



8-1995

A parallel finite element algorithm for 3D incompressible flow in velocity-vorticity form

Kwai Lambeck Wong
University of Tennessee

Follow this and additional works at: https://trace.tennessee.edu/utk_graddiss

Recommended Citation

Wong, Kwai Lambeck, "A parallel finite element algorithm for 3D incompressible flow in velocity-vorticity form." PhD diss., University of Tennessee, 1995.
https://trace.tennessee.edu/utk_graddiss/6151

This Dissertation is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a dissertation written by Kwai Lambeck Wong entitled "A parallel finite element algorithm for 3D incompressible flow in velocity-vorticity form." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Engineering Science.

A. J. Baker, Major Professor

We have read this dissertation and recommend its acceptance:

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

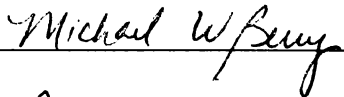
I am submitting herewith a dissertation written by Kwai Lam Wong entitled "A Parallel Finite Element Algorithm For 3D Incompressible Flow In Velocity-Vorticity Form". I have examined the final copy of this dissertation for form and content and recommend that it be accepted for the degree of Doctor of Philosophy, with a major in Engineering Science.

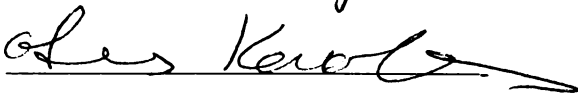


A. J. Baker, Ph.D

Major Professor

We have read this dissertation
and recommend its acceptance:







Accepted for the Council:



Associate Vice Chancellor

and Dean of The Graduate School

**A Parallel Finite Element Algorithm For 3D
Incompressible Flow In Velocity-Vorticity Form**

A Dissertation

Presented for the

Doctor of Philosophy

The University of Tennessee, Knoxville

Kwai Lam Wong

(黃桂林)

August 1995

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my major advisor, Professor A. J. Baker, for his genuine, unrelenting, and passionate guidance throughout these seemingly endless years of educational pursuit. He is always resourceful, helpful and most importantly understandable. As a mentor, he has been providing tremendous morale as well as technical support for the past years.

I wish to express my greatest appreciation towards our fellow members of the "CFD club", Mr. Dee H. Wong, Mr. Joseph Michael Barton, and particularly the wise one, Mr. David Chaffin, with whom I constantly consult. Dr. Paul Williams, Dr. Subrata Roy, and Dr. Joe Iannelli also provided numerous help and valuable insight throughout these years; my full gratitude to them.

I would like to thank Dr. Christian Halloy and staff of the Joint Institute of Computational Science for their immense support and advice for the parallel computing platform. Computing support at the UT CFD Laboratory and the Computer Science Department, which provides the most important computational tool of my research, the CM5 parallel computer, are gratefully acknowledge.

I also would like to gratefully thank my doctoral committee members: Professors T. G. Carley, O. Karakaskian and Dr. Michael Berry, for their enthusiastic participation and generous technical support of this research.

Last, but not the least, I would like to express my deepest thank to Professor John Forrester for his guidance and mentoring of my teaching career in the Department of Engineering Science and Mechanics at the University of Tennessee.

ABSTRACT

In the last decade, developments and advancement in computer technology, especially the availability of the massively parallel machine, have escalated the numerical treatment of complex fluid flow problems to a new height. Numerical simulation of incompressible viscous fluid flow, often associated with practical industrial and environmental situations, is receiving intense scrutiny to perform in the promising distributed parallel computing environment. On the other hand, the field of computational fluid dynamics continues to explore and exploit unified and versatile formulations, in contention with the notorious divergence-free velocity field constraint, for incompressible Navier-Stokes equations that encompass fluid flow in two- and three-dimensions. The velocity-vorticity formulation for the incompressible Navier-Stokes equations is chosen with the full extent to resolve these issues.

In the present dissertation, a new finite element implementation for two- and three-dimensional incompressible fluid flow is developed in the velocity-vorticity form. Pressure is eliminated analytically by taking the curl of the momentum equations, and vorticity is introduced as the active variable. The formulation consists of the three derived vorticity transport equations in conjunction with three velocity Poisson equations. Satisfaction of the continuity constraint is cast onto the specific treatment of the kinematic vorticity boundary condition for the no slip wall. A divergence-free solution is guaranteed with equal order finite element interpolation functions for all state variables.

The fully coupled system for the velocity-vorticity formulation on solved on a distributed-memory parallel computer. The CM5 machine with 32 processors at the University of Tennessee is utilized to perform computations for 3-D problems. The biconjugate gradient stablized (BiCGSTAB) sparse iterative solver is employed to yield an efficient parallel solution algorithm.

A broad range of verifications and benchmark test problems in two and three dimensions are examined to establish the validity and accuracy of the newly developed velocity-vorticity formulation for the unsteady laminar incompressible Navier-Stokes equations

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	General Review	1
1.1.1	Primitive Velocity-Pressure Formulation	2
1.1.2	Streamvector-Vorticity Formulation	4
1.1.3	Velocity-Vorticity Formulation	5
1.2	Present Research Investigation	7
2	VELOCITY-VORTICITY FORMULATION	11
2.1	Governing Equations	11
2.1.1	Primitive Variable Form	11
2.1.2	Velocity-Vorticity Form	13
2.1.3	Two Dimensional Formulation	16
2.2	Boundary Conditions	16
2.2.1	Velocity Poisson Equations	16

2.2.2	Vorticity Transport Equations	17
2.2.3	Present Vorticity Kinematic Equations	20
3	PARALLEL FINITE ELEMENT ALGORITHM	25
3.1	Finite Element Formulation	25
3.1.1	Galerkin Weak Statement	26
3.1.2	Discrete Finite Element Weak Statement For The Velocity	27
3.1.3	Discretized Finite Element Weak Statements For The Vorticity	28
3.1.4	Discretized Finite Element Weak Statement For The Temperature	30
3.1.5	Newton Statement	31
3.1.6	Jacobian Matrix	32
3.1.7	Vorticity Kinematic Boundary Conditions	34
4	PARALLEL COMPUTATIONAL PROCEDURE	36
4.1	CM5 Parallel Computer	36
4.2	Parallel Strategy	37
4.3	Sparse Matrix Data Structure	40
4.4	Sparse Iterative Solvers	42

4.4.1	Conjugate Gradient Squared Method (CGS)	42
4.4.2	BiConjugate Gradient Stabilized Method (BiCGSTAB)	44
4.4.3	Preconditioner	46
4.5	Matrix Vector Multiplication	46
4.6	Vector Product Computation	54
4.7	Boundary Condition Implementation	55
5	RESULTS AND DISCUSSIONS	57
5.1	Straight Channel Flow	58
5.1.1	Fully-Developed Flow	58
5.1.2	Developing Flow	61
5.2	Backward Facing Step	68
5.3	Lid Driven Cavity	71
5.3.1	Lid-Driven Cavity in Two Dimensions	71
5.3.2	Lid-Driven Cavity in Three Dimension	80
5.4	Thermal Cavity	94
6	SUMMARY AND CONCLUSIONS	110

BIBLIOGRAPHY	113
VITA	122

LIST OF TABLES

5.1	: Entrance length for developing flow in a square duct	66
5.2	: Flow characteristics of backward facing step	69
5.3	: Location of vortices for 2-D lid-driven cavity	78
5.4	: Centerline minimum u-velocity by Guj and Stella (1993) at Re=1000	94
5.5	: U-velocity along the vertical centerline of the 3-D driven cavity . . .	95

LIST OF FIGURES

2.1	: Schematic of kinematic vorticity boundaries	18
3.1	: Layout of jacobian matrix	33
3.2	: Imposing kinematic vorticity equation in jacobian matrix	35
4.1	: A schematic of the computational domain	39
4.2	: A finite element mesh for the example problem	48
5.1	: Geometry and boundary conditions for 2-D channel flow	59
5.2	: Geometry for 3-D square channel flow	60
5.3	: Boundary condition for straight channel flow	60
5.4	: Initial and steady state solutions for 2-D fully developed channel flow	62
5.5	: Initial and steady state center plane solutions for 3-D fully developed channel flow	63
5.6	: Velocity profile for square channel flow a) centerline profile b) per- spectives of axial flow distributions.	65

5.7	: Velocity Profile for 3-D developing channel flow at the exit	67
5.8	: Geometry and boundary conditions for backward facing step	68
5.9	: Velocity profile for backward facing step a) entire duct, b) inlet section	70
5.10	: Geometry and key features of the 2-D driven cavity	72
5.11	: Finite element mesh for 2-D driven cavity	73
5.12	: Streamline contour plot of the 2-D driven cavity at $Re=400$	73
5.13	: Vorticity contour plot for 2-D driven cavity at $Re=400$	74
5.14	: Streamline contour plot of the 2-D driven cavity at $Re=1000$	75
5.15	: Vorticity contour plot for 2-D driven cavity at $Re=1000$	75
5.16	: Streamline contour plot of the 2-D driven cavity at $Re=2000$	76
5.17	: Vorticity contour plot for 2-D driven cavity at $Re=2000$	76
5.18	: Streamline contour plot of the 2-D driven cavity at $Re=3200$	77
5.19	: Vorticity contour plot for 2-D driven cavity at $Re=3200$	77
5.20	: Solution details of 2-D driven cavity for $Re = 1000$, $M=64 \times 64$	79
5.21	: Non-uniform finite element mesh for 3-D driven cavity, $M=32^3$	81
5.22	: Perspective 3-D summary for driven cavity solution, $Re=100$, $M=32^3$	82

5.23	: Perspective 3-D summary for driven cavity solution, $Re=400$, $M=32^3$	83
5.24	: Perspective 3-D summary for driven cavity solution, $Re=1000$, $M=32^3$	84
5.25	: 2-D planar projections of mid-plane velocity vector field for 3-D driven cavity, $Re = 100, 400, 1000$, $M=32^3$	86
5.26	: Ω_x contour plot for 3-D driven cavity, $Re=100, 400, 1000$, $M=32^3$	88
5.27	: Ω_y contour plot for 3-D driven cavity, $Re=100, 400, 1000$, $M=32^3$	89
5.28	: Ω_z contour plot for 3-D driven cavity, $Re=100, 400, 1000$, $M=32^3$	90
5.29	: 3-D mid-plane centerline u-velocity plot at $Re=100$	91
5.30	: 3-D mid-plane centerline u-velocity plot at $Re=400$	92
5.31	: 3-D mid-plane centerline u-velocity plot at $Re=1000$	93
5.32	: Non-uniform finite element mesh for 3-D thermal cavity, $M=32^3$	97
5.33	: Velocity vector and temperature contour at $Ra = 10^4$ for 3-D thermal cavity	98
5.34	: Velocity vector and temperature contour at $Ra = 1.6 \times 10^5$ for 3-D thermal cavity	99
5.35	: Velocity vector and temperature contour at $Ra = 10^6$ for 3-D thermal cavity	100

5.36 : Velocity vector and temperature contour at $Ra = 1.6 \times 10^7$ for 3-D thermal cavity, $M = 32^3$, original mesh	101
5.37 : Velocity vector and temperature contour at $Ra = 1.6 \times 10^7$ for 3-D thermal cavity, $M = 32^3$, refined to fixed temperature walls	102
5.38 : Streamlines for the projected velocity vector field on the symmetry plane $Ra = 10^4 - 1.6 \times 10^7$, $M = 32^3$	105
5.39 : Temperature profiles in the centerline of the symmetry plane for $Ra = 10^4$, $Ra = 1.6 \times 10^5$, $Ra = 10^6$, and $Ra = 1.6 \times 10^7$	106
5.40 : Centerline velocity profiles of the symmetry plane for $Ra = 10^4$, $Ra = 1.6 \times 10^5$, $Ra = 10^6$, and $Ra = 1.6 \times 10^7$	107
5.41 : Temperature profile for $Ra = 10^6$ at various z location on the symmetry plane	108
5.42 : Temperature profile for $Ra = 10^5$ for air in the centerline of the symmetry plane	109

INTRODUCTION

1.1 General Review

Batchelor (1967) mentioned in his classic fluid dynamics book that: "I regard flow of an incompressible viscous fluid as being the center of fluid dynamics by virtue of its fundamental nature and its practical importance." With the advent and availability of computer resources, Computational Fluid Dynamics (CFD) has emerged as an important and economical research and development tool in the scientific world. Numerical simulation of incompressible viscous fluid flow is commonly used nowadays to solve practical industrial and environmental fluid flow problems. The finite element method, with a rigorous mathematical history and a wide range of applicability, is currently a hot topic of research in CFD especially for three dimensional incompressible viscous flow simulation.

Different finite element formulations to predict viscous incompressible flows have been introduced in the last decade. They may be classified into two major categories, primitive variable formulations, and derived variable (vorticity) formulations. Gresho & Sani (1987) point out that the pressure appearing in the incompressible Navier-Stokes equations (INS) is not a thermodynamic variable. Instead, it is a mathematical

lagrangian constraint that enforces conservation of mass via a divergence-free velocity field throughout the entire flow field at any time. The primary challenge of all INS formulations is to satisfy the continuity equation, $\nabla \cdot \mathbf{u} = 0$, either contending with how the pressure variable works in a CFD theory, or working around it.

1.1.1 Primitive Velocity-Pressure Formulation

For a direct finite element formulation for the INS equations, the momentum and the continuity equations are solved simultaneously. A specific combination of mixed finite element bases for velocity and pressure has to be used to guarantee a stable convergent solution. This set of bases is required to satisfy the div-stability condition, also known as the LBB or the inf-sup condition, as stated by Ladyzhenskaya (1969) , Babuska (1973) , and Brezzi (1974). The terminal solution matrix resulting from the direct method is ill-conditioned due to a large number of zeros on the diagonal, corresponding to the absence of the pressure variable in the continuity equation. In three dimensions, the direct mixed finite element method is also expensive to compute, hence its application tends to be limited for two dimensional incompressible fluid flow problems.

In the penalty method (Temam 1968, Hughes, Liu & Brooks 1979), the pressure term is replaced by a constraint expression comprised of the divergence of an approximation to the velocity and a lagrange multiplier. Solutions to the momentum equations are then "encouraged" to satisfy continuity as the lagrange multiplier approaches infinity. Penalty solutions are proven to converge to exact solutions in linear

Stokes flow only. The major advantage of the penalty method is the total elimination of the appearance of pressure in the momentum equations, hence it can be postprocessed from the obtained velocity field. The truly significant disadvantage is the high degree of ill-conditioning introduced into the terminal matrix algebra statement. For fluid flow at large Reynolds number, the Newton jacobian matrix is close to singular which makes it extremely expensive to solve and requires direct, e.g., Gaussian elimination, methodology.

A "pressure-relaxation" method is by far the most widely used formulation for production CFD codes. The momentum solution is initially started with a pseudo pressure distribution. The velocity is then projected to a divergence free vector field by solving a Poisson equation for a correction field. Velocity correction and/or pressure correction is then used to update the state variables iteratively. Gresho (1991*b*) states that a divergence free velocity field may be so-obtained by solving the momentum equations and a pressure Poisson equation with proper boundary conditions imposed. It is appropriate that the pressure Poisson equation, in such cases, has to be used to capture the genuine pressure field. However, the formulation is intrinsically transient since it is evolved from the basic ideas of projection method developed by Chorin (1968). As a result, a large time step may not be used. Thus obtaining a steady-state solution for a three dimensional problem could be expensive.

A detailed comparison of various pressure relaxation schemes is documented by Williams (1993). The Continuity Constraint Method (CCM) developed by Williams (1993) addresses the key theoretical issues. An upwind finite element scheme for three

dimensional incompressible flow problems based on such methodology is reported by Fujima, Tabata & Fukasawa (1994).

1.1.2 Streamvector-Vorticity Formulation

In contrast to primitive variable formulations, pressure can be eliminated analytically by taking the curl of the momentum equations, and the vorticity vector, Ω , is introduced as a dependent variable. The compatibility equation, $\nabla^2 \Psi + \Omega = 0$, exists with the streamfunction vector, Ψ , which may be used to solve INS equations in conjunction with the derived vorticity equation. By definition, the streamvector definition satisfies the continuity equation identically. The streamfunction-vorticity formulation is very successful for two dimensional flow computations that are not pressure-difference driven.

The first finite element implementation of the 2-D $\omega - \psi$ method is reported by Baker (1973). However, the formulation suffers from severe limitations when extended to three dimensional streamvector-vorticity form since the streamvector is not unique and must itself be divergence-free, or constrained to be so. A gradient of any scalar function may be added to the streamvector without affecting the velocity field. Specific restrictions have to be imposed on the formulation to ensure a solenoidal flow. As a remedy, Aregbesola & Burley (1977) introduced the vector and scalar potential vorticity formulation. The velocity vector field is decomposed into potential and divergence-free components. The continuity constraint is satisfied automatically by the scalar potential. However, boundary conditions for the scalar and the vector

potentials are often difficult to implement, especially at an outflow boundary. A solution for a three dimensional curved duct flow was presented by Yang & Camarero (1991).

1.1.3 Velocity-Vorticity Formulation

By using the definition of vorticity, the continuity equation, and the curl of the momentum equations, the incompressible Navier-Stokes equations can be rewritten exclusively in terms of velocity and vorticity. The formulation consists of three velocity Poisson equations and three vorticity transport equations.

There are several distinctive advantages to using the velocity-vorticity formulation for INS equations. First, pressure does not explicitly appear in the field equations, thus difficulties associated with the determination of the pressure boundary conditions, especially at the outflow boundary (Gresho 1991*c*), are avoided. Second, the boundary conditions for the formulation are clearer and easier to impose than for the vector potential formulation. Third, it can be directly extended to three dimensional flow computations.

The formulation is shown by Gunzburger & Peterson (1988) to be well posed. The incompressibility constraint is satisfied implicitly within the context of the velocity Poisson equations. However, in order to solve the velocity-vorticity equations system successfully, one must make up a boundary condition for the vorticity on boundary segments where the velocity is specified, e. g. no-slip. Such boundary conditions represent a crucial ingredient towards resolving a mass-conserving convergent

solution. It is pointed out by Daube (1992) that satisfying the continuity equation is equivalent to enforcing the definition of the vorticity as the curl of the velocity field, and that this requirement reduces to boundary conditions coupling the velocity and vorticity on a solid wall.

The velocity-vorticity formulation was first reported by Fasel (1976) to study the stability of the boundary layers in two dimensions. It was extended by Dennis, Ingham & Cook (1979) and Agarwal (1973) to compute steady three dimensional cavity flows. Gatski, Grosch & Rose (1982) applied the compact finite difference Keller-Box scheme to unsteady incompressible flows. Farouk & Booz (1984) presented a numerical study of the natural and forced convection and heat transfer in a two-dimensional annulus.

When the velocity-vorticity formulation is written in terms of a standard finite difference discretization, the numerical solution may not satisfy the continuity constraint and is found to be very sensitive to the discretization of the equation used to evaluate the vorticity at a solid wall via the vorticity kinematic boundary condition. In order to satisfy the constraint that the velocity field be divergence free, a spatial discretization based on a staggered grid arrangement is generally employed. Using such grid arrangement, Orlandi (1987) was able to obtain a solution for flow over a two dimensional backward facing step using a block ADI method. Steady state solutions for two dimensional driven cavity flows at $Re \leq 5000$, were reported by Guj & Stella (1988) using the false transient method with parabolized Poisson velocity equations on a staggered grid. An influence matrix technique was used by Daube (1992) to solve an axisymmetric flow in a closed cylinder.

A velocity-vorticity formulation of the two dimensional Navier-Stokes equations using the finite element method was first reported by Guevremont, Habashi & Hafez (1990). Their methodology used mixed finite element interpolation functions for the velocity components and the vorticity similar to that employed in the primitive variable formulation. A quadratic finite element interpolation was used for the velocity components while linear was used for the vorticity. A fully coupled Newton matrix system of equations was solved simultaneously by a direct solver. The vorticity boundary condition at a no-slip wall was imposed by evaluating the weighted area integral of the definition of vorticity in terms of the velocity at each boundary element. The formulation was also used to solve a subsonic internal enclosed flow problem.

Solutions for the three dimensional driven cavity benchmark using a finite difference method in conjunction with a staggered grid arrangement were reported by Dacles & Hafez (1990) and Napolitano & Pascazio (1991). A direct extension of the false transient method for three dimensional driven cavity solutions for $Re \leq 2000$ was reported by Guj & Stella (1993). A three dimensional extension of the velocity-vorticity formulation using the finite element method was also presented by Guevremont, Habashi, , Kotiuga & Hafez (1993) for driven cavity flow at $Re = 100$ and $Re = 400$.

1.2 Present Research Investigation

Quartapelle (1993) observes that: " this pair of variables (velocity and vorticity) is most suited from the viewpoint of a fluid dynamic description of incompressible

viscous flow. In fact the vorticity is governed by an extensively studied and well understood dynamical equation while the velocity field, which by definition embodies the kinematical aspect of the problem, can be related to the vorticity field by a very simple elliptic equation.”

As mathematically viable as the velocity-vorticity formulation can be, numerical solutions for the three dimensional incompressible Navier-Stokes equations using such a formulation remain a challenge for researchers. Solutions for simple fluid flow problems are predominantly solved by using finite differences on regular staggered grids. On the other hand, the finite element solutions presented by Guevremont et al. (1993) inherit the rigidity of the staggered grid arrangement with a pair of unequal order shape functions chosen for the velocity components and the vorticity components.

The three dimensional velocity-vorticity formulation contains six variables. Often, the system of velocity-vorticity equations has to be solved coupled together to guarantee a solenoidal velocity field Guj & Stella (1993). Solving such a fully coupled system implicitly on a sequential machine would be prohibitively expensive. Hence, efficient parallel numerical algebra procedures represent an important issue to be pursued. It is these reasons that motivate this investigation of a new time accurate, parallel finite element algorithm for the incompressible Navier-Stokes equations in velocity-vorticity form.

The present finite element methodology uses equal order of finite element interpolation functions for the velocity and vorticity components on a finite element mesh. A new second order kinematic vorticity boundary condition is derived to guarantee

the satisfaction of the incompressibility constraint. The three dimensional system of velocity-vorticity equations is solved on a distributed-memory parallel computer. The Newton jacobian matrix and residual vector within elements can be calculated simultaneously on multiple processor nodes. The conjugate gradient squared (CGS) (Sonneveld 1989) and the biconjugate gradient stabilized (BiCGSTAB) (Van der Vorst 1992) sparse iterative solvers are used to solve the terminal matrix statement. They can be effectively implemented on a distributed memory parallel machine such as the CM5. The global Newton jacobian matrix is not formed; instead, vector data needed for matrix-vector multiplication operations are transported through a message passing communication subroutine to the corresponding processor node.

Although these sparse solvers are effective in parallel computations, they are not robust. For large Reynolds (Re) or Rayleigh (Ra) number flows, where convection dominates the flow processes, the Newton jacobian becomes ill-conditioned and stiff. As the condition number of the jacobian matrix increases, these conjugate-like sparse solvers have a tendency to diverge. An effective preconditioner, like the incomplete LU decomposition which can vastly improve the condition number of the matrix, is unfortunately difficult for efficient parallel implementation. One viable way to overcome the problem is to keep the time step small, $\Delta t < 1.0$, hence the well-conditioned mass matrix of the finite element procedure for the unsteady time term will help to diagonalize the terminal jacobain matrix, and consequently will reduce its condition number.

A discussion of the mathematical formulation in the velocity-vorticity form is presented in Chapter 2. In Chapter 3, a detail description of the parallel finite element procedure is illustrated. Chapter 4 summarizes the implementation of the parallel programming procedure. In Chapter 5, relevant test cases and benchmarks used to validate various aspects of the formulation are given and finally, conclusions of this research are presented in Chapter 6.

VELOCITY-VORTICITY FORMULATION

2.1 Governing Equations

2.1.1 Primitive Variable Form

The nondimensional laminar incompressible Navier-Stokes equations system with Boussinesq gravity body-force approximation is

Continuity :

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

Momentum :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla P + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} - \frac{\text{Gr}}{\text{Re}^2} \Theta \hat{\mathbf{g}} \quad (2.2)$$

Energy :

$$\frac{\partial \Theta}{\partial t} + (\mathbf{u} \cdot \nabla) \Theta = \frac{1}{\text{RePr}} \nabla^2 \Theta \quad (2.3)$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) = (u, v, w)$ is the velocity vector field, t is the time, $\mathbf{x} = (x, y, z)$ is the spatial coordinate, Θ is the potential temperature, $\hat{\mathbf{g}}$ is the gravity unit vector and P is the kinematic pressure. The nondimensional parameters are Reynolds number (Re), Prandtl number (Pr), and Grashof number (Gr) defined as,

$$\text{Re} = \frac{U_r L_r}{\nu} ; \quad \text{Pr} = \frac{\nu}{\alpha} ; \quad \text{Gr} = \frac{\beta g \Delta T_r L_r^3}{\nu^2}$$

where L_r and U_r are the reference length and velocity respectively, g is the gravity acceleration, ν is the kinematic viscosity, β is the coefficient of volume expansion, α is the thermal diffusivity, and ΔT_r is the reference temperature difference.

For isothermal flow, such as straight channel flow and the lid-driven cavity, the energy equation is not used and the Boussinesq buoyancy term is neglected in the momentum equations.

The system of equations (2.1) - (2.3) is subjected to suitable boundary conditions for $t > 0$ and initial conditions at $t = 0$. Typical boundary conditions are

$$\mathbf{u}(\mathbf{x}_S, t) = \mathbf{b}(\mathbf{x}_S, t) \quad (2.4)$$

$$\Theta(\mathbf{x}_{\tilde{S}}, t) = \mathbf{f}(\mathbf{x}_{\tilde{S}}, t) \quad (2.5)$$

$$(\hat{\mathbf{n}} \cdot \nabla) \mathbf{u}|_C = 0 \quad (2.6)$$

$$\hat{\mathbf{n}} \cdot \nabla \Theta|_{\tilde{C}} = 0 \quad (2.7)$$

where $\hat{\mathbf{n}}$ is the unit normal to the boundary, \tilde{S} is the boundary surface segment for a known temperature, \mathbf{f} , and \tilde{C} is the heat flux boundary which, here, is assumed to be adiabatic for simplicity, S is the boundary surface segment for a known velocity field, \mathbf{x}_S is a spatial point on S , \mathbf{b} is the boundary velocity, and C is the outflow boundary for the velocity. When the boundary is a rigid no-slip solid wall in contact with the fluid, the boundary velocity, \mathbf{b} , is zero. For a simplified case where outflow boundary does not exist, $C = \emptyset$, the boundary velocity, \mathbf{b} , must satisfy the global condition

$$\oint \hat{\mathbf{n}} \cdot \mathbf{b} dS = 0 \quad ; \quad t \geq 0 \quad (2.8)$$

Boundary conditions of Dirichlet or Neumann type may be enforced for the pressure on inflow or outflow boundaries.

The initial conditions are

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) ; \quad \nabla \cdot \mathbf{u}_0 = 0 \quad (2.9)$$

$$\Theta(\mathbf{x}, 0) = \Theta_0(\mathbf{x}) \quad (2.10)$$

where \mathbf{u}_0 is an initial solenoidal velocity field and Θ_0 is the initial temperature.

Equations (2.1) - (2.10) represent a well-posed system for the incompressible Navier-Stokes equations in primitive variable form which will admit a unique solution (Quartapelle 1993).

2.1.2 Velocity-Vorticity Form

The vorticity vector, $\boldsymbol{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$, is defined as

$$\boldsymbol{\Omega} = \nabla \times \mathbf{u} \quad (2.11)$$

Taking the curl of the vorticity vector definition, (2.11), together with the incompressibility constraint (2.1) and the vector identity,

$$\nabla \times \boldsymbol{\Omega} = \nabla \times \nabla \times \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u} \quad (2.12)$$

the velocity Poisson vector equation is

$$\nabla^2 \mathbf{u} = -\nabla \times \boldsymbol{\Omega} \quad (2.13)$$

For the momentum equation, the nonlinear term is first expressed in the following form

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = (\nabla \times \mathbf{u} \times \mathbf{u}) + \nabla\left(\frac{1}{2}u^2\right) = \boldsymbol{\Omega} \times \mathbf{u} + \nabla\left(\frac{1}{2}u^2\right) \quad (2.14)$$

Again, taking the curl of the momentum equation, (2.2), all the gradient fields are eliminated. The momentum equation becomes

$$\frac{\partial \boldsymbol{\Omega}}{\partial t} + \nabla \times (\boldsymbol{\Omega} \times \mathbf{u}) = \frac{1}{\text{Re}} \nabla^2 \boldsymbol{\Omega} - \frac{\text{Gr}}{\text{Re}^2} \nabla \times \Theta \hat{\mathbf{g}} \quad (2.15)$$

Consider now the vector identity

$$\nabla \times (\boldsymbol{\Omega} \times \mathbf{u}) = (\mathbf{u} \cdot \nabla)\boldsymbol{\Omega} - (\boldsymbol{\Omega} \cdot \nabla)\mathbf{u} + \boldsymbol{\Omega} \nabla \cdot \mathbf{u} - \mathbf{u} \nabla \cdot \boldsymbol{\Omega} \quad (2.16)$$

But $\nabla \cdot \boldsymbol{\Omega} = 0$ and $\nabla \cdot \mathbf{u} = 0$, hence the vorticity equation assumes the form

$$\frac{\partial \boldsymbol{\Omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\Omega} - (\boldsymbol{\Omega} \cdot \nabla)\mathbf{u} = \frac{1}{\text{Re}} \nabla^2 \boldsymbol{\Omega} - \frac{\text{Gr}}{\text{Re}^2} \nabla \times \Theta \hat{\mathbf{g}} \quad (2.17)$$

which displays, in order, the unsteady term, the convection term, the stretching term, the viscous diffusion term and the buoyancy term.

Hence, the velocity-vorticity formulation for the laminar incompressible Navier-Stokes equations system with Boussinesq approximation in 3-D can be written as

$$\mathcal{L}(\mathbf{u}) = \nabla^2 \mathbf{u} + \nabla \times \boldsymbol{\Omega} = 0 \quad (2.18)$$

$$\mathcal{L}(\boldsymbol{\Omega}) = \frac{\partial \boldsymbol{\Omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\Omega} - (\boldsymbol{\Omega} \cdot \nabla)\mathbf{u} - \frac{1}{\text{Re}} \nabla^2 \boldsymbol{\Omega} + \frac{\text{Gr}}{\text{Re}^2} \nabla \times \Theta \hat{\mathbf{g}} = 0 \quad (2.19)$$

$$\mathcal{L}(\Theta) = \frac{\partial \Theta}{\partial t} + (\mathbf{u} \cdot \nabla)\Theta - \frac{1}{\text{RePr}} \nabla^2 \Theta = 0 \quad (2.20)$$

For an isothermal system, the formulation is subjected to the following initial and boundary conditions,

$$\boldsymbol{\Omega}(\mathbf{x}, 0) = \nabla \times \mathbf{u}_0(\mathbf{x}) \quad (2.21)$$

$$\mathbf{u}(\mathbf{x}_S, t) = \mathbf{b}(\mathbf{x}_S, t) \quad (2.22)$$

$$(\hat{\mathbf{n}} \cdot \nabla) \mathbf{u}|_C = 0 \quad (2.23)$$

$$(\hat{\mathbf{n}} \cdot \nabla) \boldsymbol{\Omega}|_C = 0 \quad (2.24)$$

$$\nabla \cdot \mathbf{u}(\mathbf{x}_S, t) = 0 \quad (2.25)$$

$$\nabla \cdot \boldsymbol{\Omega}(\mathbf{x}_S, t) = 0 \quad (2.26)$$

$$\hat{\mathbf{n}} \cdot \boldsymbol{\Omega}(\mathbf{x}_S, t) = \hat{\mathbf{n}} \cdot \nabla_S \times \mathbf{b}(\mathbf{x}_S, t) \quad (2.27)$$

The equation system (2.18)-(2.27) in the velocity-vorticity form for $C = \emptyset$ is proved by (Quartapelle 1993) to be equivalent to the primitive velocity-pressure form, (2.1)-(2.10).

Such formulation consists of three velocity Poisson equations that couple the velocity and vorticity components kinematically via the continuity constraint, and three vorticity transport equations that describes the dynamics of flow mechanisms. Gunzburger & Peterson (1988) point out that the problem is well posed even though just \mathbf{u} is specified at the solid wall. However, to successfully solve the momentum equation, a boundary condition for the vorticity on the no-slip wall must be specified. Such a boundary condition is given in Equation (2.27). As indicated by Quartapelle

(1993), Equation (2.27) is simply the component of the equation $\Omega = \nabla \times \mathbf{u}$ normal to S , in which the velocity boundary condition $\hat{\mathbf{n}} \times \mathbf{u}|_S = \hat{\mathbf{n}} \times \mathbf{b}$ has been taken into account to express $\hat{\mathbf{n}} \nabla \times \mathbf{u}$ in terms of $\hat{\mathbf{n}} \nabla_S \times \mathbf{b}$.

2.1.3 Two Dimensional Formulation

The isothermal velocity-vorticity formulation in two dimensions consists of two velocity Poisson equations and a single vorticity transport equation,

$$\mathcal{L}(u) = \nabla^2 u + \frac{\partial \Omega}{\partial y} = 0 \quad (2.28)$$

$$\mathcal{L}(v) = \nabla^2 v - \frac{\partial \Omega}{\partial x} = 0 \quad (2.29)$$

$$\mathcal{L}(\Omega) = \frac{\partial \Omega}{\partial t} + (\mathbf{u} \cdot \nabla) \Omega - \frac{1}{Re} \nabla^2 \Omega = 0 \quad (2.30)$$

which is greatly simplified. The stretching term vanishes in the two dimensional setting, and the $\nabla \times \Omega$ condition is not required. Instead of using the velocity Poisson equations, Gatski et al. (1982) solved the continuity equation, $\nabla \cdot \mathbf{u} = 0$, the vorticity definition, $\nabla \times \mathbf{u} = \Omega$, and the vorticity transport equation, (2.30), with a compact finite difference scheme.

2.2 Boundary Conditions

2.2.1 Velocity Poisson Equations

Boundary conditions for the velocity Poisson equations are readily available. At the inflow boundary, a Dirichlet boundary condition is imposed with a known inlet

velocity profile. A homogeneous Neumann boundary condition may be enforced at the flow exit. On solid walls, no-slip boundary conditions with $\mathbf{b} = \mathbf{0}$ in Equation (2.22) are appropriate.

2.2.2 Vorticity Transport Equations

By specifying the velocity distribution at the inlet, the inflow vorticity distribution is also known. A homogeneous Neumann boundary condition for the vorticity is appropriate at a flow exit. However, careful determination of the vorticity boundary condition at a no-slip wall is critical.

Daube (1992) points out that the requirement to satisfy the continuity equation reduces to the boundary conditions coupling the velocity and vorticity at the solid wall. A convergent solution which is divergence-free has been found by many researchers to be sensitive to the numerical implementation of the "make-up" vorticity kinematic boundary condition. Guj & Stella (1993) indicate that a solenoidal velocity field for an arbitrary vorticity distribution may be assured only by coupling the velocity Poisson equations and the vorticity transport equations. The conservation of mass, and hence the conservation of the solenoidality of vorticity field, are satisfied implicitly in the bulk of the fluid by the kinematic velocity Poisson equation and are to be imposed explicitly on the boundary of the fluid domain by the kinematic vorticity boundary condition.

A new second order vorticity kinematic boundary condition is developed and implemented in the present formulation. Using Equation (2.27), the vorticity at a

no-slip wall is represented by retaining only the normal component of the equation (2.11),

$$\Omega_{wall} = \hat{n} \cdot \nabla_S \times \mathbf{u} \quad (2.31)$$

A schematic of a typical flow geometry is shown in Figure 2.1, where w_0 , w_1 , and w_2 represent computational node points on the wall, next to the wall, and second next to the wall along an axis normal to boundary walls.

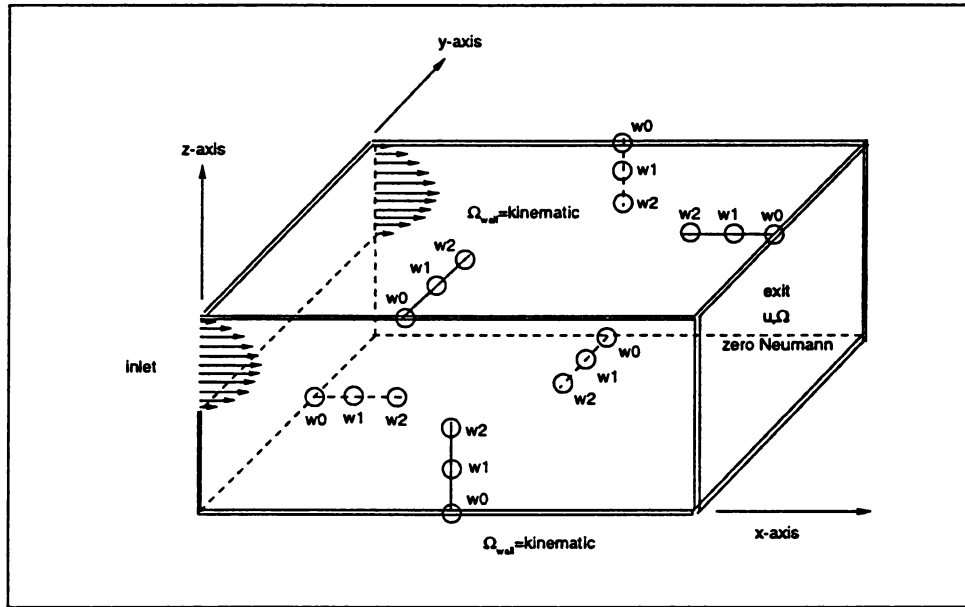


Figure 2.1 : Schematic of kinematic vorticity boundaries

For the wall normal to the x-axis, from Equation (2.31), and (2.11), the vorticity components at the wall are

$$\Omega_x = 0 ; \quad \Omega_y = -\frac{\partial w}{\partial x} ; \quad \Omega_z = \frac{\partial v}{\partial x} \quad (2.32)$$

For the wall normal to the y-axis, the vorticity components are

$$\Omega_x = \frac{\partial w}{\partial y} ; \Omega_y = 0 ; \Omega_z = -\frac{\partial u}{\partial y} \quad (2.33)$$

For the wall normal to the z-axis, the vorticity components are

$$\Omega_x = -\frac{\partial v}{\partial z} ; \Omega_y = \frac{\partial u}{\partial z} ; \Omega_z = 0 \quad (2.34)$$

Various orders for interpolation of the kinematic wall equations for vorticity were presented and studied by Agarwal (1981). They are the first order equation,

$$\Omega_{wall} = \frac{(q_{wall+1} - q_{wall})}{\Delta h} + \mathcal{O}(\Delta z) \quad (2.35)$$

the second order equation,

$$\Omega_{wall} = \frac{(-q_{wall+2} + 4q_{wall+1} - 3q_{wall})}{\Delta 2h} + \mathcal{O}(\Delta z)^2 \quad (2.36)$$

the third order equation,

$$\Omega_{wall} = \frac{(2q_{wall+3} - 9q_{wall+2} + 18q_{wall+1} - 11q_{wall})}{\Delta 6h} + \mathcal{O}(\Delta z)^3 \quad (2.37)$$

and the Wood's formula (Roache, 1972),

$$\Omega_{wall} = \frac{2(q_{wall+1} - q_{wall})}{\Delta h^2} - \frac{\Omega_{wall+1}}{2} + \mathcal{O}(\Delta z)^2 \quad (2.38)$$

where Ω denotes a vorticity component, q denotes a velocity component, and Δh is the grid size.

Equations (2.35) and (2.38) were tested in the present finite element framework on a fully developed two dimensional channel. Neither kinematic vorticity boundary conditions was able to produce a correct solution. Using a bilinear finite element interpolation, Equation (2.36) and (2.37) will put an entry outside the banded structure of

the terminal jacobian matrix. Thus, Equation (2.37), as implemented in conjunction with a biquadratic finite element interpolation, was found to be inappropriate.

A fairly different kinematic vorticity boundary enforcement was presented by Guevremont et al. (1990). Instead of using the normal component of the vorticity definition at wall, which acts on a line, a finite element volume integral weak formulation for the vorticity definition is used,

$$\int_V N_\Omega (\Omega - \nabla \times \mathbf{u}) dV = 0 \quad (2.39)$$

where N is the finite element basis for Ω . This equation was not shown to enforce continuity of the flow field. In the case of an open channel flow, mass conservation is accounted for by using the continuity equation to modify the surface integral of the normal velocity equation at an exit (Guevremont et al. 1990).

2.2.3 Present Vorticity Kinematic Equations

The kinematic vorticity boundary condition is imposed at the end of the finite element assembly process. A vorticity kinematic equation will then replace the discretized nodal vorticity equation at the corresponding node on a no-slip wall. These vorticity kinematic equations are obtained by using a Taylor series expansion of Equations (2.31). A detailed description of derivation for one of the equations follows.

Using Taylor series expansion for the x-velocity component, u , at wall normal to the z-axis,

$$u_{w1} = u_{wo} + \Delta z \frac{du_{w0}}{dz} + \frac{\Delta z^2}{2} \frac{d^2 u_{w0}}{dz^2} + \mathcal{O}(\Delta z)^3 \quad (2.40)$$

Using Equation (2.30)

$$u_{w1} = u_{w0} + \Delta z \Omega_{y,w0} + \frac{\Delta z^2}{2} \frac{d\Omega_{y,w0}}{dz} + \mathcal{O}(\Delta z)^3 \quad (2.41)$$

Expanding $\Omega_{y,w1}$ in terms of $\Omega_{y,w0}$,

$$\Omega_{y,w1} = \Omega_{y,w0} + \Delta z \frac{d\Omega_{y,w0}}{dz} + \mathcal{O}(\Delta z)^2 \quad (2.42)$$

Combining Equations (2.41) and (2.42),

$$u_{w1} = u_{w0} + \Delta z \Omega_y + \frac{\Delta z^2}{2} \left(\frac{\Omega_{y,w1} - \Omega_{y,w0}}{\Delta z} \right) + \mathcal{O}(\Delta z)^3 \quad (2.43)$$

the result is

$$(\Omega_{y,w1} + \Omega_{y,w0}) - \frac{2}{\Delta z} (u_{w1} - u_{w0}) + \mathcal{O}(\Delta z)^2 = 0 \quad (2.44)$$

For the wall normal parallel to the positive x-axis, the resultant vorticity kinematic equations are,

$$\Omega_{x,w0} = 0 \quad (2.45)$$

$$(\Omega_{y,w1} + \Omega_{y,w0}) + \frac{2}{\Delta x} (w_{w1} - w_{w0}) = 0 \quad (2.46)$$

$$(\Omega_{z,w1} + \Omega_{z,w0}) - \frac{2}{\Delta x} (v_{w1} - v_{w0}) = 0 \quad (2.47)$$

For the wall normal parallel to the negative x-axis, the vorticity kinematic equations are,

$$\Omega_{x,w0} = 0 \quad (2.48)$$

$$(\Omega_{y,w1} + \Omega_{y,w0}) - \frac{2}{\Delta x} (w_{w1} - w_{w0}) = 0 \quad (2.49)$$

$$(\Omega_{z,w1} + \Omega_{z,w0}) + \frac{2}{\Delta x}(v_{w1} - v_{w0}) = 0 \quad (2.50)$$

Derivation for the other two derivatives follows this process. For the wall normal parallel to the positive y-axis, the vorticity kinematic equations are,

$$(\Omega_{x,w1} + \Omega_{x,w0}) - \frac{2}{\Delta y}(w_{w1} - w_{w0}) = 0 \quad (2.51)$$

$$\Omega_{y,w0} = 0 \quad (2.52)$$

$$(\Omega_{z,w1} + \Omega_{z,w0}) + \frac{2}{\Delta y}(u_{w1} - u_{w0}) = 0 \quad (2.53)$$

For the wall normal parallel to the negative y-axis, the vorticity kinematic equations are,

$$(\Omega_{x,w1} + \Omega_{x,w0}) + \frac{2}{\Delta y}(w_{w1} - w_{w0}) = 0 \quad (2.54)$$

$$\Omega_{y,w0} = 0 \quad (2.55)$$

$$(\Omega_{z,w1} + \Omega_{z,w0}) - \frac{2}{\Delta y}(u_{w1} - u_{w0}) = 0 \quad (2.56)$$

For the wall normal parallel to the positive z-axis, the vorticity kinematic equations are,

$$(\Omega_{x,w1} + \Omega_{x,w0}) + \frac{2}{\Delta z}(v_{w1} - v_{w0}) = 0 \quad (2.57)$$

$$(\Omega_{y,w1} + \Omega_{y,w0}) - \frac{2}{\Delta z}(u_{w1} - u_{w0}) = 0 \quad (2.58)$$

$$\Omega_{z,w0} = 0 \quad (2.59)$$

For the wall normal parallel to the positive z-axis, the vorticity kinematic equations are,

$$(\Omega_{x,w1} + \Omega_{x,w0}) + \frac{2}{\Delta z}(v_{w1} - v_{w0}) = 0 \quad (2.60)$$

$$(\Omega_{y,w1} + \Omega_{y,w0}) - \frac{2}{\Delta z}(u_{w1} - u_{w0}) = 0 \quad (2.61)$$

$$\Omega_{z,w0} = 0 \quad (2.62)$$

A higher order kinematic vorticity equation is also derived when the biquadratic finite element basis is used for both velocity and vorticity. It is important to derive a compatible order of vorticity kinematic boundary condition that utilizes the entire element nodal point map in the normal direction. Using the Taylor expansion again for the vorticity component on the bottom wall, see Figure 2.1,

$$u_{w1} = u_{w0} + \Delta z \frac{du}{dz} + \frac{\Delta z^2}{2} \frac{d^2u}{dz^2} + \frac{\Delta z^3}{6} \frac{d^3u}{dz^3} + \mathcal{O}(\Delta z)^4 \quad (2.63)$$

$$u_{w2} = u_{w0} + 2\Delta z \frac{du}{dz} + \frac{(2\Delta z)^2}{2} \frac{d^2u}{dz^2} + \frac{(2\Delta z)^3}{6} \frac{d^3u}{dz^3} + \mathcal{O}(\Delta z)^4 \quad (2.64)$$

Using Equation (2.30),

$$u_{w1} = u_{w0} + \Delta z \Omega_{y,w0} + \frac{\Delta z^2}{2} \frac{d\Omega_{y,w0}}{dz} + \frac{\Delta z^3}{6} \frac{d^2\Omega_{y,w0}}{dz^2} + \mathcal{O}(\Delta z)^4 \quad (2.65)$$

$$u_{w2} = u_{w0} + 2\Delta z \Omega_{y,w0} + \frac{(2\Delta z)^2}{2} \frac{d\Omega_{y,w0}}{dz} + \frac{(2\Delta z)^3}{6} \frac{d^2\Omega_{y,w0}}{dz^2} + \mathcal{O}(\Delta z)^4 \quad (2.66)$$

Eliminating the $\mathcal{O}(\Delta z)^3$ term from Equations (2.65) and (2.66),

$$(8u_{w1} - u_{w2}) = 7u_{w0} + 6\Delta z \Omega_{y,w0} + 2\Delta z^2 \frac{d^2\Omega_{y,w0}}{dz^2} + \mathcal{O}(\Delta z)^4 \quad (2.67)$$

Expanding $\Omega_{y,w2}$ and $\Omega_{y,w1}$ in terms of $\Omega_{y,w0}$,

$$\Omega_{y,w1} = \Omega_{y,w0} + \Delta z \frac{d\Omega_{y,w0}}{dz} + \frac{\Delta z^2}{2} \frac{d^2\Omega_{y,w0}}{dz^2} + \mathcal{O}(\Delta z)^3 \quad (2.68)$$

$$\Omega_{y,w1} = \Omega_{y,w0} + 2\Delta z \frac{d\Omega_{y,w0}}{dz} + \frac{2(\Delta z)^2}{2} \frac{d^2\Omega_{y,w0}}{dz^2} + \mathcal{O}(\Delta z)^3 \quad (2.69)$$

Eliminating the $\mathcal{O}(\Delta z)^2$ term from Equations (2.68) and (2.69),

$$(4\Omega_{y,w1} - \Omega_{y,w2}) = 3\Omega_{y,w0} + 2\Delta z \frac{d\Omega_{y,w0}}{dz} + \mathcal{O}(\Delta z)^3 \quad (2.70)$$

Combining Equations (2.67) and (2.70),

$$(8u_{w1} - u_{w2}) = 7u_{w0} + 6\Delta z \Omega_{y,w0} + \Delta z (4\Omega_{y,w1} - \Omega_{y,w2} - 3\Omega_{y,w0}) + \mathcal{O}(\Delta z)^4 \quad (2.71)$$

The result is

$$(3\Omega_{y,w0} + 4\Omega_{y,w1} - \Omega_{y,w2}) + \frac{1}{\Delta z} (7u_{w0} - 8u_{w1} - u_{w2}) + \mathcal{O}(\Delta z)^3 = 0 \quad (2.72)$$

Similar kinematic vorticity equations are developed and used for each computational node on various solid walls. These third order equations are tested in two dimensional channel flow with a biquadratic finite element basis function for velocity and vorticity. Accurate divergence free velocity solutions were obtained. However, an unacceptable solution is found when a second order type equations, e.g. Equation (2.44), is implemented in conjunction with a biquadratic finite element basis function.

PARALLEL FINITE ELEMENT ALGORITHM

3.1 Finite Element Formulation

The three dimensional velocity-vorticity formulation for the laminar incompressible Navier-Stokes equations in a cartesian coordinate system with gravity acting in the z (vertical) direction is,

$$\mathcal{L}(u) = -\nabla^2 u + \frac{\partial \Omega_y}{\partial z} - \frac{\partial \Omega_z}{\partial y} = 0 \quad (3.1)$$

$$\mathcal{L}(v) = -\nabla^2 v + \frac{\partial \Omega_z}{\partial x} - \frac{\partial \Omega_x}{\partial z} = 0 \quad (3.2)$$

$$\mathcal{L}(w) = -\nabla^2 w + \frac{\partial \Omega_x}{\partial y} - \frac{\partial \Omega_y}{\partial x} = 0 \quad (3.3)$$

$$\mathcal{L}(\Omega_x) = \frac{\partial \Omega_x}{\partial t} + (\mathbf{u} \cdot \nabla) \Omega_x - (\boldsymbol{\Omega} \cdot \nabla) u - \frac{1}{Re} \nabla^2 \Omega_x - \frac{Gr}{Re^2} \frac{\partial \Theta}{\partial y} = 0 \quad (3.4)$$

$$\mathcal{L}(\Omega_y) = \frac{\partial \Omega_y}{\partial t} + (\mathbf{u} \cdot \nabla) \Omega_y - (\boldsymbol{\Omega} \cdot \nabla) v - \frac{1}{Re} \nabla^2 \Omega_y + \frac{Gr}{Re^2} \frac{\partial \Theta}{\partial x} = 0 \quad (3.5)$$

$$\mathcal{L}(\Omega_z) = \frac{\partial \Omega_z}{\partial t} + (\mathbf{u} \cdot \nabla) \Omega_z - (\boldsymbol{\Omega} \cdot \nabla) w - \frac{1}{Re} \nabla^2 \Omega_z = 0 \quad (3.6)$$

$$\mathcal{L}(\Theta) = \frac{\partial \Theta}{\partial t} + (\mathbf{u} \cdot \nabla) \Theta - \frac{1}{RePr} \nabla^2 \Theta = 0 \quad (3.7)$$

3.1.1 Galerkin Weak Statement

The semi-discrete finite element approximation assumes

$$\sigma \approx \sigma^h \equiv \bigcup \sigma_e$$

where σ denotes the continuum physical domain of $\mathcal{L}(\mathbf{q})$, Equations (3.1)-(3.7), and σ^h denotes the discretization of the continuum domain as the union of σ_e , the individual finite element sub-domains. Each member of the state variable

$$\mathbf{q}(\mathbf{x}, t) = \{u, v, w, \Omega_x, \Omega_y, \Omega_z, \Theta\}$$

within an element is written in terms of the spatial finite element basis functions, also called the trial functions or the element shape functions, $\{N_k(\mathbf{x})\}$, and time dependent nodal expansion coefficients as

$$\mathbf{u}_e(\mathbf{x}, t) \equiv \{N_k(\mathbf{x})\}^T \{\mathbf{U}(t)\}_e \quad (3.8)$$

$$\Omega_e(\mathbf{x}, t) \equiv \{N_k(\mathbf{x})\}^T \{\mathbf{O}(t)\}_e \quad (3.9)$$

$$\Theta_e(\mathbf{x}, t) \equiv \{N_k(\mathbf{x})\}^T \{T(t)\}_e \quad (3.10)$$

$$\mathbf{q} \approx \mathbf{q}^h \equiv \bigcup \mathbf{q}_e ; \quad \mathbf{q} \equiv (\mathbf{u}, \Omega, \Theta) \quad (3.11)$$

Here k is the polynomial degree of interpolation for the basis function, $\mathbf{U}(t) = (U, V, W)$ is the nodal velocity vector array, $\mathbf{O}(t) = (O_x, O_y, O_z)$ is the nodal vorticity vector, and $T(t)$ is the nodal temperature.

The finite element semidiscretization of a (Galerkin) weak statement is obtained by determining the stationary point of the residual of the velocity-vorticity formulation, Equations (3.1) - (3.7), over the domain, σ , for arbitrary choice of a weight

function (or test function). For a Galerkin statement, the weight function is chosen to be identical to the trial function, to within some boundary condition details, which minimizes the approximation error in the sense of distributions for the discretized state variable. The Galerkin finite element weak statement for the velocity-vorticity formulation is thus

$$GWS(\mathbf{q}^h) \equiv \{F\mathbf{Q}\} = \mathcal{S}_{e=1}^M \left(\int_{\sigma^h} \{N_k(\mathbf{x})\} \mathcal{L}(\mathbf{q}^h) dV \right) = \{0\} \quad (3.12)$$

where $\mathbf{Q} \equiv (\mathbf{U}, \mathbf{O}, T)$, \mathcal{S} denotes the finite element assembly procedure carrying local element entries into the global matrix array, and M is the number of elements in the discretized domain.

3.1.2 Discrete Finite Element Weak Statement For The Velocity

Writing $\{N_k(\mathbf{x})\} \equiv \{N\}$, for simplicity the Galerkin weak statement for $\mathcal{L}(u)$ is

$$GWS(u^h) \equiv \{FU\} = \mathcal{S}_{e=1}^M \left(\int_{\sigma^h} \{N\} \left(-\nabla^2 u^h + \frac{\partial \Omega_y^h}{\partial z} - \frac{\partial \Omega_z^h}{\partial y} \right) dV \right) = 0 \quad (3.13)$$

Applying Green's theorem, and recognizing that all interior generated surface integrals vanish identically yields

$$\begin{aligned} \{FU\} = \mathcal{S}_{e=1}^M \left(\int_{\sigma^h} \nabla \{N\} \cdot \nabla u^h + \{N\} \left(\frac{\partial \Omega_y^h}{\partial z} - \frac{\partial \Omega_z^h}{\partial y} \right) dV \right. \\ \left. - \oint_{\partial\sigma \cap \partial\sigma_e} \{N\} (\hat{\mathbf{n}} \cdot \nabla u^h) d\tau \right) = 0 \end{aligned} \quad (3.14)$$

where $\partial\sigma$ is the boundary of σ . The surface integral in Equation (3.14), will be zero for any homogenous Neumann boundary condition. Substituting finite element basis function forms for u^h and Ω^h , then

$$\{FU\} = \mathcal{S}_{e=1}^M \{FU\}_e \quad (3.15)$$

where

$$\begin{aligned} \{FU\}_e = & \int_{\sigma_e} \nabla\{N\} \cdot \nabla\{N\}^T \{U\}_e dV + \int_{\sigma_e} \{N\} \frac{\partial\{N\}^T}{\partial z} \{O_y\}_e dV \\ & - \int_{\sigma_e} \{N\} \frac{\partial\{N\}^T}{\partial y} \{O_z\}_e dV = 0 \end{aligned} \quad (3.16)$$

Following the compact finite element notation given by Baker (1983), the discretized finite element weak statements for the state variable velocity resolution are,

$$\{FU\}_e = [c2kke]\{U\}_e + [c20ze]\{O_y\}_e - [c20ye]\{O_z\}_e = 0 \quad (3.17)$$

$$\{FV\}_e = [c2kke]\{V\}_e + [c20xe]\{O_z\}_e - [c20ze]\{O_x\}_e = 0 \quad (3.18)$$

$$\{FW\}_e = [c2kke]\{W\}_e + [c20ye]\{O_x\}_e - [c20xe]\{O_y\}_e = 0 \quad (3.19)$$

where

$$[c2kke] = \int_{\sigma_e} \nabla\{N\} \cdot \nabla\{N\}^T dV \quad (3.20)$$

$$[c20xe] = \int_{\sigma_e} \{N\} \frac{\partial\{N\}^T}{\partial x} dV \quad (3.21)$$

$$[c20ye] = \int_{\sigma_e} \{N\} \frac{\partial\{N\}^T}{\partial y} dV \quad (3.22)$$

$$[c20ze] = \int_{\sigma_e} \{N\} \frac{\partial\{N\}^T}{\partial z} dV \quad (3.23)$$

3.1.3 Discretized Finite Element Weak Statements For The Vorticity

The Galerkin weak statement for $\mathcal{L}(\Omega_x)$ is

$$\begin{aligned} GWS(\Omega_x^h) \equiv \{FO_x\} = & S_{e=1}^M \left(\int_{\sigma^h} \{N\} \left(\frac{\partial\Omega_x^h}{\partial t} + (\mathbf{u}^h \cdot \nabla)\Omega_x^h - (\Omega^h \cdot \nabla)\mathbf{u}^h \right. \right. \\ & \left. \left. - \frac{1}{Re} \nabla^2 \Omega_x^h - \frac{Gr}{Re^2} \frac{\partial\Theta^h}{\partial y} \right) dV \right) = 0 \end{aligned} \quad (3.24)$$

Applying Green's theorem, dropping the surface integral, and using Equations (3.8)

- (3.10), then yields

$$\{FO_x\} = S_{e=1}^M \{FO_x\}_e \quad (3.25)$$

where

$$\begin{aligned} \{FO_x\}_e &= \int_{\sigma_e} \{N\} \{N\}^T \frac{\partial \{O_x\}}{\partial t} dV \\ &+ \int_{\sigma_e} \{N\} \left[\frac{\partial \{N\}^T}{\partial x}, \frac{\partial \{N\}^T}{\partial y}, \frac{\partial \{N\}^T}{\partial z} \right] \cdot [\{U\}, \{V\}, \{W\}] \{N\}^T \{O_x\} dV \\ &- \int_{\sigma_e} \{N\} \left[\frac{\partial \{N\}^T}{\partial x}, \frac{\partial \{N\}^T}{\partial y}, \frac{\partial \{N\}^T}{\partial z} \right] \cdot [\{O_x\}, \{O_y\}, \{O_z\}] \{N\}^T \{U\} dV \\ &+ \frac{1}{Re} \int_{\sigma_e} \nabla \{N\} \cdot \nabla \{N\}^T \{O_x\} dV \\ &- \frac{Gr}{Re^2} \int_{\sigma_e} \{N\} \frac{\partial \{N\}^T}{\partial y} \{T\} dV \end{aligned} \quad (3.26)$$

The discretized finite element weak statements for the discretized vorticity resolution are,

$$\begin{aligned} \{FO_x\}_e &= [c200e] \frac{d\{O_x\}_e}{dt} \\ &+ (\{U\}_e^T [c30x0e] + \{V\}_e^T [c30y0e] + \{W\}_e^T [c30z0e]) \{O_x\}_e \\ &- (\{O_x\}_e^T [c30x0e] + \{O_y\}_e^T [c30y0e] + \{O_z\}_e^T [c30z0e]) \{U\}_e \\ &+ \frac{1}{Re} [c2kke] \{O_x\}_e - \frac{Gr}{Re^2} [c20ye] \{T\}_e = 0 \end{aligned} \quad (3.27)$$

$$\begin{aligned} \{FO_y\}_e &= [c200e] \frac{d\{O_y\}_e}{dt} \\ &+ (\{U\}_e^T [c30x0e] + \{V\}_e^T [c30y0e] + \{W\}_e^T [c30z0e]) \{O_y\}_e \\ &- (\{O_x\}_e^T [c30x0e] + \{O_y\}_e^T [c30y0e] + \{O_z\}_e^T [c30z0e]) \{V\}_e \\ &+ \frac{1}{Re} [c2kke] \{O_y\}_e + \frac{Gr}{Re^2} [c20xe] \{T\}_e = 0 \end{aligned} \quad (3.28)$$

$$\begin{aligned}
\{FO_z\}_e &= [c200e] \frac{d\{O_z\}_e}{dt} \\
&+ \left(\{U\}_e^T [c30x0e] + \{V\}_e^T [c30y0e] + \{W\}_e^T [c30z0e] \right) \{O_z\}_e \\
&- \left(\{O_x\}_e^T [c30x0e] + \{O_y\}_e^T [c30y0e] + \{O_z\}_e^T [c30z0e] \right) \{W\}_e \\
&+ \frac{1}{Re} [c2kke] \{O_z\}_e = 0
\end{aligned} \tag{3.29}$$

where

$$[c200e] = \int_{\sigma_e} \{N\} \{N\}^T dV \tag{3.30}$$

$$[c30x0e] = \int_{\sigma_e} \{N\} \frac{\partial \{N\}}{\partial x} \{N\}^T dV \tag{3.31}$$

$$[c30y0e] = \int_{\sigma_e} \{N\} \frac{\partial \{N\}}{\partial y} \{N\}^T dV \tag{3.32}$$

$$[c30z0e] = \int_{\sigma_e} \{N\} \frac{\partial \{N\}}{\partial z} \{N\}^T dV \tag{3.33}$$

3.1.4 Discretized Finite Element Weak Statement For The Temperature

The Galerkin weak statement for $\mathcal{L}(\Theta)$ is,

$$\{FT\} = S_{e=1}^M \left(\int_{\sigma^h} \{N\} \left(\frac{\partial \Theta}{\partial t} + (\mathbf{u}^h \cdot \nabla) \Theta^h - \frac{1}{RePr} \nabla^2 \Theta^h \right) dV \right) = 0 \tag{3.34}$$

Applying Green's Theorem and using the Equations (3.8) - (3.10), the discretized finite element weak statement for the temperature is,

$$\begin{aligned}
\{FT\}_e &= [c200e] \frac{d\{T\}_e}{dt} \\
&+ \left(\{U\}_e^T [c30x0e] + \{V\}_e^T [c30y0e] + \{W\}_e^T [c30z0e] \right) \{T\}_e \\
&+ \frac{1}{RePr} [c2kke] \{T\}_e = 0
\end{aligned} \tag{3.35}$$

3.1.5 Newton Statement

The semi-discretized finite element velocity-vorticity weak statements form the following system of algebraic-ordinary differential equations,

$$\{FU\} = \{RU\} = 0 \quad (3.36)$$

$$\{FO\} = [M] \frac{dO}{dt} + \{RO\} = 0 \quad (3.37)$$

$$\{FT\} = [M] \frac{dT}{dt} + \{RT\} = 0 \quad (3.38)$$

where $[M]$ is the global assembled mass matrix associated with the time term and $\{RQ\}$ is the global steady state residual vectors which represents all other weak statement terms. Equations (3.37) and (3.38) are a system of ordinary differential equation, for which the θ_o - implicit, one step Euler scheme is used to integrate through the transient solution. The terminal computable algebraic statements are then,

$$\{FU\} = \{RU\} = [D]\{U\} + \{S(O)\} = 0 \quad (3.39)$$

$$\{FO\} = [M]\{O_{n+1} - O_n\} + \Delta t (\theta_o \{RO\}_{n+1} + (1 - \theta_o) \{RO\}_n) = 0 \quad (3.40)$$

$$\{FT\} = [M]\{T_{n+1} - T_n\} + \Delta t (\theta_o \{RT\}_{n+1} + (1 - \theta_o) \{RT\}_n) = 0 \quad (3.41)$$

where n denotes the time station t_n , $t_{n+1} = t_n + \Delta t$, $\{S(O)\}$ is the source term contribution from vorticity, $\theta_o = 0.5$ represents the trapezoidal rule, while $\theta_o = 1$ represents the full implicit backward Euler integration scheme.

The residual vector $\{RQ\}$ is a nonlinear function of the state variables, U , O , and T . Equations (3.39) - (3.41) represent a coupled, nonlinear system of algebraic equations that must be solved iteratively. The Newton-Raphson algorithm is used and the subsequent procedure is,

$$\{Q\}_{n+1}^0 = \{Q\}_n \quad ; \quad Q \equiv (U, V, W, O_x, O_y, O_z, T)$$

$$\{FQ\}_{n+1}^0 = \{FQ\}_n$$

for $p = 0, 1, 2, \dots$ until convergence $n + 1$

$$[JAC] = [JQQ] = \left[\frac{\partial \{FQ\}}{\partial \{Q\}} \right]_{n+1}^p$$

$$[JAC] \{\delta Q\}_{n+1}^{p+1} = -\{FQ\}_{n+1}^p$$

$$\{Q\}_{n+1}^{p+1} = \{Q\}_{n+1}^p + \{\delta Q\}_{n+1}^{p+1}$$

where $[JAC]$ is the jacobian matrix of the terminal non-linear algebraic statement.

3.1.6 Jacobian Matrix

The residual equations, (3.39) - (3.41), are differentiated with respect to the state variables to produce the Newton jacobian. The resultant 7x7 block matrix is shown in Figure 3.1, each term in which is formed via differentiation of $\{FQ\}$ on the element domain σ_e , then assembled to global form.

JUU	0	0	0	JUO_y	JUO_z	0
0	JVV	0	JVO_x	0	JVO_z	0
0	0	JUW	JUO_x	JUO_y	0	0
JO_xU	JO_xV	JO_xW	JO_xO_x	JO_xO_y	JO_xO_z	JO_xT
JO_yU	JO_yV	JO_yW	JO_yO_x	JO_yO_y	JO_yO_z	JO_yT
JO_zU	JO_zV	JO_zW	JO_zO_x	JO_zO_y	JO_zO_z	0
JTU	JTV	JTW	0	0	0	JTT

Figure 3.1 : Layout of jacobian matrix

The element contributions to the global block sparse matrix are:

$$[JUU] = [JVV] = [JWW] = [c2kke]$$

$$[JVO_y] = -[JWO_x] = [c20xe]$$

$$[JWO_x] = -[JUO_z] = [c20ye]$$

$$[JUO_y] = -[JVO_x] = [c20ze]$$

$$[JO_xU] = \theta_o \Delta t \left[\{O_x\}_e^T [c30x0e] - [vconv] \right]$$

$$[JO_yV] = \theta_o \Delta t \left[\{O_y\}_e^T [c30y0e] - [vconv] \right]$$

$$[JO_zW] = \theta_o \Delta t \left[\{O_z\}_e^T [c30z0e] - [vconv] \right]$$

$$[JO_jU_k] = \theta_o \Delta t \{O_j\}_e^T [c30k0e]$$

$$[JO_xO_x] = [c200e] + \theta_o \Delta t \left[\{U\}_e^T [c30x0e] + [uconv] \right] + \frac{1}{Re} [c2kke]$$

$$[JO_yO_y] = [c200e] + \theta_o \Delta t \left[\{V\}_e^T [c30y0e] + [uconv] \right] + \frac{1}{Re} [c2kke]$$

$$[JO_zO_z] = [c200e] + \theta_o\Delta t[\{W\}_e^T[c30z0e] + [uconv] + \frac{1}{Re}[c2kke]]$$

$$[JO_jO_k] = \theta_o\Delta t\{U_j\}_e^T[c30k0e]$$

$$[JO_xT] = -\theta_o\Delta t\frac{Gr}{Re^2}[c20ye]$$

$$[JO_yT] = \theta_o\Delta t\frac{Gr}{Re^2}[c20xe]$$

$$[JTT] = [c200e] + \theta_o\Delta t[+[uconv] + \frac{1}{RePr}[c2kke]]$$

$$[JTU_k] = \theta_o\Delta t\{T\}_e^T[c30k0e]$$

where

$$k, j \equiv (x, y, z), U_x = U, U_y = V, U_z = W$$

$$[uconv] = -[\{U\}_e^T[c30x0e] + \{V\}_e^T[c30y0e] + \{W\}_e^T[c30z0e]]$$

$$[vconv] = -[\{O_x\}_e^T[c30x0e] + \{O_y\}_e^T[c30y0e] + \{O_z\}_e^T[c30z0e]]$$

3.1.7 Vorticity Kinematic Boundary Conditions

The general form of the vorticity kinematic equations (2.45)-(2.62) are,

$$\begin{aligned} f(\Omega_{wall}) &= a_{x0}\Omega_{x,w0} + a_{x1}\Omega_{x,w1} + a_{y0}\Omega_{y,w0} + a_{y1}\Omega_{y,w1} \\ &\quad + a_{z0}\Omega_{z,w0} + a_{z1}\Omega_{z,w1} + b_{x0}u_{w0} + b_{x1}u_{w1} \\ &\quad + b_{y0}v_{w0} + b_{y1}v_{w1} + b_{z0}w_{w0} + b_{z1}w_{w1} = 0 \end{aligned} \quad (3.42)$$

For example, Equation (2.44), the coefficients are

$$a_{x0} = a_{x1} = a_{z0} = a_{z1} = b_{y0} = b_{y1} = b_{z0} = b_{z1} = 0$$

$$a_{y0} = a_{y1} = 1 \quad b_{x0} = \frac{2}{\Delta z} \quad b_{x1} = -\frac{2}{\Delta z}$$

To apply the kinematic vorticity boundary conditions, the residual vorticity equations at the boundary nodes where no-slip velocity are imposed are replaced by the vorticity kinematic equations, $f(\Omega_{wall})$. At the same time for the jacobian matrix, the entire row of the corresponding vorticity component at the boundary node is zeroed out and replaced by the coefficients a and b of $f(\Omega_{wall})$ at the corresponding entries, Figure 3.2.

(q)	(u)	(v)	(w)	(Ω_x)	(Ω_y)	(Ω_z)
(...)
(Ω_y)	$JO_y U$	$JO_y V$	$JO_y W$	$JO_y O_x$	$JO_y O_y$	$JO_y O_z$
($\Omega_{y,wall}$)	$0 \dots, b_{x0}, b_{x1}, \dots 0$	$\dots 0 \dots$	$\dots 0 \dots$	$\dots 0 \dots$	$0 \dots, a_{y0}, a_{y1}, \dots 0$	$\dots 0 \dots$
(Ω_y)	$JO_y U$	$JO_y V$	$JO_y W$	$JO_y O_x$	$JO_y O_y$	$JO_y O_z$
(...)

Figure 3.2 : Imposing kinematic vorticity equation in jacobian matrix

PARALLEL COMPUTATIONAL PROCEDURE

Solving the Newton system, Equation (3.39)-(3.41), for the fully coupled velocity-vorticity FE weak statement in three dimensions requires enormous computer time and memory. In a parallel computing environment, the burden of excessive functional execution and memory storage may be distributed over a number of processors. The finite element method, which involves a large amount of computations among individual elements, can be done efficiently in parallel. If Gaussian elimination is employed for the solution of the terminal global matrix statement, the degree of parallelism will deteriorate. Alternatively, the new class of CG-like sparse iterative solvers, such as CGS (Sonneveld 1989) and BiCGSTAB (Van der Vorst 1992), is parallel-efficient, hence implemented for this project. These solvers mainly involve operations of matrix-vector product and vector inner product, hence are ideal for parallel implementation, provided suitable preconditioning can be determined. Details of the developed parallel computational procedure are presented in this chapter.

4.1 CM5 Parallel Computer

The Thinking Machines CM5 at the University of Tennessee (Department of Computer Science) has 32 parallel processing nodes and one front-end control pro-

cessor(host). Each node is a 32 Mhz Sparc processor equipped with 4 vector units. There is a full 32Mbytes of memory, four banks of 8 Mbytes each, in each processor. Processing nodes are connected in a three level fat tree network.

The CM5 computer may be programmed as a single instruction multiple data (SIMD) machine or as a multiple instruction multiple data (MIMD) machine (*CM5 User's Guide, Version 1.0* 1993). SIMD programming requires implementation in CM Fortran which distributes data elements of arrays evenly over processors. MIMD programming uses CMMD message passage subroutines to exchange necessary information between functional units within a parallel program. These functional units are assigned to specific processors by the programmer.

The MIMD programming technique is used in the present methodology. A host-node program is constructed in standard Fortran 77, in conjunction with CMMD library subroutines for message passing. Input data are processed at the host and passed along to corresponding processing nodes to initiate execution. Primary computations are carried out concurrently at all processing nodes. Results are then gathered at the host and output for postprocessing. Vector units of the CM5 were not utilized for the purpose of portability. The floating-point performance of the scalar unit of each node is approximately 5 Mflops.

4.2 Parallel Strategy

The major concern in designing a MIMD type program is the cost of communication. Message passing is fairly expensive compared to floating-point computation

within processors. The other factor is data allocation across processors. The developed program is tailored to minimize both communications and data memories over processors. A typical finite element maximum mesh size of $M=32 \times 32 \times 32$ is used to match the CM5 number of processors. The natural arrangement is to subdivide the computational mesh into 32 subdomains along one direction and assign each one to a processor. A schematic of the computational grid with some major parameters is shown in Figure 4.1. The same programming instructions will be executed simultaneously on processors, however data for computation at each processor will be different. The sequence of computation is summarized as follows:

**** HOST:** preprocess input data, pass to corresponding nodes

*** NODE:** receive input data from host

1. start time integration loop
2. start Newton iteration loop
3. do $i=1, nelemp$
 - form element jacobian matrix contributions, $[JQQ]_e$
 - form residual right hand side, $\{FQ\}_e$
 - assemble to form $[JQQ]$
 - assemble to form $\{FQ\}$
4. apply Dirichlet boundary condition
5. apply vorticity kinematic boundary condition
6. solve : CGS, BiCGSTAB

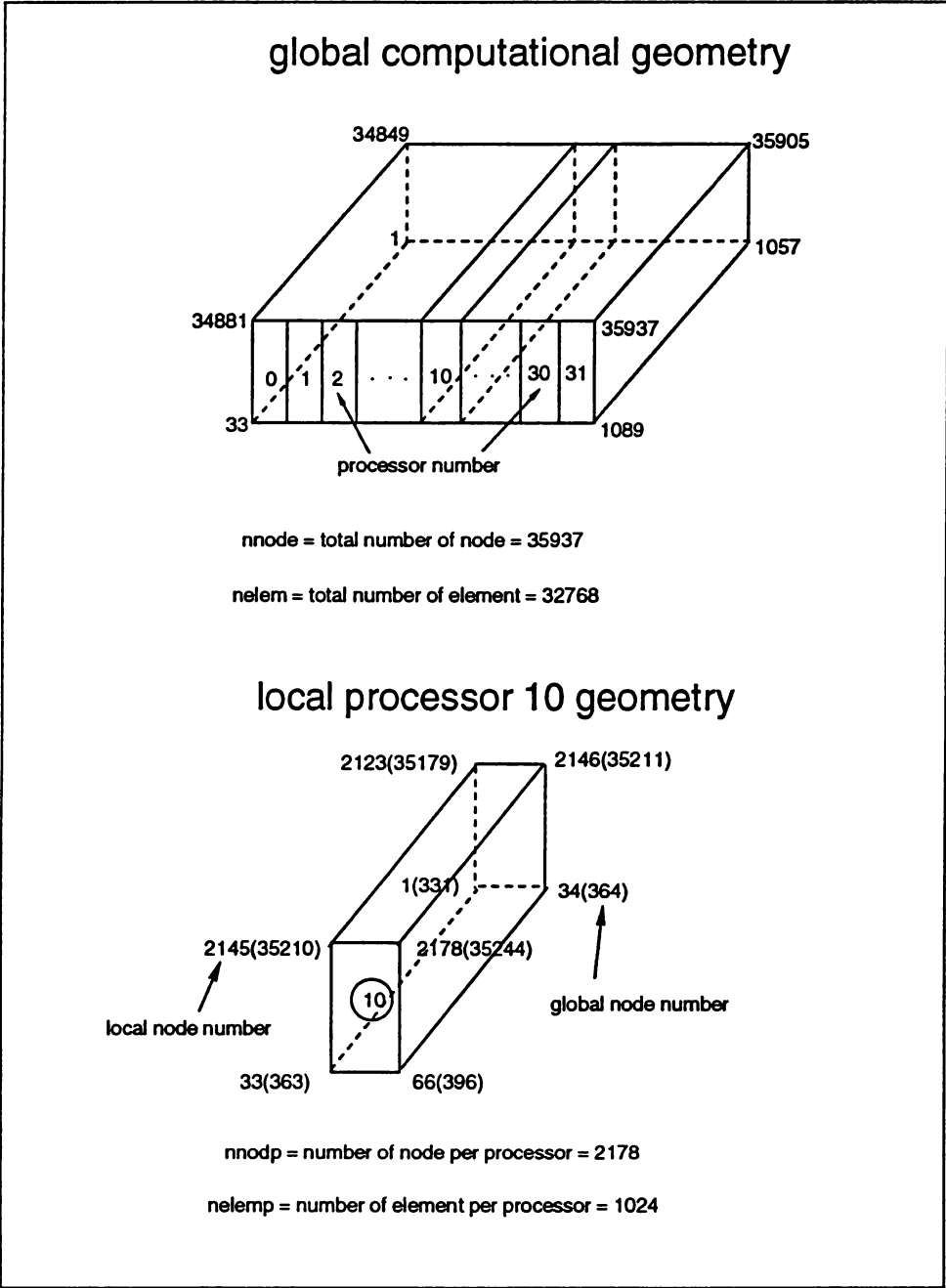


Figure 4.1 : A schematic of the computational domain

- diagonal scaling preconditioner (Barrett et al, 1994)
- vector inner product (message passing)
- matrix vector product (message passing)

* **NODE**: send solution to host

** **HOST**: gather nodal data and output printout

4.3 Sparse Matrix Data Structure

The terminal Newton jacobian for the fully coupled velocity-vorticity formulation is a 7x7 block matrix. Each jacobian block is a sparse matrix with a large number of zeros. A compressed row storage scheme is used to minimize the memory storage. Nonzeros of the jacobian matrix are stored in a floating-point vector, **SMXY**, in a contiguous rowwise order. Two addressing integer vectors hold information for the location of a element entry of the matrix. The first vector, **JC**, stores the column indices of element entries and has the same length as **SMXY**. The second one, **NR**, contains pointers to positions where each new row begins. An example of a 4x4 sparse matrix as follows,

$$[A] = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ a_{41} & 0 & a_{43} & a_{44} \end{bmatrix} \Rightarrow \{SMXY\} = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{32} \\ a_{33} \\ a_{34} \\ a_{41} \\ a_{43} \\ a_{44} \end{bmatrix} ; \{JC\} = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 1 \\ 2 \\ 3 \\ 2 \\ 3 \\ 4 \\ 1 \\ 3 \\ 4 \end{bmatrix} ; \{NR\} = \begin{bmatrix} 1 \\ 4 \\ 7 \\ 10 \\ 13 \end{bmatrix}$$

The integer pointer arrays, JC and NR, are constructed and stored in each processor. Since the structure of every jacobian matrix block is the same, only one set of JC and NR is needed. A total of twenty seven SMXY vectors are required for the isothermal fully coupled velocity-vorticity equations system. Six more are needed when the energy equation is added. The vector length of SMXY is 37637 for M=32x32x32. Thirty three SMXY vectors require approximately 10 MBytes of memory. A total of 14 Mbytes of memory per node is needed for the developed program.

The pseudocode for the matrix vector product $x = Ar$, using compressed row storage format, can be written as,

```
** for i = 1 , n
```

```
  x(i) = 0
```

```
  for j = NR(i) , NR(i+1) -1
```

$$x(i) = x(i) + SMXY(j) * r(JC(j))$$

**** end**

which is a major kernel for the sparse iterative solvers.

4.4 Sparse Iterative Solvers

The CGS and BiCGSTAB sparse iterative solvers are used to compute the solution of $Ax = b$. These solvers are an extension of the conjugate gradient method (Golub & Van Loan 1989), for a symmetric positive definite matrix, to a nonsymmetric matrix. These are Krylov subspace methods (Freund, Golub & Nachtigal 1991) based on the nonsymmetric Lanczos process. Their distinct advantage is that they operate on vector recursions, thus storage requirements per iteration are low and roughly constant.

The major kernel of these solvers is a matrix vector multiplication operation which can be efficiently implemented on a parallel computer. A comprehensive description of various Krylov subspace iterative solvers is given by Barrett et al. (1994).

4.4.1 Conjugate Gradient Squared Method (CGS)

The CGS method was developed by Sonneveld (1989). It is a modification of the BiConjugate Gradients (BiCG) method by Fletcher (1976). In the BiCG method, two sequences of mutually orthogonal residual vectors, r and \tilde{r} , are generated. They are updated along two search directions augmented by multiplication of the matrix,

A , and its transpose, A^T . The idea of the CGS method is to contract the residual vector r twice by the matrix, A , while \tilde{r} remains unchanged. The convergence rate is observed to be twice as fast as that of BiCG. CGS requires about the same number of operations per iteration as BiCG, but does not involve computation of the matrix transpose. The pseudocode for the CGS algorithm (Dias da Cunha & Hopkins 1992) is,

****CGS Algorithm**

- * choose x_0
- * $r_0 = Q(b - Ax_0)$
- * $p_0 = s_0 = r_0 = \tilde{r}_0$
- * $\rho_0 = \tilde{r}_0^T r_0$
- * for $k = 1, 2, 3, \dots$
 - $w_{k-1} = QAp_{k-1}$
 - $\xi_{k-1} = \tilde{r}_0^T w_{k-1}$
 - $\alpha_{k-1} = \rho_{k-1} / \xi_{k-1}$
 - $t_{k-1} = s_{k-1} - \alpha_{k-1} w_{k-1}$
 - $w_{k-1} = s_{k-1} + t_{k-1}$
 - $x_k = x_{k-1} + \alpha_{k-1} w_{k-1}$
 - $r_k = r_{k-1} - \alpha_{k-1} QAw_{k-1}$
 - check stopping criterion
 - $\rho_k = \tilde{r}_0^T r_k$

- $\beta_k = \rho_k / \rho_{k-1}$
- $s_k = r_k + \beta_k t_{k-1}$
- $w_k = t_{k-1} + \beta_k p_{k-1}$
- $p_k = s_k + \beta_k w_k$

****END CGS**

where Q is a preconditioning matrix for A .

4.4.2 BiConjugate Gradient Stabilized Method (BiCGSTAB)

The CGS method has a tendency to give an irregular convergence pattern, BiCGSTAB was derived by Van der Vorst (1992) to give a considerably smoother behavior. The pseudocode for the BiCGSTAB method (Dias da Cunha & Hopkins 1992) is

****BiCGSTAB Algorithm**

- * choose x_0
- * $r_0 = Q(b - Ax_0)$
- * $r_0 = \tilde{r}_0$
- * $p_0 = v_0 = 0$
- * $\rho_0 = \alpha_0 = w_0 = 1$
- * for $k = 1, 2, 3, \dots$
 - $\rho_k = \tilde{r}_0^T r_{k-1}$

- $\beta_k = (\rho_k \alpha_{k-1}) / (\rho_{k-1} w_{k-1})$
- $p_k = r_{k-1} + \beta_k (p_{k-1} - w_{k-1} v_{k-1})$
- $v_k = Q A p_k$
- $\xi_k = \tilde{r}_0^T v_k$
- $\alpha_k = \rho_k / \xi_k$
- $s_k = r_{k-1} - \alpha_k v_k$
- *if* $\|s\| < 10^{-16}$; $x_k = x_{k-1} + \alpha_k p_k$ *stop*
- $t_k = Q A s_k$
- $w_k = (t_k^T s_k) / (t_k^T t_k)$
- $x_k = x_{k-1} + \alpha_k p_k + w_k s_k$
- $r_k = s_k - w_k t_k$
- check stopping criterion

****END BiCGSTAB**

CGS requires two matrix vector products and two vector inner products. BiCGSTAB requires two more vector operations of inner product.

The stopping criterion (Barrett et al. 1994) for these iteration methods is

$$\|r_k\|_\infty < \epsilon \|b\|_\infty \quad (4.1)$$

where ϵ is the error tolerance relative to $\|b\|$, the magnitude of the right hand side residual vector. For the three dimensional cavity flow simulations conducted in this project, BiCGSTAB is used primarily and ϵ is chosen to be 10^{-6} , and $\|b\| \approx 100$ for driven cavity and $\|b\| \approx 10$ for thermal cavity in three dimensions.

4.4.3 Preconditioner

The convergence rate of iterative methods depends on the condition number of the jacobian matrix A . The linear system, $Ax = b$, may be transformed by some suitable choice of preconditioning matrix, Q , such that the condition number of the transform matrix, $A^* = QA$, is improved, and the transformed linear system, $QAx = Qb$, has the same solution as the original system. The ideal but impractical choice of Q is $Q = A^{-1}$; then the condition number of the preconditioned matrix is 1! Applying preconditioning to matrix A incurs extra computations, hence there is a trade-off between the efficiency and convergence of an algorithm.

One of the most popular and effective preconditioning techniques is based on incomplete factorization of the coefficient matrix, A , (Saad 1989). The conventional LU factorization method for solving A directly will generate full triangular matrices L and U because of generated fill-in entries to zero positions of A . The incomplete LU preconditioner is derived such that during the factorization process certain fill-in entries are ignored. However, such a preconditioning strategy will severely down-grad the parallelism of CGS and BiCGSTAB methods. In this project, the simple Jacobi(diagonal scaling) preconditioner, $Q = [diag(A)]^{-1}$, is used. The effect of preconditioning may be small, however it can be efficiently implemented in parallel.

4.5 Matrix Vector Multiplication

The goal is to simulate the result of a global matrix vector product under the framework of the finite element method by a series of local matrix and vector algebraic

steps in conjunction with message passing subroutines over a number of processors. In order to minimize memory storage, non-overlapping regions of subdomains are assigned to processors. A simple example, composed of two, 3-D trilinear basis elements, A and B, illustrates the developed computing procedure.

Assume for simplicity that there are only two processors in the parallel computer, hence element A is assigned to processor 1 and element B is assigned to processor 2. A finite element mesh for the example is shown in Figure 4.2. Global node numbers are used to indicate indices for the finite element assembled matrices. There is only one interface boundary in this example. Nodes 3, 4, 9, and 10 are shared by both elements, hence communication is required along this interface for computation of matrix-vector products between processor 1 and processor 2.

The weak statement matrix and residual vector for element A in processor 1 are

$$[SAXY] = \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 & a_1^4 & a_1^7 & a_1^8 & a_1^9 & a_1^{10} \\ a_2^1 & a_2^2 & a_2^3 & a_2^4 & a_2^7 & a_2^8 & a_2^9 & a_2^{10} \\ a_3^1 & a_3^2 & a_3^3 & a_3^4 & a_3^7 & a_3^8 & a_3^9 & a_3^{10} \\ a_4^1 & a_4^2 & a_4^3 & a_4^4 & a_4^7 & a_4^8 & a_4^9 & a_4^{10} \\ a_7^1 & a_7^2 & a_7^3 & a_7^4 & a_7^7 & a_7^8 & a_7^9 & a_7^{10} \\ a_8^1 & a_8^2 & a_8^3 & a_8^4 & a_8^7 & a_8^8 & a_8^9 & a_8^{10} \\ a_9^1 & a_9^2 & a_9^3 & a_9^4 & a_9^7 & a_9^8 & a_9^9 & a_9^{10} \\ a_{10}^1 & a_{10}^2 & a_{10}^3 & a_{10}^4 & a_{10}^7 & a_{10}^8 & a_{10}^9 & a_{10}^{10} \end{bmatrix}; \{RA\} = \begin{bmatrix} ra1 \\ ra2 \\ ra3 \\ ra4 \\ ra7 \\ ra8 \\ ra9 \\ ra10 \end{bmatrix}$$

The finite element matrix and residual vector for element B in processor 2 are

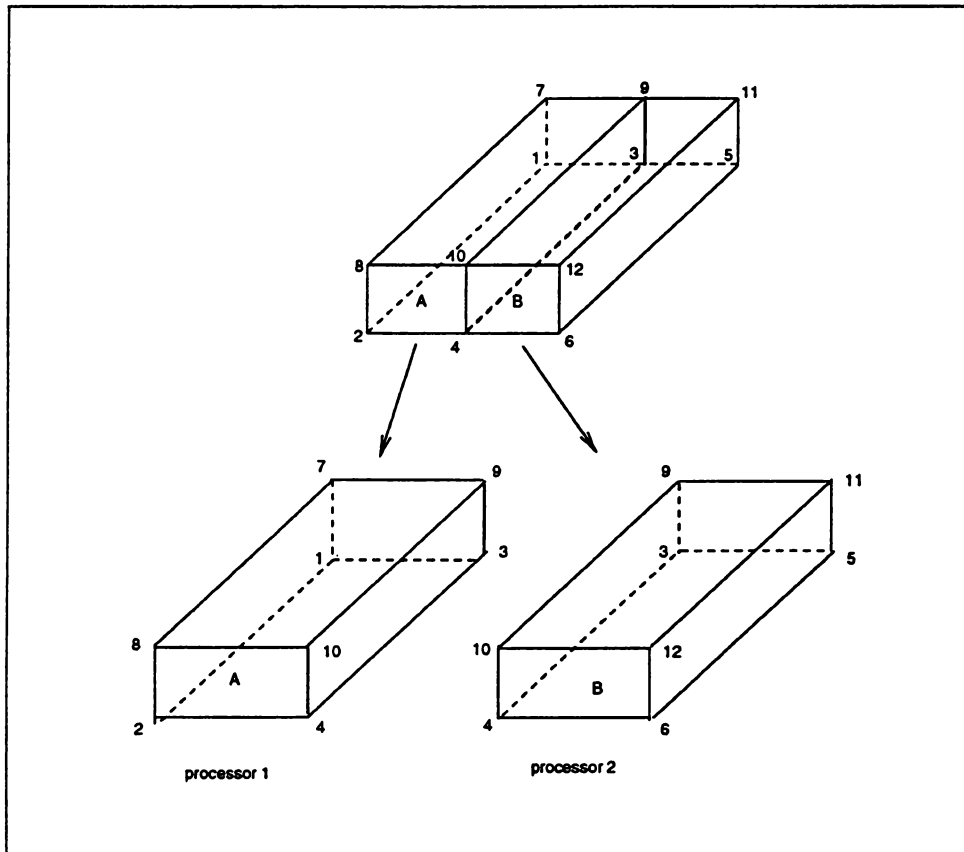


Figure 4.2 : A finite element mesh for the example problem

$$[SBXY] = \begin{bmatrix} b_3^3 & b_3^4 & b_3^5 & b_3^6 & b_3^9 & b_3^{10} & b_3^{11} & b_3^{12} \\ b_4^3 & b_4^4 & b_4^5 & b_4^6 & b_4^9 & b_4^{10} & b_4^{11} & b_4^{12} \\ b_5^3 & b_5^4 & b_5^5 & b_5^6 & b_5^9 & b_5^{10} & b_5^{11} & b_5^{12} \\ b_6^3 & b_6^4 & b_6^5 & b_6^6 & b_6^9 & b_6^{10} & b_6^{11} & b_6^{12} \\ b_9^3 & b_9^4 & b_9^5 & b_9^6 & b_9^9 & b_9^{10} & b_9^{11} & b_9^{12} \\ b_{10}^3 & b_{10}^4 & b_{10}^5 & b_{10}^6 & b_{10}^9 & b_{10}^{10} & b_{10}^{11} & b_{10}^{12} \\ b_{11}^3 & b_{11}^4 & b_{11}^5 & b_{11}^6 & b_{11}^9 & b_{11}^{10} & b_{11}^{11} & b_{11}^{12} \\ b_{12}^3 & b_{12}^4 & b_{12}^5 & b_{12}^6 & b_{12}^9 & b_{12}^{10} & b_{12}^{11} & b_{12}^{12} \end{bmatrix}; \{RB\} = \begin{bmatrix} rb3 \\ rb4 \\ rb5 \\ rb6 \\ rb9 \\ rb10 \\ rb11 \\ rb12 \end{bmatrix}$$

Assembling SAXY and SBXY, the global finite element matrix is

$$[SGXY] = \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 & a_1^4 & 0 & 0 & a_1^7 & a_1^8 & a_1^9 & a_1^{10} & 0 & 0 \\ a_2^1 & a_2^2 & a_2^3 & a_2^4 & 0 & 0 & a_2^7 & a_2^8 & a_2^9 & a_2^{10} & 0 & 0 \\ a_3^1 & a_3^2 & a_3^3 + b_3^3 & a_3^4 + b_3^4 & b_3^5 & b_3^6 & a_3^7 & a_3^8 & a_3^9 + b_3^9 & a_3^{10} + b_3^{10} & b_3^{11} & b_3^{12} \\ a_4^1 & a_4^2 & a_4^3 + b_4^3 & a_4^4 + b_4^4 & b_4^5 & b_4^6 & a_4^7 & a_4^8 & a_4^9 + b_4^9 & a_4^{10} + b_4^{10} & b_4^{11} & b_4^{12} \\ 0 & 0 & b_5^3 & b_5^4 & b_5^5 & b_5^6 & 0 & 0 & b_5^9 & b_5^{10} & b_5^{11} & b_5^{12} \\ 0 & 0 & b_6^3 & b_6^4 & b_6^5 & b_6^6 & 0 & 0 & b_6^9 & b_6^{10} & b_6^{11} & b_6^{12} \\ a_7^1 & a_7^2 & a_7^3 & a_7^4 & 0 & 0 & a_7^7 & a_7^8 & a_7^9 & a_7^{10} & 0 & 0 \\ a_8^1 & a_8^2 & a_8^3 & a_8^4 & 0 & 0 & a_8^7 & a_8^8 & a_8^9 & a_8^{10} & 0 & 0 \\ a_9^1 & a_9^2 & a_9^3 + b_9^3 & a_9^4 + b_9^4 & b_9^5 & b_9^6 & a_9^7 & a_9^8 & a_9^9 + b_9^9 & a_9^{10} + b_9^{10} & b_9^{11} & b_9^{12} \\ a_{10}^1 & a_{10}^2 & a_{10}^3 + b_{10}^3 & a_{10}^4 + b_{10}^4 & b_{10}^5 & b_{10}^6 & a_{10}^7 & a_{10}^8 & a_{10}^9 + b_{10}^9 & a_{10}^{10} + b_{10}^{10} & b_{10}^{11} & b_{10}^{12} \\ 0 & 0 & b_{11}^3 & b_{11}^4 & b_{11}^5 & b_{11}^6 & 0 & 0 & b_{11}^9 & b_{11}^{10} & b_{11}^{11} & b_{11}^{12} \\ 0 & 0 & b_{12}^3 & b_{12}^4 & b_{12}^5 & b_{12}^6 & 0 & 0 & b_{12}^9 & b_{12}^{10} & b_{12}^{11} & b_{12}^{12} \end{bmatrix}$$

The assembled residual vector and the global matrix vector products are

$$\{RMT\} = \begin{bmatrix} ra1 \\ ra2 \\ ra3 + rb3 \\ ra4 + rb4 \\ rb5 \\ rb6 \\ ra7 \\ ra8 \\ ra9 + rb9 \\ ra10 + rb10 \\ rb11 \\ rb12 \end{bmatrix} \quad \{GX\} = [SGXY]\{RMT\} = \begin{bmatrix} gx1 \\ gx2 \\ gx3 \\ gx4 \\ gx5 \\ gx6 \\ gx7 \\ gx8 \\ gx9 \\ gx10 \\ gx11 \\ gx12 \end{bmatrix}$$

The value of $\{GX\}$ at node 3 is,

$$\begin{aligned} gx3 = & (a_3^1)(ra1) + (a_3^2)(ra2) + (a_3^3 + b_3^3)(ra3 + rb3) + (a_3^4 + b_3^4)(ra4 + rb4) + \\ & (b_3^5)(rb5) + (b_3^6)(rb6) + (a_3^7)(ra7) + (a_3^8)(ra8) + (a_3^9 + b_3^9)(ra9 + rb9) + \\ & (a_3^{10} + b_3^{10})(ra10 + rb10) + (b_3^{11})(rb11) + (b_3^{12})(rb12) \end{aligned} \quad (4.2)$$

A sequence of operations is carried out among processor nodes to produce exactly the same global matrix-vector product $\{GX\}$ as illustrated in the following,

****STEP 1 :** Put values of residual vectors on interface boundary in new vectors $\{VPA\}$ and $\{VPB\}$,

$$\{VPA\} = \begin{bmatrix} ra3 \\ ra4 \\ ra9 \\ ra10 \end{bmatrix} \quad \{VPB\} = \begin{bmatrix} rb3 \\ rb4 \\ rb9 \\ rb10 \end{bmatrix}$$

**STEP 2 : Pass $\{VPA\}$ to processor 2 and $\{VPB\}$ to processor 1 and add them to the corresponding position of residual vector, $\{RA\}$ and $\{RB\}$,

$$\{RA\} + \{VPB\} = \{TRA\} = \begin{bmatrix} ra1 \\ ra2 \\ ra3 + rb3 \\ ra4 + rb4 \\ ra7 \\ ra8 \\ ra9 + rb9 \\ ra10 + rb10 \end{bmatrix} \quad \{RB\} + \{VPA\} = \{TRB\} = \begin{bmatrix} rb3 + ra3 \\ rb4 + ra4 \\ rb5 \\ rb6 \\ rb9 + ra9 \\ rb10 + ra10 \\ rb11 \\ rb12 \end{bmatrix}$$

**STEP 3 : Multiple SAXY and SBXY by $\{TRA\}$ and $\{TRB\}$,

$$[SAXY]\{TRA\} = \{PRA\} = \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 & a_1^4 & a_1^7 & a_1^8 & a_1^9 & a_1^{10} \\ a_2^1 & a_2^2 & a_2^3 & a_2^4 & a_2^7 & a_2^8 & a_2^9 & a_2^{10} \\ a_3^1 & a_3^2 & a_3^3 & a_3^4 & a_3^7 & a_3^8 & a_3^9 & a_3^{10} \\ a_4^1 & a_4^2 & a_4^3 & a_4^4 & a_4^7 & a_4^8 & a_4^9 & a_4^{10} \\ a_7^1 & a_7^2 & a_7^3 & a_7^4 & a_7^7 & a_7^8 & a_7^9 & a_7^{10} \\ a_8^1 & a_8^2 & a_8^3 & a_8^4 & a_8^7 & a_8^8 & a_8^9 & a_8^{10} \\ a_9^1 & a_9^2 & a_9^3 & a_9^4 & a_9^7 & a_9^8 & a_9^9 & a_9^{10} \\ a_{10}^1 & a_{10}^2 & a_{10}^3 & a_{10}^4 & a_{10}^7 & a_{10}^8 & a_{10}^9 & a_{10}^{10} \end{bmatrix} \begin{bmatrix} ra1 \\ ra2 \\ ra3 + rb3 \\ ra4 + rb4 \\ ra7 \\ ra8 \\ ra9 + rb9 \\ ra10 + rb10 \end{bmatrix} = \begin{bmatrix} pa1 \\ pa2 \\ pa3 \\ pa4 \\ pa7 \\ pa8 \\ pa9 \\ pa10 \end{bmatrix}$$

$$[SBXY]\{TRB\} = \{PRB\} = \begin{bmatrix} b_3^3 & b_3^4 & b_3^5 & b_3^6 & b_3^9 & b_3^{10} & b_3^{11} & b_3^{12} \\ b_4^3 & b_4^4 & b_4^5 & b_4^6 & b_4^9 & b_4^{10} & b_4^{11} & b_4^{12} \\ b_5^3 & b_5^4 & b_5^5 & b_5^6 & b_5^9 & b_5^{10} & b_5^{11} & b_5^{12} \\ b_6^3 & b_6^4 & b_6^5 & b_6^6 & b_6^9 & b_6^{10} & b_6^{11} & b_6^{12} \\ b_9^3 & b_9^4 & b_9^5 & b_9^6 & b_9^9 & b_9^{10} & b_9^{11} & b_9^{12} \\ b_{10}^3 & b_{10}^4 & b_{10}^5 & b_{10}^6 & b_{10}^9 & b_{10}^{10} & b_{10}^{11} & b_{10}^{12} \\ b_{11}^3 & b_{11}^4 & b_{11}^5 & b_{11}^6 & b_{11}^9 & b_{11}^{10} & b_{11}^{11} & b_{11}^{12} \\ b_{12}^3 & b_{12}^4 & b_{12}^5 & b_{12}^6 & b_{12}^9 & b_{12}^{10} & b_{12}^{11} & b_{12}^{12} \end{bmatrix} \begin{bmatrix} rb3 + ra3 \\ rb4 + ra4 \\ rb5 \\ rb6 \\ rb9 + ra9 \\ rb10 + ra10 \\ rb11 \\ rb12 \end{bmatrix} = \begin{bmatrix} pb3 \\ pb4 \\ pb5 \\ pb6 \\ pb9 \\ pb10 \\ pb11 \\ pb12 \end{bmatrix}$$

Values of $\{PRA\}$ and $\{PRB\}$ at node 3 are,

$$\begin{aligned} pra3 = & (a_3^1)(ra1) + (a_3^2)(ra2) + (a_3^3)(ra3 + rb3) + (a_3^4)(ra4 + rb4) + \\ & (a_3^7)(ra7) + (a_3^8)(ra8) + (a_3^9)(ra9 + rb9) + (a_3^{10})(ra10 + rb10) \end{aligned} \quad (4.3)$$

$$\begin{aligned} prb3 = & (b_3^3)(ra3 + rb3) + (b_3^4)(ra4 + rb4) + (b_3^5)(rb1) + (b_3^2)(rb2) + \\ & (b_3^9)(ra9 + rb9) + (b_3^{10})(ra10 + rb10) + (b_3^{11})(rb11) + (b_3^{12})(rb12) \end{aligned} \quad (4.4)$$

It can be seen that $gx3 = pra3 + prb3$.

****STEP 4 :** Extract values of $\{PRA\}$ and $\{PRB\}$ on interface boundary, put them in $\{VPA\}$ and $\{VPB\}$, and pass them to adjacent processors,

$$\{VPA\} = \begin{bmatrix} pra3 \\ pra4 \\ pra9 \\ pra10 \end{bmatrix} ; \quad \{VPB\} = \begin{bmatrix} prb3 \\ prb4 \\ prb9 \\ prb10 \end{bmatrix}$$

****STEP 5 :** Add $\{VPA\}$ and $\{VPB\}$ to $\{PRA\}$ and $\{PRB\}$ and the sequence is complete.

$$\{PRA\} + \{VPB\} = \{FRA\} = \begin{bmatrix} fa1 \\ fa2 \\ fa3 \\ fa4 \\ fa7 \\ fa8 \\ fa9 \\ fa10 \end{bmatrix} ; \quad \{PRB\} + \{VPA\} = \{FRB\} = \begin{bmatrix} fb3 \\ fb4 \\ fb5 \\ fb6 \\ fb9 \\ fb10 \\ fb11 \\ fb12 \end{bmatrix}$$

The jacobian matrix and residual vectors produced by the system of velocity-vorticity equations are

$$[JQQ] = \begin{bmatrix} JUU & 0 & 0 & 0 & JU\Omega_y & JU\Omega_x & 0 \\ 0 & JVV & 0 & JV\Omega_x & 0 & JV\Omega_z & 0 \\ 0 & 0 & JWW & JW\Omega_x & JW\Omega_y & 0 & 0 \\ J\Omega_x U & J\Omega_x V & J\Omega_x W & J\Omega_x \Omega_x & J\Omega_x \Omega_y & J\Omega_x \Omega_z & J\Omega_x \Theta \\ J\Omega_y U & J\Omega_y V & J\Omega_y W & J\Omega_y \Omega_x & J\Omega_y \Omega_y & J\Omega_y \Omega_z & J\Omega_y \Theta \\ J\Omega_z U & J\Omega_z V & J\Omega_z W & J\Omega_z \Omega_x & J\Omega_z \Omega_y & J\Omega_z \Omega_z & 0 \\ J\Theta U & J\Theta V & J\Theta W & 0 & 0 & 0 & J\Theta \Theta \end{bmatrix} ; \{RVEC\} = \begin{bmatrix} RU \\ RV \\ RW \\ R\Omega_x \\ R\Omega_y \\ R\Omega_z \\ R\Theta \end{bmatrix}$$

They are composed of block matrices and vectors of the same structure. Matrix vector multiplications are carried out on individual nonzero blocks and their results are summed. The final block vector is

$$\begin{bmatrix} [JUU]RU + [JU\Omega_y]R\Omega_y + [JU\Omega_x]R\Omega_x \\ [JVV]RV + [JV\Omega_x]R\Omega_x + [JV\Omega_z]R\Omega_z \\ [JWW]RW + [JW\Omega_x]R\Omega_x + [JW\Omega_y]R\Omega_y \\ [J\Omega_x U]RU + [J\Omega_x V]RV + [J\Omega_x W]RW + [J\Omega_x \Omega_x]R\Omega_x + [J\Omega_x \Omega_y]R\Omega_y + [J\Omega_x \Omega_z]R\Omega_z + [J\Omega_x \Theta]R\Theta \\ [J\Omega_y U]RU + [J\Omega_y V]RV + [J\Omega_y W]RW + [J\Omega_y \Omega_x]R\Omega_x + [J\Omega_y \Omega_y]R\Omega_y + [J\Omega_y \Omega_z]R\Omega_z + [J\Omega_y \Theta]R\Theta \\ [J\Omega_z U]RU + [J\Omega_z V]RV + [J\Omega_z W]RW + [J\Omega_z \Omega_x]R\Omega_x + [J\Omega_z \Omega_y]R\Omega_y + [J\Omega_z \Omega_z]R\Omega_z \\ [J\Theta U]RU + [J\Theta V]RV + [J\Theta W]RW + [J\Theta \Theta]R\Theta \end{bmatrix}$$

As a result, a total of thirty-three block matrix vector products are performed in a processor to complete a global operation of matrix vector multiplication.

4.6 Vector Product Computation

The CGS and BiCGSTAB procedures require vector product operations. A summary of this operation is illustrated. Given vectors \mathbf{v} and \mathbf{t} , their representations in elements A and B are

$$\{IVA\} = \begin{bmatrix} va1 \\ va2 \\ va3 \\ va4 \\ va7 \\ va8 \\ va9 \\ va10 \end{bmatrix}; \{ITA\} = \begin{bmatrix} ta1 \\ ta2 \\ ta3 \\ ta4 \\ ta7 \\ ta8 \\ ta9 \\ ta10 \end{bmatrix}; \{IVB\} = \begin{bmatrix} vb3 \\ vb4 \\ vb5 \\ vb6 \\ vb9 \\ vb10 \\ vb11 \\ vb12 \end{bmatrix}; \{ITB\} = \begin{bmatrix} tb3 \\ tb4 \\ tb5 \\ tb6 \\ tb9 \\ tb10 \\ tb11 \\ tb12 \end{bmatrix}$$

After adding values of \mathbf{v} and \mathbf{t} along interface (common) boundary nodes, the new vectors are

$$\{VA\} = \begin{bmatrix} va1 \\ va2 \\ va3 + vb3 \\ va4 + vb4 \\ va7 \\ va8 \\ va9 + vb9 \\ va10 + vb10 \end{bmatrix}; \{TA\} = \begin{bmatrix} ta1 \\ ta2 \\ ta3 + tb3 \\ ta4 + tb4 \\ ta7 \\ ta8 \\ ta9 + tb9 \\ ta10 + tb10 \end{bmatrix}; \{VB\} = \begin{bmatrix} vb3 + va3 \\ vb4 + va4 \\ vb5 \\ vb6 \\ vb9 + va9 \\ vb10 + va10 \\ vb11 \\ vb12 \end{bmatrix}; \{TB\} = \begin{bmatrix} tb3 + ta3 \\ tb4 + ta4 \\ tb5 \\ tb6 \\ tb9 + ta9 \\ tb10 + tb10 \\ tb11 \\ tb12 \end{bmatrix}$$

The vector product of $\{VA\}$ and $\{IVA\}$ in processor 1 is

$$\begin{aligned} vpa = & (va1)(ta1) + (va2)(ta2) + (va3 + vb3)(ta3) + (va4 + vb4)(ta4) + \\ & (va7)(ta7) + (va8)(ta8) + (va9 + vb9)(ta9) + (va10 + vb10)(ta10) \end{aligned} \quad (4.5)$$

The vector product of $\{VB\}$ and $\{IVB\}$ in processor 2 is

$$v_{pb} = (vb1)(tb1) + (vb2)(tb2) + (va3 + vb3)(tb3) + (va4 + vb4)(tb4) + \\ (vb7)(tb7) + (vb8)(tb8) + (va9 + vb9)(tb9) + (va10 + vb10)(tb10) \quad (4.6)$$

The global vector product is obtained by summing values of vector products in processors 1 and 2. Thus, a global gather operation is required. The CM5 provides library subroutines, `cmmd_reduce_double(variable,operation)`, that will perform a restricted set of global operations for a variable across processors. The operation, `cmmd_combiner_dadd`, is used for the summation of the inner product values among processors. For infinite norm calculation, the maximum value is obtained using the operation, `cmmd_combiner_dmax`.

4.7 Boundary Condition Implementation

Since the Newton algorithm is used in solving the system of equations, the variable, δQ , at Dirichlet nodes is always zero. Communication among processors is not needed for application of boundary conditions of Dirichlet type. The entire row of the jacobian matrix of a Dirichlet boundary node is zeroed out except at the diagonal element which is set to one. The same location in the residual vector is also set to zero. The global matrix vector product will be the same because entries at Dirichlet boundary nodes are always zero.

Application of the kinematic vorticity boundary condition requires special treatment for nodes on interface boundaries. Assume that the kinematic vorticity bound-

ary condition is needed on node 4. Then in Figure 4.2, the entire row 4 of SAXY is zeroed out. Coefficients of the kinematic boundary equations at nodes 4 and 10 are inserted for a_4^4 and a_4^{10} . These values are divided by two if they are located on interface boundary shared by two processors. The matrix and the residual vector in processor 1 become

$$[SAXY] = \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 & a_1^4 & a_1^7 & a_1^8 & a_1^9 & a_1^{10} \\ a_2^1 & a_2^2 & a_2^3 & a_2^4 & a_2^7 & a_2^8 & a_2^9 & a_2^{10} \\ a_3^1 & a_3^2 & a_3^3 & a_3^4 & a_3^7 & a_3^8 & a_3^9 & a_3^{10} \\ 0 & 0 & 0 & (kv4)/2 & 0 & 0 & 0 & (kv10)/2 \\ a_7^1 & a_7^2 & a_7^3 & a_7^4 & a_7^7 & a_7^8 & a_7^9 & a_7^{10} \\ a_8^1 & a_8^2 & a_8^3 & a_8^4 & a_8^7 & a_8^8 & a_8^9 & a_8^{10} \\ a_9^1 & a_9^2 & a_9^3 & a_9^4 & a_9^7 & a_9^8 & a_9^9 & a_9^{10} \\ a_{10}^1 & a_{10}^2 & a_{10}^3 & a_{10}^4 & a_{10}^7 & a_{10}^8 & a_{10}^9 & a_{10}^{10} \end{bmatrix} ; \{RA\} = \begin{bmatrix} ra1 \\ ra2 \\ ra3 \\ (rvk4)/2 \\ ra7 \\ ra8 \\ ra9 \\ ra10 \end{bmatrix}$$

In processor 2, the matrix and residual vector are

$$[SBXY] = \begin{bmatrix} b_3^3 & b_3^4 & b_3^5 & b_3^6 & b_3^9 & b_3^{10} & b_3^{11} & b_3^{12} \\ 0 & (kv4)/2 & 0 & 0 & 0 & (kv10)/2 & 0 & 0 \\ b_5^3 & b_5^4 & b_5^5 & b_5^6 & b_5^9 & b_5^{10} & b_5^{11} & b_5^{12} \\ b_6^3 & b_6^4 & b_6^5 & b_6^6 & b_6^9 & b_6^{10} & b_6^{11} & b_6^{12} \\ b_9^3 & b_9^4 & b_9^5 & b_9^6 & b_9^9 & b_9^{10} & b_9^{11} & b_9^{12} \\ b_{10}^3 & b_{10}^4 & b_{10}^5 & b_{10}^6 & b_{10}^9 & b_{10}^{10} & b_{10}^{11} & b_{10}^{12} \\ b_{11}^3 & b_{11}^4 & b_{11}^5 & b_{11}^6 & b_{11}^9 & b_{11}^{10} & b_{11}^{11} & b_{11}^{12} \\ b_{12}^3 & b_{12}^4 & b_{12}^5 & b_{12}^6 & b_{12}^9 & b_{12}^{10} & b_{12}^{11} & b_{12}^{12} \end{bmatrix} ; \{RB\} = \begin{bmatrix} rb3 \\ (rvk4)/2 \\ rb5 \\ rb6 \\ rb9 \\ rb10 \\ rb11 \\ rb12 \end{bmatrix}$$

If the kinematic boundary condition is not along an interface, then the entire values of vk and rvk should be used as supposed multiplied by one-half.

RESULTS AND DISCUSSIONS

A series of test cases are set up to verify and benchmark aspects of the parallel velocity-vorticity finite element formulation for the incompressible Navier-Stokes equations. These test cases can be categorized into two classes of problems: verification and benchmarking. Verification is the comparison between computational results and the analytical "closed form" solution. The fully developed flow in a straight rectangular cross-section channel is the first-verification case for the developed velocity-vorticity formulation. The solution computed by the present formulation is compared with the analytical solution to establish the validity and accuracy of the algorithm boundary condition form.

Benchmarking involves the comparison of computed results to those produced by an independent computational model. The second test case is a benchmark comparison of developing flow in a square cross-section channel. A two dimensional benchmark problem of the step-wall difuser with separation and reattachment of the primary vortex is also tested. The well-known lid-driven cavity benchmark will have to be examined since ample details of the flow solution for a wide range of Reynolds numbers are documented. The last case studied is a 3-D thermal cavity. Quantitative computational solutions are presented to give a comprehensive view of this 3-D

buoyancy-driven flow problem. Furthermore, coupling of the energy equation in this test case truly completes the scope for the incompressible Navier-Stokes thermal fluid flow problem statement in the velocity-vorticity form.

The solutions of two dimensional test cases are computed by a serial program on workstations with a banded direct solver. The solutions in three dimensions are obtained with the CM5 parallel computer augmented with the described CG-type iterative sparse solvers.

5.1 Straight Channel Flow

5.1.1 Fully-Developed Flow

Laminar isothermal flow in a straight channel is a simple but important verification test case for incompressible fluid flow. For two dimensional flow, Figure 5.1, the analytical velocity profile is parabolic, hence the vorticity profile is linear.

For the three dimensional case, the steady state velocity and vorticity profiles are (White 1974)

$$u = \frac{48 \xi_u(y, z, h)}{\pi^3 \varphi} \quad (5.1)$$

$$v = w = 0 \quad (5.2)$$

$$\Omega_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} = 0 \quad (5.3)$$

$$\Omega_y = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} = \frac{48 \xi_z(y, z, h)}{\pi^3 \varphi} \quad (5.4)$$

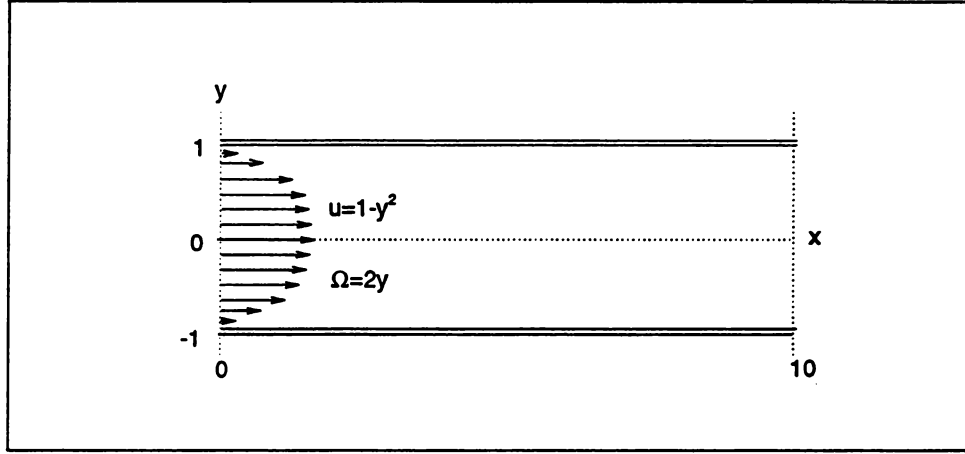


Figure 5.1 : Geometry and boundary conditions for 2-D channel flow

$$\Omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = \frac{-48}{\pi^3} \frac{\xi_y(y, z, h)}{\varphi} \quad (5.5)$$

$$\xi_u(y, z, h) = \sum_{n=1,3,5}^N (-1)^{\frac{n-1}{2}} \left[1 - \frac{\cosh(n\pi y/2h)}{\cosh(n\pi/2)} \right] \frac{\cos(n\pi z/2h)}{n^3} \quad (5.6)$$

$$\xi_z(y, z, h) = \sum_{n=1,3,5}^N (-1)^{\frac{n-1}{2}} \left[1 - \frac{\cosh(n\pi y/2h)}{\cosh(n\pi/2)} \right] \frac{-(\sin n\pi z/2h)(n\pi)}{2n^3} \quad (5.7)$$

$$\xi_y(y, z, h) = \sum_{n=1,3,5}^N (-1)^{\frac{n-1}{2}} \left[-(n\pi) \frac{\sinh(n\pi y/2h)}{2 \cosh(n\pi/2)} \right] \frac{\cos(n\pi z/2h)}{n^3} \quad (5.8)$$

$$\varphi = 1 - \frac{192}{\pi^5} \sum_{n=1,3,5}^N \frac{\tanh(n\pi/2)}{n^5} \quad (5.9)$$

where h is the duct half-height and N is a large integer, e.g, $N=200$ is used. The computational geometry of the 3-D square channel is shown in Figure 5.2. The uniform mesh discretization is $M=16 \times 16 \times 16$ consisting of 4913 nodes and 4096 trilinear elements. For boundary conditions, Figure 5.3, steady state analytical velocity

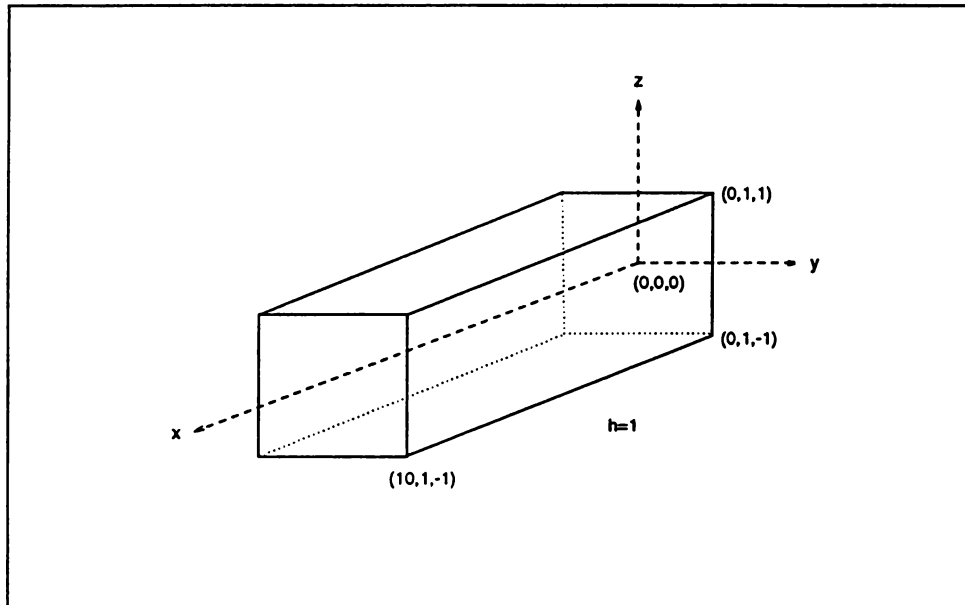


Figure 5.2 : Geometry for 3-D square channel flow

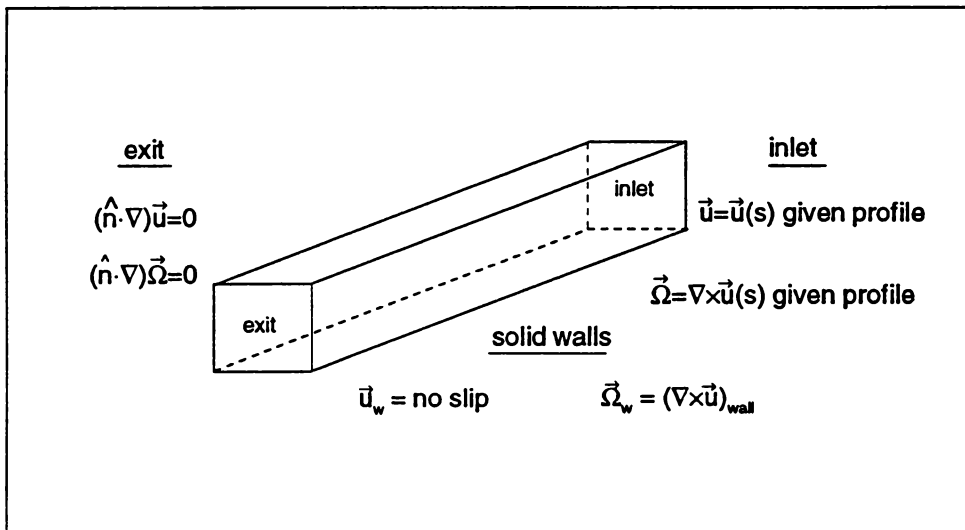


Figure 5.3 : Boundary condition for straight channel flow

and vorticity profiles are prescribed across the inlet, and vanishing Neumann conditions for velocity and vorticity are exact and imposed at the exit plane. Boundary conditions for the velocity at channel walls are no-slip, hence the developed kinematic vorticity boundary conditions are imposed for components of vorticity. The flow inside the channel is initialized at zero. A large time step size is taken with the backward Euler integration scheme to drive the solution to steady state in one single time step.

For a uniform mesh size of $M=8 \times 8$ with 81 nodes and 64 bilinear elements, the computed steady-state solution for the 2-D channel is tabulated in Figure 5.4. To 7 significant digits, the analytical parabolic velocity profile is predicted throughout the channel. The computed center plane solution for the 3-D square duct is tabulated in Figure 5.5. As in 2-D, the entrance velocity profile is propagated throughout the channel to 4 significant digits. These data clearly verify that the chosen velocity-vorticity formulation, with new second order vorticity kinematic boundary condition, is indeed mass conserving for this verification case.

5.1.2 Developing Flow

Developing flow in a three dimensional square channel has been extensively reported in the CFD literature. Han (1960) calculated analytically the velocity profiles and correlated the entrance length (to fully developed flow) for rectangular channel flow with respect to the Reynolds number, Re , based on hydraulic diameter, D_h . The

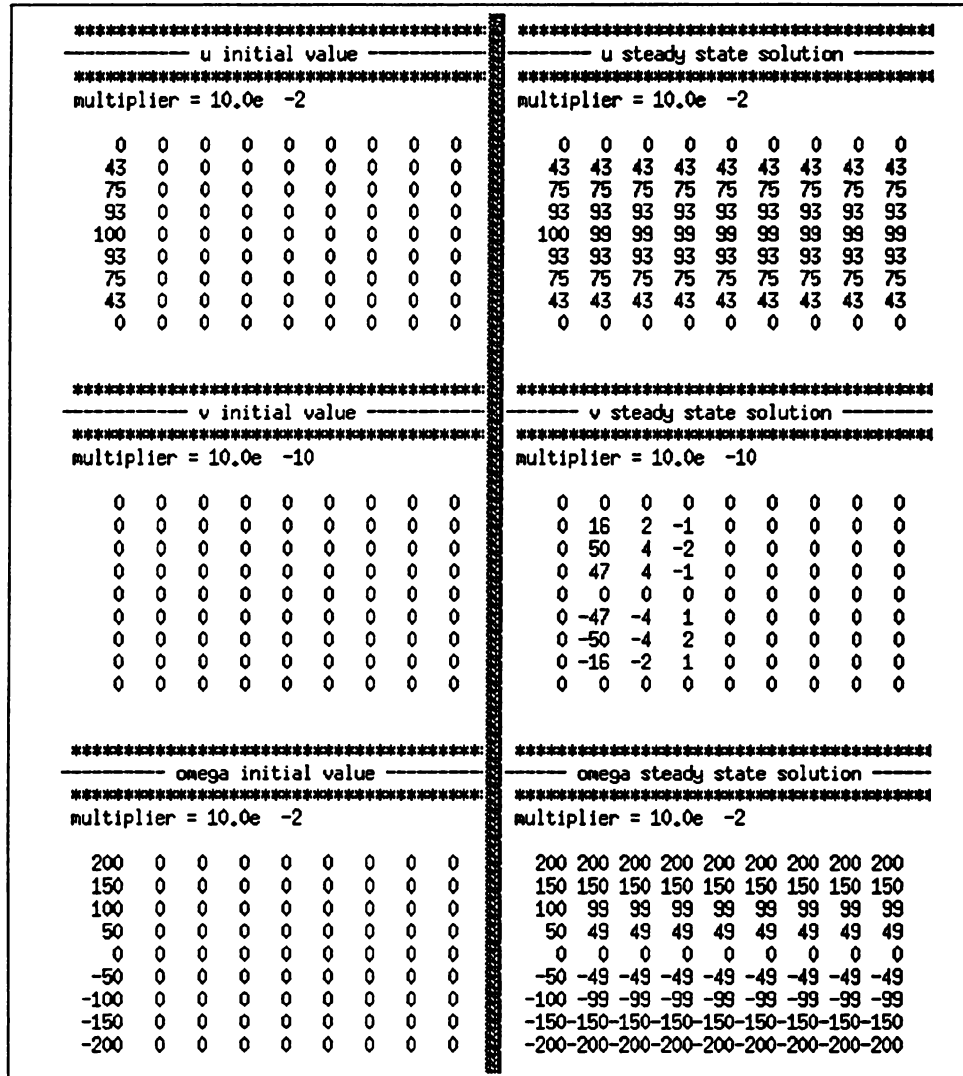


Figure 5.4 : Initial and steady state solutions for 2-D fully developed channel flow

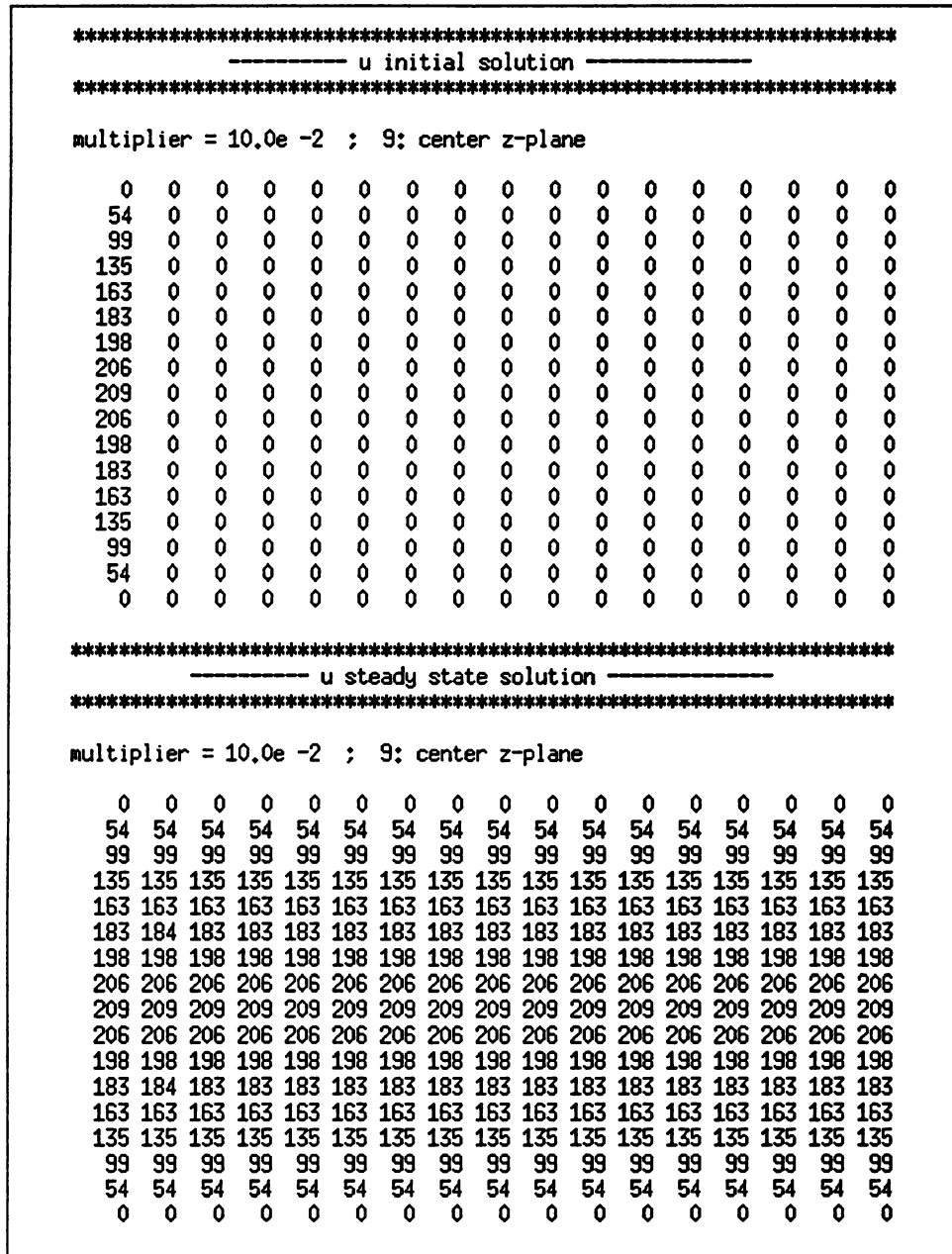


Figure 5.5 : Initial and steady state center plane solutions for 3-D fully developed channel flow

correlated entrance length, L_{cel} , expression is

$$L_{cel} = x_{el}/(D_h \times Re) \quad (5.10)$$

$$D_h = \frac{4ab}{a + b} \quad (5.11)$$

where a and b are transverse measures of the rectangular duct and x_{el} is the computed distance to where the centerline velocity reaches 99% of the fully-developed value. For a duct of square cross-section, then $a = h = b$ and $D_h = 2h$.

A detailed examination of 3-D square cross-section channel flow is reported by Williams (1993). The same channel geometry ($h = 0.5$) is used, and the slug inflow velocity profiles is defined as $u_{in} = 1.065$ everywhere except at the wall nodes where $u = 0$, and $v_{in} = 0 = w_{in}$ everywhere. A boundary layer is built up in the wall region as the flow develops until a fully developed steady state velocity profile is recovered beyond the entrance length.

For a uniform mesh of $M=32 \times 32 \times 32$, on $0 \leq x \leq 10$, the computed steady state axial velocity profile at select axial stations throughout the 3-D channel is shown in Figure 5.6. The velocity profile at the exit plane is compared with the analytical solution in Figure 5.7. Again, the present result is in excellent agreement with the analytical solution. These results confirm that the velocity-vorticity formulation does indeed satisfy the continuity equation. Aregbesola & Burley (1977) used the vorticity, vector-scalar potential method to compute the 3-D square channel. Their calculated value for the correlated entrance length is $L_{cel} = 0.086$ for $Re = 100$ on a $M=32 \times 13 \times 13$

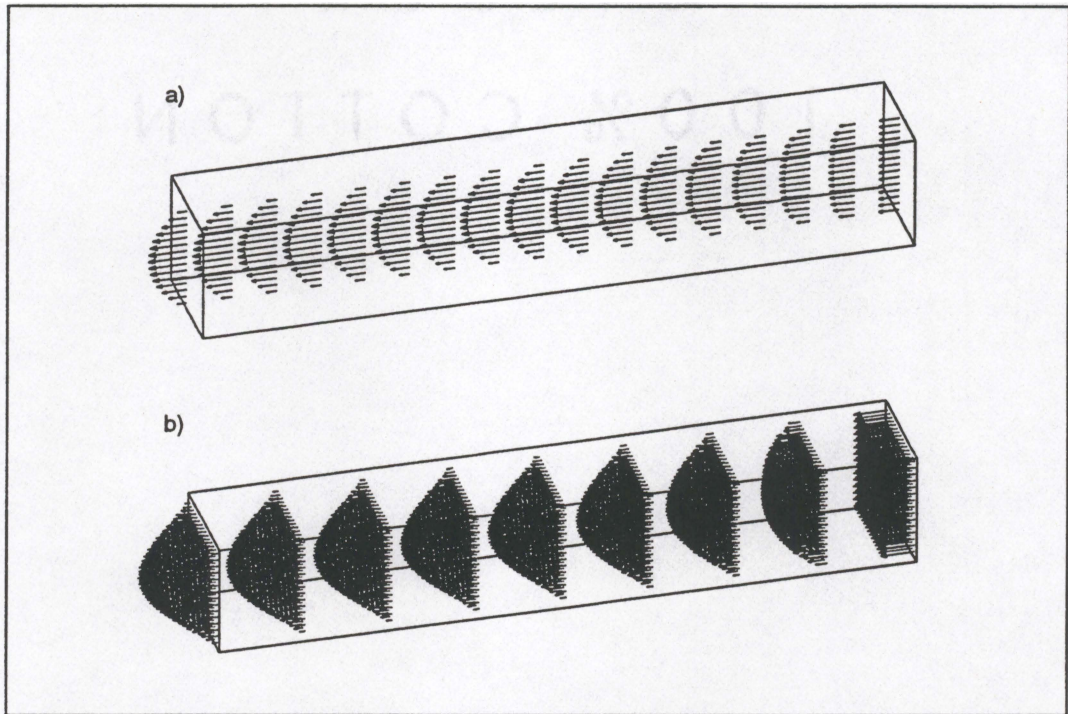


Figure 5.6 : Velocity profile for square channel flow a) centerline profile b) perspectives of axial flow distributions.

Table 5.1 : Entrance length for developing flow in a square duct

Source	Correlated Entrance Length	Mesh
Han (1960)	0.075	Analytical
Goldstein and Kreid (1967)	0.090	Experimental
Aregbesola and Burley (1977)	0.086	M=32x13x13 (uniform)
Williams (1991)	0.078	M=100x30x30 (non-uniform)
Present results	0.082	M=32x32x32 (uniform)

uniform mesh. The computed correlated entrance length by Williams (1993) using an M=100x30x30 non-uniform mesh and an inviscid entrance adjustment region is $L_{cel} = 0.078$. An experimental measurement done by Goldstein & Kreid (1967) gave a value of the correlated entrance length of $L_{cel} = 0.09$.

The results of the computed correlated entrance length for the 3-D channel are compared to the present result in Table 5.1. The present solution yields a correlated entrance length of 0.082, which is 9.3 % different from Han's analytical solution based upon boundary layer linearization approximations to the Navier-stokes equations. The present solutions, however, is 5.1 % different from Williams' computed solutions, which has three times more elements in the axial direction than the present results.

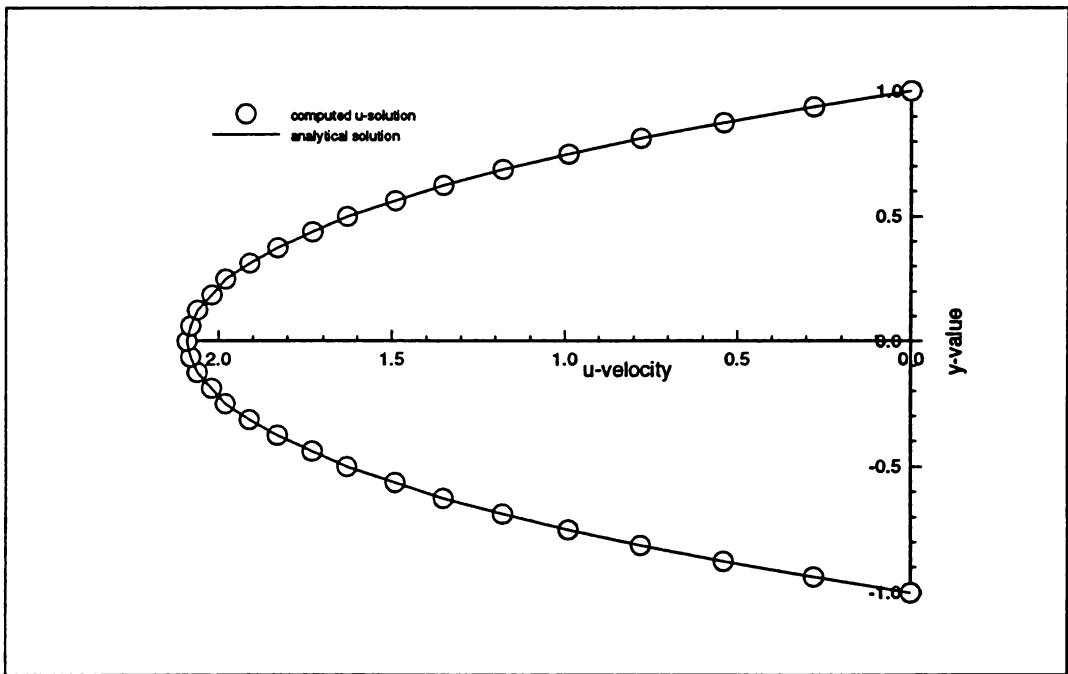


Figure 5.7 : Velocity Profile for 3-D developing channel flow at the exit

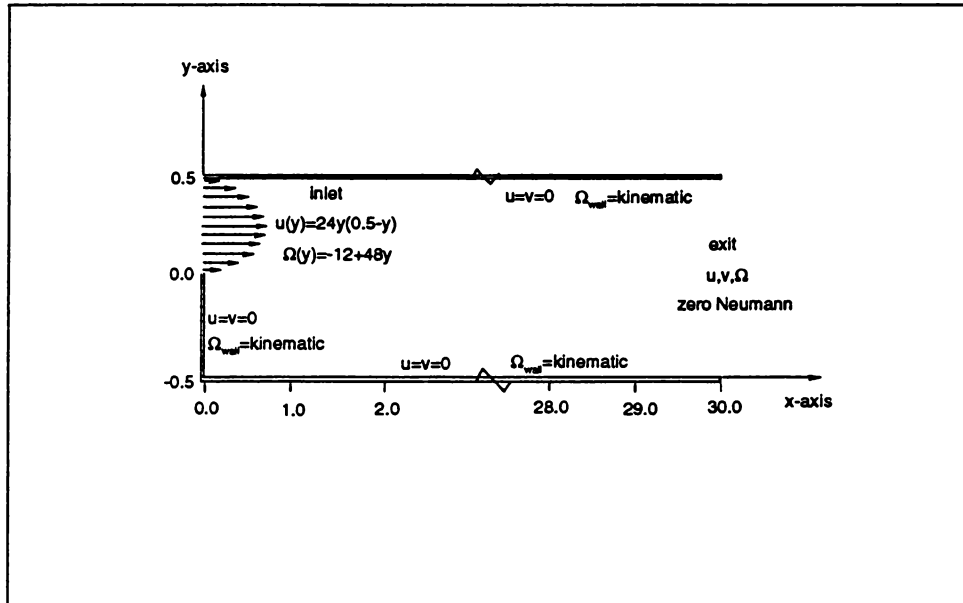


Figure 5.8 : Geometry and boundary conditions for backward facing step

5.2 Backward Facing Step

The 2-D backward facing step is a benchmark problem to study the robustness of the formulation for flow separation prediction. The test problem posted for the Symposium for Open Boundary Conditions (Gresho 1991a) is used. The geometry and boundary conditions for the test problem are given in Figure 5.8.

As the flow enters the channel, it separates at the step corner and forms a primary(lower) recirculation eddy, the size of which depends on Reynolds number. A second eddy will form on the upper wall as the Reynolds' number exceeds 400. The Reynolds number of the test case is 800, hence a second eddy forms on the top wall. An extensive study of this benchmark is published by Gartling (1990).

Table 5.2 : Flow characteristics of backward facing step

Flow Characteristics	Present Result	Gartling(1990)
Number of elements	64x64 = 4096	30x600 = 18000
Type of elements	Bilinear	Biquadratic
Center of lower eddy	(3.17 , -0.2)	(3.35 , -0.2)
Length of lower eddy	5.7	6.10
Separation point of upper eddy	(4.8 , 0.5)	(4.85 , 0.5)
Reattachment point of upper eddy	(10.4 , 0.5)	(10.48 , 0.5)

An axially non-uniform $M=64 \times 64$ mesh containing 4225 nodes and 4096 bilinear elements is used for the benchmark assessment. A velocity vector plot of the computed solution after 120 time steps with $\Delta t = 2.0$ is shown in Figure 5.9(a). The extremum change in δQ between time steps is of order 10^{-3} . A close up view of the velocity profile at the entrance section of the backstep is shown in Figure 5.9(b). The streamline contours thereon clearly illustrate the formation of the primary and secondary recirculation regions.

The comparison of the present results to the data of Gartling (1990) is summarized in Table 5.2. The reattachment point, where the u-velocity changes sign, hence the vorticity vanishes, of the lower primary eddy is 6.6 % shorter. A finer mesh near the recirculation region is needed to obtain a better agreement. The separation point

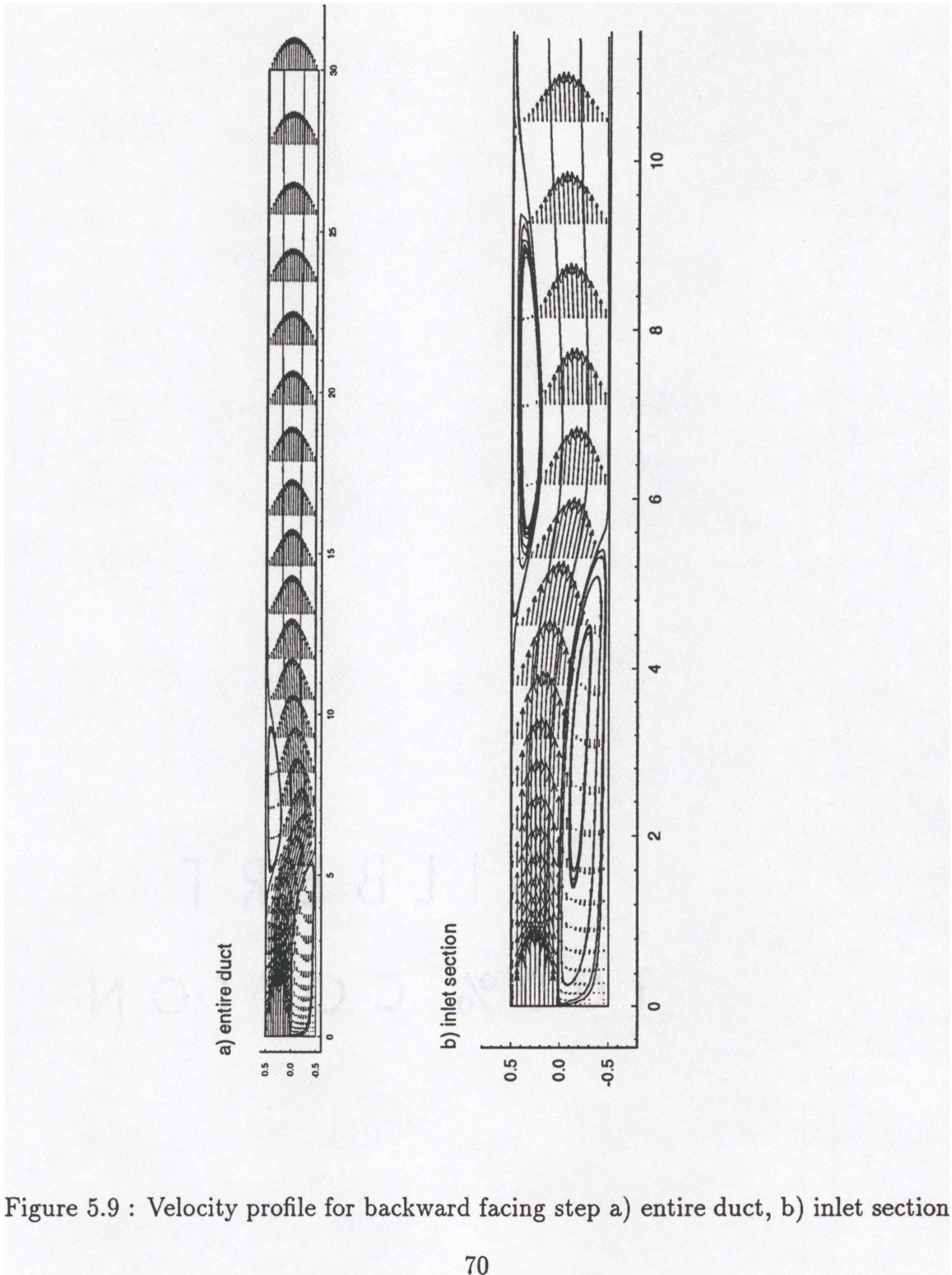


Figure 5.9 : Velocity profile for backward facing step a) entire duct, b) inlet section

and reattachment point of the upper eddy are within 2 % of the results computed by Gartling (1990).

5.3 Lid Driven Cavity

The lid driven cavity is a classic benchmark for laminar incompressible flow. The accepted comparative 2-D solution for various Reynolds numbers is documented by Ghia, Ghia & Shin (1982) using a multigrid streamfunction-vorticity formulation. Using an explicit finite element approach with mass lumping, Gresho, Chan, Lee & Upson (1984) also present a study of the 2-D problem. Agarwal (1981) is the first to report a numerical solution for the 3-D problem using a velocity-vorticity formulation with a finite difference scheme. With the increase of supercomputing capability in recent years, the 3-D lid-driven cavity problem has emerged as a standard benchmark for 3-D incompressible Navier-Stokes algorithms. A series of 3-D lid-driven cavity results at various Reynolds numbers are reported recently by Ku, Hirsh & Taylor (1987), Gatski, Grosch & Rose (1989), Napolitano & Pascazio (1991), Guevremont et al. (1993), Guj & Stella (1993), Fujima et al. (1994), Babu & Korpela (1994), and Jiang, Lin & Povinelli (1994).

5.3.1 Lid-Driven Cavity in Two Dimensions

The configuration and nomenclature of the 2-D lid-driven cavity flow is identified by Ghia et al. (1982). For $Re \leq 3200$, the basic features of the flow solution are given in Figure 5.10. Geometries and properties of the primary vortex (PV), the secondary

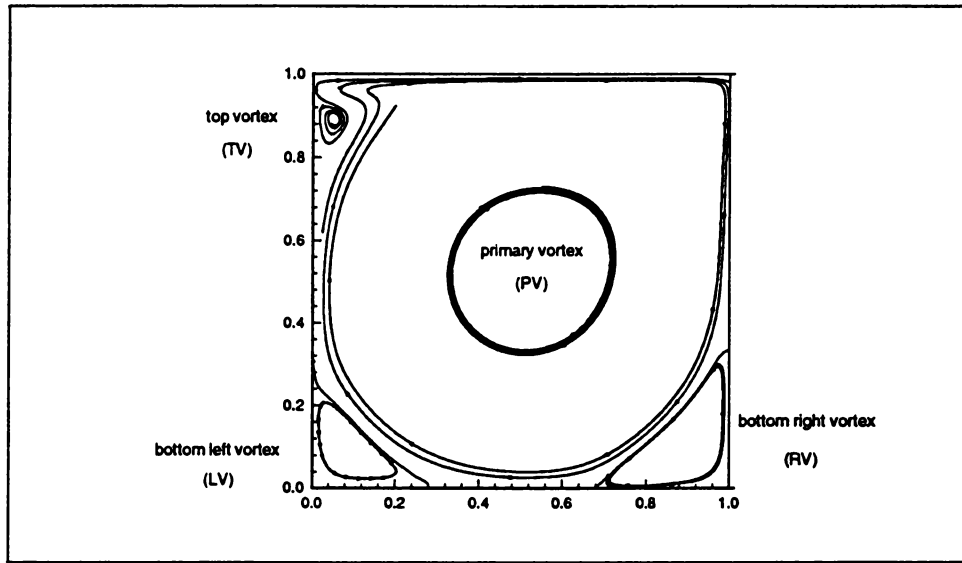


Figure 5.10 : Geometry and key features of the 2-D driven cavity

lower right corner vortex (RV), the secondary lower left corner vortex (LV), and the secondary top left corner vortex (TV) have been documented by Ghia et al. (1982).

A nonuniform cartesian $M=32 \times 32$ mesh, Figure 5.11, with 1089 nodes and 1024 bilinear elements is used for the present study, for $Re = 400$, $Re = 1000$, $Re = 2000$, and $Re = 3200$. Plots of velocity vector with streamlines and vorticity contours for $Re = 400$ are shown in Figure 5.12 and Figure 5.13, respectively. At this low Reynolds number, only the secondary RV can be clearly distinguished. The vorticity contours are smooth throughout the entire solution domain even at the top right corner singularity, indicating this mesh is providing adequate resolution.

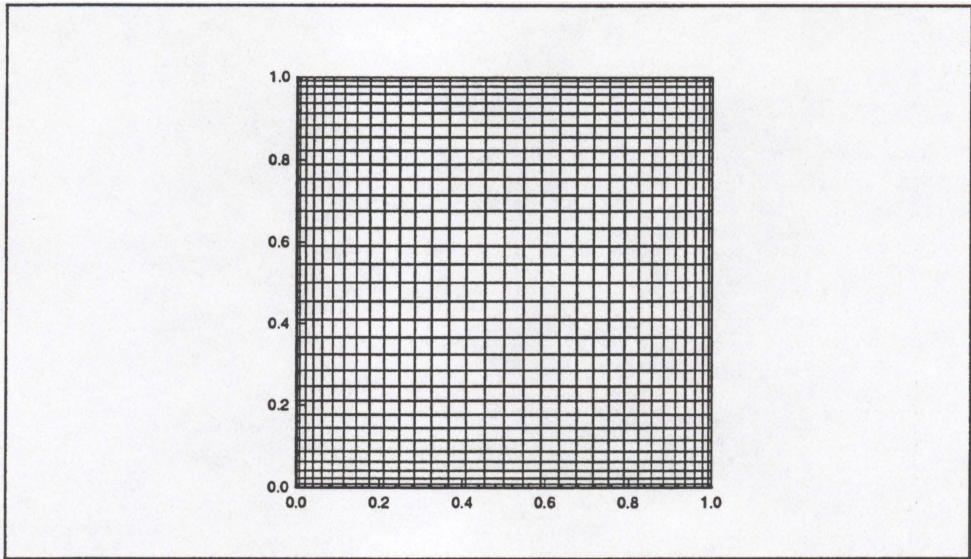


Figure 5.11 : Finite element mesh for 2-D driven cavity

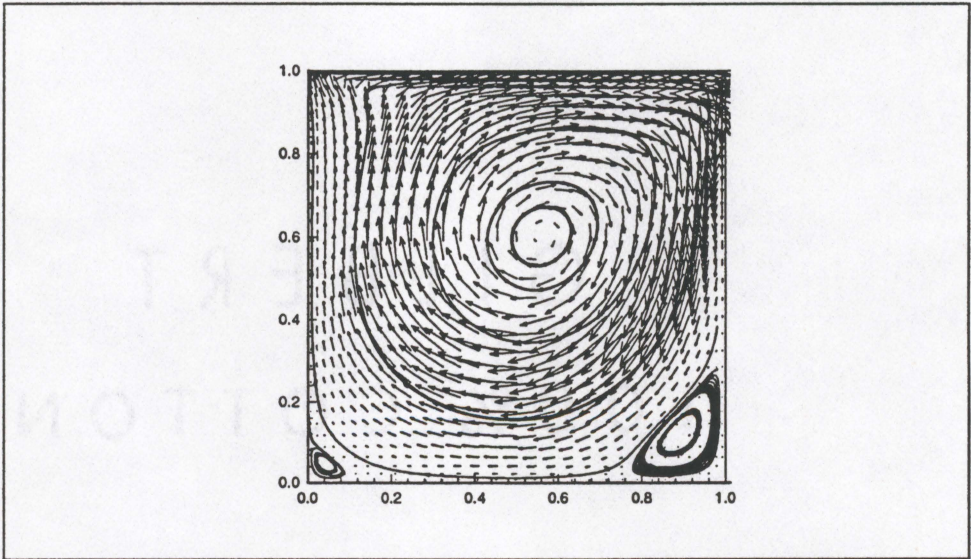


Figure 5.12 : Streamline contour plot of the 2-D driven cavity at $Re=400$

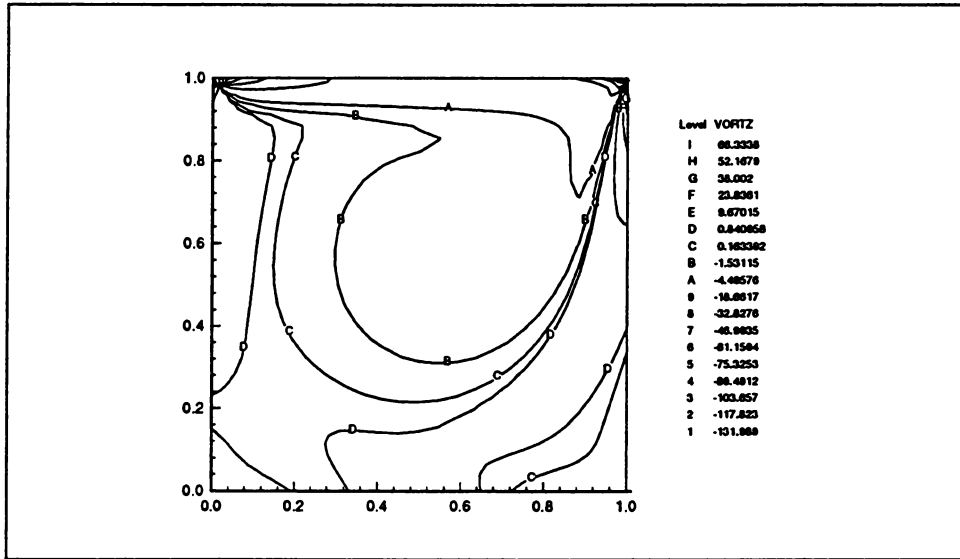


Figure 5.13 : Vorticity contour plot for 2-D driven cavity at $Re=400$

At $Re = 1000$, a secondary LV now appears, Figure 5.14. In comparison to $Re = 400$, the center of the PV moves towards the center of the cavity. The vorticity contours, shown in Figure 5.15, are smooth and non-dispersive.

At $Re = 2000$, the secondary RV and LV are clearly evident in Figure 5.16 and the TV is shown to be forming. A very modest dispersive error is evident in the vorticity contour leading into the singular corner, Figure 5.17.

At $Re = 3200$, the velocity vector plot, Figure 5.18, reveals all major flow features for the 2-D lid-driven cavity benchmark. All secondary vortices are clearly shown and well established. Location of the PV center has moved to the position where changes will be minimal for $Re > 3200$. From Figure 5.19, the vorticity contours are more clustered along the moving surface, and dispersion error is very evident in the right

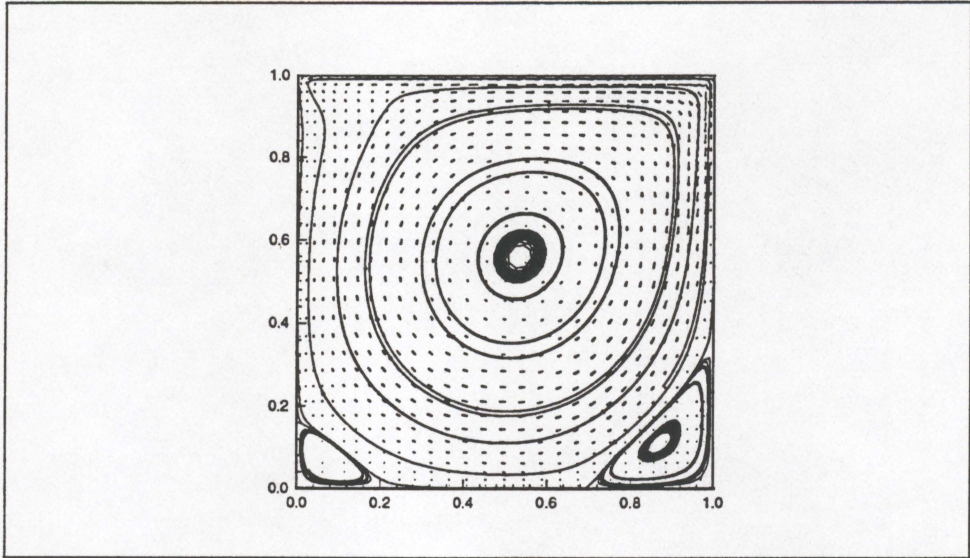


Figure 5.14 : Streamline contour plot of the 2-D driven cavity at $Re=1000$

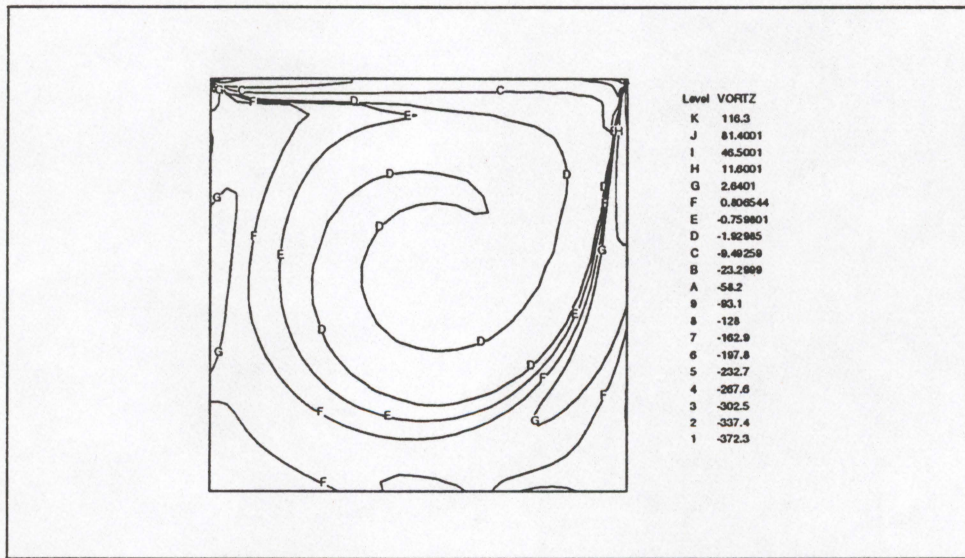


Figure 5.15 : Vorticity contour plot for 2-D driven cavity at $Re=1000$

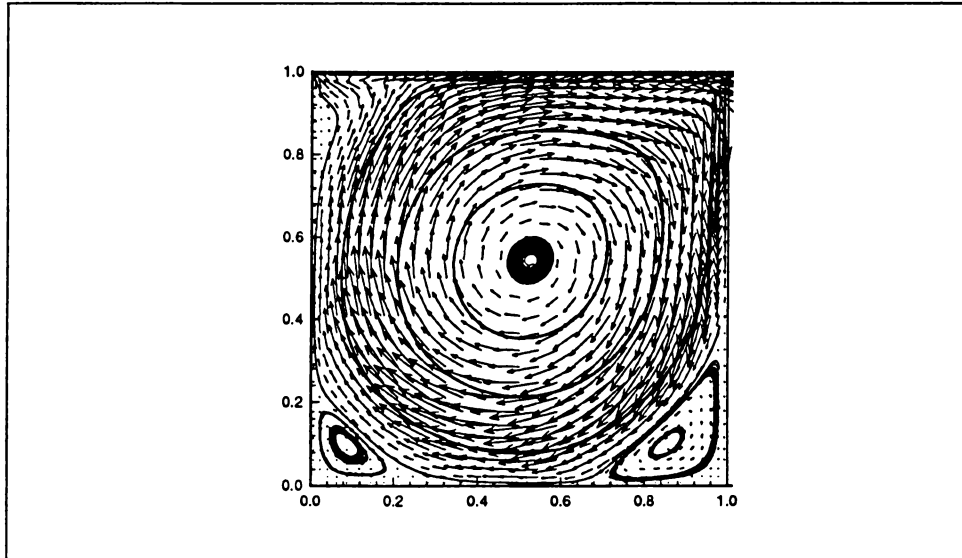


Figure 5.16 : Streamline contour plot of the 2-D driven cavity at $Re=2000$

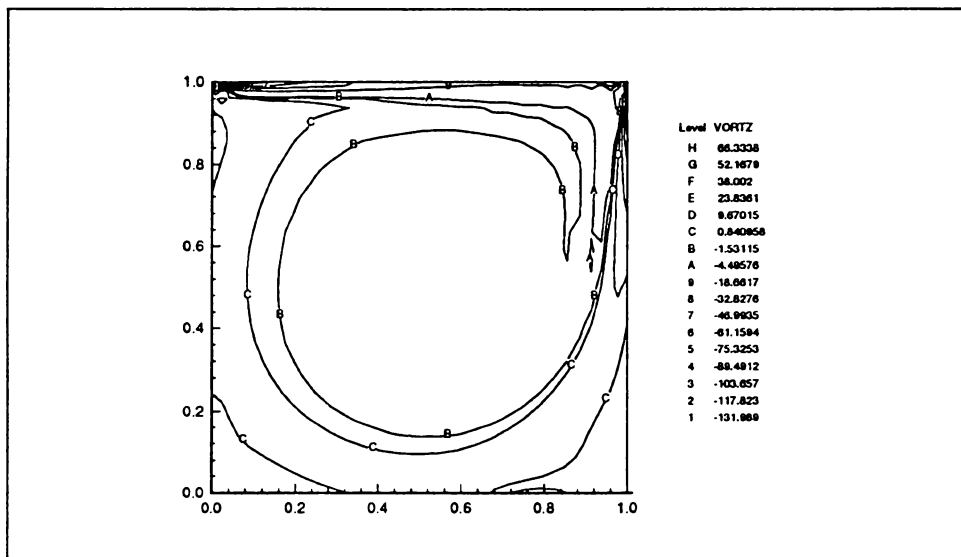


Figure 5.17 : Vorticity contour plot for 2-D driven cavity at $Re=2000$

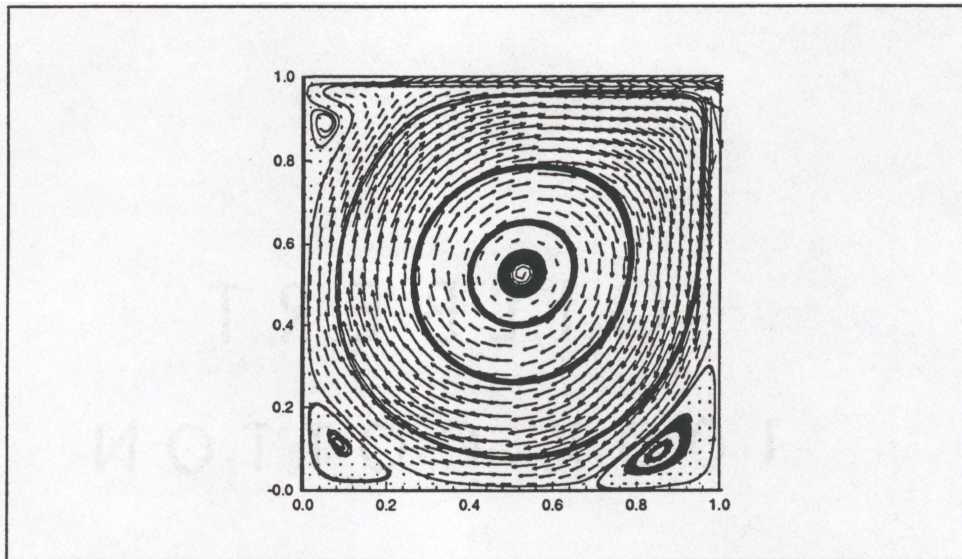


Figure 5.18 : Streamline contour plot of the 2-D driven cavity at $Re=3200$

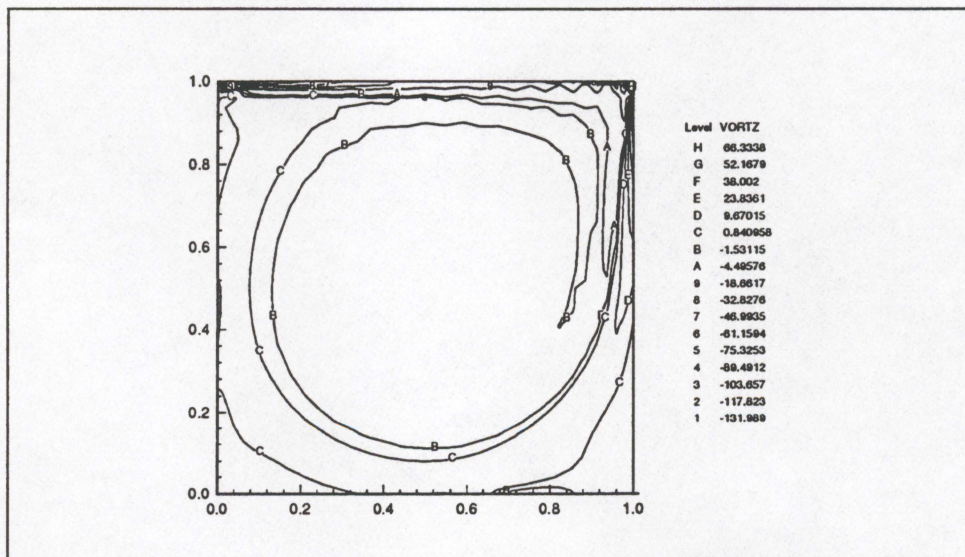


Figure 5.19 : Vorticity contour plot for 2-D driven cavity at $Re=3200$

Table 5.3 : Location of vortices for 2-D lid-driven cavity

source	vortices	$Re = 400$	$Re = 1000$	$Re = 3200$
Ghia et al	PV	(0.555,0.606)	(0.531,0.563)	(0.517,0.547)
present results	PV	(0.558,0.603)	(0.535,0.560)	(0.522,0.540)
Ghia et al	RV	(0.891,0.125)	(0.859,0.109)	(0.813,0.086)
present results	RV	(0.892,0.121)	(0.870,0.111)	(0.832,0.088)
Ghia et al	LV	(0.051,0.047)	(0.086,0.078)	(0.086,0.109)
present results	LV	(0.048,0.045)	(0.084,0.074)	(0.085,0.114)

top hand region of the cavity. Hence, this mesh is not adequate to resolve the vorticity gradients in this region.

Locations of the center of the primary and secondary vortices are summarized in Table 5.3. The present results are within 6 % ($Re = 400, 1000$) and 5 % ($Re = 3200$) of the results of Ghia (1982) in which the final grid ($M=129 \times 129$) is much larger.

A non-uniform 64×64 mesh for $Re = 1000$ was also tested; solution details are shown in Figure 5.20. The minimum u velocity along the vertical line through the geometric center of the cavity is 3.6 % lower than Ghia's prediction. The reattachment lengths for the right and left bottom eddies are within 3 % of Ghia's solution. Location of the PV is improved to (0.532,0.563), which is within 0.2 % of Ghia's solution. However, very little improvement for locations of RV, (0.867,0.111), and LV, (0.082,0.077), is observed for this finer mesh solution.

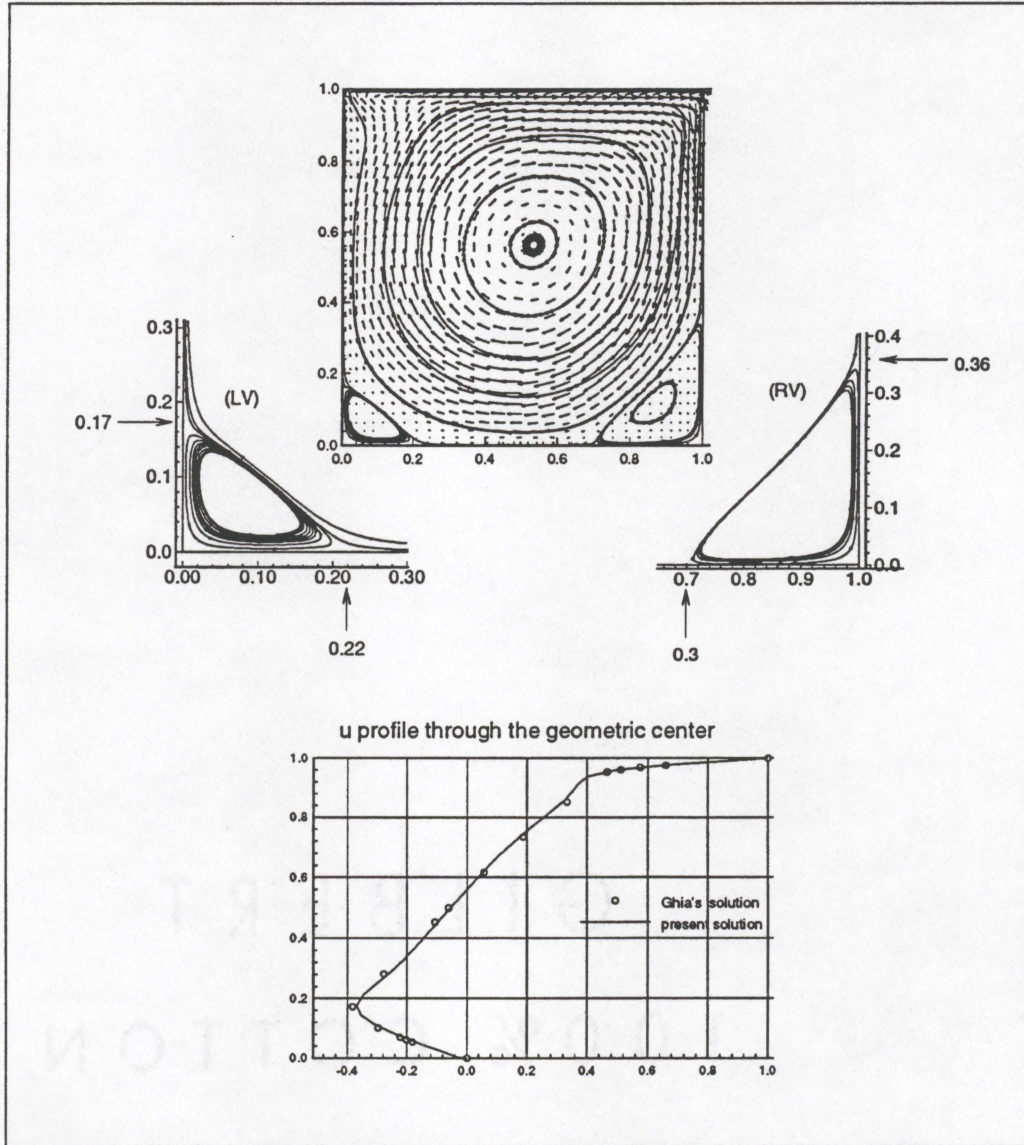


Figure 5.20 : Solution details of 2-D driven cavity for $Re = 1000$, $M=64 \times 64$.

5.3.2 Lid-Driven Cavity in Three Dimension

Significant computational resources are required to resolve the complexity of the 3-D driven cavity benchmark. There are six state variable members at each node for the velocity-vorticity formulation. A nonuniform cartesian $M=32 \times 32 \times 32$ mesh, shown in Figure 5.21, contains 32768 trilinear elements, and the Newton statement contains 215622 equations. Solving implicitly such a huge matrix statement on a sequential computer would be prohibitively time consuming. Even the CM5 parallel computer at the University of Tennessee was utilized to its full extent to solve this problem. The total memory requirement is approximately 416 Mbytes. With 32 processors on this CM5, 13 Mbytes of local processor memory is used.

The 3-D cavity is a unit cube, with the top wall moving parallel to the positive x -axis at unit velocity, $u=1$. The vorticity boundary conditions are kinematic on every boundary node of the cube. The initial condition is zero vorticity everywhere, since the top plate velocity will instantaneously generate vorticity which will be propagated throughout the whole computational domain until the flow field reaches a steady state solution. In the present study, the steady state solutions for $Re = 100$, $Re = 400$, and $Re = 1000$ are computed.

The 3-D sectional perspective views for the velocity vector field and the vorticity contours are shown in Figure 5.22, Figure 5.23, and Figure 5.24. The vorticity plots at x -mid-plane for Ω_x , y -mid-plane for Ω_y , and z -mid-plane for Ω_z confirm the carrying of important flow information. Contour lines of the vortices shown on each mid-plane

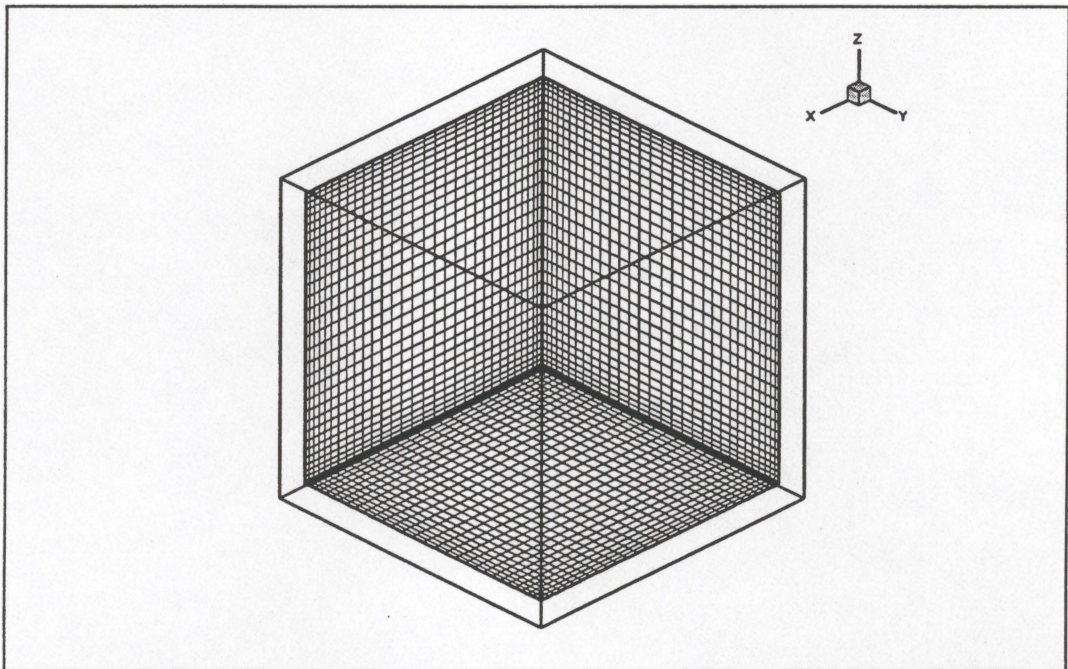


Figure 5.21 : Non-uniform finite element mesh for 3-D driven cavity, $M=32^3$

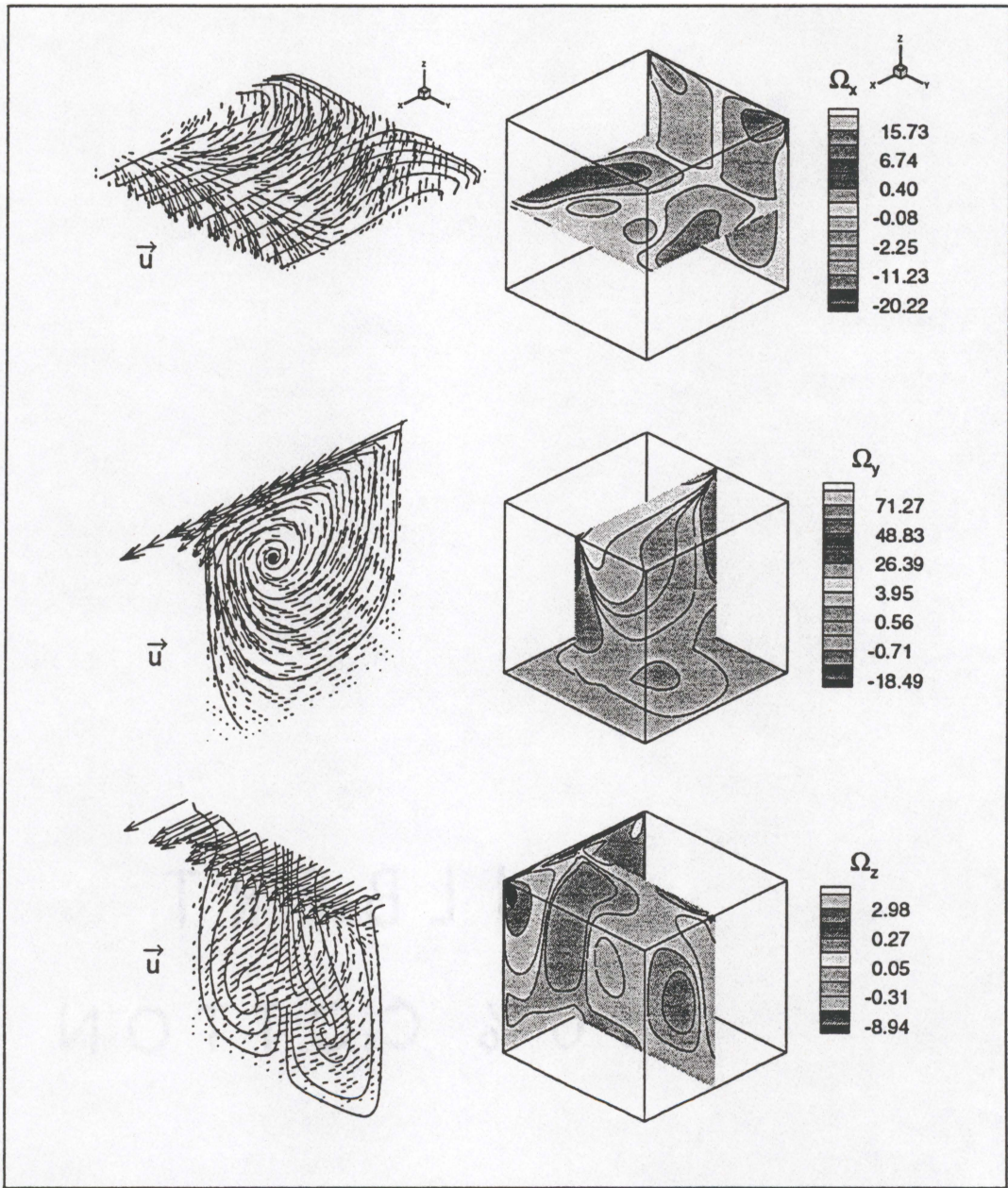


Figure 5.22 : Perspective 3-D summary for driven cavity solution, $Re=100$, $M=32^3$

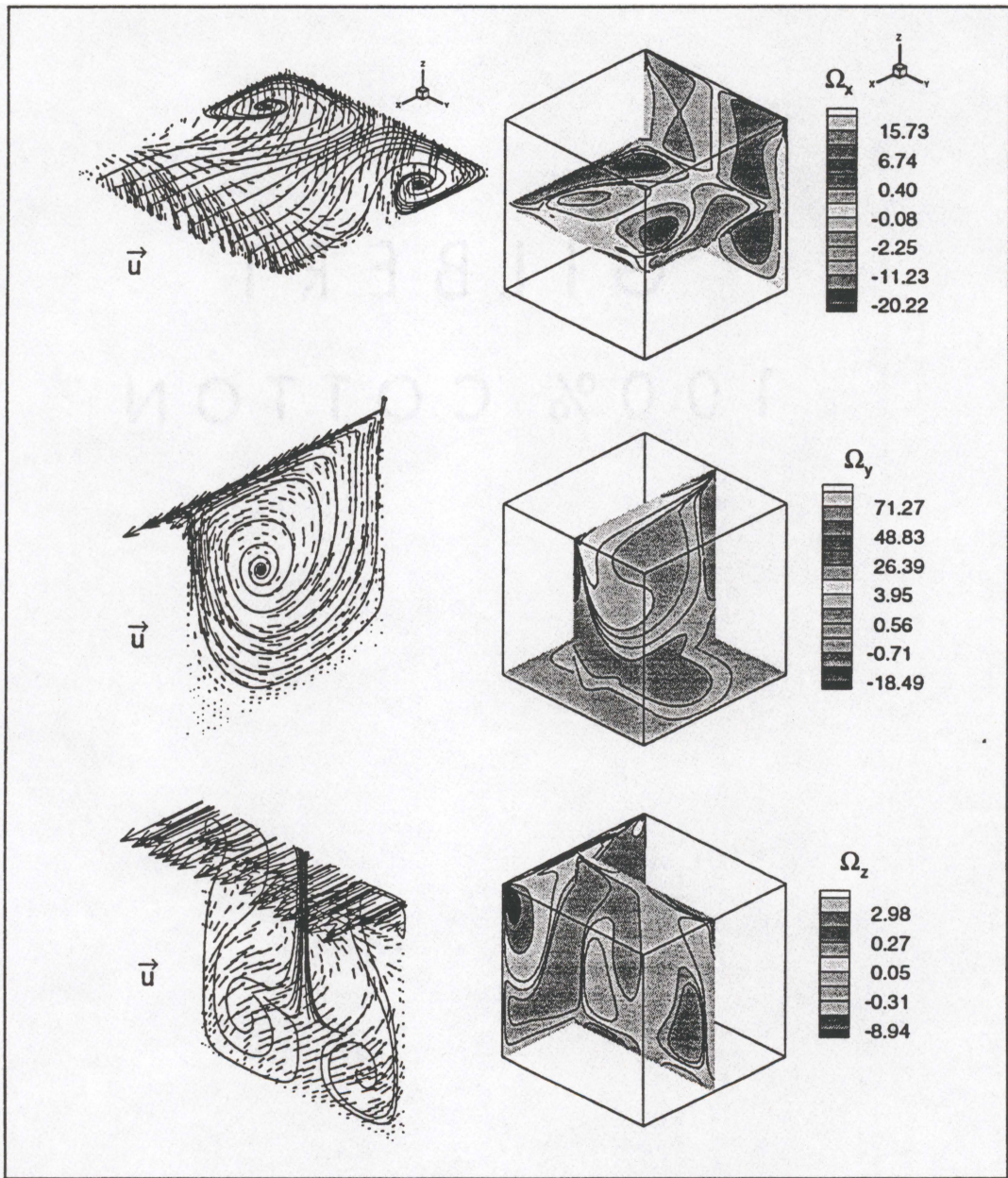


Figure 5.23 : Perspective 3-D summary for driven cavity solution, $Re=400$, $M=32^3$

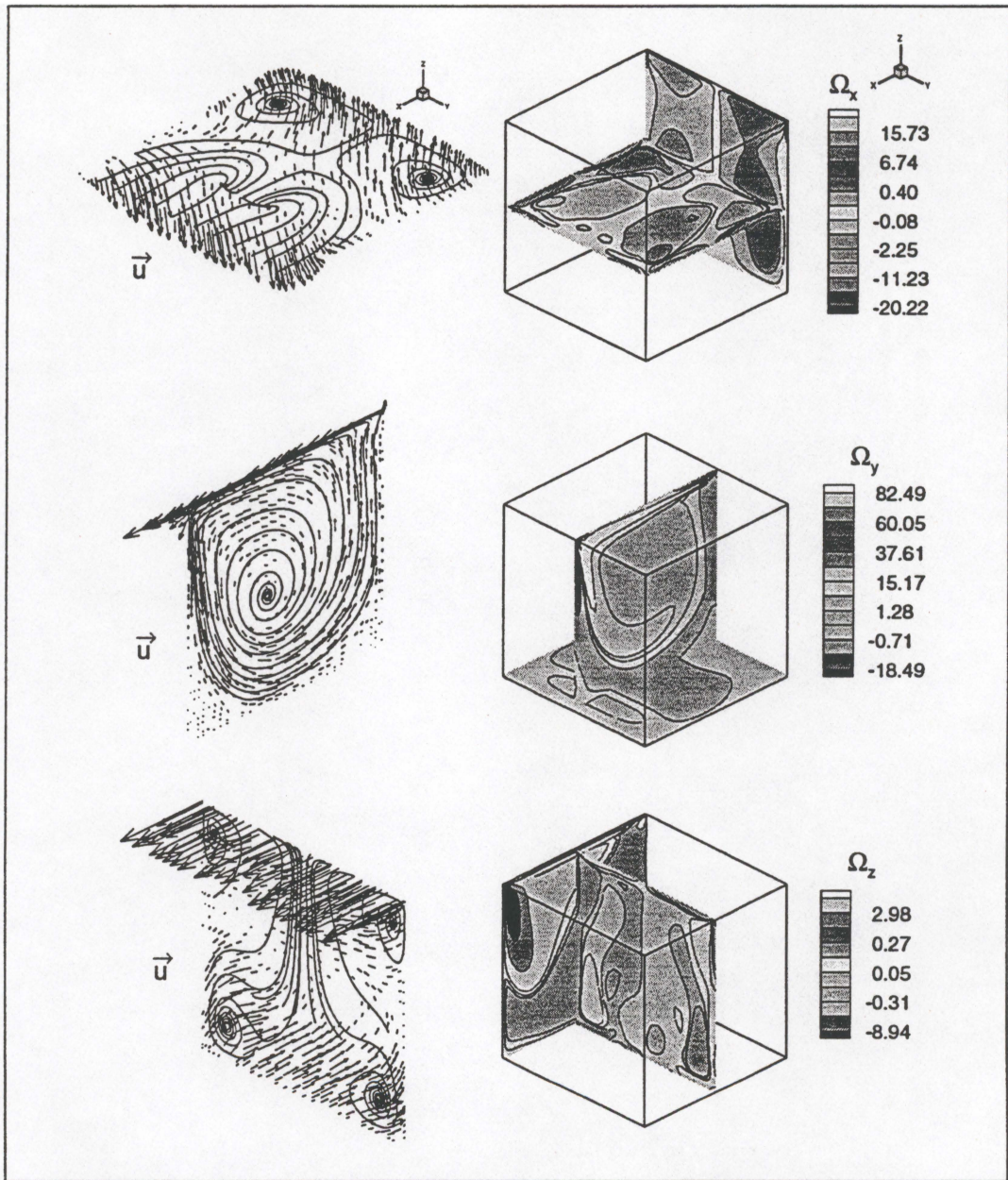


Figure 5.24 : Perspective 3-D summary for driven cavity solution, $Re=1000$, $M=32^3$

on the velocity vector plots correspond to the 2-D streamtraces projected from the 3-D velocity vector field.

Two dimensional planar projections of the velocity vector field at $Re = 100$, $Re = 400$, and $Re = 1000$ on the three centroidal planes of the cube are shown in Figure 5.25. The velocity vector plot on the x-z centroidal plane resembles the 2-D solution. The axis of the primary vortex starts off in the upper right half region, hence gradually move towards the center as the Reynolds number increases. Note that the secondary vortices, RV, LV, and TV identified in two dimensions, Figure 5.10, do not occur in the 3-D problem. Instead, the flow is fully three dimensional such that these secondary vortices turn sideways in the y direction and upwards in the z-direction.

As can be seen in the velocity vector plot on the y-z centroidal plane, a pair of vortices appear near the centerline and move out towards the lower corners as Reynolds number increases. Two small recirculation cells are also emerging at the top corners as the Reynolds number goes through $Re = 400$ to $Re = 1000$.

As for the x-y centroidal plane, at $Re = 100$ the flow is basically returning, symmetrically, normally along the x-y boundary. However, at $Re = 400$, the flow is now turning in the z-direction, which results in a pair of 2-D projected vortices from the 3-D velocity vectors. A second pair of projected vortices also appears as the Reynolds number reaches 1000, Figure 5.25.

Similar pattern for the 2-D center-plane velocity vector plots for $Re = 100$, $Re = 400$, and $Re = 1000$, are reported by Guj and Stilla (1993), Fujima et al (1994), and Jiang et al. (1994).

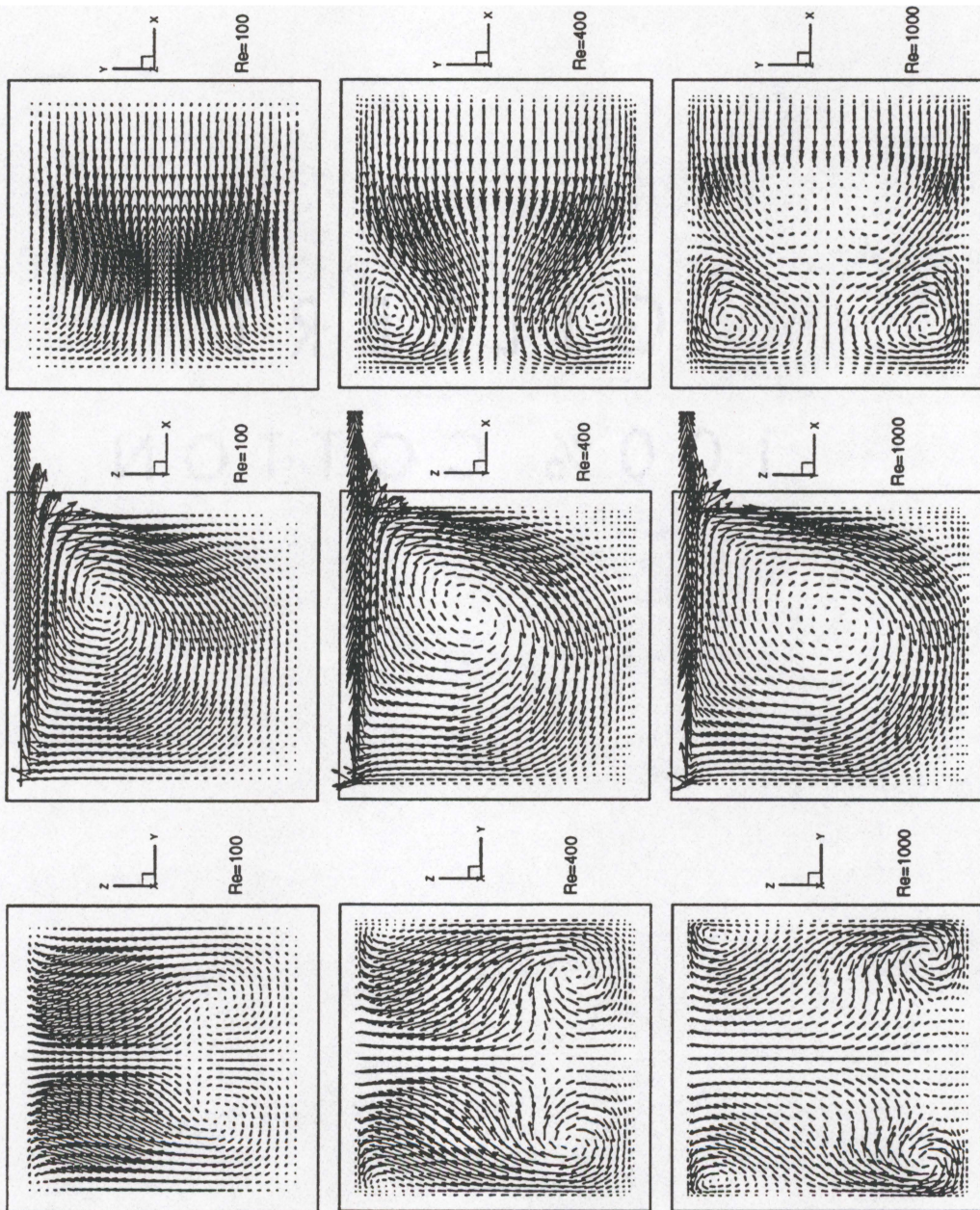


Figure 5.25 : 2-D planar projections of mid-plane velocity vector field for 3-D driven cavity, $Re = 100, 400, 1000$, $M=32^3$

There are three state variable members for vorticity in the three dimensional velocity-vorticity formulation, Ω_x , Ω_y , and Ω_z . Contour plots of these vorticities for $Re = 100$, $Re = 400$, and $Re = 1000$ on each mid-center plane are shown in Figure 5.26, Figure 5.27, and Figure 5.28. As Re increases, the vorticity contour lines become packed closer together due to the stiff boundary layer effect. The centers of vortices are observed to move towards the corners. For $Re = 1000$, the shape of the vorticity contour lines have changed significantly. They cluster closer to each other and become more complicated.

The u-velocity component distribution on the vertical plane centerline is used as a measure of accuracy for the 3-D lid-driven cavity benchmark. For $Re = 100$, the u-velocity component data reported by Ku et al. (1987), Guj and stella (1993), Fujima et al. (1994), and Jiang et al. (1994) agree quite well with each other. For the present solution, the flow field at $Re = 100$ is started from rest, and the standard GWS solution algorithm could produce the steady-state solution using a single large time step, $\Delta t = 1000$. The solution computed on the uniform mesh, shown as triangles in Figure 5.29, is seen to miss the well agreed accurate solution reported by Jiang et al. (1994). Using the same $M=32 \times 32 \times 32$ mesh, but now non-uniformly refined to the corners of the cube, the solution shown as circles in Figure 5.29 matches almost exactly with that given by Jiang et al. (1994). The maximum difference is 4.3 % at the peak of the u-velocity profile.

A timing test for the testcase is conducted on the CM5 with 32 processor nodes. It takes an average of 7.7 seconds of elapsed wall-clock time to complete one single

Figure 5.26 : Ω_x contour plot for 3-D driven cavity, $Re=100, 400, 1000, 1000, 400, 1000, M=32^3$

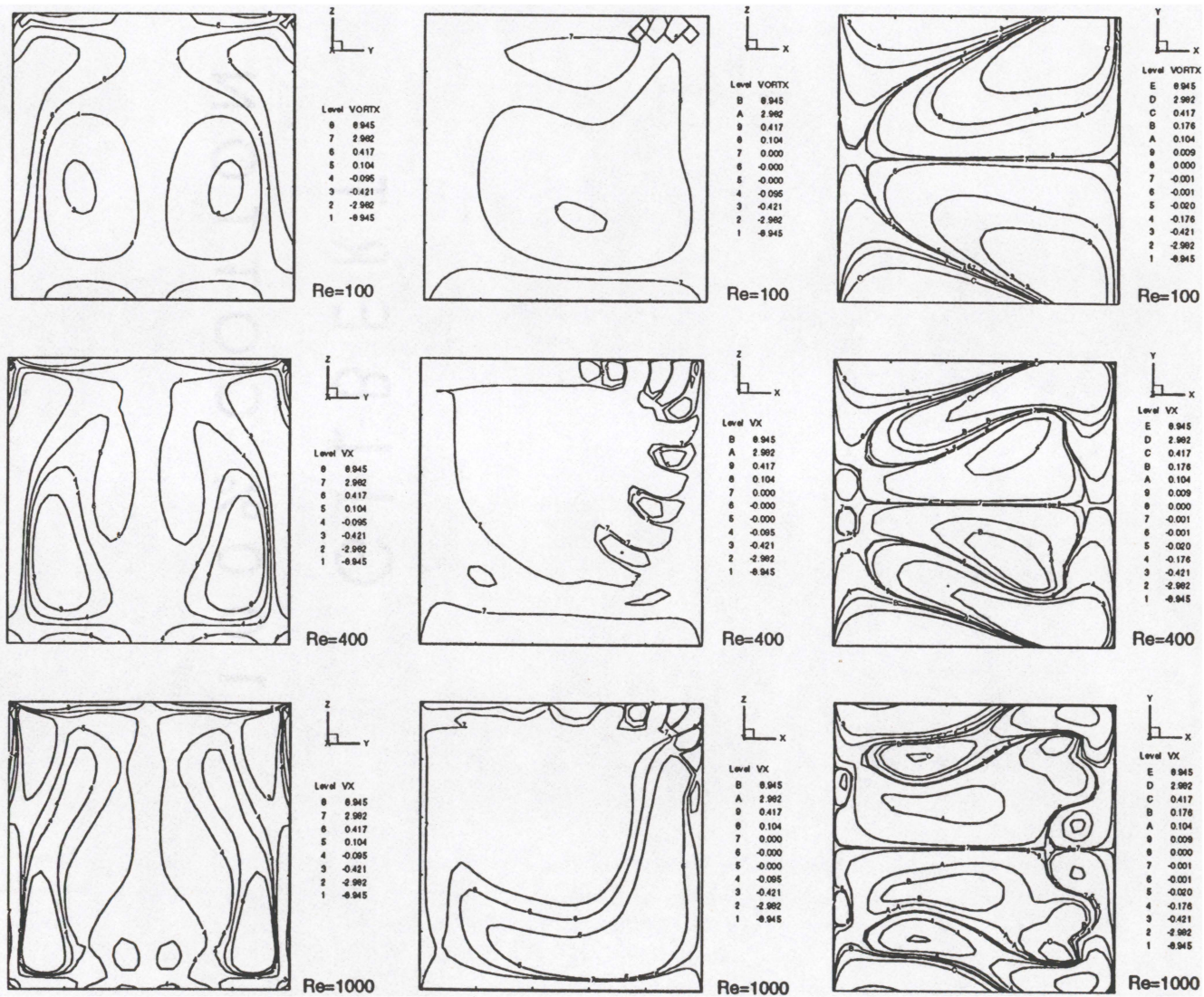


Figure 5.27 : Ω_y contour plot for 3-D driven cavity, $Re=100, 400, 1000, M=32^3$

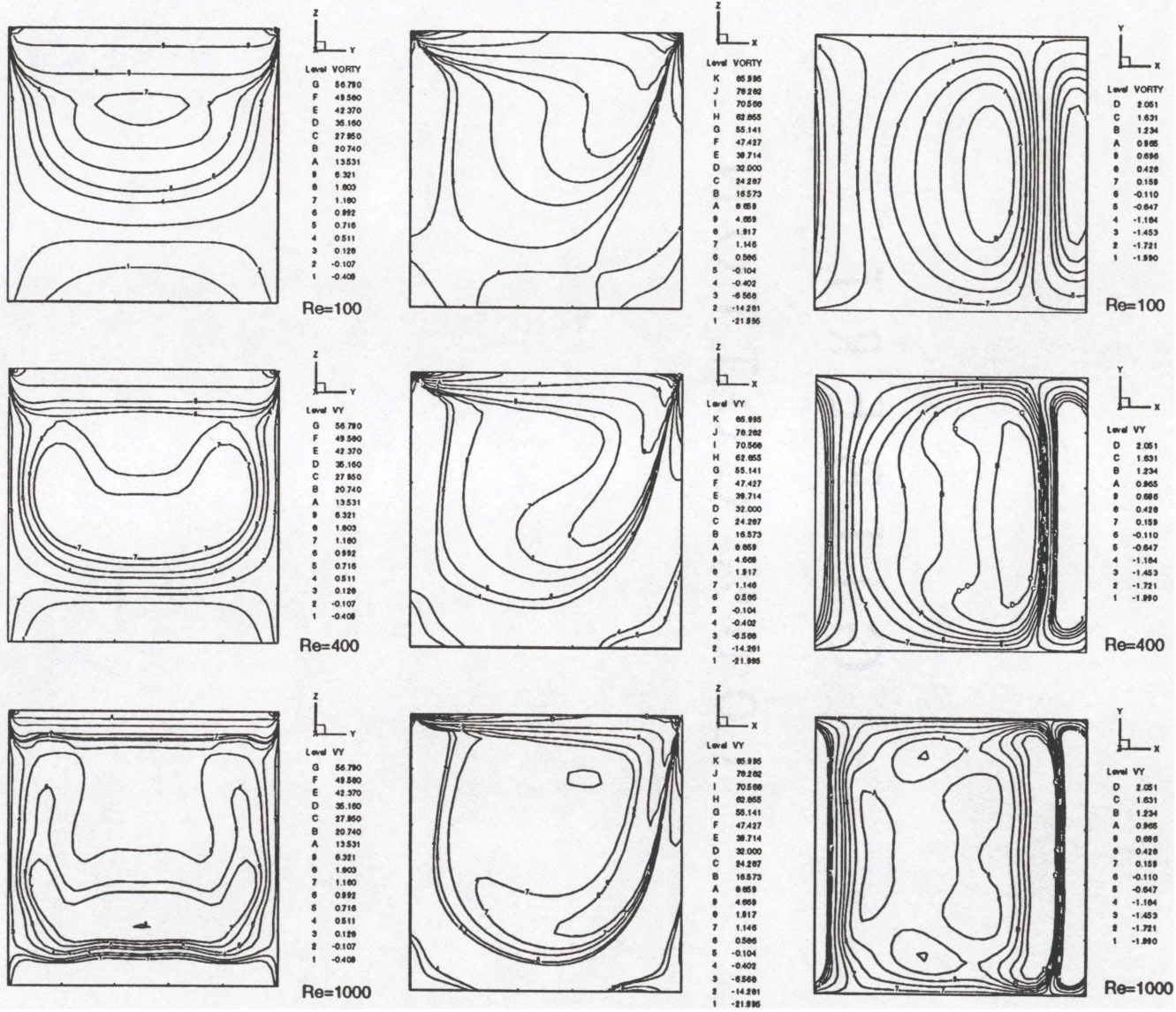
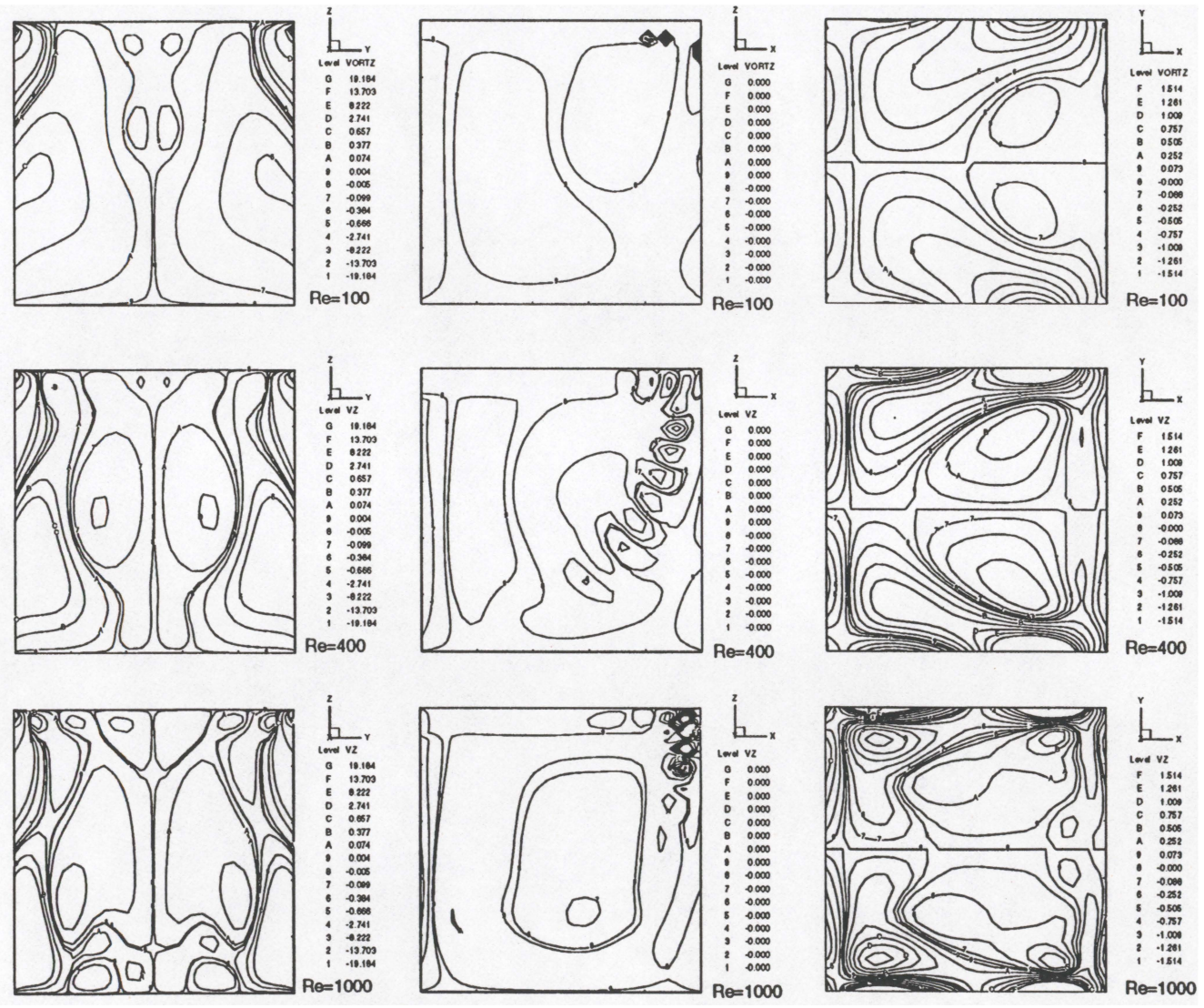


Figure 5.28 : Ω_z contour plot for 3-D driven cavity, $Re=100, 400, 1000, M=3^2^3$



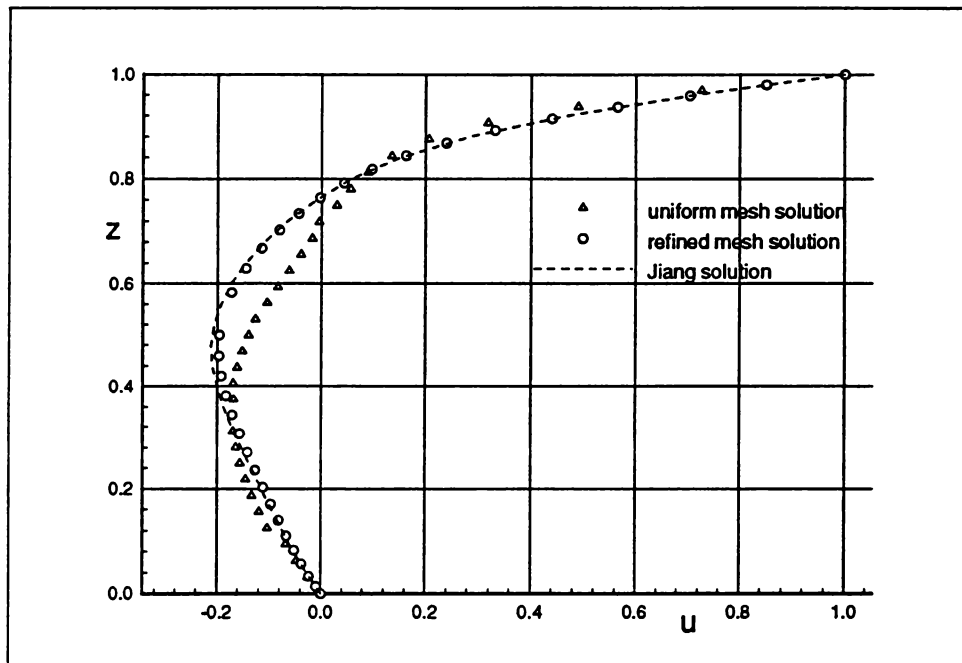


Figure 5.29 : 3-D mid-plane centerline u-velocity plot at $Re=100$

BiCGSTAB iteration with diagonal scaling preconditioner. An average of 125 seconds of elapsed time is needed to form and assemble the jacobian matrix and residual vectors for each Newton step. A fixed overhead of preprocessing and postprocessing requires approximately 4 minutes. For $Re = 100$, four Newton steps and a total of 377 BiCGSTAB iterations are required to converge the extremum Newton error, δQ_{max} to 6.26×10^{-3} occurring for Ω_y . Consequently, a total of 62 minutes is spent to achieve the steady state solution for $Re = 100$.

For $Re = 400$, the size of the time step is reduced to $\Delta t = 1.0$. After 25 time steps, the steady state is assumed with an extremum Newton residual of $\delta Q = 1.3 \times 10^{-2}$ for the vorticity components, which is nominally one order of magnitude

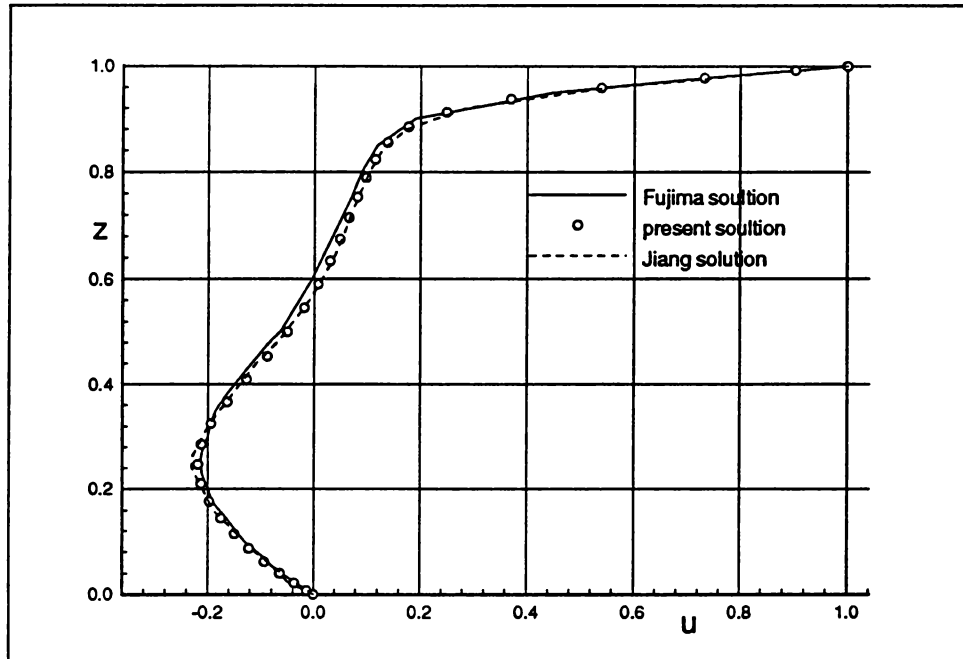


Figure 5.30 : 3-D mid-plane centerline u-velocity plot at $Re=400$

larger than that for the velocity components. The number of Newton iterations is 66 with 4679 BiCGSTAB iterations. The runtime of the testcase is calculated based on the total elapsed time accumulated from the total number of Newton iterations, the total number of BiCGSTAB iteration, and the overhead. A total of approximately 12.4 hours was required to obtain the solution for $Re = 400$ on the CM5. The centerline u-velocity profile of the present solution is compared with those of Jiang et al. (1994) and Fujima et al.(1994) in Figure 5.30. The negative peak u-velocity predicted by Jiang is 0.2341 while 0.2189 is predicted by the present solution.

For $Re = 1000$, the size of the time step is further reduced to $\Delta t = 0.4$. After 37 time steps, the steady state is assumed with an extremum $\delta Q = 3.3 \times 10^{-2}$, requiring

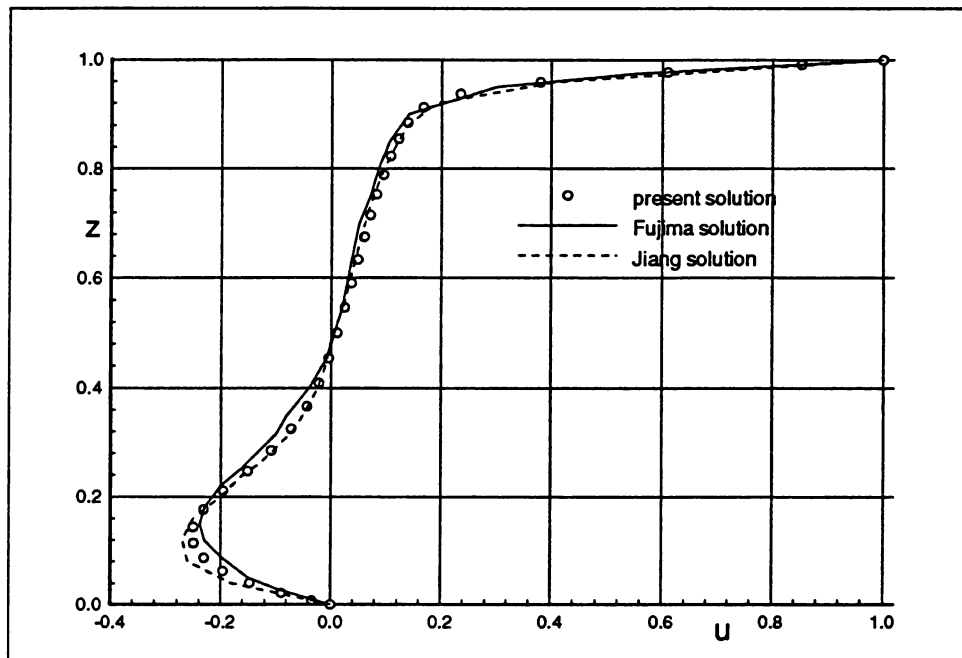


Figure 5.31 : 3-D mid-plane centerline u-velocity plot at $Re=1000$

170 newton steps and 20249 BiCGSTAB iterations. A total of approximately 49.3 hours was required to obtain the solution for $Re = 1000$ on the CM5. The present solution is compared with those of Jiang et al. (1994) and Fujima et al. (1994) in Figure 5.31. The negative peak u-velocity predicted by Jiang is -0.275. The present value is -0.250, which is in better agreement than the data of Fujima (1994).

A mesh sensitivity analysis conducted by Guj and Stella (1993), summarized in Table 5.4, shows the same magnitude of error for their computed solution using the velocity-vorticity formulation with uniform mesh refinement. The minimum centerline u-velocity component for their $M=31 \times 31 \times 32$ mesh is -0.22. For the $M=101 \times 101 \times 82$ mesh, the minimum centerline u-velocity is -0.27, which is within 2% of that reported

Table 5.4 : Centerline minimum u-velocity by Guj and Stella (1993) at $Re=1000$

Equivalent Mesh Size	centerline u_{min}	% error
31x31x32	-0.2203	21.9
45x45x38	-0.2455	12.9
67x67x54	-0.2629	6.9
101x101x82	-0.2725	3.4
extrapolated solution	-0.2820	base solution (error free)

by Jiang et al. (1994) for a uniform $M=50 \times 52 \times 50$ mesh. Due to the limited memory resources on the CM5 at the University of Tennessee, only a 32 cube mesh lid-driven cavity solution could be obtained. Note that the present value of -0.25 is an improvement over the comparison mesh ($M=31 \times 31 \times 32$) solution of Guj and Stella (1993). The values of u-velocity along the vertical line through the geometric center of the 3-D driven cavity for $Re = 100$, $Re = 400$, and $Re = 1000$ are given in Table 5.5.

5.4 Thermal Cavity

Buoyancy driven flow due to natural convection in an enclosed 3-D rectangular box is the direct extension of the 2-D thermal cavity benchmark. Potential temperature boundary conditions are one and zero on the front and back walls, along the x-axis, while both side walls along the y-axis and the floor and ceiling are adiabatic,

Table 5.5 : U-velocity along the vertical centerline of the 3-D driven cavity

z location	Re = 100	Re = 400	Re = 1000
1.00	1.0	1.0	1.0
0.9921	0.9427	0.9031	0.8529
0.9779	0.8383	0.7319	0.6093
0.9594	0.7076	0.5391	0.3802
0.9375	0.5651	0.3693	0.2338
0.9126	0.4249	0.2489	0.1667
0.8851	0.2983	0.1775	0.1386
0.8553	0.1912	0.1387	0.1216
0.8232	0.1041	0.1156	0.1070
0.7890	0.0340	0.0981	0.0938
0.7529	-0.0232	0.0821	0.0817
0.7149	-0.0713	0.0659	0.0702
0.6752	-0.1124	0.0488	0.0591
0.6338	-0.1468	0.0297	0.0479
0.5907	-0.1741	0.0075	0.0363
0.5461	-0.1932	-0.0190	0.0241
0.5	-0.2033	-0.0513	0.0106
0.4538	-0.2041	-0.0885	-0.0044
0.4092	-0.1973	-0.1273	-0.0220
0.3661	-0.1853	-0.1640	-0.0440
0.3247	-0.1702	-0.1938	-0.0723
0.2850	-0.1535	-0.2127	-0.1085
0.2470	-0.1364	-0.2189	-0.1512
0.2109	-0.1195	-0.2129	-0.1950
0.1767	-0.1031	-0.1974	-0.2308
0.1446	-0.0873	-0.1755	-0.2502
0.1148	-0.0719	-0.1498	-0.2496
0.0873	-0.0571	-0.1222	-0.2305
0.0625	-0.0427	-0.0936	-0.1957
0.0405	-0.0290	-0.0648	-0.1470
0.0220	-0.0164	-0.0374	-0.0896
0.0078	-0.0060	-0.0138	-0.0342
1.0	0.0	0.0	0.0

Figure 5.32. Velocity boundary conditions are all no slip, and the kinematic vorticity boundary conditions are imposed at every boundary node.

The first numerical solution of the 3-D thermal cavity problem is reported by Mallinson & de Vahl Davis (1977). Using the Continuity Constraint Method, Williams (1993) presents a solution for the so called "window cavity problem" of Mallinson and de Vahl Davis (1977) at $Ra = 1.5 \times 10^5$. Reddy, Reddy & Akay (1992) present a thermal cavity solution for $Ra = 10^4$ using the penalty finite element method. A numerical study of 3-D natural convection for air in the cubic enclosure is reported by Fusegi, Hyun, Kuwahara & Farouk (1991) for Rayleigh number from 10^3 to 10^6 using a control-volume based finite difference staggered mesh procedure together with the pressure correction algorithm, SIMPLE (Patankar 1980).

The three velocity Poisson equations, three momentum equations with the Boussinesq buoyancy term, and the energy equation, are solved simultaneously. For the non-uniform cartesian $M=32 \times 32 \times 32$ mesh with 35937 nodes, Figure 5.32 the total number of equations is 251559. In the present nondimensionalization, $Gr = Re^2$ and $Ra = Re^2 Pr$. The Rayleigh numbers tested are $Ra = 1.0 \times 10^4$, $Ra = 1.6 \times 10^5$, $Ra = 1.0 \times 10^6$, and $Ra = 1.6 \times 10^7$ with a fixed Prandtl number of 1.0. Select 3-D sectional perspective views of the velocity vector and temperature solution distributions for various Rayleigh numbers are shown in Figures 5.33 - 5.36. For $Ra = 1.6 \times 10^7$, the mesh of the computational domain is further refined to the walls to capture the thermal boundary layer effects, Figures 5.37.

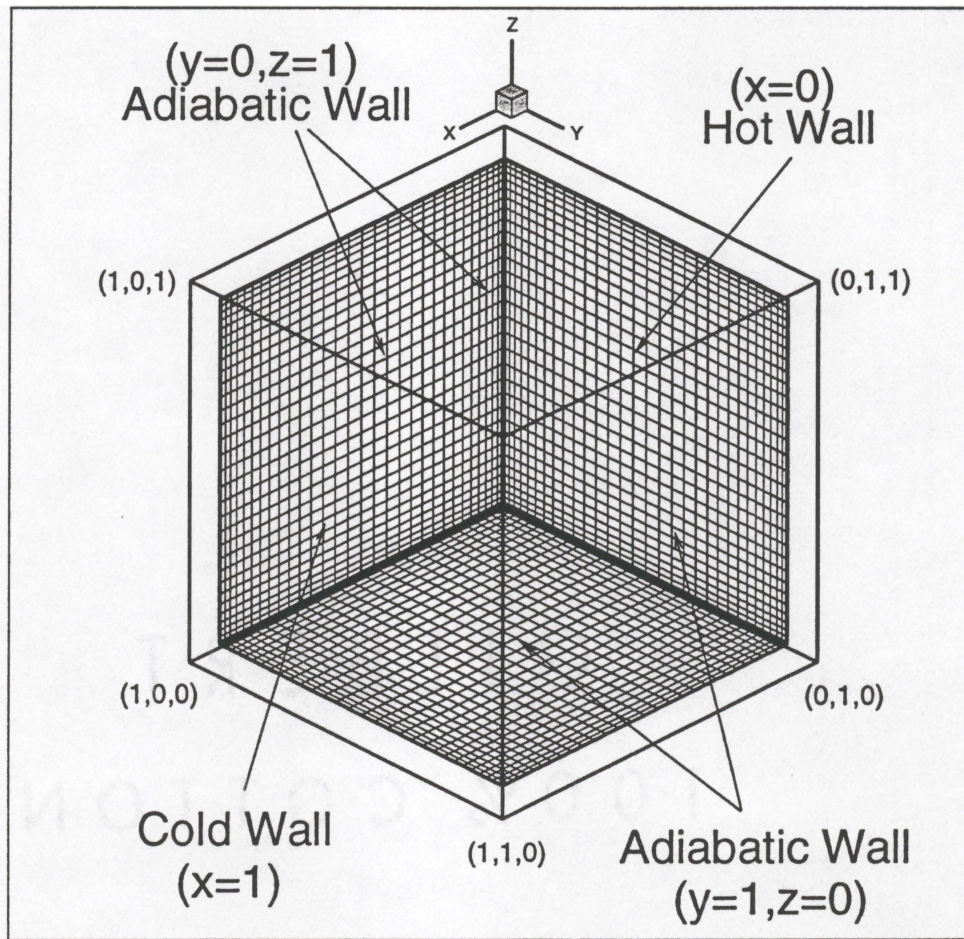


Figure 5.32 : Non-uniform finite element mesh for 3-D thermal cavity, $M=32^3$

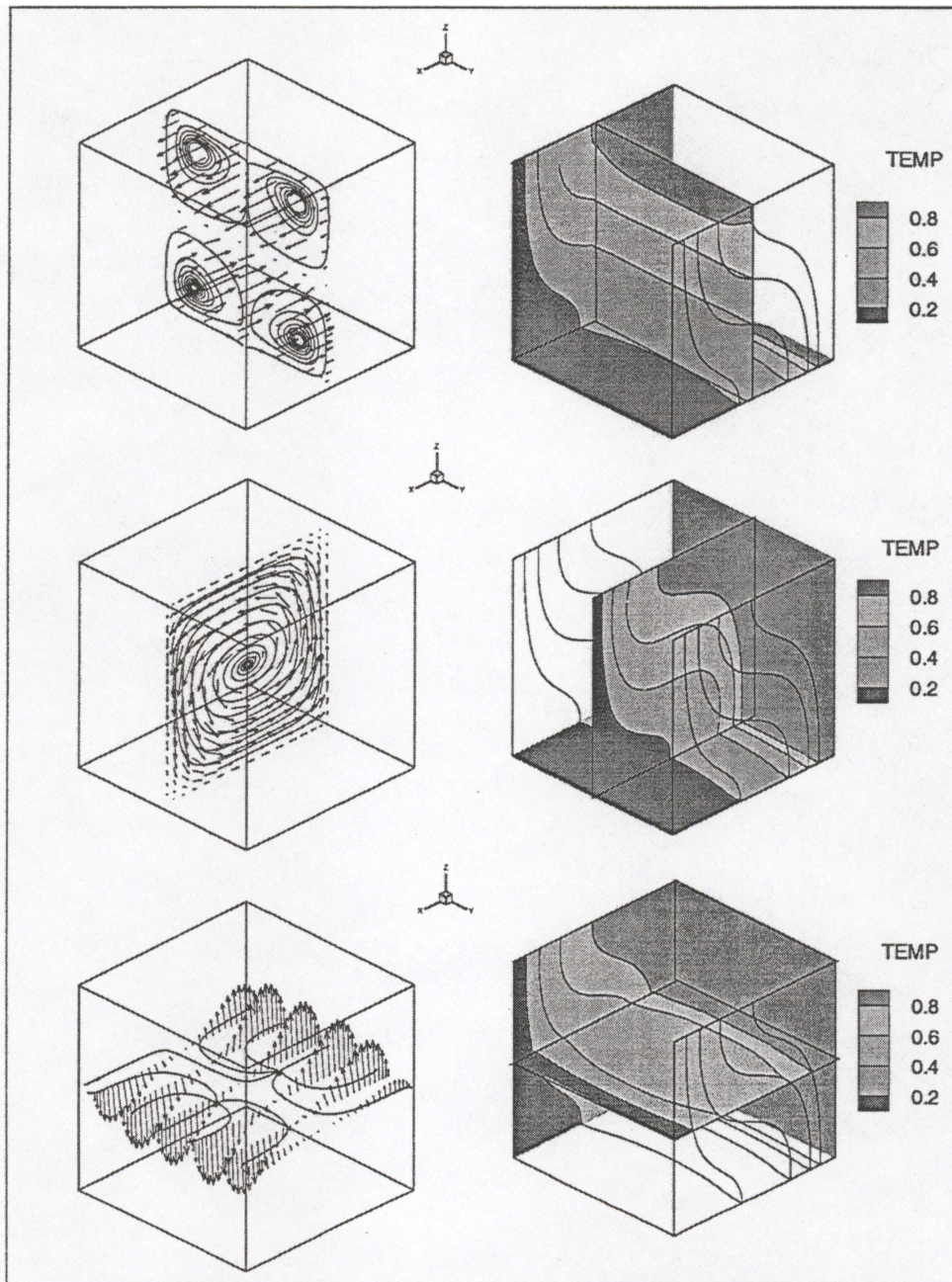


Figure 5.33 : Velocity vector and temperature contour at $Ra = 10^4$ for 3-D thermal cavity

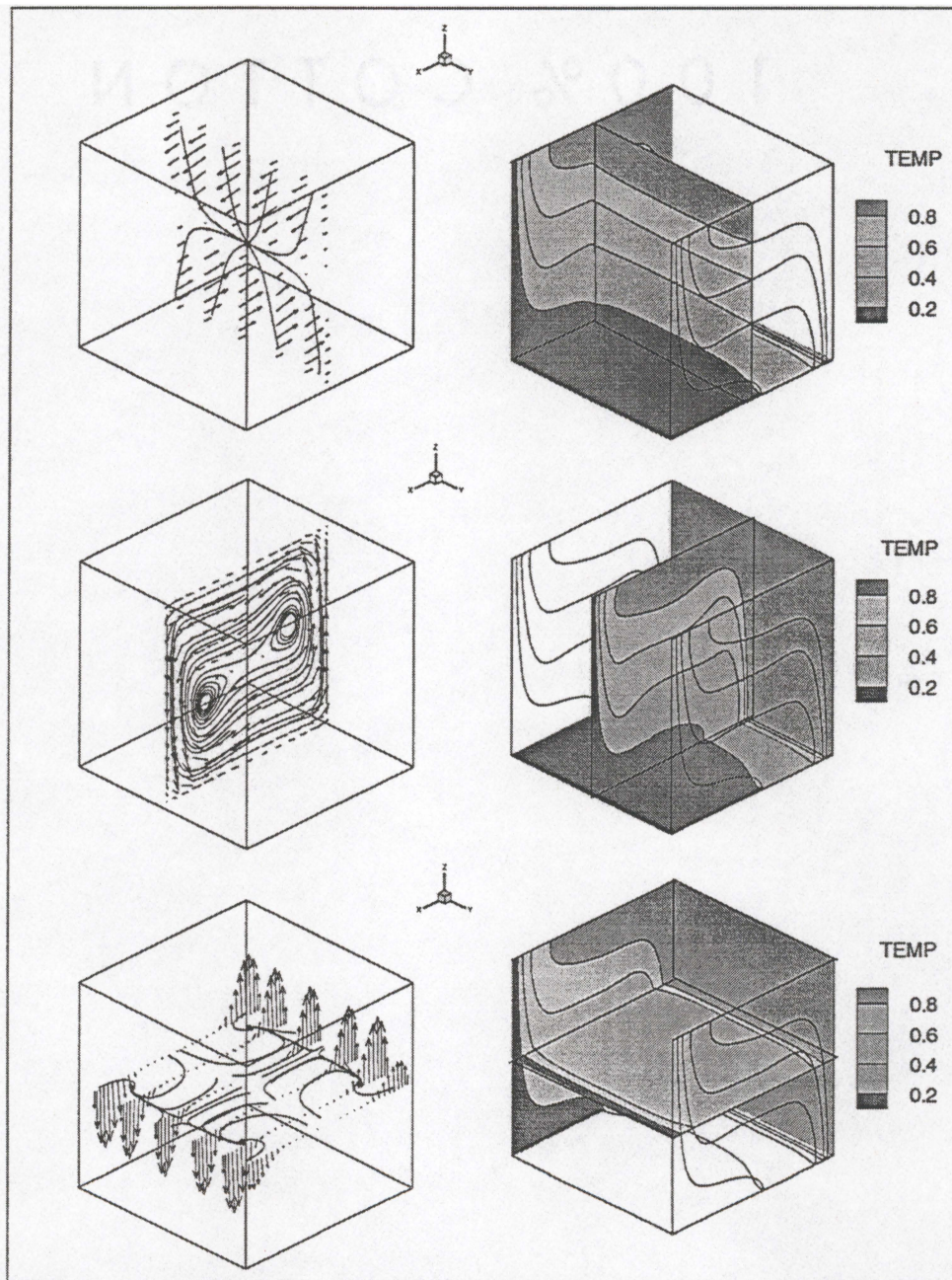


Figure 5.34 : Velocity vector and temperature contour at $Ra = 1.6 \times 10^5$ for 3-D thermal cavity

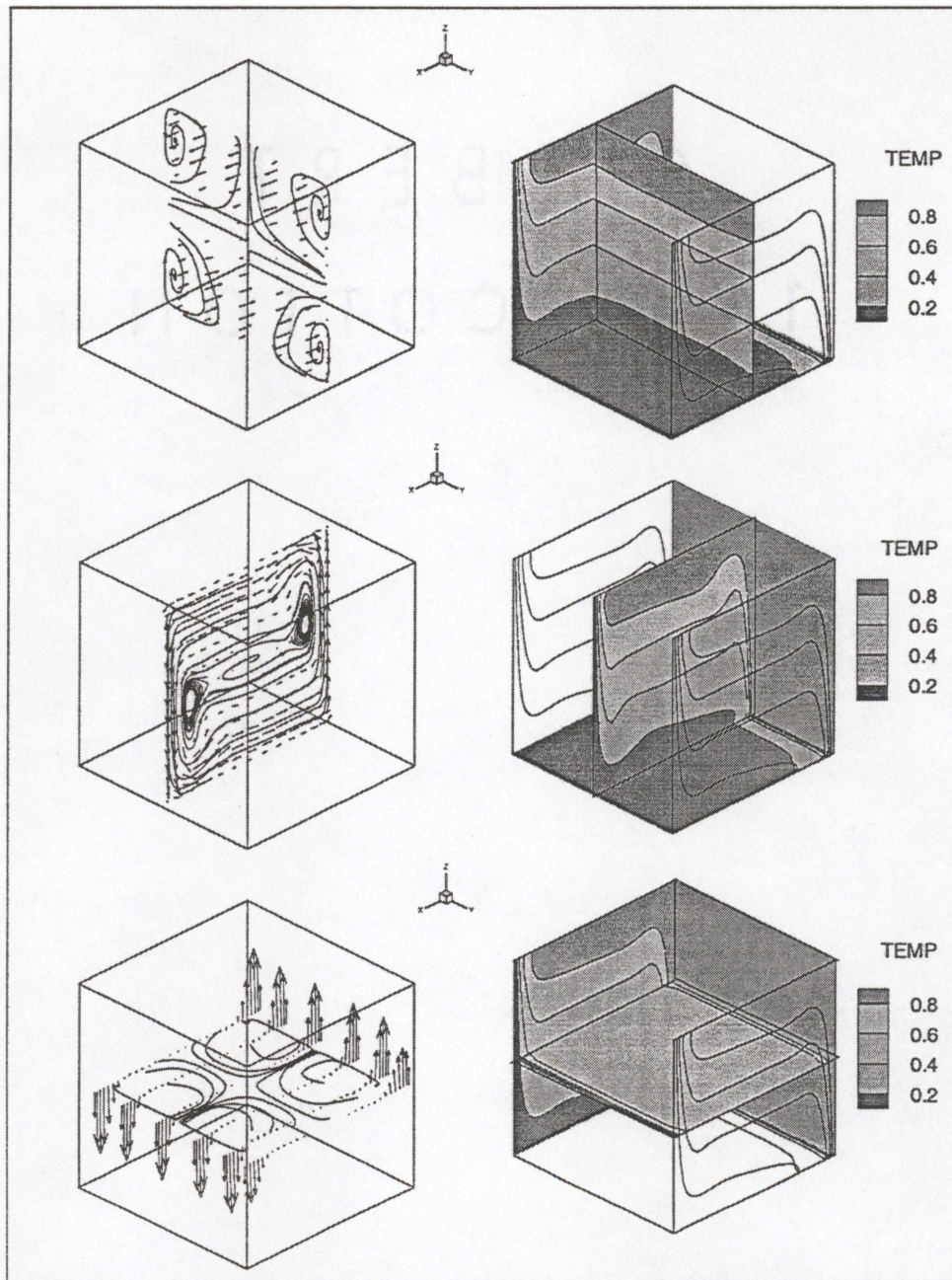


Figure 5.35 : Velocity vector and temperature contour at $Ra = 10^6$ for 3-D thermal cavity

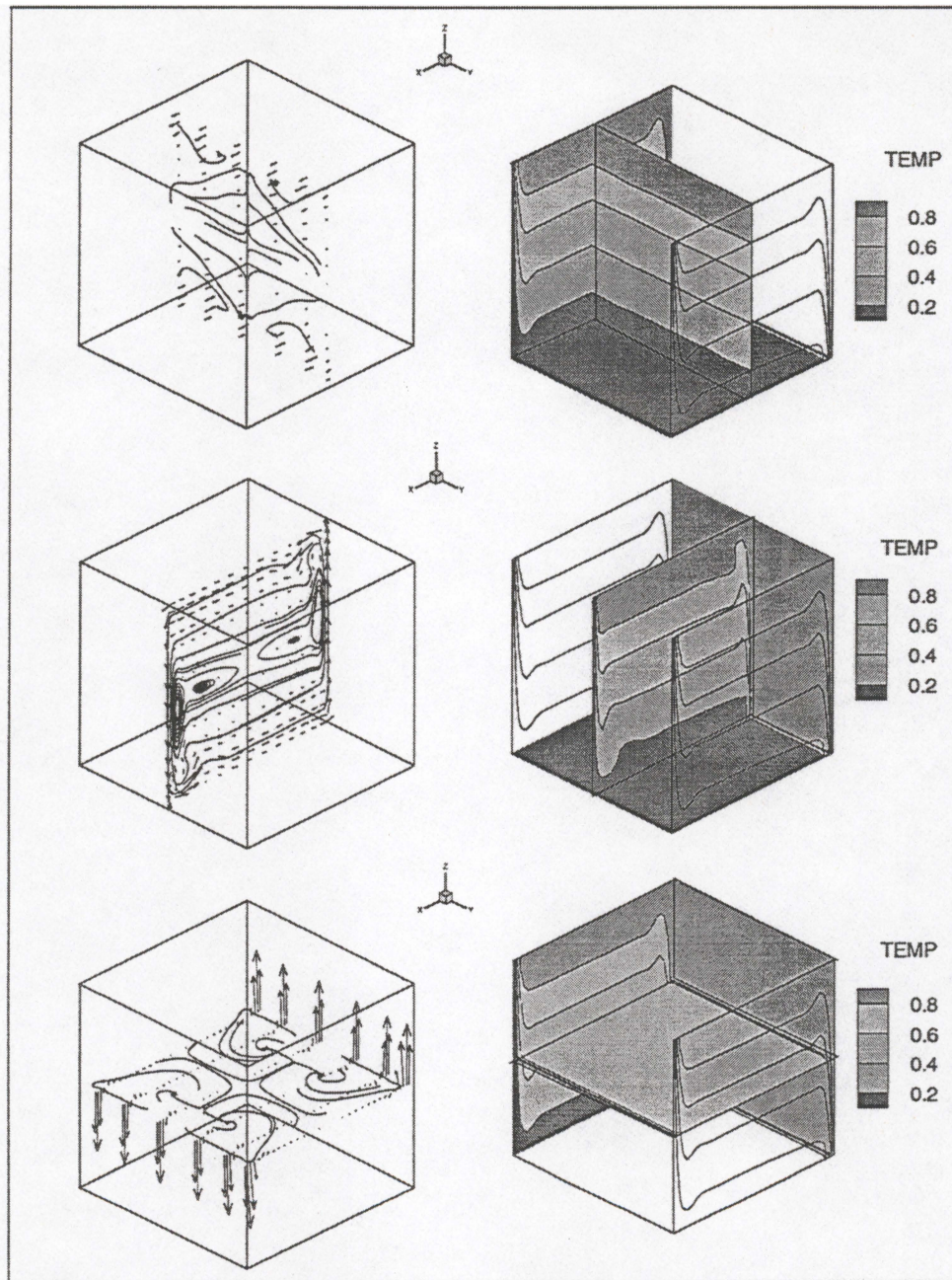


Figure 5.36 : Velocity vector and temperature contour at $Ra = 1.6 \times 10^7$ for 3-D thermal cavity, $M = 32^3$, original mesh

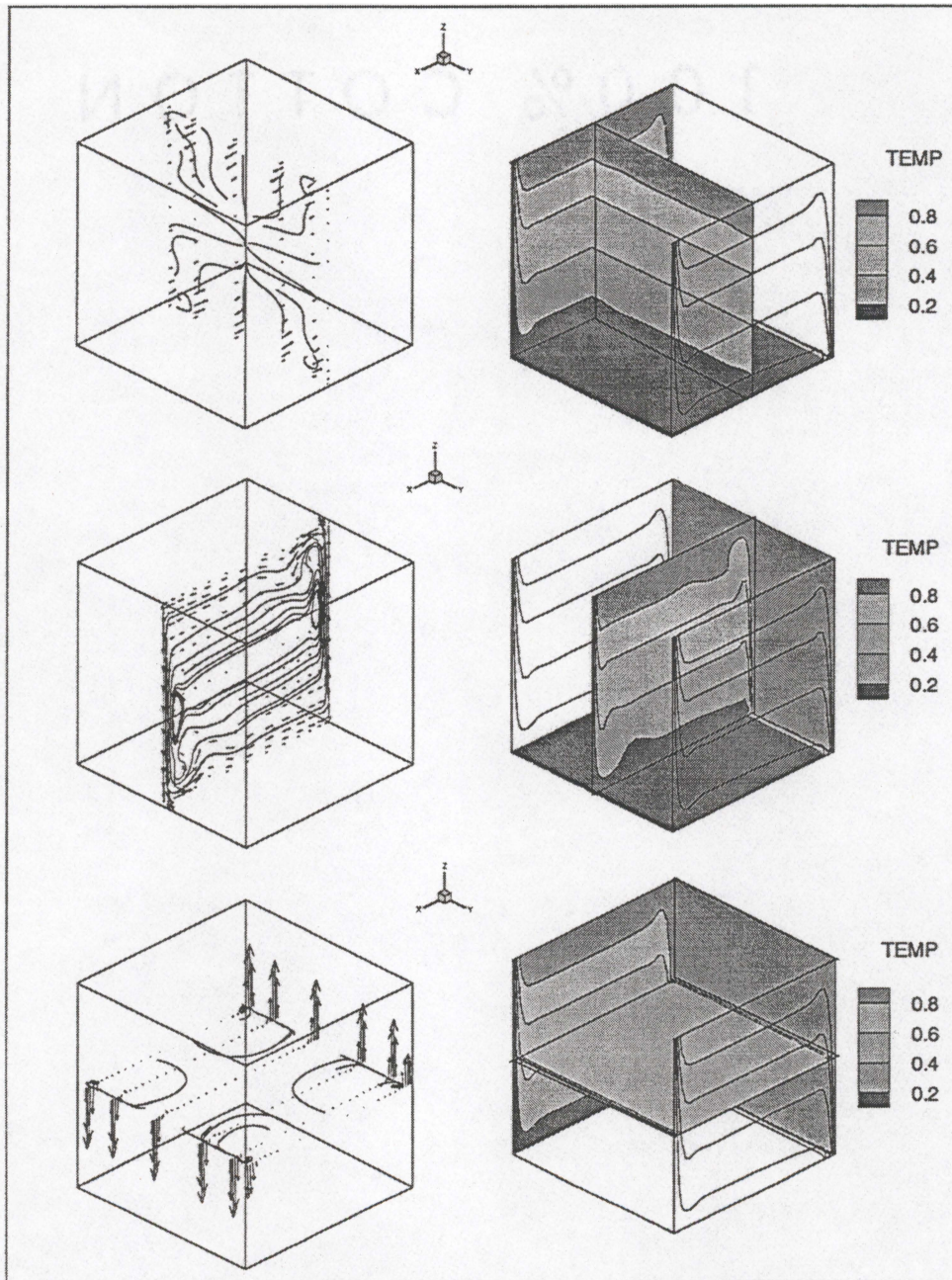


Figure 5.37 : Velocity vector and temperature contour at $Ra = 1.6 \times 10^7$ for 3-D thermal cavity, $M = 32^3$, refined to fixed temperature walls

A timing test indicates that it takes an average of 9.2 seconds of elapsed wall-clock time to complete one BiCGSTAB iteration and 137 seconds to form and assemble the jacobian matrix and residual vectors for each Newton step. For $Ra = 10^4$, the computation is started with a zero initial solution field. For the large $\Delta t = 500$, 6 Newton steps and 464 BiCGSTAB iterations are taken to achieve a solution converged to extremum $\delta Q = 3.9 \times 10^{-3}$. Consequently, a total of 1.48 hours was used. The two dimensional planar projections of the velocity vector field for $Ra = 10^4$ at the symmetry plane reveals only one single recirculation cell, Figure 5.33.

For $Ra = 1.6 \times 10^5$, the solution for $Ra = 10^4$ is used as the initial condition. A single integration step at $\Delta t = 500$ is again used, and 6 Newton steps and 581 BiCGSTAB iterations are needed to converge to extremum $\delta Q = 4.2 \times 10^{-3}$. In the symmetry plane, two recirculation cells are now observed, Figure 5.34.

For $Ra = 10^6$, the previous result is used as the initial condition. The size of the time step is reduced to $\Delta t = 5$. After 7 time steps, the steady state solution is assumed with extremum $\delta Q = 1.2 \times 10^{-2}$, requiring 18 Newton iterations and 1465 BiCGSTAB iterations. The recirculation cells at the symmetry plane are moving towards the corners, and the isotherms are packing closer to the Dirichlet walls, Figure 5.35.

For $Ra = 1.6 \times 10^7$, the size of the time step is further reduced to $\Delta t = 0.5$. The steady-state solution is assumed after 53 time steps with the solution for $Ra = 10^6$ as the initial condition. The extremum iterate at convergence is $\delta Q = 5.1 \times 10^{-2}$ for Ω_y , with 157 Newton steps and 10006 BiCGSTAB iterations used. Accounting for

the time to obtain the initial condition, a total of approximately 39.1 wall-clock hours are required to attain this solution on the CM5. The effect of the thermal boundary layer to the flow field is obvious, Figure 5.36 and Figure 5.37. Recirculation cells and isotherms are further compressed towards the fixed temperature walls.

The summary streamline plot with projected velocity on the symmetry plane for various Rayleigh numbers is shown in Figure 5.38. It shares close resemblance to that of the 2-D thermal cavity reported by Williams (1993) and Henkes & Hoogendoorn (1993).

At high Rayleigh number, isotherms with sharp gradient near the walls are observed. The effect of the thermal boundary layer is dominating the flow field. Recirculation cells cluster near the walls and move towards the corners. The effect of increasing Rayleigh number on the temperature profile, the u-velocity and the w-velocity components on the symmetry plane is shown in Figure 5.39 and Figure 5.40. The temperature profiles at various vertical locations (z-coordinate) on the symmetry plane for $Ra = 10^6$ are shown in Figure 5.41.

A case study for air, $Pr = 0.71$, in the 3-D thermal cavity is also conducted for $Ra = 10^5$. The temperature profile in the centerline of the symmetry plane is compared with the results reported by Fusegi et al. (1991), Figure 5.42. The agreement between the solutions is good. A maximum u-velocity of 0.1468 at z (vertical) location of 0.1453 in the symmetry plane was reported by Fusegi et al. (1991) with a non-uniform $M=62 \times 62 \times 62$ mesh. The present formulation yields a maximum u-velocity of 0.1415 at $z=0.1453$, a 3.6 % difference.

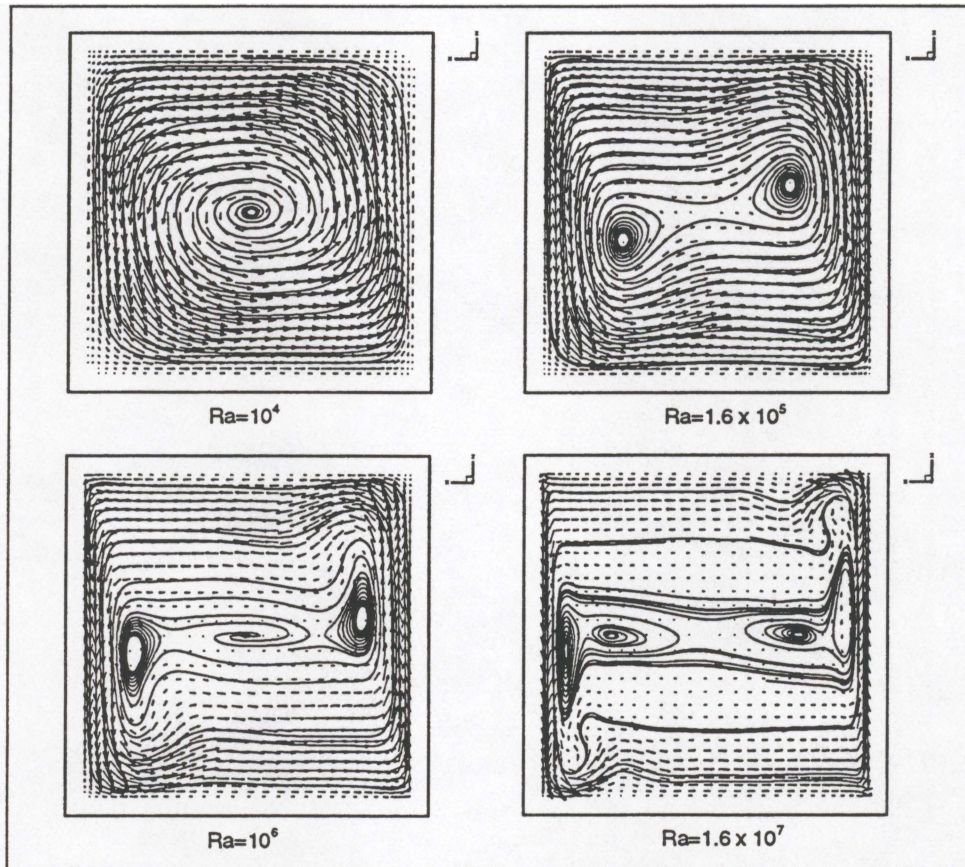


Figure 5.38 : Streamlines for the projected velocity vector field on the symmetry plane $Ra = 10^4 - 1.6 \times 10^7, M = 32^3$

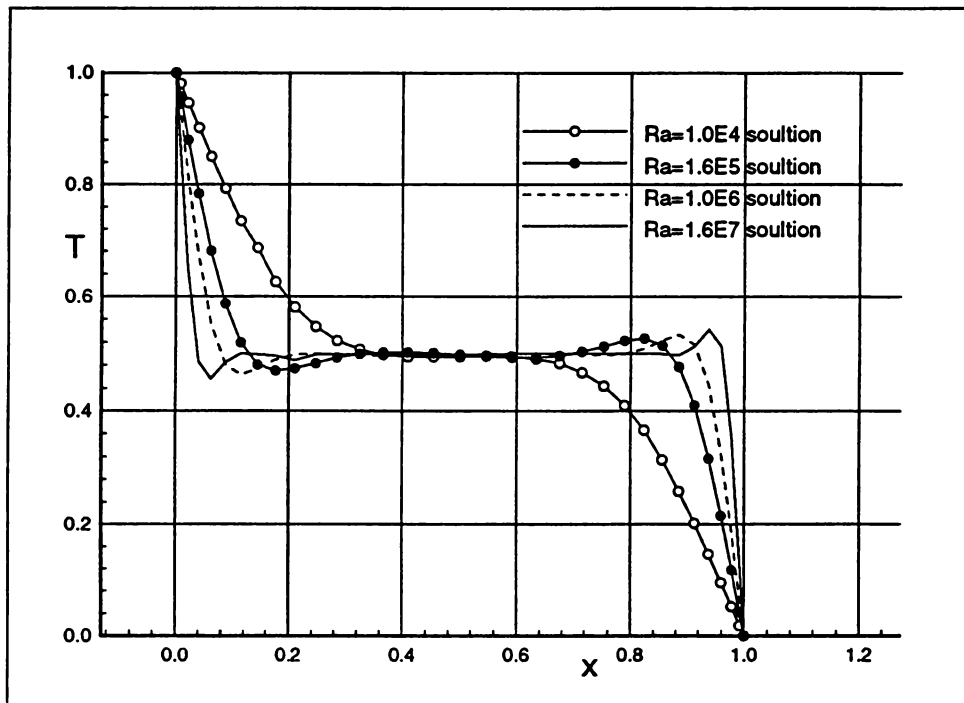


Figure 5.39 : Temperature profiles in the centerline of the symmetry plane for

$Ra = 10^4$, $Ra = 1.6 \times 10^5$, $Ra = 10^6$, and $Ra = 1.6 \times 10^7$

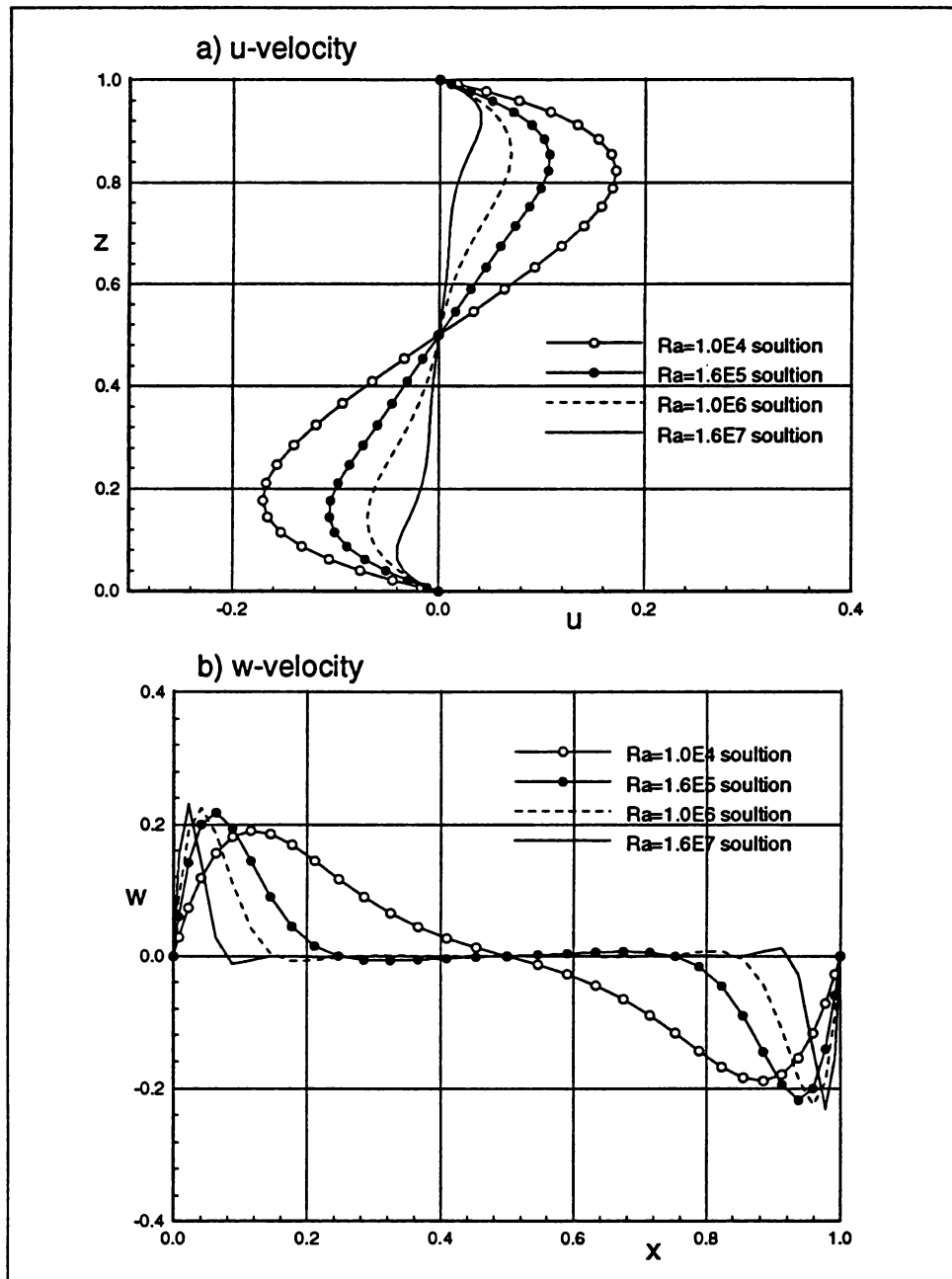


Figure 5.40 : Centerline velocity profiles of the symmetry plane for $Ra = 10^4$, $Ra = 1.6 \times 10^5$, $Ra = 10^6$, and $Ra = 1.6 \times 10^7$

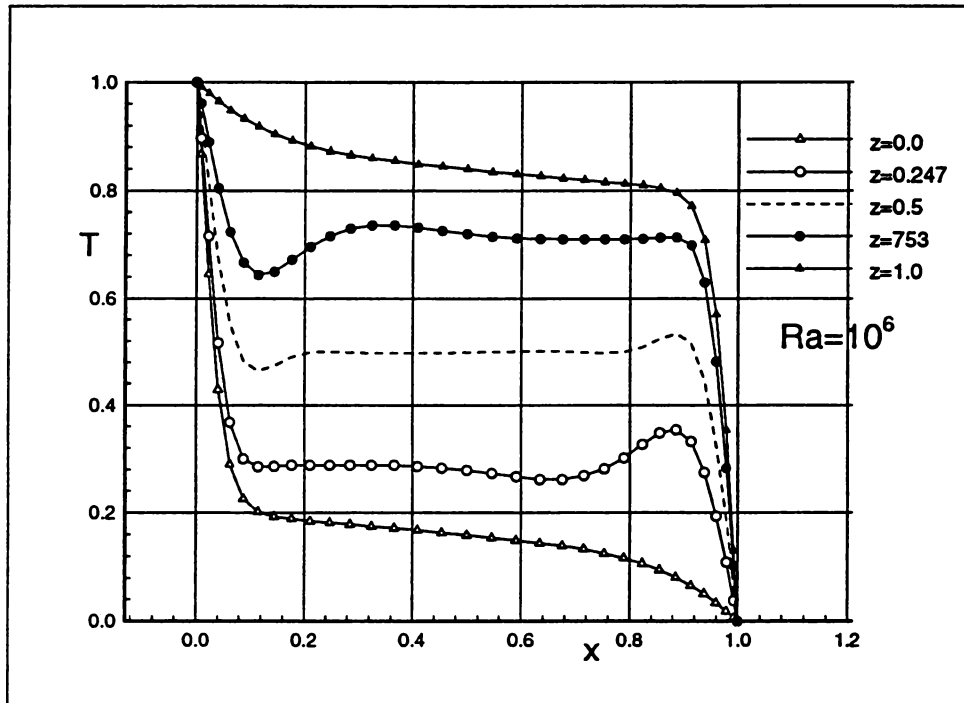


Figure 5.41 : Temperature profile for $Ra = 10^6$ at various z location on the symmetry plane

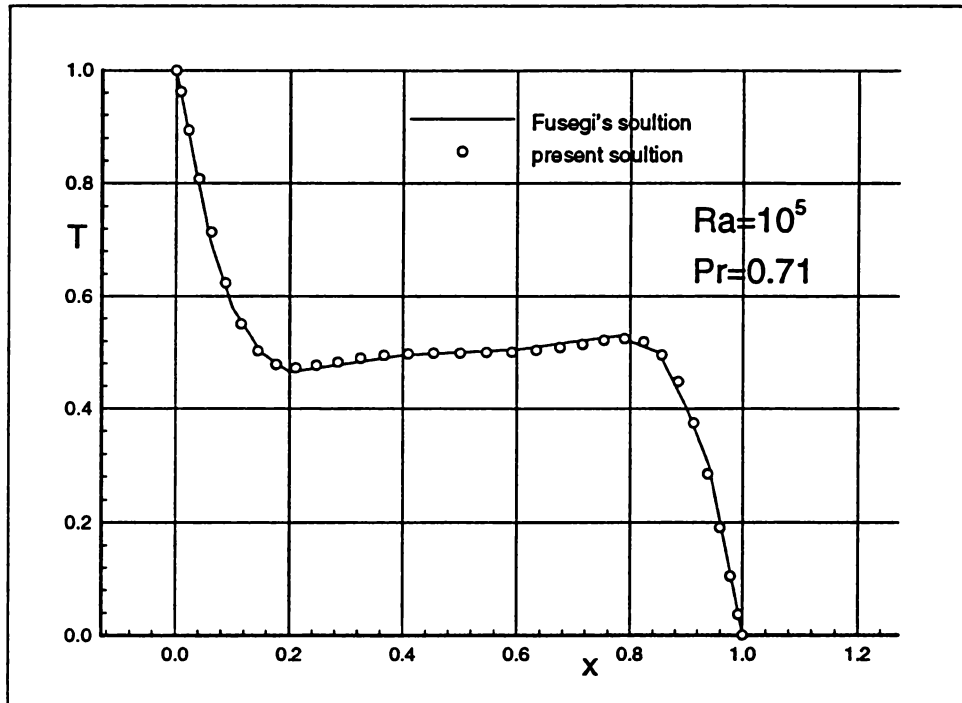


Figure 5.42 : Temperature profile for $Ra = 10^5$ for air in the centerline of the symmetry plane

SUMMARY AND CONCLUSIONS

A new parallel finite element CFD algorithm in velocity-vorticity form has been developed to produce solutions for the unsteady, laminar incompressible Navier-Stokes equations in three dimensions. The incompressibility constraint is enforced through a new second order kinematic vorticity boundary condition imposed at nodes on no slip walls. Such kinematic vorticity boundary conditions are shown to be consistent with the mathematical well-posed system proved by Quartapelle (1993) and Duabe (1992).

The fully coupled formulation, which contains six variables for an isothermal system, and seven variables if the energy equation is included, is solved in parallel on a Thinking Machines CM5 parallel computer with 32 processors. The BiCGSTAB sparse iterative solver is incorporated to yield an efficient parallel finite element algorithm. An implicit time integration scheme allows the use of a large time step to achieve a fast steady-state solution for $Re \leq 100$ or $Ra \leq 10^5$. Reducing the size of the time step can improve the condition number of the terminal Newton jacobian matrix for high Reynolds or Rayleigh number solutions, which has been used in conducting this research.

As original contributions to incompressible CFD, this dissertation has presented:

1. a time accurate 3-D finite element incompressible Navier-Stokes solver with equal order interpolation functions for which pressure boundary conditions are not necessary,
2. a mathematically consistent vorticity kinematic boundary condition for the no-slip wall,
3. a viable strategy to obtain convergent and accurate solution for high Reynolds and Rayleigh number fluid flow problem in conjunction of the CG-type sparse iterative solvers,
4. a CM5 efficient parallel finite element CFD code for non-pressure difference driven flow simulation,
5. an implementation of sparse data structures with matrix-vector multiplication kernel highly optimized for the finite element method,
6. a verification and benchmarking study of the velocity-vorticity formulation for isothermal and nonisothermal flows in three dimensions.

Due to the limitation of the size of the CM5 at the University of Tennessee, a convergence analysis via progression to refined mesh was not conducted. Therefore, various non-uniform meshings were used to assess error control. Scalability and speed up analyses could also be performed if a larger machine was available. The basic $M=32 \times 32 \times 32$ mesh used in the 3-D benchmarking, however limited, is able to produce solutions of engineering-adequate accuracy for the driven cavity and thermal cavity

problems at high Reynolds number and Rayleigh numbers. The obvious extension of the formulation for the kinematic vorticity condition is to encompass flows in arbitrary geometries. However, the simplicity of the velocity-vorticity formulation, bearing with the shortfall to solve a fully coupled Newton system, is proven to provide a mathematically viable alternative to a primitive variables, CFD-approximate pressure formulation.

BIBLIOGRAPHY

Bibliography

- Agarwal, R. k. (1973), Third-Order-Accurate Upwind Scheme for Navier-Stokes Solutions in Three Dimensions, *in* K. N. Ghia, T. J. Mueller & B. Patel, eds, 'Computers in Flow Predictions and Fluid Dynamics Experiments', ASME, New York, pp. 73–82.
- Aregbesola, Y. A. S. & Burley, D. M. (1977), 'The Vector and Scalar Potential Method for the Numerical Solution of Two- and Three-Dimensional Navier-Stokes Equations', *Journal of Computational Physics* **24**, 398–415.
- Babu, V. & Korpela, S. A. (1994), 'Numerical Solution of the Incompressible, Three-Dimensional Navier-Stokes Equations', *Computers & Fluids* **23**, 675–691.
- Babuska, I. (1973), 'The Finite Element Method with Lagrange Multipliers', *Numer. Math.* **20**, 179–192.
- Baker, A. J. (1973), 'Finite Element Solution Algorithm for Viscous Incompressible Fluid Dynamics', *International Journal for Numerical Methods in Fluids* **6**, 89–101.
- Baker, A. J. (1983), *Finite Element Computational Fluid Mechanics*, Hemisphere Publishing Corporation, New York.

- Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. & Van der Vorst, H. (1994), *TEMPLATES for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia.
- Batchelor, G. K. (1967), *An Introduction to Fluid Dynamics*, Cambridge University Press, Cambridge.
- Brezzi, F. (1974), 'On the Existence, Uniqueness, and Approximation of Saddle-Point Problems Arising from Lagrange Multipliers', *RAIRO Anal. Numer.* **8**, 129–151.
- Chorin, A. J. (1968), 'Numerical Solution of the Navier-Stokes Equations', *Mathematics of Computation* **22**, 745–762.
- CM5 User's Guide, Version 1.0* (1993), Thinking Machines Corporation, Cambridge, MA.
- Dacles, J. & Hafez, M. (1990), 'Numerical Methods for 3-D Viscous Incompressible Flows Using Velocity Vorticity Formulation', *AIAA paper*, *AIAA-90-0237*.
- Daube, O. (1992), 'Resolution of the 2D Navier-Stokes Equations in Velocity-Vorticity Form by Means of an Influence Matrix Technique', *Journal of Computational Physics* **103**, 402–414.
- Dennis, S. C. R., Ingham, D. & Cook, R. N. (1979), 'Finite-difference Methods for Calculation Steady Incompressible Flows', *Journal of Computational Physics* **33**, 325–339.

- Dias da Cunha, R. & Hopkins, T. (1992), *PIM 1.0 : The Parallel Iterative Methods Package for Systems of Linear Equations, User's Guide (Fortran 77 Version)*, University of Kent at Canterbury.
- Farouk, B. & Booz, O. (1984), 'Numerical Investigation of Supercritical Taylor-Vortex Flow for a Wide Gap', *Journal of Fluid Mechanics* **138**, 21–27.
- Fasel, H. (1976), 'Investigation of the Stability of Boundary Layers by a Finite-Difference Model of the Navier-Stokes Equations', *Journal of Fluid Mechanics* **78**, 355–383.
- Fletcher, R. (1976), Conjugate Gradient Methods for Indefinite Systems, in 'Lecture Notes Math.', Vol. 506, Springer-Verlag, New York, pp. 73–89.
- Freund, R., Golub, G. & Nachtigal, N. (1991), *Iterative Solution of Linear Systems*, Tech. Report NA-91-05, Stanford University, Stanford, CA.
- Fujima, S., Tabata, M. & Fukasawa, Y. (1994), 'Extension to Three Dimensional Problems of the Upwind Finite Element Scheme Based on the Choice of Up- and Downwind Points', *Computer Methods in Applied Mechanics and Engineering* **112**, 109–132.
- Fusegi, T., Hyun, J. M., Kuwahara, K. & Farouk, B. (1991), 'A Numerical Study of Three-Dimensional Natural Convection in a Differentially Heated Cubical Enclosure', *International Journal of Heat and Mass Transfer* **34**, 1543–1557.

- Gartling, D. (1990), 'A Test Problem for Outflow Boundary Conditions-Flow Over A Backward-Facing Step', *International Journal for Numerical Methods in Fluids* **11**, 953–967.
- Gatski, T. B., Grosch, C. E. & Rose, M. E. (1982), 'A Numerical Study of the Two-Dimensional Navier-Stokes Equations in Vorticity-Velocity Variables', *Journal of Computational Physics* **6**, 1–22.
- Gatski, T. B., Grosch, C. E. & Rose, M. E. (1989), 'The Numerical Solution of the Navier-Stokes Equations for 3-Dimensional, Unsteady, Incompressible Flow by Compact Schemes', *Journal of Computational Physics* **82**, 298–329.
- Ghia, U., Ghia, K. N. & Shin, C. T. (1982), 'High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and A Multigrid Method', *Journal of Computational Physics* **48**, 387–411.
- Goldstein, R. J. & Kreid, D. K. (1967), 'Measurement of Laminar Flow Development in A Square Duct Using A Laser-Doppler Flowmeter', *Journal of Applied Mechanics* **34**, 813–818.
- Golub, G. & Van Loan, C. (1989), *Matrix Computations*, The Johns Hopkins University Press, Baltimore.
- Gresho, P. M. (1991 *a*), A Summary Report on the 14 July 91 Minisymposium on Outflow Boundary Conditions for Incompressible Flow, Technical report, Lawrence Livermore National Laboratory. UCRL-JC-108157.

- Gresho, P. M. (1991*b*), ‘Incompressible Fluid Mechanics : Some Fundamental Issues’, *Annual Review of Fluid Mechanics* **23**, 413–453.
- Gresho, P. M. (1991*c*), ‘Some Current CFD Issues Relevant to the Incompressible Navier-Stokes Equations’, *Computer Methods in Applied Mechanics and Engineering* **87**, 201–252.
- Gresho, P. M., Chan, S. T., Lee, R. L. & Upson, C. D. (1984), ‘A Modified Finite Element Method for Solving the Time-Dependent, Incompressible Navier-Stokes Equations. Part 1: Theory’, *International Journal for Numerical Methods in Fluids* **4**, 557–598.
- Gresho, P. M. & Sani, R. L. (1987), ‘On Pressure Boundary Conditions for the Incompressible Navier-Stokes Equations’, *International Journal for Numerical Methods in Fluids* **7**, 1111–1145.
- Guevremont, G., Habashi, W. G., , Kotiuga, P. L. & Hafez, M. M. (1993), ‘Finite Element Solution of the 3D Compressible Navier-Stokes Equations by a Velocity-Vorticity Method’, *Journal of Computational Physics* **107**, 176–187.
- Guevremont, G., Habashi, W. G. & Hafez, M. M. (1990), ‘Finite Element Solution of the Navier-Stokes Equations by a Velocity-Vorticity Method’, *International Journal for Numerical Methods in Fluids* **11**, 661–675.
- Guj, G. & Stella, F. (1988), ‘Numerical Solutions of High-Re Recirculating Flows in Vorticity-Velocity Form’, *International Journal for Numerical Methods in Fluids* **8**, 405–416.

- Guj, G. & Stella, F. (1993), 'A Vorticity-Velocity Method for the Numerical Solution of 3D Incompressible Flows', *Journal of Computational Physics* **106**, 286–298.
- Gunzburger, M. D. & Peterson, J. S. (1988), 'On Finite Element Approximations of the Streamfunction-Vorticity and Velocity-Vorticity Equations', *International Journal for Numerical Methods in Fluids* **8**, 1229–1240.
- Han, L. S. (1960), 'Hydrodynamics Entrance Lengths for Incompressible Laminar Flow in Rectangular Ducts', *Journal of Applied Mechanics* **27**, 403–409.
- Henkes, R. A. W. M. & Hoogendoorn, C. J. (1993), 'Scaling of the Laminar Natural-Convection Flow in a Heated Square Cavity', *International Journal of Heat and Mass Transfer* **36**, 2913–2925.
- Hughes, T. J. R., Liu, W. K. & Brooks, A. (1979), 'Finite Element Analysis of Incompressible Viscous Flows by Penalty Method', *Journal of Computational Physics* **30**, 1–60.
- Jiang, B. N., Lin, T. L. & Povinelli, L. A. (1994), 'Large-Scale Computation of Incompressible Viscous Flow by Least-Squares Finite Element Method', *Computer Methods in Applied Mechanics and Engineering* **114**, 213–231.
- Ku, H. C., Hirsh, R. S. & Taylor, T. D. (1987), 'A Pseudospectral Method for Solution of the Three-Dimensional Convection Diffusion Difference Schemes', *Journal of Computational Physics* **70**, 439–462.

- Ladyzhenskaya, O. (1969), *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach, New York.
- Mallinson, G. D. & de Vahl Davis, G. (1977), 'Three-Dimensional Natural Convection in A Box: A Numerical Study', *Journal of Fluid Mechanics* **83**, 1–31.
- Napolitano, M. & Pascazio, G. (1991), 'A Numerical Method for the Vorticity-Velocity Navier-Stokes Equations in Two and Three Dimensions', *Computer & Fluids* **19**, 489–495.
- Orlandi, P. (1987), 'Vorticity-Velocity Formulation for High Re Flows', *Computers & Fluids* **15**, 137–149.
- Patankar, S. V. (1980), *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Company, New York.
- Quartapelle, L. (1993), *Numerical Solution of the Incompressible Navier-Stokes Equations*, Birkhauser Verlag, Basel.
- Reddy, M. P., Reddy, J. N. & Akay, H. U. (1992), 'Penalty Finite Element Analysis of Incompressible Flows Using Element by Element Solution Algorithm', *Computer Methods in Applied Mechanics and Engineering* **100**, 169–205.
- Saad, Y. (1989), 'Krylov Subspace Methods on Supercomputers', *SIAM J. Sci. Statist Comput.* **10**, 1200–1232.
- Sonneveld, P. (1989), 'CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems', *SIAM J. Sci. Statist Comput.* **10**, 36–52.

- Temam, R. (1968), 'A Method of Approximation of the Solution for Navier-Stokes Equations', *Bulletin of Soc. Math. France* **96**, 115–152.
- Van der Vorst, H. (1992), 'Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear systems', *SIAM J. Sci. Statist Comput.* **13**, 631–644.
- White, F. (1974), *Viscous Fluid Flow*, McGraw-Hill, New York.
- Williams, P. T. (1993), CCM Continuity Constraint Method: A Finite-Element Computational Fluid Dynamics Algorithm for Incompressible Navier-Stokes Fluid Flows, Technical report, Oak Ridge National Laboratory ORNL/TM-12389.
- Yang, H. & Camarero, R. (1991), 'Internal Three-Dimensional Viscous Flow Solution Using the Vorticity-Potential Method', *International Journal for Numerical Methods in Fluids* **12**, 1–15.

VITA

Kwai Lam Wong was born in Kwantung, China. He and his mother moved to Hong Kong when Kwai was four years old. He was enrolled at the University of Tennessee, Knoxville in Fall, 1978, where he graduated with a Bachelor of Science in Aerospace Engineering in 1982. Continuing his graduate studies at the UTK, he received a Master degree in Mathematics in 1986. Upon his graduation, he ventured into the restaurant entrepreneurship. He was a partner and manager of the Dynasty chain restaurant until 1993. In 1988, he was re-enrolled at the doctorate program in the Department of Engineering Science and Mechanics. He is married to the former Noella Lee and they have a cheerful four years old daughter named Fionnie.