K-1) - CFX(I,J,K)
) * HZ(K)
C
  255 CONTINUE
C
C**** OBSTACTLES CHECKED
C
      IF(NSOURCES.NE.0) THEN
      DO 400 III=1,NSOURCES
      I1 = ISOURCE(III,1)
      I2 = ISOURCE(III,2)
      J1 = ISOURCE(III,3)
      J2 = ISOURCE(III,4)
      K1 = ISOURCE(III,5)
      K2 = ISOURCE(III,6)
      DO 400 I=I1,I2
      DO 400 J=J1,J2
      DO 400 K=K1,K2
      CC(I,J,K) = CC(I,J,K) + S(III)
  400 CONTINUE
      END IF
C
      IF(NOBS.NE.0) THEN
      DO 300 III=1,NOBS
      I1 = IOB(III,1)
      I2 = IOB(III,2)
      J1 = IOB(III,3)
      J2 = IOB(III,4)
      K1 = IOB(III,5)
      K2 = IOB(III,6)
      DO 300 I=I1,I2
      DO 300 J=J1,J2
      DO 300 K=K1,K2
      CC(I,J,K) = 0.0
  300 CONTINUE
      END IF
```

```
C
      DO 1300 I=1,L
      DO 1300 J=1,M
      DO 1300 K=1,N
      C(I,J,K) = C(I,J,K) + DT * CC(I,J,K)
 1300 CONTINUE
C
C
C CALCULATE AVERAGE CHANGES OF  C
C
      DCAVE = 0.0
      CAVE = 0.0
      CMAX = 0.0
      DCMAX = 0.0
      IMAX = 3
      JMAX = 3
      KMAX = 3
      IDMAX = 3
      JDMAX = 3
      KDMAX = 3
C
      DO 110 K=3,N-1
      DO 110 J=3,M-1
      DO 110 I=3,L-1
      CAVE = CAVE + C(I,J,K)
      DCAVE = DCAVE + CC(I,J,K)**2
      IF(C(I,J,K).GT.CMAX) THEN
      IMAX = I
      JMAX = J
      KMAX = K
      CMAX = C(I,J,K)
      END IF
      If(ABS(CC(I,J,K)).GT.DCMAX) THEN
      IDMAX = I
      JDMAX = J
      KDMAX = K
      DCMAX = ABS(CC(I,J,K))
      ENDIF
C
  110 CONTINUE
C
      CALL BCONSUB
C
C*****CALCULATE  AVERAGE CHANGES
C
      CAVE =  CAVE * VOLM
      DCAVE = SQRT(DCAVE * VOLM)
C
      KTIM = TIME() - JTIME1
     IF(KSTP.EQ.IKSTP.OR.MOD(KSTP,20).EQ.0) WRITE(6,205)
                  WRITE(6,206)   KSTP,KTIM,CAVE,DCAVE,I-
MAX,JMAX,KMAX,CMAX,
     1IDMAX,JDMAX,KDMAX,DCMAX
      IF(KSTP.EQ.IKSTP.OR.MOD(KSTP,NPRINT).EQ.0) THEN
C
C
          IF(KSTP.EQ.IKSTP.OR.MOD(KSTP,20*NPRINT).EQ.0)
WRITE(10,205)
C
                WRITE(10,206)   KSTP,KTIM,CAVE,DCAVE,I-
MAX,JMAX,KMAX,CMAX,
     1IDMAX,JDMAX,KDMAX,DCMAX
      END IF
C
  205 FORMAT(1H , '..KSTP..KTIM'
     1              ,'......CAVE'
     2              ,'.....DCAVE...I...J...K......CMAX',
     3               '...I...J...K.....DCMAX')
  206 FORMAT(1H , 2I6,1P2E10.3,3I4,1PE10.3,3I4,1PE10.3)
C
C
      IF(MOD(KSTP,100).EQ.0) THEN
      OPEN (23,FILE='UNIT23',FORM='UNFORMATTED')
      WRITE(23) C,KSTP
      CLOSE(23)
      END IF
      KSTP = KSTP + 1
C
      GO TO 200
C
 2000 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE BCONSUB
C
C
C
      PARAMETER (L=24,M=26,N=19)
C
      COMMON /CONTR/
     1 TITLE(20)
     2,HPE    ,DT
     3,KSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD  ,ZSC  ,HZSC
C
      COMMON /BCON/
```

```
      1 DX(L+1)    ,DY(M+1)    ,DZ(N+1),NOBS,IOB(100,6)
             2,NIN    ,IIN(100,7),UIN(100),NSOURCES,ISOUR-
CE(100,6),S(100)
C
       COMMON
      2 U  (L,M,N),V  (L,M,N),W  (L,M,N)
      3,D  (L,M,N),CFX(L,M,N)
      5,C  (L,M,N),CC(L,M,N)
      7,XD (L),YD (M),ZD (N)
      8,HX (L),HY (M),HZ (N)
      9,XIP(L),YIP(M),ZIP(N)
C
C
       COMMON /SCAL/ ICON(3),NPRINT
C
C*****SET FREE SLIP WALL 1,2
C
       DO 10 K=2,N-1
       DO 10 J=2,M-1
C
       C (2,J,K) = C (3  ,J,K)
       C (L,J,K) = C (L-1,J,K)
C
   10 CONTINUE
C
C*****SET FREE SLIP WALL 3,4
C
       DO 20 K=2,N-1
       DO 20 I=2,L-1
C
       C (I,2,K) = C (I,  3,K)
       C (I,M,K) = C (I,M-1,K)
C
C
   20 CONTINUE
C
C*****SET FREE SLIP WALL 5,6
C
       DO 30 J=2,M-1
       DO 30 I=2,L-1
C
       C (I,J,2) = C (I,J,  3)
       C (I,J,N) = C (I,J,N-1)
C
   30 CONTINUE
C
C*****BOUNDARY CONDITION SET
C
       DO 100 III=1,NIN
C
       I1 = IIN(III,2)
       I2 = IIN(III,3)
       J1 = IIN(III,4)
       J2 = IIN(III,5)
       K1 = IIN(III,6)
       K2 = IIN(III,7)
C
       SEV = UIN(III)
C
                                                        GO        TO
(110,120,130,140,150,160,170,180,190,200),IIN(III,1)
C
C*****C VALUE
C
  110 DO 161 K=K1,K2
       DO 161 J=J1,J2
       DO 161 I=I1,I2
  161 C (I,J,K) = SEV
       GO TO 100
C
C*****HEAT FLUX SPECIFIED ( X DIRECTION )
C
  120 DO 231 K=K1,K2
       DO 231 J=J1,J2
C
       RAM1 = D(I1,J,K) +HPE
       RAM  =   RAM1                     * HX(I1)
       C(I2,J,K) = SEV / RAM + C(I1,J,K)
C
  231 CONTINUE
C
       GO TO 100
C
C*****HEAT FLUX SPECIFIED ( Y DIRECTION )
C
  130 DO 241 K=K1,K2
       DO 241 I=I1,I2
C
       RAM1 = D(I,J1,K) + HPE
       RAM  =   RAM1                     * HY(J1)
       C(I,J2,K) = SEV / RAM + C(I,J1,K)
C
  241 CONTINUE
C
       GO TO 100
C
C*****HEAT FLUX SPECIFIED ( Z DIRECTION )
C
  140 DO 251 J=J1,J2
       DO 251 I=I1,I2
C
       RAM1 = D(I,J,K1) + HPE
       RAM  =   RAM1                     * HZ(K1)
       C(I,J,K2) = SEV / RAM + C(I,J,K1)
C
  251 CONTINUE
C
       GO TO 100
C
C*****HEAT TRANSFER COEFFICIENT SPECIFIED ( X DIRECTION
)
C
  150 DO 261 K=K1,K2
       DO 261 J=J1,J2
C
       RAM1 = HPE + D(I1,J,K)
       RAM  =   RAM1                     * HX(I1)
       C(I2,J,K) = SEV / RAM * ( C(I2,J,K) - C(I1,J,K) )
+ C(I1,J,K)
C
  261 CONTINUE
C
       GO TO 100
C
C*****HEAT TRANSFER COEFFICIENT SPECIFIED ( Y DIRECTION
)
C
  160 DO 271 K=K1,K2
       DO 271 I=I1,I2
C
       RAM1 = D(I,J1,K) + HPE
       RAM  =   RAM1                     * HY(J1)
       C(I,J2,K) = SEV / RAM * ( C(I,J2,K) - C(I,J1,K) )
+ C(I,J1,K)
C
  271 CONTINUE
C
       GO TO 100
C
C*****HEAT TRANSFER COEFFICIENT SPECIFIED ( Z DIRECTION
)
C
  170 DO 281 J=J1,J2
       DO 281 I=I1,I2
C
       RAM1 = HPE + D(I,J,K1)
       RAM  =   RAM1                     * HY(J1)
       C(I,J,K2) = SEV / RAM * ( C(I,J,K2) - C(I,J,K1) )
+ C(I,J,K1)
C
  281 CONTINUE
C
       GO TO 100
C
C*****WALL TEMPERATURE SPECIFIED ( X DIRECTION )
C
  180 DO 291 K=K1,K2
       DO 291 J=J1,J2
C
  291 C(I2,J,K) = 2.0 * SEV - C(I1,J,K)
C
       GO TO 100
C
C*****WALL TEMPERATURE SPECIFIED ( Y DIRECTION )
C
  190 DO 301 K=K1,K2
       DO 301 I=I1,I2
C
  301 C(I,J2,K) = 2.0 * SEV - C(I,J1,K)
C
       GO TO 100
C
C*****WALL TEMPERATURE SPECIFIED ( Z DIRECTION )
C
  200 DO 311 J=J1,J2
       DO 311 I=I1,I2
C
  311 C(I,J,K2) = 2.0 * SEV - C(I,J,K1)
C
       GO TO 100
C
C*****
C
  100 CONTINUE
C
       CALL BSET
C
       RETURN
C
       END
C
C DUMMY ROUTINE
C
       SUBROUTINE BSET
C
       RETURN
```

```
      END
C*****OUTPUT*********************************************
********************
C
      SUBROUTINE OUTPUT
C
      CHARACTER*4 THEMA,PLAN,TERM
C
      PARAMETER (L=24,M=26,N=19)
C
      COMMON /CONTR/
     1 TITLE(20)
     2,HPE    ,DT
     3,KSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD  ,ZSC  ,HZSC
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1),NOBS,IOB(100,6)
             2,NIN   ,IIN(100,7),UIN(100),NSOURCES,ISOUR-
CE(100,6),S(100)
C
      COMMON
     2 U   (L,M,N),V   (L,M,N),W   (L,M,N)
     3,D   (L,M,N),CFX(L,M,N)
     5,C   (L,M,N),CC(L,M,N)
     7,XD  (L),YD  (M),ZD  (N)
     8,HX  (L),HY  (M),HZ  (N)
     9,XIP(L),YIP(M),ZIP(N)
C
C
      COMMON /SCAL/ ICON(3),NPRINT
C
      THEMA = '    C'
  200 CONTINUE
C
C*****Y-Z PLANE
C
      PLAN = ' Y-Z'
      TERM = ' I ='
      IC = ICON(1)
      INDEX = 1
C
      CALL PRINT1 (THEMA,PLAN,TERM,IC,INDEX)
C
C*****Z-X PLANE
C
      PLAN = ' Z-X'
      TERM = ' J ='
      IC = ICON(2)
      INDEX = 2
C
      CALL PRINT1 (THEMA,PLAN,TERM,IC,INDEX)
C
C*****X-Y PLANE
C
      PLAN = ' X-Y'
      TERM = ' K ='
      IC = ICON(3)
      INDEX = 3
C
      CALL PRINT1 (THEMA,PLAN,TERM,IC,INDEX)
C
      RETURN
      END
C
C
C
C
C
C*****PRINT**********************************************
************************
C
      SUBROUTINE PRINT1(THEMA,PLAN,TERM,IC,INDEX)
C
      CHARACTER*4 THEMA,PLAN,TERM
C
      PARAMETER (L=24,M=26,N=19)
C
      COMMON /CONTR/
     1 TITLE(20)
     2,HPE    ,DT
     3,KSTM,INIC,MAXT,VOLM,KSTP
     4,ZCD  ,ZSC  ,HZSC
C
      COMMON /BCON/
     1 DX(L+1)    ,DY(M+1)    ,DZ(N+1),NOBS,IOB(100,6)
             2,NIN   ,IIN(100,7),UIN(100),NSOURCES,ISOUR-
CE(100,6),S(100)
C
      COMMON
     2 U   (L,M,N),V   (L,M,N),W   (L,M,N)
     3,D   (L,M,N),CFX(L,M,N)
     5,C   (L,M,N),CC(L,M,N)
     7,XD  (L),YD  (M),ZD  (N)
     8,HX  (L),HY  (M),HZ  (N)
     9,XIP(L),YIP(M),ZIP(N)
C
C
      COMMON /SCAL/ ICON(3),NPRINT
C
C*****OUTPUT TITLE
C
      WRITE(10,600) TITLE
      WRITE(10,601) THEMA,PLAN,TERM,IC,KSTP
      GO TO ( 10,20,30 ),INDEX
C
C*****Y-Z PLANE
C
   10 CONTINUE
      KA = 1
  100 CONTINUE
      KK = 6 * ( KA-1 ) + 1
      KE = 6 * KA
C
      IF ( (KE-N) .GE. 0 ) GO TO 103
C
      WRITE(10,602) (K,K=KK,KE)
      WRITE(10,101) (J,(C(IC,J,K),K=KK,KE),J=1,M)
      WRITE(10,102)
      KA = KA + 1
      GO TO 100
C
  103 CONTINUE
      WRITE(10,602) (K,K=KK,N)
      DO 104 J=1,M
      WRITE(10,101) J,(C(IC,J,K),K=KK,N)
  104 CONTINUE
C
      RETURN
C
C*****Z-X PLANE
C
   20 CONTINUE
      IA = 1
  200 CONTINUE
      II = 6 * ( IA-1 ) + 1
      IE = 6 * IA
C
      IF ( (IE-L) .GE. 0 ) GO TO 203
C
      WRITE(10,602) (I,I=II,IE)
      WRITE(10,101) (K,(C(I,IC,K),I=II,IE),K=1,N)
      WRITE(10,102)
      IA = IA + 1
      GO TO 200
C
  203 CONTINUE
      WRITE(10,602) (I,I=II,L)
      DO 204 K=1,N
      WRITE(10,101) K,(C(I,IC,K),I=II,L)
  204 CONTINUE
C
      RETURN
C
C*****X-Y PLANE
C
   30 CONTINUE
      JA = 1
  300 CONTINUE
      JJ = 6 * ( JA-1 ) + 1
      JE = 6 * JA
C
      IF ( (JE-M) .GE. 0 ) GO TO 303
C
      WRITE(10,602) (J,J=JJ,JE)
      WRITE(10,101) (I,(C(I,J,IC),J=JJ,JE),I=1,L)
      WRITE(10,102)
      JA = JA + 1
      GO TO 300
C
  303 CONTINUE
      WRITE(10,602) (J,J=JJ,M)
      DO 304 I=1,L
      WRITE(10,101) I,(C(I,J,IC),J=JJ,M)
  304 CONTINUE
C
      RETURN
C
C*****FORMAT STATEMENT
C
  101 FORMAT (1H ,I5,6E12.5)
  102 FORMAT (1H ///         )
C
  600 FORMAT (1H ///1H ,20A4)
 601 FORMAT (1H ,A4,'-DISTRIBUTION ON ',A4,' PLAN',A4,I4
     1                              ,'    KSTP=',I10///)
  602 FORMAT (1H ,6I12)
C
      END
 ?
```

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 89-4211

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
JANUARY 1990

4. TITLE AND SUBTITLE

A Numerical Method for Calculating Indoor Airflows Using a Turbulence Model

5. AUTHOR(S)

T. Kurabuchi, J. B. Fang, Richard A. Grot

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

[ ] DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This report describes a numerical method for simulating indoor air flows in a building using a k-e turbulence method. The model treats three dimensional non-isothermal turbulent flows using the Boussinesq approximation for buoyance. It solves the resulting nonlinear system of momentum, energy and turbulence equations by an explicit time marching technique to obtain a solution to either a steady state or transient flow. An upwind/central combination scheme with arbitrary specification for the switching parameter is used to approximate the convective terms. This switching parameter can be specified at each point in the flow regime allowing for different strategies in different flow regions. The switching technique includes both the central and hybrid schemes found in the literature. A pressure relaxation method is used to satisfy the Poisson equation for continuity. The model handles a variety of flow, pressure, temperature and heat flux boundary conditions including prescribed inflows, outflows by either prescribing the flow or pressure, wall boundary conditions together with heat flux and temperature and/or heat transfer coefficients specified on the boundary. Volumetric heat sources can also be included. The model has the ability of handling an arbitrary number of obstacles in the flow region. This permits the modeling of the effect of furniture and partitions on the flow field and also provides a means for modeling multi-room air flows. The predicted air flows can be used in a companion computer model for predicting the three dimension dispersion of contaminants in a building. The computer code for this model exists both in a vectorized version for the Cyber 205 supercomputer and in a non-vectorized version which has been successfully run on a Sun 3/260 workstation a floating point processor board (based on a Weitek 1167) under a UNIX operating system on a Compaq 386/25 computer equipped

see attached continuation

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

Room air movement, contaminant dispersal, turbulent flow, numerical simulation, indoor air quality, bouyancy driven flows.

13. AVAILABILITY

[X] UNLIMITED
[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
[ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
122

15. PRICE
A06

ELECTRONIC FORM

Continued from abstract

with either an Intel 80387 or a Weitek 3167 coprocessor under an extended DOS operating system. The relative performance of these systems for the examples considered in this report are 1 second per iteration for the Cyber 205,9 seconds for the Compaq 386/25 with a Weitek 3167 copressor, 30 seconds for the Compaq 386 with a 387 copressor and 90 seconds for the Sun 3/260 under UNIX. Isothermal simulations seem to converge in approximately 10,000 iterations and non-isothermal simulations in approximately 30,000 iterations. Several ideal and practical applications of the model are presented and the results of the simulations are compared with existing experimental data contained in the literature.