TRANSIENT THREE-DIMENSIONAL PREDICTIONS OF TURBULENT FLOWS IN CYLINDRICAL AND CARTESIAN COORDINATE SYSTEMS

By

AHMED A. BUSNAINA

Bachelor of Science
University of Al Fateh
Tripoli, Libya
1976

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1979

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
July, 1983

TRANSIENT THREE-DIMENSIONAL PREDICTIONS OF TURBULENT FLOWS IN CYLINDRICAL AND CARTESIAN COORDINATE SYSTEMS

Thesis Approved:

David G. Lilley
Thesis Adviser

Robert M. Worth

A. J. Ghajar

James W. Maxwell

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my gratitude to my adviser, Dr. D. G. Lilley, for his valuable suggestions and advice throughout the progress of this research. Further advice and criticism of my committee members, Dr. P. M. Moretti, Dr. J. W. Maxwell, and Dr. A. J. Ghajar is greatly acknowledged.

Special gratitude and appreciation is expressed to my loving wife, Zainab, and our children, Wedad and Ibrahim. Their support, understanding, sacrifices, and encouragement have been overwhelming. I deeply appreciate the encouragement and financial support given by my parents, Mr. and Mrs. Ali Busnaina.

Part of this work was funded by the Office of Water Research and Technology and Oklahoma Water Resources Research Institute under Cooperative Program Project A-098-OKLA.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

C_1	k-ε model constant
C_2	k-ε model constant
C_D	k-ε model constant
C_μ	k-ε model constant
D_p	propeller diameter, m
D*	nondimensional prope-ler diameter, $D^* = D_p/H$
DF	dilution factor, $DF = (\rho_r - \rho_2)/(\rho_1 - \rho_2)$
E	constant in the log-law
Frd	densimetric Froude number, $Frd = v_p/[g(\Delta\rho/\rho_1)H]^{1/2}$
g	gravitational acceleration, m/s^2
G	generation rate of turbulence energy, J/kg-s
H	total depth of model, m
k	kinetic energy of turbulence per unit mass, J/kg
k_P	kinetic energy at point P
K	constant in the log-law
K*	nondimensional propeller distance, normalized with respect to H
L	propeller depth below the surface, m
ℓ	length scale of turbulence, m
L*	nondimensional propeller depth, $L^* = L/H$
m_1	mass fraction of fluid of density ρ_1 (epilimnion)
m_2	mass fraction of fluid of density ρ_2 (hypolimnion)
p	time-mean static pressure deviation from hydrostatic pressure, Pa

P	typical point in flowfield
Q_p	propeller flow rate, kg/s
Q_r	release flow rate, kg/s
Q*	nondimensional flow rate, $Q^* = Q_p/Q_r$
R	jet-to-crossflow velocity ratio, $R = u_j/u_{in}$
r,z,θ	cylindrical coordination and radians for θ
r_o	deflected jet radius, m
S_k	source term for k
S_ε	source term for ε
t	time, s
u, v, w	time-mean velocity components in x, y, z or r, z, θ coordinate directions, m/s
U_d	dilution jet velocity
v_o	chamber inlet axial velocity
v_p	propeller velocity, m/s
V_p	component of velocity parallel to the wall in the near wall region, m/s
x, y, z	cartesian coordinates, m
y_p	normal distance from nearby wall to point P, m
Z_T	thermocline location (measured from the bottom), m
Z_T^*	nondimensional thermocline location, $Z_T^* = Z_T/H$
ε	turbulent energy dissipation rate, m^2/s^3
μ	absolute viscosity, $N \cdot s/m^2$
μ_t	turbulent viscosity, $N \cdot s/m^2$
ν	kinematic viscosity, m^2/s
ρ_1	epilimnion or top fluid density, kg/m^3

ρ_2	hypolimnion or bottom fluid density, kg/m^3
ρ_r	release fluid density, kg/m^3
σ_k	Schmidt number for k
σ_ε	Schmidt number for ε
σ_{sc}	Schmidt number
τ	Turbulent stress tensor, N/m^2
ϕ	general dependent variable or swirl vane angle

CHAPTER I

INTRODUCTION

1.1 The Need for Numerical Prediction

Traditional methods of design involve a great deal of experiments, time, and expense. This is true for most of the design situations, where an engineer has to seek an optimum path between irreconcilable alternatives. The general aim of most investigations is to provide information which is useful to designers by "characterizing" or "modeling" certain features of the phenomenon in question. Investigations may be theoretical or experimental; the two approaches are complimentary. Up to now designers have relied heavily on the experimental approach. The mathematical modeling approach is now finding favor and being used to supplement existing design procedures. Also, empirical trial-and-error design procedures are becoming increasingly more expensive. Hence, the role of numerical modeling becomes very important where realistic questions of design, sizing, inlet and outlet boundaries can be investigated more easily, quickly, and economically.

Numerical procedures for predicting three-dimensional turbulent reacting and nonreacting flows consist of two main parts: mathematical models of physical processes (turbulence, species diffusion, chemical reactions, and two-phase effects) and computer programs for solving the resulting differential equations. In deciding on and justifying the use of a particular procedure, the user raises some questions: how much

time is needed to understand the code and use it to obtain a solution? Does it predict with sufficient accuracy? what is the cost?

Existing three-dimensional flowfields prediction codes, although incorporating many complexities and being efficient in their solution algorithms, present a major struggle to the code user who is faced with the task of understanding, amending, and utilizing the available codes (1).

There is a need for simplified techniques for persons with little or no experience in computational fluid dynamics CFD, and into which user-oriented complexities can be easily added. Such techniques are of great practical importance to experimental researchers who can use them to supplement their experiments and broaden their investigations by using the ability of the numerical model to adapt easily, qucikly and economically to different conditions and boundary conditions and boundary configurations.

1.2 General Research Goal

The goals of the present work are:

1. To develop simple computing codes that are easy to use and yet accurate enough for a wide variety of fluid dynamics problems.

2. Codes into which user-oriented complexities can be easily added.

3. The variety of fluid dynamics applications of the codes would include two and three-dimensional turbulent flowfields, buoyancy effects and species diffusion.

1.3 Techniques to be Used

This thesis develops a method which, in a series of computer programs of approximately 700 statements each, solves 2-D and fully 3-D time-dependent turbulent flow equations in cartesian or cylindrical coordinates. Turbulence is to be simulated by means of the two-equation k-ε turbulence model (2), and species diffusion and buoyancy effects are to be properly simulated. The computational method is based on the transient 2-D Los Alamos SOLA prediction technique (3) (for laminar flows), which is a finite difference scheme based on the marker and cell method.

Consideration is given here to a primitive, pressure-velocity variable, finite difference code which will be developed to predict turbulent 3-D transient flows. The techniques to be developed here incorporate the following:

1. A finite difference procedure in which the dependent variables are the velocity components and pressure, formulated in cartesian and cylindrical coordinates.

2. The pressure is deduced from the continuity equation and the latest velocity field, using a guess-and-correct procedure for the latter.

3. The procedure incorporates displaced grids for the three velocity components, which are placed between the nodes where pressure and other variables are stored.

4. The codes are designed for persons with little or no experience in CFD with the purpose of demonstrating that many useful and difficult problems can be solved without restoring to large, complicated computer programs. As specific complexities can be added easily, the

codes provide a basis for developing many new numerical capabilities.

5. The codes incorporate the two-equation k-ε turbulence model (4).

1.4 Applications

In order to demonstrate the capability of the developed codes, several applications are considered. These applications include laminar and turbulent, two-dimensional and three-dimensional flow problems in cartesion and cylindrical coordinates. The applications to be considered are as follows:

1. Local destratifications of reservoirs: this application consists of a low-energy axial flow propeller placed just below the surface so as to provide a downward directed jet of fluid and thereby locally mix reservoirs near the release structure of the dam.

2. Deflected turbulent jet: this application arises when a turbulent jet enters normally into a uniform steady cross-flow.

3. Gas turbine combustor: this application involves modeling nonreacting turbulent flow in a 60 degree segment of the combustor.

Further details, background and problem schematics for these applications are given later in Chapter V.

CHAPTER II

LITERATURE REVIEW

Numerical simulation of fluid dynamics problems concentrates on the physical process of interest and closely resembles a physical experiment rather than a theoretical solution. A numerical analyst sets up his equations and boundary conditions, runs his program and then analyzes the computed results. This is very similar to what an experiment does in terms of (experimental) set up, runs and results analysis. Moreover, actual discovery (5) of physical phenomena is possible by using numerical experiments. For example, Campbell and Mueller (6) discovered the phenomenon of subsonic ramp-induced separation numerically before verifying it experimentally in the wind tunnel.

Computational fluid dynamics has progressed remarkably in the last two decades. This can be attributed mainly to the modern large-scale computers introduced in this period such as CDC 3600 and IBM 4794 in the 1960's and CDC 6000 and 7000 series and IBM 360 and 370 series in the 1970's.

The first two sections of this review discuss the background to time-dependent and steady state simulation of incompressible flow in cartesion and cylindrical coordinates. The third section surveys irregular boundary approaches. The turbulence modeling review is presented in the final section.

2.1 Time-Dependent Schemes

Time-dependent finite difference codes usually use an explicit solution procedure utilizing a time-marching process. Such techniques can be used to model 2-D and 3-D flow problems. An extensive review and discussion of these techniques and their applications can be found in textbooks by Roache (5) and Chow (7).

2.1.1 Two-Dimensional Simulation

Significant prediction procedures of two-dimensional laminar fluid flow were first presented by the Los Alamos group in 1965 (8, 9). The concept of numerical simulation or computer experiment was then introduced for the first time. Moreover, it was the first time the famous Marker and Cell (MAC) method was introduced. The method used a primitive variable explicit solution procedure for laminar transient flows simulation.

The MAC technique requires the simultaneous solution of the difference equations of pressure and velocities (i.e. Poissan equation for pressure and conservation of momentum equation). The Marker particles introduced in the incompressible flow calculation are only for the purpose of indicating fluid configurations especially in free surface flows. Other developments based on the MAC method continued with the Los Alamos group being the major contributor. The SMAC method (10) is somewhat a simplified MAC method in which pressure is not solved, but the continuity equation is satisfied directly by solving the Poissan equation for the potential function. An implicit continuous-fluid Eulerian (ICE) technique has been used to model fluid flow for all Mach numbers

from zero (incompressible limit) to infinity (hypersonic limit) (11). The method utilizes an arbitrary equation of state and the fully viscous stress tensor. A further development of the same method (ICE) included a multiphase fluid flow simulation (12, 13). An arbitrary Lagrangian-Eulerian technique for all flow speeds (14) was introduced using a finite-difference mesh with vertices that may be moved with the fluid (Lagrangian), be held fixed (Eulerian), or be moved in any other prescribed manner. The method has also been developed for multifluid flow calculations at all Mach numbers (15).

A program for multicomponent chemically-reactive flows at all flow speeds RICE (16) was also introduced by Los Alamos. Another technique, the original SOLA code (3), was designed for persons with little or no experience in computational fluid dynamics.

The SOLA technique involves an iterative process for adjusting cell pressure and velocities to satisfy the incompressibility condition instead of solving Poisson's equation for pressure. The simplified SOLA technique includes the basic algorithm SOLA for confined flows and SOLA-SURF, a modified SOLA for modeling free or curved rigid surface boundaries. The code coordinates were easy to change from two-dimensional cartesian to cylindrical (axisymmetric) coordinates. SOLA-ICE (17) a version of the SOLA code for transient compressible fluid flow includes the Implicit Continuous-Fluid Eulerian (ICE) technique.

Another development based on the concept of a fractional volume of fluid (VOF) was introduced as SOLA-VOF (18). The code was designed for treating complicated free boundary configurations with more efficiency and flexibility. The code used a variable mesh version of the SOLA code SOLA-VM.

Recent developments by the author based on the SOLA Code include species diffusion and buoyancy for stratified flow simulation (19-21), and swirl flow modeling (22).

2.1.2 Three-Dimensional Simulation

The first three-dimensional version of the Marker and Cell method was introduced in 1972 (23). The method used an iterative process to modify the cell pressure and velocity to satisfy the continuity requirement. The code also included thermal buoyancy as an option, using the Boussinesq approximation, and solving the energy equation for temperature.

A three-dimensional code with an Eulerian computing mesh for fluid flows at all speeds (25) was introduced in 1975. The method was used to model subsonic (incompressible) and hypersonic (compressible) flows.

Recent development by the author also based on the MAC method includes species diffusion and buoyancy for stratified flow simulation (26). The above mentioned 3-D codes were used to simulate transient laminar imcompressible flow in cartesian coordinates. They did not, however, address turbulent flow situations. A more recent development by the atuhor did include turbulent flow simulation, using a two-equation k-ε turbulence model with buoyancy and species diffusion (27). Applications of the code were presented for a turbulent deflected jet and local destratification of reservoires.

2.2 Steady-State Schemes

Steady-state finite difference codes usually use an implicit solution procedure. Applications of the steady-state schemes will be

discussed for 2-D and 3-D flow situations. An excellent review and discussion of the steady-state schemes and their applications can be found in textbooks by Patankar (28) and Roache (5).

2.2.1 Two-Dimensional Simulation

There are two types of steady flow problems; parabolic (boundary layer type with prominent directions, processing one coordinate direction with first-order derivative only) and elliptic (recirculating type with upstream influence, possessing second-order derivatives in all directions). An example (29) of sophisticated marching method for 2-D boundary layer flows involves one-dimensional storage, automatic expanding grid using nondimensional stream function ψ instead of r as radial coordinate, and implicit solution procedure.

Relaxation methods for 2-D recirculating flows involve two-dimensional storage and often implicit solution procedures. Examples of pertinent reference include the work of Imperial College, London, on stream function ψ-ω (30) and primitive pressure-velocity p-u-v (31) formulations.

2.2.2 Three-Dimensional Simulation

Three-dimensional parabolic boundary layer flows may be solved by marching methods which often use two-dimensional storage, primitive formulation and explicit or semi-implicit solution procedure (32). Sample calculations concern fuel being injected into a furnace of rectangular cross section, resulting in a diffusion flame (32).

Relaxation methods are used for three-dimensional elliptic recirculating flows. Typically, they make use of three-dimensional storage,

primitive formulation, and an implicit solution procedure (34-43). Some of the most interesting practical combustion phenomena are in this category including furnaces, combustion chambers, diffusers, compressor cascades, and automotive reciprocating engines. Flow patterns and heat transfer inside a furnace were investigated (34) with the same mathematical model - a relaxation method based on the original TEACH code (31). A typical problem is the flow and reaction inside a segment of an annular combustion chamber. This type of flow is steady but fully 3-D, with axial recirculation due in part to the laterally induced additional air supply. In such problems a single-step combustion model is usually used (35-37), 39-41). Methods using a three-dimensional spray combustion model for two-phase reacting flows in gas turbine combustors based on the same procedure have also been reported (42, 43).

2.3 Irregular Boundary Schemes

Several methods for flow problems with irregular boundaries exist. Finite-element methods which use a triangular cell-mesh have the advantage of fitting irregular boundaries with relative ease. The finite-element method has been applied to viscous laminar flow problems in two-dimensional (44, 45) and three-dimensional (46) configurations.

Independent variable transformation techniques are considered as an accurate approach for numerical presentation of irregular boundaries (47). These methods map an irregular physical domain onto a transformed rectangular domain through coordinate transformation equations which are either directly specified (analytical transformation) (48-51), or effected by finite difference solution of two boundary value problems (numerical transformation) (52-56). Such techniques were modified for simula-

ing 3-D flow problems with irregular boundaries using a conformal mapping technique for computation of steady three-dimensional supersonic flows (57).

Numerical transformation method were also used to model three-dimensional flow. Such applications include using orthogonal curvilinear coordinates to model flow in an accelerating rectangular 90 degree elbow (58). Nonorthogral coordinates were also widely used for three-dimensional flow problems, such as compressible potential flow around arbitrary geometries (59), and incompressible flow in ducts of arbitrarily varying rectangular or polar cross sections (60). Although these methods (especially 3-D techniques) are suitable for modeling flow problems with irregular boundaries, most of them did not address turbulent flow simulation because of the difficulties involved. Such difficulties involve variables positioning and simulation of turbulence and its boundary condition in complex grid forms.

These techniques, however, are not going to be used in the present study.

2.4 Turbulence Simulation

Fluid flow problems of practical importance are almost always turbulent. The turbulent motion is highly random, three-dimensional and time-dependent. Numerical solutions for three-dimensional time-dependent flows are well advanced. However, in spite of these advances, there is no exact method for turbulent flow modeling. The reason is that current computer storage and speed are not sufficient to solve the full equations for turbulent flow, resolving very small eddies when the flow domain exceeds by many orders of magnitude the size of these

eddies (4).

Hence, the only economically feasible approach is to solve the statistically-averaged equations governing the time-mean flow. The averaging is carried out over all the turbulent motions in such a way that the averaging time is long compared with the time scale of the large-scale motion.

This approach introduces turbulent stress terms, turbulent viscosity μ_t, and eddy viscosity ν_t with a turbulence model to determine turbulent and scalar transport terms. An excellent review of turbulence models and their applications can be found in Refs. 2 and 61.

2.4.1 No-Equation Models

One of the first attempts to model turbulence was proposed by Prandtl (62). The Prandtl mixing length model assumes that eddy viscosity is proportional to mean flow fluctuating turbulent velocity V_t and a mixing length ℓ_m. Considering thin shear layers only V_t is set equal to the local mean velocity gradiant times the mixing length:

$$V_t = \ell_m \left|\frac{\partial u}{\partial y}\right| \tag{2.1}$$

Hence, the eddy viscosity can be expressed as:

$$\nu_t = \ell_m^2 \left|\frac{\partial u}{\partial y}\right| \tag{2.2}$$

The mixing length parameter ℓ_m can be specified by simple empirical formulae in many simple flow situations (63, 64). For example in a round free jet $\ell_m = .075\ \delta$ where δ is the local mixing layer width, that is δ = width of 1/2 jet. This implies that the mixing length model lacks the universality of the empirical input. Moreover, based on its

implied assumption that turbulence is in local equilibrium, the model ignores the influence of turbulence production. Another obvious disadvantage is that whenever the velocity gradiant is zero, the model predicts the eddy viscosity to be zero. This is a clear indication that the mixing length model neglects the transport and history effect of turbulence. Other no-equation models are described in Ref. 2.

2.4.2 One-Equation Models

Models that account for transport and history effects of turbulence were developed in order to overcome the main disadvantage of the mixing length model. A transport equation for a velocity scale such as $\sqrt{k}$ where k is the kinetic energy of turbulence motion per unit and a measure of the itensity of the turbulent fluctuations and defined as:

$$k = \frac{1}{2}\left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}\right) \tag{2.3}$$

The turbulence energy dissipation rate per unit volume ε is given by:

$$\varepsilon = C_D \, k^{3/2}/\ell \tag{2.4}$$

where ℓ is the length scale and C_D is an empirical constant. This is based on the assumption that ε is governed by the large-scale turbulent motion characterized by velocity scale $\sqrt{k}$ and ℓ. The eddy viscosity as a function of the velocity scale and length scale is given by:

$$\nu_t = C_\mu \sqrt{k} \; \ell \tag{2.5}$$

where C_μ is an empirical constant.

This relation is known as the Kolmogorov-Prandtl expression (65, 66). The length scale ℓ is defined by empirical relations similar to those of the mixing length ℓ_m. These empirical relations work well for simple shear layers (67), but it is not possible to prescribe them in complex flows. Therefore, the one-equation model was limited mainly to shear-layer flows. However, for strongly accelerating boundary layer the model gave fairly poor predictions (68).

2.4.3 Two-Equation Models

A transport equation for the length scale was suggested to accompany the k-equation and give the so-called k-ℓ turbulence model. A length scale equation is not necessary in the k-ℓ model; any combination of the form $z = k^m \ell^n$, where m and n are constants will suffice. Two differential equations are solved for k and z. Other combinations with k such as the k-w model has been proposed; but the k-ε model became the most popular, mainly because the ε-equation requires no extra terms at the wall. The k-ε model uses the Kolmogorov-Prandtl relation which relates the eddy viscosity to k and ε as:

$$\nu_t = C_\mu \, k^2/\varepsilon \tag{2.6}$$

The model contains five empirical constants which will be discussed with the governing equations in Chapter III. The model is not applicable in the near-wall region (viscous layer) and, hence, the so-called wall-function approach is used to account for the near-wall viscous effect. The k-ε model has been widely tested and used to simulate free shear layers (69), wall boundary layers and duct flows. Ref. 69 presented predictions of the velocity profile development in mixing layer at the

end of a splitter plate using the mixing length model, one-equation k-ε model and the two-equation k-ε model. When comparing predictions made using the three models, the velocity profiles were best predicted by the k-ε model. Calculations of wall boundary layers under the influence of free-stream turbulence showed that the friction factor was well predicted using the k-ε model (70).

Most confined separating flows were successfully predicted using the k-ε model. Predictions of unconfined separating flows, however, were not always satisfactory. The k-ε model also fails to predict confined strongly swirling flows (71). Further applications of the k-ε model to a variety of flow problems are presented in Refs. (63 and 69). Attempts to include buoyancy effect in the k-ε turbulence model are discussed in Chapter III.

2.4.4 Multi-Equation Models

In the k-ε model, the eddy viscosity and diffusivity are assumed to be isotropic. This assumption is not valid for all flow situations, for instance in complex flows eddy viscosity and diffusivity depend on the stress or the flux component considered. Moreover, the k-ε model assumes that the local state of turbulence can be characterized by one velocity scale $\sqrt{k}$ and that the individual stresses $\overline{u_j' u_j'}$ can be related to that scale, dropping the primes from here onward. Hence, models using transport equations for $\overline{u_j u_j}$ and $\overline{u_j \phi}$ were proposed to account for the different development of the individual stresses and scalar fluxes. An extensive discussion of such models and their application can be found in Refs. (71- 74). The suggested stress equation model assumes local isotropy so that the dissipation of the three

normal stress terms is the same and thus 2/3 of the total dissipation. The stress/flux-equation model is often called second-order scheme in the literature. The following transport equations for $\overline{u_i u_j}$ and $\overline{u_i \phi}$ were proposed in Refs. (72 and 73) as one of the stress-equation models. Other models somewhat different from the one introduced below are discussed in Refs. (74 and 75).

$$\underbrace{\frac{\partial \overline{u_i u_j}}{\partial t}}_{\text{rate of change}} + \underbrace{U_\ell \frac{\partial \overline{u_i u_j}}{\partial x_\ell}}_{\text{convective transport}} = \underbrace{c_s \frac{\partial}{\partial x_\ell}\left(\frac{k}{\varepsilon}\overline{u_k u_\ell}\frac{\partial \overline{u_i u_j}}{\partial x_k}\right)}_{\text{diffusive transport}} - \underbrace{\overline{u_i u_\ell}\frac{\partial U_j}{\partial x_\ell} - \overline{u_j u_\ell}\frac{\partial U_i}{\partial x_\ell}}_{P_{ij} = \text{stress production}}$$

$$\underbrace{-c_t \frac{\varepsilon}{k}\left(u_i u_j - \frac{2}{3}\delta_{ij}k\right) - c_2\left(P_{ij} - \frac{2}{3}\delta_{ij}P\right) - c_3\left(G_{ij} - \frac{2}{3}\delta_{ij}G\right)}_{\text{pressure strain}}$$

$$\underbrace{-\beta\left(g_t u_j \phi + g_j u_i \phi\right)}_{G_{ij} = \text{buoyancy production}} \underbrace{- \frac{2}{3}\varepsilon\delta_{ij}}_{\text{viscous dissipation}} \qquad (2.7)$$

$$\underbrace{\frac{\partial \overline{u_i \phi}}{\partial t}}_{\text{rate of change}} + \underbrace{U_t \frac{\partial u_i \phi}{\partial x_\ell}}_{\text{convective transport}} = \underbrace{c_{s\phi}\frac{\partial}{\partial x_\ell}\left(\frac{k}{\varepsilon}\overline{u_k u_\ell}\frac{\partial \overline{u_i \phi}}{\partial x_k}\right)}_{\text{diffusive transport}} - \underbrace{\overline{u_i u_j}\frac{\partial \phi}{\partial x_j} - \overline{u_j \phi}\frac{\partial U_i}{\partial x_j}}_{\text{mean-flow production}}$$

$$\underbrace{-\beta g_i \overline{\phi^2}}_{\text{buoyancy production}} - \underbrace{c_{1\phi}\frac{\varepsilon}{k}u_i\phi + c_{2\phi}\overline{u_\ell \phi}\frac{\partial U_i}{\partial x_\ell} + c_{3\phi}g_i\overline{\phi^2}}_{\text{pressure scrambling}} \qquad (2.8)$$

The buoyancy terms in the two equations simulate the buoyancy production and its effect on turbulance generation and dissipation. Although shear-stress models are supposed to account for the shortcoming of the k-ε model, they also need wall corrections in local-equilibrium shear layers. In this type of shear layers the rate of change, convection and diffusion terms in Equation (2.7) are absent and the ratios of the individual stresses $\overline{u_i' u_j'}$ to one another are determined by the pressure-strain term only. Flow in an asymmetric channel with one of the walls roughened so that the shear stress ratio between the walls is 5:1 was predicted (74). Predictions showed that shear stress - unlike predictions made with eddy viscosity models - did not vanish at the maximum velocity location. Another application of the stress-equation model was the simulation of a fully developed flow in a square duct where gradients of Reynolds stresses cause a turbulence-driven secondary motion. This type of motion cannot be predicted with an eddy-viscosity model such as the k-ε model. The stress-equation model was capable of simulating this type of motion (75), but not without modification to the pressure-strain model in Equation (2.7).

A stress-equation model using differential transport equations for individual turbulent stresses and fluxes consists of a large number of these equations which makes the model costly and difficult to use. In order to make the model application practical and simple, the so-called algebraic stress/flux models have been developed by reducing the differential equations into algebraic expressions (77, 78). The simplest approximation was done by neglecting the convection and diffusion terms. Another approximation based on the assumption that the transport of

$\overline{u_i' u_j'}$ is proportional to the transport of k was suggested (77). Such an approximation yields algebraic expressions for $\overline{u_i' u_j'}$ and $\overline{u_i' \phi'}$ from the original model's differential equations. This approximation requires the calculation of the k and ε equations because they appear in the expressions and, therefore, considered as an extended k-ε model.

The algebraic stress/flux model is applicable only when the transport of $\overline{u_i' u_j'}$ and $\overline{u_i' \phi'}$ is not very important. They are not also applicable to flows with counter-gradient diffusion such as boundary layers. However, in modeling flows of practical engineering relevance success has been claimed over the k-ε model. One such flow, a plane wall jet in stagnant surroundings, was simulated using the k-ε model and the algebraic stress model (78). The former model overpredicted the jet spreading rate by 30% where the latter (algebraic stress model) predicted it correctly. This was not, however, without wall correction to the pressure-strain model.

2.5 Closure

This literature survey was given to provide a general background about the different numerical modeling possibilities. State of the art numerical methods are surveyed and discussed along other areas of the on-going research activities.

The present work emphasizes: the transient SOLA code (3-D in cartesian and cylinderical), turbulent flows (using the k-ε turbulence model), species diffusion and buoyancy effects. Several applications of the developed codes are considered in order to demonstrate their capability.

CHAPTER III

THEORETICAL MODEL

A typical simulation of the turbulent three-dimensional cartesian and cylindrical versions technique is discussed in this chapter. Other problems such as laminar or two-dimensional simulations can be handled as a simplified version of the developed technique.

3.1 Governing Equations

The turbulent Reynolds equations for conservation of mass, momentum, turbulent kinetic energy, and its dissipation rate are taken in the conservative form for two different coordinate systems. Two sets of governing differential equations are presented in this section: The first set is expressed in three-dimensional cartesian coordinates with species diffusion and buoyancy forces included.

These equations are employed in modeling one of the applications (local destratification of lakes) which will be discussed in the forthcoming chapters. The equations for incompressible stratified fluid flow are taken in the common form (26):

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x_i}(u_i \phi) = \frac{1}{\rho_l}\left[\frac{\partial}{\partial x_i}\left(\frac{\mu_{eff}}{\sigma_\phi}\frac{\partial \phi}{\partial x_i}\right) + s_\phi\right] \qquad (3.1)$$

where ϕ stands for any of the dependent variables u, v, w, m_1, k, and ε. When $\phi = 1$ the above equation represents the conservation of mass (continuity) equation. The first three variables u, v, and w are time

mean velocity components u_i in x_i directions x, y, and z , respectively. A buoyancy term occurs in the vertical z-direction equation (26). The local mass fraction m_1 of fluid having a density of ρ_1 allows the local density to be found using

$$m_1 + m_2 = 1$$

$$\rho = m_1 \rho_1 + m_2 \rho_2 \tag{3.2}$$

Density changes are slight and allow incompressible equations to be used, with the addition of an upward buoyancy force.

The second set of equations are expressed in three dimensional cylindrical polar system of coordinates, and can be taken in the common form:

$$\frac{\partial \phi}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(\rho r u \phi) + \frac{1}{r}\frac{\partial}{\partial \theta}(\rho w \phi) + \frac{\partial}{\partial z}(\rho v \phi)$$

$$= \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\mu_{eff}}{\sigma_\phi}\frac{\partial \phi}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{\mu_{eff}}{\sigma_\phi}\frac{\partial \phi}{r \partial \phi}\right) + \frac{\partial}{\partial x}\left(\frac{\mu_{eff}}{\sigma_\phi}\frac{\partial \phi}{\partial z}\right) \tag{3.3}$$

$$+ S_\phi$$

where ϕ stands for any of the dependent variables, u, v, w, m_1, k, and ε, and $\phi = 1$ for the continuity equation. The time-mean velocity components u, v, and w are in the r, z, and θ directions, respectively. The turbulent viscosity is calculated from the standard k-ε turbulence model (4).

$$\mu_t = \rho C_\mu k^2/\varepsilon \ ,$$

$$\mu_{eff} = \mu_t + \mu \tag{3.4}$$

where k is the turbulence energy and $\varepsilon [= k^{3/2}/\ell]$ is the energy

dissipation rate, both of these being obtained from solution of their respective partial differential equations. The equations differ primarily in their Schmidt numbers σ_ϕ and final source terms S_ϕ, as indicated in Table I for cartesian coordinates and Table II for cylindrical coordinates. Turbulence and other constants appearing in the tables are given the usual recommended values: $C_D = 1.0$, $C_\mu = 0.09$, $C_1 = 1.44$, and $C_2 = 1.92$. These values have been used in a wide variety of turbulent flow situations and exhibited good predictive capability (4).

3.2 The Grid System

The three-dimensional grid systems used for cartesian and cylindrical coordinates are divided into cell divisions with uniform spacings. The solution domain is complemented by a layer of cells on all sides, so as to allow easy simulation of the required boundary conditions BCs. These fictitious cells increase the number of cells in each direction. Figure 1 illustrates the total mesh arrangement for the cartesian coordinate system and Fig. 2 represents the cylindrical coordinates mesh arrangement. Figure 1 also represents the physical problem of the local destratification application, which contains a downward flowing jet of fluid from a propeller. Initially, two fluids occupy positions above and below the interface as shown so that their mass fractions are $m_1 = 1$ and $m_2 = 0$ (for $z > Z_T$, the height of the interface) and vice versa. Also shown is how the inlet and outlet flows are handled. The exit release flow is via a rectangular area in the release structrue with the flow rate specified a priori. Figure 2 represents the physical problem of a can combustor

application, which involves a swirling jet penetration simulation in 60 degrees segment of a gas turbine combustor (1, 40).

Figures 3 and 4 portray a single cell in cartesian and cylindrical coordinates respectively, and show the location of each field variable p, u, v, w, and m_1, relative to this (I,J,K)-cell. The pressure and mass fraction variables are located at the center of each cell and the velocities are on the faces as follows:

u_{ijk} = x-direction velocity located at center of right face of the (I,J,K)-cell touching the (I+1, J, K)-cell.

v_{ijk} = y-direction velocity located at center of top face of the (I,J,K)-cell touching the (I, J+1, K)-cell.

w_{ijk} = z-direction velocity located at center of near face of the (I,J,K)-cell touching the (I, J, K+1)-cell.

Thus normal velocities lie directly on the physical boundaries of the solution domain, while the tangential velocities and pressure are displaced half a cell interval inside the flowfield. In this way the exterior fictitious cells are particularly convenient when applying the boundary conditions.

3.3 The Boundary Conditions

Finite difference equations simulating the problem are set up and solved by way of a time-march process applied to cells within the flow domain of interest. Cells touching the boundary thus utilize the value on the boundary (in the case of a normal velocity) or values half a cell distant beyond the boundary (in the case of tangential velocities).

Specifications of normal velocities at an outflow boundary often pose a problem, as it can have detrimental upstream influence. One might merely impose the zero-normal gradient or continuative condition and set these values equal to their immediate upstream values (3). When primary interest is being focused on the final steady-state solution, it has been found (21) that a suitable constant may be added to each such extrapolated value, with advantage to the rapidity of convergence. This constant value is chosen so as to make the total outlet flux equal to the total inlet flux, thus ensuring the requirement of a macroscopic mass balance. To illustrate the technique further, consider the application of a deflected jet in a cross flow as an example. The total inlet flux is represented by the sum of the cross flow and the deflected jet inlet fluxes. The outlet flux is calculated using the current values of the outlet velocities. Then the constant additive (UINC) is calculated using the inlet and outlet fluxes as:

$$UINC = (FLUXIN - FLUXOU)/AREA$$

where AREA represents the outlet area. This constant is then added to each extrapolated axial velocity (parallel flow) at the outflow boundary. Outlet boundary specification is imposed only after each time-step and not after each pass through the mesh during the pressure iteration.

At planes of symmetry, the usual zero normal velocity and free-slip tangential velocity specifications are applicable.

Inlet boundary specification is imposed after each time-step and

during the pressure iteration using prespecified inlet values.

The inlet value for the turbulent kinetic energy k is taken as $0.03u^2$ and the dissipation ε is given by

$$\varepsilon = k^{3/2}/\ell \tag{3.5}$$

where ℓ is the length scale of turbulence, taken typically as 3 percent of the characteristic size of the passage.

At a plane of symmetry both k and ε are given the zero-gradient condition. At a wall, k is treated similarly but values are specified just inside the computational field at a point P in terms of the local value of k via (4)

$$\varepsilon_P = \frac{C_\mu^{3/4} k_P^{3/2}}{k.y_P} \tag{3.6}$$

Wall shear stress is evaluated via the modified log-law with the assumption of a constant shear region near the wall.

3.4 Wall Functions

Wall functions are used to avoid the need for detailed calculations near the wall. The one-dimensional Couette flow characterization (diffusion perpendicular to the wall is dominant)is used to link velocities, k and ε on the wall to those in the logarithmic region. The wall functions are used in the momentum equations and k-generation terms of near-wall points (2,4).

3.4.1 Northern Wall

Wall function formulation in three-dimensional flow requires that the tangential velocities u and w are expressed in terms of total

tangential velocity near the northern top wall (32) $V = (u^2 + w^2)^{1/2}$ for cartesian and $V = (v^2 + w^2)^{1/2}$ for cylindrical coordinates. This velocity is correlated by the universal velocity profile

$$V^+ = (1/K) \ln (EY^+) \tag{3.7}$$

where K and E are constans. The dimensionless total velocity V^+ and total distance Y^+ are obtained by nondimensionalizing with respect to the total shear velocity $(\tau_t/\rho)^{1/2}$. Thus the total tangential shear stress at the north wall $\tau_t = (\tau^2_{yz} + \tau^2_{yx})^{1/2}$ for cartesian coordinates and $\tau_t = (\tau^2_{rz} + \tau^2_{\theta})^{1/2}$ for cylindrical coordinates is obtained from

$$V(\tau_k\rho)^{1/2}/\tau_t = (1/K) \ln [Ey (\tau_k/\rho)^{1/2}/\nu] \tag{3.8}$$

where τ_k is an approximation for τ_t very near the wall. τ_k is formulated by deleting the convection and diffusion of turbulent kinetic energy from the k-transport equation (where they are considered negligible in this region (2)) and invoking isotropic viscosity μ_{eff} leads to

$$\tau_k = (C_D C_\mu)^{1/2} \rho k \tag{3.9}$$

and thus from Eqn. (3.8) and (3.9)

$$\tau_t = -V_P K \rho C_\mu^{1/4} C^{1/4} k_P / \ln(E Y_P^+) \tag{3.10}$$

where the negative sign is inserted since τ_t and V_P must have opposite directions, suffix P refers to the conditions at the near-wall grid node. K and E are constants possessing the values 0.4186 and 9.793, respectively. The total tangential shear stress is then substituted

in the k-generation term (near the wall) as follows:

$$G = 2\,\mu_{eff}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right] + \tau_t^2/\mu_{eff} + \mu_{eff}\left[\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right]^2 \tag{3.11}$$

for cartesian coordinates and

$$G = 2\mu_{eff}\left[\left(\frac{\partial u}{\partial r}\right)^2 + \left(\frac{1}{r}\frac{\partial w}{\partial \theta} + \frac{u}{r}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right] + \tau_t^2/\mu_{eff} + \mu_{eff}\left[r\frac{\partial}{\partial r}\left(\frac{w}{r}\right) + \frac{1}{r}\frac{\partial u}{\partial \theta}\right]^2 \tag{3.12}$$

for cylindrical coordinates.

For the x-momentum equation the shear stress component of τ_t is given by

$$\tau_{yx} = \mu_{eff}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{3.13}$$

for cartesian coordinates and for z-momentum equation in cylindrical coordinates

$$\tau_{rz} = \mu_{eff}\left(\frac{\partial v}{\partial r} + \frac{\partial u}{\partial z}\right) \tag{3.14}$$

for cylindrical gradients $\frac{\partial v}{\partial x}$ and $\frac{\partial u}{\partial z}$ approach zero near the north wall (31). The above wall function formulation, Equation (3.10), is multiplied by the factor cos θ'(= u/V) to obtain τ_{yx} and cos θ'(= v/V) to obtain τ_{rz}, where θ' is the angle between the total tangential velocity vector near the wall and the axial velocity vector which is u in the cartesian coordinates and v in the cylindrical. The angle θ' is also considered constant near the wall (79). Thus the wall functions for the x-momentum equation in cartesian coordinate is

$$\tau_{yx} = [\, K \rho C_{\mu}^{1/4} C_D^{1/4} k_p^{1/2} / \ln (E Y_p^+)] \, u_p \qquad (3.15)$$

and for z-momentum equation in cylindrical coordinate the shear stress is

$$\tau_{rz} = [K\rho C_{\mu}^{1/4} C_D^{1/4} k_p^{1/2} / \ln (E Y_p^+)] \, v_p \qquad (3.16)$$

The quantities with subscript p are evaluated at the appropriate near-wall points as shown in Fig. 5(a) for cartesian and 5(b) for cylindrical coordinates. Then, for z-momentum equation in cartesian and θ-momentum equation in cylindrical coordiantes the shear stress along the north wall will be given by

$$\tau_{yz} = \mu_{eff} \frac{\partial w}{\partial y} \qquad (3.17)$$

for cartesian and

$$\tau_{r\theta} = \mu_{eff} \, r \frac{\partial}{\partial r} \left(\frac{w}{r}\right) \qquad (3.18)$$

for cylindrical coordinates. Applying the factor $\sin\theta'$ to the τ_t wall function yields

$$\tau_{r\theta} = \tau_{yz} = [K \rho \, C_{\mu}^{1/4} C_D^{1/4} k_p^{1/2} / \ln (E Y_p^+)] \, w_p \qquad (3.19)$$

The shear stress will be substituted in the momentum equations near the northern wall. For example, consider the x-momentum equation in cartesian coordinates:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} (u^2) + \frac{\partial}{\partial y} (uv) + \frac{\partial}{\partial z} (uw) = - \frac{\partial P}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \qquad (3.20)$$

In Equation (3.20) τ_{yx} will have a different value near the northern wall and therefore

$$\frac{\partial \tau_{yx}}{\partial y} = (\tau_{yxS} - \tau_{yxN})/\Delta y \qquad (3.21)$$

where the shear stress at the wall is subtracted from the shear stress inside the field adjacent to the wall. In this way τ_{yxN} will be the wall function formulating Equation (3.15).

3.4.2. Southern Wall

Wall functions along the southern wall are formulated similar to the northern wall formulation. Referring to Equation (3.21) of the shear stress τ_{yx} gradient, τ_{yxS} will be the wall function formulation at the southern wall where

$$\frac{\partial \tau_{yx}}{\partial y} = (\tau_{yxN} - \tau_{yxS})/\Delta y \qquad (3.22)$$

where τ_{yxS} is equal to τ_{yx} in Equation (3.15).

3.4.3 Western Wall

Wall functions along the west wall are similarly formulated. The total tangential velocity is now $V = (v^2 + w^2)^{1/2}$ for cartesian and $V = (u^2 + w^2)^{1/2}$ for cylindrical coordinates. Equation (3.10) for τ_t is obtained using the new total tangential velocity V for the k-generation term and substituted in a similar fashion to the northern and southern wall procedure. Momentum wall functions are also obtained for y-momentum and r-momentum equations for cartesian and cylindrical coordinates, respectively. Thus for the y-momentum equation

$$\tau_{xy} = [K \rho C_{\mu}^{1/4} C_D^{1/4} k_P^{1/2} / \ln (E Y_P^+)] v_P \quad (3.23)$$

and for r-momentum

$$\tau_{zr} = [K \rho C_{\mu}^{1/4} C_D^{1/4} k_P^{1/4} / \ln (E Y_P^+)] u_P \quad (3.24)$$

Then for the z-momentum and θ-momentum equations in cartesian and cylindrical coordinates respectively the shear stress will be

$$\tau_{xz} = \tau_{z\theta} = [K \rho C_{\mu}^{1/4} C_D^{1/4} k^{1/2} / \ln (E Y_P^+)] w_P \quad (3.25)$$

The quantities with subscript p are evaluated at the appropriate near-wall points. Then, these relations are substituted in the momentum equation near the western wall. Consider the z-momentum equation in cartesian coordinates for example

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial x} (wu) + \frac{\partial}{\partial y} (wv) + \frac{\partial}{\partial z} (w^2) = - \frac{\partial P}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \quad (3.26)$$

In Equation (25) τ_{xz} will have a different value near the western wall and therefore

$$\frac{\partial \tau_{xz}}{\partial x} = (\tau_{xzE} - \tau_{xzW}) / \Delta x \quad (3.27)$$

where τ_{xz_W} will equal to the wall function formulation (3.25).

3.4.4 Eastern Wall

Wall functions along the eastern wall are formulated similar to the western wall for the k-generation term. Refereing to Equation (3.27), τ_{xzE} will be at the wall and expressed as

$$\frac{\partial \tau_{xz}}{\partial x} = (\tau_{xzW} - \tau_{xzE}) / \Delta x \quad (3.28)$$

where τ_{xzE} is equal to τ_{xz} in Equation (3.25). The shear stress gradient at the wall in Equations (3.21, 3.22, 3.27 and 3.28) for north, south, west, and east walls respectively should always be negative (relative to the flow direction). To insure a negative shear gradient at the wall, the shear stress at the wall should be subtracted from the shear stress inside the field adjacent to the wall as shown in the above mentioned equations.

CHAPTER IV

THE SOLUTION PROCEDURE

Computations for one calculation cycle (time-step) can be summarized in three steps:

1. Computing the velocity at all internal points via application of momentum principles.
2. Adjusting the cell pressure and velocities iteratively to satisfy the continuity equation.
3. When convergence is achieved, the pressure and velocity values will be at the advanced time-level and can be used to start the calculation for the next time-step.

4.1 The Finite Difference Equations

Finite difference representations are required of the governing PDEs. The usual intuitive estimates of one-sided first-derivatives, centered first-derivatives and centered second-derivatives are used in representing the momentum equations. Superscripts n and (blank) are used to denote values at time-level t and t + Δt, respectively. Portrayed now are equations enabling one such forward time-step to be accomplished. Thus, starting from initial field values throughout the domain of interest, a time-march process is used so as to advance toward the final steady-state solution, which is usually of special interest as opposed to the en route calculations.

In Equations (3.1) and (3.3) the time-derivatives are approximated by forward one-sided derivatives; most spatial derivatives are approximated by central differences based on values at time-level t. Special techniques are required in computational fluid dynamics. For instance, in the representation of the convection terms, a certain amount of upstream differencing is required. The difference equations representing the PDEs may be written explicitly as:

$$u_{ijk} = u^n_{ijk} + \Delta t \left(\frac{1}{\Delta x}(p^n_{ijk} - p^n_{i+1,j,k}) + g_x - FUX - FUY - FUZ + VISX\right)$$

$$v_{ijk} = v^n_{ijk} + \Delta t\left(\frac{1}{\Delta y}(p^n_{ijk} - p^n_{i,j+1,k}) + g_y - FVX - FVY - FVZ + VISY\right)$$

$$w_{ijk} = w^n_{ijk} + \Delta t\left(\frac{1}{\Delta z}(p^n_{ijk} - p^n_{i,j,k+1}) + g_z - FWX - FWY - FWZ + VISZ\right) \qquad (4.1)$$

$$s_{ijk} = S^n_{ijk} + \Delta t\,(-FMX - FMY - FMZ + VIMX)$$

$$k_{ijk} = k^n_{ijk} + \Delta t\,(-FKX - FKY - FMZ + VISK)$$

$$\varepsilon_{ijk} = \varepsilon^n_{ijk} + t\,(-FEX - FEY - FEZ + VISE)$$

where $S_{i,j,k}$ is the symbol used for the mass fraction m_1 in the finite difference equation. The four terms on the right hand side of each equation FUX, FUY, etc., are shown in Table 1 for cartesian and Table 2 for cylindrical coordinate systems (Appendix A). The right hand side

terms represent convection, diffusion, and source terms in Equation (3.1) and (3.3). For the turbulence quantities k and ε, the source term linearization is handled in the manner recommended for always positive variables (28). In all six of these forward marching equations, donor cell differencing is used with the convection terms. In this way, a coefficient α takes constant value between 0 and 1, so giving the desired amount of upstream differencing (3). A value of 0 gives merely central differencing as in the original MAC code and numerical instability problems arise; a value of 1 gives the full upstream or donor cell for which, although less accurate, is stable provided among other things that the fluid is not allowed to pass through more than one cell in one time-step.

Although Equation (4.1) accomplishes one forward time-step based on coservation of momentum principles, the newly calculated velocities will not, in general, satisfy the continuity requirement, as expressed by the central finite difference form of the continuity equation in cartesian coordinates Equation (3.1):

$$\frac{1}{\Delta x}(u_{ijk} - u_{i-1,j,k}) + \frac{1}{\Delta y}(v_{ijk} - v_{i,j-1,k}) + \frac{1}{\Delta z}(w_{ijk} - w_{i,j,k-1}) = 0 \tag{4.2}$$

or from Equation (3.3) which gives the cylindrical version of the continuity equation:

$$\frac{1}{r_{ijk}(\Delta r)}(r_{ijk}\,u_{ijk} - r_{i-1,j,k}u_{i-1,j,k}) + \frac{1}{r_{ijk}(\Delta\theta)}(w_{ijk} - w_{i,j,k-1}) + \frac{1}{\Delta z}(v_{ijk} - v_{i,j-1,k}) = 0 \tag{4.3}$$

Terms here are evaluated at time-level $t + \Delta t$.

4.2 Iterative Procedures

This incompressibility condition is imposed by iteratively adjusting the cell pressure and velocities (3, 23, 26). That is, if the divergence D of a cell is positive (the left hand side of Equations (4.2) and (4.3) is positive), there is a net mass outflow from that cell. This is corrected by reducing the cell pressure. If the divergence is negative, an increase in cell pressure is appropriate.

First, consider the cartesian 3-D situation. When a cell pressure changes from p to p + Δp, the velocity components on the 6 faces of that cell change, given from a linear analysis from Equation (4.1), by the amounts:

$$
\begin{aligned}
u_{i,j,k} &= u_{i,j,k} + \Delta t\ \Delta p/\Delta x \\
u_{i-1,j,k} &= u_{i-1,j,k} - \Delta t\ \Delta p/\Delta x \\
v_{i,j,k} &= v_{i,j,k} + \Delta t\ \Delta p/\Delta y \\
v_{i,j-1,k} &= v_{i,j-1,k} - \Delta t\ \Delta p/\Delta y \\
w_{i,j,k} &= w_{i,j,k} + \Delta t\ \Delta p/\Delta z \\
w_{i,j,k-1} &= w_{i,j,k-1} - \Delta t\ \Delta p/\Delta z
\end{aligned}
\qquad (4.4)
$$

Substitution of these in equation (4.2) yields the amount of correction to p required as:

$$\Delta p = -D/(2\ \Delta t\ (1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2))$$

where D is the current (nonzero) value of the left hand side of equation (4.2). Pressure update iteration continues until the Ds of <u>all</u> the cells are less than some prescribed small positive quantity.

Equations in cylindrical coordinates are handled in a similar manner:

$$
\begin{aligned}
u_{ijk} &= u_{ijk} + \Delta t\ \Delta p\ /\Delta r \\
u_{i-1,j,k} &= u_{i-1,j,k} - \Delta t\ \Delta p/\Delta r \\
v_{ijk} &= v_{ijk} + \Delta t\ \Delta p/\ \Delta z \\
v_{i,j-1,k} &= v_{i,j-1,k} - \Delta t\ \Delta p/\Delta z \\
w_{ijk} &= w_{ijk} + \Delta t\ \Delta p/(r\Delta\theta) \\
w_{i,j,k-1} &= w_{i,j,k-1} - \Delta t\ \Delta p/(r\Delta\theta)
\end{aligned}
\qquad (4.6)
$$

Also, similarly, the amount of pressure correction required is:

$$\Delta p = -D/(2\Delta t\ (1/\Delta r^2 + 1/\Delta z^2 + 1/(r\Delta\theta)^2)) \qquad (4.7)$$

where D is the current nonzero value of the left hand side of Equation (4.3). The Equations (4.4) and (4.6) are applied with an over-relaxation factor ω between 1 and 2, a value near 1.8 is considered optimal in order to speed up the convergence of the pressure iteration process. After the continuity equation is sufficiently well satisfied, the values are accepted as new-time values and preparation for the next step of the time-march can begin.

In the lake destratification application, the diffusion equation (conservation of chemical species) is applied at each step of the time-march to the low-density fluid 1 (initially the top layer of fluid in the solution domain), and this enables the respective proportions of fluid 1 and fluid 2 (the high density fluid initially forming the bottom layer of fluid in the solution domain) to be calculated, and hence the density at all locations.

4.3 Convergence and Stability

Convergence to the steady-state solution is established by taking many forward time-steps. The choice of time increment must be restricted (for stability) in two ways. First, fluid should not pass through more than one cell in one time-step. Therefore, Δt must be less than (usually 0.25 to 0.33 times) the minimum cell transit time taken over all cells.

$$\Delta t < \min \{ \frac{\Delta x}{|u|} , \frac{\Delta y}{|v|} , \frac{\Delta z}{|w|} \} \tag{4.8}$$

Secondly, with a nonzero kinematic viscosity, momentum must not diffuse more than approximately one cell in one time-step, for which an estimate is:

$$\nu \Delta t < \frac{1}{2} \min \{\Delta x^2, \Delta y^2, \Delta z^2\} \tag{4.9}$$

A similar criterion is given in (23).

When the time-step is so restricted, the required amount of upstream (donor cell) differencing must be achieved by choosing a slightly larger than (typically 1.2 to 1.5 times) the largest of the right hand side members of:

$$1 \geq \alpha > \max \{ \frac{|u|\Delta t}{\Delta x} , \frac{|v|\Delta t}{\Delta y} , \frac{|w|\Delta t}{\Delta z} \} \tag{4.10}$$

where the maximum is taken over all cells. If α is chosen to be too large, stability is being achieved at the expense of the introduction of an unnecessarily large amount to diffusion-like truncation errors (called numerical smoothing).

CHAPTER V

APPLICATIONS AND DISCUSSION

In order to demonstrate the capability of the codes described in Chapter 3, several applications are considered. In each case involving turbulence simulation, predictions are given via the use of the recommended values (4) of the empirical constants in the turbulence model. Typically, a rather coarse grid system is used in the demonstrations, although clearly finer meshes are needed if accuracy is paramount. The three applications consist of fully three-dimensional problems. All applications will include laminar or turbulent non-reacting flow simulation.

5.1 Local Destratification of Reservoirs

The Garton pump consists of a low-energy axial flow propeller placed just below the surface so as to provide a downward directed jet of fluid and thereby locally mix reservoirs near the release structure of the dam. In this way high-quality epilimnion water is transported downward, so obtaining local destratification and improved release water quality in the vicinity of low-level release structures. The flowfield is fully three-dimensional and the solution procedure needs to include species diffusion and buoyancy forces. Figure 6 shows a schematic of the mixing so produced when there is no exist beneath the propeller. The jet or plume penetrates some distance below the

level of the thermocline. Figure 7 illustrates the practical application of localized mixing in the proximity of the release structure of a dam with low-level release gate. The flowfield so produced is fully three-dimensional. Earlier work (21) was restricted to this axisymmetric formulation about a vertical axis; and good results were portrayed for cases in which the propeller was close to the release structure. More recently, a three-dimensional version (26) has been developed using a constant eddy viscosity model, based on round turbulent free jet theory. Comparison of those predictions with the main source of experimental data (80) confirmed that the main dynamic effects are modeled better than the previous two-dimensional predictions. The present study extends the three-dimensional simulation to include turbulent mixing via the two-equation k-ε turbulence model (27).

5.1.1 Special Boundary Conditions

Figure 7 represents the physical problem, and contains a downward flowing jet of fluid from the propeller. Initially, two fluids occupy positions above and below the interface as shown so that their mass fractions are $m_1 = 1$ and $m_2 = 0$ (for $z > Z_T$, the height of the interface) and vice versa. A grid system of 7 x 5 x 9 internal cells (in x, y and z directions, respectively) is employed. (This corresponds to 10 x 8 x 12 grid lines). No slip boundary conditions with the wall function approach (in the turbulent prediction) are taken at the vertical dam wall and the horizontal floor of the reservoir. Free slip conditions are taken at the free surface which is also assumed to maintain a constant height. The propeller is positioned sufficiently far beneath the surface for surface effects to be neglected. Velocities are

specified a priori with flat profiles at the propeller and exit locations. Use is made of the plane of symmetry through the center of the propeller and the exit, free slip conditions are applied there. The available volume of the flow domain is very limited. To allow the outlet of release water without dramatically decreasing the fluid level, there is a compensating incoming flow which is distributed among the top and bottom layers in amounts equal to the epilimnetic and hypolimnetic water released. The peripheral inflow is at all elevations in such a way that eplimnetic water enters into the upper layer and does not distrub the density profile. The large area of the inflow allows it to be at a low velocity in order to avoid any disturbance to the flow-field. Similarly, hypolimnetic water is fed into the bottom layer.

5.1.2 Laminar Predictions

The ability to characterize the flowfield in the vicinity of low-level release structures during destratification is of prime importance in addressing questions of design and sizing of applicable propeller pumps. Comparison of results with hydraulic laboratory model data of Moon (80) shows that the numerical simulation of local destratification phenomena is a useful predictive tool. The dilution factor DF (release water quality) is found to be a function of turbulent viscosity μ_t, densimetric Froude number Frd, nondimensional metalimnion (interface) height Z_T^*, nondimensional flow rate Q*, and nondimensional propeller diameter D*, horizontal distance K*, and depth L*.

A standard base case is taken with parameters

μ_t/μ = 142

K* = 0.211

Frd = 1.6 and 2.0

Q* = 2.6

D* = 0.131

from which certain changes are made and individually assessed in the subsequent discussion. Figures 8-12 show the computed effect of variation of each of these parameters on the dilution factor DF, while the other parameters are maintained at their respective base values, with the parameters Z_T^* = 0.6 and L* = 0.211 held constant for all the figures. An indication of the effect of these two parameters on the axisymmetric flowfield were investigated (21). Appropriate experimental results are also shown where available, and it is to be observed that the main dynamic effects are modeled adequately and show the same trends as the available physical data. A constant turbulent viscosity is used in the predictions. Such a value is appropriate for a round turbulent free jet (64, 81). Hinze (81), for example, asserts that the turbulent viscosity in a round free jet is approximately constant and given by

$$\mu_t = 0.00196\ \rho(x + a)u_m \tag{5.1}$$

in terms of station maximum axial velocity u_m. In terms of jet initial velocity u_{mo} this can be written

$$\mu_t = 0.00196\ A\ \rho u_{mo} d \tag{5.2}$$

where A is a parameter 5.4 and 6.39. Our work in general uses the latter value in accordance with the early recommendation (82). For the particular propeller diameter and initial jet velocity used in

most of the subsequent predictions, this choice gives

$$\mu_t/\mu = 142 \qquad (5.3)$$

Figure 8 shows the effect of varying the magnitude of the turbulent viscosity μ_t on the dilution factor DF. Only a minor effect is to be seen indicating that results are not too susceptible to the particular choice. Since the computed effects were minor, it may be concluded that the precise modeling of turbulence is not crucial to obtaining good results. The present work also incorporates a constant turublent Schmidt number equal to unity used in the species equation. The result is a good similitude between temperature-stratified field tests, chemically-stratified laboratory tests, and numerical predictions.

Another question is the difference between axisymmetric problems on which the previous theory was based (21) and asymmetric problems on which the experimental data of Moon (80) were obtained, and with which those results were compared. Identical results could not be expected, but the work reported (21) showed that similar parametric trends were to be observed in both geometries. This would undoubtly not be the case if the geometry of the asymmetric case was no longer compact with the release gate laterally far away from the plume. With the present 3-D prediction procedure, this effect can be investigated. Figure 9 shows the effect of the nondimensional distance from the release structure to the propeller axis K*. The figure shows that the dilution factor is maximum when the propeller is next to the wall and it decreases gradually as the propeller is moved further away from the release structure. It can be inferred from this that the previous axisymmetric simulation tends to over-predict the release water quality

as compared with the practical case.

In Figure 10, the dilution factor DF is shown to be a strong function of the densimetric Froude number Frd in both predictive and laboratory data. The general result is that release water quality (fraction of epilimnetic (top) water in the exit stream) increases with higher values of the Froude number, such conditions being achieved with higher jet velocities from the propeller and/or a lower degree of stratification in the lake. The figure clearly shows the difference between the 2-D and 3-D models: the former over-predicting the release water quality and the latter giving more accurate predictions when compared to the experimental data. From the designer's view, the important result is the Froude number at which the dilution factor rises sharply, since that determines the velocity required at the propeller for the given conditions. Clearly, the 3-D simulation is superior to the 2-D simulation, and this Froude number is predicted better via the 3-D approach.

Figure 11 illustrates that the dilution factor DF is a strong function of the flow-rate ratio Q*. Generally, release water quality improves with the increase in the value of Q*, a condition achieved with low release rates and high propeller flow rates. The figure shows the relationship found in both hydraulic laboratory model and computer simulations. Again, the 2-D model over-predicts the dilution factor whereas the 3-D model shows realistic results, especially at higher values of Q*, which are better for obtaining higher water quality in the release flow.

The effect of changing the propeller diameter, but retaining the propeller flow rate Q* the same, is shown in Figure 12. Over-

prediction with the 2-D approach was present as expected. The 3-D approach, however, shows good agreement between predicted and experimental evidence of the effect of normalized diameter D* on dilution factor DF. These and other results show that a value of D* = 0.211 gives best results at this value of Q* = 2.6. At different values of Q*, different values of D* give optimum results (21).

Transient behavior of the flowfield is exhibited in the vector velocity plots and the three-dimensional representation of the vertical velocity at the thermocline (the interface). The plots are shown in a series of time-frames, using a time-interval of 2 seconds which is equivalent to 40 time-steps. This two-second time-interval may be comprehended better when it is considered that the jet velocity V_j = 0.2134 m/s and the propeller height above the reservoir bottom h = 0.42. With h/V_j = 1.968 s(the time required for the jet to reach the bottom). Eight plots are produced starting with the first time-step to the final steady state solution (which corresponds to 16 seconds or 320 time-steps).

Figure 13 shows the first four frames of the laminar vector velocity plots. In the first frame, the fluid near the reservoir bottom is being affected by the presence of the propeller after as little as one time-step. Fluid in the free shear layer surrounding the jet shows further movement in the second frame. This movement continues to increase slowly in the third frame. Very little change is shown in the last four frames in Figure 14. This indicates that the solution is approaching the final steady state solution rapidly. Velocity vectors are plotted at the vertical x-y plane which is the plane of symmetry in the flowfield.

The same trend is shown in the three-dimensional velocity plots at the thermocline (interface between the top and bottom fluid layers). These plots represent the vertical velocity v plotted as a function of x and z coordinates where the vertical coordinate is y. The plots (offering a bottom view of the vertical velocity at the interface) are shown in a series of time-frames, utilizing the same time-interval used for the vector velocity plots. The first four time-frames are displayed in Figure 15. The first frame exhibits initial jet plunge into the fluid showing the initial jet width or area. The second frame shows an increase of about 150 percent in the jet area indicating that the jet is spreading at the interface. As shown by the vector velocity plots, transient behavior in the last four frames, displayed in Figure 16 shows very little change. This is an indication of approaching the steady state solution. These three-dimensional velocity plots offer a view of the action generated by the propeller jet pentration in the fluid at the interface.

5.1.3 Turbulent Predictions

This section extends the three-dimensional laminar simulation to include turbulent mixing via the two-equations k-ε turbulence model.

The dilution factor DF (release water quality) is found to be a function of turbulent viscosity μ_t, densimetric Froude number Frd, non-dimensional thermocline height Z_T^*, nondimensional flow rate Q*, nondimensional propeller diameter D*, horizontal distance K*, and depth L*. Results are now discussed about the application of the present 3-D turbulent code to this problem with the following base values of the parameters: Frd = 2.0, D* = 0.131, K* = 0.211, L* = 0.211,

$Q^* = 2.6$, and $Z_T^* = 0.6$. These values are taken so as to represent the same conditions as the available experimental data (80).

The rate of progress of the transient computations toward the steady state solution is illustrated in Figure 17. Prediction of the dilution factor DF as a function of iteration time is shown for both a laminar and a turbulent flow with k-ε model. The latter approach is closer to the experimental evidence. It may also be noted that steady state conditions have established themselves in approximately 17 seconds (corresponding to about 320 forward steps).

In Figure 18, the dilution factor DF is shown to be a strong function of the densimetric Froude number Frd in both predictive and laboratory data. The above base values are retained for the other parameters. The general result is that release water quality (fraction of epilimnetric (top) water in the exit stream) increases with higher values of the Froude number, such conditions being achieved with higher jet velocities from the propeller and/or a lower degree of stratification in the lake. The figure clearly shows the difference between the use of a constant eddy viscosity model and the k-ε turbulence model: the former under-predicting the release water quality and the latter giving more accurate predictions when compared to the experimental data. From the designer's view, the important result is the Froude number at which the dilution factor rises sharply, since that determines the velocity required at the propeller for the given conditions. Clearly, the 3-D turbulent simulation is superior to the constant eddy viscosity model, and this Froude number is predicted better via this approach. The computations were performed with the standard k-ε turbulence model for the mixing. The influence of buoyancy was via the buoyancy

term in the vertical momentum equation. No attempt has been made to include the effects of buoyancy on the generation rates of the turbulence parameters k and ε, a practice advocated elsewhere (63, 83), since there is as yet no adequate generalization of the rather meagre evidence.

The relationship between the dilution factor DF and the flow rate ratio Q* is represented in Fig. 19. The figure shows the dilution factor strongly dependent upon Q* for Q* < 0.6 whereas Q* increases the release flow rate decreases. Comparison between Figure 19 and the corresponding laminar predictions in Figure 11 shows that the turbulence simulation gives a slightly better prediction at a high value of Q* (move mixing). On the otherhand, the laminar simulation gives better predictions at lower values of Q* which indicates move entrainment and less mixing.

The normalized propeller diamter D* effect on the dilution factor is shown in Fig. 20. Predictions compared with the laminar simulation in Figure 12 show better agreement with experiments at a low value of D*. On the other hand, the laminar model gives much better predictions at higher value of D*. This can be explained by the fact that with the large area for the jet (for higher values of D*) and since the propeller axis is at the same position, the resulting jet will be very close to the release structure and eventually to the release gate. This also indicates more entrainment than mixing since the large size of the jet will occupy much of the area between the jet axis and the release structure and eventually will displace the fluid in that area. Also, as indicated in the discussion of Figure 19 this

emphasis the fact that the turbulent model performs better in cases involving more mixing and less entrainment and the opposite is true for the laminar model.

The relation between the dilution factor DF and the normalized metalimnion location (thermocline or interface) Z_T^* is shown in Figure 21. The normalized metalimnion location Z_T^* range is from 0 to 1.0. At a Z_T^* value of zero, the fluid in the reservoir consists only of epilmnion water (top layer). On the other hand, when Z_T^* has the value of 1.0 the reservoir fluid will contain hypolimion water (bottom layer) only. This clearly indicates that the dilution factor will be unity in the first case ($Z_T^* = 0.0$) because all the release fluid will be high quality water (epilmnion). In the second case ($Z_T^* = 1.0$) the dilution factor will be zero because the release flow will consist of low quality water (hypolimnion) only. The figure shows that the dilution factor is a strong function of the metalmnion location Z_T^*. The practical value observed in reservoirs for Z_T^* is between 0.5 and 0.6.

Transient behavior of the flowfield is shown for the turbulent flow case in terms of vector velocity and 3-D velocity plots. The plots are shown in a similar fashion to the laminar flow plots using the same number of frames and time-interval. Figure 22 shows the first four frames of turbulent vector velocity plots. Comparing the first frame with the laminar one (Figure 13), it is clear that the turbulent jet does not affect the fluid as far as the reservoir bottom in the first time-step.

The fact of the matter is that the jet barely affects the flow beneath the interface (thermocline). This can be explained by the fact that the turbulent jet has a higher viscosity than its laminar counterpart

and, thus, spreads faster in the fluid and does not pentrate as far. Furthermore, the turbulent jet does not reach the reservoir bottom in the second frame, although it traveled significantly further than it did in the first frame. The turbulent jet does pentrate as far as the bottom in the third time-frame. The last four frames displayed in Figure 23 show more mixing taking place in the reservoir and around the jet specially in the last two frames.

The 3-D velocity plots for the turbulent prediction are presented for three cases of turbulence model variations:

1. k-ε turbulence model without the wall shear stress calculations.

2. k-ε turbulence model with the wall shear stress calculations used in the turbulence model only.

3. k-ε turbulence model with the wall shear stress calculations used in the turbulence model as well as the momentum equations.

In the first case shown in Figures 24 and 25, where the only turbulence introduced into the flowfield is by means of the propeller jet, the plots are very similar to the laminar simulation plots. Apart from the first frame in which the jet appears to spread faster than its laminar counterpart, the rest of the frames are almost identical to the laminar plots.

The second case plots, shown in Figures 26 and 27, exhibit a completely different behavior as compared to the first case and the laminar plots. The first frame is similar to its first case counterpart, but advancing through the frames shows that the jet spreads faster. Also, the indication of mixing is present where velocities are negative (an indication of the flow going up) around the jet. Although the 3-D velocity plots is still showing more mixing around the jet in

the last two time-frames, the last frame is considered to be the steady state solution according to Fig. 17.

The third case shown in Figures 28 and 29 is somewhat similar to the second case, although the negative velocities around the propeller are higher which indicates more mixing. This case, however, did not yield exactly a steady state solution compared to the second where the dilution factor reached a constant steady state value in Figure 17. The dilution factor in the third case was still decreasing after thirty seconds. The instability is generated by the amount of shear stress imposed near the wall in the momentum equation which results from the thick boundary layer near the wall. This thick boundary layer was unavoidable because of the uniform coarse grid used in the computations.

The second case is the one used in the computations and it is also the one used to generate the vector velocity plots.

5.1.4 Closure

The fully three-dimensional simulation represents a low-cost basic tool to show the influence of design parameters on local destratifications in reservoirs with low-level release structures. Comparison with hydraulic models shows that this numerical simulation of the local destratification phenomenon is useful for the prediction of the dilution factor (released water quality). The main dynamic effects are modeled adequately. The dilution factor is found to be a function of the densimetric Froude number, the ratio of propeller flow rate to the release flow rate, the propeller diameter, the propeller distance from the release structure, and the metalimnion (interface) location.

5.2 Deflected Turbulent Jet

An interesting three-dimensional problem arises when a turbulent jet enters normally into a uniform steady cross-flow. Figure 30 shows a schematic of the problem, which has previously been discussed (28), predicted (38) and experimentally investigated (84-87). Such problem configurations arise in chimney plumes, flow under a V/STOL aircraft, film cooling and dilution air mixing in combustor applications. Clearly, the problem is more complicated than corresponding free jet flows into quiescent surroundings, which are axisymmetric parabolic problems. Earlier predictions (38) utilized a proprietary three-dimensional implicit computer code to obtain steady state predictions. This sophisticated code is more complicated to use than the present code. Moreover, it is not generally available. Hence, the capability of the present straightforward explicit solution scheme is now illustrated for this problem.

5.2.1 Special Boundary Conditions

The simulation is for the turbulent jet of inlet velocity v_j, see Figure 30. Use is made of the vertical symmetry plane. A horizontal top plane is located 18 jet radii above the bottom plane. A grid system of 7 x 9 x 9 internal cells (in x,y, and z directions, respectively) is used. Uniform steady crossflow is specified at the upstream plane of the main inflow with velocity u_{in}. The usual zero normal gradients are taken at the exit plane. Free slip boundary conditions are used on the top, bottom, and side planes.

5.2.2 Predictions

Figure 31 shows the predicted velocity vectors in the yz-plane through x = 0. This gives a clear indication of the behavior, magnitude, and direction of the deflected jet. These predictions are for the jet to cross-flow velocity ratio $R = u_j/u_{in} = 6$, and the standard values in the k-ε turbulence model are used (4). Velocity vector plots of this type allow the jet trajectory(the line joining points of maximum velocity) to be determined.

Experimental work for comparison purposes include: Jordinson (84), Keffer and Baines (85)(who studied the structure of turbulence), Ramsey and Goldstein (86) (who presented velocity and temperature plots in cross-sectional and symmetry planes), and Chassaing et al (87) (who studied both cylindrical and co-axial jets). The predicted effect of jet to cross-flow velocity ratio R on jet trajectories is illustrated in Figure 32 with appropriate experimental data for comparison. Earlier predictions (38) are very much in agreement with the present study as shown in the figure. Effect of grid size on the solution is illustrated in Figure 33 for a jet to cross-flow velocity ratio R = 2. Predictions made with larger grid size (as expected) show better agreement with the experimental data than the small grid size predictions. The difference between the two predictions, however, is relatively small (7 percent) and considering the high cost of using a large grid size (about 23 minutes of computer time compared to 8 minutes for small grid size) the smaller grid size may be considered as a more practical choice. Considering the scatter in the data, the quality of the predictions is very good, thus confirming that the

main dynamic and turbulence mixing effects are modeled adequately in the present code.

5.2.3 Closure

The round turbulent jet emerging into uniform cross-flow requires a fully three-dimensional simulation for which one solution procedure is admirably suited. Comparison of predictions with experimental data and earlier predictions shows that this 3-D simulation is useful for the prediction of the jet centerline location and velocity vectors. Predictions show, as expected, that the location of jet centerline is a strong function of the jet-to-cross-flow velocity ratio.

5.3 Dilution Jets in Gas Turbine Combustors

Dilution jets in gas turbine combustors is a design area which receives considerable attention during the design and development of combustors. Lateral jets are used to maintain a stable burning zone, dilute near-stoichiometric mixtures and to cool and evenly mix the products before introduction to the turbine as shown in Figure 34 (88). High temperature streaks in a combustor outlet traverse or a skewed radial temperature profile (hub oriented) are detrimental to turbine vanes and blades (89). Also with the advent of high temperature-rise combustors, the dilution mixing will become an increasingly more difficult problem. High temperature-rise combustors require an additional amount of air for combustion and linear cooling which in return reduces the amount of air available for the dilution process.

Present design practices rely on extensive rig and engine development testing in order to adjust the exit temperature traverse to meet

the demands of engine hot section durability. This development testing is guided by empirical design analysis and experience of design engineers. These empirical design techniques, however, are becoming inadequate when used for predicting stringent temperature traverse requirements and low-pollution generating combustors.

Earlier modeling of the dilution zone utilized analytical as well as computational techniques. A simple analytical model was used for predicting dilution jet trajectories inside a can combustor with zero swirl (89). Also the characteristic-time model (90) which is a semi-empricial correlation for gas turbine emissions, ignition, and flame stabilization is considered as one of the successful models for modeling the dilution zone (91). These types of models form the basis of the most current preliminary design systems.

Computational or numerical models which provide 3-D predictions of velocities, pressure, temperatures, species, and concentratins are relatively new. Examples of application of these type of models have been presented in the literature (40, 42). A good review of the application of these models is also available (1). These 3-D codes, however, are not released for the general use. Moreover, they are more complicated to use than the present code and will be a major obstacle for users with little experience is computational fluid dynamics.

5.3.1 Special Boundary Conditions

The finite-difference grid employed here exhibits 10 x 10 x 7 internal cells in the z, r, and θ directions, with velocity components v, u, and w, respestively. The three-dimensional grid system is shown in Figure 2. The dilution jet is located one diameter downstream of the

inlet. The combustor is 0.6 m in length and 0.3 m in diameter. Also, since the flow pattern repeats itself in each 60 deg. sector in the combustor cross section, the grid system (or the solution domain) covers only one 60 degrees sector.

The θ-boundary conditions of the solution domain are cyclic ones (periodic). That is the given boundary condition will be that all the θ-planes separated by multiples of 60 degrees are corresponding planes with identical profiles. Thus the θ-boundary conditions are handeled in a manner that preserve the cyclic status by letting the inelt profile and gradient of the first θ-plane equal the outlet profile and gradient of the last θ-plane. This periodic boundary condition applies not only to velocities and pressure, but also to all the other dependent variables. Inlet axial velocity v_o has the value of 10.0 m/s which corresponds to a Reynolds number based on inlet diameters of 10^5.

5.3.2 Predictions

Predictions were made using inlet velocity profiles based on mean flowfield measurements (71) for the axial velocity v_o and tangent velocity w_o. Radial velocity u_o, however, is given the value of zero at the inlet. Predicted time-mean velocities are shown (with the radial scale stretched by a factor of two) for three cases:

1. Nonswirling flow with and without a dilution jet (shown in Figure 35).

2. Swirling flow with and without a dilution jet in a low Reynolds number flow (shown in Figure 36).

3. Swirling flow with and without a dilution jet in a high Reynolds number flow (shown in Figure 38).

In the first case, shown in Figure 35(a), the velocity vectors are showing magnitude and direction in the θ-plane for nonswirling flow without dilution. The predicted velocity vectors show the expected corner recirculation zone as a result of the sudden expansion. Figure 35(b) illustrates the effect of the dilution jet with a jet-to-inlet velocity ratio of $u_d/v_o = 1$. The predicted velocity vectors show the pentration and deflection of the dilution jet in the combustor. The amount of deflection corresponds to a jet to cross-flow velocity ratio R of 4 as shown in Figure 32 for cartesian coordinates. This is a result of the change in the chamber area which reduces the inlet velocity and makes the dilution jet to average axial velocity ratio close to 4.

In the second case, velocity vectors for swirling flow with swirl vane angle $\theta = 45°$ and zero dilution jet velocity are shown in Figure 36(a). The figure shows a swirling low Reynolds number flow (Re = 8000) that exhibits the expected central and corner recirculation zones observed in experiments (71). The corresponding flow with a dilution jet that has a jet-to-inlet velocity ratio of 1.0 is shown in Figure 36(b). Both vector velocity plots show essentially the same pattern with the corner and central recirculation zones being present. The effect of swirl on the dilution jet pentration is shown in the latter vector velocity plot where the jet shows a shorter pentration than the zero swirl case. Earlier predictions for a low Reynolds number flow with a dilution jet-to-inlet velocity ratio of 1.0 is shown in Figure 37 (40). The shown predictions are for a hot flow with single-phase, diffusion-controlled combustion, and 45° swirl vane angle. These predictions were made using a version of the TEACH code with a similar combustion chamber. Notice that there is no scale stretch in this

figure. Velocity vectors in the above mentioned figure exhibit the same trend shown previously in Figure 36(b).

In the third case, high Reynolds number swirling flow ($Re = 10^5$) with and without a dilution jet are shown in Figure 38. Figure 38(a) shows swirling flow with swirl vane angle $\phi = 45°$ and a zero dilution jet velocity. The figure exhibits a larger recirculation zone at the axis (central), compared to the low Reynolds flow, but it shows little or no recirculation at the corner. This may be explained by the strong centrifugal effect which promotes a very large forward velocity near the wall and almost eliminates the corner region. Figure 38(b) shows a swirling flow with the same degree of swirl and a dilution jet-to-inlet velocity ratio of 1.0. The figure displays a similar pattern to Figure 38(a) with little or no corner recirculation and a larger central recirculation zone. The dilution jet pentration is far less than low Reynolds number jet pentration, basically because of the strong centrifugal force which causes the jet additional deflection. Figure 39 shows velocity vectors with swirl and dilution and high Reynolds number flow in the seven θ-planes from K=1 to K=7. The figure shows that the flow is by no means symmetric around the dilution jet plane, mainly because of the swirl velocity which tends to deflect the jet in the θ direction.

5.3.3 Closure

Dilution jets in gas turbine combustors is a fully three-dimensional problem which must be simulated using cylindrical coordinates system. The developed prediction procedure represents a basic tool to show the

influence of design parameters on the dilution zone in combustors. Predicted results show the expected trends in the flow such as the size of corner and central recirculation zones and dilution jet deflection. Comparison with earlier predictions shows that the quality of the predicted results is very good; thus, confirming that the main dynamic and the turbulence mixing effects are modeled adequately. Discrepancy in predicted velocity profiles compared to the measured ones can be overcome by moving the downstream boundary condition away from the field of interest.

CHAPTER VI

CLOSURE

6.1 Conclusions

A prediction procedure for fully three-dimensional transient turbulent swirling flows has been developed. Based on the Los Alamos 2-D SOLA ideas, the transient Reynolds equations of an incompressible fluid in cartesian or cylindrical coordinates are solved via their associated finite difference equations in terms of the primitive pressure velocity variables. Turbulence is simulated by means of the two-equation k-ε turbulence model. Species diffusion and buoyancy forces are included. The developed code is a simplified yet effective prediction procedure for use by persons with little or no experience in computational fluid dynamics. The code thus represents a basic tool, to which user-oriented complexities and sophistications can easily be added as required.

Three applications of the code are presented:

1. Local destratification near the release structure of a resevoir.
2. The deflection of a jet entering normally into a uniform cross-flow.
3. Dilution jets in gas turbine combustors.

Predicted results of the three applications exhibit good agreement with available experimental data and earlier predictions (obtained using different codes), showing that a useful characterization of fully

three-dimensional turublent swirling flows is now available.

6.2 Recommendations for Further Work

Improvements and modifications to further enhance the code capability should be continued in several areas. First, nonuniform grid capability could be added to the code to provide a fine grid for the boundary layer calculations near interesting flow regions and walls without increasing computer storage requirements. Second, the range of applications could be broadened even further by adding chemical reaction and compressible flow simulations to the code. This requires using the energy equations, the equation of state, and species equations. It would be particularly useful for combustor applications. Third, possible use of an implicit solution algorithm should be investigated as an option in the code. This would be beneficial for stability reasons in transient cases. In steady state type applications, this would also reduce computer time requirements.

REFERENCES

1. Lilley, D. G., "Flowfield Modeling in Practical Combustors: A Review," Journal of Energy, Vol. 3, No. 4, July-August 1979, pp. 193-210.

2. Launder, B. E. and Spalding, D. B., Mathematical Models of Turbulence, Academic Press, London, 1972.

3. Hirt, C. W., Nichols, B. D. and Romero, N. C., "SOLA - A Numerical Solution Algorithm for Transient Fluid Flows," Report LA-5852, 1975, Los Alamos Scientific Laboratory, Los Alamos, NM.

4. Launder, B. E. and Spalding, D. B., "The Numerical Computation of Turbulent Flow," Computer Methods in Applied Mechanics and Engineering, Vol. 3, 1974, pp. 269-89.

5. Roache, P. J., Computational Fluid Dynamics. Hermosa, Albuquerque, NM, 1972.

6. Campbell, D. R., and Mueller, T. J., "A Numerical and Experimental Investigation of Incompressible Laminar Ramp - Induced Separated Flow," UNDAS TN-1068-M1, Dept. of Aero-Space Engineering, University of Notre Dame, Notre Dame, Indiana, 1968.

7. Chow, C. Y., An Introduction to Computational Fluid Mechanics, John Wiley and Sons, New York, 1979.

8. Harlow, F. H., and Fromm, J. E., "Computer Experiments in Fluid Dynamics," Scientific American, Vol. 212, No. 3, 1965, pp. 104-110.

9. Harlow, F. H., and Welch, J. E., "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, (The MAC Method)," Physics of Fluids, Vol. 8, 1965, pp. 2182-2189.

10. Amsden, A. A., and Harlow, F. H., "The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows," Report LA-4379, 1970, Los Alamos Scientific Laboratory, Los Alamos, NM.

11. Harlow, F. H., and Amsden, A. A., "A Numerical Fluid Dynamics Calculation Method for Flow Speeds," Journal of Comp. Physics, Vol. 8, 1971, pp. 197-213.

12. Amsden, A. A., and Harlow, F. H., "KACHINA: An Eulerian Computer

Program for Multi-Field Flows," Los Alamos Scientific Laboratory Report, LA-5680, 1974, Los Alamos, NM.

13. Harlow, F. H., and Amsden, A. A., "Numerical Calculation of Multiphases Flow," Journal of Comp. Physics, Vol. 17, pp. 19-52, January, 1975.

14. Amsden, A. A., and Hirt, C. W., "YAQUI: An Arbitrary Labrangian-Eulerian Computer Program for Fluid Flow at all Speeds," Los Alamos Scientific Laboratory Report LA-5100, 1973, Los Alamos, NM.

15. Harlow, F. H., and Amsden, A. A., "Multifluid Flow Calculations at all Mach Numbers," Journal of Comp. Physics, Vol. 16, pp. 1-19, 1974.

16. Rivard, W. C., Farmer, O. A., and Butler, T. D., "RICE: A Computer Program for Multicomponent Chemically Reactive Flows at ALL Speeds," Los Alamos Scientific Laboratory, Report LA-5812, March, 1975, Los Alamos, NM.

17. Cloutman, L. D., Hirt, C. W., and Romero, N. C., "SOLA-ICE: A Numerical Solution Algorithm for Transient Compressible Fluid Flows," Report LA-6236, 1976, Los Alamos Scientific Laboratory, Los Alamos, NM.

18. Hirt, C. W. and Nichols, B. D., "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," Journal of Physics, Vol. 39, 1981. pp. 201-225.

19. Busnaina, A. A., "Numerical Simulation of Local Destratification of Lakes," Thesis presented to the Oklahoma State University at Stillwater, OK, in 1979, in partial fulfillment of the requirement for the degree of Master of Science.

20. Busnaina, A. A., "Local Destratification of Lakes - A Numerical Solution," Proceedings of the 11th Southwestern Graduate Research Conference in Applied Mechanics, Stillwater, OK, April 11-12, 1980, pp. 2.26-2.33.

21. Busnaina, A. A., Lilley, D. G., and Moretti, P. M., "Prediction of Local Destratification of Lakes," Journal of Hydraulics Division, ASCE, HY3, Proc. Paper 16094, March 1981, pp. 259-272.

22. Busnaina, A. A., and Lilley, D. G., "Numerical Simulation of Swirling Flow in a Cyclone Chamber," Proceeding, Symposium on the Fluid Mechanics of Combustion Systems, Joint ASME/ASCE Fluids Engineering and Applied Mechanics Conference, Boulder, Colorado, June 22-24, 1981. pp. 169-178.

23. Hirt, C. W., and Cook, J. L., "Calculating Three-Dimensional Flows Around Structures and Over Rough Terrain," Journal of Comp.

Physics, Vol. 10, 1972, pp. 324-340.

24. Nichols, B. D., and Hirt, C. W., "Calculating Three-Dimensional Free Surface Flows in the Vicinity of Submerged and Exposed Structures," Journal of Comp. Physics, Vol. 12, 1973, pp. 234-246.

25. Pracht, W. E., "Calculating Three-Dimensional Fluid Flows at All Speeds with an Eulerian-Langrangian Computing Mesh," Journal of Comp. Physics, Vol. 17, 1975, pp. 132-159.

26. Busnaina, A. A., and Lilley, D. G., "Computer Prediction of Local Destratification Near Low-Level Release Structures of Reservoires," Paper No. 81-FE-11 presented at the ASME Fluids Engineering Division 1981 Spring Meeting, Boulder, Colorado, June 22-24, 1981. J. of Fluids Engineering, 1982 (in press).

27. Busnaina, A. A., and Lilley D. G., "A Basic Code for the Prediction of Transient Three-Dimensional Turbulent Flowfields," Proceeding, Symposium on Three-Dimensional Turbulent Shear Flows, ASME Fluids Engineering Division 1982 Spring meeting, St. Louis, Missouri, June 7-10, 1982.

28. Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Hemisphere-McGraw-Hill, New York, 1980.

29. Spalding, D. B., GENMIX: A General Computer Program for Two-Dimensional Parabolic Phenomena, Pergamon Press, Oxford, 1977.

30. Gosman, A. D., Pun, W. M., Runchal, A. K., Spalding, D. B., and Wolfshtein, M. W., Heat and Mass Transfer in Recirculating Flows, Academic Press, London, 1969.

31. Gosman, A. D., and Pun W. M., "Calculation of Recirculating Flows," Report No. HTS/74/2 Dept. of Mech. Engr., Imperial College, London, 1974.

32. Patankar, S. V., and Spalding, D. B., "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Prabolic Flows," International Journal of Heat and Mass Transfer, Vol. 15, 1972, pp. 1787-1806.

33. Patankar, S. V., and Spalding, D. B., "A Computer Model for Three-Dimensional Flow in Furnaces," 14th Symposium (International on Combustion, The Combustion Institute, Pittsburg, Pennsylvania, 1973, pp. 604-614.

34. Pai, B. R., Michelfelder, S., and Spalding, D. B., "Prediction of Furnace Heat Transfer with a Three-Dimensional Mathematical Model," International Journal of Heat and Mass Transfer, Vol. 21, 1978, pp. 571-580.

35. Patankar, S. V., "Numerical Prediction of Three-Dimensional Flows," Studies in Convection, edited by B. E. Launder, Vol. 1,

Academic Press, London, 1975, pp. 1-78.

36. Caretto, L. S., Gosman, A. D., Patankar, S. V., and Spalding, D. B., "Tow Calculation Procedures for Steady-Three-Dimensional Flows with Recirculation," Proceedings of 3rd International Conference on Numerical Methods in Fluid Mechanics, Vol. 2, edited by H. Cabannes and R. Teman, Springer-Verag, Germany, 1973, pp. 60-68.

37. Patankar, S. V., and Spalding, D. B., "Simultaneous Predictions of Flow Pattern and Radiation for Three-Dimensional Flames," paper in Heat Transfer and Flames, edited by N. H. Afgan and J. M. Beer, Scripta Book Co. (Hemisphere-Wiley), Washington, D. C., 1974.

38. Patankar, S. V., Basu, D. K., and Alpay, S. A., "Prediction of the Three-Dimensional Velocity Fields of a Deflected Turbulent Jet," Journal of Fluids Engineering, Vol. 99, 1977, pp. 758-762.

39. Abou Ellail, M. M. M., Gosman, A. D., Lockwood, F. C., and Megahed, I. E. A., "Description and Validation of a Three-Dimensional Procedure for Combustion Chamber Flows." Turbulent Combustion, edited by L. A. Kennedy, Progress in Astronautics and Aeronautics, Vol. 58, AIAA, New York, 1978, pp. 163-190.

40. Serag-Eldin, M. A., and Spalding, D. B., "Computations of Three-Dimensional Gas Turbine Combustion Chamber Flows," ASME 78-GT-142, London, England, 1978.

41. Swithenbank, J., Turan, A., and Felton, P. G., "3-Dimensional 2-Phase Mathematical Modeling of Gas Turbine Combustors," project SQUID (ONR) Workshop, Gas Turbine Combustor Design Problems, meeting held at Purdue University, W. Lafayette, Indiana, May 31-June 1, 1978.

42. Boysan, F., Ayers, W. H., Swithenbank, J., and Pan, Z., Three-Dimensional Model of Spray Combustion in Gas Turbine Combustors," AIAA-81-0324, 19th Aerospace Sciences Meeting, St. Louis, Missouri, 1981.

43. Boysan, F., and Swithenbank, J., "Spray Evaporation in Recirculating Flow," International Symposium on Combustion, Leeds, 1978.

44. Gallagher, R. H., Oden, J. T., Taylor, C., and Zienkiewicz, O. C., "Finite Elements in Fluids," Vol. 2, John Wiley and Sons, New York, New York, 1975.

45. Chung, T. J., "Finite Element Analysis in Fluid Dynamics," McGraw-Hill, New York, New York, 1978.

46. Hamad, A., and Baskharone, E., "Analysis of the Three-Dimensional Flow in a Turbine Scroll," Symposium on Flow in Primary,

Non-Rotating Passages in Turbomachines, ASME Winter Annual Meeting, New York, New York, December 2-7, 1979, pp. 99-105.

47. Roache, P. J., "A Review of Numerical Techniques," Proceedings, First International Conference on Ship Hydrodynamics, Bethesda, Maryland, Oct. 20-22, 1975, pp. 609-640.

48. Sills, J. A., "Transformation for Infinite Regions and Their Application to Flow Problems," AIAA J., Vol. 7, No. 1 (1969), pp. 117-123.

49. Markatos, N. C., Spalding, D. B., and Tatchell, D. G., "Combustion of Hydrogen Injected into a Supersonic Airstream (The SHIP Computer Program)," NASA-CR-2802, April, 1977.

50. Roache, P. J., "A Semidirect Method for Internal Flows in Flush Inlets," AIAA Third Compt. Fluid Dynamics Conf., Albuquerque, New Mexico, June 27-28, 1977, pp. 149-155.

51. Moretti, G., and Abbett, M., "A Time-Dependent Computational Method for Blunt Body Flows," AIAA J., Vol. 4, No. 12 (1966), pp. 2136-2141.

52. Pope, S. B., "The Calculation of Turbulent Recirculating Flows in General Orthogonal Coordinates," J. of Comp. Physics, Vol. 26 (1978), pp. 197-217.

53. Pope, S. B., "The Calculation of Separated Boundary Layers," Symposium on Turbulent Shear Flows, Pennsylvania State University, April, 1977.

54. Thompson, J. F., Thames, F. C., and Mastin, C. W., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinates System for Fields Containing any Number of Arbitrary Two-Dimensional Bodies," J. of Comp. Physics, Vol. 15, No. 3, 1974, pp. 299-319.

55. Thompson, J. F., et al., "Use of Numerically Generated Body-Fitted Coordinate Systems for Solution of the Navier-Stokes Equations," Proceeding, AIAA Second Comp. Fluid Dynamics Conf., Hartford, Connecticut, June 19-20, 1975, pp. 68-80.

56. Amsden, A. A., and Hirt, C. W., "A Simple Scheme for Generating General Curvilinear Grids," J. of Comp. Physics, Vol. 11, 1973, pp. 348-359.

57. Moretti, G., "Conformal Mapping for Computation of Steady, Three-Dimensional, Supersonic Flows," Numerical/Laboratory Computer Methods in Fluid Mechanics, Editor: A. A. Pouring, ASME Publication, 1976.

58. Moore, J., and Moore, J. G., "A Calculation Procedure for Three-Dimensional, Viscous, Compressible Duct Flow," Symposium on

Flow in Primary, Non-Rotating Passages in Turbomachines, ASME Winter Meeting, New York, New York, December 2-7, 1979, pp. 75-88.

59. Forester, C. K., "Numerical Prediction of Compressible Flow for Arbitrary Geometries," Symposium on Flow in Primary, Non-Rotating Passages in Turbomachines, ASME Winter Meeting, New York, New York, Dec. 2-7, 1979, pp. 115-124.

60. Ghia, U., and Sathyanarayana, K., "Analysis and Solution of Three-Dimensional Viscous Flow in Non-Rotating Passages in Turbomachines", ASME Winter Meeting, New York, Dec. 2-7, 1979, pp. 179-188.

61. Rodi, W., "Progress in Turbulence Modeling for Incompressible Flows," AIAA Paper No. 81-0045, AIAA 19th Aerospace Science Meeting, St. Louis, Missouri, Jan. 12-15, 1981.

62. Prandtl, L., "Uber die ausgebildete Turbulenz," ZAMM, 5, 1925, p. 136.

63. Rodi, W., Turbulence Models and Their Application in Hydraulics, International Association for Hydraulic Research, Delft, The Netherlands, 1980.

64. Schlichting, H., Boundary Layer Theory, 6th ed., McGraw-Hill, New York, 1968.

65. Kolmogorov, A. N., "Equations of Turbulent Motion of an Incompressible Fluid," Izv. Akad. Nauk. SSR Seria Fizicheska Via., No. 1-2, 1942, pp. 56-58, English Translation Imperial College, Mech. Engr. Dept. Report ON/6, 1968.

66. Prandtl, L., "Uber ein neues Formel-system fur die ausgebildete Turbulenz," Nachr, Akad. Wiss., Gottingen, Math.-Phys. Klasse, 1945, p. 6.

67. Bradshaw, P., Ferriss, D. H., and Atwell, N. P., "Calculation of Boundary Layer Development Using the Turbulent Energy Equation," J. of Fluid Mech., 28, 1967, pp. 593-616.

68. Hassid, S., and Proeh, M., "A Turbulent Energy Model for Flows with Drag Reduction," ASME Paper 75-fe-H, 1975.

69. Launder, B. E., Morse, A. P., Rodi, W., and Spalding, D. B., "The Prediction of Free-Shear Flows - A Comparison of the Performance of Six Turbulence Models," Proceeding, NASA Langley Free Turbulent Shear Flows Conf., Vol. 1, 1973, NASA pp. 320.

70. Huffman, G. D., Zimmerman, D. R., and Bennet, W. A., "The Effect of Free-Stream Turbulence Level on Turbulent Boundary Layer Behaviour", AGARD A-AG 164, 1972, pp. 91-115.

71. Rhode, D. L., Lilley, D. G., and McLaughlin, D. K., "Mean Flowfields in Axisymmetric Combustor Geometries with Swirl," AIAA Paper No. 82-0177, AIAA 20th Aerospace Science Meeting, Orlando, Florida, Jan. 11-14, 1982.

72. Launder, B. E., Reece, G. J., and Rodi, W., "Progress in the Development of a Reynolds Stress Turbulence Closure," J. of Fluid Mech., Vol. 68, 1975, pp. 537-66.

73. Gibson, M. M., and Launder, B. E., "Ground Effects on Pressure Flutution in Atmospheric Boundary Layer," J. of Fluid Mech., Vol. 86, 1978, p. 491

74. Lumley, J. L., and Khajeh-Nouri, B., "Computation of Turbulent Transport," Adv. in Geophys. A18, 1974, pp. 169-192.

75. Lewellen, W. S., Teske, M. and Donaldson, C. du P., "Variable Density Flows Computed by a Second-Order Closure Description of Turbulence," AIAA J., Vol. 14, 3, 1976, pp. 382-387.

76. Reece, G. J., "A Generalized Reynolds-Stress Model of Turbulence," Ph.D. Thesis, University of London, 1977.

77. Rodi, W., "A New Algebraic Relation for Calculating the Reynolds Stresses," ZAMM, 56, 1976, pp. T219-221.

78. Ljuboja, M., and Rodi, W., "Calculation of Turbulent Wall Jets with an Algebraic Reynolds-Stress Model," J. of Fluids Engineering, Vol. 102, 1980, pp. 350-356.

79. Blackshall, R. G. and Landis, F., "The Boundary-Layer Velocity Distribution in Turbulent Swirling Pipe Flow," Journal of Basic Engr., ASME Trans., Dec. 1969, pp. 728-733.

80. Moon, J. J., "Enhancement of Release Water Quality by Localized Mixing - A Hydraulic Model Study, "Thesis presented at the Oklahoma State University at Stillwater, OK, in 1978, in partial fulfillment of the requirements for the degree of Master of Science.

81. Hinze, J. O., Turbulence, 2nd ed., McGraw-Hill, New York, New York, 1975, p. 539.

82. Hinze, J. O., and Van Der Hegge Zijnen, B. G., "Transfer of Heat and Matter in the Turbulence Mixing Zone of an Axially Symetrical Jet," Applied Science Res., Vol. A1, 1949, pp. 435-461.

83. Hossain, M. S. and Rodi, W., Influence of Buoyancy on the Turbulent Intensities in Horizontal and Vertical Jets in Heat Transfer and Turbulent Buoyant Convection. Studies and Applications for Natural Environment, Buildings, Engineering

Systems, (D. B. Spalding and N. Afgan, eds.) Hemisphere Publishing Corp., Washington, D. C., USA, 1977.

84. Jordinson, R., "Flow in a Jet Directed Normal to the Wind," Aeronautical Research Council, R & M No. 3074, 1958.

85. Keffer, J. F. and Baines, W. D., "The Round Turbulent Jet in a Cross-Wind," Journal of Fluid Mechanics, Vol. 15, Pt. 4, 1963, pp. 481-496.

86. Ramsey, J. W. and Goldstein, R. S., "Interaction of a Heated Jet with Deflecting Stream," NASA Cr-72613, HTL TR No. 92, April 1970.

87. Chassaing, P., "Physical Characteristics of Subsonic Jets in a Cross-Stream," Journal of Fluid Mechanics, Vol. 62, Part 1, 1973, pp. 41-64.

88. Mellor, A. M., "Gas Turbine Engine Pollution," Progr. Energy Combust. Sci., 1, pp. 111-133, 1976.

89. Novick, A. S., Arvin, J. R., and Quinn, R. E., "Development of a Gas Turbine Combustor Dilution Zone Design Analysis," AIAA Paper No. 79-1194, AIAA/SAE/ASME 15th Joint Propulsion Conference, Las Vegas, Nevada, June 8-10, 1979.

90. Mellor, A. M., "Semi-Empirical Correlations for Gas Turbine Emissions, Ignition and Flame Stabilization," AGARD Conference Reprint No. 275, Combustion Modeling, AGARD, pp. 24.2-24.13, 1979.

91. Sturgess, G., "Gas Turbine Combustor Design Challenges for 1980's," AIAA Paper No. 80-1285, AIAA/SAE/ASME 16th Joint Propulsion Conference, Hartford, Connecticut, June 30-July 2, 1980.

APPENDIX A

TABLES

TABLE I

SCHMIDT NUMBERS AND SOURCE TERMS
IN THE GENERAL EQUATION (3.1)

ϕ	σ_ϕ	S_ϕ
1	0	0
u	1	$-\frac{\partial p}{\partial x} + \frac{\partial}{\partial x_i}\left(\mu_{eff} \frac{\partial u_i}{\partial x}\right)$
v	1	$-\frac{\partial p}{\partial x} + \frac{\partial}{\partial x_i}\left(\mu_{eff} \frac{\partial u_i}{\partial y}\right)$
w	1	$-\frac{\partial p}{\partial x} + \frac{\partial}{\partial x_i}\left(\mu_{eff} \frac{\partial u_i}{\partial z}\right) - g(\rho - \rho_1)$
m_1	1	0
k	1	$G - C_D \rho \varepsilon$
ε	1.3	$(C_1 G\varepsilon - C_2 \rho \varepsilon^2)/k$

The rate of generation of k by the action of velocity gradients is taken as:

$$G = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \frac{\partial u_i}{\partial x_j}$$

TABLE II

SCHMIDT NUMBERS AND SOURCE TERMS
IN THE GENERAL EQUATION (3.3)

ϕ	σ_ϕ	S_ϕ
1	0	0
u	1	$\frac{w^2}{r} - \frac{\partial p}{\partial r} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu_{eff}\frac{\partial u}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(r\mu_{eff}\frac{\partial (w/r)}{\partial r}\right)$ $- 2\frac{\mu_{eff}}{r}\left[\frac{1}{r}\frac{\partial w}{\partial \theta} + \frac{v}{r}\right] + \frac{\partial}{\partial z}\left(\mu_{eff}\frac{\partial v}{\partial r}\right)$
v	1	$-\frac{\partial p}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu_{eff}\frac{\partial u}{\partial z}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\mu_{eff}\frac{\partial w}{\partial z}\right)$ $+ \frac{\partial}{\partial z}\left(\mu_{eff}\frac{\partial v}{\partial z}\right)$
w	1	$-\frac{vw}{r} - \frac{\partial p}{r\partial \theta} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu_{eff}\left[\frac{1}{r}\frac{\partial u}{\partial \theta} - \frac{w}{r}\right]\right)$ $+ \frac{\mu_{eff}}{r}\left[r\frac{\partial (w/r)}{\partial r} + \frac{1}{r}\frac{\partial u}{\partial \theta}\right]$ $+ \frac{1}{r}\frac{\partial}{\partial \theta}\left(\mu_{eff}\left[\frac{\partial w}{r\partial \theta} + \frac{2u}{r}\right]\right) + \frac{\partial}{\partial z}\left(\mu_{eff}\frac{\partial v}{\partial \theta}\right)$
m_1	1	0
k	1	$G - C_D \rho\varepsilon$
ε	1.3	$(C_1 G \varepsilon - C_2 \rho\varepsilon^2)/k$

The rate of generation of k by the action of velocity gradients is taken as:

$$G = \mu_t \left\{ 2\left(\frac{\partial v}{\partial z}\right)^2 + 2\left(\frac{\partial u}{\partial r}\right)^2 + 2\left(\frac{\partial w}{r\partial \theta} + \frac{u}{r}\right)^2 + \left(\frac{\partial v}{\partial r} + \frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial w}{\partial z} + \frac{\partial v}{r\partial \theta}\right)^2 + \left(\frac{\partial u}{r\partial \theta} + \frac{\partial w}{\partial r} - \frac{w}{r}\right)^2 \right\}$$

APPENDIX B

FIGURES

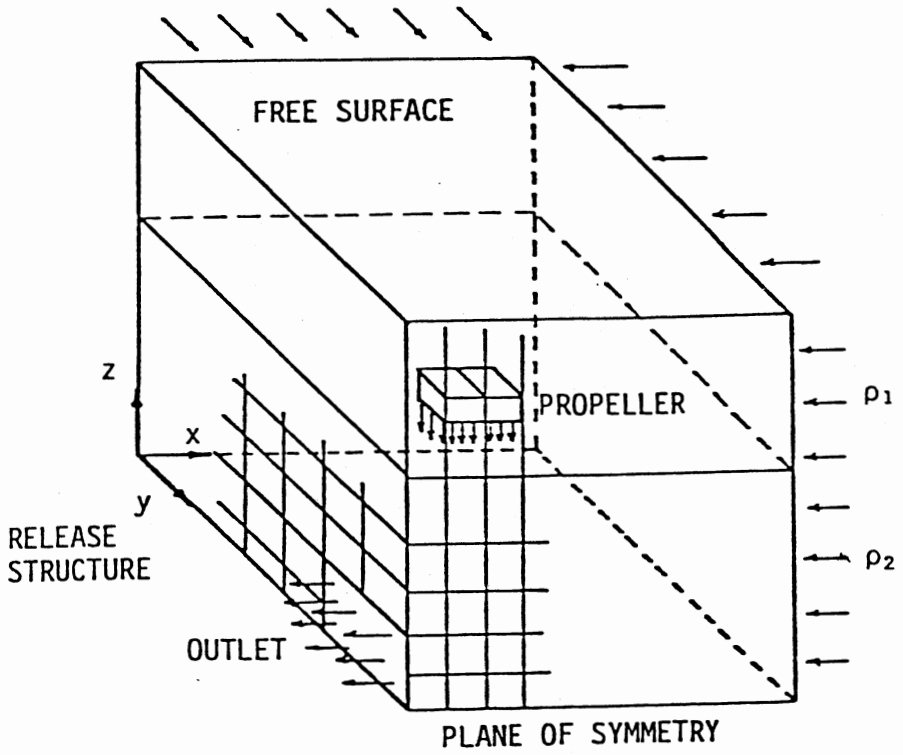


Figure 1. Fully 3-D Cartesian Grid Schematic.

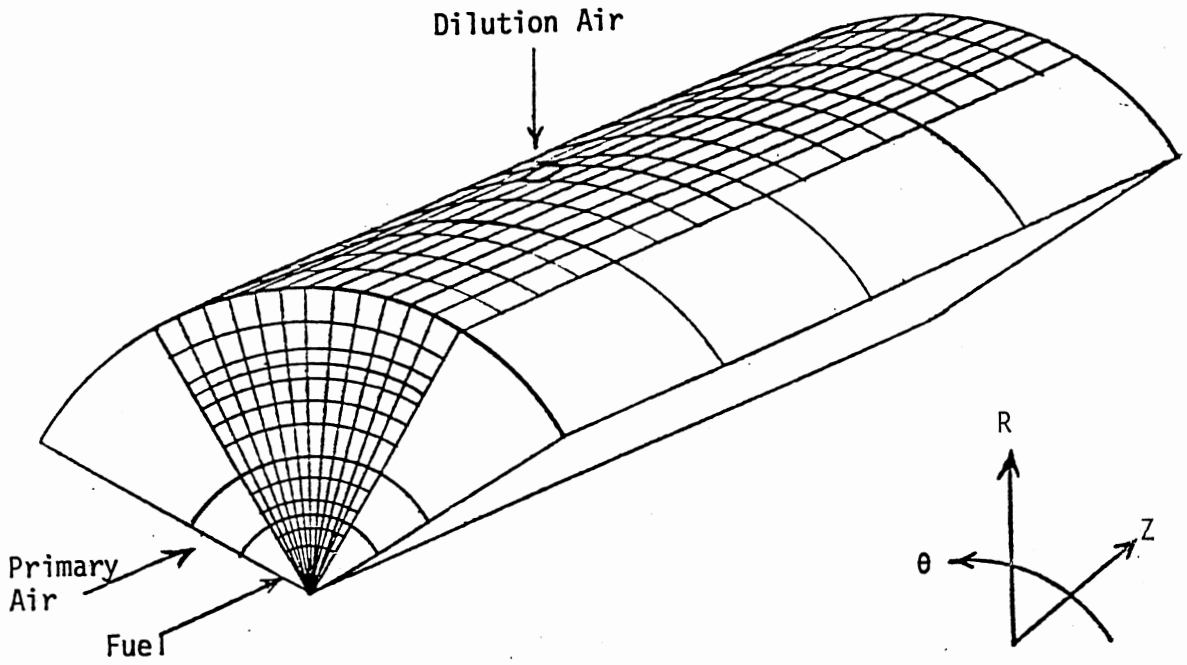


Figure 2. Fully 3-D Cylindrical Grid Schematic.

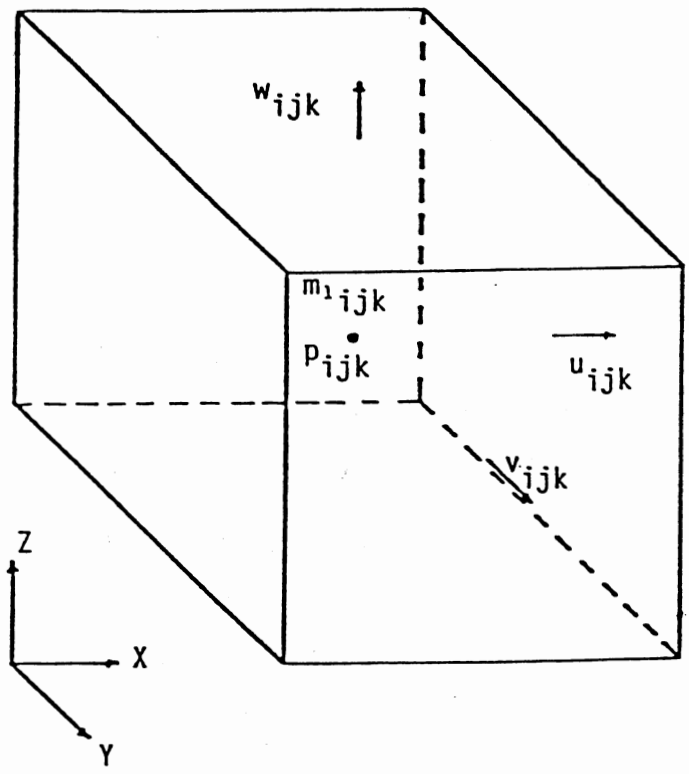


Figure 3. Arrangement of Finite Difference Variables in a Typical Cell in Cartesian Coordinates.

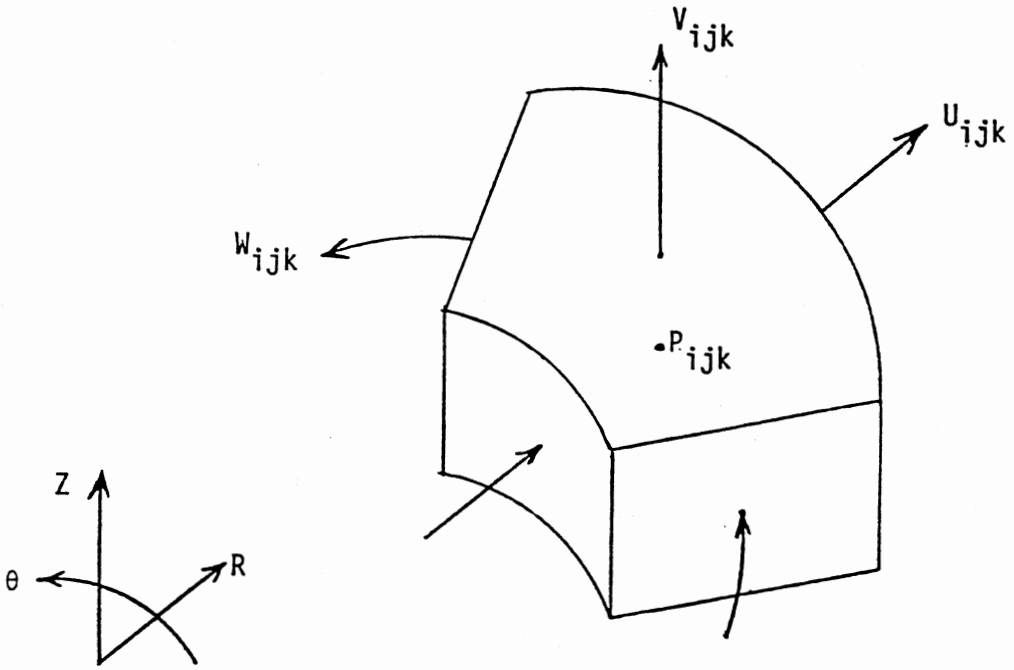


Figure 4. Arrangement of Finite Difference Variables in a Typical Cell in Cylindrical Coordinates.

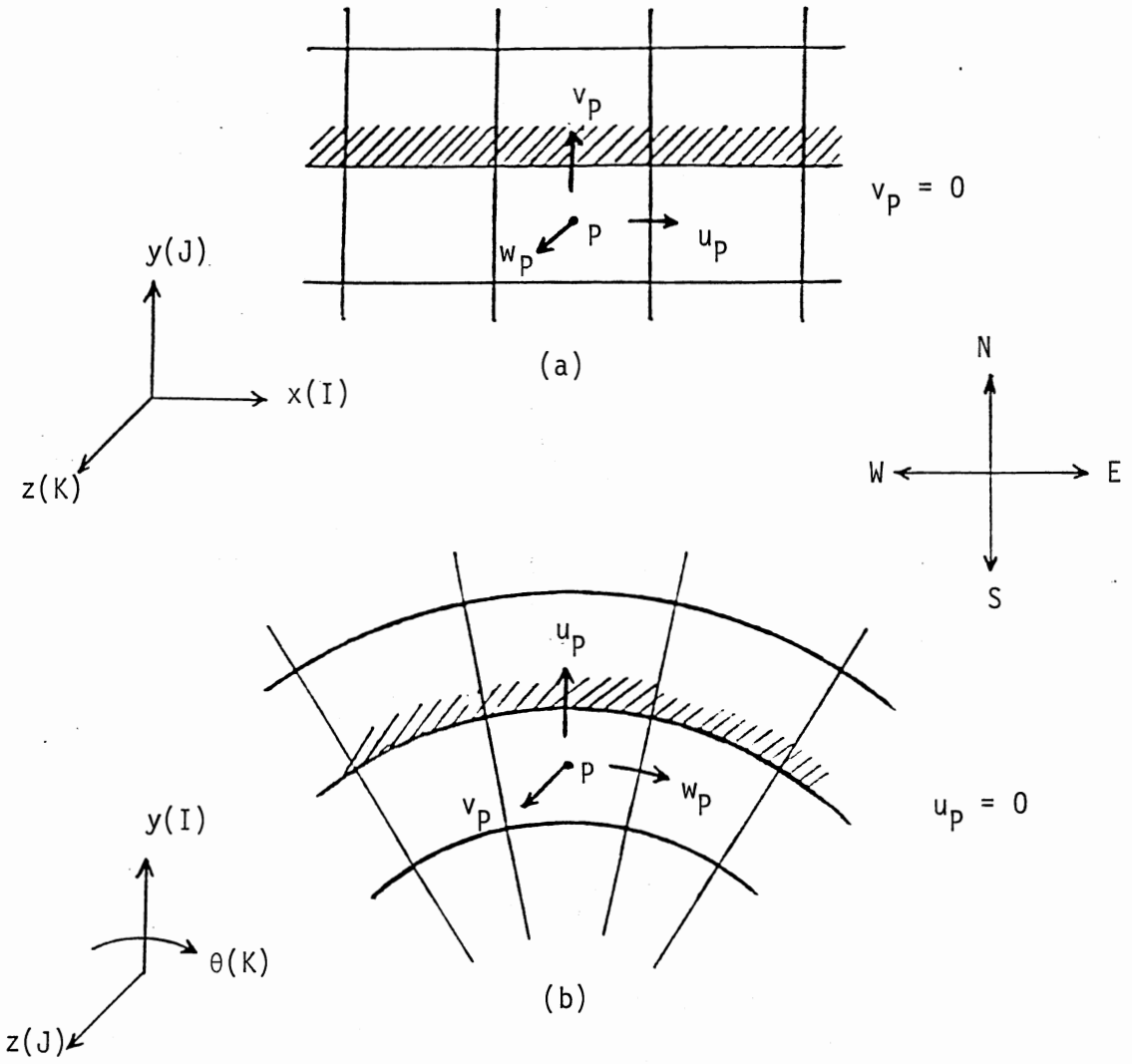


Figure 5. Arrangement of Finite Difference Variables Near the Wall.

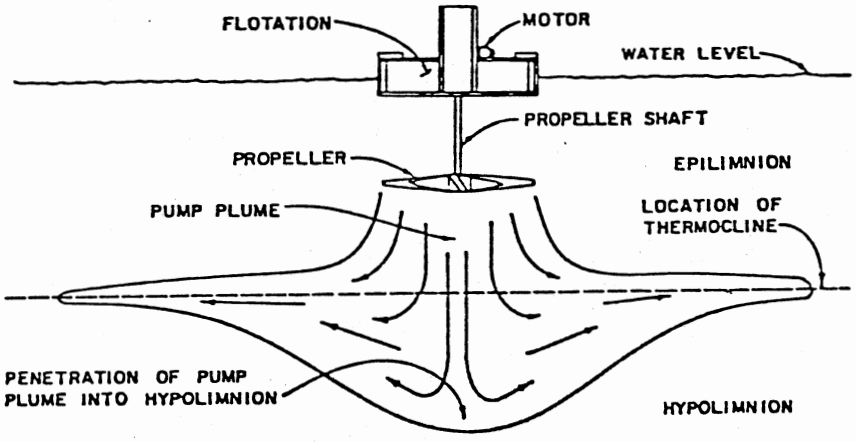


Figure 6. Schematic of a Typical Propeller Pump and the Flowfield Produced Without Exit Flow (80).

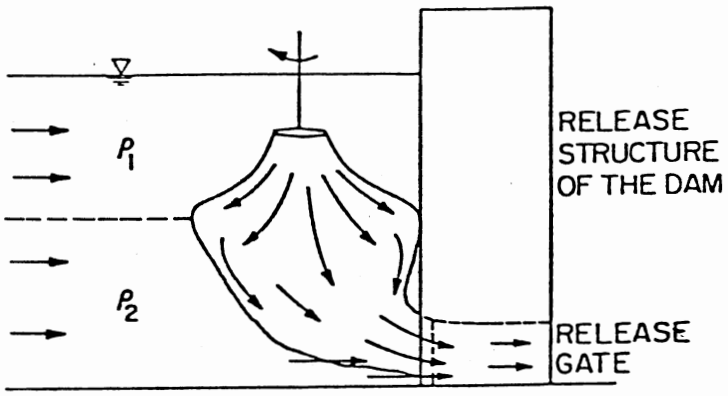


Figure 7. Schematic of a Typical Propeller Pump and the Flowfield Produced with Exit Flow via a Low Level Release Structure of the Dam (20).

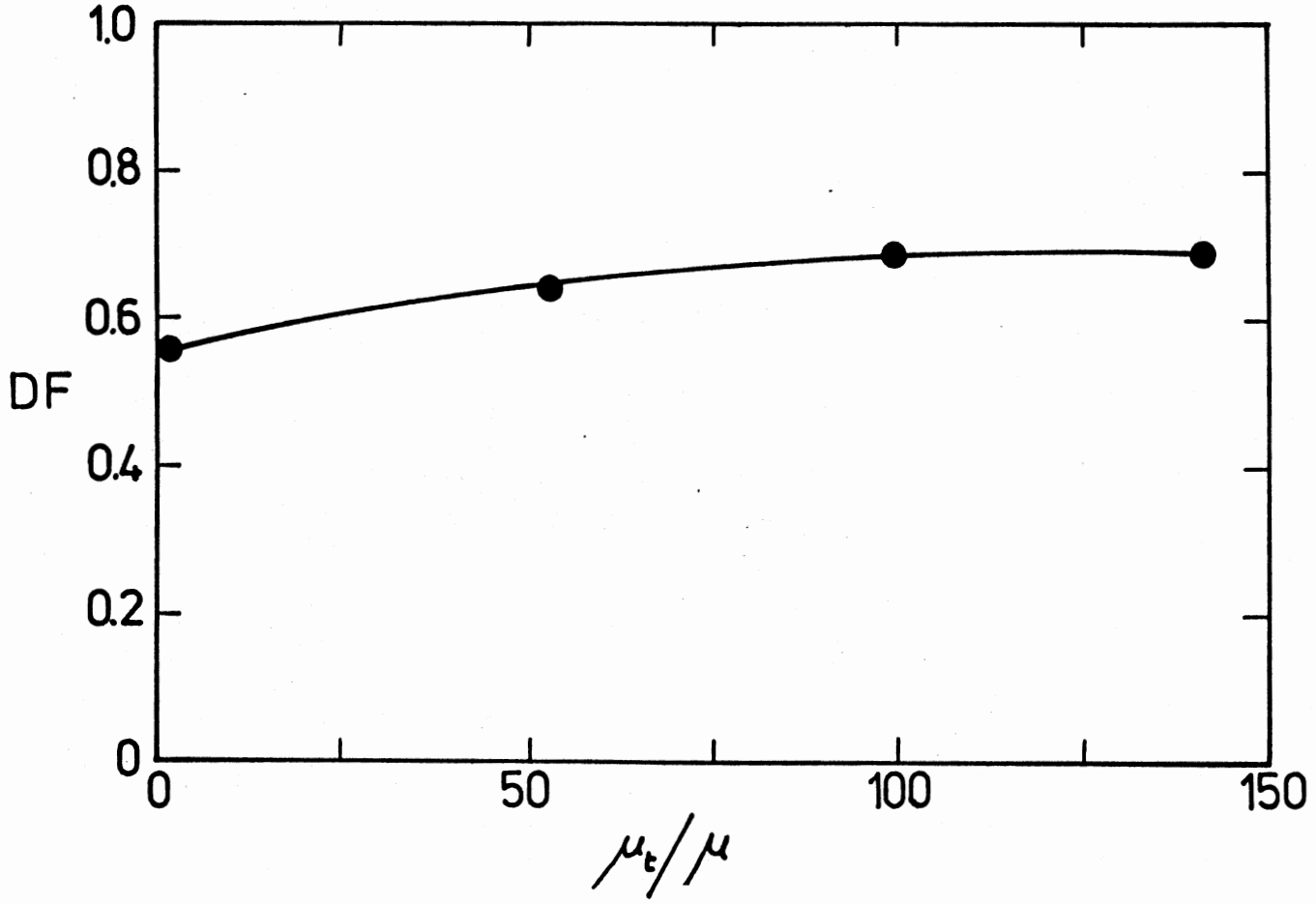


Figure 8. Predicted Effect of Turbulent Viscosity μ_t on Dilution Factor DF [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].

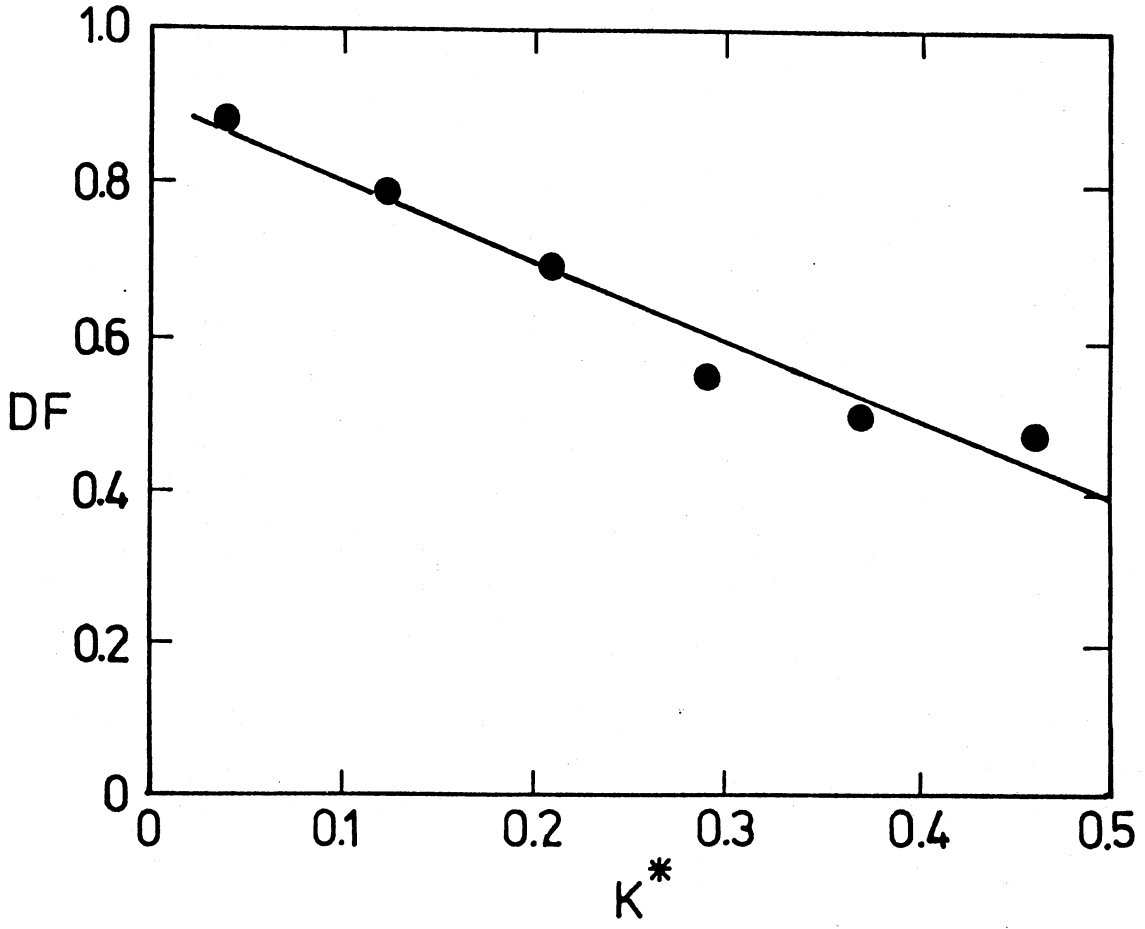


Figure 9. Predicted Dilution Factor DF as a Function of Nondimensional Horizontal Distance from Release Structure K* [Q* = 2.6, D* = 0.131, Frd = 2.0].

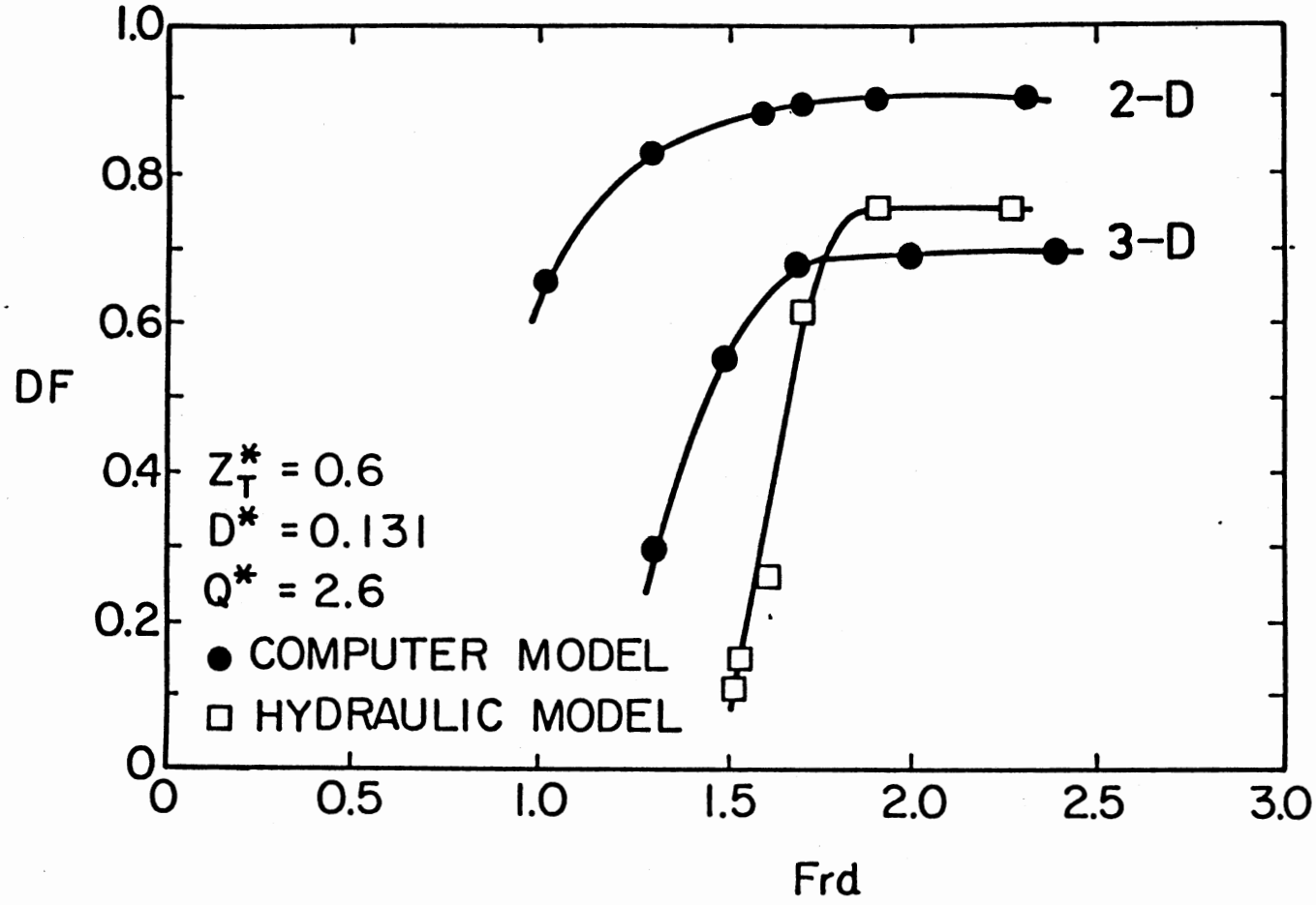


Figure 10. Dilution Factor as a Function of Froude Number F_{rd} (26) [K* = 0.211].

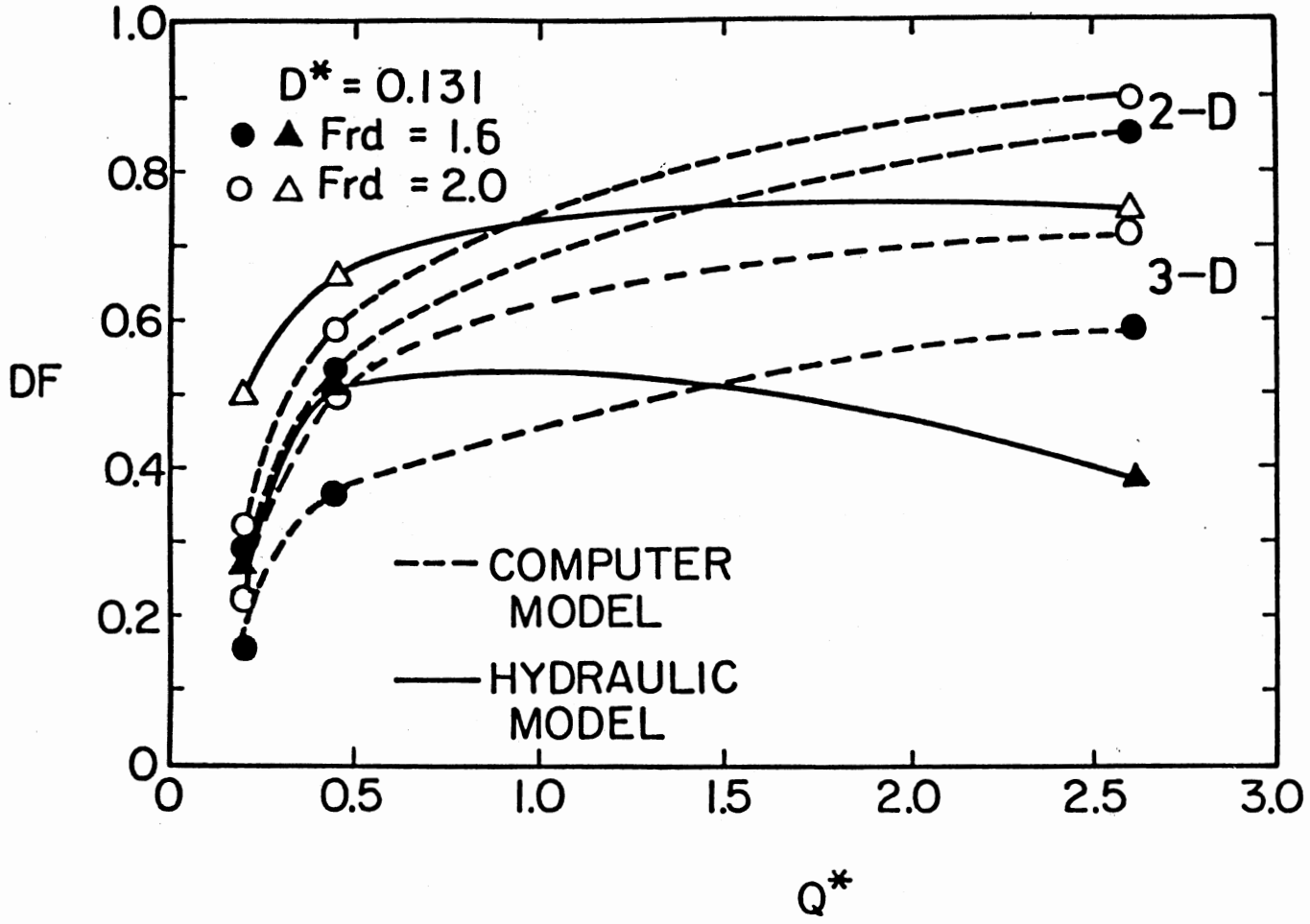


Figure 11. Dilution Factor as a Function of Nondimensional Flow Rate Q* (26) [K* = 0.211].

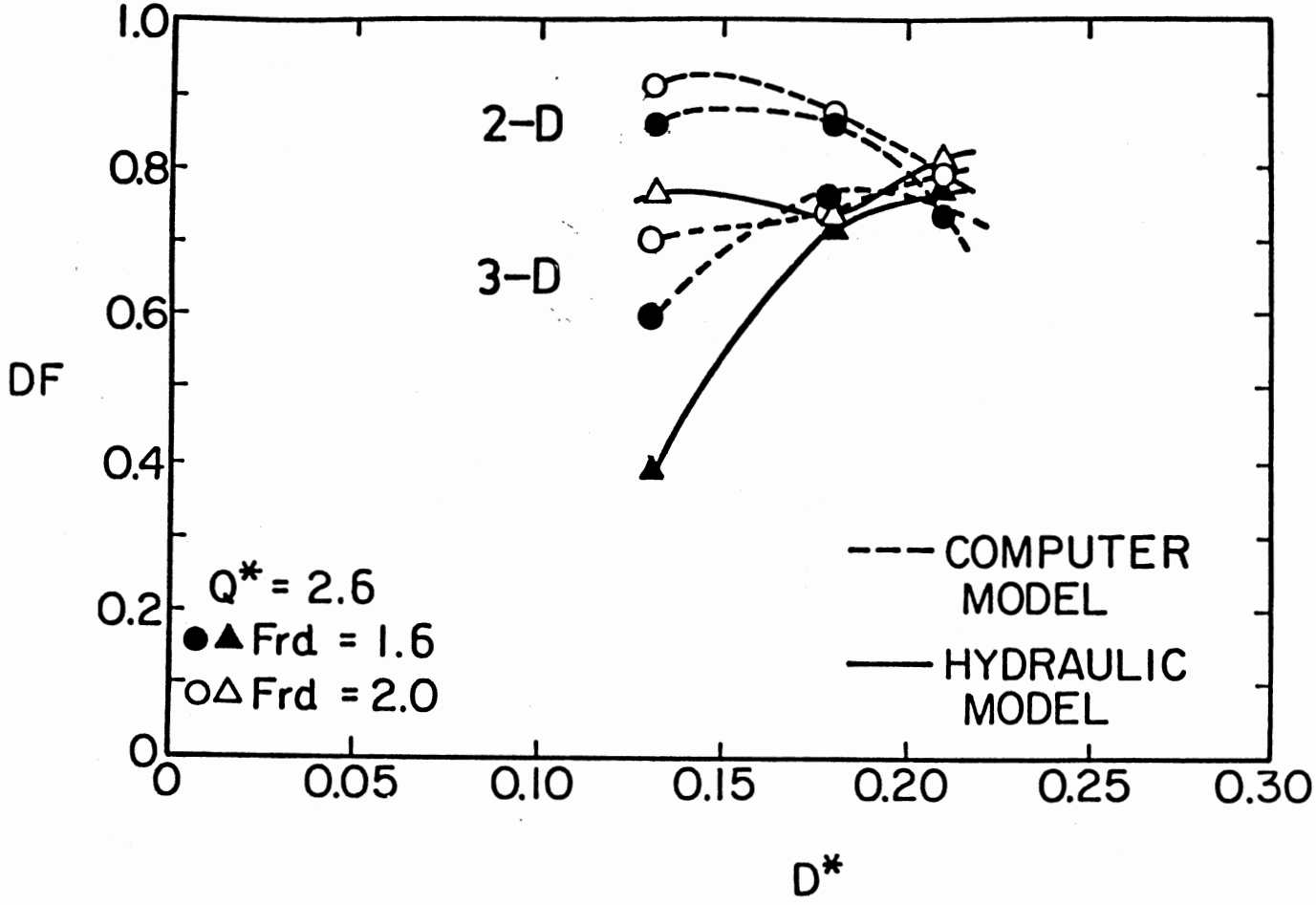


Figure 12. Dilution Factor as a Function of Nondimensional Propeller Diameter D* (26) [K* = 0.211].

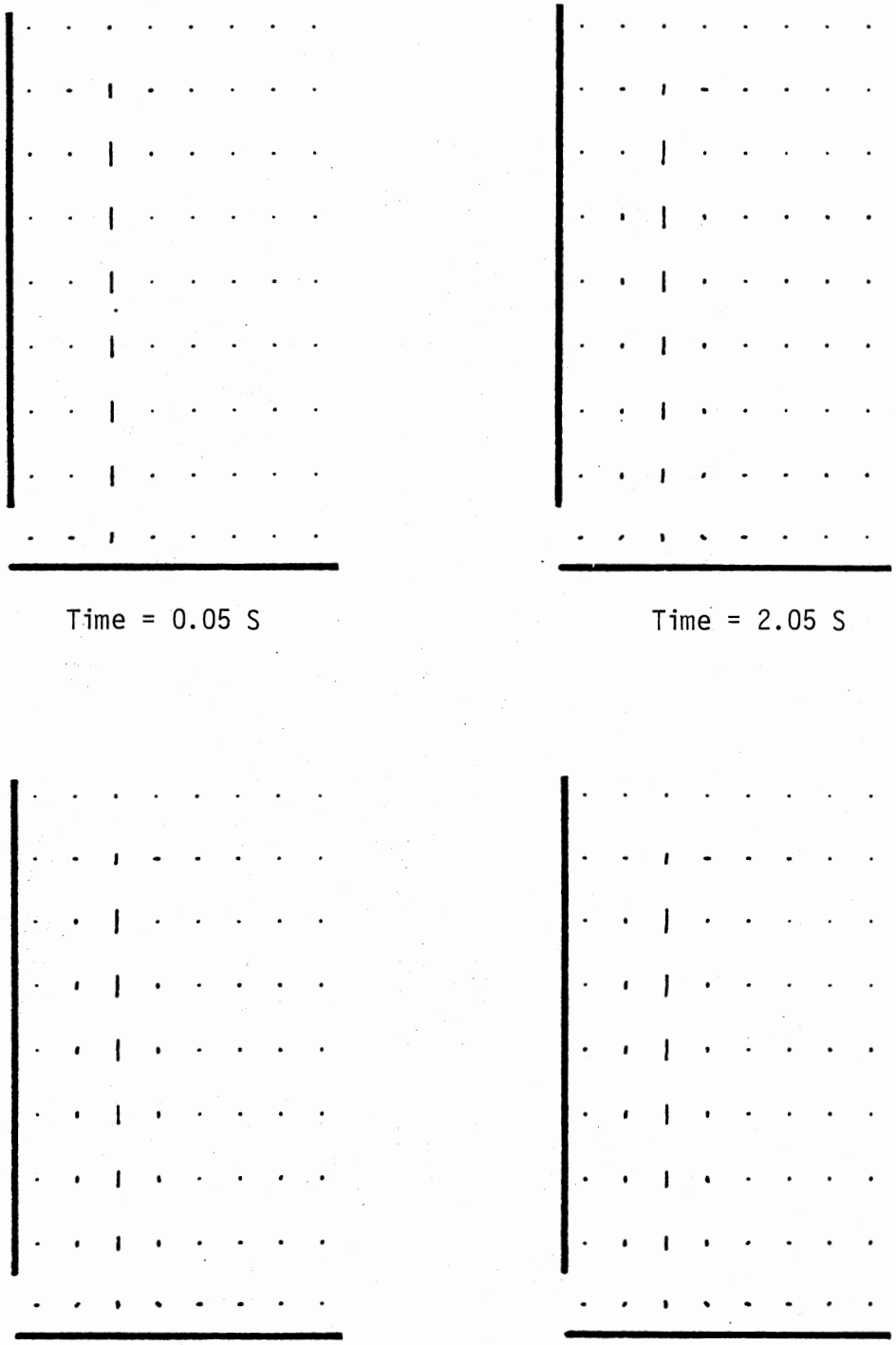
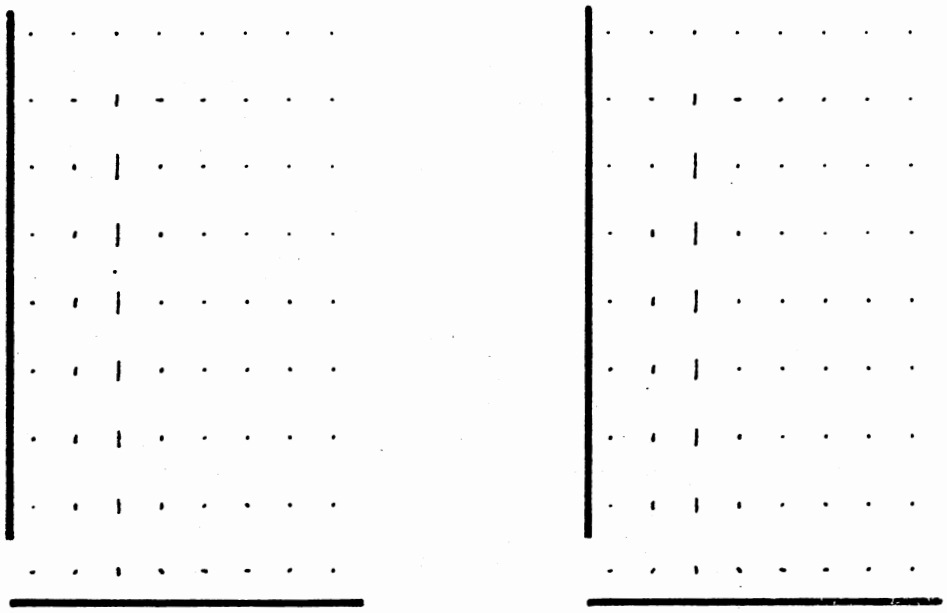
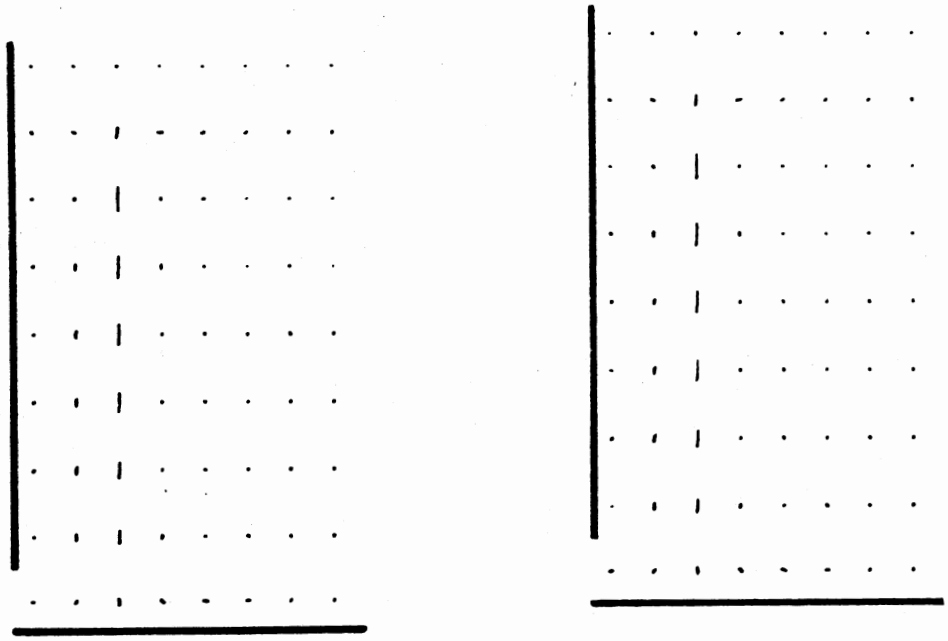


Figure 13. Velocity Vectors in zx-plane (at the Plane of Symmetry) Showing Magnitude and Direction of the Flow in the First Four Time Frames of the Laminar Predictions [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



Time = 8.05 S

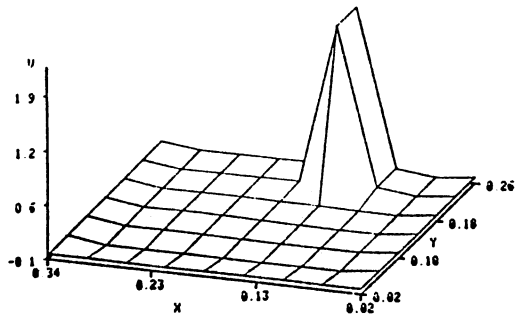
Time = 10.05 S



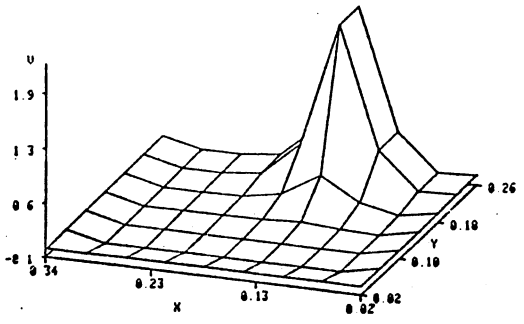
Time = 12.05 S

Time = 14.05 S

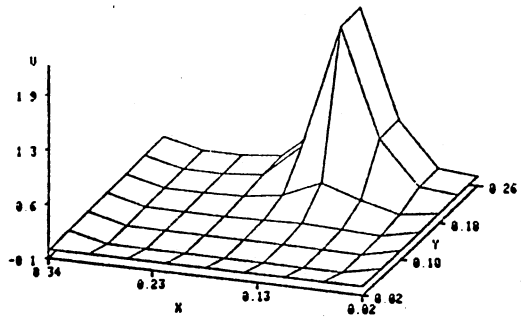
Figure 14. Velocity Vectors in zx-plane (at the Plane of Symmetry) Showing Magnitude and Direction of the Flow in the Last Four Time Frames of the Laminar Predictions [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



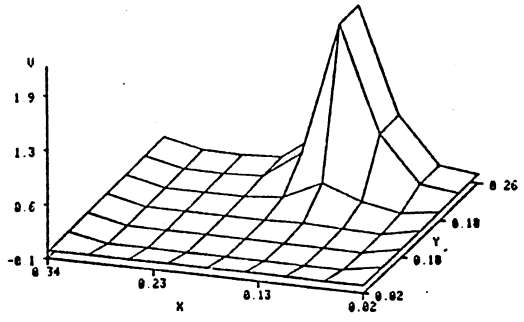
Time = 0.05 S



Time = 2.05 S

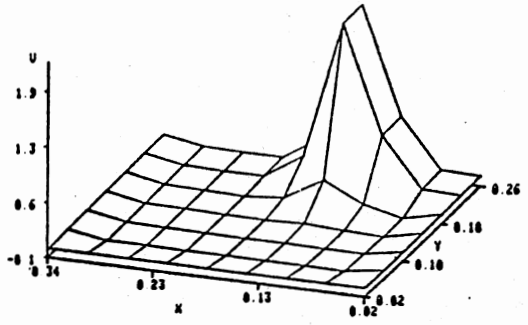


Time = 4.05 S

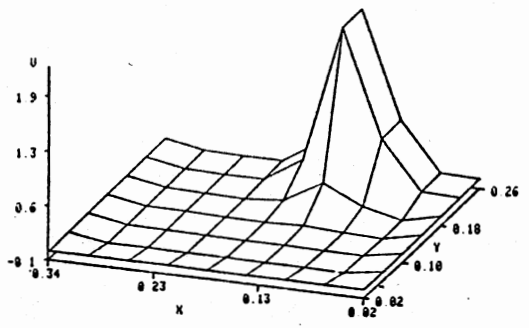


Time = 6.05 S

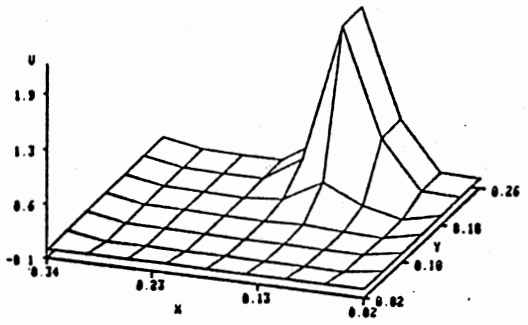
Figure 15. Three-Dimensional Representation of the Vertical Velocity at the Interface is Shown for the First Four Time Frames of the Laminar Predictions [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



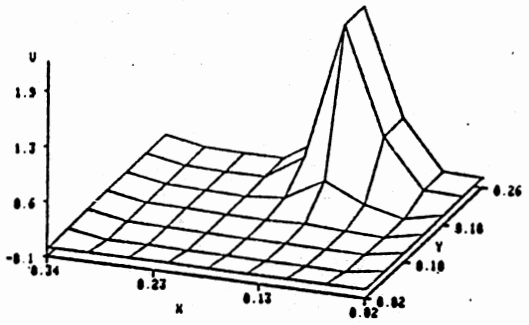
Time = 8.05 S



Time = 10.05 S



Time = 12.05 S



Time = 14.05 S

Figure 16. Three-Dimensional Representation of the Vertical Velocity at the Interface is Shown for the Last Four Time Frames of the Laminar Predictions [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].

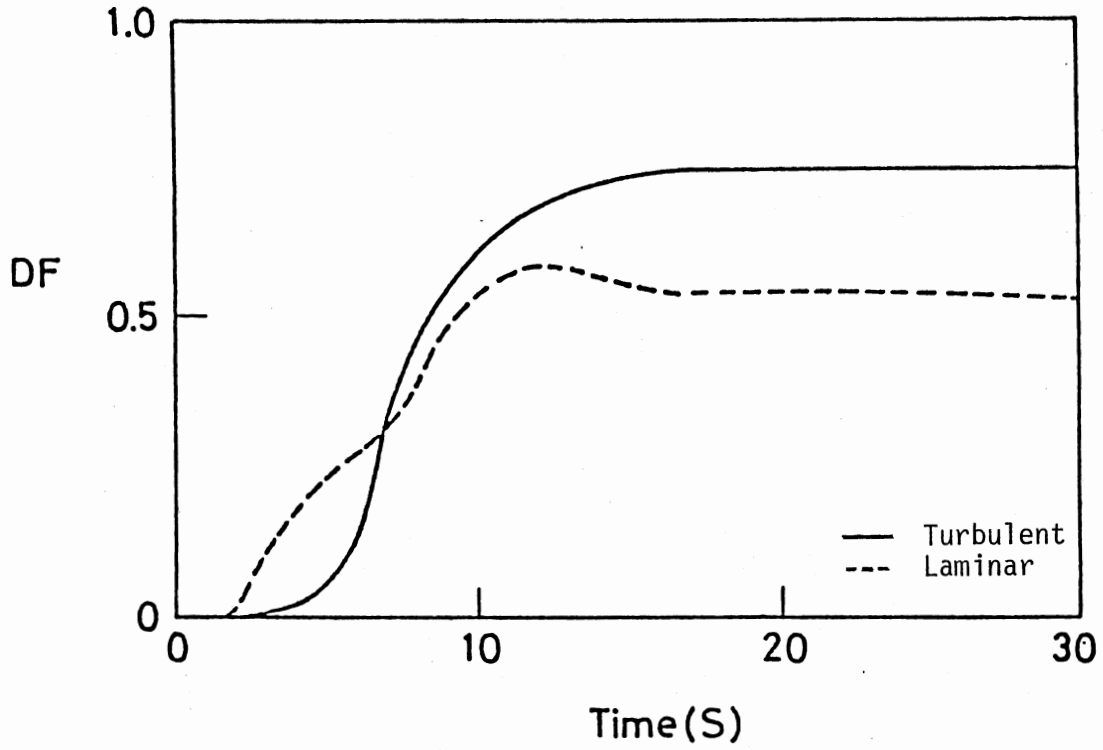


Figure 17. Prediction of Dilution Factor DF as a Function of Time [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].

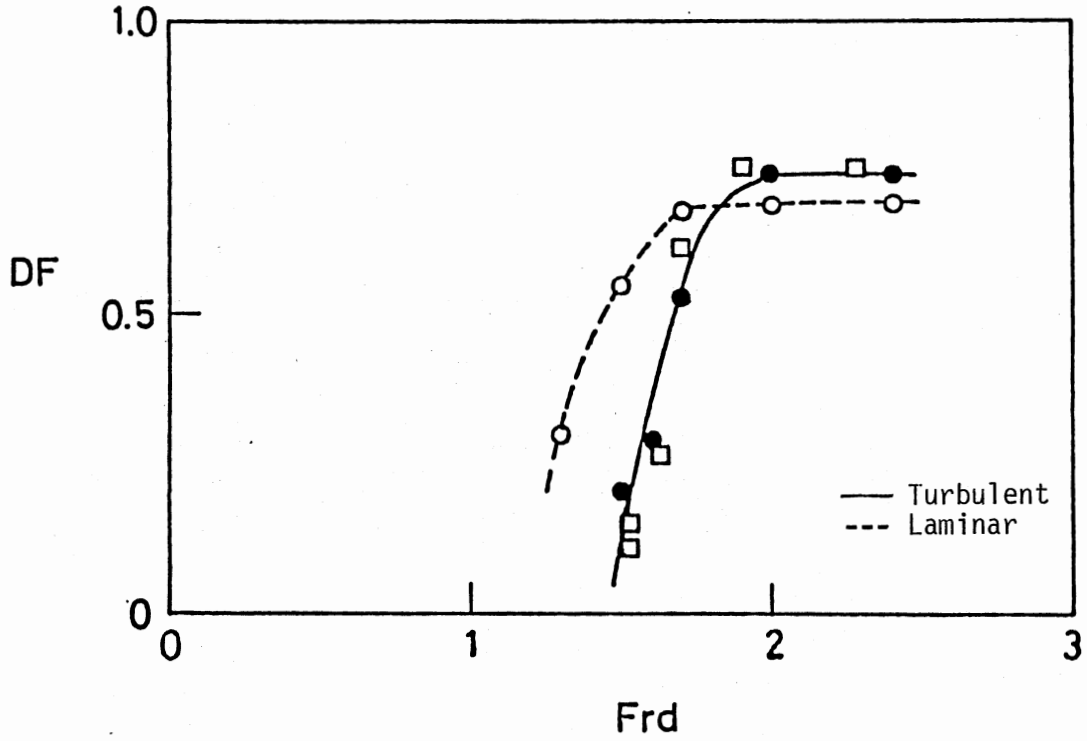


Figure 18. Dilution Factor as a Function of Froude Number F_{rd} (80) [Q* = 2.6, D* = 0.131, K* = 0.211].

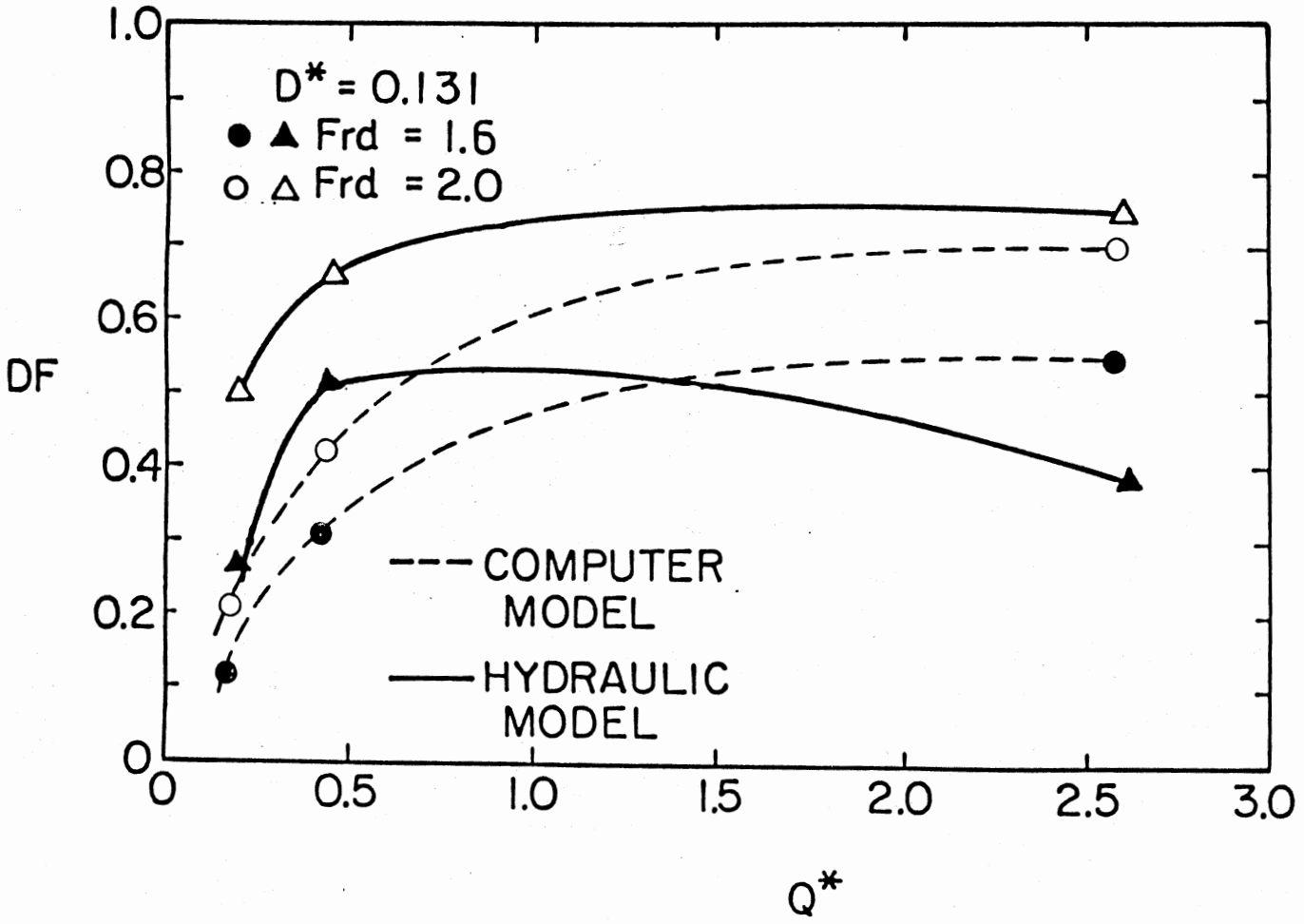


Figure 19. Dilution Factor as a Function of Nondimensional Flow Rate Q* [Turbulent Predictions and Experiments (80)] [K* = 0.211].

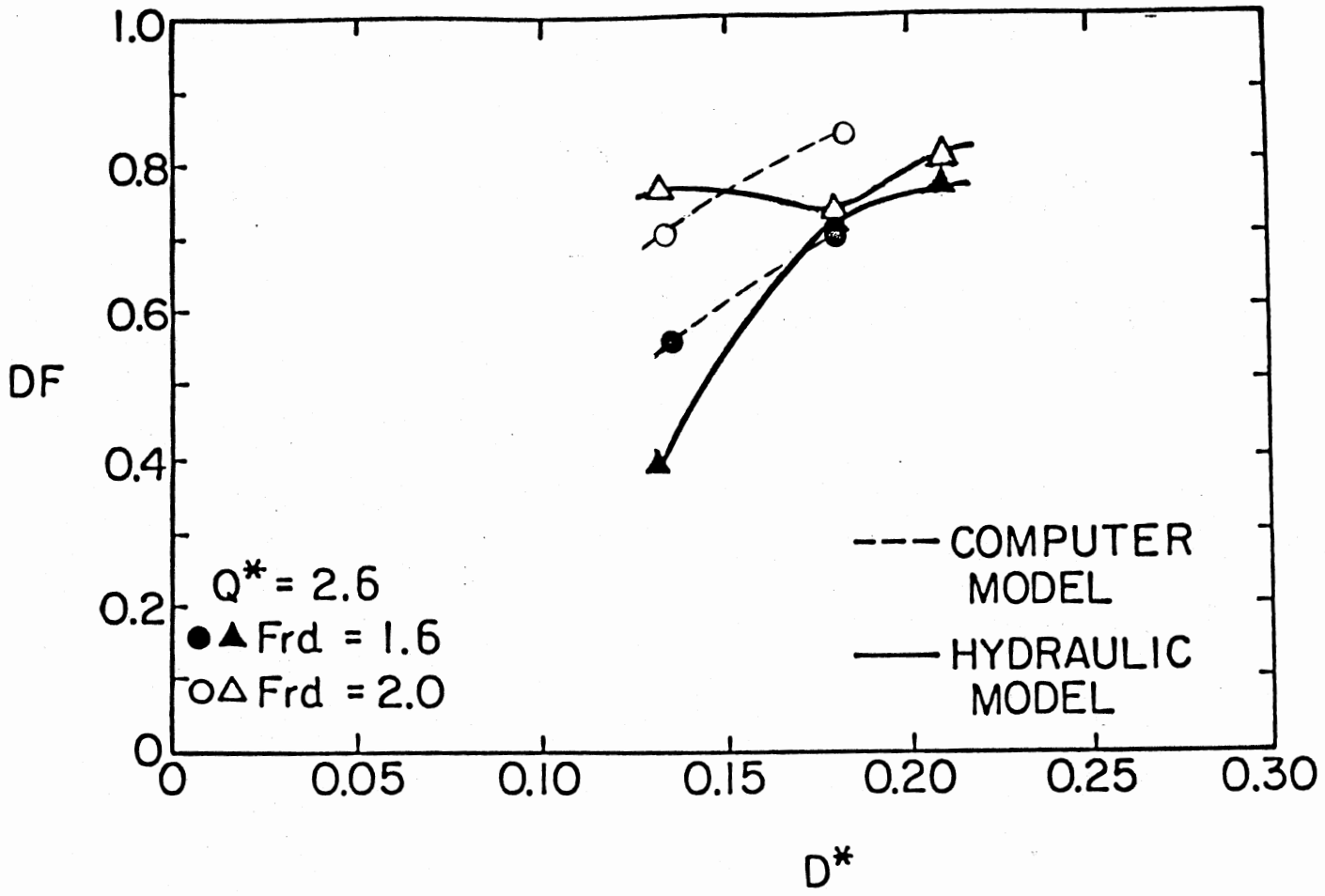


Figure 20. Dilution Factor as a Function of Nondimensional Propeller Diameter D* [Turbulent Predictions and Experiments (80)] [K* = 0.211].

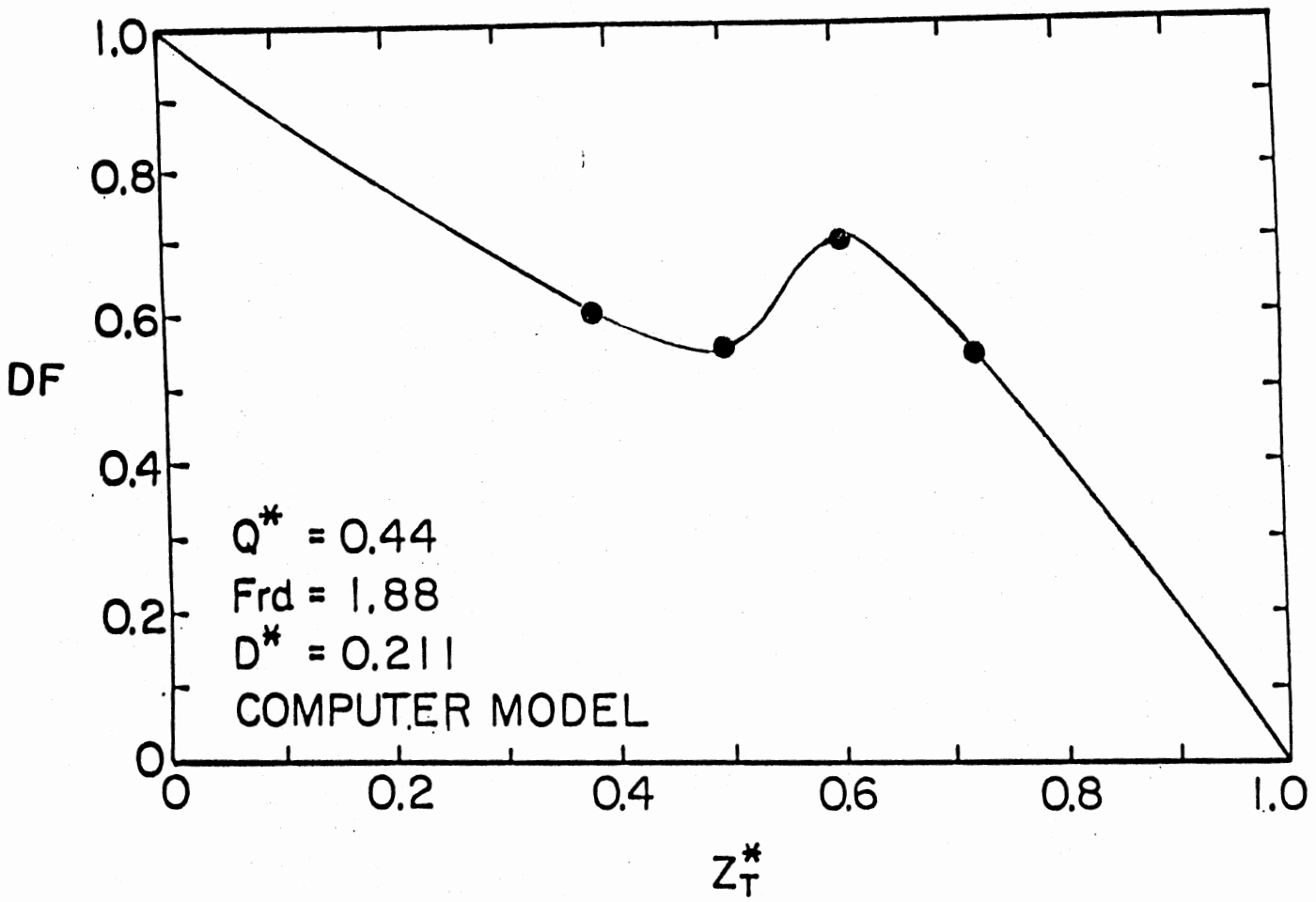


Figure 21. Dilution Factor as a Function of the Normalized Metalimnion Location (Interface) Z_T^* (Turbulent Predictions) [K* = 0.211].

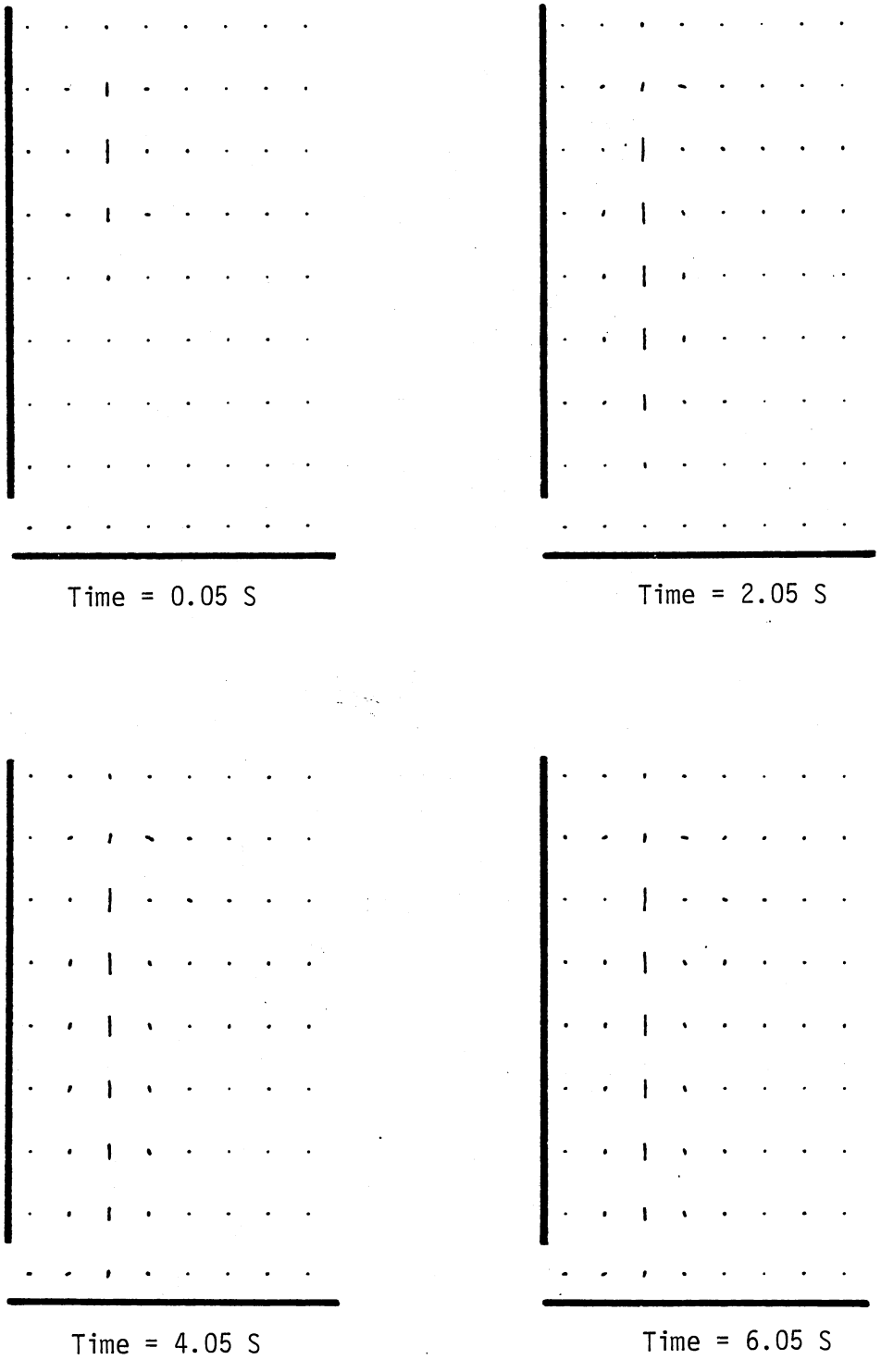


Figure 22. Velocity Vectors in xz-plane (at the Plane of Symmetry) Showing Magnitude and Direction of the Flow in the First Four Time Frames of the Turbulent Predictions [Q = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].

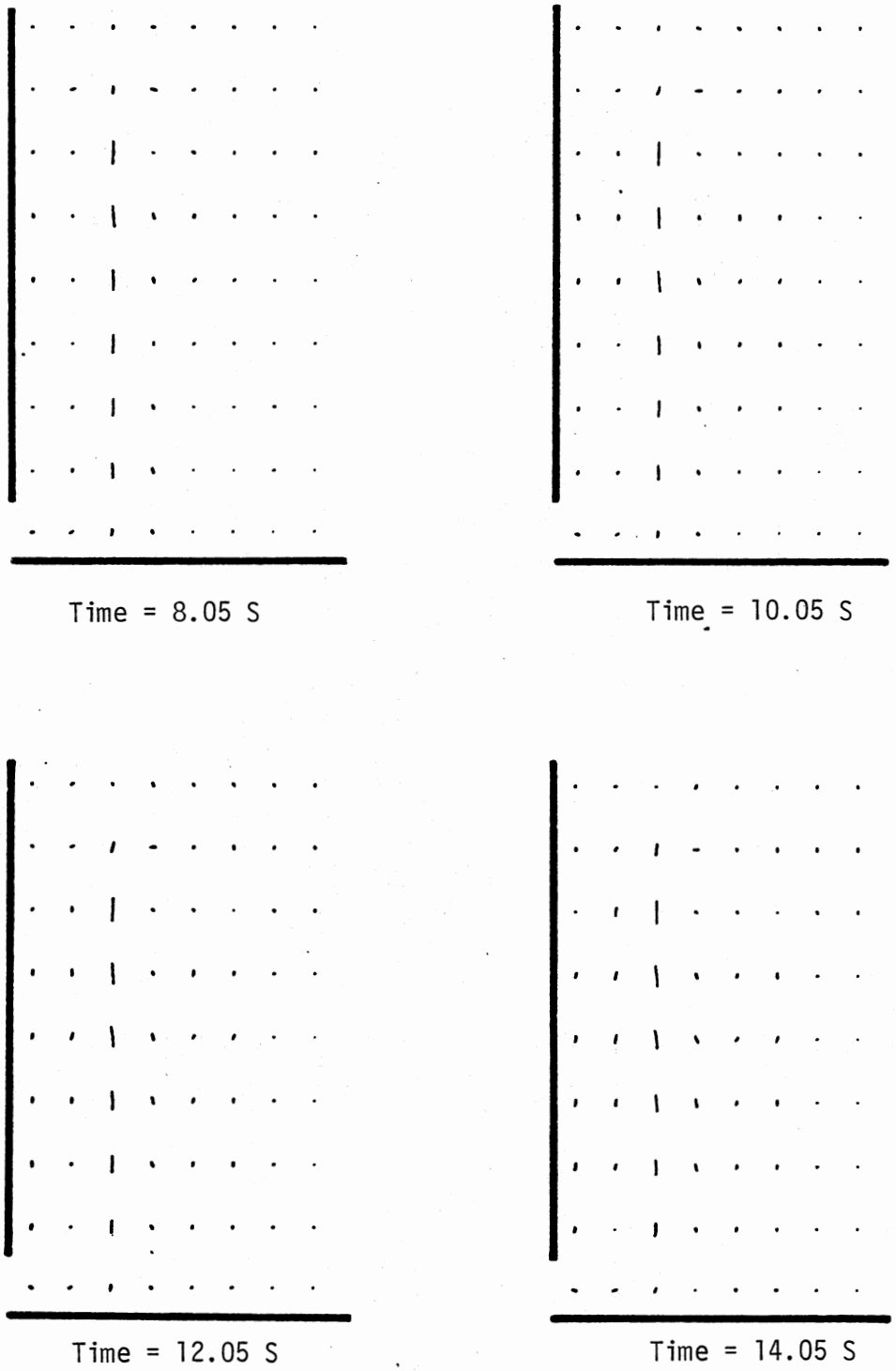
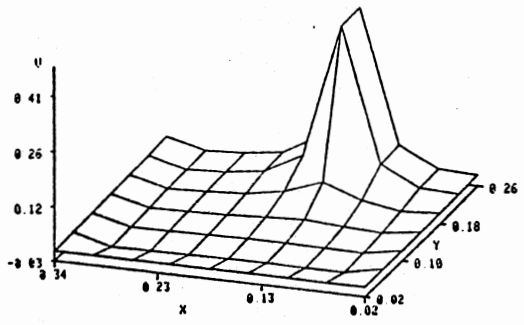
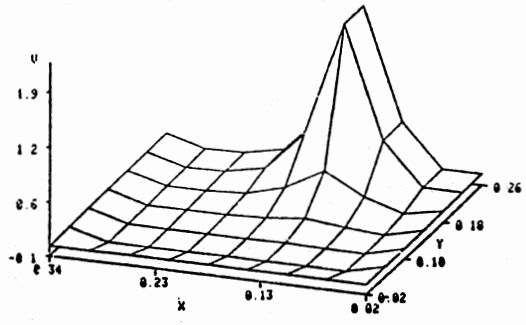


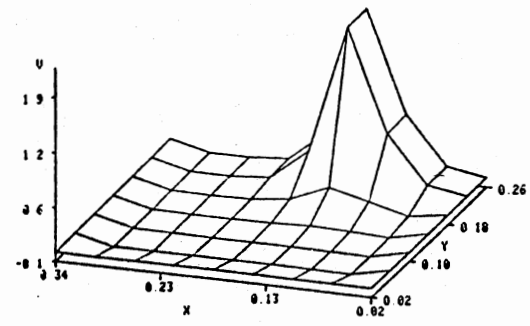
Figure 23. Velocity Vectors in xz-plane (at the Plane of Symmetry) Showing Magntiude and Direction of the Flow in the Last Four Time Frames of the Turbulent Predictions [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



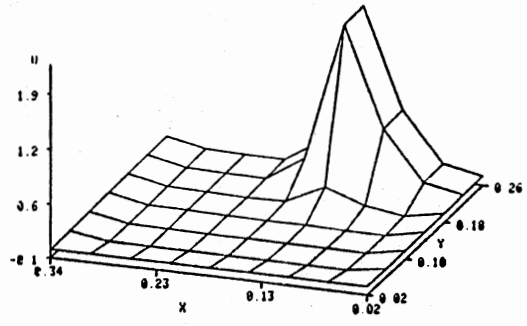
Time = 0.05 S



Time = 2.05 S

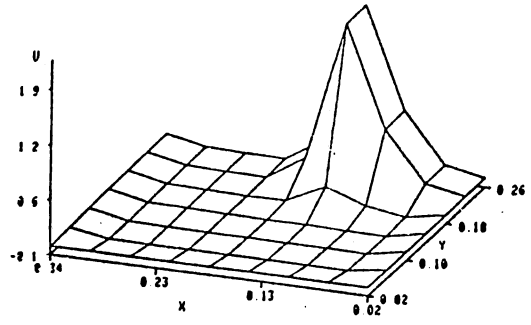


Time = 4.05 S

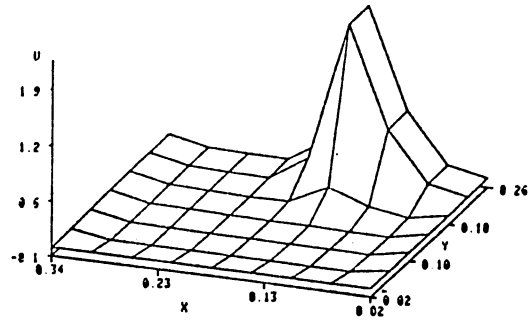


Time = 6.05 S

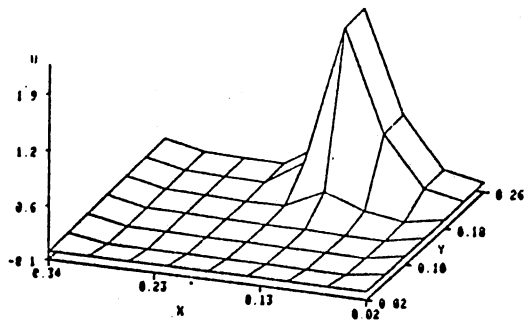
Figure 24. Three-Dimensional Representation of the Vertical Velocity at the Interface is Shown for the First Four Time Frames of the Turbulent Predictions (Case 1) [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



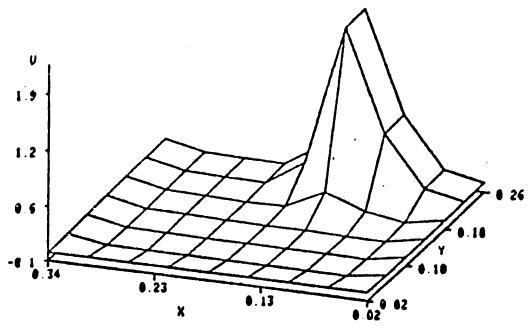
Time = 8.05 S



Time = 10.05 S

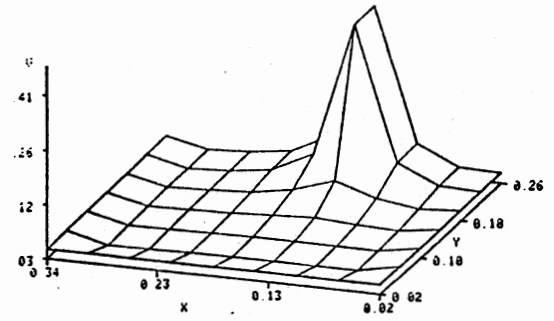


Time = 12.05 S

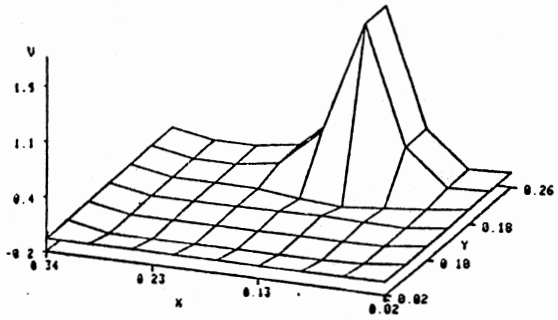


Time = 14.05 S

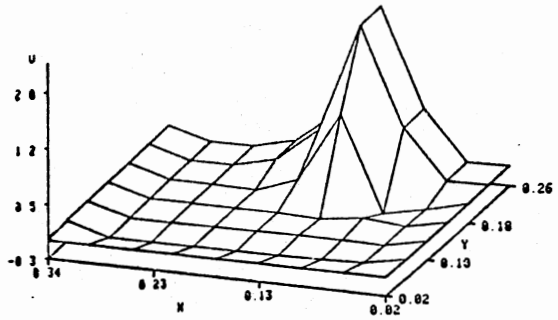
Figure 25. Three-Dimensional Representation of the Vertical Velocity at the Interface is Shown for the Last Four Time Frames of the Turbulent Predictions (Case 1) [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



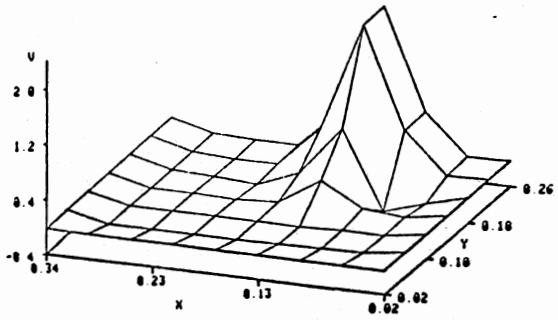
Time = 0.05 S



Time = 2.05 S

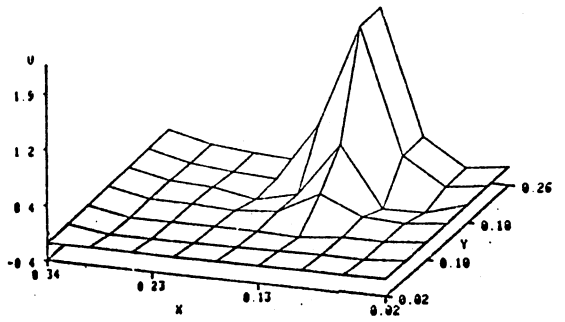


Time = 4.05 S

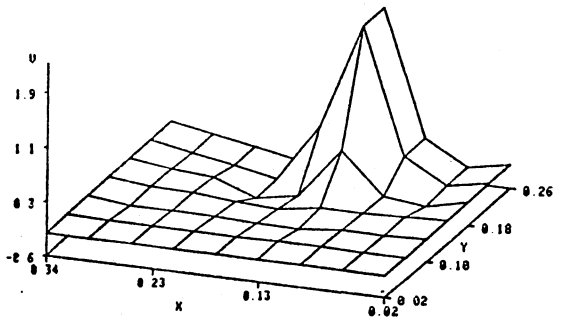


Time = 6.05 S

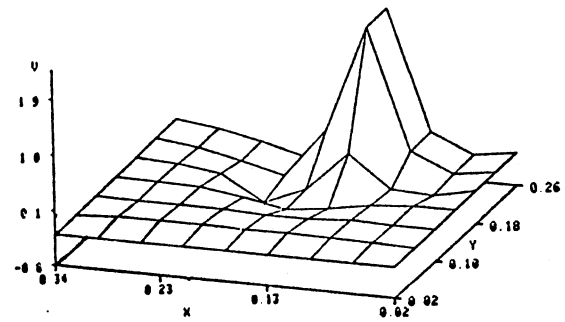
Figure 26. Three-Dimensional Representation of the Vertical Velocity at the Interface is Shown for the First Four Time Frames of the Turbulent Predictions (Case 2) [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



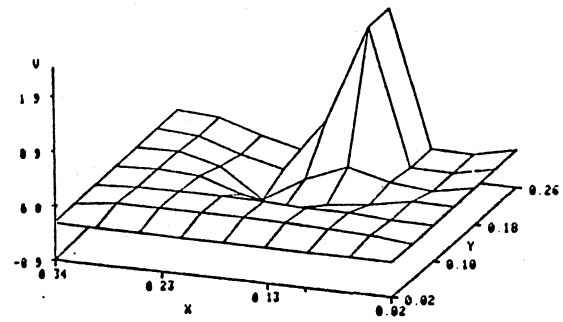
Time = 8.05 S



Time = 10.05 S

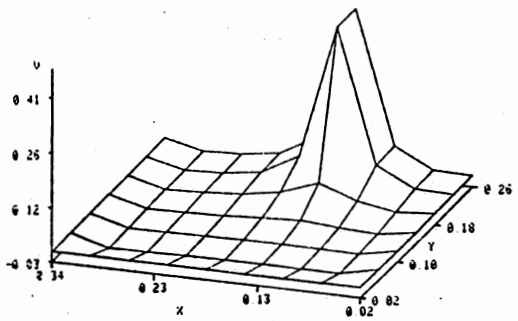


Time = 12.05 S

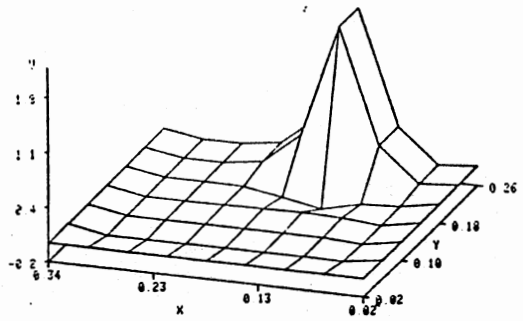


Time = 14.05 S

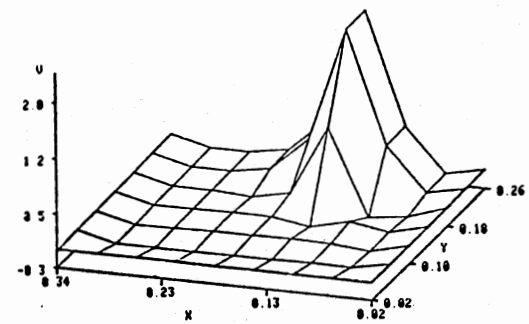
Figure 27. Three-Dimensional Representation of the Vertical Velocity at the Interface is Shown for the Last Four Time Frames of the Turbulent Predictions (Case 2) [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



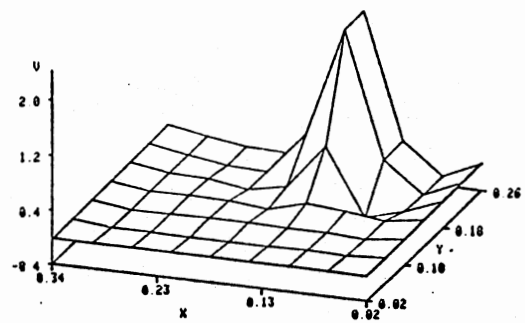
Time = 0.05 S



Time = 2.05 S

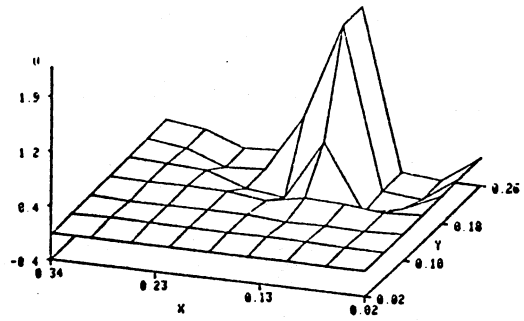


Time = 4.05 S

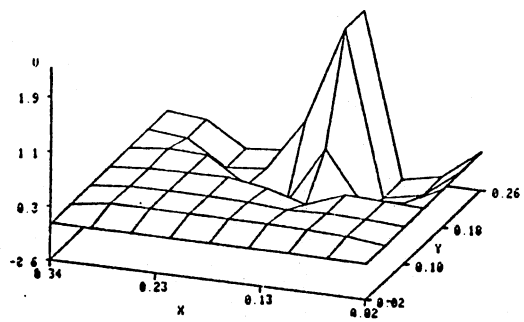


Time = 6.05 S

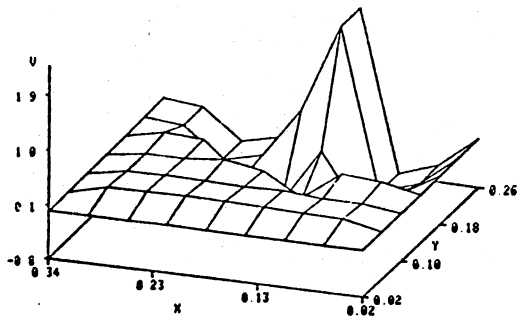
Figure 28. Three-Dimensional Representation of the Vertical Velocity of the Interface is Shown for the First Four Time Frames of the Turbulent Predictions (Case 3)[Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].



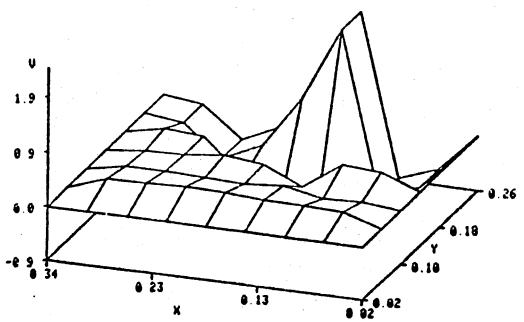
Time = 8.05 S



Time = 10.05 S



Time = 12.05 S



Time = 14.05 S

Figure 29. Three-Dimensional Representation of the Vertical Velocity at the Interface is Shown for the Last Four Time Frames of the Turbulent Predictions (Case 3) [Q* = 2.6, D* = 0.131, Frd = 2.0, K* = 0.211].

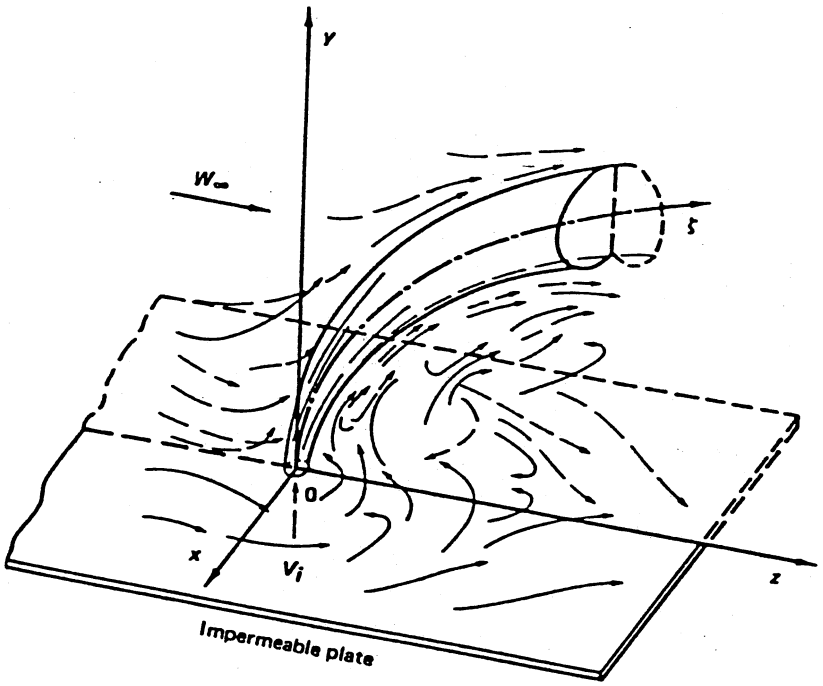


Figure 30. Schematic of the Deflected-Jet Problem (28).

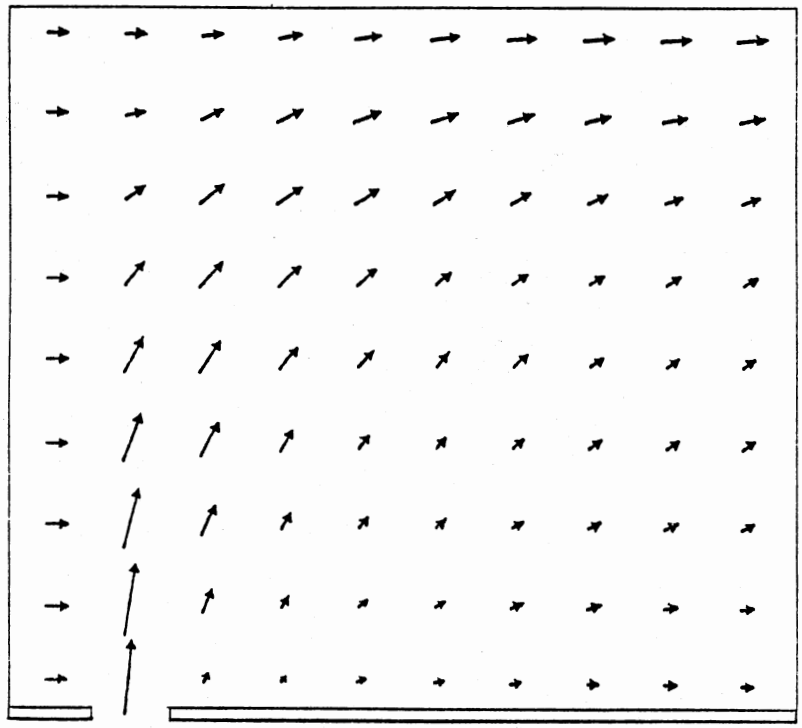


Figure 31. Velocity Vectors in yz-plane at x = 0 showing Magnitude and Direction of the Jet [for Velocity Ratio R = u_j/u_{in} = 6].

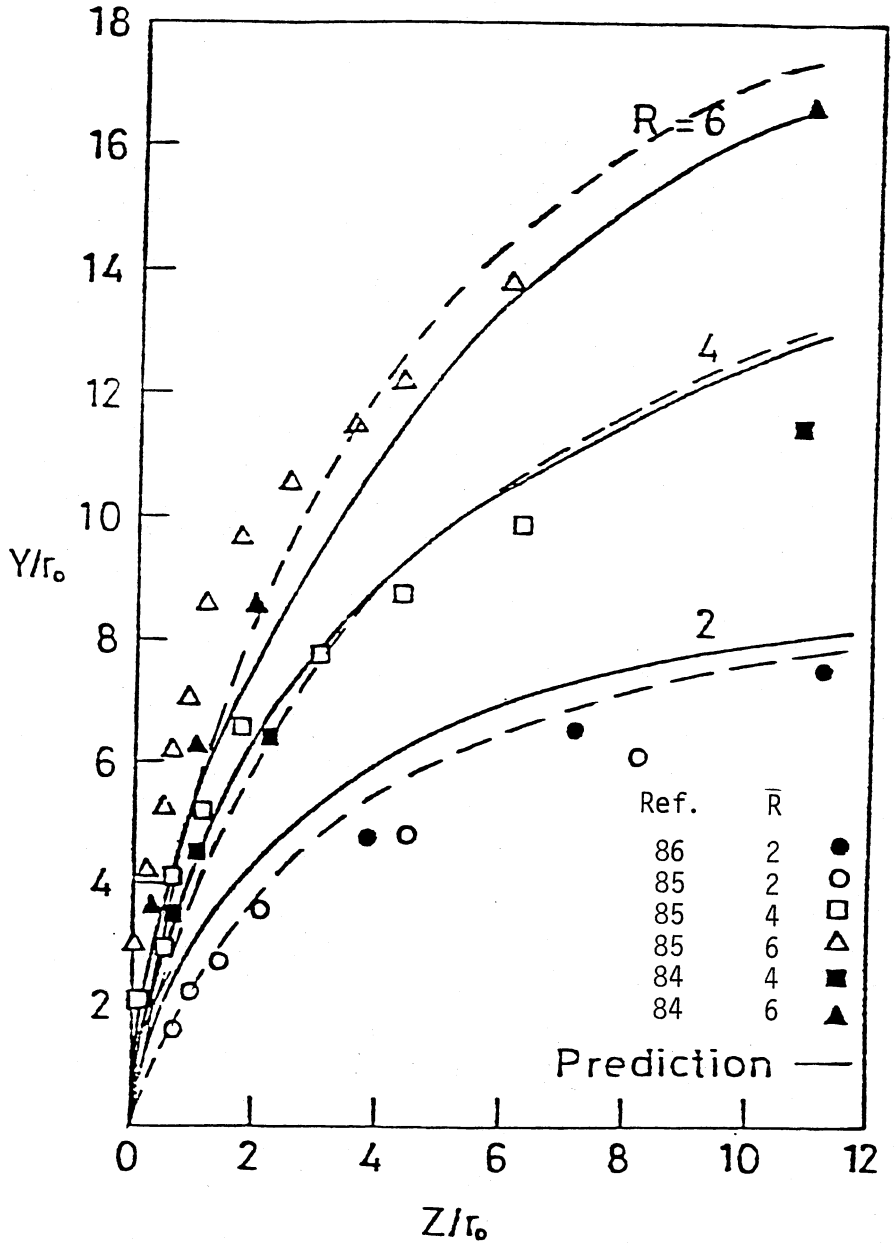


Figure 32. Location of the Jet Centerline for Different Jet-to-Crossflow Velocity Ratios R = u_j/u_{in} [--- Predictions (38)].

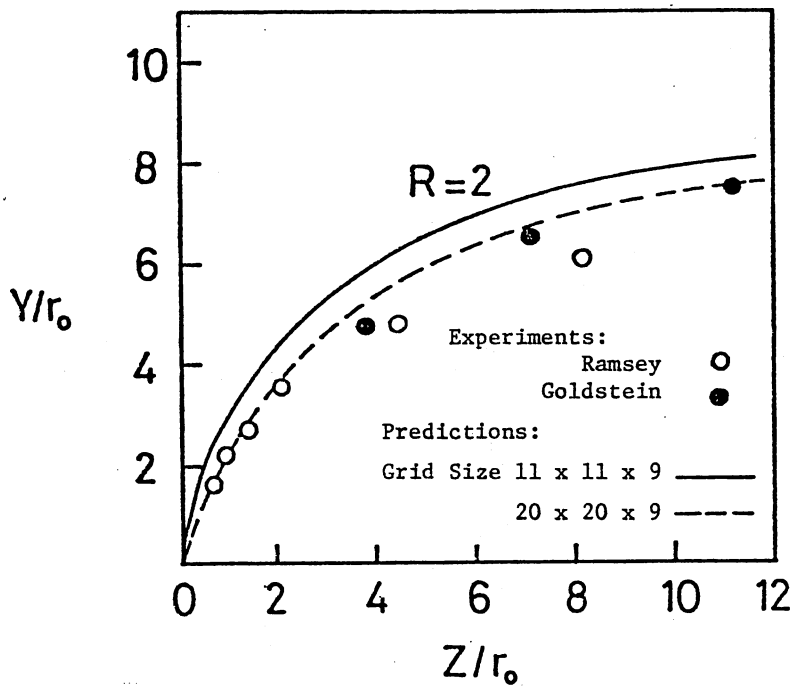


Figure 33. Location of the Jet Centerline with Jet-to-Crossflow Velocity Ratio R = 2, Predicted with Two Grid Sizes.

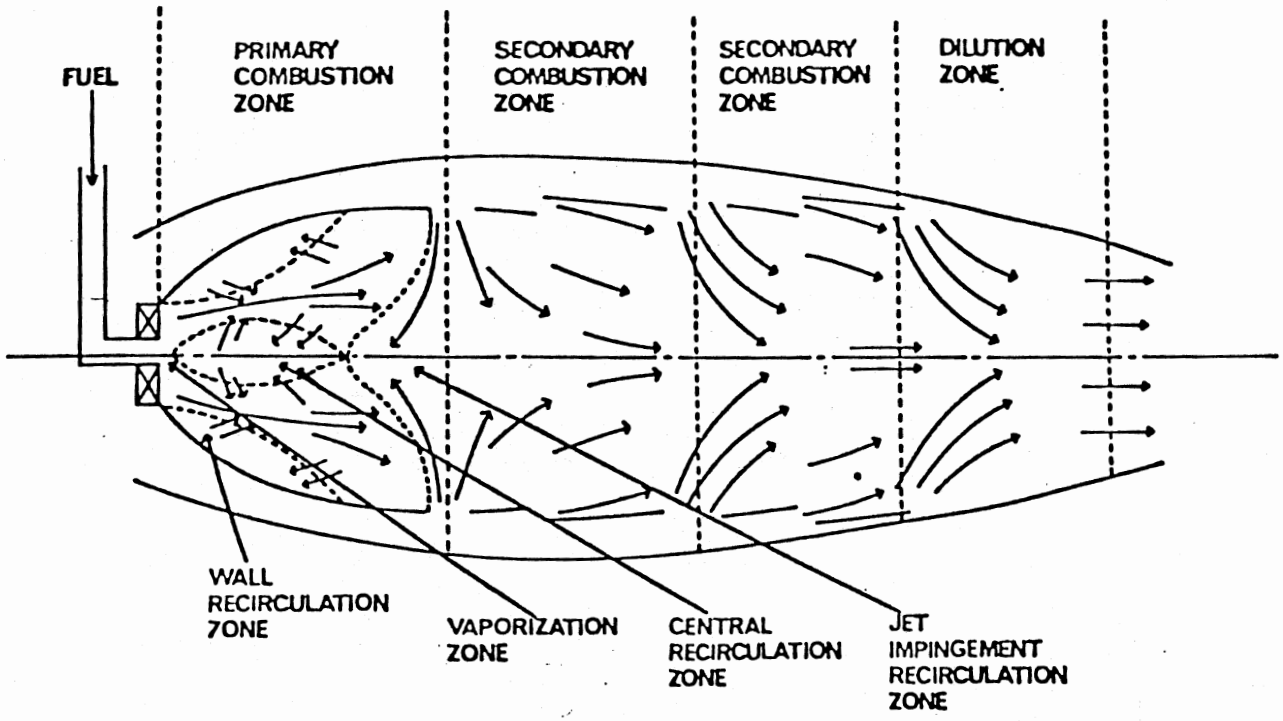
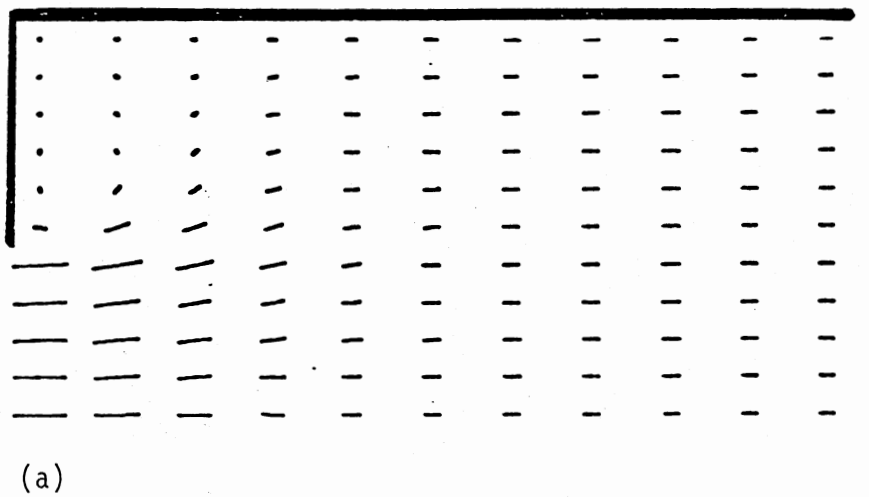
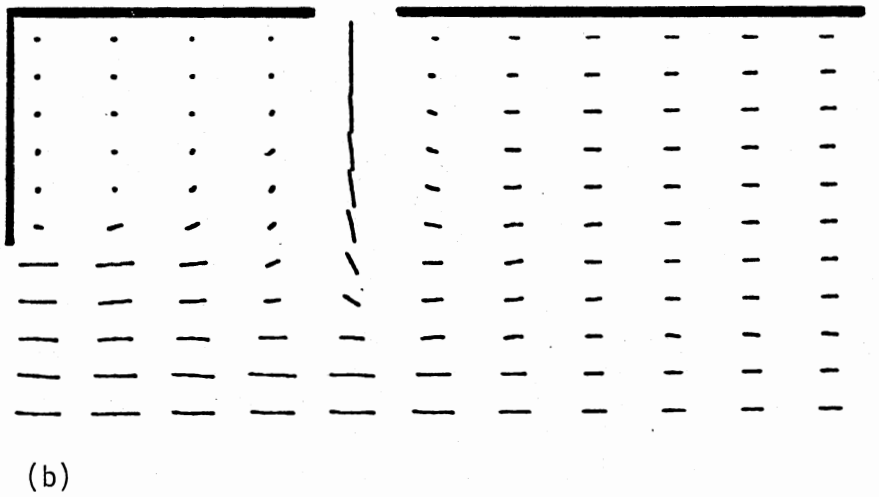


Figure 34. Combustor Schematic Showing Primary Combustion, Secondary Combustion and Dilution Zones (88).

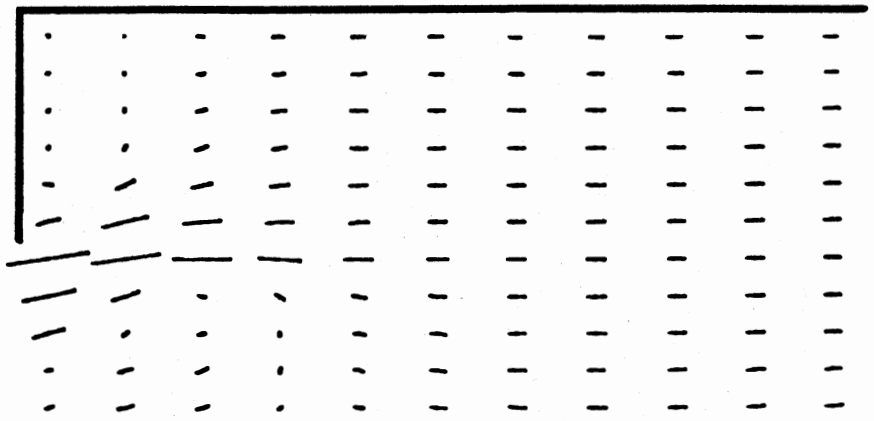


(a)

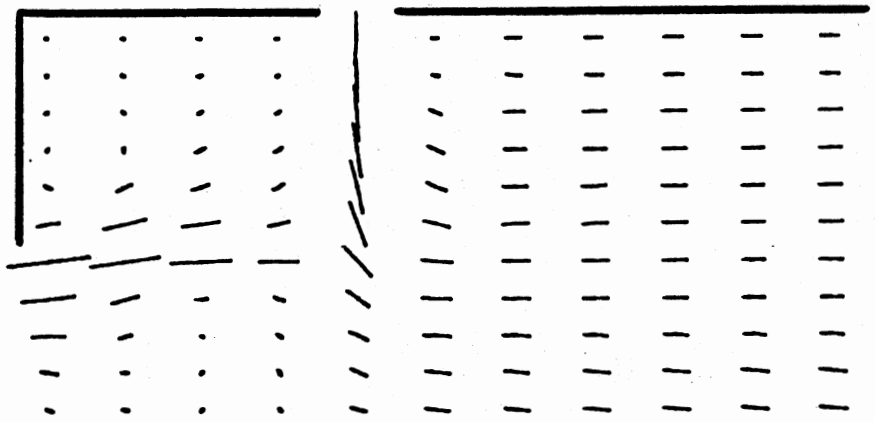


(b)

Figure 35. Velocity Vectors Showing Magnirude and Directions with Swirl Vane Angle $\phi = 0°$ and Dilution Jet Velocity: (a) $u_d = 0.0$, (b) $u_d = v_o$.



(a)



(b)

Figure 36. Velocity Vectors Showing Magnitude and Directions with Swirl Vane Angle ϕ = 45°, Low Reynolds Number, and with a Dilution Jet Velocity (a) U_d = 0.0, (b) $U_d = V_o$.

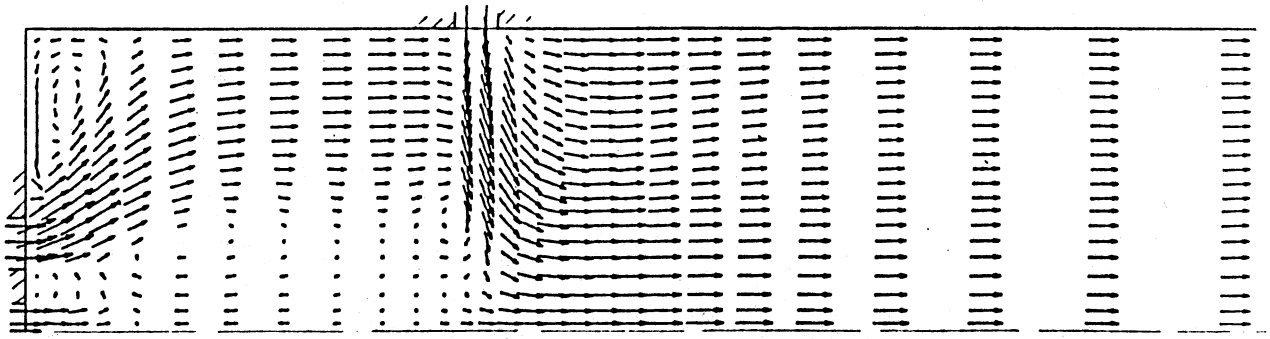
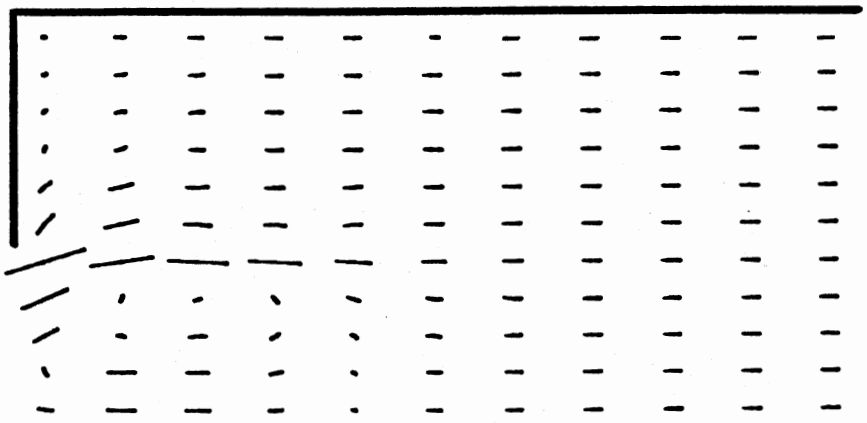
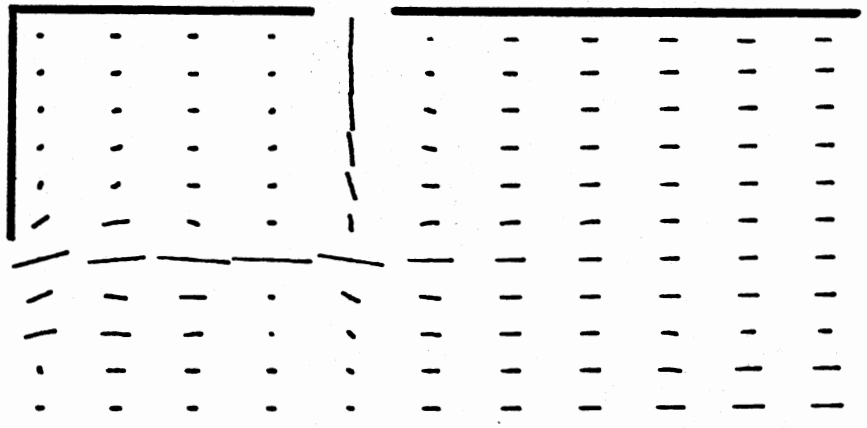


Figure 37. Velocity Vectors Showing Magnitude and Directions with Swirl Vane Angle ϕ = 45° and Dilution Jet Velocity $U_d = V_o$ for Low Reynolds Number Flow with Combustion (40). (This chamber has a Different Aspect Ratio Compared to the Previous Shown Chambers).



(a)



(b)

Figure 38. Velocity Vectors Showing Magnitude and Directions with Swirl Vane Angle ϕ = 45°, High Reynolds Number, and with a Dilution Jet Velocity (a) U_d = 0.0, (b) $U_d = V_o$.

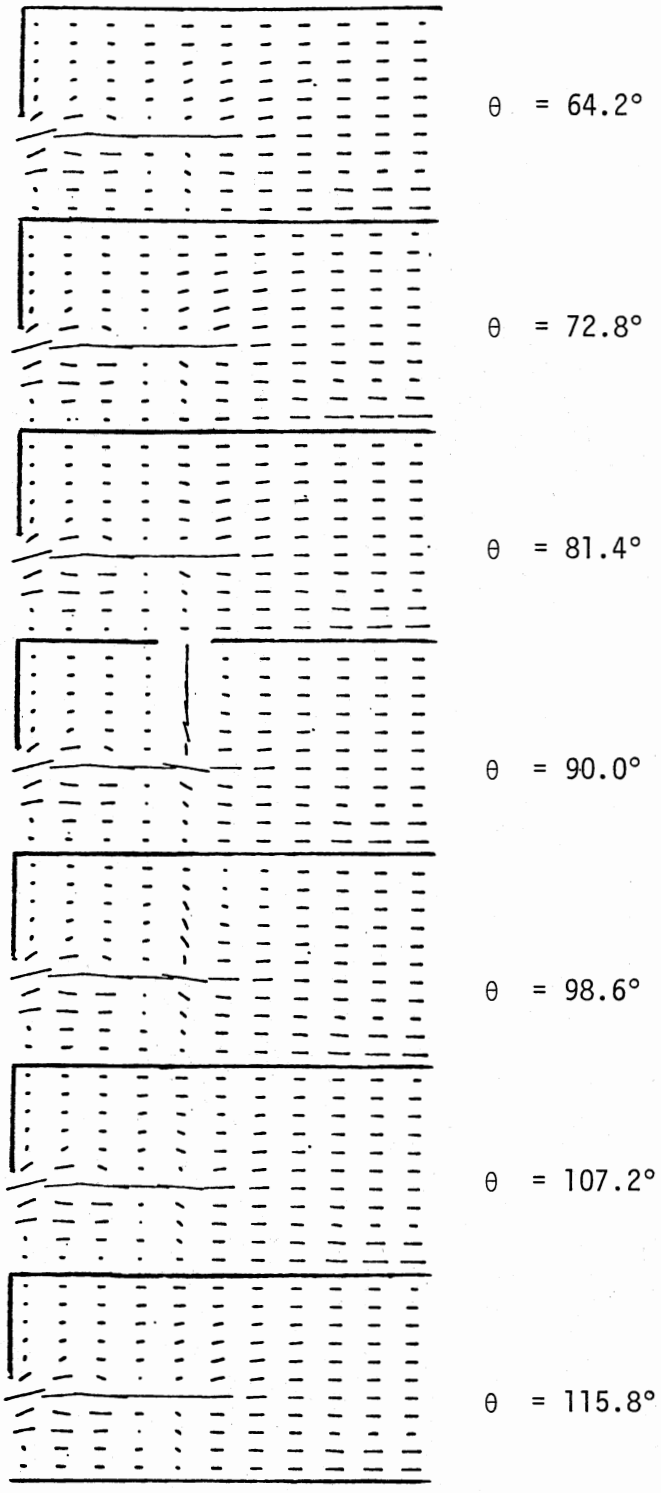


Figure 39. Velocity Vectors Showing Magnitude and Directions with Swirl Vane Angle ϕ = 45° and Dilution Jet Velocity $u_d = v_o$ in θ-planes (high Re number).

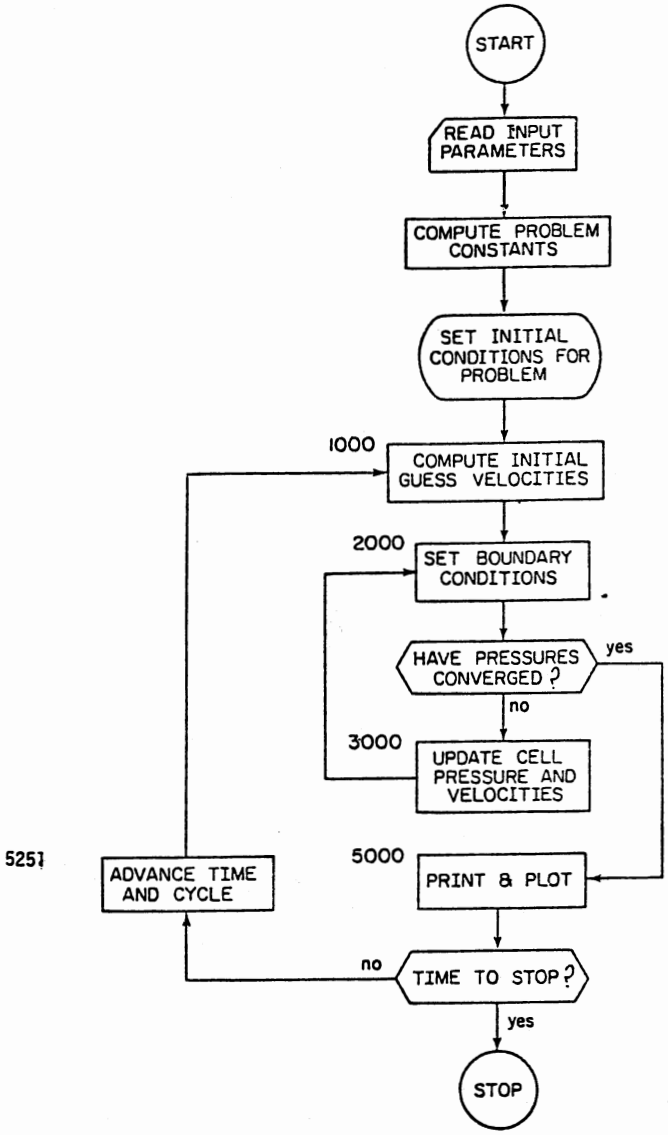


Figure 40. Flow Chart of the Computer Program (3).

APPENDIX C

USER'S GUIDE TO COMPUTER PROGRAMS

This User's Guide presents the important Fortram variables and defines them. It also gives a test case with the standard conditions used in order to demonstrate a sample run of the program. Changing the dimensions, number of cells, fluid properties, input velocities and their locations can be easily handled by changing the corresponding defined Fortran variables. Other changes such as considering a different geometry (that includes a sloping wall for example) require an alteration in the boundary condition section in the program. Such changes could be easily made by referring to the original SOLA report which provides specific instruction for boundary condition alterations. Higher velocity values (subsonic) could be accommodated easily in the code provided that the time-step is reduced by the same factor.

The program output gives the velocity, pressure, mass fraction m_1, and density values everywhere in the solution domain. The value of the dilution factor (in the first application) is given in the beginning of every cycle along with the time in seconds. The cycle [number of forward time-steps] and iterations [number of iterations for pressure and velocity correction at each time-step] are also given. It should be mentioned here that the steady state conditions have been obtained before computations cease at a cycle number of 300.

A good indication of convergence is the iteration number. When it approaches a small value (close to one) this means that the flow is approaching steady state, as the momentum forward time-steps are then

not disturbing the flowfield as measured by lack of mass continuity. The program's flow chart is shown in Figure 40.

1. Local Destratification of Reservoirs Application

A description is given of the important points needed in order to use the first of the programs listed in Appendix D.

1.1 Important Fortran Variables

These parameter values are all supplied in the first part of the computer program in lines 1 to 32.

IP — Gives the propeller location in x-direction (measured from the Release structure)

JP — Gives the propeller location in the y-direction

KP — Gives the propeller location in the z-direction (measured from the bottom). Used to specify propeller depth, L = (KM1-KP)ΔZ, m

KR — Gives Release Gate height, h = (KR-1)ΔZ,m

JR — Gives Release Gate width, w = (JMAX-JR)Δy, m

KIN — Gives interface location, Z_T = (KIN-1)ΔZ, m

IBAR — Gives length of the flow domain (IBAR)ΔX, m

JBAR — Gives width of the flow domain, (JBAR)ΔY, m

KBAR — Gives total depth of the flow domain, (KBAR)ΔZ, m

DELX — Cell length, m

DELY — Cell height, m

DELZ — Cell width, m

VINITL Initial velocity of propeller, m/s

QSTAR Flow rate ratio, Q*

FRD Densimetric Froude number, F_{rd}

1.2 Standard Conditions

These conditions simulate a flowfield with (D* = .183 or D = 0.089m), and corresponding to the standard case discussed in Section 5.1.

IP = 3 corresponding to K* = .211

JP = 6

KP = 8 corresponding to L* = .211

JR = 5

KR = 2

KIN = 6 corresponding to $Z_T^* = .6$

IBAR = 7

JBAR = 9 corresponding to H = 0.478 m

KBAR = 5

DELX = 0.07, m

DELY = 0.06, m

DELZ = 0.04, m

VINITL = 0.2134 initial velocity V_o = 0.2134 m/s

QSTAR = 2.6

FRD = 2.0

2. Deflected Turbulent Jet Application

A description is given of the important points needed in order to use the second of the programs listed in Appendix D.

2.1 Important Fortran Variables

DELX = cell length, m

DELY = cell height, m

DELZ = cell width, m

IBAR = number of cells in x-direction

JBAR = number of cells in y-direction

KBAR = number of cells in z-direction

JJET = J-index of the jet height

KJET = K-index of the jet location in the z-direction

IJET = I-index of the jet location in the x-direction

IJETF = final cell of the jet in the z-direction

VISCOS = absolute laminar viscosity, $N \cdot s/m^2$

DEN = density

SC = Schmidt number

RAT = jet-to-cross flow ratio

UJET = jet velocity m/s

UIN = inlet cross flow velocity

DELT = time-step s

2.2 Standard Conditions

These conditions correspond to the standard case discussed in Section 5.2.

DELX = 0.02 m

DELY = 0.015 m

DELZ = 0.01 m

IBAR = 9

JBAR = 9

KBAR = 7

JJET = 1

KJET = 8

IJET = 5

IJETF = 5

VISCOS = 1.983×10^{-5} $N \cdot s/m^2$

DEN = 1.2 Kg/m^3

SC = 1.0

RAT = 6.0

UIN = 0.1 m/s

DELT = 0.05 s

3. Dilution Jets in Gas Turbine Combustors Application

A description is given of the important points needed in order to use the third of the programs listed in Appendix D.

3.1 Important Fortran Variables

IN1 = I-index of first entrance plane, within calculation domain

IN2 = I-index of last entrance plane, within calculation domain

RIN = Inside radius, m

VINELT = Inlet velocity, m/s

RAT = Ratio of the dilution jet velocity to the inlet velocity

UDIL = Dilution jet velocity, m/s

KDIL = K-index of the dilution jet location in θ-direction

JDIL = J-index of the dilution jet location in z-direction

LFS = Index for swirl angle

ITMAX = Maximum number of iteration

CYCMAX = Maximum number of cycles (time-steps)

DELR = Cell height in R-direction, m

DELZ = Cell length in z-direction, m

DELTH = Cell width in θ-direction, m

DELT = Time step, sec

IBAR = Number of cells in R-direction

JBAR = Number of cells in z-direction

KBAR = Number of cells in θ-direction

VISCOS = Absolute laminar viscosity, $N \cdot s/m^2$

DEN = Density, K_g/m^3

SC = Schmidt number

3.2 Standard Conditions

These conditions correspond to the standard case discussed in Section 5.3.

IN1 = 2

IN2 = 6

RIN = 0.0, m

VINLET = 10.0 m/s

UDIL = 10.0 m/s

KDIL = 5

JDIL = 6

LFS = 2

ITMAX = 60

CYCMAX = 300

DELR = 0.05, m

DELZ = 0.01, m

DELTH = 0.14957 Radians

IBAR = 10

JBAR = 10

KBAR = 7

VISCOS = 1.983×10^{-5} $N \cdot s/m^2$

DEN = 1.2 K_g/m^3

SC = 1.0

APPENDIX D

COMPUTER PROGRAM LISTINGS

```
C
C*************************************************************************
C**
C**
C**          THREE-DIMENSIONAL PREDICTION TECHNIQUE FOR TURBULENT
C**            RECIRCULATING FLOWFIELDS IN CARTESIAN COORDINATES
C**
C**                                PH.D. THESIS
C**                                     BY
C**                               AHMED A. BUSNAINA
C**
C**
C**
C**                                  MAY, 1983
C**              SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING
C**                         OKLAHOMA STATE UNIVERSITY
C**                           STILLWATER , OKLAHOMA
C**
C**
C**
C*************************************************************************
C*************************************************************************
C
C-------- APPLICATION 1  : LOCAL DESTRATIFICATION OF RESERVOIRS --------
C
C*************************************************************************
C*************************************************************************
C
C
C-------- THIS PROGRAM SOLVES THREE DIMENSIONS NAVIER STOKES EQUATION.
C--------THE FLUID SHOULD BE INCOMPRESSIBLE I.E. MACH NUM SHOULD NOT BE
C--------  GREATER THEN .3.
C--------THE DATA THAT SHOULD BE ENTERED IS,
C             DELX     IS THE SIZE OF MESH IN X-D.
C             DELY     IS THE SIZE OF MESH IN Y-D.
C             DELZ     IS THE SIZE OF MESH IN Z-D.
C             IBAR     NUMBER OF DEVISIONS IN X-D.
C             KBAR     NUMBER OF DEVISIONS IN Z-D.
C             JBAR     NUMBER OF DEVISIONS IN Y-D.
C             DELT IS THE TIME INCREMENT AND IT IS LIMITED BY STABULITY
C              IT SHOULD BE SMALLER MAX SIZE MESH OVER MAX VELOCITY.
C              TYPICALLY .25 TO .33 TIMES IT.
C             EPSI     IS THE CONVERGENCE CRITERIA.
C             NU       IS THE KINEMATIC VISCOSITY.
C             VI       IS THE INITIAL VELOCITY IN Y-D.
C             UI       IS THE INITIAL VELOCITY IN X-D.
C             WI       IS THE INITIAL VELOCITY IN Z-D.
C             GX,GY,GZ  ARE ACCELERATIONS.
C             ALPHA SHOULD BE BETWEEN 1. AND(DELT*VELOCITY)/MESH SIZE.
C---------- KM STANDS FOR YHE LOCATION  OF THE SLICE TO BE PLOTTED.
C---------- LC IS THE NUMBER OF CYCLE AT WHICH THE PLOTTER WILL BE CALLE
C
C
C-----------------------------------------------------------------------
```

```
C
C
      INTEGER CYCLE
      REAL  NU,MU,MUN
       DIMENSION  U(10,11,10),V(10,11,10),W(10,11,10),P(10,11,10),
     #  UN(10,11,10),VN(10,11,10),WN(10,11,10),S(10,11,10),SN(10,11,10)
     #  ,DEN(10,11,10),S2(10,11,10),B1(10,11,10)
      DIMENSION IBLOCK(10)
C
      DIMENSION TK(10,11,10),TKN(10,11,10),E(10,11,10),EN(10,11,10),
     # MU(10,11,10),GG(10,11,10),MUN(10,11,10)
C
      DIMENSION YPLUSS(10,10),TAUS(10,10),XPLUSW(11,10),TAUW(11,10)
C
      DIMENSION XX(10,11,3),YY(10,11,3),X(330),Y(330)
C
C-----------------------------------------------------------------------
C--------DATA CARDS.----------------------------------------------------
      DELX=0.04
      DELY=0.06
      DELZ=0.04
      IBAR=7
      JBAR=9
      KBAR=5
      JIN=6
      JP=8
      KP=6
      IP=4
      JR=2
      KR=5
      CMU=0.09
      CD=1.0
      C1=1.44
      C2=1.92
      SK=1.0
      SE=1.21
      PK=0.4187
      CAPPA=PK
      ELOG=9.793
      URFVIS=0.7
      NU=1.3E-6
      VISCOS=1.3E-3
      VISMAX=142.0*VISCOS
      DEN1=1000.0
      REN=10000.0
      RLARGE=DELY*JBAR
      SC=1.0
C     D*=0.131
C     FROUDE NUMBER = 2.0
      FRD=2.0
      DR=1.0+(0.49)/(1.595*32.2*(FRD)**2)
      VINITL=0.2134
      QSTAR=2.6
      DEN2=DEN1*DR
```

```
C
      DATA  U,UN,V,VN,W,WN,P /7700 * 0.0 /
      DATA   EPSI,GX,GY,GZ,OMG,DZRO /1.E-3,0.0,-9.81,0.0,1.7,1.0/
      ALPHA=0.6
C
C-----------------------------------------------------------------------
C
      UI= 0.0
      VI=0.0
      WI=0.0
      PI=0.0
      UINT=0.2134
      ASR=DELY*DELZ*2.0
      ASP=DELX*DELZ
      RDX = 1/DELX
      RDY = 1/ DELY
      RDZ = 1/ DELZ
      IMAX=IBAR+2
      JMAX = JBAR+2
      KMAX = KBAR + 2
      IM1=IBAR+1
      JM1=JBAR+1
      KM1=KBAR+1
      IM2=IBAR
      JM2=JBAR
      KM2=KBAR
C
      DELT=0.05
      DTMAX1=0.33*DELY/VINITL
      DTMAX2=0.5*DELX**2*DELY**2*DELZ**2/((DELX**2+DELY**2+DELZ**2)*NU)
      ALFAMI=1.5*VINITL*DELT/DELY
      BETA=OMG/(2*DELT*(1/(DELX**2)+1/(DELY**2)+1/(DELZ**2)))
      WRITE(6,92)DELT,DTMAX1,DTMAX2,DR
      WRITE(6,93)  ALPHA,ALFAMI
C
C     DELT  REDUCTION  FOR  STABILITY
C
      IF(DELT.GT.DTMAX1) DELT=DTMAX1
 92   FORMAT(//,2X,'DELT = ',E12.4,2X,'DTMAX1 = ',E12.4,2X,'DTMAX2 = ',
     @ E12.4,' DR = ',E12.6,/)
 93   FORMAT(/,2X,'ALPHA = ',E12.4,2X,'ALFAMI = ',E12.4,//)
C
      JNP=JIN+1
      QPROP=VINITL*ASP
      QREL=QPROP/QSTAR
      UOUT=QREL/ASR
C-----------------------------------------------------------------------
      T=0
      ITER=0
      CYCLE=0
C
C     INITIAL FIELD VALUES.
C
      DO  560  I=2,IM1
```

```
      DO  560  J=2,JM1
      DO  560  K=2,KM1
      U(I,J,K)=UI
      V(I,J,K)=VI
      P(I,J,K)=PI
560   W(I,J,K)=WI
C
C
C
C
C
C   ---------- INITIAL  DENSITY  AND  MASS  FRACTION  FIELD
C
      DO  555  I=1,IMAX
      DO  555  J=1,JIN
      DO  555  K=1,KMAX
      S(I,J,K)=0.0
      S2(I,J,K)=1.0
      DEN(I,J,K)=DEN2
 555  CONTINUE
      DO  556  I=1,IMAX
      DO  556  K=1,KMAX
      DO  556  J=JNP,JMAX
      S(I,J,K)=1.0
      S2(I,J,K)=0.0
      DEN(I,J,K)=DEN1
 556  CONTINUE
C
C-----------  INITIAL TURB ENERGY & DISSIAPATION
C
      TIN=0.03*(UINT**2)*20
      EIN=(TIN**1.5)/DELZ
      VISC=(UINT*RLARGE*2.*DEN1)/REN
       IF(EIN.LT.1.E-6) EIN=1.E-6
C
      DO  557  I=1,IMAX
      DO  557  J=1,JMAX
      DO  557  K=1,KMAX
       TK(I,J,K)=TIN
       E(I,J,K)=EIN
       MU(I,J,K)=VISC
 557  CONTINUE
C
C
C-----------------------------------------------------------------------
C
      ASSIGN 5000 TO KRET
      GOTO 2000
1000  CONTINUE
      ITER=0
      FLG=1.0
      ASSIGN 3000 TO KRET
C
C-----------------------------------------------------------------------
```

```
C
      DO 1100 I=2,IM1
      DO 1100 J=2,JM1
      DO 1100 K=2,KM1
C
      TMXE=   (MUN(I+1,J,K)+MUN(I,J,K))/2.0
C
      TMXW=   (MUN(I-1,J,K)+MUN(I,J,K))/2.0
C
      TMYN=   (MUN(I,J+1,K)+MUN(I,J,K))/2.0
C
      TMYS=   (MUN(I,J-1,K)+MUN(I,J,K))/2.0
C
      TMZO=   (MUN(I,J,K+1)+MUN(I,J,K))/2.0
C
      TMZI=   (MUN(I,J,K-1)+MUN(I,J,K))/2.0
C
C
      TMUN=(MUN(I+1,J,K)+MUN(I,J,K)+MUN(I+1,J+1,K)+MUN(I,J+1,K))/4.0
      TMUS=(MUN(I,J,K)+MUN(I+1,J,K)+MUN(I,J-1,K)+MUN(I+1,J-1,K))/4.0
      TMUO=(MUN(I,J,K+1)+MUN(I+1,J,K)+MUN(I+1,J,K+1)+MUN(I,J,K))/4.0
      TMUI=(MUN(I,J,K)+MUN(I+1,J,K)+MUN(I,J,K-1)+MUN(I+1,J,K-1))/4.0
      TMVE=(MUN(I,J,K)+MUN(I+1,J,K)+MUN(I+1,J+1,K)+MUN(I,J+1,K))/4.0
      TMVW=(MUN(I,J,K)+MUN(I,J+1,K)+MUN(I-1,J,K)+MUN(I-1,J+1,K))/4.0
      TMVO=(MUN(I,J,K)+MUN(I,J,K+1)+MUN(I,J+1,K)+MUN(I,J+1,K+1))/4.0
      TMVI=(MUN(I,J,K)+MUN(I,J+1,K)+MUN(I,J+1,K-1)+MUN(I,J,K-1))/4.0
      TMWE=(MUN(I,J,K+1)+MUN(I,J,K)+MUN(I+1,J,K)+MUN(I+1,J,K+1))/4.0
      TMWW=(MUN(I,J,K+1)+MUN(I,J,K)+MUN(I-1,J,K)+MUN(I-1,J,K+1))/4.0
      TMWN=(MUN(I,J,K)+MUN(I,J,K+1)+MUN(I,J+1,K)+MUN(I,J+1,K+1))/4.0
      TMWS=(MUN(I,J,K)+MUN(I,J,K+1)+MUN(I,J-1,K)+MUN(I,J-1,K+1))/4.0
C
      DVXY=(TMUN*(VN(I+1,J,K)-VN(I,J,K))-TMUS*(VN(I+1,J-1,K)-VN(I,J-1,K
     1 )))*RDX*RDY
      DWXZ=(TMUO*(WN(I+1,J,K)-WN(I,J,K))-TMUI*(WN(I+1,J,K-1)-WN(I,J,K-
     1 1)))*RDX*RDZ
      DUYX=(TMVE*(UN(I,J+1,K)-UN(I,J,K))-TMVW*(UN(I-1,J+1,K)-UN(I-1,J,
     1 K)))*RDY*RDX
      DWYZ=(TMVO*(WN(I,J+1,K)-WN(I,J,K))-TMVI*(WN(I,J+1,K-1)-WN(I,J,K-
     2 1)))*RDY*RDZ
      DUZX=(TMWE*(UN(I,J,K+1)-UN(I,J,K))-TMWW*(UN(I-1,J,K+1)-UN(I-1,J,
     2 K)))*RDZ*RDX
      DVZY=(TMWN*(VN(I,J,K+1)-VN(I,J,K))-TMWS*(VN(I,J-1,K+1)-VN(I,J-1,
     2 K)))*RDZ*RDY
C
      VISX=(RDX**2*(TMXE*(UN(I+1,J,K)-UN(I,J,K))-TMXW*(UN(I,J,K)-UN(I-1
     2 ,J,K)))*2.+RDY**2*(TMYN*(UN(I,J+1,K)-UN(I,J,K))-TMYS*(UN(I,J,K)-
     3 UN(I,J-1,K)))+DVXY+RDZ**2*(TMZO*(UN(I,J,K+1)-UN(I,J,K))-TMZI*(UN(
     4 I,J,K)-UN(I,J,K-1)))+DWXZ)/DEN1
      FUX  =  RDX/4*  ((UN(I,J,K) +UN(I+1,J,K))**2
     1   +ALPHA*ABS(UN(I,J,K)+ UN(I+1,J,K))*( UN(I,J,K)-UN(I+1,J,K))
     2  -(UN(I-1,J,K)+UN(I,J,K))**2
     3   -ALPHA * ABS(UN(I-1,J,K)+UN(I,J,K))*(UN(I-1,J,K) - UN(I,J,K)))
      FUY=RDY/4*((VN(I,J,K)+VN( I+1,J,K))*(UN(I,J,K)+UN(I,J+1,K))
     1  +ALPHA *ABS(VN(I,J,K) +VN(I+1,J,K))*( UN(I,J,K)-UN(I,J+1,K))
```

```
     2     -(VN(I  ,J-1,K)+VN( I+1,J-1,K))* (UN(I,J-1,K)+UN(I,J,K))
     3  -ALPHA* ABS (VN(I,J-1,K)+VN(I+1,J-1,K))
     4                          *( UN(I,J-1,K) - UN(I,J,K)))
       FUZ = RDZ/4*(( WN(I,J,K) +WN(I+1,J,K))*(UN(I,J,K)+UN(I,J,K+1))
     1  + ALPHA*ABS(WN(I,J,K)+WN(I+1,J,K))*(UN(I,J,K)-UN(I,J,K+1))
     2 -(WN(I,J,K-1) +WN(I+1,J,K-1))*(UN(I,J,K-1) + UN(I,J,K))
     3  -ALPHA*ABS(WN(I,J,K-1) +WN(I+1,J,K-1))*( UN(I,J,K-1)-UN(I,J,K)))
C
       VISZ=(RDX**2*(TMXE*(WN(I+1,J,K)-WN(I,J,K))-TMXW*(WN(I,J,K)-WN(I-
     1 1,J,K)))+DUZX+RDY**2*(TMYN*(WN(I,J+1,K)-WN(I,J,K))-TMYS*(WN(I,J,K
     2 )-WN(I,J-1,K)))+DVZY+RDZ**2*(TMZO*(WN(I,J,K+1)-WN(I,J,K))-TMZI*(
     3 WN(I,J,K)-WN(I,J,K-1)))*2.)/DEN1
       FWZ=RDZ/4    *  (    (WN(I,J,K)    +WN(I,J,K+1))  **2
     1 +ALPHA *    ABS(    WN(I,J,K)    +WN(I,J,K+1))    *    (  WN(I,J,K)
     2 -WN(I,J,K+1))
     3 -(     WN(I,J,K-1)    +WN(I,J,K))    **2
     4 -ALPHA    * ABS(    WN(I,J,K-1)    +WN(I,J,K))  *(WN(I,J,K-1)
     5  -WN(I,J,K)     )     )
      FWX=  RDX/4*  ((UN(I,J,K) +UN(I,J,K+1))*(WN(I,J,K) +WN(I+1,J,K))
     1 +ALPHA    *ABS( UN(I,J,K)     +UN(I,J,K+1))    * (WN(I,J,K)
     2 -WN(I+1,J,K)       )
     3 -(UN(I-1,J,K)    +UN(I-1,J,K+1)      )*(WN(I-1,J,K)     +WN(I,J,K))
     4  -ALPHA     *ABS(    UN(I-1,J,K)   +UN(I-1,J,K+1)  ) *
     5     (WN(I-1,J,K)     -WN(I,J,K)       )       )
      FWY=RDY/4 * ((VN(I,J,K)+VN(I,J,K+1))*(WN(I,J,K) + WN(I,J+1,K))
     1  +ALPHA  *ABS(    VN(I,J,K)    +VN(I,J,K+1)      )    *( WN(I,J,K)
     2  -WN(I,J+1,K) )
     3   -( VN(I,J-1,K)    +VN(I,J-1,K+1)  )*(WN(I,J-1,K) +WN(I,J,K)    )
     4  -ALPHA  * ABS (  VN(I,J-1,K)   +VN(I,J-1,K+1)     )*
     5  (    WN(I,J-1,K)     -WN(I,J,K)      )       )
C
       VISY=(RDX**2*(TMXE*(VN(I+1,J,K)-VN(I,J,K))-TMXW*(VN(I,J,K)-VN(I-
     1 1,J,K)))+ DUYX + RDY**2* (TMYN* (VN(I,J+1,K)-VN(I,J,K))-TMYS*
     2 (VN(I,J,K)-VN(I,J-1,K)))* 2. + RDZ**2* (TMZO* (VN(I,J,K+1)-
     3 VN(I,J,K))-TMZI* (VN(I,J,K)-VN(I,J,K-1)))+ DWYZ )/DEN1
       FVZ=    RDZ/4 * ((WN(I,J,K) + WN(I,J+1,K)) * ( VN(I,J,K)
     1+VN(I,J,K+1) )+ALPHA *ABS( WN(I,J,K) +WN(I,J+1,K) ) *(VN(I,J,K)
     2   -VN(I,J,K+1) ) - ( WN(I,J,K-1) +WN(I,J+1,K-1) ) *(VN(I,J,K-1)
     3 +VN(I,J,K)) -ALPHA * ABS( WN(I,J,K-1) +WN(I,J+1,K-1)  )*
     4   (  VN(I,J,K-1) - VN(I,J,K) ) )
       FVX = RDX/ 4 *((UN(I,J,K) + UN(I,J+1,K))*(VN(I,J,K )+VN(I+1,J,K))
     1  + ALPHA *ABS(UN(I,J,K) +UN(I,J+1,K))*(VN(I,J ,K)-VN(I+1,J,K))
     2  -(UN(I-1,J,K)+UN(I-1 , J+1,K))*(VN( I-1,J,K)+ VN(I,J,K))
     3  - ALPHA*ABS(UN(I-1,J,K) +UN(I-1,J+1,K)) *(VN(I-1,J,K)-VN(I,J,K))
     4     )
       FVY = RDY/4*((VN(I,J,K)  +VN(I,J+1,K))**2
     1  + ALPHA*ABS(VN(I,J,K)+VN(I,J+1,K))*( VN(I,J,K) -VN(I,J+1,K))
     2  -(VN(I,J-1,K)+VN(I,J,K) )**2
     3  -ALPHA*ABS(VN(I,J-1,K) +VN(I,J,K))*(VN (I,J-1,K) -VN(I,J,K)))
C
       FMX=(UN(I,J,K)*(SN(I,J,K)+SN(I+1,J,K))+ALPHA*ABS(UN(I,J,K))*
     1 (SN(I,J,K)-SN(I+1,J,K))-UN(I-1,J,K)*(SN(I-1,J,K)+SN(I,J,K))-
     2 ALPHA*ABS(UN(I-1,J,K))*(SN(I-1,J,K)-SN(I,J,K)))/(2.*DELX)
C
```

```
      FMY=(VN(I,J,K)*(SN(I,J,K)+SN(I,J+1,K))+ALPHA*ABS(VN(I,J,K))*
     1 (SN(I,J,K)-SN(I,J+1,K))-VN(I,J-1,K)*(SN(I,J-1,K)+SN(I,J,K))-
     2 ALPHA*ABS(VN(I,J-1,K))*(SN(I,J-1,K)-SN(I,J,K)))/(2.*DELY)
C
      FMZ=(WN(I,J,K)*(SN(I,J,K)+SN(I,J,K+1))+ALPHA*ABS(WN(I,J,K))*
     1 (SN(I,J,K)-SN(I,J,K+1))-WN(I,J,K-1)*(SN(I,J,K-1)+SN(I,J,K))-ALPHA
     2 *ABS(WN(I,J,K-1))*(SN(I,J,K-1)-SN(I,J,K)))/(2.*DELZ)
C
      VIMX=(RDX**2*(TMXE*(SN(I+1,J,K)-SN(I,J,K))-TMXW*(SN(I,J,K) -
     1 SN(I-1,J,K)))+RDY**2*(TMYN*(SN(I,J+1,K)-SN(I,J,K))-TMYS*(SN
     2 (I,J,K)-SN(I,J-1,K)))+RDZ**2*(TMZO*(SN(I,J,K+1)-SN(I,J,K))-
     3 TMZI*(SN(I,J,K)-SN(I,J,K-1))))/(DEN1*SC)
C
C     IF(I.EQ.2) GO TO 1200
C     IF(J.EQ.2) GO TO 1200
C
      DVX=((VN(I,J,K)+VN(I+1,J,K)+VN(I,J-1,K)+VN(I+1,J-1,K)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I-1,J,K)+VN(I-1,J-1,K)))/
     2  (4.*DELX)
      DUY=((UN(I,J,K)+UN(I,J+1,K)+UN(I-1,J+1,K)+UN(I-1,J,K)) -
     1  (UN(I,J,K)+UN(I,J-1,K)+UN(I-1,J-1,K)+UN(I-1,J,K))) /
     2  (4.*DELY)
      DWY=((WN(I,J,K)+WN(I,J+1,K)+WN(I,J+1,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I,J-1,K)+WN(I,J-1,K-1))) /
     2  (4.*DELY)
      DVZ=((VN(I,J,K)+VN(I,J,K+1)+VN(I,J-1,K)+VN(I,J-1,K+1)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I,J-1,K-1)+VN(I,J,K-1))) /
     2  (4.*DELZ)
      DUZ=((UN(I,J,K)+UN(I,J,K+1)+UN(I-1,J,K)+UN(I-1,J,K+1)) -
     1  (UN(I,J,K)+UN(I-1,J,K)+UN(I-1,J,K-1)+UN(I,J,K-1))) /
     2  (4.*DELZ)
      DWX=((WN(I,J,K)+WN(I+1,J,K)+WN(I+1,J,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I-1,J,K-1)+WN(I-1,J,K))) /
     2  (4.*DELX)
      GG(I,J,K) = MUN(I,J,K)* (2.*(((UN(I,J,K)-UN(I-1,J,K))/DELX)**2 +
     1  ((VN(I,J,K)-VN(I,J-1,K))/DELY)**2 +((WN(I,J,K) -WN(I,J,
     2  K-1))/DELZ)**2) + (DVX+DUY)**2 + (DWY+DVZ)**2 +
     3    (DUZ+DWX)**2 )
1200  VISK=((RDX**2* (TMXE* (TKN(I+1,J,K)-TKN(I,J,K)) - TMXW* (
     1 TKN(I,J,K)-TKN(I-1,J,K))) + RDY**2* (TMYN* (TKN(I,J+1,K) -
     2 TKN(I,J,K)) -TMYS* (TKN(I,J,K)-TKN(I,J-1,K))) + RDZ**2*
     3 (TMZO* (TKN(I,J,K+1)-TKN(I,J,K))-TMZI* (TKN(I,J,K)-TKN
     4 (I,J,K-1))))/SK)/DEN1
C
      SORK1=DEN(I,J,K)*EN(I,J,K)/DEN1
      SORK= GG(I,J,K)/DEN1
      SORK2=1.0+DELT * DEN(I,J,K)*EN(I,J,K) /(TKN(I,J,K)*DEN1)
      VISK=VISK+SORK
C
      FKX= (UN(I,J,K)*(TKN(I,J,K)+TKN(I+1,J,K)) + ALPHA * ABS
     1 (UN(I,J,K))*(TKN(I,J,K)-TKN(I+1,J,K)) - UN(I-1,J,K) *
     2 (TKN(I-1,J,K)+TKN(I,J,K)) - ALPHA * ABS(UN(I-1,J,K)) *
     3 (TKN(I-1,J,K)-TKN(I,J,K))) /(2.*DELX)
      FKY= (VN(I,J,K)*(TKN(I,J,K)+TKN(I,J+1,K))+ ALPHA *ABS
```

```
     1 (VN(I,J,K)) *(TKN(I,J,K) - TKN(I,J+1,K)) -VN(I,J-1,K) *
     2 (TKN(I,J-1,K)+TKN(I,J,K)) - ALPHA * ABS(VN(I,J-1,K)) *
     3 (TKN(I,J-1,K)-TKN(I,J,K)))/(2.*DELY)
       FKZ= (WN(I,J,K)*(TKN(I,J,K)+TKN(I,J,K+1)) + ALPHA * ABS
     1 (WN(I,J,K))* (TKN(I,J,K)-TKN(I,J,K+1)) - WN(I,J,K-1) *
     2 (TKN(I,J,K-1)+TKN(I,J,K)) - ALPHA * ABS(WN(I,J,K-1)) *
     3 (TKN(I,J,K-1)-TKN(I,J,K)))/(2.*DELZ)
C
C      CALCULATE NEW TIME VALUES.
C
       U(I,J,K) = UN(I,J,K) +DELT*(RDX*(P(I,J,K)  -P(I+1,J,K))
     1   + GX + VISX - FUX -FUY -FUZ)
       V(I,J,K)=VN(I,J,K)+DELT*(RDY*(P(I,J,K)-P(I,J+1,K))/DEN1+GY*
     1 (DEN(I,J,K)-DEN1)/DEN1-FVX-FVY-FVZ+VISY)
       W(I,J,K) =   WN(I,J,K)  +  DELT*(RDZ *   (P(I,J,K) -P(I,J,K+1))
     1 +GZ -FWX -FWY -FWZ +VISZ)
       TK(I,J,K) =(TKN(I,J,K) + DELT* (-FKX-FKY-FKZ +VISK))/SORK2
C      WRITE(6,37) I,J,K,SORK,SORK1,SORK2,VISK,TK(I,J,K)
       FKSUM=FKX+FKY+FKZ
C      WRITE(6,39) I,J,K,FKX,FKY,FKZ,FKSUM
C      IF(TK(I,J,K).LT.0.0) TK(I,J,K)=TIN
       IF(TK(I,J,K).LT.1.E-3) TK(I,J,K)=1.E-3
C
       VISOLD=MUN(I,J,K)
       IF(EN(I,J,K).LT.1.E-3) EN(I,J,K)=1.E-3
       MU(I,J,K)=(CMU*DEN(I,J,K)*(TKN(I,J,K))**2)/EN(I,J,K) + VISCOS
       MU(I,J,K)=URFVIS*MU(I,J,K)+(1.-URFVIS)*VISOLD
          IF(MU(I,J,K).LT.VISC) MU(I,J,K)=VISC
            IF(MU(I,J,K).GT.VISMAX) MU(I,J,K)=VISMAX
C
       S(I,J,K)=SN(I,J,K)+DELT*(-FMX-FMY-FMZ+VIMX)
       IF(CYCLE.EQ.1)  S(I,J,K)=SN(I,J,K)
       S2(I,J,K)=1.0-S(I,J,K)
       DEN(I,J,K)=S(I,J,K)*DEN1+S2(I,J,K)*DEN2
       B1(I,J,K)=GY*(DEN(I,J,K)-DEN1)/DEN1
C
C37    FORMAT(1X,3(I2,1X),5(E14.6))
C39    FORMAT(1X,3(I2,1X),'CONV IN X,Y,Z',4(E14.6))
C
1100   CONTINUE
C-------------------------------------------------------------------
C
2000   CONTINUE
C
C
       QS=0.0
       QS2=0.0
C
       DO  551  J=2,JR
       DO  551  K=KR,KM1
        QS=QS+S(2,J,K)*DELZ*DELY*UOUT
        QS2=QS2+S2(2,J,K)*DELZ*DELY*UOUT
 551   CONTINUE
C
```

```
      AT=(KBAR*DELZ+IBAR*DELX)*(JM1-JIN)*DELY
C
      AB=(KBAR*DELZ+IBAR*DELX)*(JIN-1)*DELY
      UT=QS/AT
      UB=QS2/AB
C
      DENAC=(QS*DEN1+QS2*DEN2)/((KMAX-5)*(JR-1)*DELY*DELZ*UOUT)
C
      DF=(DENAC-DEN2)/(DEN1-DEN2)
C
C
C  ==========BOUNDARY  CONDITIONS.====================
C
C     VERTICAL
C       X - Y  PLANE
C
      DO  20  J=1,JMAX
      DO  20  I=1,IMAX
C
C    ------  END  PLANE  --  FREE  SLIP    K=1
C
C
      U(I,J,1)=U(I,J,2)
      V(I,J,1)=V(I,J,2)
      S(I,J,1)=S(I,J,2)
      TK(I,J,1)=TK(I,J,2)
C     TK(I,J,1)=0.0
      E(I,J,1)=E(I,J,2)
      MU(I,J,1)=MU(I,J,2)
C
C     SYMMETRY  PLANE  FREE SLIP     K=KMAX
C
C
      W(I,J,KM1)=0.0
      W(I,J,KMAX)=W(I,J,KM1)
      U(I,J,KMAX)=U(I,J,KM1)
      V(I,J,KMAX)=V(I,J,KM1)
      S(I,J,KMAX)=S(I,J,KM1)
      TK(I,J,KMAX)=TK(I,J,KM1)
C     TK(I,J,KMAX)=0.0
      E(I,J,KMAX)=E(I,J,KM1)
       MU(I,J,KMAX)=MU(I,J,KM1)
 20   CONTINUE
C
C      Y  -  Z  PLANE    FREE SLIP      I=1        THE DAM WALL
C
      DO  21  J=1,JMAX
      DO  21  K=1,KMAX
      S(1,J,K)=S(2,J,K)
      TK(1,J,K)=TK(2,J,K)
C     TK(1,J,K)=0.0
C     E(2,J,K)=(((CMU*CD)**0.75)*TK(2,J,K)**1.5)/(CD*PK*DELX*0.5)
      E(1,J,K)=E(2,J,K)
      MU(1,J,K)=MU(2,J,K)
```

```
      W(1,J,K)=W(2,J,K)
      V(1,J,K)=V(2,J,K)
       U(1,J,K)=0.0
 21   CONTINUE
C
C     HORIZONTAL
C       X - Z  PLANE
C
      DO  22  I=1,IMAX
      DO  22  K=1,KMAX
C
C      TOP   FREE SLIP         J=JMAX
C
      V(I,JM1,K)=0.0
      V(I,JMAX,K)=V(I,JM1,K)
      U(I,JMAX,K)=U(I,JM1,K)
      W(I,JMAX,K)=W(I,JM1,K)
      S(I,JMAX,K)=S(I,JM1,K)
      TK(I,JMAX,K)=TK(I,JM1,K)
C     TK(I,JMAX,K)=0.0
      E(I,JMAX,K)=E(I,JM1,K)
      MU(I,JMAX,K)=MU(I,JM1,K)
C
C    ----  BOTTOM   --  FREE SLIP        J=1
C
      V(I,1,K)=0.0
      U(I,1,K)=U(I,2,K)
      W(I,1,K)=W(I,2,K)
      S(I,1,K)=S(I,2,K)
      TK(I,1,K)=TK(I,2,K)
C     TK(I,1,K)=0.0
C     E(I,2,K)=(((CMU*CD)**0.75)*TK(I,2,K)**1.5)/(CD*PK*DELY*0.5)
      E(I,1,K)=E(I,2,K)
      MU(I,1,K)=MU(I,2,K)
 22   CONTINUE
C
C     PROPPELER  B.CS.
C
      JPM1=JP-1
C
      DO  27  K=KP,KM1
      DO  27  I=IP,4
      U(I,JP,K)=0.0
      V(I,JP,K)=-VINITL
      V(I,JPM1,K)=-VINITL
      W(I,JP,K)=0.0
      TK(I,JP,K)=0.03*(VINITL**2)*20.0
      E(I,JP,K)=(TK(I,JP,K)**1.5)/DELZ
 27   CONTINUE
C
C     OUTLET FLOW   B.CS.
C
      DO  30  J=2,JR
      DO  30  K=KR,KM1
```

```
      U(1,J,K)=-UOUT
      V(1,J,K)=0.0
      W(1,J,K)=0.0
 30   CONTINUE
C
C     INLET  BOUNDARY  CONDITIONS
C
      DO  25  J=2,JIN
      DO  25  K=2,KM1
      U(IMAX,J,K)=-UB
      U(IM1,J,K)=-UB
      TK(IMAX,J,K)=0.03*UB**2*10.0
      E(IMAX,J,K)=(TK(IMAX,J,K)**1.5)/(0.01*JBAR*DELY)
      S(IMAX,J,K)=0.0
      V(IMAX,J,K)=V(IM1,J,K)
      W(IMAX,J,K)=W(IM1,J,K)
      MU(IMAX,J,K)=MU(IM1,J,K)
 25   CONTINUE
C
      JINP=JIN+1
C
      DO  26  J=JINP,JM1
      DO  26  K=2,KM1
      U(IMAX,J,K)=-UT
      U(IM1,J,K)=-UT
      TK(IMAX,J,K)=0.03*UT**2*10.0
      E(IMAX,J,K)=(TK(IMAX,J,K)**1.5)/(0.01*JBAR*DELY)
      W(IMAX,J,K)=W(IM1,J,K)
      V(IMAX,J,K)=V(IM1,J,K)
      MU(IMAX,J,K)=MU(IM1,J,K)
      S(IMAX,J,K)=1.0
 26   CONTINUE
C
C
      DO  35  J=2,JIN
      DO  35  I=2,IM1
      S(I,J,1)=0.0
      S(I,J,2)=0.0
      TK(I,J,1)=0.03*UB**2*10.0
      E(I,J,1)=(TK(I,J,1)**1.5)/(0.01*JBAR*DELY)
      MU(I,J,1)=MU(I,J,2)
 35    W(I,J,1)=UB
C
      DO  36  J=JINP,JMAX
      DO  36  I=2,IM1
      S(I,J,1)=1.0
      S(I,J,2)=1.0
      TK(I,J,1)=0.03*UT**2*10.0
      E(I,J,1)=(TK(I,J,1)**1.5)/(0.01*JBAR*DELY)
      MU(I,J,1)=MU(I,J,2)
 36   W(I,J,1)=UT
C
      IF(CYCLE.EQ.0)  GO  TO  2990
      IF(ITER.GT.0)  GO  TO  2990
```

```
C
C       DO 1101 I=1,IMAX
C       WRITE(6,47) I
C       WRITE(6,57)
C       DO 1102 JJ=1,JMAX
C       J=JMAX-JJ+1
C1102   WRITE(6,44) (TK(I,J,K),K=1,KMAX)
C       WRITE(6,58)
C       DO 1103 JJ=1,JMAX
C       J=JMAX-JJ+1
C1103   WRITE(6,44) (E(I,J,K),K=1,KMAX)
C
C1101   CONTINUE
C
        DO 1110 I=1,IMAX
        DO 1110 J=1,JMAX
        DO 1110 K=1,KMAX
         EN(I,J,K)=E(I,J,K)
 1110   CONTINUE
C
C
C-------------- DISSIAPATION  EQUATION  -----------------
C
        DO  1111 I=2,IM1
        DO  1111 J=2,JM1
        DO  1111 K=2,KM1
C
         IF(TKN(I,J,K).LT.1.E-3) TKN(I,J,K)=1.E-3
C
        TMXE= (MUN(I+1,J,K)+MUN(I,J,K))/2.0
C
        TMXW= (MUN(I-1,J,K)+MUN(I,J,K))/2.0
C
        TMYN= (MUN(I,J+1,K)+MUN(I,J,K))/2.0
C
        TMYS= (MUN(I,J-1,K)+MUN(I,J,K))/2.0
C
        TMZO= (MUN(I,J,K+1)+MUN(I,J,K))/2.0
C
        TMZI= (MUN(I,J,K-1)+MUN(I,J,K))/2.0
C
        FEX=(UN(I,J,K)*(EN(I,J,K)+EN(I+1,J,K))+ALPHA* ABS(UN(I,J,K))*
     1 (EN(I,J,K)-EN(I+1,J,K))-UN(I-1,J,K)*(EN(I-1,J,K)+EN(I,J,K))
     2 -ALPHA *ABS(UN(I-1,J,K)) *(EN(I-1,J,K)-EN(I,J,K))) /
     3 (2. * DELX )
        FEY=( VN(I,J,K)* ( EN(I,J,K)+ EN(I,J+1,K)) + ALPHA*ABS(VN(I,J,K)
     1 ) * ( EN(I,J,K)- EN(I,J+1,K)) - VN(I,J-1,K)*(EN(I,J-1,K)+EN(I,J,K
     2 )) - ALPHA * ABS(VN(I,J-1,K)) * ( EN(I,J-1,K)- EN(I,J,K))) /
     3 (2. * DELY )
        FEZ=(WN(I,J,K) * ( EN(I,J,K)+ EN(I,J,K+1)) +ALPHA*ABS( WN(I,J,K
     1 ))*(EN(I,J,K)-EN(I,J,K+1)) - WN(I,J,K-1) * (EN(I,J,K-1)+EN(I,J,K
     2 )) - ALPHA * ABS(WN(I,J,K-1)) * ( EN(I,J,K-1)- EN(I,J,K))) /
     3 (2. * DELZ)
        DVX=((VN(I,J,K)+VN(I+1,J,K)+VN(I,J-1,K)+VN(I+1,J-1,K)) -
```

```
     1   (VN(I,J,K)+VN(I,J-1,K)+VN(I-1,J,K)+VN(I-1,J-1,K)))/
     2   (4.*DELX)
         DUY=((UN(I,J,K)+UN(I,J+1,K)+UN(I-1,J+1,K)+UN(I-1,J,K)) -
     1   (UN(I,J,K)+UN(I,J-1,K)+UN(I-1,J-1,K)+UN(I-1,J,K))) /
     2   (4.*DELY)
         DWY=((WN(I,J,K)+WN(I,J+1,K)+WN(I,J+1,K-1)+WN(I,J,K-1)) -
     1   (WN(I,J,K)+WN(I,J,K-1)+WN(I,J-1,K)+WN(I,J-1,K-1))) /
     2   (4.*DELY)
         DVZ=((VN(I,J,K)+VN(I,J,K+1)+VN(I,J-1,K)+VN(I,J-1,K+1)) -
     1   (VN(I,J,K)+VN(I,J-1,K)+VN(I,J-1,K-1)+VN(I,J,K-1))) /
     2   (4.*DELZ)
         DUZ=((UN(I,J,K)+UN(I,J,K+1)+UN(I-1,J,K)+UN(I-1,J,K+1)) -
     1   (UN(I,J,K)+UN(I-1,J,K)+UN(I-1,J,K-1)+UN(I,J,K-1))) /
     2   (4.*DELZ)
         DWX=((WN(I,J,K)+WN(I+1,J,K)+WN(I+1,J,K-1)+WN(I,J,K-1)) -
     1   (WN(I,J,K)+WN(I,J,K-1)+WN(I-1,J,K-1)+WN(I-1,J,K))) /
     2   (4.*DELX)
         GG(I,J,K) = MUN(I,J,K)* (2.*(((UN(I,J,K)-UN(I-1,J,K))/DELX)**2 +
     1   ((VN(I,J,K)-VN(I,J-1,K))/DELY)**2 +((WN(I,J,K) -WN(I,J,
     2   K-1))/DELZ)**2) + (DVX+DUY)**2 + (DWY+DVZ)**2 +
     3    (DUZ+DWX)**2 )
         VISEX=RDX**2*(TMXE* ( EN(I+1,J,K)- EN(I,J,K))-TMXW * (EN(I,J,
     1 K)- EN(I-1,J,K)))
         VISEY=RDY**2 *(TMYN *( EN(I,J+1,K)- EN(I,J,K))-TMYS*( EN(I,J,K)
     1 - EN(I,J-1,K)))
         VISEZ=RDZ**2 *(TMZO *( EN(I,J,K+1)- EN(I,J,K))-TMZI*(EN(I,J,K)
     2 - EN(I,J,K-1)))
         VIS=(VISEX +VISEY +VISEZ )/(SE*DEN1)
         GC=GG(I,J,K)
         ENC=EN(I,J,K)
         DENC=DEN(I,J,K)
         TKNC=TKN(I,J,K)
         TKC=TK(I,J,K)
         SORE= GG(I,J,K)*C1 * EN(I,J,K) /(TKN(I,J,K)*DEN1)
         SORE1=C2*DEN(I,J,K)*(EN(I,J,K)**2)/(TKN(I,J,K)*DEN1)
        SORE2=1.0 + DELT*C2* DEN(I,J,K) *EN(I,J,K) /(TKN(I,J,K)*DEN1)
         VISE=VIS + SORE
C
C
C38      FORMAT(1X,3(I2,1X),7(E12.5))
C
C
        E(I,J,K)=(EN(I,J,K) + DELT * (-FEX-FEY-FEZ+VISE))/SORE2
C        WRITE(6,38) I,J,K,SORE,SORE1,SORE2,VIS,VISE,E(I,J,K),GC
        IF(E(I,J,K).LT.0.0) E(I,J,K)=EIN
         IF(E(I,J,K).LT.1.E-3) E(I,J,K)=1.E-3
        FESUM=FEX+FEY+FEZ
C       WRITE(6,39) I,J,K,FEX,FEY,FEZ,FESUM
C
C
1111     CONTINUE
C
C
2990     CONTINUE
```

```
C
      GOTO KRET,(3000,5000)
3000  CONTINUE
      IF(FLG.EQ.0.)GO TO4000
      ITER=ITER+1
      IF(ITER.LT.50)  GO  TO  3050
      IF(CYCLE .LT.320) GO TO 4000
      T= 1E+10
      GOTO 5000
3050  FLG=0.0
C
C-----------------------------------------------------------------------
C  ====CONVERGENCE===
C     PRESSURE ITERATIONS.
C
      DO 3500 I=2,IM1
      DO 3500 J=2,JM1
      DO 3500 K=2,KM1
     D=1/DELX*(U(I,J,K)-U(I-1,J,K)) +1/DELY*(V(I,J,K) -V(I,J-1,K))
    +  + 1/DELZ*(W(I,J,K)-W(I,J,K-1))
      IF(ABS(D/DZRO).GE.EPSI)  FLG=1.
      DELP=-D*BETA
      P(I,J,K)=P(I,J,K)+DELP
      U(I,J,K)=U(I,J,K)+DELT*DELP/DELX
      U(I-1,J,K)=U(I-1,J,K)-DELT*DELP/DELX
      V(I,J,K)=V(I,J,K)+DELT*DELP/DELY
      V(I,J-1,K)=V(I,J-1,K)-DELT*DELP/DELY
      W(I,J,K)=W(I,J,K)+DELT*DELP/DELZ
      W(I,J,K-1)=W(I,J,K-1)-DELT*DELP/DELZ
3500  CONTINUE
      GOTO 2000
4000  CONTINUE
5000  CONTINUE
C
       GO TO 5245
C------------SHEAR STREES ON THE BOTTOM WALL
C
      DO  5100  I=2,IM1
      DO  5100  K=2,KM1
       J=2
      UAVG=(U(I,J,K)+U(I+1,J,K))/2.
      WAVG=(W(I,J,K)+W(I,J,K+1))/2.
      UEFF=SQRT(UAVG*UAVG+WAVG*WAVG)
      DENU=DEN(I,J,K)
      YP=DELY/2.
      SQRTK=SQRT(TK(I,J,K))
      CMUPQ=CMU**0.25
      YPLUSS(I,K)=DENU*CMUPQ*SQRTK*YP/VISCOS
      IF(YPLUSS(I,K).LE.11.63) GO TO 5101
      TMULT=DENU*CMUPQ*SQRTK*CAPPA/ALOG(ELOG*YPLUSS(I,K))
      TAUS(I,K)=-TMULT*UEFF
      GO  TO  5102
5101  TAURX=-VISCOS*UAVG/YP
      TAURW=-VISCOS*WAVG/YP
```

```
      TAUS(I,K)=SQRT(TAURX*TAURX+TAURW*TAURW)
5102  DUZ=((U(I,J,K)+U(I,J,K+1)+U(I-1,J,K)+U(I-1,J,K+1)) -
     1  (U(I,J,K)+U(I-1,J,K)+U(I-1,J,K-1)+U(I,J,K-1))) /
     2  (4.*DELZ)
      DWX=((W(I,J,K)+W(I+1,J,K)+W(I+1,J,K-1)+W(I,J,K-1)) -
     1  (W(I,J,K)+W(I,J,K-1)+W(I-1,J,K-1)+W(I-1,J,K))) /
     2  (4.*DELX)
      GG(I,J,K)=MU(I,J,K)*2.*(((U(I,J,K)-U(I-1,J,K))/DELX)**2+
     1 ((V(I,J,K)-V(I,J-1,K))/DELY)**2 + ((W(I,J,K)-W(I,J,K-1))/DELZ)
     2 **2 )+ TAUS(I,K)**2/MU(I,J,K) +MU(I,J,K)* (DUZ+DWX)**2
5100   CONTINUE
C
C----------- SHEAR STRESS ON THE SIDE WALL
C
       DO  5200  J=2,JM1
       DO  5200  K=2,KM1
        I=2
       IF(J.EQ.JR.AND.K.GE.KR) GO TO 5200
       VAVG=(V(I,J,K)+V(I,J+1,K))/2.
       WAVG=(W(I,J,K)+W(I,J,K+1))/2.
       VEFF=SQRT(VAVG*VAVG+WAVG*WAVG)
       XP=DELX/2.
       DENV=DEN(I,J,K)
       SQRTK=SQRT(TK(I,J,K))
       XPLUSW(J,K)=DENV*CMUPQ*SQRTK*XP/VISCOS
       IF(XPLUSW(J,K).LE.11.63) GO TO 5201
       TMULT=DENV*CMUPQ*SQRTK*CAPPA/ALOG(ELOG*XPLUSW(J,K))
       TAUW(J,K)=-TMULT*VEFF
       GO TO 5202
5201   TAUXR=VISCOS*VAVG/XP
       TAUXW=VISCOS*WAVG/XP
       TAUW(J,K)=SQRT(TAUXR*TAUXR+TAUXW*TAUXW)
5202  DWY=((WN(I,J,K)+WN(I,J+1,K)+WN(I,J+1,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I,J-1,K)+WN(I,J-1,K-1))) /
     2  (4.*DELY)
      DVZ=((VN(I,J,K)+VN(I,J,K+1)+VN(I,J-1,K)+VN(I,J-1,K+1)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I,J-1,K-1)+VN(I,J,K-1))) /
     2  (4.*DELZ)
      GG(I,J,K)=MU(I,J,K)*2.*(((U(I,J,K)-U(I-1,J,K))/DELX)**2 +
     1 ((V(I,J,K)-V(I,J-1,K))/DELY)**2 + ((W(I,J,K)-W(I,J,K-1)) /
     2 DELZ)**2)+ TAUW(J,K)**2/MU(I,J,K) + MU(I,J,K)*(DVZ+DWY)**2
5200  CONTINUE
C
5245    CONTINUE
C
C========PRINT=====
       WRITE(6,49)  ITER,T,CYCLE
       PRINT 54,DF
 54    FORMAT(//,5X,'DILUTION FACTOR', 2X,1E12.4,/)
       WRITE(6,666) V(3,3,3),W(3,3,3),U(3,3,3),V(3,2,4),W(3,2,4),
     # U(3,2,4)
C
       DO 5280 IW=1,320,40
       IF(CYCLE.EQ.IW) GO TO 646
```

```
5280    CONTINUE
        GO TO 636
646     CONTINUE
C
        DO 5300 I=2,IMAX
        DO 5300 J=2,JMAX
        K=6
        UA=(U(I,J,K)+U(I-1,J,K))/2.
        VA=(V(I,J,K)+V(I,J-1,K))/2.
        XX(I,J,1)=(0.4*I)-0.2-UA/2.
        XX(I,J,2)=XX(I,J,1)+UA/2.
        XX(I,J,3)=XX(I,J,2)+UA/2.
        YY(I,J,1)=(0.6*J)-0.3-VA/2.
        YY(I,J,2)=YY(I,J,1)+VA/2.
        YY(I,J,3)=YY(I,J,2)+VA/2.
5300    CONTINUE
        M=0
        DO 5400 I=2,IMAX
        DO 5400 J=2,JMAX
        DO 5400 L=1,3
        M=M+1
        X(M)=XX(I,J,L)
        Y(M)=YY(I,J,L)
5400    CONTINUE
C
        MAX=JM1*IM1*3.0
        DO 5450 I=1,MAX
        WRITE(14,559) X(I),Y(I)
5450    CONTINUE
559     FORMAT(E12.5,5X,E12.5)
C
        DO 5460 I=1,IMAX
        DO 5460 K=1,KMAX
        J=JIN
        V(1,J,K)=0.0
        V(IMAX,J,K)=0.0
        V(I,J,1)=0.0
        V(I,J,KMAX)=0.0
        V(I,J,K)=-V(I,J,K)
5460    CONTINUE
         DO 5470 K=1,KMAX
        J=JIN
        WRITE(15,558) (V(I,J,K),I=1,IMAX)
5470    CONTINUE
558     FORMAT(1X,10E11.4)
C
        DO 5480 I=1,IMAX
        DO 5480 K=1,KMAX
        J=JIN
        V(1,J,K)=V(2,J,K)
        V(IMAX,J,K)=V(IM1,J,K)
        V(I,J,1)=V(I,J,2)
        V(I,J,KMAX)=V(I,J,KM1)
        V(I,J,K)=-V(I,J,K)
```

```
5480   CONTINUE
636    CONTINUE
C
       IF(CYCLE.EQ.0)  GO TO 5152
       IF(CYCLE.EQ.1)  GO TO 5152
       IF(CYCLE.EQ.2)  GO TO 5152
       DO  66  II=10,400,50
C      DO  66  II=3,400
         IF(CYCLE.EQ.II)  GO TO 5152
 66    CONTINUE
C
          GOTO 5251
C
 5152  CONTINUE
C
C-------------------------------------------------------------------------
C-----------    OUTPUT   -------------------------------------------------
C
C-------------------------------------------------------------------------
C
5230   CONTINUE
       DO 5250 KK = 1,KMAX
       K=KMAX-KK+1
       WRITE(6,47) K
       WRITE(6,48)
       DO 5252 JJ =  1, JMAX
       J = JMAX -JJ+1
 5252  WRITE(6,44) (U(I,J,K),I=1,IMAX)
       WRITE(6,50)
       DO 5253 JJ = 1,JMAX
       J = JMAX -JJ+1
 5253  WRITE(6,44) (V(I,J,K),I=1,IMAX)
       WRITE(6,51)
       DO 5254 JJ = 1,JMAX
       J = JMAX -JJ+1
 5254  WRITE(6,44) (W(I,J,K),I=1,IMAX)
       WRITE(6,52)
       DO 5255 JJ = 1,JMAX
       J = JMAX -JJ+1
 5255  WRITE(6,44) (P(I,J,K),I=1,IMAX)
       WRITE(6,55)
       DO 5256 JJ = 1,JMAX
       J = JMAX -JJ+1
5256    PRINT 44,(S(I,J,K),I=1,IMAX  )
       WRITE(6,56)
       DO 5257 JJ=1,JMAX
       J=JMAX-JJ+1
 5257   PRINT 44,(DEN(I,J,K),I=1,IMAX)
       WRITE(6,57)
       DO  5258  JJ=1,JMAX
       J=JMAX-JJ+1
 5258  WRITE(6,44) (TK(I,J,K),I=1,IMAX)
       WRITE(6,58)
       DO  5259  JJ=1,JMAX
```

```
      J=JMAX-JJ+1
 5259 WRITE(6,44) (E(I,J,K),I=1,IMAX)
      WRITE(6,59)
      DO  5260  JJ=1,JMAX
      J=JMAX-JJ+1
 5260 WRITE(6,44) (MU(I,J,K),I=1,IMAX)
      WRITE(6,60)
      WRITE(6,44) (YPLUSS(I,K),I=1,IMAX)
C
      WRITE(6,61)
      WRITE(6,44) (XPLUSW(J,K),J=1,JMAX)
C
5250  CONTINUE
5251  CONTINUE
      DO 6100 I=1,IMAX
      DO 6100 J=1,JMAX
      DO 6100 K=1,KMAX
      VN(I,J,K)=V(I,J,K)
      UN(I,J,K)=U(I,J,K)
      WN(I,J,K)=W(I,J,K)
      SN(I,J,K)=S(I,J,K)
      TKN(I,J,K)=TK(I,J,K)
      EN(I,J,K)=E(I,J,K)
      MUN(I,J,K)=MU(I,J,K)
 6100 CONTINUE
      T=T+DELT
      IF(T.GT.320*DELT)  GO TO 6500
      IF(CYCLE .EQ.0 )GOTO 6200
C     IF(ITER.LE.1)  GO TO 6550
6200   CYCLE=CYCLE+1
      GOTO 1000
6500  STOP
 6550 WRITE(6,53)
      STOP
C
C---------------------------------------------------------------------
C
 47   FORMAT(//,20X,'K =',I2,/,19X,  7('='))
 49   FORMAT(//,6X,'ITER=',I5,9X,'TIME=',1PE12.5,10X,'CYCLE=',I4,/)
 44   FORMAT(1X,12(E11.4))
 48   FORMAT(//,2X,'U-FIELD',/)
 50   FORMAT(//,2X,'V-FIELD',/)
 51   FORMAT(//,2X,'W-FIELD',/)
 52   FORMAT(//,2X,'P-FIELD',/)
53    FORMAT (//,10X,7('='),'CONVERGENCE HAS  BEEN REACHED. ')
 55   FORMAT(//,2X,'S-FIELD',/)
 56   FORMAT(//,2X,'DEN-FIELD',/)
 57   FORMAT(//,2X,'TK-FIELD',/)
 58   FORMAT(//,2X,'E -FIELD',/)
 59   FORMAT(//,2X,'MU-FIELD',/)
 60   FORMAT(//,2X,'YPLUSS ',4X,'J = 2 ',/)
 61   FORMAT(//,2X,'XPLUSW ',4X,'I = 2 ',/)
 666  FORMAT(//,10X,'V(3,3,3)=',E11.4,2X,'W(3,3,3)=',E11.4,2X,
     # 'U(3,3,3)=',E11.4,//,10X,'V(3,2,4)=',E11.4,2X,'W(3,2,4)=',E11.4,
```

```
# 2X,'U(3,2,4)=',E11.4,//)
  END
```

```
C
C
C*************************************************************************
C**          THREE-DIMENSIONAL PREDICTION TECHNIQUE FOR TURBULENT
C**            RECIRCULATING FLOWFIELDS IN CARTESIAN COORDINATES
C**
C**
C**                              PH.D. THESIS
C**                                   BY
C**                             AHMED A. BUSNAINA
C**
C**
C**
C**                              MAY, 1983
C**                 SCHOOL OF MECHANICAL AND AROSPACE ENGINEERING
C**                           OKLAHOMA STATE UNIVERSITY
C**                              STILLWATER , OKLAHOMA
C**
C**
C**
C*************************************************************************
C*************************************************************************
C
C------- APPLICATION 2 : DEFLECTED TURBULENT JET IN A CROSSFLOW --------
C
C*************************************************************************
C*************************************************************************
C
C
C
C
C-------- THIS PROGRAM SOLVES THREE DIMENSIONAL NAVIER STOKES EQUATION.
C--------THE FLUID SHOULD BE INCOMPRESSIBLE I.E. MACH NUM SHOULD NOT BE
C--------  GREATER THEN .3.
C--------THE DATA THAT SHOULD BE ENTERED IS,
C             DELX     IS THE SIZE OF MESH IN X-D.
C             DELY     IS THE SIZE OF MESH IN Y-D.
C             DELZ     IS THE SIZE OF MESH IN Z-D.
C             IBAR     NUMBER OF DEVISIONS IN X-D.
C             KBAR     NUMBER OF DEVISIONS IN Z-D.
C             JBAR     NUMBER OF DEVISIONS IN Y-D.
C             DELT IS THE TIME INCREMENT AND IT IS LIMITED BY STABULITY
C              IT SHOULD BE SMALLER MAX SIZE MESH OVER MAX VELOCITY.
C              TYPICALLY .25 TO .33 TIMES IT.
C             EPSI     IS THE CONVERGENCE CRITERIA.
C             NU       IS THE KINEMATIC VISCOSITY.
C             VI       IS THE INITIAL VELOCITY IN Y-D.
C             UI       IS THE INITIAL VELOCITY IN X-D.
C             WI       IS THE INITIAL VELOCITY IN Z-D.
C             GX,GY,GZ  ARE ACCELERATIONS.
C             ALPHA SHOULD BE BETWEEN 1. AND(DELT*VELOCITY)/MESH SIZE.
C
C-----------------------------------------------------------------------
C
```

```
C
C-----------------------------------------------------------------------
C
C
      INTEGER CYCLE
      REAL  NU,MU,MUN
       DIMENSION  U(11,11,11),V(11,11,11),W(11,11,11),P(11,11,11),
     #  UN(11,11,11),VN(11,11,11),WN(11,11,11)
      DIMENSION IBLOCK(11),VEL(11,11),TR(11,11)
C
      DIMENSION TK(11,11,11),TKN(11,11,11),E(11,11,11),EN(11,11,11),
     # MU(11,11,11),GG(11,11,11),MUN(11,11,11)
C
      DIMENSION YPLUSS(11,11),TAUS(11,11),XPLUSW(11,11),TAUW(11,11)
C
       DIMENSION XX(11,11,3),YY(11,11,3),X(111,3),Y(111,3)
C
C
C-----------------------------------------------------------------------
C--------DATA CARDS.----------------------------------------------------
      DELX=0.02
      DELY=0.02
      DELZ=0.01
      IBAR=9
      JBAR=9
      KBAR=7
      JJET=1
      KJET=8
      IJET=3
      CMU=0.09
      CD=1.0
      C1=1.44
      C2=1.92
      SK=1.0
      SE=1.21
      PK=0.4187
      CAPPA=PK
      ELOG=9.793
      URFVIS=0.7
      NU=15.68E-6
      VISCOS=1.983E-5
      DEN=1.2
      SC=1.0
C ---------- VELOCITY RATIO BETWEEN JET & MAIN STREAM
C
        RAT=6.0
        UIN=0.1
        UJET=RAT*UIN
C
      DATA  U,UN,V,VN,W,WN,P /9317 * 0.0 /
      DATA   EPSI,GX,GY,GZ,OMG,DZRO /1.E-3,0.0,0.0,0.0,1.7,1.0/
      ALPHA=0.6
C
C-----------------------------------------------------------------------
```

```
C
      UI= UIN
      VI=0.0
      WI=0.0
      PI=0.0
      ASJ=DELX*DELZ
      RDX = 1/DELX
      RDY = 1/ DELY
      RDZ = 1/ DELZ
      IMAX=IBAR+2
      JMAX = JBAR+2
      KMAX = KBAR + 2
      IM1=IBAR+1
      JM1=JBAR+1
      KM1=KBAR+1
      IM2=IBAR
      JM2=JBAR
      KM2=KBAR
C
      DELT=0.05
      DTMAX1=0.33*DELY/UJET
      DTMAX2=0.5*DELX**2*DELY**2*DELZ**2/((DELX**2+DELY**2+DELZ**2)*NU)
      ALFAMI=1.5*UJET*DELT/DELY
      BETA=OMG/(2*DELT*(1/(DELX**2)+1/(DELY**2)+1/(DELZ**2)))
      WRITE(6,92)DELT,DTMAX1,DTMAX2,DR
      WRITE(6,93)  ALPHA,ALFAMI
C
C     DELT  REDUCTION  FOR  STABILITY
C
      IF(DELT.GT.DTMAX1) DELT=DTMAX1
 92   FORMAT(//,2X,'DELT = ',E12.4,2X,'DTMAX1 = ',E12.4,2X,'DTMAX2 = ',
     @ E12.4,' DR = ',E12.6,/)
 93   FORMAT(/,2X,'ALPHA = ',E12.4,2X,'ALFAMI = ',E12.4,//)
C
C-----------------------------------------------------------------------
      T=0
      ITER=0
      CYCLE=0
C
C     INITIAL FIELD VALUES.
C
      DO  560  I=1,IMAX
      DO  560  J=1,JMAX
      DO  560  K=1,KMAX
      U(I,J,K)=UI
      V(I,J,K)=VI
      P(I,J,K)=PI
560   W(I,J,K)=WI
C
C
C
C
C
C
```

```
C-----------  INITIAL TURB ENERGY & DISSIAPATION
C
       TIN=0.01*(UIN)**2
       EIN=(TIN**1.5)/0.3
        IF(EIN.LT.1.E-4) EIN=1.E-4
C
       DO  557  I=1,IMAX
       DO  557  J=1,JMAX
       DO  557  K=1,KMAX
        TK(I,J,K)=TIN
        E(I,J,K)=EIN
         IF(EIN.LE.1.E-06) MU(I,J,K)=VISCOS
        MU(I,J,K)=CMU*DEN*(TIN**2)/EIN
 557   CONTINUE
C
C
C---------------------------------------------------------------------
C
       ASSIGN 5000 TO KRET
       GOTO 2000
1000   CONTINUE
       ITER=0
       FLG=1.0
       ASSIGN 3000 TO KRET
C
C---------------------------------------------------------------------
C
       DO 1100 I=2,IM1
       DO 1100 J=2,JM1
       DO 1100 K=2,KM1
C
       TMXE=   (MUN(I+1,J,K)+MUN(I,J,K))/2.0
C
       TMXW=   (MUN(I-1,J,K)+MUN(I,J,K))/2.0
C
       TMYN=   (MUN(I,J+1,K)+MUN(I,J,K))/2.0
C
       TMYS=   (MUN(I,J-1,K)+MUN(I,J,K))/2.0
C
       TMZO=   (MUN(I,J,K+1)+MUN(I,J,K))/2.0
C
       TMZI=   (MUN(I,J,K-1)+MUN(I,J,K))/2.0
C
C
       TMUN=(MUN(I+1,J,K)+MUN(I,J,K)+MUN(I+1,J+1,K)+MUN(I,J+1,K))/4.0
       TMUS=(MUN(I,J,K)+MUN(I+1,J,K)+MUN(I,J-1,K)+MUN(I+1,J-1,K))/4.0
       TMUO=(MUN(I,J,K+1)+MUN(I+1,J,K)+MUN(I+1,J,K+1)+MUN(I,J,K))/4.0
       TMUI=(MUN(I,J,K)+MUN(I+1,J,K)+MUN(I,J,K-1)+MUN(I+1,J,K-1))/4.0
       TMVE=(MUN(I,J,K)+MUN(I+1,J,K)+MUN(I+1,J+1,K)+MUN(I,J+1,K))/4.0
       TMVW=(MUN(I,J,K)+MUN(I,J+1,K)+MUN(I-1,J,K)+MUN(I-1,J+1,K))/4.0
       TMVO=(MUN(I,J,K)+MUN(I,J,K+1)+MUN(I,J+1,K)+MUN(I,J+1,K+1))/4.0
       TMVI=(MUN(I,J,K)+MUN(I,J+1,K)+MUN(I,J+1,K-1)+MUN(I,J,K-1))/4.0
       TMWE=(MUN(I,J,K+1)+MUN(I,J,K)+MUN(I+1,J,K)+MUN(I+1,J,K+1))/4.0
       TMWW=(MUN(I,J,K+1)+MUN(I,J,K)+MUN(I-1,J,K)+MUN(I-1,J,K+1))/4.0
```

```
      TMWN=(MUN(I,J,K)+MUN(I,J,K+1)+MUN(I,J+1,K)+MUN(I,J+1,K+1))/4.0
      TMWS=(MUN(I,J,K)+MUN(I,J,K+1)+MUN(I,J-1,K)+MUN(I,J-1,K+1))/4.0
C
      DVXY=(TMUN*(VN(I+1,J,K)-VN(I,J,K))-TMUS*(VN(I+1,J-1,K)-VN(I,J-1,K
     1 )))*RDX*RDY
      DWXZ=(TMUO*(WN(I+1,J,K)-WN(I,J,K))-TMUI*(WN(I+1,J,K-1)-WN(I,J,K-
     1 1)))*RDX*RDZ
      DUYX=(TMVE*(UN(I,J+1,K)-UN(I,J,K))-TMVW*(UN(I-1,J+1,K)-UN(I-1,J,
     1 K)))*RDY*RDX
      DWYZ=(TMVO*(WN(I,J+1,K)-WN(I,J,K))-TMVI*(WN(I,J+1,K-1)-WN(I,J,K-
     2 1)))*RDY*RDZ
      DUZX=(TMWE*(UN(I,J,K+1)-UN(I,J,K))-TMWW*(UN(I-1,J,K+1)-UN(I-1,J,
     2 K)))*RDZ*RDX
      DVZY=(TMWN*(VN(I,J,K+1)-VN(I,J,K))-TMWS*(VN(I,J-1,K+1)-VN(I,J-1,
     2 K)))*RDZ*RDY
C
      VISX=(RDX**2*(TMXE*(UN(I+1,J,K)-UN(I,J,K))-TMXW*(UN(I,J,K)-UN(I-1
     2 ,J,K)))*2.+RDY**2*(TMYN*(UN(I,J+1,K)-UN(I,J,K))-TMYS*(UN(I,J,K)-
     3 UN(I,J-1,K)))+DVXY+RDZ**2*(TMZO*(UN(I,J,K+1)-UN(I,J,K))-TMZI*(UN(
     4 I,J,K)-UN(I,J,K-1)))+DWXZ)/DEN
      FUX  =  RDX/4*  ((UN(I,J,K) +UN(I+1,J,K))**2
     1   +ALPHA*ABS(UN(I,J,K)+ UN(I+1,J,K))*( UN(I,J,K)-UN(I+1,J,K))
     2  -(UN(I-1,J,K)+UN(I,J,K))**2
     3   -ALPHA * ABS(UN(I-1,J,K)+UN(I,J,K))*(UN(I-1,J,K) - UN(I,J,K)))
      FUY=RDY/4*((VN(I,J,K)+VN( I+1,J,K))*(UN(I,J,K)+UN(I,J+1,K))
     1  +ALPHA *ABS(VN(I,J,K) +VN(I+1,J,K))*( UN(I,J,K)-UN(I,J+1,K))
     2     -(VN(I  ,J-1,K)+VN( I+1,J-1,K))* (UN(I,J-1,K)+UN(I,J,K))
     3  -ALPHA* ABS (VN(I,J-1,K)+VN(I+1,J-1,K))
     4                          *( UN(I,J-1,K) - UN(I,J,K)))
      FUZ = RDZ/4*(( WN(I,J,K) +WN(I+1,J,K))*(UN(I,J,K)+UN(I,J,K+1))
     1  + ALPHA*ABS(WN(I,J,K)+WN(I+1,J,K))*(UN(I,J,K)-UN(I,J,K+1))
     2 -(WN(I,J,K-1) +WN(I+1,J,K-1))*(UN(I,J,K-1) + UN(I,J,K))
     3  -ALPHA*ABS(WN(I,J,K-1) +WN(I+1,J,K-1))*( UN(I,J,K-1)-UN(I,J,K)))
C
      VISZ=(RDX**2*(TMXE*(WN(I+1,J,K)-WN(I,J,K))-TMXW*(WN(I,J,K)-WN(I-
     1 1,J,K)))+DUZX+RDY**2*(TMYN*(WN(I,J+1,K)-WN(I,J,K))-TMYS*(WN(I,J,K
     2 )-WN(I,J-1,K)))+DVZY+RDZ**2*(TMZO*(WN(I,J,K+1)-WN(I,J,K))-TMZI*(
     3 WN(I,J,K)-WN(I,J,K-1)))*2.)/DEN
      FWZ=RDZ/4   *  (   (WN(I,J,K)   +WN(I,J,K+1))  **2
     1 +ALPHA *   ABS(   WN(I,J,K)   +WN(I,J,K+1))   *   (  WN(I,J,K)
     2 -WN(I,J,K+1))
     3 -(    WN(I,J,K-1)   +WN(I,J,K))   **2
     4 -ALPHA   * ABS(   WN(I,J,K-1)   +WN(I,J,K))  *(WN(I,J,K-1)
     5  -WN(I,J,K)    )    )
      FWX=  RDX/4*  ((UN(I,J,K) +UN(I,J,K+1))*(WN(I,J,K) +WN(I+1,J,K))
     1 +ALPHA   *ABS( UN(I,J,K)    +UN(I,J,K+1))   * (WN(I,J,K)
     2 -WN(I+1,J,K)       )
     3 -(UN(I-1,J,K)   +UN(I-1,J,K+1)     )*(WN(I-1,J,K)    +WN(I,J,K))
     4  -ALPHA    *ABS(   UN(I-1,J,K)  +UN(I-1,J,K+1)  ) *
     5    (WN(I-1,J,K)    -WN(I,J,K)       )       )
      FWY=RDY/4 * ((VN(I,J,K)+VN(I,J,K+1))*(WN(I,J,K) + WN(I,J+1,K))
     1  +ALPHA  *ABS(   VN(I,J,K)   +VN(I,J,K+1)     )   *( WN(I,J,K)
     2  -WN(I,J+1,K) )
     3   -( VN(I,J-1,K)   +VN(I,J-1,K+1)  )*(WN(I,J-1,K) +WN(I,J,K)   )
```

```
     4  -ALPHA  * ABS (  VN(I,J-1,K)  +VN(I,J-1,K+1)    )*
     5  (   WN(I,J-1,K)    -WN(I,J,K)     )      )
C
       VISY=(RDX**2*(TMXE*(VN(I+1,J,K)-VN(I,J,K))-TMXW*(VN(I,J,K)-VN(I-
     1 1,J,K)))+ DUYX + RDY**2* (TMYN* (VN(I,J+1,K)-VN(I,J,K))-TMYS*
     2 (VN(I,J,K)-VN(I,J-1,K)))* 2. + RDZ**2* (TMZO* (VN(I,J,K+1)-
     3 VN(I,J,K))-TMZI* (VN(I,J,K)-VN(I,J,K-1)))+ DWYZ )/DEN
       FVZ=   RDZ/4 * ((WN(I,J,K) + WN(I,J+1,K)) * ( VN(I,J,K)
     1+VN(I,J,K+1) )+ALPHA *ABS( WN(I,J,K) +WN(I,J+1,K) ) *(VN(I,J,K)
     2   -VN(I,J,K+1) ) - ( WN(I,J,K-1) +WN(I,J+1,K-1) ) *(VN(I,J,K-1)
     3 +VN(I,J,K)) -ALPHA * ABS( WN(I,J,K-1) +WN(I,J+1,K-1)  )*
     4   (  VN(I,J,K-1) - VN(I,J,K) ) )
       FVX = RDX/ 4 *((UN(I,J,K) + UN(I,J+1,K))*(VN(I,J,K )+VN(I+1,J,K))
     1  + ALPHA *ABS(UN(I,J,K) +UN(I,J+1,K))*(VN(I,J ,K)-VN(I+1,J,K))
     2  -(UN(I-1,J,K)+UN(I-1 , J+1,K))*(VN( I-1,J,K)+ VN(I,J,K))
     3  - ALPHA*ABS(UN(I-1,J,K) +UN(I-1,J+1,K)) *(VN(I-1,J,K)-VN(I,J,K))
     4     )
       FVY = RDY/4*((VN(I,J,K)  +VN(I,J+1,K))**2
     1  + ALPHA*ABS(VN(I,J,K)+VN(I,J+1,K))*( VN(I,J,K) -VN(I,J+1,K))
     2  -(VN(I,J-1,K)+VN(I,J,K) )**2
     3  -ALPHA*ABS(VN(I,J-1,K) +VN(I,J,K))*(VN (I,J-1,K) -VN(I,J,K)))
C
C
C      IF(I.EQ.2) GO TO 1200
       IF(J.EQ.2) GO TO 1200
C
       DVX=((VN(I,J,K)+VN(I+1,J,K)+VN(I,J-1,K)+VN(I+1,J-1,K)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I-1,J,K)+VN(I-1,J-1,K)))/
     2  (4.*DELX)
       DUY=((UN(I,J,K)+UN(I,J+1,K)+UN(I-1,J+1,K)+UN(I-1,J,K)) -
     1  (UN(I,J,K)+UN(I,J-1,K)+UN(I-1,J-1,K)+UN(I-1,J,K))) /
     2  (4.*DELY)
       DWY=((WN(I,J,K)+WN(I,J+1,K)+WN(I,J+1,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I,J-1,K)+WN(I,J-1,K-1))) /
     2  (4.*DELY)
       DVZ=((VN(I,J,K)+VN(I,J,K+1)+VN(I,J-1,K)+VN(I,J-1,K+1)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I,J-1,K-1)+VN(I,J,K-1))) /
     2  (4.*DELZ)
       DUZ=((UN(I,J,K)+UN(I,J,K+1)+UN(I-1,J,K)+UN(I-1,J,K+1)) -
     1  (UN(I,J,K)+UN(I-1,J,K)+UN(I-1,J,K-1)+UN(I,J,K-1))) /
     2  (4.*DELZ)
       DWX=((WN(I,J,K)+WN(I+1,J,K)+WN(I+1,J,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I-1,J,K-1)+WN(I-1,J,K))) /
     2  (4.*DELX)
       GG(I,J,K) = MUN(I,J,K)* (2.*(((UN(I,J,K)-UN(I-1,J,K))/DELX)**2 +
     1  ((VN(I,J,K)-VN(I,J-1,K))/DELY)**2 +((WN(I,J,K) -WN(I,J,
     2  K-1))/DELZ)**2) + (DVX+DUY)**2 + (DWY+DVZ)**2 +
     3    (DUZ+DWX)**2 )
1200   VISK=((RDX**2* (TMXE* (TKN(I+1,J,K)-TKN(I,J,K)) - TMXW* (
     1 TKN(I,J,K)-TKN(I-1,J,K))) + RDY**2* (TMYN* (TKN(I,J+1,K) -
     2 TKN(I,J,K)) -TMYS* (TKN(I,J,K)-TKN(I,J-1,K))) + RDZ**2*
     3 (TMZO* (TKN(I,J,K+1)-TKN(I,J,K))-TMZI* (TKN(I,J,K)-TKN
     4 (I,J,K-1))))/SK)/DEN
C
```

```
      SORK1=EN(I,J,K)
      SORK= GG(I,J,K)/DEN
      SORK2=1.0+DELT * EN(I,J,K) /(TKN(I,J,K))
      VISK=VISK+SORK
C
      FKX= (UN(I,J,K)*(TKN(I,J,K)+TKN(I+1,J,K)) + ALPHA * ABS
     1 (UN(I,J,K))*(TKN(I,J,K)-TKN(I+1,J,K)) - UN(I-1,J,K) *
     2 (TKN(I-1,J,K)+TKN(I,J,K)) - ALPHA * ABS(UN(I-1,J,K)) *
     3 (TKN(I-1,J,K)-TKN(I,J,K))) /(2.*DELX)
      FKY= (VN(I,J,K)*(TKN(I,J,K)+TKN(I,J+1,K))+ ALPHA *ABS
     1 (VN(I,J,K)) *(TKN(I,J,K) - TKN(I,J+1,K)) -VN(I,J-1,K) *
     2 (TKN(I,J-1,K)+TKN(I,J,K)) - ALPHA * ABS(VN(I,J-1,K)) *
     3 (TKN(I,J-1,K)-TKN(I,J,K)))/(2.*DELY)
      FKZ= (WN(I,J,K)*(TKN(I,J,K)+TKN(I,J,K+1)) + ALPHA * ABS
     1 (WN(I,J,K))* (TKN(I,J,K)-TKN(I,J,K+1)) - WN(I,J,K-1) *
     2 (TKN(I,J,K-1)+TKN(I,J,K)) - ALPHA * ABS(WN(I,J,K-1)) *
     3 (TKN(I,J,K-1)-TKN(I,J,K)))/(2.*DELZ)
C
C     CALCULATE NEW TIME VALUES.
C
      U(I,J,K) = UN(I,J,K) +DELT*(RDX*(P(I,J,K)  -P(I+1,J,K))
     1   + GX + VISX - FUX -FUY -FUZ)
      V(I,J,K)=VN(I,J,K)+DELT*(RDY*(P(I,J,K)-P(I,J+1,K))+GY
     1 -FVX-FVY-FVZ+VISY)
      W(I,J,K) =   WN(I,J,K)  +  DELT*(RDZ *   (P(I,J,K) -P(I,J,K+1))
     1 +GZ -FWX -FWY -FWZ +VISZ)
      TK(I,J,K) =(TKN(I,J,K) + DELT* (-FKX-FKY-FKZ +VISK))/SORK2
C     WRITE(6,37) I,J,K,SORK,SORK1,SORK2,VISK,TK(I,J,K)
      FKSUM=FKX+FKY+FKZ
C     WRITE(6,39) I,J,K,FKX,FKY,FKZ,FKSUM
C     IF(TK(I,J,K).LT.0.0) TK(I,J,K)=TIN
      IF(TK(I,J,K).LT.1.E-4) TK(I,J,K)=1.E-4
C
      VISOLD=MUN(I,J,K)
      IF(EN(I,J,K).LT.1.E-4) EN(I,J,K)=1.E-4
      MU(I,J,K)=(CMU*DEN*(TKN(I,J,K))**2)/EN(I,J,K) + VISCOS
      MU(I,J,K)=URFVIS*MU(I,J,K)+(1.-URFVIS)*VISOLD
C
C
C37   FORMAT(1X,3(I2,1X),5(E14.6))
C39   FORMAT(1X,3(I2,1X),'CONV IN X,Y,Z',4(E14.6))
C
1100  CONTINUE
C-----------------------------------------------------------------------
C
2000  CONTINUE
C
C
C
C  ==========BOUNDARY  CONDITIONS.====================
C
C     VERTICAL
C       X - Y  PLANE
C
```

```
      DO  20  J=1,JMAX
      DO  20  I=1,IMAX
C
C     ------  END  PLANE  --  FREE  SLIP    K=1
C
C
      U(I,J,1)=U(I,J,2)
      V(I,J,1)=V(I,J,2)
      W(I,J,1)=0.0
      TK(I,J,1)=TK(I,J,2)
      E(I,J,1)=E(I,J,2)
      MU(I,J,1)=MU(I,J,2)
C
C     SYMMETRY  PLANE  FREE SLIP     K=KMAX
C
C
      W(I,J,KM1)=0.0
      W(I,J,KMAX)=W(I,J,KM1)
      U(I,J,KMAX)=U(I,J,KM1)
      V(I,J,KMAX)=V(I,J,KM1)
      TK(I,J,KMAX)=TK(I,J,KM1)
      E(I,J,KMAX)=E(I,J,KM1)
       MU(I,J,KMAX)=MU(I,J,KM1)
 20   CONTINUE
C
C      Y  -  Z  PLANE    FREE SLIP      I=1     INLET FREE STREAM VEL.
C
      DO  21  J=1,JMAX
      DO  21  K=1,KMAX
      TK(1,J,K)=0.01*UIN**2
      E(1,J,K)=(TK(1,J,K)**1.5)/0.3
      MU(1,J,K)=MU(2,J,K)
      W(1,J,K)=0.0
      V(1,J,K)=0.0
       U(1,J,K)=UIN
 21   CONTINUE
C
C     HORIZONTAL
C       X - Z  PLANE
C
      DO  22  I=1,IMAX
      DO  22  K=1,KMAX
C
C      TOP   FREE SLIP        J=JMAX
C
      V(I,JM1,K)=0.0
      V(I,JMAX,K)=V(I,JM1,K)
      U(I,JMAX,K)=U(I,JM1,K)
      W(I,JMAX,K)=W(I,JM1,K)
      TK(I,JMAX,K)=TK(I,JM1,K)
      E(I,JMAX,K)=E(I,JM1,K)
      MU(I,JMAX,K)=MU(I,JM1,K)
C
C     ----  BOTTOM   --  FREE SLIP       J=1
```

```
C
      V(I,1,K)=0.0
      U(I,1,K)=U(I,2,K)
      W(I,1,K)=W(I,2,K)
      TK(I,1,K)=TK(I,2,K)
      E(I,2,K)=(((CMU*CD)**0.75)*TK(I,2,K)**1.5)/(CD*PK*DELY*0.5)
      E(I,1,K)=E(I,2,K)
      MU(I,1,K)=MU(I,2,K)
 22   CONTINUE
C
C     INLET B.CS. FOR JET
C
C
      DO  27  K=KJET,KM1
      DO  27  I=IJET,3
      U(I,JJET,K)=0.0
      V(I,JJET,K)=UJET
      W(I,JJET,K)=0.0
      TK(I,JJET,K)=0.03*(UJET**2)
      E(I,JJET,K)=(TK(I,JJET,K)**1.5)/0.01
 27   CONTINUE
C
        AREA=DELZ*DELY*JBAR*KBAR
        FLUXIN=AREA*UIN+UJET*DELX*DELZ
C
        UOUT=FLUXIN/AREA
C
        FLUXOU=0.0
        DO 26 J=2,JM1
        DO 26 K=2,KM1
 26       FLUXOU=FLUXOU+DELZ*DELY*U(IM2,J,K)
        UINC=(FLUXIN-FLUXOU)/AREA
C
       IF(ITER.GT.0) GO TO 29
C
C       OUTLET BOUNDARY CONDITIONS
C
       DO 25 J=2,JM1
       DO 25 K=2,KM1
       U(IM1,J,K)=U(IM2,J,K)+UINC
C      U(IM1,J,K)=UOUT
       U(IMAX,J,K)=U(IM1,J,K)
       V(IMAX,J,K)=V(IM1,J,K)
       W(IMAX,J,K)=W(IM1,J,K)
       MU(IMAX,J,K)=MU(IM1,J,K)
       TK(IMAX,J,K)=TK(IM1,J,K)
        E(IMAX,J,K)=E(IM1,J,K)
25      CONTINUE
C
C
C
29    IF(CYCLE.EQ.0)  GO  TO  2990
      IF(ITER.GT.0)   GO   TO  2990
C
```

```
C       DO 1101 I=1,IMAX
C       WRITE(6,47) I
C       WRITE(6,57)
C       DO 1102 JJ=1,JMAX
C       J=JMAX-JJ+1
C1102   WRITE(6,44) (TK(I,J,K),K=1,KMAX)
C       WRITE(6,58)
C       DO 1103 JJ=1,JMAX
C       J=JMAX-JJ+1
C1103   WRITE(6,44) (E(I,J,K),K=1,KMAX)
C
C1101   CONTINUE
C
        DO 1110 I=1,IMAX
        DO 1110 J=1,JMAX
        DO 1110 K=1,KMAX
         EN(I,J,K)=E(I,J,K)
 1110   CONTINUE
C
C
C--------------- DISSIAPATION  EQUATION  -----------------
C
        DO  1111 I=2,IM1
        DO  1111 J=3,JM1
        DO  1111 K=2,KM1
C
         IF(TKN(I,J,K).LT.2.E-4) TKN(I,J,K)=2.E-4
C
        TMXE= (MUN(I+1,J,K)+MUN(I,J,K))/2.0
C
        TMXW= (MUN(I-1,J,K)+MUN(I,J,K))/2.0
C
        TMYN= (MUN(I,J+1,K)+MUN(I,J,K))/2.0
C
        TMYS= (MUN(I,J-1,K)+MUN(I,J,K))/2.0
C
        TMZO= (MUN(I,J,K+1)+MUN(I,J,K))/2.0
C
        TMZI= (MUN(I,J,K-1)+MUN(I,J,K))/2.0
C
        FEX=(UN(I,J,K)*(EN(I,J,K)+EN(I+1,J,K))+ALPHA* ABS(UN(I,J,K))*
     1 (EN(I,J,K)-EN(I+1,J,K))-UN(I-1,J,K)*(EN(I-1,J,K)+EN(I,J,K))
     2 -ALPHA *ABS(UN(I-1,J,K)) *(EN(I-1,J,K)-EN(I,J,K))) /
     3 (2. * DELX )
        FEY=( VN(I,J,K)* ( EN(I,J,K)+ EN(I,J+1,K)) + ALPHA*ABS(VN(I,J,K)
     1 ) * ( EN(I,J,K)- EN(I,J+1,K)) - VN(I,J-1,K)*(EN(I,J-1,K)+EN(I,J,K
     2 )) - ALPHA * ABS(VN(I,J-1,K)) * ( EN(I,J-1,K)- EN(I,J,K))) /
     3 (2. * DELY )
        FEZ=(WN(I,J,K) * ( EN(I,J,K)+ EN(I,J,K+1)) +ALPHA*ABS( WN(I,J,K
     1 ))*(EN(I,J,K)-EN(I,J,K+1)) - WN(I,J,K-1) * (EN(I,J,K-1)+EN(I,J,K
     2 )) - ALPHA * ABS(WN(I,J,K-1)) * ( EN(I,J,K-1)- EN(I,J,K))) /
     3 (2. * DELZ)
        DVX=((VN(I,J,K)+VN(I+1,J,K)+VN(I,J-1,K)+VN(I+1,J-1,K)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I-1,J,K)+VN(I-1,J-1,K)))/
```

```
     2  (4.*DELX)
        DUY=((UN(I,J,K)+UN(I,J+1,K)+UN(I-1,J+1,K)+UN(I-1,J,K)) -
     1  (UN(I,J,K)+UN(I,J-1,K)+UN(I-1,J-1,K)+UN(I-1,J,K))) /
     2  (4.*DELY)
        DWY=((WN(I,J,K)+WN(I,J+1,K)+WN(I,J+1,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I,J-1,K)+WN(I,J-1,K-1))) /
     2  (4.*DELY)
        DVZ=((VN(I,J,K)+VN(I,J,K+1)+VN(I,J-1,K)+VN(I,J-1,K+1)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I,J-1,K-1)+VN(I,J,K-1))) /
     2  (4.*DELZ)
        DUZ=((UN(I,J,K)+UN(I,J,K+1)+UN(I-1,J,K)+UN(I-1,J,K+1)) -
     1  (UN(I,J,K)+UN(I-1,J,K)+UN(I-1,J,K-1)+UN(I,J,K-1))) /
     2  (4.*DELZ)
        DWX=((WN(I,J,K)+WN(I+1,J,K)+WN(I+1,J,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I-1,J,K-1)+WN(I-1,J,K))) /
     2  (4.*DELX)
        GG(I,J,K) = MUN(I,J,K)* (2.*(((UN(I,J,K)-UN(I-1,J,K))/DELX)**2 +
     1  ((VN(I,J,K)-VN(I,J-1,K))/DELY)**2 +((WN(I,J,K) -WN(I,J,
     2  K-1))/DELZ)**2) + (DVX+DUY)**2 + (DWY+DVZ)**2 +
     3    (DUZ+DWX)**2 )
        VISEX=RDX**2*(TMXE* ( EN(I+1,J,K)- EN(I,J,K))-TMXW * (EN(I,J,
     1 K)- EN(I-1,J,K)))
        VISEY=RDY**2 *(TMYN *( EN(I,J+1,K)- EN(I,J,K))-TMYS*( EN(I,J,K)
     1 - EN(I,J-1,K)))
        VISEZ=RDZ**2 *(TMZO *( EN(I,J,K+1)- EN(I,J,K))-TMZI*(EN(I,J,K)
     2 - EN(I,J,K-1)))
        VIS=(VISEX +VISEY +VISEZ )/(SE*DEN)
        GC=GG(I,J,K)
        ENC=EN(I,J,K)
        TKNC=TKN(I,J,K)
        TKC=TK(I,J,K)
        SORE= GG(I,J,K)*C1 * EN(I,J,K) /(TKN(I,J,K)*DEN)
        SORE1=C2 *(EN(I,J,K)**2)/(TKN(I,J,K))
        SORE2=1.0 + DELT * C2  *EN(I,J,K) /(TKN(I,J,K))
        VISE=VIS + SORE
C
C
C38     FORMAT(1X,3(I2,1X),7(E12.5))
C
C
       E(I,J,K)=(EN(I,J,K) + DELT * (-FEX-FEY-FEZ+VISE))/SORE2
C       WRITE(6,38) I,J,K,SORE,SORE1,SORE2,VIS,VISE,E(I,J,K),GC
C      IF(E(I,J,K).LT.0.0) E(I,J,K)=EIN
        IF(E(I,J,K).LT.1.E-4) E(I,J,K)=1.E-4
       FESUM=FEX+FEY+FEZ
C      WRITE(6,39) I,J,K,FEX,FEY,FEZ,FESUM
C
C
1111    CONTINUE
C
C
2990    CONTINUE
C
        GOTO KRET,(3000,5000)
```

```
3000    CONTINUE
        IF(FLG.EQ.0.)GO TO 4000
        ITER=ITER+1
        IF(ITER.LT.150)  GO  TO  3050
        IF(CYCLE .LT. 64) GO TO 4000
        T= 1E+10
        GOTO 5000
3050    FLG=0.0
C
C-------------------------------------------------------------------
C  ====CONVERGENCE===
C       PRESSURE ITERATIONS.
C
        DO 3500 I=2,IM1
        DO 3500 J=2,JM1
        DO 3500 K=2,KM1
       D=1/DELX*(U(I,J,K)-U(I-1,J,K)) +1/DELY*(V(I,J,K) -V(I,J-1,K))
     +   + 1/DELZ*(W(I,J,K)-W(I,J,K-1))
        IF(ABS(D/DZRO).GE.EPSI)  FLG=1.
        DELP=-D*BETA
        P(I,J,K)=P(I,J,K)+DELP
        U(I,J,K)=U(I,J,K)+DELT*DELP/DELX
        U(I-1,J,K)=U(I-1,J,K)-DELT*DELP/DELX
        V(I,J,K)=V(I,J,K)+DELT*DELP/DELY
        V(I,J-1,K)=V(I,J-1,K)-DELT*DELP/DELY
        W(I,J,K)=W(I,J,K)+DELT*DELP/DELZ
        W(I,J,K-1)=W(I,J,K-1)-DELT*DELP/DELZ
3500    CONTINUE
        GOTO 2000
4000    CONTINUE
5000    CONTINUE
C
C------------SHEAR STREES ON THE BOTTOM WALL
C
        DO  5100  I=2,IM1
        DO  5100  K=2,KM1
         J=2
        UAVG=(U(I,J,K)+U(I+1,J,K))/2.
        WAVG=(W(I,J,K)+W(I,J,K+1))/2.
        UEFF=SQRT(UAVG*UAVG+WAVG*WAVG)
        DENU=DEN
        YP=DELY/2.
        SQRTK=SQRT(TK(I,J,K))
        CMUPQ=CMU**0.25
        YPLUSS(I,K)=DENU*CMUPQ*SQRTK*YP/VISCOS
        IF(YPLUSS(I,K).LE.11.63) GO TO 5101
        TMULT=DENU*CMUPQ*SQRTK*CAPPA/ALOG(ELOG*YPLUSS(I,K))
        TAUS(I,K)=-TMULT*UEFF
        GO  TO  5102
5101    TAURX=-VISCOS*UAVG/YP
        TAURW=-VISCOS*WAVG/YP
        TAUS(I,K)=SQRT(TAURX*TAURX+TAURW*TAURW)
5102    DUZ=((U(I,J,K)+U(I,J,K+1)+U(I-1,J,K)+U(I-1,J,K+1)) -
     1   (U(I,J,K)+U(I-1,J,K)+U(I-1,J,K-1)+U(I,J,K-1))) /
```

```
     2  (4.*DELZ)
        DWX=((W(I,J,K)+W(I+1,J,K)+W(I+1,J,K-1)+W(I,J,K-1)) -
     1  (W(I,J,K)+W(I,J,K-1)+W(I-1,J,K-1)+W(I-1,J,K))) /
     2  (4.*DELX)
        GG(I,J,K)=MU(I,J,K)*2.*(((U(I,J,K)-U(I-1,J,K))/DELX)**2+
     1 ((V(I,J,K)-V(I,J-1,K))/DELY)**2 + ((W(I,J,K)-W(I,J,K-1))/DELZ)
     2 **2 )+ TAUS(I,K)**2/MU(I,J,K) +MU(I,J,K)* (DUZ+DWX)**2
5100    CONTINUE
C
C
        GO TO 5210
C
C----------- SHEAR STRESS ON THE SIDE WALL
C
        DO  5200  J=2,JM1
        DO  5200  K=2,KM1
         I=2
        VAVG=(V(I,J,K)+V(I,J+1,K))/2.
        WAVG=(W(I,J,K)+W(I,J,K+1))/2.
        VEFF=SQRT(VAVG*VAVG+WAVG*WAVG)
        XP=DELX/2.
        DENV=DEN
        SQRTK=SQRT(TK(I,J,K))
        XPLUSW(J,K)=DENV*CMUPQ*SQRTK*XP/VISCOS
        IF(XPLUSW(J,K).LE.11.63) GO TO 5201
        TMULT=DENV*CMUPQ*SQRTK*CAPPA/ALOG(ELOG*XPLUSW(J,K))
        TAUW(J,K)=-TMULT*VEFF
        GO TO 5202
5201    TAUXR=VISCOS*VAVG/XP
        TAUXW=VISCOS*WAVG/XP
        TAUW(J,K)=SQRT(TAUXR*TAUXR+TAUXW*TAUXW)
5202   DWY=((WN(I,J,K)+WN(I,J+1,K)+WN(I,J+1,K-1)+WN(I,J,K-1)) -
     1  (WN(I,J,K)+WN(I,J,K-1)+WN(I,J-1,K)+WN(I,J-1,K-1))) /
     2  (4.*DELY)
       DVZ=((VN(I,J,K)+VN(I,J,K+1)+VN(I,J-1,K)+VN(I,J-1,K+1)) -
     1  (VN(I,J,K)+VN(I,J-1,K)+VN(I,J-1,K-1)+VN(I,J,K-1))) /
     2  (4.*DELZ)
       GG(I,J,K)=MU(I,J,K)*2.*(((U(I,J,K)-U(I-1,J,K))/DELX)**2 +
     1 ((V(I,J,K)-V(I,J-1,K))/DELY)**2 + ((W(I,J,K)-W(I,J,K-1)) /
     2 DELZ)**2)+ TAUW(J,K)**2/MU(I,J,K) + MU(I,J,K)*(DVZ+DWY)**2
5200   CONTINUE
C
5210    CONTINUE
C
C========PRINT=====
       WRITE(6,49)  ITER,T,CYCLE
       WRITE(6,666) V(3,3,3),W(3,3,3),U(3,3,3),V(3,2,4),W(3,2,4),
     # U(3,2,4)
        IF(CYCLE.NE.320) GO TO 636
C
        DO 5300 I=1,IMAX
        DO 5300 J=1,JMAX
        K=8
        XX(I,J,1)=(0.4*I)-0.2-U(I,J,K)/2.
```

```
        XX(I,J,2)=XX(I,J,1)+U(I,J,K)/2.
        XX(I,J,3)=XX(I,J,2)+U(I,J,K)/2.
        YY(I,J,1)=(0.6*J)-0.3-V(I,J,K)/2.
        YY(I,J,2)=YY(I,J,1)+V(I,J,K)/2.
        YY(I,J,3)=YY(I,J,2)+V(I,J,K)/2.
5300     CONTINUE
        DO 5400 I=1,IMAX
        DO 5400 J=1,JMAX
        DO 5400 L=1,3
        M=I*J
        X(M,L)=XX(I,J,L)
        Y(M,L)=YY(I,J,L)
5400    CONTINUE
C
C
        WRITE(14) X
        WRITE(14) Y
C
C
636      CONTINUE
C
         DO 5500 I=1,IMAX
         DO 5500 J=1,JMAX
          VEL(I,J)=SQRT(V(I,J,8)**2)+(U(I,J,8)**2)
          TR(I,J)=(J-1)*DELY -DELY/2.
C
5500    CONTINUE
C
C
C
        IF(CYCLE.EQ.0)  GO TO 5152
       IF(CYCLE.EQ.1)  GO TO 5152
       IF(CYCLE.EQ.2)  GO TO 5152
       IF(CYCLE.EQ.320) GO TO 5152
       DO  66  II=10,400,50
C      DO  66  II=3,400
         IF(CYCLE.EQ.II)  GO TO 5152
 66    CONTINUE
C
          GOTO 5251
C
 5152  CONTINUE
C
C-------------------------------------------------------------------------
C-----------   OUTPUT   ------------------------------------------------
C
C-------------------------------------------------------------------------
C
5230    CONTINUE
       DO 5250 KK = 1,KMAX
       K=KMAX-KK+1
       WRITE(6,47) K
       WRITE(6,48)
       DO 5252 JJ =  1, JMAX
```

```
      J = JMAX -JJ+1
 5252 WRITE(6,44) (U(I,J,K),I=1,IMAX)
      WRITE(6,50)
      DO 5253 JJ = 1,JMAX
      J = JMAX -JJ+1
 5253 WRITE(6,44) (V(I,J,K),I=1,IMAX)
      WRITE(6,51)
      DO 5254 JJ = 1,JMAX
      J = JMAX -JJ+1
 5254 WRITE(6,44) (W(I,J,K),I=1,IMAX)
      WRITE(6,52)
      DO 5255 JJ = 1,JMAX
      J = JMAX -JJ+1
 5255 WRITE(6,44) (P(I,J,K),I=1,IMAX)
      WRITE(6,57)
      DO  5258  JJ=1,JMAX
      J=JMAX-JJ+1
 5258 WRITE(6,44) (TK(I,J,K),I=1,IMAX)
      WRITE(6,58)
      DO  5259  JJ=1,JMAX
      J=JMAX-JJ+1
 5259 WRITE(6,44) (E(I,J,K),I=1,IMAX)
      WRITE(6,59)
      DO  5260  JJ=1,JMAX
      J=JMAX-JJ+1
 5260 WRITE(6,44) (MU(I,J,K),I=1,IMAX)
      WRITE(6,60)
      WRITE(6,44) (YPLUSS(I,K),I=1,IMAX)
C
      WRITE(6,61)
      WRITE(6,44) (XPLUSW(J,K),J=1,JMAX)
        WRITE(6,62)
        DO 5261 JJ=1,JMAX
        J=JMAX-JJ+1
5261    WRITE(6,44) (VEL(I,J),I=1,IMAX)
C
        WRITE(6,63)
        DO 5262 JJ=1,JMAX
        J=JMAX-JJ+1
5262    WRITE(6,44) (TR(I,J),I=1,IMAX)
C
5250  CONTINUE
5251  CONTINUE
      DO 6100 I=1,IMAX
      DO 6100 J=1,JMAX
      DO 6100 K=1,KMAX
      VN(I,J,K)=V(I,J,K)
      UN(I,J,K)=U(I,J,K)
      WN(I,J,K)=W(I,J,K)
      TKN(I,J,K)=TK(I,J,K)
      EN(I,J,K)=E(I,J,K)
      MUN(I,J,K)=MU(I,J,K)
 6100 CONTINUE
      T=T+DELT
```

```
      IF(T.GT.320*DELT)  GO TO 6500
      IF(CYCLE .EQ.0 )GOTO 6200
C     IF(ITER.LE.1)  GO TO 6550
6200   CYCLE=CYCLE+1
      GOTO 1000
6500  STOP
 6550 WRITE(6,53)
      STOP
C
C-----------------------------------------------------------------------
C
 47   FORMAT(//,20X,'K =',I2,/,19X,  7('='))
 49   FORMAT(//,6X,'ITER=',I5,9X,'TIME=',1PE12.5,10X,'CYCLE=',I4,/)
 44   FORMAT(1X,12(E11.4))
 48   FORMAT(//,2X,'U-FIELD',/)
 50   FORMAT(//,2X,'V-FIELD',/)
 51   FORMAT(//,2X,'W-FIELD',/)
 52   FORMAT(//,2X,'P-FIELD',/)
53    FORMAT (//,10X,7('='),'CONVERGENCE HAS  BEEN REACHED. ')
 57   FORMAT(//,2X,'TK-FIELD',/)
 58   FORMAT(//,2X,'E -FIELD',/)
 59   FORMAT(//,2X,'MU-FIELD',/)
 60   FORMAT(//,2X,'YPLUSS ',4X,'J = 2 ',/)
 61   FORMAT(//,2X,'XPLUSW ',4X,'I = 2 ',/)
 62    FORMAT(//,2X,'VELOCITY-RES',/)
 63    FORMAT(//,2X,'TRAJECTORY ',/)
 666  FORMAT(//,10X,'V(3,3,3)=',E11.4,2X,'W(3,3,3)=',E11.4,2X,
     # 'U(3,3,3)=',E11.4,//,10X,'V(3,2,4)=',E11.4,2X,'W(3,2,4)=',E11.4,
     # 2X,'U(3,2,4)=',E11.4,//)
      END
```

```
C
C
C*****************************************************************************
C**
C**
C**         THREE-DIMENSIONAL PREDICTION TECHNIQUE FOR TURBULENT
C**            SWIRLING FLOWFIELDS IN CYLINDRICAL COORDINATES
C**
C**                              PH.D. THESIS
C**                                   BY
C**                             AHMED A. BUSNAINA
C**
C**
C**                                 MAY, 1983
C**               SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING
C**                          OKLANOMA STATE UNIVERSITY
C**                            STILLWATER , OKLAHOMA
C**
C**
C**
C*****************************************************************************
C*****************************************************************************
C
C-------- APPLICATION 3 : DILUTION JETS IN GAS TURBINE COMBUSTORS ------
C
C*****************************************************************************
C*****************************************************************************
C
C
C
      DIMENSION U(12,12,9),V(12,12,9),W(12,12,9),P(12,12,9),
     1UN(12,12,9),VN(12,12,9),WN(12,12,9)
      DIMENSION TK(12,12,9),TKN(12,12,9),E(12,12,9),EN(12,12,9)
     1 ,MU(12,12,9),MUN(12,12,9)
     1 ,GG(12,12,9),RPLUSS(12,12),TAUS(12,12),ZPLUSW(12,12),
     2 TAUW(12,12),RU(12,12,9),RV(12,12,9),TAUN(12,12),TNWRTE(12,12),
     3 TNWRZE(12,12),VANB(7),VEL(12,12),TR(12,12),
     3 TSWRTE(12,12),TSWRZE(12,12),TWWRZO(12,12),TWWTZO(12,12)
C
      DIMENSION XX(12,12,3),YY(12,12,3),X(432),Y(432),
     1  VV(12,9,5),UU(12,9,5),WW(12,9,5)
C
      DATA  VANB /0.0,45.0,55.0,60.0,65.0,68.0,70.0/
C
      REAL NU,MU,MUN
      INTEGER CYCLE,CYCMAX
C
C     PARAMETER SETTING
C
      EPSI=1.0E-3
      OMG=1.7
      OMG=1.7
      GR=0.
      GZ=0.
```

```
      GTH=0.
      DZRO=1.0
      IN1=2
      IN2=6
      KIN=4
      RIN=0.0
      VINLET=0.1*100.0
C
C--------- DILUTON JET DATA ----------------------------------------
C
       RAT=1.0
C
       UDIL=VINLET*RAT
       KDIL=5
       JDIL=6
C
C
C---------- SWIRL VELOCITY AND LOOP --------------------------
C
      LFS=2
      WIN=VINLET*TAN(VANB(LFS)*3.14159/180.)
C
      NS=4
       IWRITE=0
       ITMAX=60
C
C--------MAXIMUM NUMBER OF CYCLES
C
      CYCMAX=300
      NU=2.0E-4
      ALPHA=0.6
      DELR=0.015
      DELZ=0.06
      DELTH=0.14957
      DELT=0.008/100.0
      IBAR=10
      JBAR=10
      KBAR=7
      DIAM=IBAR*DELR*2.0 + RIN
C     PARAMETER
      IMAX=IBAR+2
      JMAX=JBAR+2
      KMAX=KBAR+2
      IM1=IBAR+1
      JM1=JBAR+1
      KM1=KBAR+1
      IM2=IBAR
      JM2=JBAR
      KM2=KBAR
      RDR=1./DELR
      RDZ=1./DELZ
      RDTH=1./DELTH
      CMU=0.09
      CMUPQ=CMU**0.25
```

```
      CD=1.0
      C1=1.44
      C2=1.92
      SK=1.0
      SE=1.21
      PK=0.4187
      CAPPA=PK
      ELOG=9.793
      URFVIS=0.7
      NU=15.68E-6
      VISCOS=1.983E-5
      DEN=1.2
      SC=1.0
C
C     ------------------------------
C     INITIALIZATION
      T=0
      ITER=0
      CYCLE=0
C     GUESS INITIAL VELOCITY FIELD
      DO 560 I=1,IMAX
      DO 560 J=1,JMAX
      DO 560 K=1,KMAX
       U(I,J,K)=0.0
      V(I,J,K)=VINLET/4.0
      W(I,J,K)=WIN/4.0
       UN(I,J,K)=0.0
       VN(I,J,K)=0.1
       WN(I,J,K)=0.0
       P(I,J,K)=0.0
 560  CONTINUE
      DTMAX1=0.33*DELZ/VINLET
      DTMAX2=0.5*DELR**2*DELZ**2*DELTH**2/((DELR**2+DELZ**2+DELTH**2)
     1 *NU)
      ALFAMI=1.5*VINLET*DELT/DELZ
      BETA=OMG/(2*DELT*(RDR**2+RDZ**2+RDTH**2))
      WRITE(6,92) DELT,DTMAX1,DTMAX2,ALPHA,ALFAMI
C
C       DELT REDUCTION FOR STABILITY
C
        IF(DELT.GT.DTMAX1) DELT=DTMAX1
92      FORMAT(//,2X,'DELT =',E12.4,2X,'DTMAX1 =',E12.4,2X,'DTMAX2 ='
     1 ,E12.4,2X,'ALPHA =',E12.4,2X,'ALFAMI =',E12.4,//)
C
C
      DO 540 J=1,JMAX
      DO 540 K=1,KMAX
      DO 540 I=1,IMAX
      RU(I,J,K)=(I-1.)*DELR+RIN
      RV(I,J,K)=(I-1.5)*DELR +RIN
        IF(RIN.LE.1.E-15) RU(1,J,K)=1.E-3
540   CONTINUE
C
C------------- INITIAL TURB ENERGY & DISSIAPATION ----------------
```

```
C
      TIN=0.01*(VINLET)**2
      EIN=(TIN**1.5)/0.01
      IF(EIN.LT.1.E-3) EIN=1.E-3
      TINMAX=10.0*TIN
      EINMAX=10.0*EIN
      DO 557 I=1,IMAX
      DO 557 J=1,JMAX
      DO 557 K=1,KMAX
       TK(I,J,K)=TIN
       E(I,J,K)=EIN
      MU(I,J,K)=CMU*DEN*(TIN**2)/EIN + VISCOS
557   CONTINUE
C     ---------------------------------
      ASSIGN 5000 TO KRET
      GO TO 2000
 1000 CONTINUE
      ITER=0
      FLG=1.0
      ASSIGN 3000 TO KRET
C     ------------------------------
C     APPLY MOMENTA EQUATIONS FOR TIME ADVANCE
C     U,V,W
      DO 1100  I=2,IM1
      DO 1100  J=2,JM1
      DO 1100  K=2,KM1
C
      TMRE=(MU(I+1,J,K)+MU(I,J,K))/2.
      TMRW=(MU(I-1,J,K)+MU(I,J,K))/2.
      TMTN=(MU(I,J,K+1)+MU(I,J,K))/2.
      TMTS=(MU(I,J,K-1)+MU(I,J,K))/2.
      TMZO=(MU(I,J+1,K)+MU(I,J,K))/2.
      TMZI=(MU(I,J-1,K)+MU(I,J,K))/2.
      TMUN=(MU(I,J,K)+MU(I+1,J,K)+MU(I+1,J,K+1)+MU(I,J,K+1))/4.
      TMUS=(MU(I,J,K)+MU(I+1,J,K)+MU(I+1,J,K-1)+MU(I,J,K-1))/4.
      TMUO=(MU(I,J,K)+MU(I+1,J,K)+MU(I+1,J+1,K)+MU(I,J+1,K))/4.
      TMUI=(MU(I,J,K)+MU(I+1,J,K)+MU(I+1,J-1,K)+MU(I,J-1,K))/4.
      TMWE=(MU(I,J,K)+MU(I+1,J,K)+MU(I+1,J,K+1)+MU(I,J,K+1))/4.
      TMWW=(MU(I,J,K)+MU(I-1,J,K)+MU(I-1,J,K+1)+MU(I,J,K+1))/4.
      TMWO=(MU(I,J,K)+MU(I,J,K+1)+MU(I,J+1,K+1)+MU(I,J+1,K))/4.
      TMWI=(MU(I,J,K)+MU(I,J,K+1)+MU(I,J-1,K+1)+MU(I,J-1,K))/4.
      TMVE=(MU(I,J,K)+MU(I+1,J,K)+MU(I+1,J+1,K)+MU(I,J+1,K))/4.
      TMVW=(MU(I,J,K)+MU(I-1,J,K)+MU(I-1,J+1,K)+MU(I,J+1,K))/4.
      TMVN=(MU(I,J,K)+MU(I,J,K+1)+MU(I,J+1,K+1)+MU(I,J+1,K))/4.
      TMVS=(MU(I,J,K)+MU(I,J,K-1)+MU(I,J+1,K-1)+MU(I,J+1,K))/4.
C
      TAURRE=2.*MU(I+1,J,K)*(UN(I+1,J,K)-UN(I,J,K))
      TAURRW=2.*MU(I,J,K)*(UN(I,J,K)-UN(I-1,J,K))
      DTARRU=RDR**2*(RV(I+1,J,K)*TAURRE-RV(I,J,K)*TAURRW)/RU(I,J,K)
      TAURTN=TMUN*(RU(I,J,K)*RDR*(WN(I+1,J,K)/RV(I+1,J,K)-WN(I,J,K)/
     1RV(I,J,K))+(RDTH/RU(I,J,K))*(UN(I,J,K+1)-UN(I,J,K)))
      TAURTS=TMUS*(RU(I,J,K)*RDR*(WN(I+1,J,K-1)/RV(I+1,J,K-1)-WN(I,J,K-1
     1)/RV(I,J,K-1))+(RDTH/RU(I,J,K))*(UN(I,J,K)-UN(I,J,K-1)))
      DTARTU=RDTH*(TAURTN-TAURTS)/RU(I,J,K)
```

```
      TAUTTU=2.*TMRE*(.5*RDTH*(WN(I,J,K)+WN(I+1,J,K)-WN(I,J,K-1)-
     1WN(I+1,J,K-1))+UN(I,J,K))/RU(I,J,K)**2
      TAURZO=TMUO*(RDR*(VN(I+1,J,K)-VN(I,J,K))+RDZ*(UN(I,J+1,K) -
     1 UN(I,J,K)))
      TAURZI=TMUI*(RDR*(VN(I+1,J-1,K)-VN(I,J-1,K))+RDZ*(UN(I,J,K)-
     1 UN(I,J-1,K)))
      IF(I.LE.IN2.AND.J.EQ.2) GO TO 334
        IF(J.EQ.2) TAURZO=TWWRZO(I,K)
334   DTARZU=RDZ*(TAURZI-TAURZO)
      TAURTE=TMWE*(RU(I,J,K)*RDR*(WN(I+1,J,K)/RV(I+1,J,K)-WN(I,J,K)/
     1RV(I,J,K))+RDTH*(UN(I,J,K+1)-UN(I,J,K))/RU(I,J,K))
      TAURTW=TMWW*(RU(I-1,J,K)*RDR*(WN(I,J,K)/RV(I,J,K)-WN(I-1,J,K)/
     1RV(I-1,J,K))+RDTH*(UN(I-1,J,K+1)-UN(I-1,J,K))/RU(I-1,J,K))
      IF(I.EQ.2.AND.RIN.GT.0.0) TAURTE=TSWRTE(J,K)
      IF(I.EQ.IM1) TAURTE=TNWRTE(J,K)
1125  DTARTW=RDR*(RU(I-1,J,K)**2*TAURTW-RU(I,J,K)**2*TAURTE)/
     1RV(I,J,K)**2
      TAUTTN=2.*MU(I,J,K+1)*(RDTH*(WN(I,J,K+1)-WN(I,J,K))+(UN(I,J,K+1)
     1 +UN(I-1,J,K+1))/2.)
      TAUTTS=2.*MU(I,J,K)*(RDTH*(WN(I,J,K)-WN(I,J,K-1))+(UN(I,J,K)
     1 +UN(I-1,J,K))/2.)
      DTATTW=RDTH*(TAUTTN-TAUTTS)/RV(I,J,K)**2
      TAUTZO=TMWO*(RDZ*(WN(I,J+1,K)-WN(I,J,K))+RDTH*(VN(I,J,K+1)-
     1 VN(I,J,K))/RV(I,J,K))
      TAUTZI=TMWI*(RDZ*(WN(I,J,K)-WN(I,J-1,K))+RDTH*(VN(I,J-1,K+1)-
     1 VN(I,J-1,K))/RV(I,J,K))
      IF(I.LE.IN2.AND.J.EQ.2) GO TO 335
      IF(J.EQ.2) TAUTZO=TWWTZO(I,K)
335   DTATZW=RDZ*(TAUTZI-TAUTZO)
      TAURZE=TMVE*(RDR*(VN(I+1,J,K)-VN(I,J,K))+RDZ*(UN(I,J+1,K)-
     1 UN(I,J,K)))
      TAURZW=TMVW*(RDR*(VN(I,J,K)-VN(I-1,J,K))+RDZ*(UN(I-1,J+1,K)
     1 -UN(I-1,J,K)))
        IF(I.EQ.2.AND.RIN.GT.0.0) TAURZE=TSWRZE(J,K)
        IF(I.EQ.IM1) TAURZE=TNWRZE(J,K)
      DTARZV=RDR*(RU(I-1,J,K)*TAURZW-RU(I,J,K)*TAURZE)/RV(I,J,K)
      TAUTZN=TMVN*(RDZ*(WN(I,J+1,K)-WN(I,J,K))+RDTH*(VN(I,J,K+1)-
     1 VN(I,J,K))/RV(I,J,K))
      TAUTZS=TMVS*(RDZ*(WN(I,J+1,K-1)-WN(I,J,K-1))+RDTH*(VN(I,J,K) -
     1 VN(I,J,K-1))/RV(I,J,K))
      DTATZV=RDTH*(TAUTZN-TAUTZS)/RV(I,J,K)
C
      IF(RIN.LE.1.E-15.AND.I.EQ.2) DTATZV=0.0
C
      TAUZZO=2.*MU(I,J+1,K)*RDZ*(VN(I,J+1,K)-VN(I,J,K))
      TAUZZI=2.*MU(I,J,K)*RDZ*(VN(I,J,K)-VN(I,J-1,K))
      DTAZZV=RDZ*(TAUZZO-TAUZZI)
C
C     VELOCITY IN THE RADIAL DIRECTION  R
C
      FUR=(RV(I+1,J,K)*(UN(I,J,K)+ UN(I+1,J,K))**2+ALPHA*RV(I+1,J,K)
     1*ABS(UN(I,J,K)+UN(I+1,J,K))*(UN(I,J,K)-UN(I+1,J,K))-RV(I,J,K)
     2*(UN(I-1,J,K)+UN(I,J,K))**2 -ALPHA*RV(I,J,K)*ABS(UN(I-1,J,K)+
     3 UN(I,J,K))*(UN(I-1,J,K)-UN(I,J,K)))/(4.*DELR*RU(I,J,K))
```

```
C
      FUZ =((VN(I,J,K)+VN(I+1,J,K))*(UN(I,J,K)+UN(I,J+1,K))+ALPHA*ABS(VN
     1(I,J,K)+VN(I+1,J,K))*(UN(I,J,K)-UN(I,J+1,K))-(VN(I,J-1,K)+VN(I+1,
     2J-1,K))*(UN(I,J-1,K)+UN(I,J,K))-ALPHA*ABS(VN(I,J-1,K)+VN(I+1,J-1,K
     3))*(UN(I,J-1,K)-UN(I,J,K)))/(4*DELZ)
C
      FUT=((WN(I,J,K)+WN(I+1,J,K))*(UN(I,J,K)+UN(I,J,K+1))+ALPHA*ABS(WN(
     1I,J,K)+WN(I+1,J,K))*(UN(I,J,K)-UN(I,J,K+1))-(WN(I,J,K-1)+WN(I+1,J,
     2K-1))*(UN(I,J,K-1)+UN(I,J,K))-ALPHA*ABS(WN(I,J,K-1)+WN(I+1,J,K-1)
     3 )*(UN(I,J,K-1)-UN(I,J,K)))/(4.*DELTH*RU(I,J,K))
C
      FUC=((WN(I,J,K)+WN(I+1,J,K))**2+
     1    (WN(I,J,K-1)+WN(I+1,J,K-1))**2+ALPHA*ABS(WN(I,J,K)+WN(I+1,J,K))
     2*(WN(I,J,K)-WN(I+1,J,K))+ALPHA*ABS(WN(I,J,K-1)+WN(I+1,J,K-1))*(WN(
     3I,J,K-1)-WN(I+1,J,K-1)))/(8* RU(I,J,K))
C
      VISR=(DTARRU+ DTARTU- TAUTTU+ DTARZU )/DEN
C
C     VELOCITY IN THE AXIAL DIRECTION ZETA
C
      FVR=(RU(I,J,K)*(UN(I,J,K)+UN(I,J+1,K))*(VN(I,J,K)+VN(I+1,J,K))+
     1 ALPHA*RU(I,J,K)*ABS(UN(I,J,K)+UN(I,J+1,K))*(VN(I,J,K)-VN(I+1,J
     2 ,K))-RU(I-1,J,K)*(UN(I-1,J,K)+UN(I-1,J+1,K))*(VN(I-1,J,K)+VN(I,
     3 J,K))-ALPHA*RU(I-1,J,K)*ABS(UN(I-1,J,K)+UN(I-1,J+1,K))*(VN(I-1,
     4 J,K)-VN(I,J,K)))/(4.*DELR*RV(I,J,K))
C
      FVZ=((VN(I,J,K)+VN(I,J+1,K))**2+ALPHA*ABS(VN(I,J,K)+VN(I,J+1,K))*(
     1VN(I,J,K)-VN(I,J+1,K))-(VN(I,J-1,K)+VN(I,J,K))**2-ALPHA*ABS(VN(I,J
     2-1,K)+VN(I,J,K))*(VN(I,J-1,K)-VN(I,J,K)))/(4*DELZ)
C
      FVT=((WN(I,J,K)+WN(I,J+1,K))*(VN(I,J,K)+VN(I,J,K+1))+ALPHA*ABS(WN(
     1I,J,K)+WN(I,J+1,K))*(VN(I,J,K)-VN(I,J,K+1))-(WN(I,J,K-1)+WN(I,J+1,
     2K-1))*(VN(I,J,K-1)+VN(I,J,K))-ALPHA*ABS(WN(I,J,K-1)+WN(I,J+1,K-1))
     3*(VN(I,J,K-1)-VN(I,J,K)))/(4*DELTH* RV(I,J,K))
C
C
      VISZ=(DTARZV+ DTATZV+ DTAZZV )/DEN
C
C     VELOCITY IN THE ANGULAR DIRECTION THETA
C
      FWR=(RU(I,J,K)*(UN(I,J,K)+UN(I,J,K+1))*(WN(I,J,K)+WN(I+1,J,K))+
     1 ALPHA*RU(I,J,K)*ABS(UN(I,J,K)+UN(I,J,K+1))*(WN(I,J,K)-WN(I+1,J,K)
     2 )- RU(I-1,J,K)*(UN(I-1,J,K)+UN(I-1,J,K+1))*(WN(I-1,J,K)+WN(I,J,K)
     3 )- ALPHA*RU(I-1,J,K)*ABS(UN(I-1,J,K)+UN(I-1,J,K+1))*(WN(I-1,J,K)
     4 - WN(I,J,K)))/(4.*DELR*RV(I,J,K))
C
      FWZ=((VN(I,J,K)+VN(I,J,K+1))*(WN(I,J,K)+WN(I,J+1,K))+ALPHA*ABS(VN(
     1I,J,K)+VN(I,J,K+1))*(WN(I,J,K)-WN(I,J+1,K))-(VN(I,J-1,K)+VN(I,J-1,
     2K+1))*(WN(I,J-1,K)+WN(I,J,K))-ALPHA*ABS(VN(I,J-1,K)+VN(I,J-1,K+1))
     3*(WN(I,J-1,K)-WN(I,J,K)))/(4*DELZ)
C
      FWT=((WN(I,J,K)+WN(I,J,K+1))**2+ALPHA*ABS(WN(I,J,K)+WN(I,J,K+1))*(
     1WN(I,J,K)-WN(I,J,K+1))-(WN(I,J,K-1)+WN(I,J,K))**2-ALPHA*ABS(WN(I,J
     2,K-1)+WN(I,J,K))*(WN(I,J,K-1)-WN(I,J,K)))/(4*DELTH* RV(I,J,K))
```

```
C
      FWC=((UN(I,J,K)+UN(I,J,K+1))*(WN(I,J,K)+WN(I+1,J,K))+(UN(I-1,J,K)
     1+UN(I-1,J,K+1))*(WN(I-1,J,K)+WN(I,J,K))+ALPHA*ABS(UN(I,J,K)+UN(I,J
     2,K+1))*(WN(I,J,K)-WN(I+1,J,K))+ALPHA*ABS(UN(I-1,J,K)+UN(I-1,J,K+1)
     3)*(WN(I-1,J,K)-WN(I,J,K)))/(8* RV(I,J,K))
C
      VIST=(DTARTW+ DTATTW+ DTATZW )/DEN
C
      IF(RIN.LE.1.E-15.AND.I.EQ.2) GO TO 1135
      IF(I.LE.IN2.AND.J.EQ.2) GO TO 1135
      IF(I.EQ.2.OR.I.EQ.IM1.OR.J.EQ.2) GO TO 1150
C
1135  DWR=RDR*((WN(I,J,K)+WN(I,J,K-1)+WN(I+1,J,K)+WN(I+1,J,K-1))/(4.0
     1 ) -(WN(I,J,K)+WN(I,J,K-1)+WN(I-1,J,K-1)+WN(I-1,J,K))/
     2 (4.) ) - (WN(I,J,K)/RV(I,J,K))
        IF(RIN.LE.1.E-15.AND.I.LT.4) DWR=0.0
C
      DUT=RDTH*((UN(I,J,K)+UN(I,J,K+1)+UN(I-1,J,K+1)+UN(I-1,J,K))/4.
     1 -(UN(I,J,K)+UN(I-1,J,K)+UN(I-1,J,K-1)+UN(I,J,K-1))/4.)/RV(I,J,K)
      DVT=RDTH*((VN(I,J,K)+VN(I,J,K+1)+VN(I,J-1,K+1)+VN(I,J-1,K))/4.
     1 -(VN(I,J,K)+VN(I,J-1,K)+VN(I,J,K-1)+VN(I,J-1,K-1))/4.)/RV(I,J,K)
      DWZ=RDZ*((WN(I,J,K)+WN(I,J+1,K)+WN(I,J+1,K-1)+WN(I,J,K-1))/4.
     1 -(WN(I,J,K)+WN(I,J,K-1)+WN(I,J-1,K)+WN(I,J-1,K-1))/4.)
      DUZ=RDZ*((UN(I,J,K)+UN(I,J+1,K)+UN(I-1,J,K)+UN(I-1,J+1,K))/4.
     1 -(UN(I,J,K)+UN(I-1,J,K)+UN(I,J-1,K)+UN(I-1,J-1,K))/4.)
      DVR=RDR*((VN(I,J,K)+VN(I+1,J,K)+VN(I,J-1,K)+VN(I+1,J-1,K))/4.
     1 -(VN(I,J,K)+VN(I,J-1,K)+VN(I-1,J,K)+VN(I-1,J-1,K))/4.)
C
      GP=MU(I,J,K)*(2.*((RDR*(UN(I,J,K)-UN(I-1,J,K)))**2 +
     1 (UN(I,J,K)/RU(I,J,K) +(RDTH/RV(I,J,K))*(WN(I,J,K)-WN(I,J,K-1)))
     2 **2 + (RDZ*(VN(I,J,K)-VN(I,J-1,K)))**2) )
C
        GG(I,J,K)= GP + (DWR+DUT)**2 + (DVT+DWZ)**2 + (DUZ+DVR)**2
C
        IF(CYCLE.LT.400) GO TO 1150
        IF(I.NE.2) GO TO 1150
        WRITE(6,95) I,J,K,DWR,DUT,DVT,DWZ
        WRITE(6,97) DUZ,DVR,GP,GG(I,J,K)
C
95      FORMAT(1X,3(I2,1X),'DWR,DUT,DVT,DWZ',3X,4(E12.5,1X))
97      FORMAT(2X,'DUZ,DVR,GP,GG ',3X,4(E12.5,1X))
C
C
1150  CONTINUE
C
C
C     TURBULENT ENERGY
C
      FKR=(1./RV(I,J,K))*(RU(I,J,K)*UN(I,J,K)*(TKN(I,J,K)+TKN(I+1,J,K))
     1 + ALPHA*ABS(UN(I,J,K))*RU(I,J,K)*(TKN(I,J,K)-TKN(I+1,J,K)) -
     2 UN(I-1,J,K)*RU(I-1,J,K)*(TKN(I-1,J,K)+TKN(I,J,K)) - ALPHA*
     3 ABS(UN(I-1,J,K))*RU(I-1,J,K)*(TKN(I-1,J,K)-TKN(I,J,K)))*RDR/2.
C
      FKZ=RDZ*(VN(I,J,K)*(TKN(I,J,K)+TKN(I,J+1,K))+ALPHA*ABS(VN(I,J,K
```

```
     1 ))*(TKN(I,J,K)-TKN(I,J+1,K)) -VN(I,J-1,K)*(TKN(I,J-1,K)+TKN(I,J,K
     2 )) -ALPHA*ABS(VN(I,J-1,K))*(TKN(I,J-1,K)-TKN(I,J,K)))/2.
C
      FKT=RDTH*(WN(I,J,K)*(TKN(I,J,K)+TKN(I,J,K+1)) +ALPHA*ABS(WN(I,J,K)
     1 )*(TKN(I,J,K)-TKN(I,J,K+1)) -WN(I,J,K-1)*(TKN(I,J,K-1)+TKN(I,J,K)
     2 ) -ALPHA*ABS(WN(I,J,K-1))*(TKN(I,J,K-1)-TKN(I,J,K)))/2.
C
      VISK=((RDR**2/RV(I,J,K))*(RU(I,J,K)*TMRE*(TKN(I+1,J,K)-TKN(I,J,K))
     1 -RU(I-1,J,K)*TMRW*(TKN(I,J,K)-TKN(I-1,J,K))) + (RDTH**2/RV(I,J,K
     2 )**2) *(TMTN*(TKN(I,J,K+1)-TKN(I,J,K))-TMTS*(TKN(I,J,K)-TKN(I,J,K
     3 -1))) +RDZ**2 *(TMZO*(TKN(I,J+1,K)-TKN(I,J,K))-TMZI*(TKN(I,J,K)-
     4 TKN(I,J-1,K))))/(DEN*SK)
C
      SORK=GG(I,J,K)/DEN
      SORK2=1.0+DELT* EN(I,J,K) /TKN(I,J,K)
      VISK1=VISK+SORK
C
C
C     VELOCITIES UPDATE
C     RADIAL VELOCITY
C
      U(I,J,K)=UN(I,J,K)+DELT*((P(I,J,K)-P(I+1,J,K))*RDR+GR-FUR-FUZ-FUT+
     1FUC+VISR)
C     AXIAL VELOCITY
C
      V(I,J,K)=VN(I,J,K)+DELT*((P(I,J,K)-P(I,J+1,K))*RDZ+GZ-FVR-FVZ-FVT
     1+VISZ)
C     TANGENTIAL VELOCITY
C
      W(I,J,K)=WN(I,J,K)+DELT*((P(I,J,K)-P(I,J,K+1))*RDTH/(RV(I,J,K))
     1  +GTH-FWR-FWZ-FWT-FWC+VIST)
C
       IF(W(I,J,K).LT.0.0) W(I,J,K)=0.0
C
       IF(RIN.LE.1.E-15.AND.I.EQ.2) W(I,J,K)=0.3333*W(I+1,J,K)
C
C     TURBULENCE ENERGY
C
       IF(CYCLE.LE.40) GO TO 1800
C
      TK(I,J,K)=(TKN(I,J,K)+DELT*(-FKR-FKZ-FKT+VISK1))/SORK2
      IF(TK(I,J,K).LT.1.E-3) TK(I,J,K)=1.E-3
      IF(EN(I,J,K).LT.1.E-3) EN(I,J,K)=1.E-3
      IF(TK(I,J,K).GT.TINMAX) TK(I,J,K)=TINMAX
C
1800    CONTINUE
C
      VISOLD=MUN(I,J,K)
      MU(I,J,K)=(CMU*DEN*(TKN(I,J,K))**2)/EN(I,J,K) + VISCOS
      MU(I,J,K)=URFVIS*MU(I,J,K)+(1.-URFVIS)*VISOLD
C
      IF(CYCLE.LT.400) GO TO 1100
       IF(I.NE.2) GO TO 1100
C
```

```
      WRITE(6,40) I,J,K,FKR,FKZ,FKT,VISK
      WRITE(6,41) SORK,SORK2,VISK1,GG(I,J,K)
C
       WRITE(6,96) I,J,K,FUR,FUT,FUZ,FUC,VISR
       WRITE(6,93) FVR,FVT,FVZ,VISZ
       WRITE(6,94) FWR,FWT,FWZ,FWC,VIST
C
96     FORMAT(1X,3(I2,1X),'FUR,FUT,FUZ,FUC,VISR',3X,5(E12.5,1X))
93     FORMAT(1X,'FVR,FVT,FVZ,VISZ ',2X,4(E12.5,1X))
94     FORMAT(1X,'FWR,FWT,FWZ,FWC,VIST',3X,5(E12.5,1X))
C
 40   FORMAT(1X,3(I2,1X),'FKR,Z,T,AND VISK',2X,4(E12.5,1X))
 41   FORMAT(1X,'SORK,SORK2,VISK1,GG',2X,4(E12.5,1X))
C
C
 1100 CONTINUE
C
 2000 CONTINUE
C
C
C     GENERAL BOUNDARY CONDITIONS
C
C     PERIODEC BOUNDARY CONDITIONS
C
C     BOUNDARY CONDITIONS ON THE RIGHT SIDE
      DO 2200 J=1,JMAX
      DO 2200 I=1,IMAX
      U(I,J,1)=U(I,J,KM1)
      V(I,J,1)=V(I,J,KM1)
      W(I,J,1)=W(I,J,KM1)
      P(I,J,1)=P(I,J,KM1)
      TK(I,J,1)=TK(I,J,KM1)
      E(I,J,1)=E(I,J,KM1)
      MU(I,J,1)=MU(I,J,KM1)
C     BOUNDARY CONDITIONS ON THE LEFT SIDE
      U(I,J,2)=U(I,J,KM1)
      V(I,J,2)=V(I,J,KM1)
      W(I,J,2)=W(I,J,KM1)
      P(I,J,2)=P(I,J,KM1)
      TK(I,J,2)=TK(I,J,KM1)
      E(I,J,2)=E(I,J,KM1)
      MU(I,J,2)=MU(I,J,KM1)
C
C
      U(I,J,KMAX)=U(I,J,KM1)
      V(I,J,KMAX)=V(I,J,KM1)
      W(I,J,KMAX)=W(I,J,KM1)
      P(I,J,KMAX)=P(I,J,KM1)
      TK(I,J,KMAX)=TK(I,J,KM1)
      E(I,J,KMAX)=E(I,J,KM1)
      MU(I,J,KMAX)=MU(I,J,KM1)
 2200 CONTINUE
C
C     BOUNDARY CONDITIONS ON THE WEST WALL
```

```
C     NO SLIP BCS
C
      DO 2500 I=1,IMAX
      DO 2500 K=1,KMAX
      U(I,1,K)=-U(I,2,K)
      V(I,1,K)=0.0
      W(I,1,K)=-W(I,2,K)
      TK(I,1,K)=TK(I,2,K)
      IF(RIN.LE.1.E-15.AND.I.LE.IN2) GO TO 2501
      E(I,2,K)=(((CMU*CD)**0.75)*TK(I,2,K)**1.5)/(CD*PK*DELZ*0.5)
2501  E(I,1,K)=E(I,2,K)
      MU(I,1,K)=MU(I,2,K)
C
C     BOUNDARY CONDITIONS ON THE EAST WALL
C      COMBUSTOR OUTLET
C
      U(I,JMAX,K)= U(I,JM1,K)
C     V(I,JMAX,K) =0.0
C     V(I,JM1,K)=0.0
      W(I,JMAX,K)=W(I,JM1,K)
      TK(I,JMAX,K)=TK(I,JM1,K)
      E(I,JMAX,K)=E(I,JM1,K)
      MU(I,JMAX,K)=MU(I,JM1,K)
 2500 CONTINUE
C
C     BOUNDARY CONDITIONS ON THE INNER CYLINDER
C     NO SLIP BCS
C
      DO 2020  J=1,JMAX
      DO 2020  K=1,KMAX
      IF(RIN.LE.1.E-15) GO TO 2021
      U(1,J,K)=0.0
      V(1,J,K)=-V(2,J,K)
      W(1,J,K)=-W(2,J,K)
      E(2,J,K)=(((CMU*CD)**0.75)*TK(2,J,K)**1.5)/(CD*PK*DELR*0.5)
      GO TO 2022
C     SYMMETRY AXIS WHEN  RIN=0.0
2021  V(1,J,K)=V(2,J,K)
      W(1,J,K)=-W(2,J,K)
2022  U(1,J,K)=0.0
      TK(1,J,K)=TK(2,J,K)
      E(1,J,K)=E(2,J,K)
      MU(1,J,K)=MU(2,J,K)
C
C     BOUNDARY CONDITIONS ON THE OUTER CYLINDER
C     NO SLIP BCS
C
      U(IMAX,J,K)=0.0
      U(IM1,J,K)=0.0
      V(IMAX,J,K)=-V(IM1,J,K)
      W(IMAX,J,K)=-W(IM1,J,K)
      TK(IMAX,J,K)=TK(IM1,J,K)
      E(IM1,J,K)=(((CMU*CD)**0.75)*TK(IM1,J,K)**1.5)/(CD*PK*DELR*0.5)
      E(IMAX,J,K)=E(IM1,J,K)
```

```
      MU(IMAX,J,K)=MU(IM1,J,K)
 2020 CONTINUE
C
C     SPECIAL BOUNDARY CONDITIONS
C
          IF(WIN.LE.0.0) GO TO 2111
C
C
         DO 2120 K=1,KMAX
C
C
         V(2,1,K)=0.0
         W(2,1,K)=0.0
         V(3,1,K)=0.4*VINLET
         W(3,1,K)=0.4*WIN
         V(4,1,K)=0.8*VINLET
         W(4,1,K)=0.8*WIN
         V(5,1,K)=1.2*VINLET
         W(5,1,K)=1.2*WIN
         V(6,1,K)=1.6*VINLET
         W(6,1,K)=1.6*WIN
C
 2120      CONTINUE
C
2111       CONTINUE
C
C     INLET  FLOW
C
C
      DO 2220  I=IN1 ,IN2
      DO 2220  K=1,KMAX
C
        IF(WIN.GT.0.0) GO TO 2221
C
      V(I,1,K)=VINLET
        W(I,1,K)=WIN
2221   U(I,1,K)=0.0
      TK(I,1,K)=0.03*(V(I,1,K)**2)
      E(I,1,K)=(TK(I,1,K)**1.5)/(0.01*DIAM)
 2220 CONTINUE
C
C------------  INLET BCS FOR THE DILUTION JET -------------------------
C
        U(IM1,JDIL,KDIL)=-UDIL
        U(IMAX,JDIL,KDIL)=-UDIL
C
C
        FLUXDJ=UDIL*DELZ*DELTH*RU(IM1,JDIL,KDIL)
C
C
C     OUTLET FLOW
      IF(ITER.GT.0) GO TO 2813
C     USE OF FLUXRAT TO AMEND V AT OUTLET AT ITER=0 ONLY
C     COMPUTATION OF IN AND OUT FLUXES
```

```
      FLUXIN=0.0
      DO 6112 I=IN1,IN2
      DO 6112 K=2,KM1
C        K=KIN
      FLUXIN=((RU(I,1,K)**2-RU(I-1,1,K)**2)*DELTH/2)*V(I,1,
     1K) +FLUXIN
 6112 CONTINUE
C
       FLUXIN=FLUXIN+FLUXDJ
       AROUT=((RIN+IM2*DELR)**2-RIN**2)*KM2*DELTH/2
       VOUT=FLUXIN/AROUT
C
      FLUXOU=0.0
      DO 6122 I=2,IM1
      DO 6122 K=2,KM1
         J=JM2
       FLUXOU=(RU(I,J,K)**2-RU(I-1,J,K)**2)*(DELTH/2)*V(I,J,K)+FLUXOU
6122  CONTINUE
      VINC=(FLUXIN-FLUXOU)/AROUT
C
      DO 2817 I=2,IM1
      DO 2817 K=2,KM1
C     V(I,JM1,K)=VOUT
      V(I,JM1,K)=V(I,JM2,K)+VINC
      V(I,JMAX,K)=V(I,JM1,K)
      U(I,JMAX,K)=U(I,JM1,K)
 2817 W(I,JMAX,K)=W(I,JM1,K)
 2813 CONTINUE
C
C
       IF(CYCLE.EQ.0) GO TO 2990
       IF(CYCLE.LE.40) GO TO 2990
       IF(ITER.GT.0)  GO TO 2990
C
C
      DO 1110 I=1,IMAX
      DO 1110 J=1,JMAX
      DO 1110 K=1,KMAX
       EN(I,J,K)=E(I,J,K)
1110  CONTINUE
C
C------------ DISSIAPATION EQUATION --------------------
C
      IS=3
      IF(RIN.LE.1.E-15) IS=2
C
C
      DO 1111 I=IS,IM2
      JS=3
      IF(RIN.LE.1.E-15.AND.I.LE.IN2) JS=2
C
      DO 1111 J=JS,JM1
      DO 1111 K=2,KM1
C
```

```
      TMRE=(MU(I+1,J,K)+MU(I,J,K))/2.
      TMRW=(MU(I-1,J,K)+MU(I,J,K))/2.
      TMTN=(MU(I,J,K+1)+MU(I,J,K))/2.
      TMTS=(MU(I,J,K-1)+MU(I,J,K))/2.
      TMZO=(MU(I,J+1,K)+MU(I,J,K))/2.
      TMZI=(MU(I,J-1,K)+MU(I,J,K))/2.
C
      FER=RDR*(RU(I,J,K)*UN(I,J,K)*(EN(I,J,K)+EN(I+1,J,K)) +ALPHA
     1 *RU(I,J,K)*ABS(UN(I,J,K))*(EN(I,J,K)-EN(I+1,J,K)) -UN(I-1,J,K)*
     2 RU(I-1,J,K)*(EN(I-1,J,K)+EN(I,J,K)) -ALPHA*RU(I-1,J,K)*ABS
     3 (UN(I-1,J,K))*(EN(I-1,J,K)-EN(I,J,K)))/(2.*RV(I,J,K))
C
      FEZ=RDZ*(VN(I,J,K)*(EN(I,J,K)+EN(I,J+1,K))+ALPHA*ABS(VN(I,J,K))
     1 *(EN(I,J,K)-EN(I,J+1,K))- VN(I,J-1,K)*(EN(I,J-1,K)+EN(I,J,K))-
     2 ALPHA*ABS(VN(I,J-1,K))*(EN(I,J-1,K)-EN(I,J,K)))/2.
C
      FET=RDTH*(WN(I,J,K)*(EN(I,J,K)+EN(I,J,K+1)) +ALPHA*ABS(WN(I,J,K))
     1 *(EN(I,J,K)-EN(I,J,K+1))- WN(I,J,K-1)*(EN(I,J,K-1)+EN(I,J,K))-
     2 ALPHA*ABS(WN(I,J,K-1))*(EN(I,J,K-1)-EN(I,J,K)))/2.
C
      VISE=((RDR**2/RV(I,J,K))*(RU(I,J,K)*TMRE*(EN(I+1,J,K)-EN(I,J,K))
     1 -RU(I-1,J,K)*TMRW*(EN(I,J,K)-EN(I-1,J,K))) +(RDTH**2/RV(I,J,K)
     2 **2)*(TMTN*(EN(I,J,K+1)-EN(I,J,K))-TMTS*(EN(I,J,K)-EN(I,J,K-1))
     3 ) +RDZ**2* (TMZO*(EN(I,J+1,K)-EN(I,J,K))-TMZI*(EN(I,J,K)-EN(I,J
     4 -1,K))))/(DEN*SE)
C
      SORE=GG(I,J,K)*C1 *EN(I,J,K) /(TKN(I,J,K)*DEN)
      SORE2=1.0+ DELT*C2 *EN(I,J,K)/(TKN(I,J,K))
      VISE1=VISE +SORE
C
C      TURBULENCE DISSIAPATION
C
      E(I,J,K)=(EN(I,J,K)+DELT *(-FER-FEZ-FET+VISE1))/SORE2
      IF(E(I,J,K).LT.1.E-3) E(I,J,K)=1.E-3
      IF(E(I,J,K).GT.EINMAX) E(I,J,K)=EINMAX
C
       IF(CYCLE.LT.400) GO TO 1111
       IF(I.NE.2) GO TO 1111
C
      WRITE(6,39) I,J,K,FER,FEZ,FET,VISE
      WRITE(6,38) SORE,SORE2,VISE1,GG(I,J,K)
 39   FORMAT(1X,3(I2,1X),'FER,Z,T,AND VISE',2X,4(E12.5,1X))
 38   FORMAT(1X,'SORE,SORE2,VISE1,GG',2X,4(E12.5,1X))
C
C
1111  CONTINUE
C
C
2990   CONTINUE
C
C
C     ------------------------------------------------------------------
C     PRESSURE ITERATION AND P U V UPDATE
C
```

```
       GO  TO  KRET , (3000,5000)
 3000 CONTINUE
      IF(FLG.EQ.0)GO TO 4000
      ITER=ITER+1
      STORIT=ITER
       IF(ITER.LT.ITMAX) GO TO 3050
      IF(CYCLE.LT.300)GO TO 4000
C     TERMINATION CONDITION
      T=1E+10
      GO TO 5000
 3050 FLG=0.0
C     CHECK IF CONVERGENCE HAS BEEN REACHED
      SUMD=0.0
      DO 3500 I=2,IM1
      DO 3500 J=2,JM1
      DO 3500 K=2,KM1
C
       D=(1./RV(I,J,K))*(RU(I,J,K)*U(I,J,K)-RU(I-1,J,K)*U(I-1,J,K))*
     1 RDR + (1./RV(I,J,K))*(W(I,J,K)-W(I,J,K-1))*RDTH + RDZ *
     2 (V(I,J,K)-V(I,J-1,K))
C
      IF(ABS(D/DZRO).GE.EPSI) FLG=1.0
      DELP=-(D*OMG)/(2*DELT*(1/(DELR**2)+1/(DELZ**2)+1/(((      (I-1.5)*D
     1ELR+RIN)*DELTH)**2)))
      P(I,J,K)=P(I,J,K)+DELP
      U(I,J,K)=U(I,J,K)+DELT*DELP*RDR
      V(I,J,K)=V(I,J,K)+DELT*DELP*RDZ
      W(I,J,K)=W(I,J,K)+DELT*DELP*RDTH/  RV(I,J,K)
       IF(W(I,J,K).LT.0.0) W(I,J,K)=0.0
      U(I-1,J,K)=U(I-1,J,K)-DELT*DELP*RDR
       V(I,J-1,K)=V(I,J-1,K)-DELT*DELP*RDZ
       W(I,J,K-1)=W(I,J,K-1)-DELT*DELP*RDTH/RV(I,J,K)
       IF(W(I,J,K-1).LT.0.0) W(I,J,K-1)=0.0
      SUMD = SUMD+ABS(D)
 3500 CONTINUE
C
C     CHECKPRINTS DURING PRESSURE CYCLE
      IWRITE=0
C     IF(ITER.LE.2)IWRITE=1
      IF(CYCLE.GT.2.AND.CYCLE.LT.CYCMAX)IWRITE=0
      IF(IWRITE.EQ.1) GO TO 5152
C     RETURN FROM PRINTING SECTION
 3501 CONTINUE
      IWRITE=0
C
      GO TO 2000
 4000 CONTINUE
 5000 CONTINUE
C
C------------ SHEAR STRESS ON THE BOTTOM WALL
C
      IF(RIN.LE.1.E-15) GO TO 5105
C
C
```

```
      DO 5100 J=2,JM1
      DO 5100 K=2,KM1
      I=2
      VAVG=(V(I,J,K)+V(I,J+1,K))/2.
      WAVG=(W(I,J,K)+W(I,J,K+1))/2.
      VEFF=SQRT(VAVG*VAVG + WAVG*WAVG)
      DENV=DEN
      RP=DELR/2.
      SQRTK=SQRT(TK(I,J,K))
      RPLUSS(J,K)=DENV*CMUPQ*SQRTK*RP/VISCOS
      IF(RPLUSS(J,K).LE.11.63) GO TO 5101
      TMULT=DENV*CMUPQ*SQRTK*CAPPA/ALOG(ELOG*RPLUSS(J,K))
      TAUS(J,K)=-TMULT*VEFF
      GO TO 5102
5101  TAURX=-VISCOS*VAVG/RP
      TAURW=-VISCOS*WAVG/RP
      TAUS(J,K)=SQRT(TAURX*TAURX+TAURW*TAURW)
5102  DVT=RDTH*((V(I,J,K)+V(I,J,K+1)+V(I,J-1,K+1)+V(I,J-1,K))/4.
     1 -(V(I,J,K)+V(I,J-1,K)+V(I,J,K-1)+V(I,J-1,K-1))/4.)/RV(I,J,K)
      DWZ=RDZ*((W(I,J,K)+W(I,J+1,K)+W(I,J+1,K-1)+W(I,J,K-1))/4.
     1 -(W(I,J,K)+W(I,J,K-1)+W(I,J-1,K)+W(I,J-1,K-1))/4.)
C
      GG(I,J,K)=MU(I,J,K)*(2.*((RDR*(U(I,J,K)-U(I-1,J,K)))**2 +
     1 (U(I,J,K)/RU(I,J,K) +(RDTH/RV(I,J,K))*(W(I,J,K)-W(I,J,K-1)))
     2 **2 +
     2 (RDZ*(V(I,J,K)-V(I,J-1,K)))**2) + TAUS(J,K)**2/MU(I,J,K)+ (DVT
     3 +DWZ)**2   )
C
       IF(RPLUSS(J,K).LE.11.63) GO TO 5103
C
      TSWRTE(J,K)=DENV*CMUPQ*CAPPA*SQRTK*W(I,J,K)/ALOG(ELOG*RPLUSS(J,K
     1 ))
      TSWRZE(J,K)=DENV*CMUPQ*CAPPA*SQRTK*V(I,J,K)/ALOG(ELOG*RPLUSS(J,K
     1 ))
      GO TO 5100
5103  TSWRTE(J,K)=VISCOS*W(I,J,K)/RP
      TSWRZE(J,K)=VISCOS*V(I,J,K)/RP
5100  CONTINUE
C
C
5105   CONTINUE
C
C
C------------- SHEAR STRESS ON THE TOP WALL (OUTER CYLINDER)
C
      DO 5120 J=2,JM1
      DO 5120 K=2,KM1
      I=IM1
      VAVG=(V(I,J,K)+V(I,J+1,K))/2.
      WAVG=(W(I,J,K)+W(I,J,K+1))/2.
      VEFF=SQRT(VAVG*VAVG+WAVG*WAVG)
      DENV=DEN
      RP=DELR/2.
      SQRTK=SQRT(TK(I,J,K))
```

```
      RPLUSS(J,K)=DENV*CMUPQ*SQRTK*RP/VISCOS
      IF(RPLUSS(J,K).LE.11.63) GO TO 5121
      TMULT=DENV*CMUPQ*SQRTK*CAPPA/ALOG(ELOG*RPLUSS(J,K))
      TAUN(J,K)=-TMULT*VEFF
      GO TO 5122
5121  TAUTX=-VISCOS*VAVG/RP
      TAUTW=-VISCOS*WAVG/RP
      TAUN(J,K)=SQRT(TAUTX*TAUTX+TAUXW*TAUXW)
5122   DVT=RDTH*((V(I,J,K)+V(I,J,K+1)+V(I,J-1,K+1)+V(I,J-1,K))/4.
     1 -(V(I,J,K)+V(I,J-1,K)+V(I,J,K-1)+V(I,J-1,K-1))/4.)/RV(I,J,K)
      DWZ=RDZ*((W(I,J,K)+W(I,J+1,K)+W(I,J+1,K-1)+W(I,J,K-1))/4.
     1 -(W(I,J,K)+W(I,J,K-1)+W(I,J-1,K)+W(I,J-1,K-1))/4.)
C
      GG(I,J,K)=MU(I,J,K)*(2.*((RDR*(U(I,J,K)-U(I-1,J,K)))**2 +
     1 (U(I,J,K)/RU(I,J,K) +(RDTH/RV(I,J,K))*(W(I,J,K)-W(I,J,K-1)))
     2 **2 +
     3 (RDZ*(V(I,J,K)-V(I,J-1,K)))**2) + TAUN(J,K)**2/MU(I,J,K)+ (DVT
     4 +DWZ)**2   )
C
      IF(RPLUSS(J,K).LE.11.63) GO TO 5123
C
      TNWRTE(J,K)=DENV*CMUPQ*CAPPA*SQRTK*W(I,J,K)/ALOG(ELOG*RPLUSS(J,K
     1 ))
      TNWRZE(J,K)=DENV*CMUPQ*CAPPA*SQRTK*V(I,J,K)/ALOG(ELOG*RPLUSS(J,
     1  K))
      GO TO 5120
5123  TNWRTE(J,K)=VISCOS*W(I,J,K)/RP
      TNWRZE(J,K)=VISCOS*V(I,J,K)/RP
C
5120  CONTINUE
C
C
C------------ SHEAR STRESS ON WEST WALL
C
      DO 5220 I=2,IM1
      DO 5220 K=2,KM1
      J=2
      IF(K.EQ.4.AND.I.EQ.4.OR.I.EQ.5) GO TO 5220
      UAVG=(U(I,J,K)+U(I+1,J,K))/2.
      WAVG=(W(I,J,K)+W(I,J,K+1))/2.
      VEFF=SQRT(UAVG*UAVG+WAVG*WAVG)
      DENU=DEN
      ZP=DELZ/2.
      SQRTK=SQRT(TK(I,J,K))
      ZPLUSW(I,K)=DENU*CMUPQ*SQRTK*ZP/VISCOS
      IF(ZPLUSW(I,K).LE.11.63) GO TO 5221
      TMULT=DENU*CMUPQ*SQRTK*CAPPA/ALOG(ELOG*ZPLUSW(I,K))
      TAUW(I,K)=-TMULT*VEFF
      GO TO 5222
5221  TAUXR=VISCOS*UAVG/ZP
      TAUXW=VISCOS*WAVG/ZP
      TAUW(I,K)=SQRT(TAUXR*TAUXR+TAUXW*TAUXW)
5222  DWR=RDR*((W(I,J,K)+W(I,J,K-1)+W(I+1,J,K)+W(I+1,J,K-1))/(4.*
     1 RU(I,J,K)) -(W(I,J,K)+W(I,J,K-1)+W(I-1,J,K-1)+W(I-1,J,K))/
```

```
     2 (4.*RU(I-1,J,K)) )/RV(I,J,K)
      DUT=RDTH*((U(I,J,K)+U(I,J,K+1)+U(I-1,J,K+1)+U(I-1,J,K))/4.
     1 -(U(I,J,K)+U(I-1,J,K)+U(I-1,J,K-1)+U(I,J,K-1))/4.)/RV(I,J,K)
C
      GG(I,J,K)=MU(I,J,K)*(2.*((RDR*(U(I,J,K)-U(I-1,J,K)))**2 +
     1 (U(I,J,K)/RU(I,J,K) +(RDTH/RV(I,J,K))*(W(I,J,K)-W(I,J,K-1)))
     2 **2 +
     2 (RDZ*(V(I,J,K)-V(I,J-1,K)))**2) + TAUW(I,K)**2/MU(I,J,K)+ (DWR
     3 +DUT)**2    )
C
       IF(ZPLUSW(I,K).LE.11.63) GO TO 5223
C
       TWWRZO(I,K)=DENU*CMUPQ*CAPPA*SQRTK*U(I,J,K)/ALOG(ELOG*ZPLUSW(I,K
     1 ))
       TWWTZO(I,K)=DENU*CMUPQ*CAPPA*SQRTK*W(I,J,K)/ALOG(ELOG*ZPLUSW(I,
     1  K))
C
       GO TO 5220
5223   TWWRZO(I,K)=VISCOS*U(I,J,K)/ZP
       TWWTZO(I,K)=VISCOS*W(I,J,K)/ZP
5220   CONTINUE
C
C==============  PLOTTING DATA GENERATION  ============================
C
C
         IF(CYCLE.NE.300) GO TO 636
C
       DO 5280 K=2,KM1
       DO 5300 I=2,IMAX
       DO 5300 J=2,JMAX
        UA=(U(I,J,K)+U(I-1,J,K))/2.
        VA=(V(I,J,K)+V(I,J-1,K))/2.
C
         UA=UA*0.045
         VA=VA*0.045
C
        XX(I,J,1)=(0.3*I)-0.15-UA/2.
        XX(I,J,2)=XX(I,J,1)+UA/2.
        XX(I,J,3)=XX(I,J,2)+UA/2.
        YY(I,J,1)=(0.6*J)-0.3-VA/2.
        YY(I,J,2)=YY(I,J,1)+VA/2.
        YY(I,J,3)=YY(I,J,2)+VA/2.
        VEL(I,J)=SQRT(VA**2 +UA**2 )
        TR(I,J)=(J-1)*DELZ-DELZ/2.
5300   CONTINUE
C
C
       M=0
       DO 5400 I=2,IMAX
       DO 5400 J=2,JMAX
       DO 5400 L=1,3
       M=M+1
       X(M)=XX(I,J,L)
       Y(M)=YY(I,J,L)
```

```
5400   CONTINUE
C
       DO 5450 I=1,432
        WRITE(14,555) X(I),Y(I)
5450   CONTINUE
555    FORMAT(E12.5,5X,E12.5)
C
5280   CONTINUE
C
C-----------------------------------------------------------------------
C
       DO 5500 K=2,KM1
       DO 5500 I=2,IM1
       VV(I,K,1)=V(I,1,K)
       VV(I,K,2)=(V(I,3,K)+V(I,4,K))/2.
       VV(I,K,3)=V(I,6,K)
       VV(I,K,4)=(V(I,8,K)+V(I,9,K))/2.
       VV(I,K,5)=V(I,11,K)
       UU(I,K,1)=U(I,1,K)
       WW(I,K,1)=W(I,1,K)
       UU(I,K,2)=U(I,4,K)
       WW(I,K,2)=W(I,4,K)
       UU(I,K,3)=(U(I,6,K)+U(I,7,K))/2.
       WW(I,K,3)=(W(I,6,K)+W(I,7,K))/2.
       UU(I,K,4)=U(I,9,K)
       WW(I,K,4)=W(I,9,K)
       UU(I,K,5)=U(I,11,K)
       WW(I,K,5)=W(I,11,K)
5500   CONTINUE
C
C
       DO 5550 K=2,KMAX
       DO 5550 L=1,5
       WRITE(15,556) (UU(I,K,L),I=2,IM1)
       WRITE(15,556) (VV(I,K,L),I=2,IM1)
       WRITE(15,556) (WW(I,K,L),I=2,IM1)
5550   CONTINUE
C
556     FORMAT(5(E12.5),/,5(E12.5))
C
636      CONTINUE
C
C
C================  PRINTING SECTION  ==================================
C
      PRINT 53,ITER,T,CYCLE
      PRINT 200,D
      PRINT 201, DELP
 200  FORMAT(1X,' D= ',E16.6,/)
 201   FORMAT(1X,' DELP = ',E16.6,/)
C     INTERMEDIATE PRINTING
       IF(CYCLE.LE.40) IIN=40
       IF(CYCLE.GT.40) IIN=40
       DO  363  II=10,400,IIN
```

```
           IF(CYCLE.EQ.II)  GO TO 5152
  363    CONTINUE
         IF(CYCLE.EQ.290) GO TO 5152
       IF(CYCLE.EQ.CYCMAX) GO TO 5152
       IF(CYCLE.EQ.1) GO TO 5152
          GO TO 5251
  5152 CONTINUE
C
       DO 5250 KK=1,KMAX
       K=KMAX-KK+1
       WRITE(6,42) K
       WRITE(6,48)
       DO 7001 J=1,JMAX
       JJ=JMAX-J+1
       WRITE(6,47) (U(I,JJ,K),I=1,IMAX)
  7001 CONTINUE
       WRITE(6,49)
       DO 7002 J=1,JMAX
       JJ=JMAX-J+1
       WRITE(6,47) (V(I,JJ,K),I=1,IMAX)
  7002 CONTINUE
       WRITE(6,52)
       DO 7003 J=1,JMAX
       JJ=JMAX-J+1
       WRITE(6,47) (W(I,JJ,K),I=1,IMAX)
  7003 CONTINUE
  7014 CONTINUE
       WRITE(6,51)
       DO 7004 J=1,JMAX
       JJ=JMAX-J+1
       WRITE(6,47) (P(I,JJ,K),I=1,IMAX)
  7004 CONTINUE
       WRITE(6,54)
       DO 7005 J=1,JMAX
       JJ=JMAX-J+1
       WRITE(6,47) (TK(I,JJ,K),I=1,IMAX)
  7005 CONTINUE
       WRITE(6,55)
       DO 7006 J=1,JMAX
       JJ=JMAX-J+1
       WRITE(6,47) (E(I,JJ,K),I=1,IMAX)
  7006 CONTINUE
       WRITE(6,56)
       DO 7007 J=1,JMAX
       JJ=JMAX-J+1
       WRITE(6,47) (MU(I,JJ,K),I=1,IMAX)
  7007 CONTINUE
 5250  CONTINUE
  5251 CONTINUE
C
       SUMU=0.0
       SUMV=0.0
       SUMW=0.0
       DO 6100 I=2,IM1
```

```
      DO 6100 J=2,JM1
      DO 6100 K=2,KM1
      SUMU=SUMU+ABS(UN(I,J,K)-U(I,J,K))
      SUMV=SUMV+ABS(VN(I,J,K)-V(I,J,K))
      SUMW=SUMW+ABS(WN(I,J,K)-W(I,J,K))
 6100 CONTINUE
C     RETURN TO PRESSURE ITERATION CYCLE IF THESE WERE ONLY CHECKPRINTS
C     ----------------------------------------------------------------
C
       IF(IWRITE.EQ.1)  GO TO 3501
C
C     REPACKAGING
      DO 6101 I=1,IMAX
      DO 6101 J=1,JMAX
      DO 6101 K=1,KMAX
      UN(I,J,K)=U(I,J,K)
      VN(I,J,K)=V(I,J,K)
      WN(I,J,K)=W(I,J,K)
      TKN(I,J,K)=TK(I,J,K)
      EN(I,J,K)=E(I,J,K)
      MUN(I,J,K)=MU(I,J,K)
 6101 CONTINUE
      PRINT 90
      PRINT 91,U(3,4,3),V(3,4,3),W(3,4,3),STORIT,SUMU,SUMV,SUMW,SUMD
C     ADVANCE TIME AND CYCLE
      T=T+DELT
      IF(T.GT.CYCMAX*DELT)GO TO 6500
      CYCLE=CYCLE+1
      GO TO 1000
 6500 CONTINUE
 42   FORMAT(//,20X,'K =',I2,/,19X, 7('='))
      STOP
 47    FORMAT(1X,12(E11.4))
 48   FORMAT(//,2X,8H U-FIELD,/)
 49   FORMAT(///,2X,8H V-FIELD,/)
 51   FORMAT(///,2X,8H P-FIELD,/)
 52   FORMAT(///,2X,8H W-FIELD,/)
 53   FORMAT(6X,5HITER=,I5,9X,5HTIME=,1PE12.5,10X,6HCYCLE=,I4)
 54   FORMAT(///,2X,'TK - FIELD',/)
 55   FORMAT(///,2X,'E - FIELD',/)
 56   FORMAT(///,2X,'MU - FIELD',/)
 90    FORMAT(///4X,'UMON  ',8X,'VMON  ',7X,'WMON  ',7X,'PRESS ITER',4X,
     1 'SUMU',10X,'SUMV',10X,'SUMW',10X,'SUMD'//)
 91   FORMAT(8E14.4)
C
      END
```

VITA 2

Ahmed A. Busnaina

Candidate for the Degree of

Doctor of Philosophy

Thesis: TRANSIENT THREE-DIMENSIONAL PREDICTIONS OF TURBULENT FLOWS IN CYLINDRICAL AND CARTESIAN COORDINATE SYSTEMS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Benghazi, Libya, September 2, 1953, the son of Mr. and Mrs. Ali Busnaina.

Education: Graduated from Benghazi Secondary School, Benghazi, Libya, in May, 1971; received the Bachelor of Science degree in Mechanical Engineering from University of Al Fateh in July, 1976; completed requirements for the Master of Science degree at Oklahoma State University in December, 1979; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in July, 1983.

Professional Experience: Maintenance engineer, General Wires and Electrical Products Company, Benghazi, Libya, 1976; maintenance engineer, Esso Oil Company, Briga, Libya, February, 1977, to August, 1977; teaching assistant, University of Garyounis, Benghazi, Libya, August to December, 1977; graduate research assistant, Oklahoma State University, 1978-1979; registered engineer intern (EIT), State of Oklahoma, No. ET-5857; Teaching and research associate, School of Mechanical and Aerospace Engineering, Oklahoma State University, 1980-1983.

Professional Societies: American Institute of Aeronautics and Astronautics; American Society of Mechanical Engineers; National Society of Professional Engineers; Pi Tau Sigma (Mechanical Engineering Honor Society).