Investigation and Improvement of Some Recent FVM and CVFEM Practices for Two-Dimensional, Incompressible, Viscous Fluid Flow

by

Mahmoud Afshar

Thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfilment of the requirements for the degree of

Master of Engineering

Department of Mechanical Engineering

McGill University

Montréal, Canada.

SHORT TITLE FOR M.Eng. THESIS

by

Mahmoud AFSHAR
Student No. 9001706

SHORT TITLE

Investigation and Improvement of FVMs and CVFEMs for Fluid Flow

IN THE NAME OF GOD

THE BENEFICENT, THE MERCIFUL

To the memory of my mother

ABSTRACT

Over the last decade, there has been considerable interest and success in the formulation of co-located equal-order Finite Volume Methods (FVMs) and Control-Volume Finite Element Methods (CVFEMs) for the prediction of fluid flow and heat transfer. This thesis is concerned with an evaluation and enhancement of some aspects of recent co-located equal-order FVMs and CVFEMs. In particular, the goals of this thesis are the following: (1) formulation and computer implementation of a co-located equal-order FVM that facilitates the evaluation and enhancement tasks; (2) evaluation and enhancement of iterative sequential and coupled-equation solvers; and (3) comparative evaluation of a recently proposed mass-weighted skew upwind scheme (MAW) against five well-established schemes.

The proposed equal-order co-located FVM is formulated for the prediction of steady, two-dimensional, incompressible, viscous fluid flow in planar rectangular domains. This FVM deals directly with the velocity components and pressure, or primitive variables. The rectangular domains are discretized using structured line-by-line rectilinear grids, and rectangular control volumes are constructed around each grid point. All dependent variables are co-located or stored at the same grid points, and interpolated on the same rectangular elements in an equal-order formulation. A computer program incorporating the proposed FVM has been developed and tested successfully.

On the basis of an evaluation of some available iterative solvers, two improved algorithms are proposed in this work: (1) Enhanced Sequential Solution Algorithm (ESSA); and (2) Sequential Variable Adjustment (SEVA) algorithm. Test results obtained with these algorithms are very encouraging, particulary with ESSA. The comparative evaluation of the MAW scheme shows that its performance is comparable to the skew upwind difference scheme (SUDS) of Raithby [89].

SOMMAIRE

Les Méthodes de Volumes Finis (MVFs) à collocation d'ordre égal et les Méthode aux Eléments Finis/Volumes de Contrôle (MEFVC) ont été appliquées avec succès aux écoulements de fluide avec transfert de chaleur et ont présenté un intérêt considérable durant la dernière décade. Cette thèse porte plus particulièrement sur l'évaluation et l'amélioration de certains récents aspects concernant les MVFs à collocation d'ordre égal et les MEFCVs. Les objectits de cette thèse se résument ainsi: (1) formulation et implantation d'une MFV à collocation d'ordre égal facilitant les tâches d'évalution et d'amélioration; (2) évaluation et amélioration de la procédure de solution séquenttielle des équations couplées; et (3) comparison entre un récent schéma amont orienté aux flux massiques pondérés (SAOFMP) et d'autres schémas reconnus.

La MVF à collcation d'ordre égal proposée est formulée pour la prédiction d'écoulements permanents, bi-dimensionels, incompressibles et visqueux de fluid à l'intérieur d'enceintes planes et rectangulaires cette MVF utilise les composantes de la vittesse et la pression-les variable primitives- dans sa formulation. L'enceinte rectangulaire est discrétisée en utilisant un seul maillage selon une structure rectiligne, et un volume de contrôle rectangulaire est construit outour de chaque noeud du domaine. Toutes les variables dépendantes sont évaluées aux noeuds du maillage et sont interpolées sur les aux mêmes éléments puisqu'il s'agit d'un schéma d'ordre égal. Un programme informatique incorporant la MVF proposée a été développé et vérifié.

Suivant l'évaluation de différentes procédures de solution existantes, deux algorithmes sont proposés dans ce travail: (1) Procédure de solution Séquentielle et Améliorée (PSSA); et (2) Procédure de Solution Séquentielle à Ajustment Variable (PSSAV). Les résultants obtenus avec ces procédures sont très encourageants, tout particulièrement avec PSSA. La comparaison des différents schémas démontre que le SAOFMD performe comparablement au schéma SUDS proposé par Raithby [89].

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my supervisor, Dr. B.R.Baliga, for his guidance during the course of this work. His valuable suggestions, friendly discussions and the time spent in reviewing the manuscript are well acknowledged.

This work could not have been attempted without the patience and moral support of my family, in particular my wife. Their sacrifices were great: my gratitude is profound.

The financial support provided by the National Iranian Oil Company through study leave program is highly appreciated.

Finally, the computer facilities which were made available through financial support provided by the Natural Sciences and Engineering Research Council of Canada, those which were provided by the CEntre de Recherche en Calcul Applique' (CERCA), and free computing time from the McGill Computing Centre are greatly appreciated.

Contents

List of Figures

List of Tables

NOMENCLATURE

Symbol	Description
a_P, a_{nb}, b	coefficients in the discretized convection-diffusion equation
a_P^p, a_{nb}^p, b^p	coefficients in the discretized pressure equation
a_P^u, a_{nb}^u, b^u	coefficients in the discretized x-momentum equation
a_P^v, a_{nb}^v, b^v	coefficients in the discretized y-momentum equation
A	coefficient matrix of the vector of the 4 integration point values $[\phi_k]$ for the MAW scheme
$A_{c.v}$	boundary surface area associated with a boundary node
a, b, c, d	Integration points
B	Coefficient matrix of the vector of the 4 nodal values $[\phi_{l,m}]$ for the MAW scheme
c_p	specific heat at constant pressure
CK_j	value of the CK expression at integration point j for SUDS and LSD schemes
CKP_j	value of the CKP expression at integration point j for LSD scheme
d^u, d^v	pressure coefficients derived from x- and y-momentum equations
$E\%$	percentage error
f	constant used in the MAW scheme

$\mathcal{F}$, $\mathcal{H}$	radiation heat transfer coefficients
h	convection heat transfer coefficient
H	depth of channel
$\vec{g}$	gravity vector
i, j	refers to node or element (i,j)
k	thermal conductivity
L	length
$\dot{m}$	a mass flow rate
$\vec{n}$	unit vector normal to control volume surface
N	number of iterations
N_M	number of iterations for the MAW scheme
N	a constant in the interpolation function for ϕ^c
Nu,Nu_{av}	local and average Nusselt numbers, respectively
p	Pressure
Pr	Prandtl number
q	heat flux
Ra	Rayleigh number
Re	Reynolds number
Re_w, Re_{av}	Reynolds number based on u_w and u_{av}, respectively
S, S^p, S^u, S^v	volumetric source terms in convection-diffusion, continuity, and momentum equations, respectively
S_c, S_p	coefficients in the linearized expression for S
t	time
t_M	time for the MAW scheme
T	temperature

T_h, T_c	temperatures at hot and cold walls, respectively
T_{ref}, T_{inf}	reference and ambient temperatures, respectively
ΔT	temperature difference
u	velocity component in the x-direction
u_{av}	average u-velocity at inlet: flow over backward-facing step
u_w,	velocity of lid: lid-driven flow in a square cavity
$\hat{u}$	pseudo velocity defined using discretized x-momentum equation
ΔV	volume
V_n	velocity components normal to boundary control volume surface
v	velocity component in the y-direction
$\hat{v}$	pseudo velocity defined using discretized y-momentum equation
x, y	corresponding to x and y directions in a Cartesian coordinate system
δx	length of the side of element in x-direction
Δx	length of the side of control volume in x-direction
δy	length of the side of element in y-direction
Δy	length of the side of control volume in y-direction

Greek Symbols

α	thermal diffusivity
α_u, α_v, α_T	under-relaxation parameter for u, v, and T discretization equations
β	thermal volumetric expansion coefficient
Γ	diffusion coefficient
Φ	transported scalar

ϕ	approximate value of Φ
ρ	mass density
ρ_c	mass density at cold temperature
μ	dynamic viscosity
ν	kinematic viscosity

Subscripts

1, 2, 3, 4	indices used for mass flow rate through faces of control volumes associated with each element
a, b, c, d	refers to integration points
av	refers to average value
c	pertains to cold temperature
cv	control volume
e	refers to a given element
h	pertains to hot temperature
i, j	indices that refer to nodes and elements
∞	pertains to ambient condition
k	index used in the MAW scheme for integration-point values
l, m	indices used in the MAW scheme for nodal values
M	index used for the MAW scheme
n	normal to the surface of boundary control-volumes
nb	neighbouring points in numerical molecule
ref	index used for reference quantities
T, u, v	indices used for under-relaxation parameters, pertaining to u, v, and T discretization equations
w	corresponding to lid: lid-driven flow in a square cavity

x, y	corresponding to x, y coordinate directions, respectively

Superscripts

$*$	nondimensional parameters
p	pertains to pressure
u, v	corresponding to u and v velocity components, respectively

Chapter 1

INTRODUCTION

1.1 AIMS OF THE THESIS

This thesis has two main goals. The first goal is to formulate and implement a finite volume method (FVM) for the prediction of steady, two-dimensional, viscous, incompressible fluid flow and heat transfer in planar rectangular domains. The proposed method deals directly with the velocity components and pressure, or primitive dependent variables. The rectangular domains are discretized using structured (line-by-line) rectilinear grids. Each grid point, or node, is associated with non-overlapping contiguous control volumes that collectively fill up the domain completely and exactly. All dependent variables are co-located, or stored at the same grid points, and interpolated on the same rectangular elements in an equal-order formulation. This co-located equal-order FVM is based on concepts borrowed from control-volume-based finite element methods (CVFEMs) proposed by Prakash and Patankar [86], Saabas [100], and Schneider and Raw [103].

The second, and primary, goal of this thesis is to use the aforementioned FVM to (i) evaluate and enhance an iterative sequential procedure for the solution of discretized momentum and continuity equations, proposed recently by Saabas [100];

and (ii) undertake a comparative evaluation of six schemes for the interpolation of scalar dependent variables in the convective transport terms in the governing equations. Specifically, a Mass-Weighted upwind scheme (MAW) adapted from the work of Saabas [100], the Upwind and Central Difference Schemes (UDS and CDS) as discussed by Patankar [77], the Skew Upwind Difference Scheme (SUDS) of Raithby [89], the Linear-Skew Difference (LSD) of Huget [53], and the Quadratic Upwind Interpolation for Convective Kinematics (QUICK) of Leonard [63] are considered in the comparative evaluation.

1.2 Background and Motivation

Computational Fluid Dynamics (CFD) is a subject that deals with the formulation and application of mathematical models and numerical solution methods for the computer simulation of natural and industrial fluid flow and heat transfer phenomena. CFD uses and extends ideas and concepts borrowed from applied mathematics, computer science, and several engineering science subjects such as thermodynamics, fluid dynamics, and heat transfer. Research in this interdisciplinary subject can be both challenging and satisfying, and it can result in economic, social, and health-related benefits. This is the general inspiration for the work undertaken in this thesis. Specific technical issues that provided the motivation for this research are discussed in the remainder of this section.

Since the mid-sixties, the FVMs of Harlow and Welch [44], Patankar and Spalding [76], and Raithby [89] have been successfully applied to complex fluid flow and heat transfer phenomena, but these methods are limited to regular-shaped domains. This is because they are based on staggered orthogonal grids for the velocity components and pressure, and they employ locally one-dimensional functions to interpolate the nodal values of the dependent variables. These concepts are not directly applicable

to irregular non-orthogonal grids.

The desire and the need to extend finite volume, or control volume-based, formulations to irregular domains provided the motivation for the early Control-Volume-based Finite Element Methods (CVFEMs) of Baliga [6], Baliga and Patankar [7, 8], and Prakash and Patankar [86]. These CVFEMs were constructed by combining and extending ideas contained in the aforementioned FVMs [76, 89], the work of Winslow [127], and the Finite Element Methods (FEMs) of Zienkiewicz [129], Oden et al. [75], and Taylor and Hood [116]. Today, after over fifteen years of work by several research groups around the world, CVFEMs are being applied successfully to complex two- and three-dimensional fluid flow and heat transfer phenomena [92, 100]. However, some minor difficulties continue to afflict these methods.

Most of the CVFEMs proposed in the eighties suffer from intrinsic limitations that severely restrict the scope of their applicability to practical problems [100, 10]. CVFEMs based on flow-oriented upwind schemes similar to those of Baliga and Patankar [7], LeDain-Muir and Baliga [62], and Prakash [85] have been successful in overcoming false diffusion difficulties that afflict upwind schemes that are not flow oriented [77, 89]. However, they can encounter difficulties caused by negative coefficients in the discretization equations in problems that involve high element-based Peclet numbers [100, 10].

The unequal-order two-dimensional CVFEM of Baliga and Patankar [8] avoids checkerboard-type pressure distributions, but it suffers a loss of accuracy in problems with high Reynolds number and steep pressure gradients. Furthermore, it is unwieldy to extend this method to three-dimensions [10]. In the co-located equal-order, two-dimensional, CVFEMs of Prakash [87] and Hookey and Baliga [49] the velocity components are interpolated over three-node triangular elements by functions that explicitly account for the influence of the local pressure gradient. These CVFEMs

successfully overcome the problem of spurious checkerboard-type pressure distributions that afflicted earlier co-located equal-order FEMs [116]. However, because of the manner in which the pressure gradient is explicitly included in the velocity interpolation functions, such formulations require overspecification of boundary conditions and could encounter convergence difficulties in problems with inflow and outflow boundaries [100, 10]. Schneider and Raw [103] have proposed a co-located equal-order CVFEM based on quadrilateral elements and a mass-weighted upwind scheme that ensures positive coefficients in the discretization equations, and avoids spurious spatial oscillations in the computed pressure field. However, in problems involving inflow and outflow boundaries, it appears as if their method would encounter the same difficulties as those encountered by the CVFEMs of Prakash [87] and Hookey and Baliga [49].

The recent doctoral work of Saabas [100] was undertaken with the aim of overcoming some of the difficulties discussed in the previous paragraph. The resulting equal-order co-located CVFEM uses three-node triangular and four-node tetrahedral elements to discretize calculation domains in two- and three-dimensional problems, respectively. These elements are further discretized so as to create polygonal and polyhedral control volumes around the nodes in triangular- and tetrahedral-element meshes, respectively. The procedure for constructing polygonal control volumes in two-dimensional problems is borrowed from the work of McCormick [70], and an extension of these ideas is used to construct the polyhedral control volumes in three-dimensional problems [10, 100]. The dependent variables are interpolated linearly in each element in the approximation of diffusion terms. In the approximation of convective terms, two flow-oriented and one mass-weighted upwind schemes were investigated. The mass-weighted upwind scheme is formulated to ensure that the algebraic approximations to the convection terms contribute positively to the coefficients in the

discretization equations. In each element, the velocity components in the mass flow terms are interpolated by special functions that directly account for the influence of the elemental pressure gradient and prevent the occurrence of spurious pressure oscillations. The resulting discretization equations are solved using an iterative sequential solution algorithm.

The CVFEM of Saabas [100] has been successfully implemented to steady, two- and three-dimensional, laminar and turbulent, incompressible, viscous fluid flows in irregular-shaped geometries, with and without inflow and outflow boundaries. However, this iterative sequential algorithm for the solution of the discretization equations converges rather slowly, and it requires the simultaneous storage of the coefficients in the discretized momentum and pressure equations. Furthermore, in problems with inflow and outflow boundaries, the special functions that are used to interpolate velocity components in the mass flux terms, in conjunction with the proposed outflow treatment, can lead to minor inconsistencies in the overall mass balance and further slow down the convergence of the solution procedure.

It should also be noted that the MAW scheme proposed by Saabas is useful in avoiding negative coefficients in the discretization equations. However, in problems where flow-oriented upwind functions (FLO) worked, the MAW scheme was found to be less accurate than the FLO scheme [100]. A detailed comparative evaluation of the MAW scheme against other well-known schemes such as UDS [77], SUDS [89], and QUICK [63] was not done by Saabas [100].

The objectives of this thesis were finalized after the completion of the work of Saabas [100]. There was an urgent need to improve the rate of convergence and to reduce the storage requirements of the iterative sequential solution algorithm proposed by Saabas. It was also realized that the aforementioned difficulty experienced by the outflow treatment would not occur in a co-located equal-order FVM formulation

based on rectangular elements. Such an FVM would, therefore, allow the attention to be focused on the improvement of the iterative sequential solution procedure. Furthermore, this FVM would also allow a comparative evaluation of the MAW scheme against other well-established schemes [53, 63, 77, 89] in the context of a co-located, equal-order, primitive-variables formulation: there are no works of this nature in the published literature. It is with these needs and research opportunities in mind that the aims and scope of this thesis, as discussed in section 1.1, were determined.

1.3 Synopsis of Some Finite Methods for Fluid Flow and Heat Transfer

As has been discussed by Baliga and Saabas [10], finite volume methods (FVMs), finite element methods (FEMs), and control-volume based finite element methods (CVFEMs) are all particular cases of the methods of weighted residuals (MWRs). This unifying view was first proposed by Finlayson and Scriven [28]. The formulation of FVMs and CVFEMs for fluid flow typically involves five basic steps:

1. discretization of the calculation domain using elements (in CVFEMs) or structured orthogonal or non-orthogonal grids (in FVMs);

2. further discretization of the domain so as to associate each node (in CVFEMs) or grid point (in FVMs) with control volumes;

3. prescription of element-based multidimensional functions (in CVFEMs) or locally one-dimensional piecewise functions (in FVMs) to interpolate dependent variables and thermophysical properties of the fluid;

4. use of the subdomain, or control-volume-based, MWR to derive algebraic approximations to the governing equations; and

5. prescription of a procedure to solve these algebraic (discretization) equations.

In FEMs for fluid flow, step (ii), discretization into control volumes, is not done, and the Galerkin or Petrov-Galerkin MWR [18, 129] is used to derive the discretization equations. Otherwise, the formulation of FEMs is similar to that of CVFEMs. It should also be mentioned that in both CVFEMs and FEMs, the discretization equations are assembled using an element-by-element procedure. As was mentioned earlier in section 1.2, and as is perhaps clear from this discussion, CVFEMs are constructed by a combination of ideas native to FVMs and FEMs. Thus, CVFEMs could be considered as subset of FVMs or FEMs. Indeed, the method put forward in this thesis uses an element-by-element approach to the derivation and assembly of the discretization equations, but it is limited to rectangular elements, and it employs some locally one-dimensional interpolation functions. It was thus decided to label it as a FVM.

A brief review of CVFEMs for fluid flow was given in section 1.2. Detailed reviews of CVFEMs are available in the works of Baliga and Patankar [9] and Baliga and Saabas [10], so these methods will not be discussed further in this section. Review of FEMs for fluid flow are available in books by Chung [21] and Baker [5]. Furthermore, a detailed discussion of FEMs is not directly relevant either to the motivation or the formulation of the FVM proposed in this thesis. For these reasons, FEMs are not reviewed further in this section.

This review, therefore, deals primarily with FVMs for fluid flow and heat transfer. It is divided into three sections: in the first section, FVMs for convection-diffusion problems are discussed; in the second section, methods for the storage and calculation of pressure in incompressible fluid flows are reviewed; and in the last section, methods for solution of the discretized momentum and continuity equations are discussed.

In this thesis, following the accepted definition in the published literature [77], the term convection-diffusion problems is used to denote situations in which transport of a scalar dependent variable is to be calculated in the presence of a known, or prescribed, fluid flow: transport due to the overall or gross fluid flow is termed convection; [1] and transport due to molecular interactions is termed diffusion. Solution procedures for convection-diffusion problems are a prerequisite to the formulation of methods for the numerical simulation of fluid flow [77]. As the momentum of the fluid is transported by convection and diffusion, the corresponding governing equations can be solved by procedures designed for the solution of convection-diffusion problems. However, as was mentioned earlier, special procedures are required for the calculation of pressure in incompressible fluid flows. Furthermore, special methods are also needed to solve the coupled, nonlinear, sets of discretized momentum and continuity equations [77].

1.3.1 FVMs for Convection-Diffusion Problems

The Central Difference Scheme (CDS) appears to be the first method that was used for the algebraic approximation of convection and diffusion terms in the governing partial differential equations [77, 97]. It can be shown that the CDS is essentially equivalent to the use of piecewise-linear interpolation of the dependent variable(s) between grid points. This scheme works well in the numerical simulation of conduction-type problems, because the piecewise-linear interpolation function is appropriate for the modelling of the elliptic nature of the diffusion process. However, the CDS is inappropriate for modelling of the parabolic or one-way character of convective transport [77]. Indeed, it can be shown that when the value of the local Peclet number, based on the average velocity and the distance between adjacent grid points, exceeds two, the CDS can generate negative coefficients in the algebraic discretization equations

[1] This phenomena is also referred to as advection in the literature [55].

[77, 112]. Negative coefficients in the discretization equations can lead to unphysical oscillations, or "wiggles", in the solutions [77]. They can also slow convergence or cause divergence of iterative methods for the solution of the discretization equations [77]. One way to overcome this difficulty is to use grids that are fine enough to ensure that the value of the local Peclet number does not exceed two. However, this approach is computationally expensive and usually impractical in the solution of engineering problems.

An early remedy for the above-mentioned difficulty with the CDS was the Upwind Difference Scheme (UDS), developed by Courant et al. [23], Gentry et al. [38], Barakat and Clark [11], and Runchal and Wolfshtien [98]. In UDS, a piecewise-linear function is used to approximate the diffusion terms. However, a locally one-dimensional upwind treatment is used to approximate the convection terms: in an orthogonal finite-volume grid, the value of the convected scalar dependent variable at the intersection of a grid line and a control surface is assumed to be equal to its value at the adjacent node on the upwind side of the grid line. The UDS ensures that the coefficients in the discretized equations are all positive, over the whole range of Peclet numbers. However, this scheme is not as accurate as the CDS at low Peclet numbers, and it overestimates diffusion at high Peclet Numbers [77].

In an effort to improve the UDS, Spalding [112] derived the exact solution to the equation that governs steady, one-dimensional convection-diffusion transport in the absence of source terms, and with constant properties of the fluid. Using the analytical solution, he derived the Exponential Difference Scheme (EDS) [112]. This scheme works well for the whole range of grid Peclet (Pe_Δ) numbers, but it is computationally more expensive than the CDS and UDS because of the use of exponential functions. To overcome this objection, Spalding approximated the exponential scheme with a three-part function that is equivalent to the CDS for $|Pe_\Delta| \leq 2$, and purely upwind

advection (with no diffusion contribution) for $|Pe_\Delta| > 2$. Spalding called this approximation the Hybrid Difference Scheme (HDS) [112]. A better approximation to the EDS than the HDS is the Power-law Difference Scheme (PDS) of Patankar [77]. The PDS produces more accurate solutions than HDS, but requires some what more execution time.

The UDS, EDS, HDS, and PDS are all based on the approximation of convection transport using interpolation functions that are locally one-dimensional along the grid lines. Thus all these schemes suffer from numerical errors, or *false diffusion*, when the flow is at an angle to the grid lines and there are significant gradients of the dependent variables in the cross-flow direction [90]. Similar difficulties can also be encountered by these schemes in unsteady problems and problems with large source terms [32, 90, 121].

In a critical evaluation of the UDS, Raithby [90] showed the above-mentioned errors clearly. He also mentioned that it is the *size of the error* which is important, rather than the *order of the truncation error* in a Taylor series expansion of convective flux. To overcome false or numerical diffusion, Raithby proposed the skew upwind difference scheme (SUDS) [89] in which he considered the skewness of flow to the grid lines in addition to upwind nature of convection. This scheme, however, is prone to the difficulty of negative coefficients [45, 52, 63, 80, 100, 103] in the discretization equations. Such negative coefficients imply that an increase in the value of the scalar dependent variable at a node outside a control volume could result in a net outflow of that variable from the control volume: This is not physically correct, since the scalar dependent variable has to first flow into the control volume before it leaves it. In this paper [89], Raithby also introduced the skew upstream weighted difference scheme (SUWDS), in which he considered the influence of diffusion in the mean and cross flow directions in the interpolation of the convected scalar. This scheme decreases

false diffusion significantly in comparison to UDS, but negative coefficients can still occur in the discretization equations, and, as Raithby has mentioned, there is some doubt about the practical advantages of this scheme over the SUDS scheme because of the additional computational time for the evaluation of exponential terms in SUWDS. It should also be noted that both SUDS and SUWDS could suffer a loss of accuracy in unsteady problems and in problems with significant source terms. In an effort to overcome this difficulty Lillington [67] in an enhancement to SUDS proposed a vector upstream difference scheme (VUDS). However, this enhancement appears to be some what ad-hoc [36]. Other attempts have been made to improve SUDS [36], with limited success. One of the results of such attempts is the Linear Skew Difference (LSD) of Huget [36]. Details of LSD are present in chapter 2.

A Quadratic Upstream Interpolation for Convective Kinematics (QUICK) scheme has been introduced by Leonard [63]. This scheme is equivalent to a CDS scheme that is corrected by a term proportional to an upstream-weighted term. In QUICK, the overall truncation error is of third order in the spatial grid size. However, as noted by Raithby [90], it is the size of the error which is important, and only using higher-order interpolation functions does not guarantee better accuracy [2]. The QUICK scheme gives very accurate results when the Peclet number is low [74, 52, 79, 80], but when there is strong convective transport, negative coefficients can arise in the discretization equations and lead to unphysical oscillations in the solutions and instability in iterative solution methods [26, 43, 52, 74, 79, 80, 108, 109].

Pollard and Siu [83] have worked out extended versions of QUICK that have overcome some, but not all, of the aforementioned difficulties [79]. Wong and Raithby [128] have proposed a locally analytic differencing scheme (LOADS) based on a solution to an approximation of the convection-diffusion problem. It is an enhancement to EDS, in that LOADS accounts for the influence of the volumetric source term and

local deviations from one dimensionality. After evaluating the different approximations in the context of natural convection in a square cavity, they [128] improved the poorest approximations step-by-step and come up with LOADS scheme. The main drawbacks of this scheme are the existence of negative coefficients in the discretization equations and exponential terms in the discretized-equation coefficients. A flux-spline method has been proposed by Varejao [80-6]. This scheme uses a quadratic-spline technique to accurately calculate convection-diffusion fluxes at control-volume faces in the finite-volume technique, but it admits negative coefficients in the discretization equations.

To find a framework for studying numerical errors in discrete methods, Stubley et al. [114, 115] examined the CDS and UDS with reference to a problem with an exact solution. They stated that there are two types of errors in finite difference schemes: (i) profile error, which is a measure of the degree to which the corresponding interpolation function is fitted with the exact solution; and (ii) operator error, which is the error associated in approximating the differential operators in the convection-diffusion terms. Therefore, reducing the profile error would not necessarily ensure smaller solution errors, since the approximated operator may distribute the profile error in such a way that solution error becomes larger. Based on these studies, Stubley et al. [114] proposed two new schemes, linear influence scheme (LIS) and quadratic influence scheme (QIS). In these schemes, they used approximate analytical solutions of two-dimensional flows to *influence* the coefficients in the discretization equations. However, these schemes require repeated summations of infinite series, thus the implementation of these schemes is tedious and uneconomical. Furthermore, relatively crude interpolations of the dependent variables along the gridlines are used as inputs to LIS and QIS. This raises further doubts about their viability.

Gresho and Lee [41] states that it perhaps best not to suppress the oscillations

caused by negative coefficients, since they are an indicator of the accuracy of the schemes. But Patankar [78] states that distinguishing between the real and false oscillations would be difficult, especially in turbulent flows. In an effort to improve the SUDS scheme, Hassan et al. proposed a mass-flow-weighted two-dimensional skew upwind scheme [45]. They studied the reason for negative coefficients in the discretized equations associated with SUDS, and to prevent negative coefficients, proposed that the skewing be limited so as to ensure that the contribution of the value of the dependent variable at a node external to a control volume to the outflow of the variable is always less than its contribution to inflow into that control volume. This suggestion has a strong physical basis: in steady convection-diffusion problems without source terms, for a transported scalar to go out from a control volume, it first has to come into that control volume. This concept has been later used by Schneider and Raw [103] and Saabas [100] in the context of control-volume-based finite element methods (CVFEMs). Huget [53] tested several difference schemes, and proposed three new schemes: mass-weighted upstream scheme (MWUS), modified skew upstream scheme (MSUS), and a source correction scheme (SCS). Detailed description of these schemes are available in his doctoral thesis [53].

In a technical note, Galpin et al. [33] discuss the effect of grid curvature on upstream-weighted-advection approximations in a simple uniform flow through a half-cylindrical region. They showed that in the circumferential direction, approximation of the convected variable at control-volume faces using upstream functions along the grid lines can produce significantly different values than the true upstream values, because of the curvature of the grid lines. To remedy this, they introduced a vector upwind approximation (VUWS) which is an upstream-weighted approximation with a correction term to include the effects of the grid curvature. Also, Galpin et al. [36] have divided advection schemes into four general classes: (i) profile schemes, such

as CDS or QUICK; (ii) operator schemes, such as flux-spline schemes and LOADS; (iii) upstream schemes, such as UDS or EDS; and (iv) skew upstream schemes, such as SUDS and VUWS. With respect to the physics of fluid flows, skew upstream schemes are indicated as best choice by them. Based on a study of different errors and previous experiences, they have proposed the basic full-influenced scheme [36] and two variations of it, namely, linear profile skew (LPS) and mass-weighted skew (MWS). They approximate the advection term by a upstream difference along the local streamline, which results in a correction term to the upstream value: this takes into consideration the effect of source and diffusion terms. The local upstream value is then calculated either by LPS or MWS schemes. Since they have tried to address most of the errors associate with discrete solutions, their schemes appear to be quite promising [100].

The SHARP scheme is a result of attempts by Leonard [64, 65] to improve QUICK, after earlier modifications to QUICK such as QUICKE and QUICKER [83]. In SHARP he uses a piecewise model for interpolation of the convected scalar: this includes UDS, QUICK, exponential upwind, and ad-hoc straight-line parts. In the results to a two-dimensional problem involving pure-convective transport of an oblique-step distribution of the scalar, he showed that this scheme did not have the stability problems of QUICK, but the implementation of this scheme is both tedious and expensive because of the many parts associated in the calculation of coefficients and exponential terms.

Van Doormaal et al. [124] have proposed a scheme similar to that of Galpin et al. [36] in an explicit two-dimensional form, and named it physical advection correction (PAC) scheme. They used two subclasses of it, namely grid-upstream schemes and SUDS [89] schemes, to derive corrections to the interface values of the transported scalar. They recommend the use of either linear-profile or mass-weighted schemes

[103] to complete the SUDS subclass of PAC, and state the drawbacks of each of these methods.

Karki et al. [57] used the flux-spline method of Varejao [126] in a three-dimensional lid-driven flow in a square cavity. This scheme is based on the assumption that within a control volume, the total flux in a given direction varies linearly along the coordinate direction. They found that it gives more accurate results than PDS.

By studying three-, four-, and five-point schemes Braga [16] investigated the reasons for instabilities and convergence problems in several available methods for convection-diffusion problems. He derived a new scheme which shows better performance than the QUICK scheme, but still it has overshoot and undershoot problems caused by negative coefficients in the discretization equations. In an enhancement of the QUICK scheme: Tzanos [120] used a scheme like CDS, but with modifications that ensure that cell-face values of the convected scalar are bounded by their adjacent cell-central values, and prevent wiggles in the solution at high Peclet numbers. In test problems, this scheme showed better accuracy than UDS. Tzanos [120] also recommends the use of adaptive grid methods to obtain higher accuracy with a fixed number of grid points.

With the rapid growth of the number and sophistication of convection-diffusion schemes, it is not surprising that there have been many attempts to evaluate their capabilities. Runchal [99] did a comparative evaluation of the CDS, UDS and HDS schemes. He reported that among these schemes, CDS has the best accuracy as long as it converge $|Pe_\Delta| \leq 2$. HDS on the other hand, shows better convergence and accuracy properties than UDS. A comparison between the performance of HDS and QUICK schemes in two-dimensional problems with elliptic flows at high Reynolds numbers was done by Han et al. [43]. They used these schemes to simulate axisymmetric stagnation flow , square-cavity lid-driven flow, and flow downstream of

a sudden expansion in a pipe (in the latter two cases, both laminar and turbulent cases were studied using a staggered-grid finite volume method [43]). They used the SIMPLE procedure for solution of the discretization equations, and reported that in both laminar and turbulent flows, QUICK showed remarkably more accuracy than HDS, but unlike HDS, it was not unconditionally stable. Smith and Hutton [111] have reported the results of a comparison of thirty sets of solutions (using finite-difference, finite-element, and characteristics methods) submitted by nineteen groups. All solutions pertain to a standard problem, which involves flow in a duct with a 180 degree turn. On coarse grids, all methods showed solution oscillations or false diffusion. None of the methods emerged as the best, and based on their opinion, there was no perfect scheme, and compromise between diffusive and oscillatory errors was an "artistic necessity" [111].

Four different convection-diffusion schemes, PDS, SUDS, QUICK, and LOADS were evaluated by Huang et al. [52] in six tests, including square-cavity lid-driven flow, natural convection in a square cavity, irrotational flow in a corner, and impinging-jet flows. They used staggered-grid arrangement in a finite volume model, and reported that LOADS performed well for either linear or irrotational flows, but it failed to converge for nonlinear viscous flows. PDS or any other locally one-dimensional scheme is only appropriate if the flow direction is almost parallel to grid lines. SUDS produced convergent solutions in all their test problems, but it was prone to overshoot and undershoot problems, and its running time was about two-and-a-half times that of PDS. The QUICK scheme was also prone to overshoot and undershoot problems, but its accuracy was better and its running time was 1.65 times more than that of PDS. The QUICK scheme was the best among the four schemes, as reported by Huang et al. [52].

Shyy [108] reported a study of five schemes, namely, UDS, SUDS, QUICK, Second

Order CDS (SOCDS), Second Order UDS (SOUDS) in two one-dimensional tests and one two-dimensional convection-diffusion problem. He found that the higher-order schemes would not always give better results, and in the first test, a boundary-layer type problem, UDS and SOUDS were stable, but QUICK and SOCDS generated spurious oscillations at high Peclet numbers. In the second test, a flow with a source term, SOUDS and QUICK gave comparable results. In the last test, QUICK and SOCDS again showed oscillatory solutions, and UDS and SOUDS had high numerical diffusion. Overall, SOUDS was the most satisfactory scheme. The accuracy of SUDS was better than UDS, but both of them displayed large errors in modelling of source terms, and QUICK and SOCDS was prone to solution oscillations.

Three convection-diffusion schemes, namely, HDS, QUICK, and SOUDS, were used by Shyy et al. [109] to simulate a recirculating flow in nonorthogonal curvilinear coordinates. QUICK was found to be the most restrictive scheme with regard to stability and convergence of iterative solution methods. HDS on the other hand, was prone to excessive numerical diffusion. SOUDS was the best scheme in their test. The investigation of Demuren [26] regarding false diffusion in three-dimensional turbulent flows found that QUICK produced more accurate results than UDS, but it is accompanied by overshoots and undershoots.

Patel et al. [79] compared eight discretization schemes in the simulation of two-dimensional flows in a lid-driven square cavity flow, and sudden enlargement in a circular pipe. The eight schemes were: CDS, UDS, HDS, QUICK, QUICKE, QUICKER, PDS, and EDS. They used the staggered arrangement for grids, and the SIMPLE procedure for solving the sets of coupled discretization equations. They discuss the convergence properties and boundness of each scheme, and based on the results of their tests, report that CDS, QUICK and QUICKE were the most unstable schemes at high Reynolds number; the others were always stable. On the other hand, QUICK,

QUICKE and QUICKER were found to be more accurate than others when they produced converged solutions. In terms of computer time, they found QUICKER to be the most expensive scheme, with largest necessary number of iterations for convergence. Their opinion was that although QUICK and its variants might offer the best choice in terms of accuracy, because of their other limitations, they could not be used as a general scheme for a wide variety of practical problems.

In another study, Patel et al. [80] evaluated eleven discretization schemes including CDS. UDS, HDS, PDS, QUICK, QUICKE, QUICKER, SUDS, RDS, and UPSTREAM schemes. They used these schemes to predict elliptic flow and heat transfer in supersonic jets mixing in supersonic or subsonic streams. In these tests, they used staggered grids and the SIMPLE procedure. They found that the false diffusion, even with UDS, in turbulent flows was not significant with respect to the uncertainties in the turbulence models, and, in most cases, with respect to the turbulent eddy viscosity. Only five of the schemes, UDS, HDS, EDS, PDS and UPSTREAM produced converged solutions for this problem, and these five schemes gave almost the same results. Despite the best efforts of these authors, SUDS and QUICK schemes did not converge, and in their opinion, UDS is probably the best choice for this kind of problems.

In a review of recent developments in computational heat transfer, Patankar [78] did a survey of different schemes and stated that lower-order schemes such as UDS are stable and converge monotonically, but lead to false diffusion; and higher-order schemes such as QUICK eliminate (or reduce) false diffusion, but produce solution oscillations and often fail to converge. Mohamad et al. [74] compared CDS, HDS, PDS, and the QUICK scheme in a numerical simulation of natural convection of low-Prantdl-number fluids in a cavity. Their results showed rapid convergence of the temperature field, due to high thermal diffusivity, but a large number of iterations

was required for the solution of the momentum equations. They suggest that for such a problem, a careful choice of the discretization scheme and grids must be made in order to have stable and accurate solutions at a reasonable cost. They studied both steady and transient cases, and reported that QUICK produced better results in terms of accuracy, but it was prone to solution wiggles.

A review of many comparisons of different schemes is tabulated and discussed in a paper by Zurigat and Ghajar. [130]. They have also compared Weighted Upwind Difference Scheme (WUDS) and SOUDS on four tests in a staggered-grid two-dimensional framework. They report that SOUDS gives better results than WUDS, but it creates some over and undershoots. WUDS was found to generate more false diffusion than SOUDS.

Finally, Tsui [119] has tested eight discretization schemes, including CDS, UDS, QUICK, and other higher-order schemes. He derived a general scheme from which these schemes can be obtained as particular cases. His study included examination of the coefficients of the difference equations, Taylor-series analysis, upwind connection to numerical diffusion, single-cell analysis, and one- and two-dimensional model problems. His conclusions are similar to those of Stubley et al. [114]: increasing the order of the scheme does not guarantee better accuracy.

This literature review showed that there have been numerous investigations of many different schemes for convection-diffusion problems. However, most of the studies have been conducted with staggered-grid finite-volume methods for fluid flow. Thus there seems to be a need to examine these schemes in the context of co-located equal-order FVMs. This thesis aims to fulfil a part of this need.

1.3.2 Storage and Treatment of Pressure in the Computation of Incompressible Fluid Flows

In finite methods for fluid flow that deal directly with the velocity components and pressure (primitive variables), if these dependent variables are stored at the same nodes or grid points (co-located) and interpolated with similar interpolation functions (equal order), then physically unrealistic checkerboard-type pressure distributions could be admitted as solutions. This difficulty was first discussed by Harlow and Welch [44]. Additional discussions of this problem are available in the works of Carreto, Curr and Spalding [20], Patankar and Spalding [76], and Roache [97].

One way to overcome the aforementioned difficulty with checkerboard pressure is to eliminate it from the governing equations. In two-dimensional problems, pressure can be eliminated by first cross differentiating the momentum equation, and then subtracting one of these equations from the other, to obtain a vorticity-transport equation. This equation, when combined with the definition of a stream function, is the basis of the so-called vorticity-stream function methods. Such methods have been proposed and discussed by Fromm and Harlow [29], Barakat and Clark [11], Runchal and Wolfshtien [98], Gosman et al. [40], and Roache [97].

Vorticity-stream function methods have several attractive features [77]. The pressure is eliminated from the governing equations; only two differential equations, one for vorticity and one for stream function, have to be solved, as opposed to three (two momentum and one continuity equations) in primitive-variable formulations; methods for convection-diffusion problems can be used to solve the two governing equations, so no additional developments are required for the solution of fluid flow problems; and boundary conditions pertaining to irrotational flow can be easily prescribed by simply setting vorticity at that boundary to zero. There are, however, some major disadvantages associated with vorticity-stream function methods: boundary condi-

tions on vorticity at solid walls are difficult to specify, especially at corners; boundary conditions on stream functions have to be specified iteratively in problems with multiply-connected domains; given-pressure boundary conditions present other challenges and complexities; and the stream-function concept can not be extended to three-dimensional problems.

Another approach that eliminates pressure from the governing equations is based on the use of the vorticity vector and the velocity-potential vector as dependent variables. This approach was first proposed by Aziz and Hellums [4]. It is not restricted to two-dimensional problems, but it involves the solution of four differential equations is two-dimensional problems and six differential equations in three-dimensional problems. Thus this approach is computationally more complex and more expensive than the primitive-variables approach. Furthermore, vorticity vector velocity-potential vector formulations suffer from boundary-condition difficulties similar to those experienced by stream function-vorticity formulations. In addition, it is generally accepted, at least among many engineers, that vorticity-based formulations involve concepts that are harder to visualize and interpret than those in primitive variables formulations [77]. For these reasons, numerical methods that deal directly with primitive-variables are generally preferred over methods that are based on vorticity formulations in computer simulations of practical flow problems.

In numerical methods based on primitive-variables formulations, the difficulty with checkerboard-type pressure distributions can be overcome by employing staggered grids for the velocity components and pressure. Such an arrangement ensures that each velocity component is driven by differences in pressures at adjacent, not alternate, nodes; when these velocity components are required to satisfy continuity constraints, unphysically checkerboard-type pressure distributions can not be admitted as solutions [77]. The staggered-grid approach for the velocity components and

pressure was first introduced by Harlow and Welch [44] in the MAC method. It is also used in the SIVA method of Carreto et al. [20], and it is the basis of the SIMPLE method of Patankar and Spalding [76], as well as its many extensions [77, 122].

The staggered-grid approach has been used with considerable success over the last twenty years for the solution of complex fluid flow problems in regular two- and three-dimensional geometries, discretized by orthogonal grids. It has also been used successfully with curvilinear orthogonal grids [42] and curvilinear non-orthogonal grids [57, 110]. However, the staggered-grid approach is not well-suited to implementation on non-orthogonal grids, and it can fail completely when such grids becomes highly non-orthogonal or undergo bends that exceed 90° [110]. Furthermore, it can not be extended to irregular finite element methods. It should also be noted that even when numerical methods based on staggered grids work well, they require considerably more complicated book-keeping and coding in computer programs than that necessary with co-located formulations.

The aforementioned difficulties with the staggered-grid approach motivated researchers to develop co-located primitive-variables methods with special procedures to avoid checkerboard-type pressure distributions. Hsu [51], Demirdzic [25], and Peric [81] were the first to introduce such ideas in FVMs, and similar ideas were presented by Prakash [85] in the context of control-volume finite element methods (CVFEMs). Hsu [51] developed special interpolation functions for the mass fluxes leaving the faces of the control volume surrounding a grid point [1]. These expressions are obtained using the discretized momentum equations, and they involve pseudo-velocities and differences in pressure at adjacent nodes. These special mass-flux interpolation functions are used in integral mass conservation equations to obtain the discretization equations for pressure. Rhie and Chow [96] used co-located grids for the problem of turbulent flow past an airfoil in general curvilinear coordinates. They used a correc-

tion term to the mass fluxes across the control-volume faces to remedy the problem of checkerboard-type pressure distribution. Shih and Ren [107] have also derived similar primitive-variables formulation on co-located grids.

Schneider [104] derived a quasi-one-dimensional solution to a convection-diffusion problem to calculate the velocities at interfaces of control volumes. Then the pressure gradient term was extracted from the source term and approximated using the difference in adjacent values of nodal pressures. These interfaces velocities were then used in the integral continuity equation, and the set of equations for velocities and pressure was solved iteratively. This co-located method gave poor results for high-Reynolds-number flow in a square driven cavity.

Reggio and Camarero [93] have solved the time-dependent incompressible Navier-Stokes equations in an arbitrary-shaped domain, using curvilinear co-located grids. They used overlapping grids and an opposed difference scheme for pressure and momentum fluxes in the main-flow direction to avoid oscillatory results. Mass-flux gradients were obtained by upwind differencing and pressure gradients were calculated by downwind differencing. This idea has also been extended to turbulent flows by Reggio et al. [94].

A comparison between staggered and collocated grids in the context of finite volume methods has been done by Peric et al. [82]. They did this comparison with a FVM based on orthogonal grids. The convergence properties of these two grid arrangements were investigated for three test problems: driven-cavity flow, flow over a backward-facing step, and flow in a sudden expansion in a pipe. They reported that the collocated FVM had no disadvantage compared to the FVM based on staggered grids, and in some cases, it provided faster convergence. The accuracy of results was also of the same order for both these approaches.

The role and influence of under-relaxation in co-located finite volume methods

has been investigated by Majumdar [71]. He pointed out that the results obtained by several other researchers who used FVM co-located grids were not independent of under-relaxation parameters, and he proposed an explicit under-relaxation of interface velocities in momentum equations to overcome this difficulty. Thiart [117, 118] integrated the momentum equations on staggered grids only for calculating the interface velocities. He then substituted them in the continuity equation to derive pressure and pressure-correction equations. All other derivation were done with the same grid for pressure and velocity components.

Kobayashi and Pereira [59] used a non-staggered, nonorthogonal grid to investigate the influence of under-relaxation factors and extended the ideas of Majumdar [71] to the PWIM of Peric [81]. Their method ensures that the values do not depend on under-relaxation factors. Finally, Coelho and Pereira [22] used the co-located method of Rhie and Chow [96] to solve the turbulent flow over a hill with two- and three-dimensional non-orthogonal co-located grid systems.

An extensive review and discussions of co-located equal-order CVFEMs is available in the works of Saabas [100] and Baliga and Saabas [10].

1.3.3 Solution of the Discretized Equations

In this section, methods for solution of the discretized momentum and continuity equations are reviewed. In incompressible flow problems, there is no explicit equation for pressure: When the correct pressure distribution is substituted into the momentum equations, it produces a velocity field that satisfies the continuity equation [77]. This indirect specification of pressure presents a special challenge to numerical solution methods. One approach to this problem is to directly solve linearized, coupled, sets of discretized momentum and continuity equations; the nonlinearity is handled through iterations. Another approach is to devise an explicit equation for pressure

by substituting discretized momentum equations into the discretized continuity equations; the linearized, coupled, discretized equations for the velocity components and pressure can then be solved using direct methods or iterative methods; the overall nonlinearity is again handled through iterations.

A vast number of methods are available for solving the nonlinear discretization equations. All these methods must perform two functions: linearization ; and solution of sets of linearized, coupled, algebraic equations. The linearization is usually done using successive-substitution methods or Newton-Raphson methods. Examples of the successive-substitution approach can be found in the works of Patankar [77], and Schneider et al. [101] among others. Successive-substitution methods are easy to implement, but they are only linearly convergent, when they converge [68]. The Newton-Raphson technique has quadratic convergence: but it does not converge to a solution from any arbitrary starting point, or guessed solution field [68, 34].

Hybrid methods have been developed in an effort to combine the desirable features of successive-substitution and Newton-Raphson techniques. Examples of hybrid methods include multiple linearization techniques of Levenberg [66] and Marquardt [69], and partial-rank quasi-Newton techniques of Powell [84] and Blue [13]. A hybrid steepest-descent algorithm has been proposed by Macarthur [68].

The methods that are used to solve the linearized, coupled, sets of discretized equations can be grouped into three categories: direct methods; semi-direct methods; and iterative methods. Direct methods include such approaches as Gaussian elimination, error vector propagation [97], fast Fourier transforms [61], and sparse matrix methods [95]. Direct methods basically require very large amounts of computer storage, specially in three-dimensional problems involving many dependent variables. Thus they are rarely used in FVMs for fluid flow.

Iterative methods for the solution of discretized equations include sequential so-

lution techniques such as SIMPLE [77], SIMPLER [77], and SIMPLEC [122]. Detail evaluation and review of such methods are available in the works of Schneider et al. [101], and Van Doormaal and Raithby [122]. In the sequential solution algorithms, the decoupled sets discretization equations can be solved by direct methods, such as sparse matrix techniques, or iterative methods for linear algebraic equations, such as Gauss-Seidel, Successive over-relaxation, and line-by-line methods [77]. Another iterative procedure that is gaining in popularity is the conjugate gradient method [95]. In this approach, the Gram-Schmidt orthogonalization process is used to form a sequence of vectors, where one of the vectors represents the unknowns to be solved for, and full coupling between all equations is maintained during the iteration process.

The semi-direct methods encompass those methods that can not be classified as either direct or fully iterative methods, and they represent a compromise between direct and iterative methods. The basic idea in such methods is to divide the calculation domain into segments and to solve the fully-coupled equations in each segment by a direct method. Examples of such methods include capacity matrix methods [46], block implicit relaxations [27], the subdomain methods of Braaten [15], and Vanka [125], and coupled-equation solvers of Galpin [34] and Hookey [50].

Other methods pertinent to this review include the method of false transients [72], and explicit time-stepping schemes to solve transient formulations [44]. The strongly implicit procedures of Stone [113] and Schneider and Zedan [102], and the block correction methods of Forsythe and Wasow [30] and Ames [3] provide interesting and useful options for enhancing the rate of convergence of iterative methods. Another approach to increase the rate of convergence of iterative methods is to work not with a single grid, but with a sequence of grids of increasing fineness. This is the basic ideas behind multigrid methods, such as those proposed by Brandt [17].

Comprehensive reviews of these solution methods are available in the doctoral

dissertations of Braatan [15] and Maccrthur [68]. Details of iterative methods akin to SIMPLE [77] and CELS of Galpin [34] and Hookey [50] are presented in chapter 4 of this thesis.

1.4 Outline of the Thesis

In this chapter, some of the numerical methods for fluid flow and heat transfer problems that have appeared in the published literature during the previous three decades have been reviewed. The main goals of this thesis have also been presented.

Chapter 2 presents the a finite volume method formulated (FVM) for convection-diffusion problems. First, a domain discretization scheme is described. Then, appropriate interpolation functions are prescribed for all of the dependent variables. Also, in this part, a MAW and five other schemes are discussed. Algebraic approximations to the governing equations are then derived using an element-by-element procedure. Lastly, an iterative solution procedure for the resulting coupled, non-linear, algebraic equations is discussed.

Chapter 3 is devoted to the formulation of a co-located equal-order FVM. Chapter 4 presents discussions of the Saabas scheme [100], and some sequential solution algorithms and coupled-equation line solvers. Based on these discussions, two new algorithms, SEVA and ESSA, are proposed and discussed in the remaining part of chapter 4.

Chapter 5 presents the results generated for three steady, two dimensional, fluid flow problems. These results are used to compare the convergence behaviour of the Saabas, SEVA, and ESSA schemes. Chapter 6 presents the results obtained in a comparative evaluation of the MAW scheme against five other well-established schemes.

The contributions of this thesis are discussed in chapter 7. Ideas for extensions and improvement of this work are also presented in this, the concluding chapter.

Chapter 2

CONVECTION-DIFFUSION PROBLEMS

2.1 Governing Equations

In convection-diffusion problems, the velocity profile is known (specified), and attention is focused on the distribution of scalar dependent variables of interest in the presence of convection, diffusion, and volumetric source terms [77]. Here, in keeping with well established usage in the literature [77], transport due to overall or gross fluid motion is denoted as convection[1] and transport due to molecular interactions, such as conduction, viscous transport, and mass diffusion, is labelled as diffusion. Only steady, two-dimensional, convection-diffusion problems are considered in this thesis.

The partial differential equations which govern steady, two-dimensional convection-diffusion problems can be cast in the following general form in the Cartesian coordinate system [77]:

$$\frac{\partial(\rho u \Phi)}{\partial x} + \frac{\partial(\rho v \Phi)}{\partial y} = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial \Phi}{\partial y}\right) + S \qquad (2.1)$$

[1]This phenomena is also referred to as advection in the literature [55].

The fluid flow field in this equation is specified, and it is assumed that it satisfies the steady continuity equation:

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{2.2}$$

In these equations, Φ is the scalar dependent variable of interest; u and v are the known velocity components in x and y directions, respectively; ρ is the mass density of the fluid; Γ is the diffusion coefficient associated with Φ; and S is the volumetric source term. In problems where the diffusion of Φ in not proportional to $\vec{\nabla}\Phi$, the diffusion terms that do not fit within the gradient expression are included in the source term, and equation (2.1) remains valid [77].

In the proposed finite volume method (FVM), the calculation domain is first divided into suitable elements and control volumes. Equation (2.1) in then integrated over each of the control volumes to obtain integral conservation equations. Appropriate functions are then prescribed to interpolate nodal values of the dependent variables over each element in the calculation domain. These interpolation functions are used to derive of algebraic approximations to the integral conservation equations. The algebraic approximations, or discretization equations, constitute a set of simultaneous algebraic equations that could, in general, be nonlinear, and coupled to other sets of discretization equations associated with other dependent variables. An iterative algorithm is used to solve this set, or sets, of discretization equations.

In this chapter, the aforementioned steps and a method for the solution of the set, or sets, of algebraic discretization equations are described.

2.2 Domain Discretization

As mentioned earlier in this thesis, only plane two-dimensional rectangular domains are considered in this investigation. Such domains are first divided into rectangular elements, using grid lines which are parallel to the x- and y-coordinate axes,

as shown in Fig. 2.1. It should be noted that the spacing between these grid lines could be nonuniform. Indeed, in problems with highly nonuniform distributions of the dependent variables, a corresponding nonuniform grid is computationally more efficient than a uniform grid [77]. A nonuniform grid is shown in Fig. 2.2.

After the discretization of the domain into rectangular elements, each element is subdivided into four equal rectangular regions, or sub-control volumes, by joining the midpoints of opposite sides, as shown in Fig. 2.3. Collectively, these sub-control volumes form rectangular control volumes around each node in the calculation domain, as illustrated in Fig. 2.4. These control volumes have the following desirable features: (i) they do not overlap; (ii) collectively, they fill the calculation domain completely and exactly; and (iii) their faces lie midway between adjacent nodes along the grid lines, and intersect the grid lines perpendicularly. The first two of these features facilitate the formulation of a conservative numerical scheme [77, 97]; and the third feature enables second-order accurate algebraic approximations of the diffusive fluxes, using linear interpolation of the dependent variables in each element.

2.3 Integral Conservation Equation for a Control Volume

Consider a typical control volume as shown in Fig. 2.4. Upon integration of equation (2.1) over this control volume, the integral conservation equation will be of the general form:

$$\int\int_{CV}\left[\frac{\partial(\rho u\Phi)}{\partial x}+\frac{\partial(\rho v\Phi)}{\partial y}\right]dxdydz = \int\int_{CV}\left[\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\Phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial\Phi}{\partial y}\right)+S\right]dxdydz \quad (2.3)$$

With respect to the control volume surrounding node (i,j), also denoted as a particular node 'P', in Fig.2.4, and noting that in this two-dimensional formulation the calculation domains are assumed to be of unit dimension in the z direction, Eq. 2.3 can be rearranged as follows:

$$\left[\int_a^o \rho u\Phi dy+\int_o^d \rho v\Phi dx-\int_a^o\left(\Gamma\frac{\partial\Phi}{\partial x}\right)dy-\int_o^d\left(\Gamma\frac{\partial\Phi}{\partial y}\right)dx-\int\int_{Paod}Sdxdy\right] \quad (2.4)$$

+ [similar contributions from other elements associated with node (i,j)]

+ [boundary contributions, if applicable] = 0

This form of the integral conservation equation emphasizes that it can be assembled on an element-by-element basis. Eq. 2.4 is approximated in this work by the following equation:

$$\left[(\rho u\phi)_a\frac{\delta y_j}{2}+(\rho v\phi)_d\frac{\delta x_i}{2}-\left(\Gamma\frac{\delta\Phi}{\delta x}\right)_a\frac{\delta y_j}{2}-\left(\Gamma\frac{\delta\Phi}{\delta y}\right)_d\frac{\delta x_i}{2}-(S_e)\frac{\delta x_i\delta y_j}{4}\right] \quad (2.5)$$

+ [similar contributions from other elements associated with node (i,j)]

+ [boundary contributions, if applicable] = 0

Where δx_i and δy_j are the lengths of the sides of each element (i,j) in the x and y directions, respectively, as shown in the Fig. 2.3. These lengths are distances between adjacent nodes in x and y directions.

The values of ϕ and $\vec{\nabla}\phi$ at points a, b, c, and d, that will be referred to as integration points, will be obtained based on interpolation functions which will be defined in the next section.

2.4 Interpolation Functions

This section provides interpolation functions for the thermophysical properties of the fluid, the volumetric source term, and the dependent variables Φ, u, and v.

2.4.1 Interpolation of ρ, Γ and S

Values of ρ and Γ are supplied at the centroid of the rectangular elements and are assumed to prevail over the corresponding element. The source term, S, is linearized with respect to its dependence on ϕ, if required, and expressed as follows [77]:

$$S = S_C + S_P\phi \tag{2.6}$$

The values of S_C and S_P are calculated at the centroid of each element, and assumed to prevail within the corresponding element. Hence, the contribution of element (i,j), shown in Fig. 2.3, to the integral source term in Eq. 2.4 can be written as:

$$\int\int_{Pacd} S dx dy \simeq (S_{i,j})\frac{\delta x_i \delta y_j}{4} = (S_{C_{i,j}} + S_{P_{i,j}}\phi)\frac{\delta x_i \delta y_j}{4} \tag{2.7}$$

2.4.2 Interpolation of Velocities

In each rectangular element, the mid-side values of u and v are calculated using functional distributions of the velocity components, if applicable. If only nodal values of u and v are specified, linear interpolation is used to calculate the mid-side values[2]. Thus, with reference to Fig. 2.3,

$$u_a = \frac{u_{i,j} + u_{i+1,j}}{2} \qquad u_c = \frac{u_{i,j+1} + u_{i+1,j+1}}{2}$$
$$v_b = \frac{v_{i+1,j} + v_{i+1,j+1}}{2} \qquad v_d = \frac{v_{i,j} + v_{i,j+1}}{2} \tag{2.8}$$

[2]In fluid flow problems, where the computation of u and v is required, special interpolation functions are used to interpolate these variables in the mass flux terms. These functions will be introduced in chapter 3.

2.4.3 Interpolation of Φ

As was discussed in section 1.3.1, the proper approximation of the convection term, with acceptable accuracy and without creation of unphysically numerical oscillations in the solution, has been a challenging task in computational fluid dynamics over the last 25 years. One of the goals of this thesis is to compare the accuracy of six schemes for obtaining algebraic approximations of the convection and diffusion terms. These schemes are:

1. Central Difference Scheme (CDS)
2. Upwind Difference Scheme (UDS)
3. Skew Upstream Difference Scheme (SUDS)
4. Linear Skew Difference scheme (LSD)
5. Quadratic Upstream Interpolation for Convective Kinematics (QUICK)
6. MAss Weighted difference scheme (MAW)

In the proposed method, each rectangular element contributes to the diffusive and convective transport across the boundaries of four sub-control volumes. With reference to the typical element and the notation given in Fig. 2.3, these contributions to the diffusive and convective transport terms in Eqs. 2.4 and 2.5 are approximated as follows:

$$\begin{aligned}
\int_a^o \rho u\Phi dy &= \rho_e u_a \phi_a \delta y_j/2 & \int_a^o (\Gamma\frac{\partial\Phi}{\partial x})dy &= \Gamma_e(\frac{\delta\Phi}{\delta x})_a\frac{\delta y_j}{2}\\
\int_o^c \rho u\Phi dy &= \rho_e u_c \phi_c \delta y_j/2 & \int_o^c (\Gamma\frac{\partial\Phi}{\partial x})dy &= \Gamma_e(\frac{\delta\Phi}{\delta x})_c\frac{\delta y_j}{2}\\
\int_d^o \rho v\Phi dx &= \rho_e v_d \phi_d \delta x_i/2 & \int_d^o (\Gamma\frac{\partial\Phi}{\partial y})dx &= \Gamma_e(\frac{\delta\Phi}{\delta y})_d\frac{\delta x_i}{2}\\
\int_o^b \rho v\Phi dx &= \rho_e v_b \phi_b \delta x_i/2 & \int_o^b (\Gamma\frac{\partial\Phi}{\partial y})dx &= \Gamma_e(\frac{\delta\Phi}{\delta y})_b\frac{\delta x_i}{2}
\end{aligned} \qquad (2.9)$$

In these equations, ρ_e and Γ_e are the values of density and diffusion coefficient, respectively, appropriate for the element under consideration; and u_a, v_b, u_c, and v_d are values of the velocity components at integration points a, b, c, and d, and these are obtained using the interpolation practices discussed earlier in section 2.4.2. All of the aforementioned six schemes use the same interpolation function to approximate Φ in the diffusion terms at integration points. However, the interpolation functions that are used to obtain the values of ϕ in the convection terms at integration points are different in the six schemes of interest.

The approximation of Φ in the diffusion terms will be discussed first. Then, the treatment of the convection terms in each of the six schemes will be discussed separately.

Approximation of Φ in the Diffusion Terms

In all of the above schemes, piecewise-linear interpolation of Φ along grid lines is used to approximate the diffusion terms at the integration points within each element, which results in:

$$\left(\frac{\partial\phi}{\partial x}\right)_a = \frac{\phi_{i+1,j} - \phi_{i,j}}{\delta x_i} \qquad \left(\frac{\partial\phi}{\partial x}\right)_c = \frac{\phi_{i+1,j+1} - \phi_{i,j+1}}{\delta x_i}$$
$$\left(\frac{\partial\phi}{\partial y}\right)_b = \frac{\phi_{i+1,j+1} - \phi_{i+1,j}}{\delta y_j} \qquad \left(\frac{\partial\phi}{\partial y}\right)_d = \frac{\phi_{i,j+1} - \phi_{i,j}}{\delta y_j} \tag{2.10}$$

for the gradient of ϕ in Eq. 2.9.

Approximation of Φ in the Convection Terms

Central Difference Scheme (CDS)

For the domain discretization used in the proposed method, this scheme is equivalent to the use of linear interpolation for the approximation of the integration-point

values of ϕ. This treatment, with reference to Fig. 2.3, results in the following expressions for ϕ_a, ϕ_b, ϕ_c, and ϕ_d,

$$\phi_a = \frac{\phi_{i,j} + \phi_{i+1,j}}{2} \qquad \phi_c = \frac{\phi_{i,j+1} + \phi_{i+1,j+1}}{2}$$
$$\phi_b = \frac{\phi_{i+1,j} + \phi_{i+1,j+1}}{2} \qquad \phi_d = \frac{\phi_{i,j} + \phi_{i+1,j+1}}{2} \tag{2.11}$$

Upwind Difference Scheme (UDS)

This scheme, which was proposed by Courant et al. [23], and by others [77], assumes that the value of Φ at an interface is equal to the value of ϕ at the grid point on the upwind side of the face. Thus, this scheme takes into consideration the direction of the fluid flow, at least to some extent, and, with reference to Fig. 2.3, it leads to the following equations for the integration-point values of ϕ:

Case 1: $u > 0$ and for all v

$$\phi_a = \phi_{i,j} \qquad \phi_c = \phi_{i,j+1} \tag{2.12}$$

Case 2: $u < 0$ and for all v

$$\phi_a = \phi_{i+1,j} \qquad \phi_c = \phi_{i+1,j+1} \tag{2.13}$$

Case 3: $v > 0$ and for all u

$$\phi_b = \phi_{i+1,j} \qquad \phi_d = \phi_{i,j} \tag{2.14}$$

Case 4: $v < 0$ and for all u

$$\phi_b = \phi_{i+1,j+1} \qquad \phi_d = \phi_{i,j+1} \tag{2.15}$$

Skew Upstream Difference Scheme (SUDS)

This scheme, first proposed by Raithby [89], takes into consideration the skewness of the flow to the grid lines, while accounting for the one-way or upwind behaviour of

convective transport. In this scheme, the value of ϕ at the integration point of interest is assumed to be equal to the value of ϕ at the intersection of an element side and the upstream extension of the velocity vector passing through the integration point. The value of ϕ at this intersection point is obtained by a linear interpolation of the value of ϕ at the two adjacent nodes which lie on either sides of the intersection point on the element side.

Thus with respect to Fig. 2.5 and for $u>0$ and $v>0$, when the upstream extension of the velocity vector passing through the integration point 'a' intersects the grid line between nodes "P" and "S", the value of $\phi_a = \phi_{us}$ is obtained by a linear interpolation of ϕ_P and ϕ_S: if, however, the upstream extension of the velocity vector through 'a' intersects the grid line between nodes "S" and "SE", as shown in Fig. 2.6, then it is assumed that $\phi_a = \phi_S$ [89]. Therefore, with reference to Figs. 2.3, 2.5, and 2.6,

If $u_a > 0$ and $v_a > 0$, and $\frac{|v_a|}{|u_a|} \leq \frac{2\delta y_{j-1}}{\delta x_i}$, then:

$$\phi_a = (1 - CK_a)\phi_{i,j} + CK_a\phi_{i,j-1} \tag{2.16}$$

with $CK_a = \frac{|v_a|}{|u_a|} * \frac{\delta x_i}{2\delta y_{j-1}}$

If $u_a > 0$ and $v_a > 0$, and $\frac{|v_a|}{|u_a|} > \frac{2\delta y_{j-1}}{\delta x_i}$, then:

$$\phi_a = \phi_{i,j-1} \tag{2.17}$$

If $u_a > 0$ and $v_a < 0$, and $\frac{|v_a|}{|u_a|} \leq \frac{2\delta y_j}{\delta x_i}$, then:

$$\phi_a = (1 - CK_a)\phi_{i,j} + CK_a\phi_{i,j+1} \tag{2.18}$$

with $CK_a = \frac{|v_a|}{|u_a|} * \frac{\delta x_i}{2\delta y_j}$

If $u_a > 0$ and $v_a < 0$, and $\frac{|v_a|}{|u_a|} > \frac{2\delta y_j}{\delta x_i}$, then:

$$\phi_a = \phi_{i,j+1} \tag{2.19}$$

If $u_a < 0$ and $v_a > 0$, and $\frac{|v_a|}{|u_a|} \leq \frac{2\delta y_{j-1}}{\delta x_i}$, then:

$$\phi_a = (1 - CK_a)\phi_{i+1,j} + CK_a\phi_{i+1,j-1} \tag{2.20}$$

with $CK_e = \frac{|v_e|}{|u_e|} * \frac{\delta x_i}{2\delta y_{j-1}}$

If $u_e < 0$ and $v_e > 0$, and $\frac{|v_e|}{|u_e|} > \frac{2\delta y_{j-1}}{\delta x_i}$, then:

$$\phi_e = \phi_{i+1,j-1} \tag{2.21}$$

If $u_e < 0$ and $v_e < 0$, and $\frac{|v_e|}{|u_e|} \leq \frac{2\delta y_j}{\delta x_i}$, then:

$$\phi_e = (1 - CK_e)\phi_{i+1,j} + CK_e\phi_{i+1,j+1} \tag{2.22}$$

with $CK_e = \frac{|v_e|}{|u_e|} * \frac{\delta x_i}{2\delta y_j}$

If $u_e < 0$ and $v_e < 0$, and $\frac{|v_e|}{|u_e|} > \frac{2\delta y_{j-1}}{\delta x_i}$, then:

$$\phi_e = \phi_{i+1,j+1} \tag{2.23}$$

SUDS approximations for ϕ_b, ϕ_c, and $\phi_d$are obtained analogously.

Linear Skew Difference scheme (LSD)

This scheme, proposed by Huget [53], is basically quite similar to the SUDS scheme. In this scheme, as in the SUDS scheme, the value of ϕ at an integration point is assumed to be equal to the value of ϕ at the intersection point of an element side and the upstream extension of the velocity vector passing through the integration point of interest. However, contrary to the SUDS scheme, regardless of which element side intersects with the upstream extension of the velocity vector passing through the integration point of interest, linear interpolation of the values of ϕ at two adjacent nodes, which lie on either side of the intersection point, on the element side, is used to obtain the approximation to ϕ at the integration point.

The implementation and the programming of this scheme is more involved than the CDS, UDS, and even SUDS schemes. For positive velocity components, $u > 0$ and $v > 0$, the interpolation functions for the approximation of ϕ at the integration points, see Figs. 2.3, 2.5 and Fig 2.6, are:

Case 1: $\left|\frac{v}{u}\right| \leq \frac{2\delta_y}{\delta_x}$

$$\begin{aligned} \phi_a &= (1 - CK_a)\phi_{i,j} + CK_a\phi_{i,j-1} \\ \phi_c &= (1 - CK_c)\phi_{i,j+1} + CK_c\phi_{i,j} \end{aligned} \tag{2.24}$$

Case 2: $\left|\frac{v}{u}\right| > \frac{2\delta_y}{\delta_x}$

$$\begin{aligned} \phi_a &= (1 - CKP_a)\phi_{i+1,j-1} + CKP_a\phi_{i,j-1} \\ \phi_c &= (1 - CKP_c)\phi_{i+1,j} + CKP_c\phi_{i,j} \end{aligned} \tag{2.25}$$

Case 3: $\left|\frac{u}{v}\right| \leq \frac{2\delta_x}{\delta_y}$

$$\begin{aligned} \phi_b &= (1 - CK_b)\phi_{i+1,j} + CK_b\phi_{i,j} \\ \phi_d &= (1 - CK_d)\phi_{i,j} + CK_d\phi_{i-1,j} \end{aligned} \tag{2.26}$$

Case 4: $\left|\frac{u}{v}\right| > \frac{2\delta_x}{\delta_y}$

$$\begin{aligned} \phi_b &= (1 - CKP_b)\phi_{i,j+1} + CKP_b\phi_{i,j} \\ \phi_d &= (1 - CKP_d)\phi_{i-1,j+1} + CKP_d\phi_{i-1,j} \end{aligned} \tag{2.27}$$

The definitions of the CK and CKP terms can be obtained using procedures akin to those used to obtain similar terms in SUDS. Details are obtained in the work of Huget [53].

Quadratic Upstream Interpolation for Convective Kinematics (QUICK)

This scheme, proposed by Leonard [63], improves the CDS scheme by adding to it a correction term which incorporate upstream influence in the interpolation for ϕ. This correction term takes into consideration the one-way behaviour of the convection transport, and is added to the CDS scheme to remedy some of its drawbacks. From the point of view of the mathematics, to derive an equation for calculation of ϕ at the integration point on a grid line of interest, this scheme fits a quadratic curve to

the two upstream and one downstream nodal values of ϕ. Therefore, the equations for obtaining the values of ϕ at the integration points shown in Fig. 2.3 are:

Case 1: **Uniform Mesh** $u>0$ **for all** v

$$\phi_a = \frac{\phi_{i,j} + \phi_{i+1,j}}{2} - \frac{\phi_{i-1,j} + \phi_{i+1,j} - 2\phi_{i,j}}{8}$$
$$\phi_c = \frac{\phi_{i,j+1} + \phi_{i+1,j+1}}{2} - \frac{\phi_{i-1,j+1} + \phi_{i+1,j+1} - 2\phi_{i,j+1}}{8}$$

Case 2: **Uniform Mesh** $u<0$ **for all** v

$$\phi_a = \frac{\phi_{i,j} + \phi_{i+1,j}}{2} - \frac{\phi_{i,j} + \phi_{i+2,j} - 2\phi_{i+1,j}}{8}$$
$$\phi_c = \frac{\phi_{i,j+1} + \phi_{i+1,j+1}}{2} - \frac{\phi_{i,j+1} + \phi_{i+2,j+1} - 2\phi_{i+1,j+1}}{8}$$

Case 3: **Uniform Mesh** $v>0$ **for all** u

$$\phi_b = \frac{\phi_{i+1,j} + \phi_{i+1,j+1}}{2} - \frac{\phi_{i+1,j-1} + \phi_{i+1,j+1} - 2\phi_{i+1,j}}{8}$$
$$\phi_d = \frac{\phi_{i,j} + \phi_{i,j+1}}{2} - \frac{\phi_{i,j-1} + \phi_{i,j+1} - 2\phi_{i,j}}{8} \qquad (2.28)$$

Case 4: **Uniform Mesh** $v<0$ **for all** u

$$\phi_b = \frac{\phi_{i+1,j} + \phi_{i+1,j+1}}{2} - \frac{\phi_{i+1,j} + \phi_{i+1,j+2} - 2\phi_{i+1,j+1}}{8}$$
$$\phi_d = \frac{\phi_{i,j} + \phi_{i,j+1}}{2} - \frac{\phi_{i,j} + \phi_{i,j+2} - 2\phi_{i,j+1}}{8} \qquad (2.29)$$

The forms of these equations for non-uniform grids are some more involved than those for uniform grids. Details are available in the paper by Leonard [63].

MAss Weighted difference scheme (MAW)

This scheme is based on the works of Hassan et al. [45], in the context of finite difference methods, and Schneider and Raw [103] and Saabas [100] in the context of the control-volume finite element methods. The latter adaptation is used in this work. This scheme guarantees that the algebraic approximations to the convection terms have a positive contribution to the coefficients in the discretization equations.

This is done by ensuring, at each element level, that the value of ϕ at a node outside a control volume contributes less to the outflow of ϕ from that control volume than to the inflow into the control volume. Physically, this means that for a scalar to flow out of a control volume, first, it has to come into the control volume.

Suppose that in the element shown in Fig. 2.7, there is an outflow of ϕ from the control volume associated with node (i,j) into the control volume surrounding node (i+1,j) across interface 1: this is equal to $\frac{\rho u_a \phi_a \delta y_j}{2}$. In addition, let there be a flow of ϕ from the control volume associated with node (i,j+1) to the control volume surrounding node (i,j) across interface 4: this is given by $-\frac{\rho v_d \phi_d \delta x_i}{2}$.

It is assumed that the value of ϕ_a transported across '1' has contributions from ϕ_d and $\phi_{i,j}$. Thus, ϕ_a can be expressed as:

$$\phi_a = f\phi_d + (1-f)\phi_{i,j} \tag{2.30}$$

The factor f is chosen so as to ensure, at the element level, that the transport of ϕ_d out of the control volume surrounding node (i,j) is less than or equal to its transport into this control volume: in mathematical terms, the following condition must be satisfied:

$$-\frac{\rho v_d \delta x_i}{2}\phi_d \geq \frac{\rho u_a \delta y_j}{2} f\phi_d \tag{2.31}$$

The mass flow rates across interfaces 1 and 4 are:

$$\dot{m}_1 = \frac{\rho u_a \delta y_j}{2} \qquad \dot{m}_4 = \frac{\rho v_d \delta x_i}{2}$$

Therefore,

$$f = \begin{cases} 0 & if\ \dot{m}_1 > 0\ and\ \frac{-\dot{m}_4}{\dot{m}_1} \leq 0 \\ \frac{-\dot{m}_4}{\dot{m}_1} & if\ \dot{m}_1 > 0\ and\ 0 < \frac{-\dot{m}_4}{\dot{m}_1} < 1 \\ 1 & if\ \dot{m}_1 > 0\ and\ \frac{-\dot{m}_4}{\dot{m}_1} \geq 1 \end{cases} \tag{2.32}$$

The above equation for defining f can be compactly represented by:

If $\dot{m}_1 > 0$, then:

$$f = Min[Max(\frac{-\dot{m}_4}{\dot{m}_1}, 0), 1] \tag{2.33}$$

In general, the equations for obtaining integration-point values of ϕ are the following:

Integration point a:

$$\begin{aligned} &If \quad \dot{m}_1 \geq 0 \qquad \phi_a = f\phi_d + (1-f)\phi_{i,j} \qquad f = Min[Max(\frac{-\dot{m}_4}{\dot{m}_1}, 0), 1] \\ &If \quad \dot{m}_1 \leq 0 \qquad \phi_a = f\phi_b + (1-f)\phi_{i+1,j} \qquad f = Min[Max(\frac{\dot{m}_2}{\dot{m}_1}, 0), 1] \end{aligned} \tag{2.34}$$

Integration point b:

$$\begin{aligned} &If \quad \dot{m}_2 \geq 0 \qquad \phi_b = f\phi_a + (1-f)\phi_{i+1,j} \qquad f = Min[Max(\frac{\dot{m}_1}{\dot{m}_2}, 0), 1] \\ &If \quad \dot{m}_2 \leq 0 \qquad \phi_b = f\phi_c + (1-f)\phi_{i+1,j+1} \qquad f = Min[Max(\frac{-\dot{m}_3}{\dot{m}_2}, 0), 1] \end{aligned} \tag{2.35}$$

Integration point c:

$$\begin{aligned} &If \quad \dot{m}_3 \geq 0 \qquad \phi_c = f\phi_d + (1-f)\phi_{i,j+1} \qquad f = Min[Max(\frac{\dot{m}_4}{\dot{m}_3}, 0), 1] \\ &If \quad \dot{m}_3 \leq 0 \qquad \phi_c = f\phi_b + (1-f)\phi_{i+1,j+1} \qquad f = Min[Max(\frac{-\dot{m}_2}{\dot{m}_3}, 0), 1] \end{aligned} \tag{2.36}$$

Integration point d:

$$\begin{aligned} &If \quad \dot{m}_4 \geq 0 \qquad \phi_d = f\phi_a + (1-f)\phi_{i,j} \qquad f = Min[Max(\frac{-\dot{m}_1}{\dot{m}_4}, 0), 1] \\ &If \quad \dot{m}_4 \leq 0 \qquad \phi_d = f\phi_c + (1-f)\phi_{i,j+1} \qquad f = Min[Max(\frac{\dot{m}_3}{\dot{m}_4}, 0), 1] \end{aligned} \tag{2.37}$$

Hence, in the MAW scheme, the value of ϕ at an integration point is dependent on its value at the other integration points, on the nodal values of ϕ, and on the direction and magnitude of the mass flow rate across the four interfaces within the element. Eqs. (2.34) to (2.37) can be used to obtain the following equations that relate integration-point values of ϕ to the nodal values of ϕ:

$$A_{11}\phi_a + A_{12}\phi_b + A_{13}\phi_c + A_{14}\phi_d = B_{11}\phi_{i,j} + B_{12}\phi_{i+1,j} + B_{13}\phi_{i+1,j+1} + B_{14}\phi_{i,j+1}$$

$$A_{21}\phi_a + A_{22}\phi_b + A_{23}\phi_c + A_{24}\phi_d = B_{21}\phi_{i,j} + B_{22}\phi_{i+1,j} + B_{23}\phi_{i+1,j+1} + B_{24}\phi_{i,j+1}$$

$$A_{31}\phi_a + A_{32}\phi_b + A_{33}\phi_c + A_{34}\phi_d = B_{31}\phi_{i,j} + B_{32}\phi_{i+1,j} + B_{33}\phi_{i+1,j+1} + B_{34}\phi_{i,j+1}$$

$$A_{41}\phi_a + A_{42}\phi_b + A_{43}\phi_c + A_{44}\phi_d = B_{41}\phi_{i,j} + B_{42}\phi_{i+1,j} + B_{43}\phi_{i+1,j+1} + B_{44}\phi_{i,j+1}$$

or, in a compact form:

$$[A][\phi_k] = [B][\phi_{l,m}] \tag{2.38}$$

in which, [A] is a (4 x 4) coefficient matrix of the vector of the 4 integration-point values $[\phi_k]$, and [B] is a (4 x 4) coefficient matrix of the vector of the 4 nodal values $[\phi_{l,m}]$. The above equations must be solved simultaneously in order to express each of the integration-point values of ϕ in terms of the nodal values of ϕ. Therefore:

$$[\phi_k] = [A^{-1}][B][\phi_{l,m}] \Longrightarrow [\phi_k] = [D][\phi_{l,m}] \quad where \quad [D] = [A^{-1}][B] \tag{2.39}$$

Thus, to obtain the integration point values, the matrix [A] must be inverted. Matrix [A] has special characteristics: (i) all diagonal elements have the value of one (unity) and are dominant in the matrix; (ii) in each row, at least one of the elements is zero and the other non-diagonal elements are negative values (< -1). Recognizing these special characteristics of this matrix, a very efficient matrix invertor can be established. In this work, a special pivoting strategy is used to invert matrix [A], which in comparison to the other available methods, is one of the fastest methods and provides excellent accuracy. The elements of the matrices [A] and [B], and also the inversion method for the matrix [A], are presented in Appendix A.

2.5 Derivation of the Discretized Equations

In order to obtain an algebraic approximation to the integral conservation equation for a control volume (Eq. 2.4), Eq. 2.5 is used, along with the interpolation functions

previously developed. The entire calculation domain is visited element-by-element. In each element, algebraic approximations are derived for the contribution of the element to the diffusion, convection, and source terms in the integral conservation equations associated with the control volumes surrounding the four nodes. These contributions, are then assembled in an appropriate manner. Boundary contributions are derived and added to the element contributions, if they are applicable. This section describes the derivation and assembly of these contributions in the proposed FVM.

2.5.1 Element Contribution to the Diffusion Terms

Consider the element (i,j) shown in Fig. 2.3. This element has four interfaces 1, 2, 3 and 4, with increasing numbers in the counter-clockwise direction. Two interfaces are associated with each of the four sub-control volumes in this element. Using the algebraic approximations to the integration-point values of the components of $\vec{\nabla}\phi$, as given in Eq. 2.10, and noting that the value of $\Gamma_e = \Gamma_{i,j}$ prevails over the element, the transport of ϕ by diffusion across the surfaces of the sub-control volume (i,j), in this element, can be approximated as follows:

$$(\Gamma\frac{\delta\Phi}{\delta x})_a\frac{\delta y_j}{2} + (\Gamma\frac{\delta\Phi}{\delta y})_d\frac{\delta x_i}{2} = \Gamma_{i,j}[(\frac{\phi_{i+1,j}-\phi_{i,j}}{\delta x_i})\frac{\delta y_j}{2} + (\frac{\phi_{i,j+1}-\phi_{i,j}}{\delta y_j})\frac{\delta x_i}{2}] \qquad (2.40)$$

The transport of ϕ by diffusion across the surfaces of the other three sub-control volumes in this element, (i+1,j), (i+1,j+1), and (i,j+1), are approximated in a similar manner. The contributions of the other elements are calculated by the same method, and these element contributions to the diffusion terms are assembled appropriately.

2.5.2 Element Contribution to the Convection Terms

In general, the convection contributions of element (i,j), see Fig. 2.3, to the integral conservation equations associated with nodes (i,j), (i+1,j), (i+1,j+1), and

(i,j+1) can be expressed as follows. Let:

$$\dot{m}_1 = \frac{\rho u_a \delta y_j}{2} \quad ; \quad \dot{m}_2 = \frac{\rho v_b \delta x_i}{2} \quad ; \quad \dot{m}_3 = \frac{\rho u_c \delta y_j}{2} \quad ; \quad \dot{m}_4 = \frac{\rho v_d \delta x_i}{2} \tag{2.41}$$

Then the contribution of the element (i,j) to the *outflow* of ϕ by convection across the surfaces of

(i) the control volume surrounding node (i,j) is:

$$[\int_a^o \rho u \Phi dy + \int_o^d \rho v \Phi dx] \simeq \dot{m}_1 \phi_a + \dot{m}_4 \phi_d \tag{2.42}$$

(ii) the control volume surrounding node (i+1,j) is:

$$[\int_b^o \rho v \Phi dx - \int_o^a \rho u \Phi dy] \simeq \dot{m}_2 \phi_b - \dot{m}_1 \phi_a \tag{2.43}$$

(iii) the control volume surrounding node (i+1,j+1) is:

$$[-\int_c^o \rho u \Phi dy - \int_o^b \rho v \Phi dx] \simeq -\dot{m}_3 \phi_c - \dot{m}_2 \phi_b \tag{2.44}$$

(iv) the control volume surrounding node (i,j+1) is:

$$[-\int_d^o \rho v \Phi dx + \int_o^c \rho u \Phi dy] \simeq -\dot{m}_4 \phi_d + \dot{m}_3 \phi_c \tag{2.45}$$

The expressions that give the integration-point values, ϕ_a, ϕ_b, ϕ_c, and ϕ_d, in terms of the nodal values of ϕ are different for the six schemes considered in this work. These expressions were presented in section 2.4.3. When these expressions are substituted into the above equations for the convective transport terms, algebraic approximations to the convection contributions of element (i,j) are obtained. These approximations are different for the six different schemes considered here. The convection contributions of other elements are calculated in a similar manner, and these element contributions are assembled appropriately.

2.5.3 Element Contribution to the Source Terms

Consider the element (i,j) shown in Fig. 2.3. Using approximations similar to those used in deriving Eq. 2.7, the following expressions are obtained for the contributions

of element (i,j) to the source terms in the integral conservation equations for the control volumes surrounding its four nodes:

(i) the control volume surrounding node (i,j):

$$\iint_{i,j} S dx dy \cong (S_{C_{i,j}} + S_{P_{i,j}} \phi_{i,j}) \frac{\delta x_i \delta y_j}{4} \tag{2.46}$$

(ii) the control volume surrounding node (i+1,j):

$$\int\int_{i+1,j} S dx dy \cong (S_{C_{i,j}} + S_{P_{i,j}} \phi_{i+1,j}) \frac{\delta x_i \delta y_j}{4} \tag{2.47}$$

(iii) the control volume surrounding node (i+1,j+1):

$$\int\int_{i+1,j+1} S dx dy \cong (S_{C_{i,j}} + S_{P_{i,j}} \phi_{i+1,j+1}) \frac{\delta x_i \delta y_j}{4} \tag{2.48}$$

(iv) the control volume surrounding node (i,j+1):

$$\int\int_{i,j+1} S dx dy \cong (S_{C_{i,j}} + S_{P_{i,j}} \phi_{i,j+1}) \frac{\delta x_i \delta y_j}{4} \tag{2.49}$$

In these equations, $S_{C_{i,j}}$ and $S_{P_{i,j}}$ pertain to *element* (i,j): as stated earlier, they are stored at the centroid and assumed to prevail over the element. The source-term contributions of other elements are approximated similarly, and these element contributions are assembled appropriately.

2.5.4 Discretized Equations for Internal Nodes

When algebraic approximations to the element contributions to diffusion, convection, and source terms, as derived in the previous section, are assembled appropriately for all elements, the complete algebraic discretized equations are obtained for *internal* nodes. These discretized equations can be cast in the following general forms:

CDS and UDS schemes:

$$\begin{aligned} a_{P_{i,j}} \phi_{i,j} &= a_{E_{i,j}} \phi_{i+1,j} + a_{N_{i,j}} \phi_{i,j+1} \\ &\quad + a_{W_{i,j}} \phi_{i-1,j} + a_{S_{i,j}} \phi_{i,j-1} + b_{i,j} \end{aligned} \tag{2.50}$$

SUDS, LSD and MAW schemes:

$$\begin{aligned} a_{P_{i,j}}\phi_{i,j} = {} & a_{E_{i,j}}\phi_{i+1,j} + a_{N_{i,j}}\phi_{i,j+1} + a_{W_{i,j}}\phi_{i-1,j} \\ & + a_{S_{i,j}}\phi_{i,j-1} + a_{NE_{i,j}}\phi_{i+1,j+1} + a_{NW_{i,j}}\phi_{i-1,j+1} \\ & + a_{SE_{i,j}}\phi_{i+1,j-1} + a_{SW_{i,j}}\phi_{i-1,j-1} + b_{i,j} \end{aligned} \tag{2.51}$$

QUICK scheme:

$$\begin{aligned} a_{P_{i,j}}\phi_{i,j} = {} & a_{E_{i,j}}\phi_{i+1,j} + a_{N_{i,j}}\phi_{i,j+1} + a_{W_{i,j}}\phi_{i-1,j} \\ & + a_{S_{i,j}}\phi_{i,j-1} + a_{EE_{i,j}}\phi_{i+2,j} + a_{NN_{i,j}}\phi_{i,j+2} \\ & + a_{WW_{i,j}}\phi_{i-2,j} + a_{SS_{i,j}}\phi_{i,j-2} + b_{i,j} \end{aligned} \tag{2.52}$$

2.5.5 Implementation of the Boundary Conditions

After the assembly of element contributions to the diffusion, convection, and source terms, the discretized equations for the control volumes associated with boundary nodes are not complete. To complete these equations, the transport of ϕ across the boundary must be properly included. Boundary conditions usually encountered in convection-diffusion problems can be categorized as follows:

Dirichlet or Specified-Value Boundary Condition

In this case, the value of ϕ on the boundary is known, and the discretization equation for the boundary node can be replaced by :

$$\phi_{boundary} = \phi_{specified} \tag{2.53}$$

After calculation of all the unknown ϕ values, the flux of ϕ across the boundary control surfaces can be calculated, using the overall balance on ϕ at the boundary control volume of interest. It is to be noted that for nodes located at inflow boundaries, or boundaries across which fluid flows into the calculation domain, the value of ϕ is usually specified.

Neumann or Specified-Flux Boundary Condition

In this case, the diffusion flux of ϕ normal to the boundary is specified, either as a constant value at each boundary node, or as a known distribution of flux across the boundary of interest. This boundary condition can take various forms:

Case 1: Given value : $q_{in,boundary} = q_{specified}$

Case 2: Given convection coefficient

. and reference temperature: $q_{in,boundary} = h(T_\infty - T_{boundary})$

Case 3: Radiation condition: $q_{in,boundary} = \mathcal{F}(T_{Ref}^4 - T_{boundary}^4)$

Consider element (i,j) shown in Fig. 2.3, and suppose that the side joining nodes (i,j) and (i+1,j) lies on the domain boundary. For this case, after assembling of all element contributions, as discussed earlier, the discretized equation associated with boundary node (i,j) is completed as follows:

Case 1:

$$b_{i,j} = b_{i,j} + q_{specified}(\delta x_i)/2 \tag{2.54}$$

Case 2:

$$\begin{aligned} b_{i,j} &= b_{i,j} + hT_\infty(\delta x_i)/2 \\ a_{P_{i,j}} &= a_{p_{i,j}} + h(\delta x_i)/2 \end{aligned} \tag{2.55}$$

Case 3: In the case of radiation flux, it is first linearized about a guess or available value of the boundary-node temperature $T_{i,j}^*$:

$$\begin{aligned} q_{i,j} &= \mathcal{F}[T_{ref}^2 + T_{i,j}^{*2}][T_{ref} + T_{i,j}^*][T_{ref} - T_{i,j}^*] \\ &= \mathcal{H}[T_{ref} - T_{i,j}^*] \end{aligned} \tag{2.56}$$

The discretized equation is then completed by doing the following:

$$\begin{aligned} b_{i,j} &= b_{i,j} + \mathcal{H}T_{ref}(\delta x_i)/2 \\ a_{P_{i,j}} &= a_{P_{i,j}} + \mathcal{H}(\delta x_i)/2 \end{aligned} \tag{2.57}$$

Outflow Boundary condition

An outflow boundary is one across which fluid flows out of the calculation domain. If the value of ϕ at the outflow boundary is specified, the implementation of this boundary condition is the same as that discussed earlier. However, usually, neither the value of ϕ nor its flux are known at the outflow boundary. The treatment of outflow boundary conditions in this work is identical to its treatment in Ref. [77]: it is assumed that the diffusion flux across an outflow boundary is negligible in comparison to the convection flux at this boundary. Furthermore, at an outflow boundary node, it is assumed that the value of ϕ and the outward velocity component normal to the boundary prevail over the boundary surface associated with this node.

Again, consider element (i,j) shown in Fig. 2.3, and now suppose that the side joining nodes (i+1,j) and (i+1,j+1) lies on an outflow boundary. Then, for node (i+1,j), the convection transport out of its control volume across this side is given by $\rho_e u_{i+1,j}\phi_{i+1,j}(\delta y_j)/2$. After assembly of all element contributions, the corresponding discretization equation is completed as follows:

$$a_{P_{i+1,j}} = a_{P_{i+1,j}} + \rho_{i,j} u_{i+1,j}(\delta y_j)/2 \tag{2.58}$$

Similar treatment is accorded to other nodes that lie on outflow boundaries.

2.5.6 Final Form of the Discretized Equations

The completed discretized equations can be cast in the following general form:

$$a_{P_{i,j}}\phi_{i,j} = \sum_{nb} a_{nb_{i,j}}\phi_{nb_{i,j}} + b_{i,j} \tag{2.59}$$

where the summation is taken over the neighbouring nodes around node (i,j). Eqs. 2.50 to 2.52 show the neighbour nodes which are involved in each scheme. The maximum number of neighbouring nodes for the CDS and UDS schemes are four; and for

the SUDS, LSD, QUICK and MAW schemes, the maximum number of neighbouring nodes are eight.

2.6 Solution of the Discretized Equations

The set of algebraic discretization equations derived in previous sections have to be solved simultaneously, since they are dependent on each other. If these equations happen to be nonlinear, they have to be linearized and solved iteratively. One of the methods to handle nonlinear system of equations is the successive substitution method. In this method, which is the one used in this work, the coefficients are calculated using initial guess-values or the current values of the variables. Each set of linearized algebraic equations is then solved sequentially to calculate new values for the dependent variables, and this procedure is repeated until convergence achieved.

In this work, the grids have a line-by-line structure. For the solution of each set of linear, or linearized, algebraic equations, a line-by-line tri-diagonal matrix algorithm (TDMA) is used with the CDS, UDS, SUDS, LSD, and MAW schemes: with the QUICK scheme, only, a line-by-line penta-diagonal matrix algorithm (PDMA) is used. The line-by-line TDMA that is used for CDS and UDS schemes is the same as that described in Ref. [77]. For SUDS, LSD and MAW schemes, the only difference is that extra coefficients, associated with 'NW', 'NE', 'SE' and 'SW' nodes shown in Fig 2.5, have to be accounted for; otherwise, the line-by-line TDMA is the same as that described in Ref. [77].

With the QUICK scheme, each node can have up to 4 neighbours along a grid line passing through the node: for grid lines in the x direction, an internal node P could have non-zero discretization coefficients a_P, a_W, a_{WW}, a_E *and* a_{EE}; and for grid lines in the y direction, an internal node could have non-zero discretization coefficients a_P, a_S, a_{SS}, a_N; and a_{NN}. Thus a line-by-line PDMA is needed with the

QUICK scheme. The line-by-line PDMA used in this work is taken from Ref. [31].

In both line-by-line TDMA and PDMA, four alternating-direction sweeps, in the positive x direction, positive y direction, negative x direction, and negative y direction, are used. The advantage inherent in doing four alternating direction sweeps, instead of one sweep in any one direction, is that, in general, it provides uniform and faster transmission of the influence of boundary conditions into the interior [77].

In the case of highly nonlinear sets of discretization equations, under-relaxation of the results in each iteration may be necessary to obtain a converged solution. This is also described completely in Ref. [77], and exactly the same under-relaxation method is used here for the convection-diffusion problems.

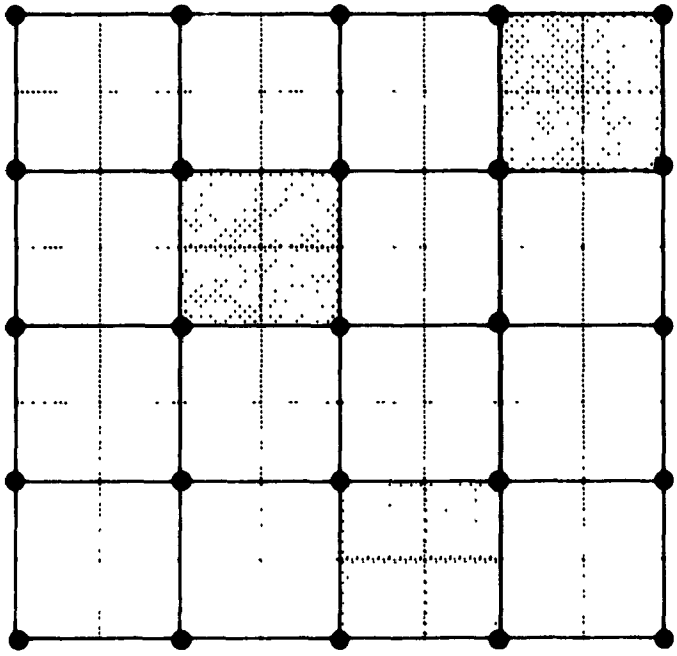


Figure 2.1: Domain Discretization with Rectangular Elements

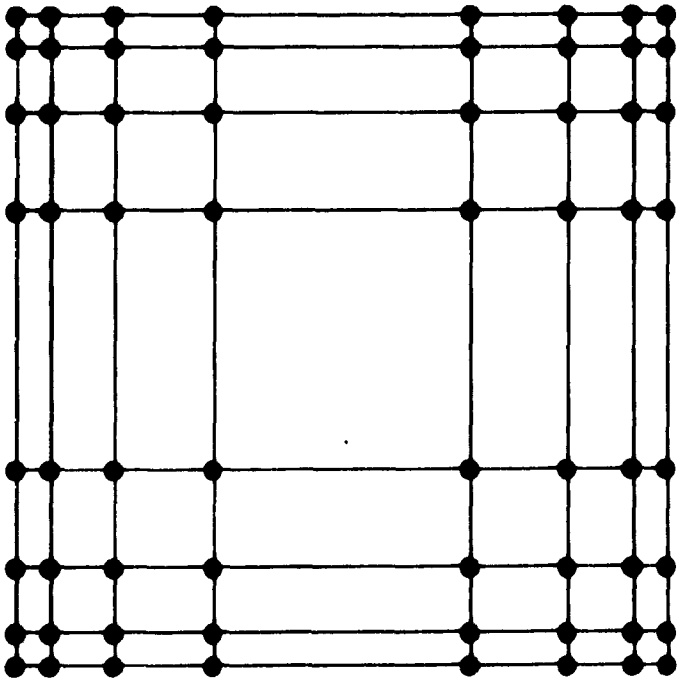


Figure 2.2: Sample Non-Uniform Grid

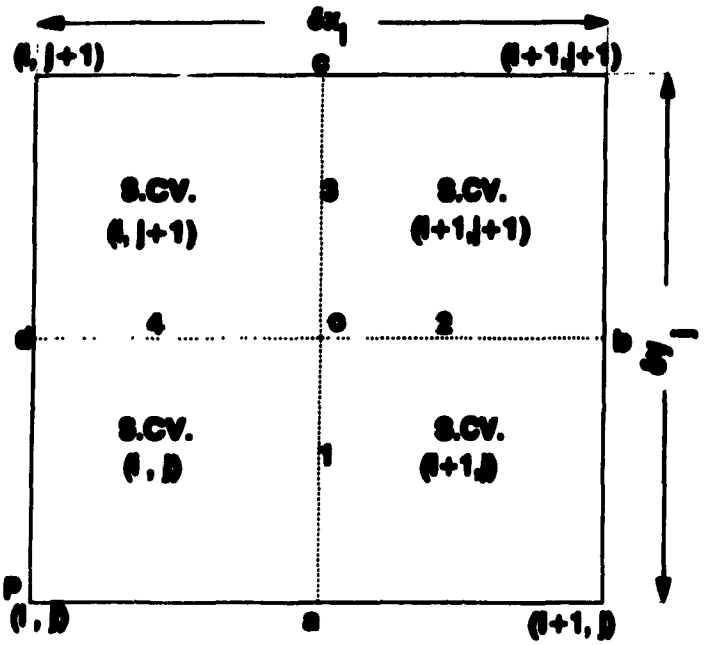


Figure 2.3: Typical Element (i,j)

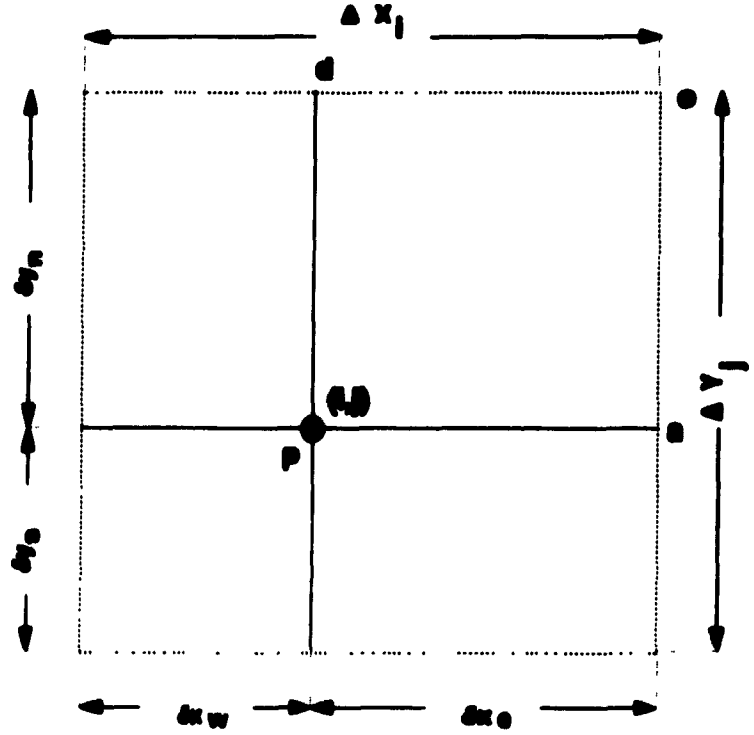


Figure 2.4: Typical Control Volume Surrounding node (i,j)

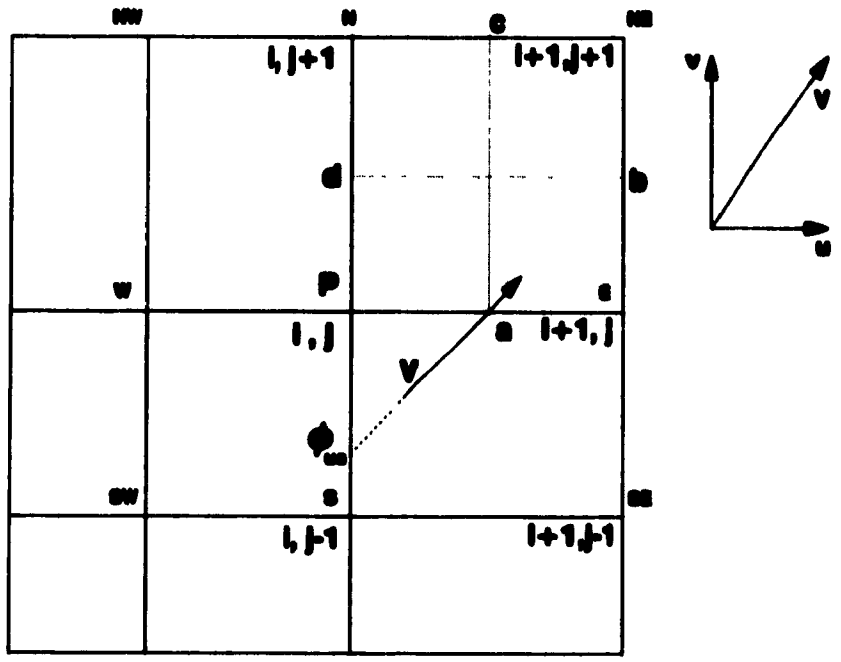


Figure 2.5: Schematic Presentation for SUDS and LSD Schemes: Case 1

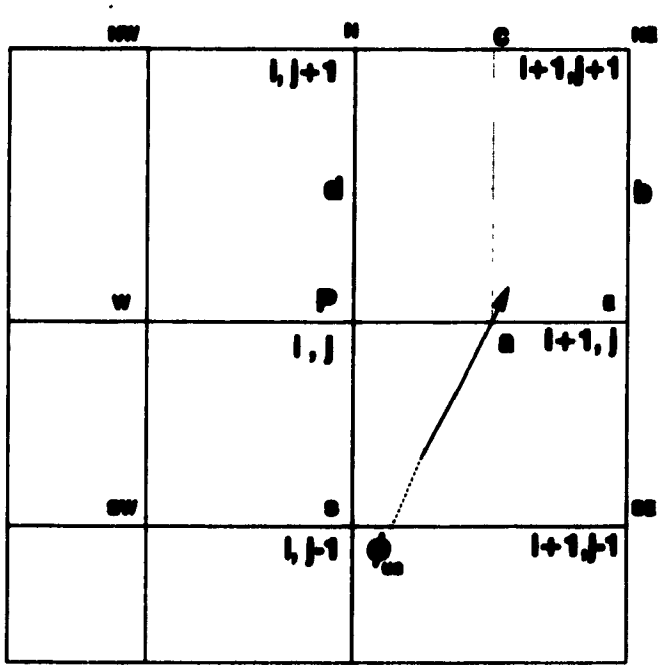


Figure 2.6: Schematic Presentation for SUDS and LSD Schemes: Case 2

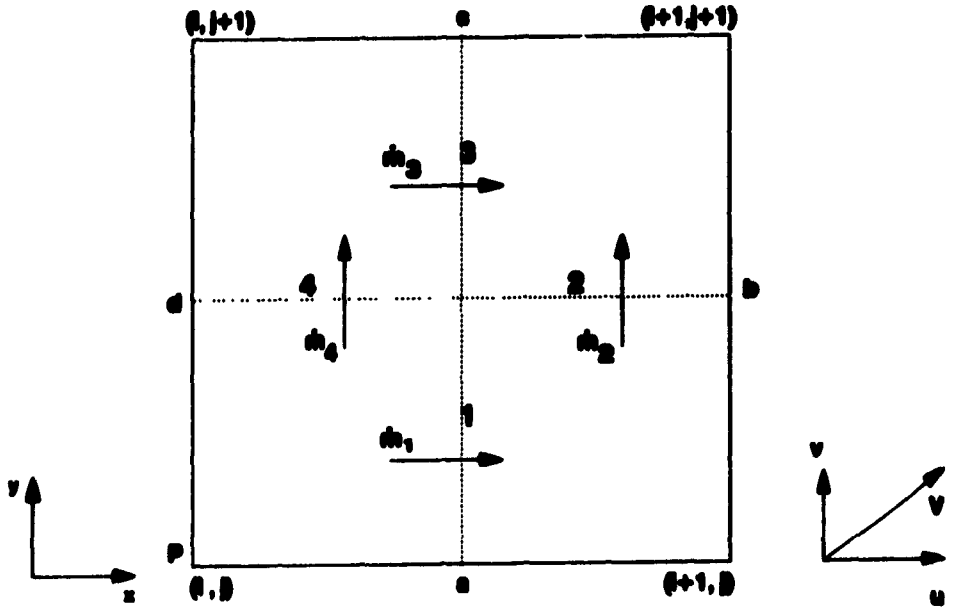


Figure 2.7: Notation and Convention Used in the MAW scheme

Chapter 3

CO-LOCATED FINITE VOLUME METHOD

3.1 Introduction

A finite volume method for steady, two dimensional, incompressible, Newtonian fluid flow in the rectangular geometries is presented in this chapter. The governing equations for such flows can be cast in a form similar to that of Eq. (2.1), the general convection-diffusion equation presented in the previous chapter. Nevertheless, additional developments are required for the solution of fluid flow problems.

As was discussed before in Section 1.3.2, the pressure, which appears in the $-\vec{\nabla}P$ term in the momentum equations, is an unknown in fluid flow problems. In incompressible fluid flows, there is no explicit equation that governs pressure. The pressure distribution is implicitly governed by the momentum and continuity equations: a correct pressure field when substituted in the momentum equations produces a flow field which satisfies the continuity equation [77]. This coupling between the velocity and pressure fields has to be properly accounted for in the solution of the momentum and continuity equations.

Another difficulty encountered in the numerical solution of primitive-variables formulations of incompressible fluid flow is the occurrence of checkerboard pressure fields. A discussion of this problem has also been presented in Section 1.3.2. As was stated there, one way to overcome this difficulty is to stagger the grids for u, v, and P [44, 77]. This approach is effective in formulations based on orthogonal grids, but it involves some tedious book-keeping, and it can not be extended (strictly) to non-orthogonal grids. Several co-located formulations, specially designed to overcome the checkerboard pressure difficulty, have also been proposed. Examples may be found in the works of Hsu [51], Rhie and Chow [96], Shih and Ren [107], Peric [81], and Prakash [85].

In this chapter, a co-located finite-volume method (FVM) patterned after the control-volume finite element method (CVFEM) of Prakash and Patankar [86] and Saabas [100] will be applied to the momentum and continuity equations to derive discretization equations for the primitive dependent variables u, v, and P. At the end of this chapter, an iterative algorithm, based on a sequential solution of u, v, and P, similar to the solution procedure of Saabas [100], will be described. In the next chapter, this algorithm will be investigated and optimized; and enhancements of this solution algorithm, which were developed in this work, will be presented and discussed.

3.2 Governing Equations

The governing equations for steady, laminar, flow of an incompressible Newtonian fluid, in a two-dimensional Cartesian coordinate system, are:

x-momentum:

$$\frac{\partial(\rho u u)}{\partial x} + \frac{\partial(\rho v u)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) + S^u \qquad (3.1)$$

y-momentum:

$$\frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho vv)}{\partial y} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right) + S^v \tag{3.2}$$

continuity:

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{3.3}$$

In these equations, u and v are velocity components in the x and y directions, respectively; p is the pressure; μ is the dynamic viscosity; ρ is the mass density; and S^u and S^v are the rate of generation or volumetric source terms for x- and y-momentum equations, respectively.

The values of μ, ρ, S^u, and S^v may be functions of dependent variables such as temperature. In such cases, additional governing equations, such as the energy equation, must also be solved simultaneously with momentum and continuity equations. However, these additional governing equations are similar to the general convection-diffusion equation described in the previous chapter, so they can be solved by the methods described in that chapter.

In this chapter, the formulation of a co-located finite volume method (FVM) for the solution of momentum and continuity equations (Eqs. 3.1-3.3) will be discussed.

3.3 Domain Discretization

In the proposed FVM, all dependent variables (velocity components, pressure, and other dependent variables, Φ are stored at the same nodal points. Therefore, only one set of control volumes have to be specified. In contrast, in FVMs based on staggered grids, velocity components are calculated at grid points which are displaced with respect to the main-grid points [77]; this necessitates the definition of additional sets of control volumes for the velocity components. Hence, computer codes for FVMs based on staggered grids require considerably more book-keeping than computer codes for

co-located FVMs, because the former have to carry all the indexing and geometric information about the staggered locations and control volumes of the velocity components. This excess book-keeping is especially large for three-dimensional problems. In another words, computer programming is much easier for co-located grids than for staggered grids.

It should also be noted that in FVMs based on co-located grids, the contribution of convection terms to the coefficients in the discretized equations is the same for the x, y, and z momentum equations. Therefore, once these contributions have been calculated for one of these momentum equations, they can also be used for the other momentum equations. However, in FVMs based on staggered grids, convection contributions for each momentum equation have to be calculated separately. Another advantage of a co-located FVM formulation based on orthogonal grids is that it offers the possibility of relatively straightforward extension to non-orthogonal grids. This is not the case with staggered-grid FVMs [110].

As was stated earlier, a co-located formulation is used in this work. Thus, the domain discretization is exactly the same as that used for convection-diffusion problems, which was discussed in section 2.2 and illustrated in Figs. 2.1 and 2.2.

3.4 Integral Conservation Equation for a Control Volume

Consider a typical control volume surrounding node (i,j), as shown in Fig. 2.4. Upon integration of equations 3.1 to 3.3 over this control volume, the conservation

equation can be written as follows:

x-momentum:

$$\int\int_{CV}\left[\frac{\partial(\rho uu)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}\right]dxdydz=$$
$$\int\int_{CV}\left[-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right)+S^u\right]dxdydz \tag{3.4}$$

y-momentum:

$$\int\int_{CV}\left[\frac{\partial(\rho uv)}{\partial x}+\frac{\partial(\rho vv)}{\partial y}\right]dxdydz=$$
$$\int\int_{CV}\left[-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right)+S^v\right]dxdydz \tag{3.5}$$

continuity:

$$\int\int_{CV}\left[\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right]dxdydz=0 \tag{3.6}$$

Noting that in this two-dimensional formulation the calculation domains are assumed to be of unit dimension in the z direction, Eqs. 3.4 - 3.6 can be rearranged as follows:

x-momentum:

$$\left[\int_a^o(\rho uu)dy+\int_o^d(\rho vu)dx+\int\int_{iaod}\left(\frac{\partial p}{\partial x}\right)dxdy-\right.$$
$$\left.\int_a^o\left(\mu\frac{\partial u}{\partial x}\right)dy-\int_o^d\left(\mu\frac{\partial u}{\partial y}\right)dx-\int\int_{Paod}S^u dxdy\right] \tag{3.7}$$

\+ [similar contributions from other elements associated with node (i,j)]

\+ [boundary contributions, if applicable] = 0

y-momentum:

$$\left[\int_a^o(\rho uv)dy+\int_o^d(\rho vv)dx+\int\int_{iaod}\left(\frac{\partial p}{\partial y}\right)dxdy-\right.$$
$$\left.\int_a^o\left(\mu\frac{\partial v}{\partial x}\right)dy-\int_o^d\left(\mu\frac{\partial v}{\partial y}\right)dx-\int\int_{Paod}S^v dxdy\right] \tag{3.8}$$

+ [similar contributions from other elements associated with node (i,j)]

+ [boundary contributions, if applicable] = 0

continuity:

$$\left[\int_a^c (\rho u)dy + \int_c^d (\rho v)dx\right] \tag{3.9}$$

+ [similar contributions from other elements associated with node (i,j)]

+ [boundary contributions, if applicable] = 0

Eqs. 3.7 - 3.9 are approximated in this work by the following equations:

x-momentum:

$$\left[(\rho uu)_a \frac{\delta y_j}{2} + (\rho vu)_d \frac{\delta x_i}{2} + \left(\frac{\partial p}{\partial x}\right)_e \frac{\delta x_i \delta y_j}{4} - \left(\mu \frac{\partial u}{\partial x}\right)_e \frac{\delta y_j}{2} - \left(\mu \frac{\partial u}{\partial y}\right)_d \frac{\delta x_i}{2} - (S_e^u)\frac{\delta x_i \delta y_j}{4}\right] \tag{3.10}$$

+ [similar contributions from other elements associated with node (i,j)]

+ [boundary contributions, if applicable] = 0

y-momentum:

$$\left[(\rho uv)_a \frac{\delta y_j}{2} + (\rho vv)_d \frac{\delta x_i}{2} + \left(\frac{\partial p}{\partial y}\right)_d \frac{\delta x_i \delta y_j}{4} - \left(\mu \frac{\partial v}{\partial x}\right)_e \frac{\delta y_j}{2} - \left(\mu \frac{\partial v}{\partial y}\right)_d \frac{\delta x_i}{2} - (S_e^v)\frac{\delta x_i \delta y_j}{4}\right] \tag{3.11}$$

+ [similar contributions from other elements associated with node (i,j)]

+ [boundary contributions, if applicable] = 0

continuity:

$$\left[(\rho u)_a \frac{\delta y_j}{2} + (\rho v)_d \frac{\delta x_i}{2}\right] \tag{3.12}$$

+ [similar contributions from other elements associated with node (i,j)]

+ [boundary contributions, if applicable] = 0

Where δx_i and δy_j are the lengths of the sides of element (i,j) in the x and y directions, respectively, as shown in Fig. 2.3. There is an extra term in these equations with respect to Eqs. 2.3 - 2.5 for convection-diffusion problems: this is the pressure term.

3.5 Interpolation Functions

This section provides interpolation functions for the thermophysical properties of the fluid, u, v, p, S^u, and S^v.

3.5.1 Interpolation of ρ, μ and S^u, and S^v

The interpolation function for ρ, μ, S^u, and S^v are the same as those described in subsection 2.4.1 for convection-diffusion problems.

3.5.2 Interpolation of Velocities for Momentum Equations

The interpolation functions discussed here are for the velocity components u and v, when they are treated as transported scalars in the momentum equations. These interpolation functions are the same as those that were defined for Φ in the convection-diffusion problems, in the previous chapter. It means that all the six schemes, CDS, UDS, SUDS, LSD, QUICK, and MAW, with all their approximations, are completely applicable here: either u or v components of velocity are substituted for Φ, and $\Gamma = \mu$, in the discretization of x- and y- momentum equations, respectively.

3.5.3 Interpolation of Pressure

Piecewise-linear interpolation functions are used to interpolate the pressure along the grid lines joining adjacent nodes. Therefore, the components of the pressure gradient at integration points for the element shown in Fig. 2.3 are approximated by:

$$\begin{aligned}
\left(\frac{\partial p}{\partial x}\right)_a &= \frac{p_{i+1,j} - p_{i,j}}{\delta x_i} \\
\left(\frac{\partial p}{\partial y}\right)_b &= \frac{p_{i+1,j+1} - p_{i+1,j}}{\delta y_j} \\
\left(\frac{\partial p}{\partial x}\right)_c &= \frac{p_{i+1,j+1} - p_{i,j+1}}{\delta x_i}
\end{aligned} \tag{3.13}$$

$$\left(\frac{\partial p}{\partial y}\right)_d = \frac{p_{i,j+1} - p_{i,j}}{\delta y_j}$$

These approximations are used to calculate the element contributions to pressure gradient terms in the integral x- and y-momentum conservation equations (Eqs. 3.10 and 3.11).

3.5.4 Interpolation of Velocities in Mass Fluxes

The velocity components, u and v, are interpolated by special functions in the discretization of mass-flux terms in the continuity, momentum and convection-diffusion equations. This special treatment is borrowed from the CVFEMs of Prakash and Patankar [86] and Saabas [100], and it is crucial to the success of this co-located FVM. Without this special treatment, or some other special treatments, unphysical checkerboard-type pressure distributions could contaminate the solution provided by co-located FVMs [25, 77, 93].

The special functions employed in this work to interpolate u and v in the mass-flux terms depend on the discretized momentum equations. Therefore, they will be presented later in this chapter.

3.6 Derivation of the Discretized Momentum Equations

The momentum equations are similar to the general convection-diffusion equation discussed in last chapter, and the same control volumes are used for both equations. Therefore, the procedures that were used for discretization of the convection-diffusion equation in section 2.5 , can also be used to discretize the x and y momentum equations. The contributions of convection, diffusion, and source terms (other

than pressure-gradient terms) to the coefficients and constants in the discretized momentum equations are calculated and assembled using the same element-by-element procedure as that discussed in chapter 2. Then, the element contributions to the pressure-gradient terms in the integral momentum-conservation equations are approximated algebraically and assembled to obtain the complete discretized equations.

At each interface of the element (i,j) in Fig. 2.3, the components of the pressure gradient are approximated via Eq. 3.13. Consider integration point a in Fig. 2.3. The pressure gradient at this integration point is approximated as:

$$\left(\frac{\partial p}{\partial x}\right)_a = \frac{p_{i+1,j} - p_{i,j}}{\delta x_i} \tag{3.14}$$

Thus, the contribution of element (i,j) to the pressure-gradient terms in the integral x-momentum equations applied to the control volume around nodes (i,j) and (i+1,j) are approximated as follows:

$$\int\int_{Paod} \left(\frac{\partial p}{\partial x}\right) dxdy = \frac{p_{i+1,j} - p_{i,j}}{\delta x_i}(\delta x_i \delta y_j/4) \tag{3.15}$$

$$\int\int_{Paob} \left((\frac{\partial p}{\partial x}\right))dxdy = \frac{p_{i+1,j} - p_{i,j}}{\delta x_i}(\delta x_i \delta y_j/4) \tag{3.16}$$

In the same way, the contributions to the discretized momentum equations at the other nodes can be calculated. This procedure is then used for all elements to complete the assembly of discretized equations for u and v at internal nodes.

The final form of the resulting discretized equations can be written as:

$$a^u_{P_{i,j}} u_{i,j} = \sum_{nb} a^u_{nb_{i,j}} u_{nb_{i,j}} + b^u_{i,j} \tag{3.17}$$

$$a^v_{P_{i,j}} v_{i,j} = \sum_{nb} a^v_{nb_{i,j}} v_{nb_{i,j}} + b^v_{i,j} \tag{3.18}$$

Again summation is taken over the neighbouring nodes around node (i,j).

3.7 Derivation of the Discretized Pressure Equation

3.7.1 Special Interpolation Functions for u and v in Mass-Flux Terms

This procedure is borrowed from the works of Prakash and Patankar [86] and Saabas [100]. The discretized momentum equations, with respect to the control volume (i,j) shown in Fig. 2.4, are rearranged as follows:

$$u_{i,j} = \frac{\sum_{nb} a^u_{nb_{i,j}} u_{nb} + b^u_{i,j}}{a^u_{P_{i,j}}} - \frac{(\overline{\partial p/\partial x})\Delta V}{a^u_{P_{i,j}}} \tag{3.19}$$

$$v_{i,j} = \frac{\sum_{nb} a^v_{nb_{i,j}} v_{nb} + b^v_{i,j}}{a^v_{P_{i,j}}} - \frac{(\overline{\partial p/\partial y})\Delta V}{a^v_{P_{i,j}}} \tag{3.20}$$

In these expressions, $\overline{\partial p/\partial x}$ and $\overline{\partial p/\partial y}$ are the average of the pressure gradients in the x and y directions, respectively, acting on the control volume (ΔV) surrounding node (i,j). These equation can be rewritten in the following form:

$$u_{i,j} = \hat{u}_{i,j} - d^u_{i,j}\overline{\frac{\partial p}{\partial x}} \tag{3.21}$$

$$v_{i,j} = \hat{v}_{i,j} - d^v_{i,j}\overline{\frac{\partial p}{\partial y}} \tag{3.22}$$

where

$$\begin{aligned} \hat{u}_{i,j} &= \frac{\sum_{nb} a^u_{nb_{i,j}} u_{nb_{i,j}} + b^u_{i,j}}{a^u_{P_{i,j}}} \\ \hat{v}_{i,j} &= \frac{\sum_{nb} a^v_{nb_{i,j}} v_{nb_{i,j}} + b^v_{i,j}}{a^v_{P_{i,j}}} \end{aligned} \tag{3.23}$$

are pseudo-velocities, and

$$\begin{aligned} d^u_{i,j} &= \frac{\Delta V}{a^u_{P_{i,j}}} \\ d^v_{i,j} &= \frac{\Delta V}{a^v_{P_{i,j}}} \end{aligned} \tag{3.24}$$

are pressure coefficients.

Consider integration point a in Fig. 2.3. The value of u_a is obtained by a linear interpolation of $\hat{u}$ and d^u at the nodes (i,j) and (i+1,j), and with the linear interpolation of pressure between these two nodes:

$$u_a = \left[\frac{\hat{u}_{i,j} + \hat{u}_{i+1,j}}{2}\right] - \left[\frac{d^u_{i,j} + d^u_{i+1,j}}{2}\right]\left(\frac{p_{i+1,j} - p_{i,j}}{\delta x_i}\right) \tag{3.25}$$

or:

$$u_a = \hat{u}_a - \frac{d^u_a}{\delta x_i}(p_{i+1,j} - p_{i,j}) \tag{3.26}$$

where:

$$\hat{u}_a = \left[\frac{\hat{u}_{i,j}+\hat{u}_{i+1,j}}{2}\right] \qquad d^u_a = \left[\frac{d^u_{i,j}+d^u_{i+1,j}}{2}\right]$$

With this assumption, the difference in pressure at adjacent nodes is responsible for driving the mass flow through an interface: Since the interface mass flow rates have to satisfy continuity requirements, checkerboard pressure fields are eliminated. In other words, although the nodal velocities that appear in momentum equations do not sense any difference between a uniform and a checkerboard pressure field, the same is not true for velocities that appear in the interface mass fluxes. Therefore, the overall system of equations, does not permit any spurious pressure oscillations in the solution [86, 100].

In the same way, the integration-point velocities for other interfaces of the element shown in Fig. 2.3, can be calculated as follows:

$$\begin{aligned} v_b &= \hat{v}_b - \frac{d^v_b}{\delta y_j}(p_{i+1,j+1} - p_{i+1,j}) \\ u_c &= \hat{u}_c - \frac{d^u_c}{\delta x_i}(p_{i+1,j+1} - p_{i,j+1}) \\ v_d &= \hat{v}_d - \frac{d^v_d}{\delta y_j}(p_{i,j+1} - p_{i,j}) \end{aligned} \tag{3.27}$$

3.7.2 Assembly of the Discretized Pressure Equations

In order to assemble the discretized pressure equations, an element-by-element procedure, similar to that used for assembling the discretized convection-diffusion equation, is used. Consider the element shown in Fig. 2.3. In this element, the values of $\hat{u}$, $\hat{v}$, d^u and d^v are first calculated for integration points a to d, using piecewise-linear interpolation of current nodal values of these quantities, as given by Eqs. 3.23-3.24. Then the contributions to the coefficients of the discretized pressure equation, for each node in this element, can be obtained by using Eqs. 3.26 and 3.27 for calculation of the mass flow rate across each interface associated with this element.

To elaborate further, consider node (i,j) and its subcontrol-volume in Fig. 2.4, which has two interfaces 1 and 4. The mass flow rates through these interfaces are:

$$\dot{m}_1 = \frac{\rho_{i,j} u_a \delta y_j}{2} \qquad \dot{m}_4 = \frac{\rho_{i,j} v_d \delta x_i}{2} \tag{3.28}$$

The total mass flow rate out of this subcontrol volume across faces 1 and 4 is:

$$(\dot{m}_1 + \dot{m}_4) = +\frac{\rho_{i,j} u_a \delta y_j}{2} + \frac{\rho_{i,j} v_d \delta x_i}{2}$$

Substitute for u_a and v_d from Eqs. 3.26-27. Then :

$$(\dot{m}_1 + \dot{m}_4) = \frac{\rho_{i,j}\delta y_j}{2}[\hat{u}_a + \frac{d^u_a(p_{i,j} - p_{i+1,j})}{\delta x_i})] + \frac{\rho_{i,j}\delta x_i}{2}[\hat{v}_d + \frac{d^v_d(p_{i,j} - p_{i,j+1})}{\delta y_j})] \tag{3.29}$$

Therefore, the mass conservation equation for the CV. surrounding node (i,j) is:

$$\{\rho_{i,j}[\hat{u}_a + \frac{d^u_a(p_{i,j} - p_{i+1,j})}{\delta x_i})]\frac{\delta y_j}{2} + \rho_{i,j}[\hat{v}_d + \frac{d^v_d(p_{i,j} - p_{i,j+1})}{\delta y_j})]\frac{\delta x_i}{2}\} \tag{3.30}$$

$+$ [similar contributions from other elements associated with node (i,j)]

$+$ [boundary contributions, if applicable] $= 0$

The contributions of element (i,j) to the coefficients of the discretized pressure equation for node (i,j) are:

$$a^p_{E_{i,j}} = \frac{\rho_{i,j}\delta y_j d^u_a}{2\delta x_i}$$

$$
\begin{aligned}
a^p_{N_{i,j}} &= \frac{\rho_{i,j}\delta x_i a^v_d}{2\delta y_j} \\
a^p_{P_{i,j}} &= a^p_{E_{i,j}} + a^p_{N_{i,j}} \\
b^p_{i,j} &= -\frac{\rho_{i,j}\delta y_j \hat{u}_a + \rho_{i,j}\delta x_i \hat{v}_d}{2}
\end{aligned} \tag{3.31}
$$

In the same way, the contributions of other elements to the coefficients of the discretized pressure equation for node (i,j) are calculated and assembled. The final form of the discretized pressure equation for node (i,j) can be written as:

$$
\begin{aligned}
a^p_{P_{i,j}} p_{i,j} &= a^p_{E_{i,j}} p_{i+1,j} + a^p_{N_{i,j}} p_{i,j+1} \\
&\quad + a^p_{W_{i,j}} p_{i-1,j} + a^p_{S_{i,j}} p_{i,j-1} + b^p_{i,j}
\end{aligned} \tag{3.32}
$$

or

$$
a^p_{P_{i,j}} p_{i,j} = \sum_{nb} a^p_{nb_{i,j}} p_{nb_{i,j}} + b^p_{i,j} \tag{3.33}
$$

The discretized pressure equations for other nodes are assembled in a similar way.

3.8 Implementation of Boundary Conditions

In fluid flow problems, either the velocity or pressure are specified at boundary nodes, or shear stresses are given at boundary surfaces. Also, at outflow boundary surfaces, neither velocity, nor the shear stresses are specified. The implementation of boundary conditions will be discussed separately for momentum and pressure equations.

3.8.1 Implementation of Boundary Conditions in Momentum Equations

The procedures used to implement boundary conditions in the x- and y-momentum equations are the same, and they are similar to the implementation of boundary con-

ditions for convection-diffusion problems as discussed in section 2.5.6. The only difference is in the additional treatment of the pseudo-velocities and pressure coefficients at specified-velocity boundary nodes. At such nodes, the following treatment is used: $\hat{u} = u$; $\hat{v} = v$; $d^u = 0$; and $d^v = 0$.

3.8.2 Implementation of Boundary Conditions in Pressure Equations

The boundary conditions for pressure (continuity) equations are either specified-pressure, specified-mass flow, or outflow boundary conditions. The treatment of each of these cases is discussed next.

Specified-Pressure Boundary Condition

In the case of specified-pressure boundary condition, the discretized equations for the boundary nodes are overwritten to:

$$p_{boundary} = p_{specified} \tag{3.34}$$

Hence, the coefficients in the discretized equations for these boundary nodes are:

$$\begin{aligned} a_P^p &= 1 \\ a_{nb}^p &= 0 \\ b^p &= p_{specified} \end{aligned} \tag{3.35}$$

Specified-Mass Flux Boundary Condition

In this case, the mass flux at the boundary node is known. If the mass flux (ρV_n) is specified at the nodes on the boundaries, the corresponding mass flow rate is approximated as $\dot{m} = \rho V_n A_{c.v}$. Here, $A_{c.v}$ is the boundary surface area associated with the boundary node under consideration, and V_n is the component of velocity normal

to $A_{c.v}$. Otherwise, if the distribution of mass flow rate along the boundary is given, it is first integrated over each of the relevant boundary surfaces, $A_{c.v}$, to calculate the boundary mass flow rate, $\dot{m}$, for the corresponding boundary node. This mass flow rate is subtracted or added, as appropriate, to the term b^p, in the discretized pressure equation.

Outflow Boundary Condition

In the outflow boundary condition, neither mass flux nor the velocities are specified. In the case that pressure is also not specified, the discretized pressure equations for the boundary nodes are completed by estimating the mass flux out of the domain, based on the most current calculated velocities, at outflow boundaries. In every iteration, these velocities are updated, and, therefore, the outflow boundary conditions are updated, until the final converged solution is obtained.

An attractive feature of the proposed co-located FVM can be discussed at this stage. In the derivation of the discretized pressure equations in the interior of the domain, the mass flow rate across any particular control volume surface is based on linear interpolation of the pseudo-velocities ($\hat{u}$, $\hat{v}$), pressure coefficients (d^u *or* d^v), and pressure values at the two nodes on either side of the surface, as described previously. The values of d^u *or* d^v are *not* set to zero at the outflow boundary nodes. Consequently, in the calculation of mass flow rate across an *outflow boundary node*, the value of velocity obtained using the special mass-flux interpolations, given by Eqs. 3.26 and 3.27, is the same as that given by the discretized momentum equations, Eqs. 3.19 and 3.20; the pressure gradient in both these sets of equations corresponds to the average pressure gradient acting on the control volume associated with the boundary node in question.

In the CVFEMs of Prakash and Patankar [86] and Saabas [100], the element pressure gradient used in the calculation of velocity components in the mass flux

terms is not necessarily the same as the control-volume-averaged pressure gradient associated with the nodes, except in the limit of very fine grids, and this inconsistency also applies at outflow boundary nodes. In this work, there is only one pressure gradient which drives flow in the either x-direction outflow ($\frac{\partial p}{\partial x}$) or the y-direction outflow ($\frac{\partial p}{\partial y}$). The value of this pressure gradient is the same in both the boundary control volume surrounding the node in question and the corresponding boundary element. Therefore, there is no inconsistency in the treatment of pressure gradient at the outflow boundary in this work.

3.9 Solution of the Discretized Equations

The discretized continuity and momentum equations form sets of coupled non-linear algebraic equations. Various available methods for solution of these equations have been discussed in section 1.3.2. In this work, the iterative, sequential solution algorithm of Saabas [100] is used to solve these equations. Some enhancements to this algorithm will be proposed in the next chapter.

The nonlinearities in the discretized equations are resolved by Picard iteration: the coefficients in these equations are evaluated using the most recent field values. In each iteration, the set of linearized discretization equations are *not* solved completely to convergence. Rather, the coefficients are updated with the partially converged values of the dependent variables, to start a new iteration. These overall iterations are repeated until specified convergence criteria are met fully.

3.9.1 Summary of the Sequential Solution Algorithm

The solution algorithm proceeds in the following steps:

1. Guess the nodal values of pressure, velocity components, and other dependent variables which are coupled to the fluid flow. Set $\hat{u} = u$; $\hat{v} = v$; $d^u = 0$; and $d^v = 0$ at all nodes.

2. Calculate the coefficients of discretized x-momentum equations (Eq. 3.17), *excluding* any boundary conditions and the contributions of the pressure gradient terms. Incorporate *only* the given-flux and outflow boundary treatments, if applicable.

3. Calculate the pseudo-velocity $\hat{u}$ and also d^u for each discretized x-momentum equation, using Eqs. 3.23 and 3.24. If the u velocities are given at the boundary nodes, the $\hat{u}$ and d^u for these nodes are overwritten to:

$$\hat{u} = u_{specified} \qquad\qquad d^u = 0$$

4. Repeat steps 2 to 3 for the discretized y-momentum equations.

5. Calculate the coefficients in the discretized pressure equation (Eq. 3.33), apply the appropriate boundary conditions, and solve the set of algebraic equations using line-by-line TDMA method [77].

6. Add up the contributions of pressure-gradient terms to the constant terms in the discretized x-momentum equation, and under-relax the equations if required.

7. Apply specified-velocity boundary conditions for x-momentum equations, if applicable, and solve these set of discretized equations, using line-by-line TDMA (PDMA for QUICK scheme).

8. Repeat steps 6 and 7 for the discretized y-momentum equations.

9. Calculate the coefficients in the discretized equations for other dependent variables that are coupled to the fluid flow. Apply the appropriate boundary con-

ditions, under-relax, and solve them. This procedure is done sequentially for each of such variables.

10. Return to step 2 if convergence is not reached yet, and repeat steps 2 to 9 until overall convergence is achieved.

11. Solve for other dependent variables that do not affect the velocity field.

3.9.2 Under-relaxation of the Discretized Equations

Since the governing equations for fluid flow are nonlinear, the set of discretized equations are solved iteratively. If during the iterations, the changes in one of the variables becomes large, the solution may begin to oscillate, and there is a risk of divergence of the whole procedure. Therefore, it is necessary to under-relax the results in each iteration to ensure convergence of the solution procedure.

In this work, an implicit under-relaxation scheme [77] is used in the solution of the discretized momentum equations. It should be noted, however, that to avoid the dependency of the results on the under-relaxation factors, under-relaxation of these equations is done only *after* the calculations of $\hat{u}$, $\hat{v}$, d^u, and d^v [71] are completed. The amount of the under-relaxation required is a problem dependent parameter [77]. Some recommendations will be made in chapters 5 and 6.

3.9.3 Solution of the Linear Discretized Equations

The same methods as those used for the solution of linearized discretization equations in convection-diffusion problems, as described in section 2-6, are also used here: Line-by-line TDMA [77] (PDMA for QUICK scheme) is utilized for the sequential solution of the set of linearized discretization equations for u, v, and P.

Chapter 4

SEVA AND ESSA SCHEMES

4.1 Introduction

As was stated in chapter 1, Saabas [100], in his doctoral dissertation, identified the causes and proposed solutions to some serious difficulties which affected the earlier CVFEMs of Baliga and Patankar [7, 8], Prakash and Patankar [86], Prakash [87], Hookey and Baliga [49], and LeDain-Muir and Baliga [62]. In particular, a MAss-Weighted skew upwind scheme for triangular and tetrahedral elements was proposed and implemented: this ensured that algebraic approximations to convective transport terms contributed positively to the coefficients in the discretized equations, without incurring the excessive false diffusion inherent in the donor-cell upwind scheme of Prakash [88]. Furthermore, Saabas [100] showed that the interpolation functions that Prakash [87] and Hookey and Baliga [49] used to interpolate the velocity components in the discretization of mass flux terms are unsuitable for problems with inflow and outflow boundaries, and he advocated a return to an earlier proposal by Prakash and Patankar [86]. The resulting nonlinear coupled sets of discretization equations for the velocity components and pressure were solved using an iterative sequential solution algorithm. However, Saabas did not optimize this solution algorithm, in terms of

the number of iterations and the CPU time required to achieve converged solutions Another drawback of the CVFEM of Saabas [100] is its relatively large computer-storage requirements, compared to earlier CVFEMs [87]. One of the objectives of this thesis is to enhance the iterative sequential solution algorithm proposed by Saabas [100], in term of both the required computer time and storage.

The discretized momentum and continuity equations form coupled sets of nonlinear algebraic equations. The solution to these nonlinear equations is obtained iteratively, by successively solving coupled sets of linearized equations. These linearized equation sets are typically obtained using either a successive substitution (Picard) method or the Newton-Raphson method. The Newton-Raphson method is characterized by quadratic convergence, when the initial, or guess, values of the unknowns lie within a certain radius of convergence (or not too far from the exact solution) [68, 34]. The rate of convergence of the Picard method is only linear, but it is more robust than the Newton-Raphson method [31]. A combination of the Picard method (to start the solution) and the Newton-Raphson procedure (to finish the solution) appears desirable [31, 69, 84], but it is outside the scope of this thesis. In this work, attention is limited to iterative solutions based on the Picard method.

One approach to the solution of the linearized, but coupled, sets of discretized momentum and continuity equations is to use the Numerical Direct Solvers (NDS) [68]. Such methods, however, require very large (perhaps excessively large) computer storage for solution of fluid flow problems, as was discussed in 1.3.3. For this reason, NDS are not used in this work.

Segregated solution algorithms solve the linearized, coupled, sets of equations sequentially. Examples of such methods include the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) [77], SIMPLE-Revised (SIMPLER) [77], SIMPLE-Consistent (SIMPLEC) [122], and related versions. These methods have been used

successfully for over two decades in the solution of incompressible, laminar and turbulent, fluid flow problems. However, there are some drawbacks to these methods, especially their slow convergence rate and weak interconnection between dependent variables during the solution procedure.

In Iterative Coupled Equations Solvers, the coupled sets of linearized discretization equations are solved simultaneously, block-by-block, line-by-line, or node-by-node, in each iteration. Examples include procedures in [12, 20, 32] . These methods enjoy some of the advantages of NDS methods, namely strong interconnections between variables and fast convergence, but they also suffer some of the disadvantages of NDS methods, particulary the large storage requirements. Furthermore, implementations of iterative coupled-equation solvers and generalization of these methods to handle the wide range of practical incompressible fluid flows are not as simple as that of segregated methods.

4.2 The Task

The set of algebraic discretization equations for steady, two-dimensional, incompressible fluid flow were derived in chapter 3. The task here is to solve these sets of discretization equations, which are repeated here for convenience,

$$a^u_{P_{i,j}} u_{i,j} = \sum_{nb} a^u_{nb_{i,j}} u_{nb_{i,j}} + b^u_{i,j} \qquad 3.17$$

$$a^v_{P_{i,j}} v_{i,j} = \sum_{nb} a^v_{nb_{i,j}} v_{nb_{i,j}} + b^v_{i,j} \qquad 3.18$$

$$a^p_{P_{i,j}} p_{i,j} = \sum_{nb} a^p_{nb_{i,j}} p_{nb_{i,j}} + b^p_{i,j} \qquad 3.33$$

These equations are nonlinear and coupled. Therefore, their solution requires proper treatment of the nonlinearity and interconnection between dependent variables. The available methods for solution of these equations were discussed in sections 1.3.3 and 4.1.

In the next section of this chapter, some segregated and coupled equation solvers will be discussed. Following that, the iterative sequential solution algorithm of Saabas [100] will be discussed in terms of its computer memory and time requirements. Then, an improved version of the Saabas scheme, namely, SEquential Variable Adjustment (SEVA) scheme, will be proposed. Finally, a scheme labelled as Enhanced Sequential Solution Algorithm (ESSA) will be put forward in the last section.

4.3 Discussion of Available Iterative Methods

In this section, two categories of iterative methods, namely, segregated methods related to SIMPLE [77] and coupled-equation-line-solver methods (CELS) [32, 50] will be discussed. As was discussed in section 1.3.3, the CELS could also be regarded as semi-direct methods. The objective of this discussion is to identify the strong and weak points of these methods.

4.3.1 Segregated Methods Related to SIMPLE

The SIMPLE algorithm and its variants solve the discretized momentum, continuity, and other governing equations sequentially, and repeat this procedure until convergence is achieved. The SIMPLE algorithm consists of the following steps [77]:

1. Guess the pressure field.
2. Based on the guess pressure, solve the momentum equations to obtain velocity components.
3. Solve a pressure-correction equation.
4. Calculate pressure p, by adding the pressure corrections to the guessed pressure.

5. Obtain new values of u, v, and w from their values in step 2, using velocity-correction equations.

6. Solve the discretization equations for other dependent variables, if they influence the flow field.

7. Treat the corrected pressure p as a new guessed pressure, return to step 2, and repeat the whole procedure until a converged situation is obtained.

This semi-implicit algorithm, has the following desirable features:

1. It is easy to implement.

2. It has been successfully applied to a wide range of fluid flow problems, including turbulent flows, natural convection flows, and flow in porous media, and flows with combustion.

3. Since the set of discretized equations are solved sequentially, in each step, only the coefficients of the current set of discretized equation are required to be calculated and stored. Therefore, the same variable names and computer memory can be used to store the coefficients of each of the set of discretized equations, anytime they are required.

4. Since in each step, only one linearized and decoupled set of discretized equations is to be solved, efficient iterative solution methods, based on block-by-block, line-by-line, or point-by-point Gauss-Seidel methods, with multigrid or block-correction techniques to enhance convergence [106], can be used with relatively low computer-storage requirements.

This algorithm also has important drawbacks, however, which makes its utilization unfavourable in many cases. The main shortcomings of this algorithm are as follows:

1. In the velocity-correction equations, the effect of neighbouring velocity corrections are omitted. Hence, the pressure-correction terms in velocity-correction equations have to compensate for this omission. This leads to rather exaggerated pressure corrections and hence under-relaxation of the pressure correction equations becomes essential [77]. This, in turn, causes the convergence of this algorithm to be relatively slow.

2. Since successive substitution is used for treatment of the nonlinear terms in the discretized equations of dependent variables, and these equations are solved separately for each of the dependent variables, the interconnection between dependent variables is weak in this solution algorithm during the overall iterations.

3. At the start of this algorithm, the coefficients of the discretized equations are calculated based on the guessed values, and in each iteration, they are recalculated based on the most recent values of dependent variables. Recalculations of these approximate coefficients in each step of this sequential iterative procedure could be unnecessarily expensive in term of computer time, and it could even slow down convergence.

In the SIMPLER algorithm [77], a separate equation is developed for the calculation of the pressure and there is no approximation in deriving this equation, so no under-relaxation is necessary for the pressure equation [77]. SIMPLEC [122] improves the SIMPLE algorithm by a better approximation of the neighbouring velocity corrections in the derivation of a simplified velocity-correction equation. In this algorithm, in the velocity correction equations, it is assumed that the correction to the neighbouring nodal velocities in the equation is equal to the correction to the nodal velocity of interest. Both SIMPLER and SIMPLEC converge faster than SIMPLE [77, 122], because of improvements in the calculation of the pressure field.

The convergence of SIMPLER and SIMPLEC still remains slow, especially when the solution is close to satisfying the convergence criteria. This is because even in these algorithms, the sets of discretization equations are solved in a sequential manner, decoupled from each other by successive substitutions of the dependent variables.

On the positive side, these algorithms can handle almost all types of grids, such as staggered grids or co-located grids in both rectilinear and curvilinear coordinate systems. Examples may be found in Refs. [77, 81, 93, 1, 117, 22].

4.3.2 Coupled-Equation Line Solvers

Two coupled-equation line solvers (CELS), one proposed by Galpin [32] in the context of a staggered-grid finite volume method, and the other proposed by Hookey [50] in the context of an equal-order co-located CVFEM, will be discussed here. These discussions are intended to facilitate the incorporation of some of the CELS concepts into the iterative sequential solution method discussed in chapter 3.

The CELS method of Galpin, is applicable to a line-by-line structured-grid FVM for a steady, two-dimensional, incompressible fluid flow. In this procedure, the unknowns, u, v, and p, are coupled using an iterative line-by-line simultaneous solution of the discretized continuity, x-momentum, and y-momentum equations. When attention is focused on the values of u, v, and p at the nodes along any particular grid line, the most recent estimations of velocities and pressure at nodes off that line are assumed to be specified (or "known"). The coupled discretized continuity, x-momentum, and y-momentum equations associated with the nodes along the line of interest are solved simultaneously as follows: the continuity equation is used to eliminate the v-velocity terms from the y-momentum equation; the resulting equation is rearranged so as to express pressure in terms of u; and this equation is used to eliminate the pressure terms from the x-momentum equation. The final result is a

linearized set of simultaneous equations that has a penta-diagonal coefficient matrix and only one unknown, u. This set of equations is solved, and then the v and p fields are easily calculated by substituting the computed u values in the aforementioned rearrangements of the y-momentum and continuity equations. This procedure is repeated line-by-line until the entire calculation domain is swept. Typically, these line-by-line sweeps are carried out in alternating directions: positive x, positive y, negative x, and negative y; and these are then repeated. For each iteration, all coefficients in the discretized continuity, x-momentum, and y-momentum equations are calculated at first, and kept constant until one complete iteration is accomplished. Then these coefficients are updated using the new calculated values of the dependent variables.

The main features of the CELS of Galpin [32] are as follows:

1. During the iterations, the interconnection between dependent variables, u, v, and p are fully respected in CELS, in contrast to rather weak coupling in the earlier SIMPLE procedures [77].

2. During the iterations, along each grid line, discretized x- and y-momentum and continuity equations are solved simultaneously and exactly. Thus at every stage of the iterations, the u and v fields satisfy mass conservation exactly.

3. Since no additional approximations are made to the u, v, and p discretized equations, the solution of these equations is less sensitive to the under-relaxation parameters used than those in the segregated methods [32].

This procedure also has important drawbacks as follows:

1. Since in each iteration, all the coefficients in discretized equations must be calculated and stored, it requires a considerable amount of computer storage. This difficulty could get particularly unmanageable in three-dimensional problems.

2. Another main shortcoming of this method [32] is that it has been derived in the context of two-dimensional incompressible fluid flow problems, in which the unknowns dependent variables are u, v, and p. The extension of this procedure to three-dimensional problems gets quite tedious, and it can not be easily generalized to solve fluid flow problems that involve additional dependent variables, such as temperature, kinetic energy of turbulence, and concentration of chemical spices. For example, to apply CELS to two-dimensional natural convection in a square enclosure, Galpin [31] had to device a special extension that allowed a simultaneous solution of u, v, p, and T discretization equations.

CELS of Hookey: A coupled-equation line solver is also used in the equal-order co-located CVFEM of Hookey [50] for the solution of two-dimensional, viscous, compressible fluid flow problems. In this CELS, Hookey rearranged the discretized equations in such a way that the value of any dependent variable at each node along a line of interest is only dependent on the values of the dependent variables at the nodes on one side of the node (denoted as "forward " nodes, for convenience). The coefficients of these rearranged equations are calculated based on the values of the coefficients in the discretized equations. Since for the last node on a line, there is no forward node, the values of dependent variables at the last node are obtained directly from these rearranged equations. The values of the dependent variables at the nodes behind the last node of that line are then computed by a simple back-substitution procedure. Because of similarity between this method and the TDMA [77], Hookey called his procedure as coupled-TDMA (CTDMA) [50].

To use this CELS, in each iteration, all coefficients in the discretized equations must be calculated and stored, based on the guessed or current values of the dependent variables. In the line-by-line application of this CELS, alternating-direction sweeps, similar to those used in the CELS of Galpin [34] are used. As the governing

equations are nonlinear, after each iteration, the overall convergence is checked, and if convergence has not been achieved, the procedure is repeated with newly calculated coefficients based on the latest available values of the dependent variables.

The CELS of Hookey shares almost all the advantages and disadvantages of Galpin's CELS. It should also be noted at this stage that in the CVFEMs of Hookey [47] and Hookey and Baliga [48], the velocity components in the momentum and continuity equations are interpolated using an element-based functions that explicitly depend on the pressure gradient in the element. This approach for velocity interpolation was investigated by Saabas [100] in the context of incompressible flow, and it was abandoned because of its drawbacks in simulating outflow problems.

4.4 Sequential Solution Algorithm of Saabas

The various steps in this solution algorithm were presented in section 3.9.1. Much of the discussion pertaining to segregated methods akin to SIMPLE (see section 4.3.1) is also applicable to the solution algorithm of Saabas [100]. Nevertheless, more discussion is required to elaborate the reasons for the large computer memory and time requirements of this method. In each iteration of this method, the coefficients in the discretized momentum equations are used in calculations of the pseudo-velocities $(\hat{u}, \hat{v})$ and pressure coefficients (d^u, d^v), which, in turn, are used in calculations of the coefficients in the discretized pressure equation. After the solution of the discretized pressure equations, the coefficients in the discretized momentum equations are required again for the solution of the velocity field. Hence, the coefficients in the discretized momentum equations must be stored separately, and the coefficients in the discretized pressure equations have also to be declared (stored) in separate arrays. In other words, *all* coefficients have to be stored in the Saabas scheme. This leads to a very large computer-storage demand, or time-consuming reading and writing of a

huge amount of data to and from disks has to be done. This difficulty is particularly serious in three-dimensional problems.

As was pointed out in section 3.9.1, in the Saabas scheme, for each dependent variable, the iterative line-by-line TDMA solution of linearized discretization equations is repeated only for a few sets of four alternating-directions sweeps in each overall iteration: in other words, the line-by-line TDMA solution is not pushed to complete convergence, because of the approximate nature of the coefficients in the linearized discretization equations. Furthermore, based on available experience, the number of these line-by-line TDMA sweeps was set to a fixed value for each dependent variable.

Thus, this method does not necessarily lead to an optimized solution of fluid flow problems [122]. If the line-by-line TDMA sweeps for the discretized pressure equations are terminated before sufficient convergence is achieved, the continuity constraint is poorly satisfied by the pseudo-velocities and pressure coefficients. These values are later used to calculate new coefficients in the momentum equations, so that the error in the satisfaction of the continuity constraint is propagated, with the possible result of divergence or slow convergence. On the other hand, it is uneconomical to drive the solution of the discretized pressure equation to a tight convergence in each overall iteration. The performance of the entire solution algorithm depends heavily on the criterion used for terminating the line-by-line TDMA sweeps in the solution of the dependent, linearized, discretization equations. In this work, efforts were made to overcome these difficulties in the solution method of Saabas [100]. The resulting improvements are discussed in the following sections.

4.5 SEquential Variable Adjustment (SEVA) Algorithm

Details of the SEVA algorithm, which is an enhancement of the Saabas scheme, will be explained in this section. This enhancement concern two issues: (i) the computer time; and (ii) the computer memory required.

To reduce the computer time requirement, two approaches were examined. In the *first approach*, some of the proposals of Van Doormaal and Raithby [122], plus additional modifications, are incorporated. This main idea in [122] proposal is to quit from the line-by line TDMA sweeps [1] in the solver for the set of discretized pressure-correction equations whenever the sum of the absolute values of the residue of all the equations in this set reaches one fifth of its initial value, at the beginning of the solution of this set of equations in the current iteration. In this work, this idea is extended to the solution of all variables. Also, as the solution nears convergence, the initial residue has a small value, and many sweeps are required to achieve the aforementioned condition on residue reduction in each iteration. It is suggested, therefore, that if the sum of the absolute values of the residues becomes less than a small value (10^{-12}), the solution is converged, and it is not necessary to continue the sweeps in the solver. Therefore, if either the first condition (overall residue reduction) is satisfied, or if the sum of the absolute values of the residues becomes less than 10^{-12}, the sweeps in the solver are stopped, and the next step is started. In the *second approach*, for the set of discretized u, v, and p equations, only one sweep is performed in the solver. These approaches were also examined in the ESSA scheme, which will be explained in the next section.

To reduce the computer memory requirements in the Saabas scheme, it is necessary

[1] The line-by-line TDMA sweeps in the solver for the discretized equations, will be referred to as sweeps from now on, for simplicity.

to make some changes in the sequence of this solution algorithm. As was discussed in chapter 3, in the Saabas scheme, the solution of the discretized velocity equations is done after the solution of the discretized pressure equations. The coefficients of the discretized velocity equations are used in the calculations of $\hat{u}$, $\hat{v}$, d^u, and d^v, and the coefficients of the discretized pressure equations are calculated based on these values. Therefore, all the coefficients in the discretized momentum equations have to be stored in separate memories. The same memory can not be used repeatedly to keep all the coefficients.

It is suggested here that the pressure equation should be solved after the solution of the velocity equations, in order to reduce the memory requirements. Based on the above suggestion and without making any other significant changes to the Saabas scheme, the SEquential Variable Adjustment (SEVA) algorithm, proceeds in the following steps.

1. Guess the pressure, velocity components, and other dependent variables which are coupled to the fluid flow. Set $\hat{u} = u$; $\hat{v} = v$; $d^u = 0$; and $d^v = 0$.

2. Calculate coefficients in the discretized u-velocity equations (Eq. 3.17), excluding contributions of the pressure gradient terms. Dump the values of coefficients of the discretized equations at boundary nodes, and also a_P terms for all nodes, in separate storage for later use in the calculation of coefficients in the discretized v-velocity equations.

3. Apply any specified-flux boundary conditions, if necessary, dump the values of a_P and constant terms in separate storage, for later use in calculations of $\hat{u}$ and d^u.

4. Add up the contributions of pressure-gradient terms to the current values of the constant terms in the discretized u-velocity equations, and under-relax the

equations, if required.

5. Apply Dirichlet boundary conditions for u-velocity equations, if needed, and solve the set of these equations, using the line-by-line TDMA method with either of the two enhancements explained at the beginning of this section.

6. Read in the values of a_P and constant terms stored in step 3. Calculate the pseudo-velocity $\hat{u}$ and also d^u. If the u velocities are given at the boundary nodes, the $\hat{u}$ and d^u for boundary nodes are overwritten to:

$$\hat{u} = u_{spec} \qquad\qquad d^u = 0$$

7. Read in the value of a_P, constant terms, and coefficients of the discretized equations at boundary nodes from step 2, and repeat steps 3-6 for the discretized v-velocity equations.

8. Calculate the coefficients in the discretized pressure equation (Eq. 3.33), apply appropriate boundary conditions, and solve this set of equations using line-by-line TDMA method, with the two enhancements explained at the beginning of this section.

9. Calculate the coefficients in the discretized equations for other dependent variables that are coupled to the fluid flow. Apply appropriate boundary conditions, under-relax, and solve these equations. This procedure is repeated sequentially for such variables.

10. Return to step 2 if the convergence criteria is not satisfied yet, and repeat steps 2-9 until overall convergence is achieved.

11. Solve for the other dependent variables that do not affect the velocity field.

In this method, some clever storage of the coefficients is needed to minimize the memory requirements, while keeping the running time of the algorithm essentially unchanged. Only the a_P and constant terms for each of discretized u- and v-velocity equation are required to be stored, and the same memory is used by the coefficients in the discretization equations for all the dependent variables at internal nodes. This leads to a considerable reduction in the computer memory requirements compared to the Saabas scheme [100].

It should also be noted that the solution of the pressure equations after the velocity equations does not violate an important feature of the Saabas method: continuity requirements are well satisfied at the start of each overall iteration. Only for the very first iteration, with guessed u, v, and p fields, this condition may not be satisfied.

4.6 Enhanced Sequential Solution Algorithms (ESSA)

4.6.1 Motivation

In section 4-3, two procedures for the solution of fluid flow problems were discussed. The main difficulty with available segregated methods like SIMPLE [77] are their slow convergence, especially in the latter steps of the iterative solution. On the other hand, these methods have desirable features, such as relatively modest memory and storage requirements, compared to coupled-equation solvers. Another important advantage of algorithms akin to SIMPLE is their generality: they can be used without any modifications to solve a wide range of complex fluid flow problems [77], such as turbulent flows. This facilitates the development of general computer codes.

Coupled-equation solvers, on the other hand, have the advantage of faster convergence [31, 68] , and this results in savings of computer time, but they require

more computer memory than segregated solution algorithms. Also, coupled-equation solvers can not be easily extended to solve fluid flow problems in which more dependent variables than just velocity components and pressure are involved; and they require special treatments (major modifications) for each additional dependent variable. For example, in the problem of natural convection in a square enclosure, the processed u-velocity discretized equations, which have a penta-diagonal band in the coefficient matrix after substitutions to get rid of v and p values, and the coupled discretized temperature equations,which have a tri-diagonal band in the coefficient matrix, must be solved together [35]. If all coupled variables are to be included in the coupled-equation solution algorithm, it would create a huge computational molecule, since each coupled neighbour has to be kept explicitly in each discretized equation. This could create storage requirements similar to those needed in numerical direct solvers, which were abandoned in this work due to their excessive demands for computer memory.

It is suggested in this work, that a combination of ideas borrowed from sequential solution algorithms, akin to the SIMPLE, and coupled-equation solvers, akin to the CELS, keeping their advantages and omitting, if possible, their drawbacks, can improve the efficiency of the solution of the sets of discretization equations. Adoption of the key ideas of the CELS method, similar to that of Galpin [32] or Hookey [50], in conjunction with a sequential algorithm, was the first approach that was examined.

The CELS of Galpin [32] is formulated in the context of staggered grids, and velocities at the interfaces of pressure control volumes, which are required for assembling the discretized continuity equations, are the nodal velocities, therefore no interpolation is required to find these velocities. Hence, in the discretized continuity and momentum equations, which later on are manipulated to form a u-velocity equation, the velocity at each node is connected only to its immediate neighbouring velocities.

Thus, the coefficient matrix for each line of nodes in the final set of u-velocity equations is penta-diagonal, and a PDMA can be used to solve these equations efficiently.

In the work of Hookey [47], the same interpolation function, are used for velocity in the approximation of both the continuity and momentum equations, and the discretized equations involve velocities only at immediate neighbours. Therefore, a relatively small molecule for each node is created in the discretized u- and v-equations, and this can be handled by methods like the CTDMA [47].

The Hookey approach for the interpolation of velocity was abandoned by Saabas [100] because of drawbacks that were discussed earlier in this thesis. In work of Saabas [100], the velocity components in the mass flux terms are expressed as in Eqs. 3.21 - 3.24, which are repeated here for convenience:

$$u_{i,j} = \hat{u}_{i,j} - d^u_{i,j}\overline{\frac{\delta p}{\delta x}} \qquad 3.21$$

$$v_{i,j} = \hat{v}_{i,j} - d^v_{i,j}\overline{\frac{\delta p}{\delta y}} \qquad 3.22$$

where:

$$\hat{u}_{i,j} = \frac{\sum_{nb} a^u_{nb_{i,j}} u_{nb} + b^u_{i,j}}{a^u_{P_{i,j}}}$$

$$\hat{v}_{i,j} = \frac{\sum_{nb} a^v_{nb_{i,j}} v_{nb} + b^v_{i,j}}{a^v_{P_{i,j}}} \qquad 3.23$$

are pseudo-velocities and:

$$d^u_{i,j} = \frac{\Delta V}{a^u_{P_{i,j}}}$$

$$d^u_{i,j} = \frac{\Delta V}{a^v_{P_{i,j}}} \qquad 3.24$$

are pressure coefficients.

The pseudo-velocity at a node is dependent on the neighbouring velocities; these neighbour-velocities can also be expressed as a sum of a pseudo-velocity plus a pressure gradient term; but the pseudo-velocities at neighbouring nodes are also dependent on their neighbouring velocities; and so on. Thus, with this interpolation for velocities, it is impossible to use the CELS methods, in the form defined by either Galpin or Hookey.

Therefore, direct adoption of the ideas from the CELS method and incorporation of these into the segregated solution algorithms, as discussed in chapter 3, is impossible, or at best very complicated, and so it was abandoned. Instead, attention was focused on the indirect use of the CELS idea in the sequential solution algorithms.

In the CELS method, all discretized equations coefficients must be calculated and stored before each iteration. These values, which are kept constant in each iteration, are used to calculate the rearranged equations coefficients in each iteration. This is in contrast to the segregated methods in which, only an individual set of the discretized equations coefficients for a particular dependent variable, is calculated and used, immediately, for the solution of these equations, and this is repeated sequentially for each dependent variable. The segregated approach has advantages and disadvantages: its main advantage is that the same storage can be used for the coefficients in the discretized equations for all dependent variables; its disadvantage is that these sets of coefficients have to be calculated in each iteration, based on the latest calculated values of the dependent variables, and this requires a considerable amount of computer time. A better approach appears to be to keep the calculated coefficients constant for few iterations and then recalculate them.

Another point which must be taken into consideration in the solution procedures for fluid flow problems is the relatively faster convergence of the velocity fields com-

pared to the pressure field. Solution of the discretized pressure equations can consume up to 80 percentage of execution time for a fluid flow problem [122]. It would be useful, therefore, if in the solution procedure, more effort is directed to improve the rate of convergence of the pressure field, in such a way that solutions to both the velocity and pressure equations converges at the same rate.

The above discussions are the basis for the Enhanced Sequential Solution Algorithm (ESSA), which will be presented in the next subsection.

4.6.2 Enhanced Sequential Solution Algorithm (ESSA)

This new algorithm adds an *inner loop* to the segregated methods, such as SIMPLE: the u, v, and p fields are updated sequentially in each cycle in the inner loop, while the coefficients of their discretized equations are kept constant. In each cycle of the inner loop, only the contributions of the pressure and other dependent variables which affect the constant terms in u- and v-velocity discretized equations are updated. In the discretized pressure equations, the only change is in the constant terms, which require updating as the pseudo-velocities are recalculated with the newly estimated values of velocities.

In this work, this enhancement is incorporated into the sequential solution algorithm of Saabas [100]. For convenience, the primary iterative procedure of Saabas will be called the *outer loop* from now on.

The **ESSA algorithm** proceeds in the following steps:

1. Guess the pressure, velocity components, and other dependent variables (other than u, v and p) which are coupled to the fluid flow. Set $\hat{u} = u$; $\hat{v} = v$; $d^u = 0$; $d^v = 0$

2. Calculate the coefficients and constant terms in the discretized u-velocity equations (Eq. 3.17), **excluding** contributions of the pressure-gradient terms and

source terms that involve dependent variables other than u, and apply appropriate boundary conditions. Store these coefficients and constant terms in separate arrays.

3. Add the contributions of source terms that depend on Φ (other than u, v, and p) to the constant terms calculated in step 2. Calculate $\hat{u}$ and d^u using the latest available u velocities (Eqs. 3.21 to 3.24), and apply appropriate boundary conditions.

4. Conduct operations similar to steps 2 and 3 for the discretized v-velocity equations, $\hat{v}$, and d^v: The storage locations for the coefficients in discretized v equation are different to those for the discretized u equation.

5. Calculate coefficients in the discretized pressure equations (Eq. 3.33), again storing these in locations that are different from those used for the u and v discretized equations.

6. Apply appropriate boundary conditions to the discretized pressure equations, and solve these equations. In the proposed FVM, a line-by-line TDMA was used in this step.

7. Add contributions of pressure-gradient terms and source terms excluded in step 1 variables to the constant terms in the discretized u-velocity equations, under-relax, apply appropriate boundary conditions, and solve this set of equations. Again, in the proposed FVM, a line-by-line TDMA was used for this step.

8. Do operations similar to step 7 in the discretized v-velocity equations.

9. Calculate coefficients in the discretized equations for other dependent variables that are coupled to the fluid flow. Apply appropriate boundary conditions, under-relax, and solve them. This procedure is repeated sequentially for such

variables, with the same storage locations for the coefficients in the discretized equations.

10. Check to see if overall convergence criteria are satisfied: If yes, go to step 12; If not, go to the next step.

11. **Start the inner loop**

    (a) Repeat steps 7 and 8.

    (b) Repeat step 3 for u, and do a similar step for v.

    (c) Recalculate the constant terms in the discretized pressure equations, and solve these equations to obtain an updated p field.

    (d) Do tasks listed in step 9, above.

    (e) Check to see if convergence criteria for the inner loop are satisfied: If yes, go to step 2; If not, go to step (a) in the inner loop. Note that steps (a) to (d) of the inner loop are repeated only until the prescribed maximum number of inner-loop iterations or cycles (KNMAX) is reached.

12. Solve for the other dependent variables that do not affect the velocity field.

13. Call desired output routines, and then stop.

The best prescribed maximum number of iterations or cycles of the inner loop (KNMAX) is problem dependent, and should be optimized for each problem. This optimization can be performed on a coarse grid. It was found in this work that the optimum values of KNMAX obtained for coarse grids are essentially optimal for fine grids too, for the same problem. Numerous numerical tests were done with ESSA for two-dimensional fluid flow problems. A review of these tests show that the optimal value of KNMAX is usually between 3 and 5. A general recommendation is KNMAX = 4.

4.6.3 Discussion of ESSA

The proposed ESSA has the following desirable features:

1. It is as simple as segregated methods, and does not need any extra manipulations of coefficients as in the coupled-equations solvers. It can be used to enhance any of the available SIMPLE-like segregated algorithms.

2. Problems that involve dependent variables other than the velocity components and pressure can be easily accommodated: This is in contrast to coupled-equation solvers (CELS) which requires special manipulation of basic solution routines to handle dependent variables other than velocity components and pressure.

3. It can be easily implemented on staggered and co-located, orthogonal or non-orthogonal, grids in two- and three-dimensional problems. This is a very important advantage of ESSA over the available CELS.

4. The extra computer time required for recalculation of the constant terms of the discretized equations in the cycles of the inner loop, is considerably less than that required to recalculate all coefficients in the discretized equations.

5. In this scheme, it is not *necessary* to keep all the coefficients of the discretized equations in main memory (RAM), as is the case in coupled-equation solvers. The coefficients for any dependent variable can be written on to an external storage device, such as a hard disk, and they can be read back into RAM when they are required. This feature could be especially important for solutions on large domains with fine grids, or in three-dimensional problems, for which a huge amount of storage would be needed to keep all the coefficients in main memory (RAM)

6. Simple tests on coarse grids can be used to optimize the maximum number of cycles in the inner loop (KNMAX) for any problem. This is because as the solution proceeds towards convergence, the number of inner-loop cycles required for convergence becomes less than KNMAX. In this case, the cycles in the inner loop are stopped, and the execution shifts to the outer loop.

Considerable savings in computer time were achieved using ESSA for the solution of three bench-mark test problems. Details of these tests are presented in the next chapter.

Chapter 5

EVALUATION OF THE SEVA AND ESSA SCHEMES

5.1 Introduction

In this chapter, the proposed methods for the solution of the coupled, nonlinear, sets of algebraic discretization equations, *SEquential Variable Adjustments (SEVA)* and *Enhanced Sequential Solution Algorithm (ESSA)*, are evaluated. Quantitative appraisal of the performance (or degradation) provided by these methods is done by comparing the results against that of the Saabas scheme [100], which was discussed in chapter 4.

Three standard, well-documented, laminar-flow test problems were selected for this evaluation:

1. Lid-driven flow in a square cavity
2. Natural convection in a square enclosure
3. Flow over a backward-facing step

All tests were performed for the steady, two-dimensional, laminar flow of an incompressible Newtonian fluid in a Cartesian coordinate system. The computer programs used in these tests were in standard FORTRAN-77 and executed on a Logix personal computer (Intel 80486 CPU, 50 MHz), using a LAHEY F77-EM/32 compiler. Other computers, such as IBM ES-9000-M320 and Hewlett-Packard HP-APOLLO Series 700 Model 710, were also used in the development phase of this work, but not in the final tests of the proposed methods.

Appropriate convergence criteria were devised to ensure consistency and accuracy in the results. The iterative solution was stopped, when this criteria were satisfied. The execution time of each test was determined using the TIME subroutine available in the LAHEY F77-EM/32 compiler. No input and output operations were conducted in the section of the computer code that was used to determine the CPU time needed to achieve convergence.

The lid-driven square-cavity flow at high Reynolds number (400 and 1000) is a good test problem to evaluate the performance of the proposed method in the solution of complex recirculating flows. Natural convection in a square enclosure allows performance evaluation in the context of problems in which there is coupling between the fluid flow and heat transfer. Finally, the problem of flow over a backward-facing step is a good test of the me' ods with the proposed treatment of outflow boundary conditions.

5.2 Preliminary Tests

In this section, the consistency and implementation (computer code) of the FVM described in chapter 3 is investigated. Lid-driven flow in a square cavity is used as the test problem, and the effect of grid refinement on the accuracy of the results is studied. In all these tests, the Saabas scheme (see section 3.9) was used to solve the

discretization equations.

5.2.1 Lid-driven flow in a square cavity

Problem Statement and Formulation

In this problem, steady, two-dimensional, laminar recirculation of an incompressible Newtonian fluid contained in a square enclosure is considered. The motion of the fluid is driven by a sliding lid. A schematic of this problem is given in Fig. 5.1. The square enclosure of side L has its lower-left corner located at the origin of the Cartesian coordinate system (x,y). All walls are fixed, except for the lid, which moves in the positive x direction with a constant velocity u_w.

The equations which describe this problem are the x- and y-momentum, and continuity equations, Eqs. 3.1 to 3.3. The following nondimensional variables and parameters are used:

$$x^* = x/L; \qquad y^* = y/L; \qquad u^* = u/u_w; \qquad v^* = v/u_w$$

$$p^* = (p - p_{ref})/\rho u_w^2; \qquad Re_w = \rho u_w L/\mu$$

The resulting non-dimensional momentum and continuity equations are of the form:

x-momentum:

$$u^*\frac{\partial u^*}{\partial x^*} + v^*\frac{\partial u^*}{\partial y^*} = -\frac{\partial p^*}{\partial x^*} + \frac{1}{Re_w}\left(\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}}\right) \tag{5.1}$$

y-momentum:

$$u^*\frac{\partial v^*}{\partial x^*} + v^*\frac{\partial v^*}{\partial y^*} = -\frac{\partial p^*}{\partial y^*} + \frac{1}{Re_w}\left(\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}}\right) \tag{5.2}$$

and continuity:

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} = 0 \tag{5.3}$$

with the following boundary conditions:

$$v^* = 0 \qquad \text{on all walls}$$

$$u^* = \begin{cases} 1 \;\; at \;\; y^* = 1, & 0 < x^* < 1 \\ 0 \;\; elsewhere \end{cases} \tag{5.4}$$

The singularities at the corners of the lid are handled by setting the velocity to zero there. The only free parameter in this problem is the Reynolds number, Re_w, based on the lid velocity. As was mentioned before, lid-driven flow in a square cavity is a problem that is well-established and well-documented in the testing of computational methods for recirculating flows. It is the subject of several papers, such as those by Burggraf [19], Bozeman and Dalton [14], Ghia et al. [39] and Schreiber and Keller [58]. Ghia et al. [39] used a stream function-vorticity finite-difference formulation and a multigrid method with a 129 x 129 node grid to solve this problem. Their results for the velocity profile along the vertical midplane of the enclosure are used to evaluate the results obtained in this work.

Numerical Details

In their bench-mark solutions for this test problem , Ghia et al. [82-2] used a 129 x 129 node grid. In these preliminary tests, the accuracy of the proposed FVM was investigated by obtaining solutions on uniform grids of 11 x 11, 21 x 21, 41 x 41, and 81 x 81 nodes. For all grids, the resulting profile of u^* along the line $x^* = 0.5$ was plotted and compared with the results of the bench-mark solution. These tests were performed using the SUDS and MAW schemes.

As was mentioned earlier, the only free parameter in this problem is the Reynolds number. The results for $Re = 1000$ are presented for this preliminary test. The under-relaxation parameters, that were used in this test are:

$$\alpha_u = \alpha_v = 0.6; \qquad \alpha_p = 1.0$$

The overall iterations were terminated in these tests when the sums of the absolute values of residues for the in each of u, v, and p sets of discretized equations were all

less than 10^{-8}.

Results

The u-velocity profiles at $x^*= 0.5$, obtained with four different uniform grids, and the bench-mark solution of Ghia et al., are shown in the Fig. 5.2. As is seen in these figure, with refinement of the grid results obtained with both the SUDS and MAW schemes advance towards the bench-mark solution. In other words, with grid refinement, the errors in the results in comparison with the bench-mark solution decrease. These preliminary tests ensure that the proposed FVM has been consistently formulated and implemented.

5.3 Investigation of ESSA and SEVA

The three aforementioned test problems were used to investigate ESSA and SEVA in comparison with the Saabas scheme to establish the following results:

1. the CPU time required to achieve converged solutions;
2. the number of iterations needed to obtain converged solutions;
3. the effect of inner- and outer-loop under-relaxation values on Items 1 and 2; and
4. the effect of the specified maximum number of inner loop cycles (KNMAX) in the ESSA scheme on Items 1 and 2. [1]

All the tests in this section were performed utilizing the MAW scheme.

[1]Items 1 and 2 will be called 'execution time' and 'number of iterations' from now on for simplicity.

5.3.1 Lid-driven flow in a square cavity

The problem statement for this problem is the same as that described in the previous section. Numerous tests were performed with this problem for a comprehensive study of the Saabas, SEVA and ESSA schemes.

Numerical Details

Nonuniform grids of 31 x 31 nodes, with a power-law type expansion of the grid lines, with power of 1.4, away from the solid boundaries in both x- and y-directions were used.

With the Saabas and the SEVA schemes, tests were done with under-relaxation parameters for u- and v-velocity discretized equations from 0.5 to 0.8; this parameter for the discretized pressure equations was set equal to 1 for all cases. With the ESSA scheme, the tests were performed to evaluate:

1. influence of the value of the under-relaxation parameter in the discretized velocity equations in the *outer-loop*, in the range 0.8 to 1.0.

2. influence of the value of the under-relaxation parameter in the discretized velocity equations in the *inner-loop*, in the range 0.5 to 0.8; and

3. effect of KNMAX: values of 1 to 5, 10, 15, and 20 were tried.

Also, to optimize the line-by-line TDMA solver [2] for the decoupled and linearized sets of discretized equations, the suggestion of Van Doormaal and Raithby [122] was investigated, as described in chapter 4. To evaluate this approach, values of 0.1 to 0.9 were examined for the ratio of the sum of the absolute residues of the discretized equations at the end of each sweep in the solver to its initial value at the start of the

[2] From now on instead of *line-by-line TDMA solver*, for simplicity only *solver* will be used.

solution of the discretized equations. Finally, a set of tests was performed with only one sweep in the solver for the u, v, and p discretized equations.

The optimum values based on results obtained with the nonuniform grid of 31 x 31 nodes were then used in tests with nonuniform grids of 51 x 51 and 81 x 81 nodes, with a power-law expansion of the grid lines, with power of 1.4, away from the solid boundaries in both x- and y-directions.

All tests were performed at $Re_w = 1000$. The convergence criteria for these tests were to satisfy the following conditions:

1. the relative change in the value of the sum of the absolute values of the constant term in the discretized pressure equation should be less than 10^{-6}; and

2. the relative change in the value of the u-velocity component at the geometric centre of the cavity should be less than 10^{-5}, or its absolute value should be less than 10^{-12}.

In inner loop, condition number 1 should be satisfied to stop the cycles.

In each test, the execution time and number of iterations required to achieve solutions that satisfied these convergence criteria were recorded.

Results

The parameters which affect the convergence rate of the ESSA scheme are the values of the under-relaxation parameters for u- and v-velocity discretized equations in the inner and in the outer loops and the number of cycles in the inner loop.

In figures 5.3, 5.4 and 5.5, the execution time and number of iterations are plotted versus the maximum number of cycles in the inner loop (KNMAX). These plots are for under-relaxation values for the u- and v-velocity discretized equations from 0.5 to 0.7 in the inner loop [RELAX(V)]. The under-relaxation values for the outer loop (ALPHAO) are 0.8, 0.9, and 1.0, respectively, in figures 5.3, 5.4, and 5.5. As is seen in

these figures, for all values of ALPHAO, the execution time and number of iterations are decreased by increasing of the value of RELAX(V). The tests with RELAX(V) $\geq$ 0.8 failed to converged solution.

In figure 5.6, the execution time and number of iterations are plotted versus KNMAX, at constant RELAX(V) = 0.7, for ALPHAO = 0.8, 0.9, and 1. As is seen in this figure, the best results are achieved when no under-relaxation is done in the outer loop (ALPHAO = 1).

Hence, the values of 0.7 for the inner-loop and 1 for the outer-loop were found to be the optimum values of under-relaxation parameters for the solution of u- and v-velocity discretized equations. As is seen in these figures, values of KNMAX between 3 to 5 give the best results in term of execution time. By increasing the value of KNMAX from 1 to around 5, the number of iterations decreases sharply, but further increase in KNMAX leads to only a gradual decrease in the number of iterations. However, the execution time increases with increasing value of KNMAX beyond than its optimum value (3 to 5). From these results, it can be concluded that for this test problem, the optimum value for KNMAX lies between 3 to 5.

All of these tests were performed utilizing the recommendations in [122] for terminating the sweeps in the solver for the decoupled linear sets of discretized equations. The ratio of the normalized total absolute residue in each set of discretized equations at the end of each sweep in the solver to its value at the start of the sweeps (RES) was chosen to be 0.2 in all the aforementioned tests.

Efforts were also made to optimize the value of RES. In figure 5.7, the execution time and number of iterations are shown for the solutions obtained with values of RES ranging from 0.2 to 0.9, for two values of KNMAX = 3 and 5. As is seen, two different values of RES perform the best: for KNMAX = 3, this value is RES = 0.5; and for KNMAX = 5, this value is RES = 0.8. None of these values is even close

to the recommended value of RES = 0.25 proposed by Van Doormaal and Raithby [122].

On the basis of the research undertaken in this work, it is suggested that a fixed value for this ratio (RES) would not produce the optimum condition for exiting from the solver. In the early stages of the overall iterations, RES must be a small value: since the values of the dependent variables (u, v, and p) are far from their values in the final solution of the problem, more sweeps are required in the solver. However, as the solution proceeds and the dependent variables progress towards their final values, it is not necessary to keep this ratio fixed at the small value that is required in the early stages of the iterative solution. In other words, the best treatment appears to be one in which the value of RES is not fixed, but is allowed to change appropriately as the solution progress.

In another try to optimize the overall solution process in ESSA, only one sweep was performed in the solver of the linearized decoupled sets of discretized equations. The results obtained for RELAX(V) = 0.5, 0.6, and 0.7 are plotted for KNMAX values from 1 to 20 in figure 5.8. In table 5.1, ESSA execution times and number of iterations obtained with the optimized RES values , and with only one sweep in the line-by-line TDMA solver, are compared. The considerable superiority of the results obtained with only one sweep in the TDMA solver are significant. It is clear that with the 31 x 31 node nonuniform grid for this problem, performing one sweep in the TDMA solver for all dependent variables lead to the optimum performance of ESSA.

In figure 5.9, the execution times and number of iterations are plotted for RELAX(V) = 0.5, 0.6, and 0.7 for the Saabas, SEVA, and ESSA schemes. In the TDMA solver, only one sweep was performed for the three schemes. As is seen in this figure, the ESSA scheme produces convergence almost two times faster than the Saabas and SEVA schemes, and the number of iterations for the ESSA scheme is almost one-third

of the corresponding values for the other two schemes. This figure clearly shows the significant influence of the ESSA scheme in decreasing both the execution time and number of iterations in comparison to the Saabas scheme. The SEVA scheme significantly decreases the storage requirements of the Saabas scheme, as was discussed in chapter 4, and it is also lead to a slight improvement in the execution times in this problem.

ESSA results obtained with the finer (51 x 51 and 81 x 81) grids are shown in figures 5.10, 5.11 and 5.12. In figure 5.10, the execution time and number of iterations for tests with a grid of 51 x 51 are plotted against RES for KNMAX = 3. The minimum execution time is obtained with RES = 0.8. In figure 5.11, the execution times and number of iterations are compared for solution on the 51 x 51 node grid, utilizing the ESSA scheme with RES = 0.2, and with only one sweep in the TDMA solver. As is seen in this figure, the execution times obtained with one sweep in the TDMA solver are almost one-half of the execution times for RES = 0.2.

In figure 5.12, the execution time and number of iterations for the solution of this problem for different grids (31 x 31, 51 x 51, and 81 x 81) are plotted for the Saabas, SEVA, and ESSA schemes. The best results obtained with each of these schemes are shown in table 5.2. The execution times for the ESSA scheme is much less than corresponding execution times for the two other schemes, by a ratio of almost 1 to 2; the number of ESSA iterations are also less than the corresponding number of iterations for the other two schemes, by a ratio of almost 1 to 4.

In summary, for the problem of lid-driven flow in a square cavity, the ESSA scheme has a significant superiority to the Saabas and SEVA schemes in the rate of convergence. The convergence behaviour of SEVA is almost similar to that of the Saabas scheme, but SEVA is considerably more efficient than the Saabas scheme in terms of computer-storage requirements.

5.3.2 Flow Over a Backward-Facing Step

Problem Statement

This problem is usually used to test the treatment of the outflow boundary condition. The flow in this problem is assumed to be steady, two-dimensional and laminar. The fluid is assumed to be Newtonian and incompressible. This problem is illustrated schematically in Fig. 5.13. The fluid flow enters from the left through a parallel-plate channel of depth $\frac{H}{2}$, flows over a step of height $\frac{H}{2}$, and leaves from the right through a channel of depth H. For non-creeping flows, the flow will detach at the upper corner of the step and reattach on the lower wall at some distance x_r downstream. The origin of a Cartesian coordinate system (x,y) is located at the upper corner of the step. The computational domain extends in the positive x direction up to a distance 30H, and from $-\frac{H}{2}$ to $+\frac{H}{2}$ in the y direction.

The equations which describe this problem are the x- and y-momentum, and continuity equations (Eqs. 3.1-3.3). The following non-dimensional variables and parameters are used:

$$x^* = x/H; \qquad y^* = y/H; \qquad u^* = u/u_{av}; \qquad v^* = v/u_{av}$$

$$p^* = (p - p_{ref})/\rho u_{av}^2; \qquad Re_{av} = \rho u_{av} H/\mu$$

The resulting non-dimensional momentum and continuity equations are of the form:

x-momentum:

$$u^* \frac{\partial u^*}{\partial x^*} + v^* \frac{\partial u^*}{\partial y^*} = -\frac{\partial p^*}{\partial x^*} + \frac{1}{Re_{av}} \left(\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}} \right) \tag{5.5}$$

y-momentum:

$$u^* \frac{\partial v^*}{\partial x^*} + v^* \frac{\partial v^*}{\partial y^*} = -\frac{\partial p^*}{\partial y^*} + \frac{1}{Re_{av}} \left(\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}} \right) \tag{5.6}$$

and continuity:

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} = 0 \tag{5.7}$$

with the following boundary conditions:

$$\begin{aligned} u^* = v^* &= 0 && \textit{on horizontal walls} \\ v^* &= 0 && \textit{at inflow boundary} \\ u^*_{av} &= 1 && \textit{at inflow boundary } (0.0 \leq y \leq 0.5;\ x^* = 0) \end{aligned} \tag{5.8}$$

At the inlet, a parabolic u-velocity profile is prescribed, and the cross-flow v-velocity is set to zero. At the outflow boundary, it is ensured that the outgoing mass flow rate is identical to the incoming mass flow rate, but none of the velocity components or pressure are specified; they are calculated as part of the solution. The only free parameter in this problem is the Reynolds number, Re_{av}, based on the average velocity at the inlet and the height of the channel.

This problem has been investigated by Hackman, Raithby, and Strong [42] ,and by Gartling [37]. Gartling used a Galerkin-based finite element method with up to 32000 elements to obtain a bench-mark solution of this problem. He solved this problem for a flow with $Re_{av} = 800$, which is a relatively high Reynolds number laminar flow. Hackman et al. solved this problem with finite volume methods on Cartesian and curvilinear orthogonal meshes at $Re_{av} = 73$ and $Re_{av} = 229$ [3].

Recirculating flows are created in the vicinity of the step in the lower part of channel, and further in the upper part of channel. The detection of the upper recirculating zone is a good test for the different models. In this work, the reattachment lengths reported by Gartling [gart] and Hackman et al. [hack] were adequately predicted in

[3]The definition of the Re_{av} is not the same in the work of Hackman et al. to the definition used by Gartling. Gartling's definition is used in this work. Proper adjustment were done to compensate for this difference of definitions.

all tests. The focus in these tests, however, was on the performances of the ESSA and Saabas schemes, not on the accuracy of the corresponding FVM solutions.

Numerical Details

Nonuniform grids of 31 x 31 nodes, with power-law expansion of the grid lines (power = 1.4) away from the step in x-direction, and uniform spacing between grid lines in y-direction were used in the tests. These tests were performed with the Saabas and ESSA schemes. Since the convergence behaviour of the SEVA scheme was almost similar to the Saabas scheme in tests with lid-driven flow in a square cavity, it was decided to concentrate attention on the comparison of the convergence behaviour of the Saabas and ESSA schemes, only.

The value of under-relaxation parameter for u- and v-velocity discretized equations (RELAX(V)) in the inner loop of the ESSA scheme and also for the Saabas scheme was varied between 0.5 to 0.7, and its value for the outer loop of the ESSA scheme (ALPHAO) was varied between 0.9 to 1. The maximum number of cycles in the inner loop (KNMAX) was assigned values of 1 to 5, 10, and 15.

To optimize the TDMA solver for the discretized equations in the ESSA scheme, tests similar to those explained in the problem of lid-driven flow in a square cavity were performed. The convergence criteria for these tests were also similar to those defined for the problem of lid-driven flow in a square cavity. Finally, some tests with finer grids, of 51 x 51 and 81 x 81 nodes with the same grid distribution as the 31 x 31 node grid, were performed to study the effect of grid refinement on the convergence rate of the Saabas and ESSA schemes.

The only free parameter in this problem is Re_{av}, and all tests were performed with $Re_{av} = 800$. The execution times and number of iterations are reported.

Results and Discussion

In figures 5.14 and 5.15, the results obtained with a grid of 31 x 31 nodes utilizing the ESSA scheme are shown. The execution time and number of iterations required are plotted versus KNMAX for ALPHAO = 0.9 and 1, and RELAX(V) =0.6 in figure 5.14. The results obtained with ALPHAO = 1 show much faster convergence and smaller number of iterations. In figure 5.15, the execution time and number of iterations are plotted versus KNMAX for RELAX(V) = 0.5 and 0.6, at ALPHAO = 1. The tests with RELAX(V) $\geq$ 0.7 failed to converge for both the ESSA and Saabas schemes. The execution time and number of iteration for RELAX(V) = 0.6 are less than the corresponding results obtained with RELAX(V) = 0.5. Therefore, for this problem, the optimum values for under-relaxation parameters are: RELAX(V) = 0.6 and ALPHAO = 1.

The results shown in figure 5.14 were obtained by utilizing the suggestion in [122] for the optimization of the TDMA solver, with RES = 0.2. The corresponding results in figure 5.15 were obtained by performing only one sweep in the TDMA solver. By comparing the execution times in these two figures, the superiority of ESSA with one sweep in the TDMA solver is evident. These results confirm the findings of the results obtained in the problem of lid-driven flow in a square cavity.

In table 5.3, the execution times and number of iterations required by the ESSA and Saabas schemes for different grids are shown. For both schemes, only one sweep was performed in the TDMA solver. The execution times are almost halved and the number of iterations are decreased by a factor of about 5 with the ESSA scheme, in comparison to the Saabas scheme, in this problem for all grid sizes.

5.3.3 Natural Convection in a Square Enclosure

Problem Statement

The numerical simulation of two-dimensional natural convection in a square enclosure is a standard test problem that is used to evaluate the effectiveness of numerical methods in the solution of coupled fluid flow and heat transfer. In this problem, steady, two-dimensional, laminar natural convection of a Newtonian fluid contained in a square enclosure is considered. This problem is illustrated schematically in Fig. 5.16. The origin of a Cartesian coordinate system (x, y) is located at the lower left corner of the square enclosure of dimension L. The acceleration due to gravity, $\vec{g}$, is directed in the negative y direction. The vertical side walls at $x = 0$ and at $x = \mathrm{L}$, are maintained at hot and cold temperatures, T_h and T_c, respectively. The two horizontal walls are considered to be adiabatic. The standard Boussinesq approximation is used in the analysis of this problem: the mass density is considered to be constant in all terms except the buoyancy term, in which it is assumed to decrease linearly with temperature:

$$\rho = \rho_c[1 - \beta(T - T_c)] \tag{5.9}$$

where β is the thermal volumetric expansion coefficient, and ρ_c is the density of the fluid at the reference temperature, T_c. All other thermophysical properties of the fluid are considered to be constant.

The equations which govern this problem are the momentum, continuity and energy equations. The following non-dimensional variables and parameters are used:

$$x^* = x/L; \qquad y^* = y/L; \qquad u^* = uL/\alpha; \qquad v^* = vL/\alpha$$

$$p^* = (p - p_{ref})L^2/\rho\alpha^2; \qquad \phi = (T - T_c)/(T_h - T_c) \tag{5.10}$$

$$Pr = \nu/\alpha; \qquad Ra = \beta g \Delta T L^3/\alpha\nu$$

where Ra is the Rayleigh number and Pr is the Prandtl number.

The non-dimensional forms of the governing equations are:

x-momentum:

$$u^*\frac{\partial u^*}{\partial x^*} + v^*\frac{\partial u^*}{\partial y^*} = -\frac{\partial p^*}{\partial x^*} + Pr\left(\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}}\right) \tag{5.11}$$

y-momentum:

$$u^*\frac{\partial v^*}{\partial x^*} + v^*\frac{\partial v^*}{\partial y^*} = -\frac{\partial p^*}{\partial y^*} + Pr\left(\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}}\right) + RaPr\phi \tag{5.12}$$

continuity:

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} = 0 \tag{5.13}$$

and energy equation:

$$u^*\frac{\partial \phi}{\partial x^*} + v^*\frac{\partial \phi}{\partial y^*} = \frac{\partial^2 \phi}{\partial x^{*2}} + \frac{\partial^2 \phi}{\partial y^{*2}} \tag{5.14}$$

with the following boundary conditions:

$$\begin{aligned} u^* = v^* &= 0 \qquad\qquad\qquad on\ \ all\ \ walls \\ \phi &= \begin{cases} 1 \quad at \quad x^* = 0, & 0 < y^* < 1 \\ 0 \quad at \quad x^* = 1, & 0 < y^* < 1 \end{cases} \\ \frac{\partial \phi}{\partial y^*} &= 0 \qquad\qquad at \quad y^* = 0,\ 1 \end{aligned} \tag{5.15}$$

It should be noted that the y-momentum equation has a source term which involves ϕ, and leads to a coupling of the fluid flow and heat transfer problems. The local Nusselt number can be evaluated for the hot wall, once the solution is achieved for the ϕ field, as:

$$Nu = \frac{(-k\partial T/\partial x)_{x=0}}{(T_h - T_c)}\left(\frac{L}{k}\right) = \left(\frac{\partial \phi}{\partial x^*}\right)_{x^*=0} \tag{5.16}$$

and the average Nusselt number on the hot wall can be evaluated as:

$$Nu_{av} = \int_0^1 \left(\frac{\partial \phi}{\partial x^*}\right)_{x^*=0} dy^* \tag{5.17}$$

Since the total amount of heat transferred from the hot wall to the fluid is the same as the amount absorbed by the cold wall from the fluid, and because the hot and cold wall areas are equal, the average Nusselt number on both walls should be the same.

A benchmark solution of this problem is available in the work done by de Vahl Davis [24]. He used a stream-function-vorticity based finite-difference method, with successively finer grids and extrapolation procedures, to arrive at a 'best' solution. Tabular data for the values of Nu_{av}, and maximum u- and v-velocity components on horizontal and vertical midplanes, respectively, are given in [24].

Numerical Details

As in the previous problems, non-uniform grids of 31 x 31 nodes, with a power-law expansion of the grid lines (power = 1.4) away from the solid boundaries in both x- and y-directions were used to perform tests on the Saabas and ESSA schemes. In these tests, the values of the under-relaxation parameters in the discretized velocity equations (RELAX(V)) in the inner loop of the ESSA scheme and for the Saabas scheme were varied between 0.5 to 0.8. No under-relaxation was used for the solution of the discretized velocity equations in the outer loop of the ESSA scheme (ALPHAO = 1).

The under-relaxation parameter for the solution of the discretized temperature equations (RELAX(T))in both the Saabas and ESSA schemes was set equal to 0.9. The maximum number of cycles in the inner loop of ESSA in these tests were : KNMAX = 1 to 5, 10, 15, and 20. To optimize the suggestions in [122] for the TDMA solver, tests were performed with the values of RES in the range 0.1 to 0.9, for values of KNMAX = 1 to 5. Tests with finer grids, of 51 x 51 and 81 x 81 nodes, with the same power-law distribution as 31 x 31 node grids were performed to study the effect of grid refinement on the convergence behaviour of the Saabas and ESSA schemes for this problem. These tests were performed for RELAX(V) = 0.6 and 0.7,

with ALPHAO = 1 and RELAX(T) = 0.9.

The convergence criteria for these tests required the satisfaction of the following conditions:

1. the relative change in the value of average Nusselt number on the left wall should be less than 10^{-5};
2. the summation of average Nusselt number on the left and right walls divided by the its value on the left wall should be less than 10^{-5}; and
3. the relative change in the value of the sum of the absolute values of the constant terms in the discretized pressure equation should be less than 10^{-6}.

In inner loop, condition number 1 should be satisfied to stop the cycles.

These tests were performed with air as the fluid, for $Ra = 10^6$. The execution times and number of iterations are reported for the tests in which converged solutions were obtained.

Results and Discussion

The results obtained for the 31 x 31 node non-uniform grid utilizing the ESSA scheme are shown in figure 5.17. The execution times and number of iterations for converged solution are plotted versus KNMAX, for 3 different values of RELAX(V), 0.5, 0.6, and 0.7. The solutions with RELAX(V) $\geq$ 0.8 failed to converge. As is seen in this figure, the solutions obtained with RELAX(V) = 0.7 require more execution time than the solutions obtained with RELAX(V) less than 0.7. This reflects the fact that the iterative solution procedure did not converge smoothly and monotonically for RELAX(V) = 0.7. Therefore, for this problem, the best convergence rate is obtained with RELAX(V) = 0.6.

The tests with RELAX(T) = 1.0 failed to converge, but as is shown in this figure 5.17, RELAX(V) = 0.6 and RELAX(T) = 0.9 work well. In this problem, similar

to the previous problems, no under-relaxation is required for the u- and v-velocity discretized equations in the outer-loop with the ESSA scheme. Indeed, ALPAHO = 1 gives the best convergence behaviour.

In figure 5.18, the results obtained for optimization of Res values for terminating iterations in the TDMA solver are shown. The execution times and number of iterations are plotted versus KNMAX for three values of RES: 0.1, 0.2, and 0.3. Tests with higher values for RES either failed to converge or diverged. Two cases converge faster than the other cases, namely, KNMAX = 2 and RES = 0.3, and KNMAX = 3 and RES = 0.1.

For this problem, in contrast with the experience in the previous problems, the use of only one sweep in the TDMA solver was not very successful. It appear that with such a high Rayleigh number, the larger buoyancy term in the v-velocity equation requires more sweeps in the TDMA solver for a faster convergence of the overall iterative procedure.

The results for tests with a finer grid of 51 x 51 nodes utilizing the ESSA scheme are shown in figure 5.19 for two values of RELAX(V): 0.6, and 0.7. For tests with RELAX(V) = 0.7, the fastest convergence is obtained with KNMAX = 20. For tests with RELAX(V) = 0.6, the results obtained for KNMAX = 1 are much better than the other results in term of the execution time.

In table 5.4, the results obtained with the Saabas and ESSA schemes are compared for different number of nodes in the domain discretization [4]. As is seen, the execution times and number of iterations for tests with the ESSA scheme are much lower than those for corresponding tests with the Saabas scheme.

These results confirm the conclusions drawn from studies of the previous problems: the performance of the ESSA scheme is significantly better than that of the Saabas

[4]The results shown with the Saabas scheme for a grid with 31 x 31 nodes were obtained with RELAX(V) = 0.6 and RELAX(T) = 0.8. The test with RELAX(T) = 0.9 for this case diverged.

scheme in terms of both the execution time and number of iterations required to achieve converged solution.

5.4 Summary

In this chapter, three schemes for solution of the discretized equations, namely, the Saabas, SEVA, and ESSA schemes, were evaluated. These studies were performed using three well-established test problems: lid-driven laminar flow in a square cavity; laminar flow over a backward-facing step; and natural convection in a square enclosure. The results show that the ESSA scheme requires significantly less execution time and number of iterations to produce converged solutions in comparison to the Saabas and SEVA schemes. The reduction in the execution time was up to 65%, and the reduction in number of iterations was up to 75%, in comparison to the Saabas and SEVA schemes.

The SEVA scheme was tested in the problem of lid-driven flow in a square cavity, and it shows almost the same convergence behaviour as the Saabas scheme, but it provides a considerable reduction in the computer-storage requirements.

Also, the recommendations of Van Doormaal and Raithby [122] for optimization of the line-by-line TDMA solver for the linearized, decoupled sets of discretized equations was examined. It was found that this idea improves the convergence rate of the overall solution, but the optimum choice for the first two problems is to use only one sweep of the line-by-line TDMA solver. The results with this optimization show superior performance to the results obtained with the recommendations in [122], with a fixed value of RES, in terms of overall execution times required to obtain converged solutions.

Optimization Method		Execution Time	Number of Iter.
Suggestion in [122]	KMAX = 3	112.71	47
	KMAX = 5	126.06	36
One sweep in the TDMA solver		80.36	26

Table 5.1: **Lid-Driven Flow in A Square Cavity: Performance of ESSA with Optimized RES Values and with Only One Sweep in TDMA Solver**

GRID SIZE	SAABAS	SEVA	ESSA
31 x 31	137.59	167.85	73.05
51 x 51	761.87	724.52	401.06
81 x 81	4039.50	4483.23	3565.54

(a)

GRID SIZE	SAABAS	SEVA	ESSA
31 x 31	199	164	44
51 x 51	396	383	83
81 x 81	730	907	287

(b)

Table 5.2: **Lid-Driven Flow in A Square Cavity: Convergence Behaviour of Saabas, SEVA and ESSA Schemes: (a) Execution Times; (b): Number of Iterations**

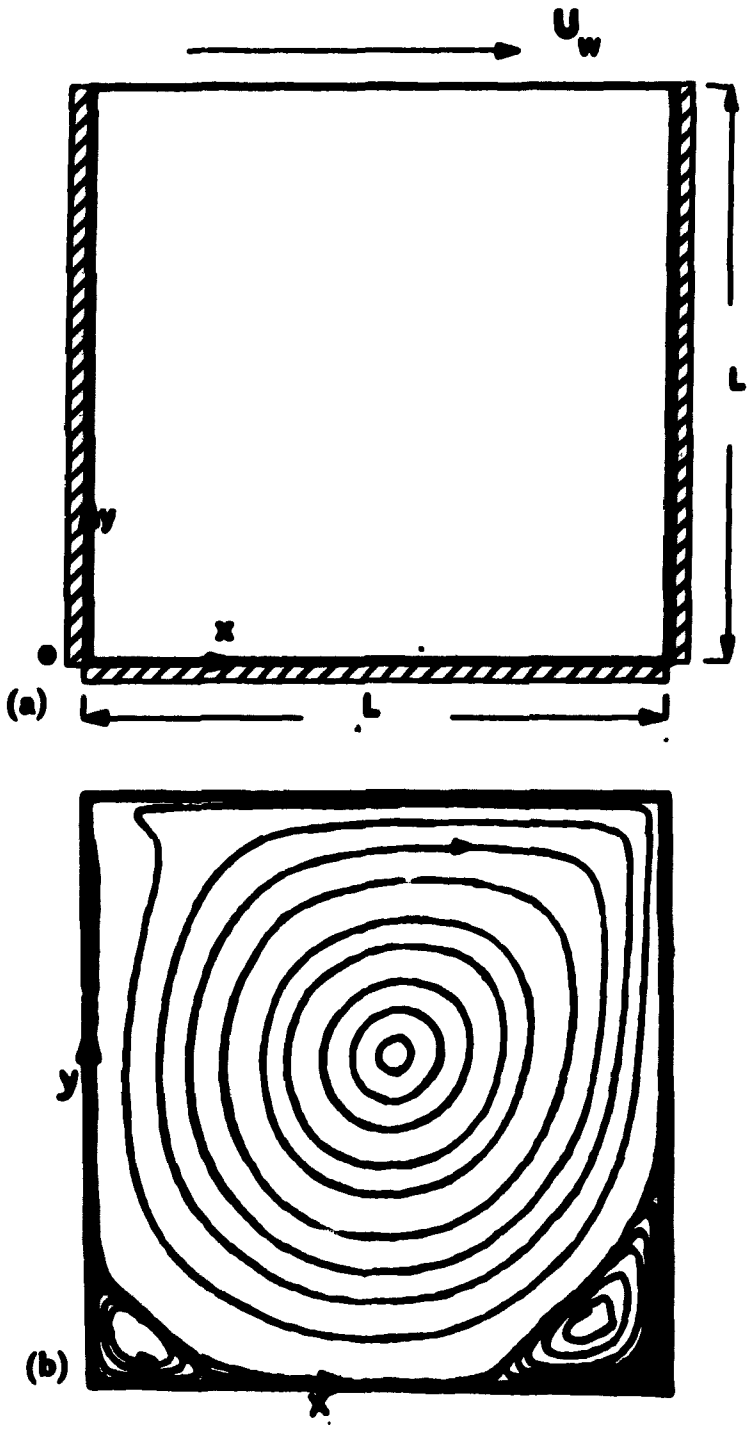


Figure 5.1: Schematic Representation of Lid-Driven Flow in A Square Cavity: (a) Geometry; and (b) Flow Pattern

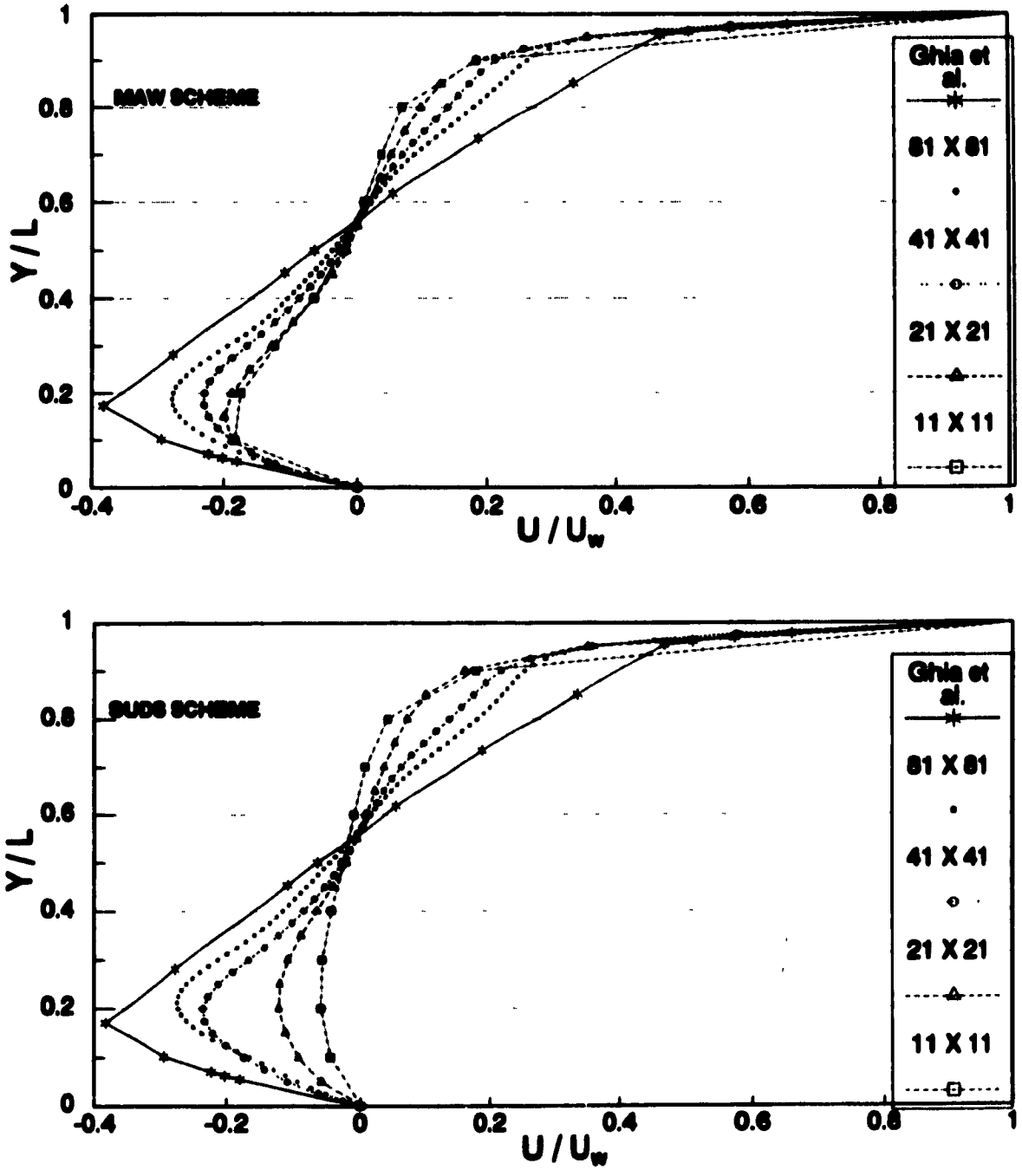


Figure 5.2: Preliminary Tests to Establish Consistency of the Proposed FVM: (a) the Results Obtained with the MAW Scheme; (b) Results Produced by the SUDS Scheme

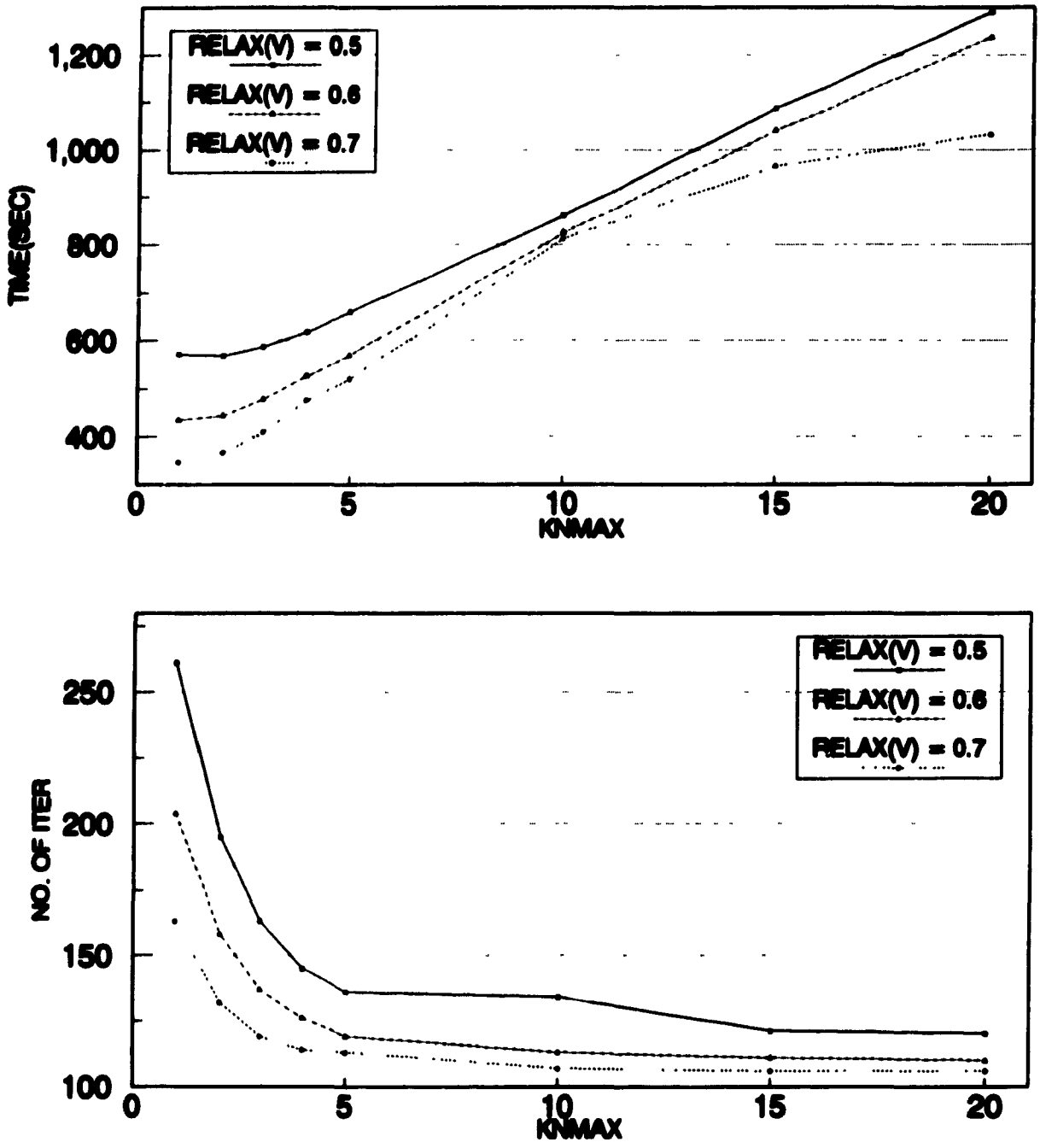


Figure 5.3: Lid-Driven Flow in A Square Cavity: Convergence Behaviour of ESSA: ALPHAO = 0.8; RELAX(V) = 0.5, 0.6, and 0.7

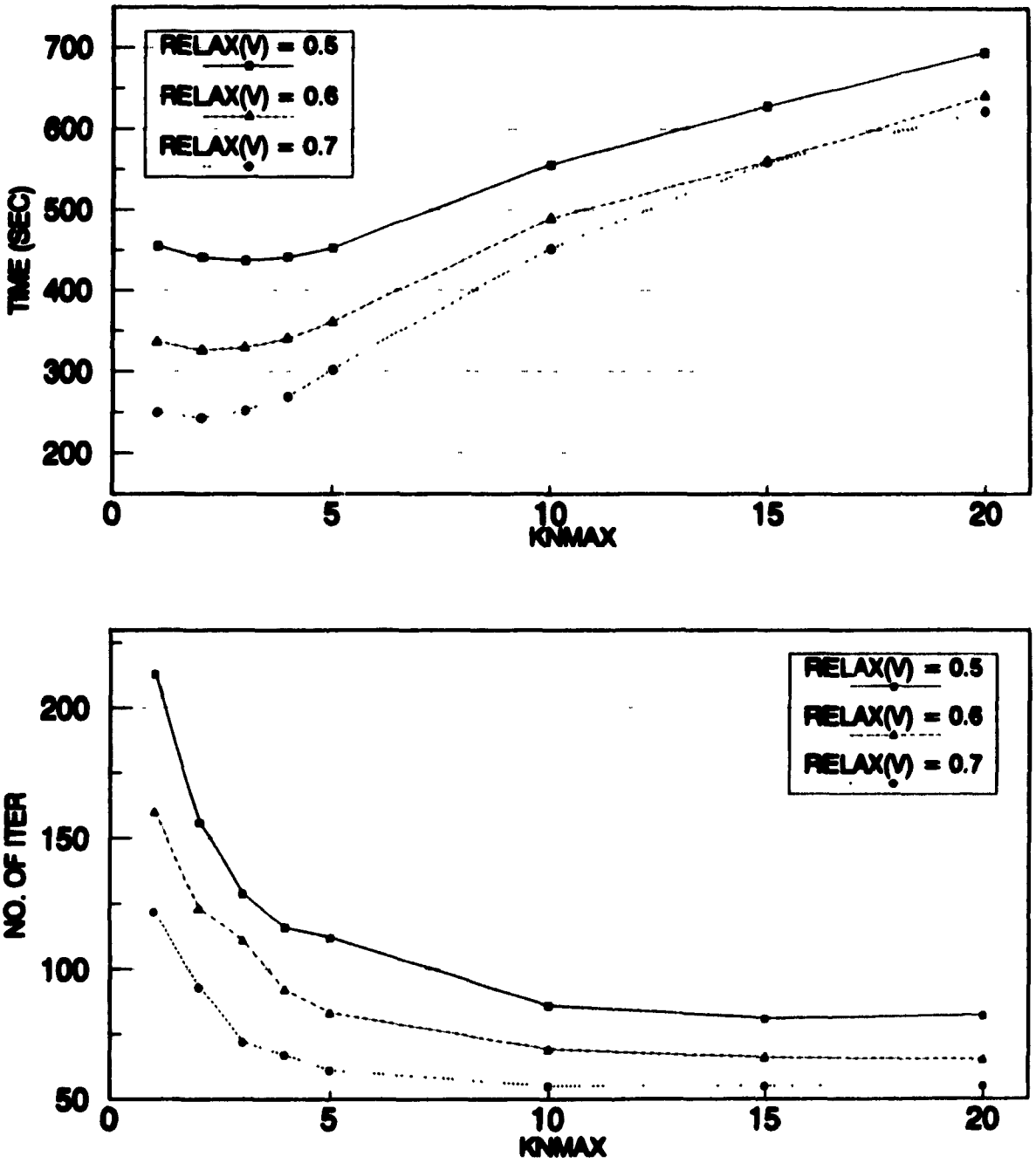


Figure 5.4: Lid-Driven Flow in A Square Cavity: Convergence Behaviour of ESSA: ALPHAO = 0.9; RELAX(V) = 0.5, 0.6, and 0.7

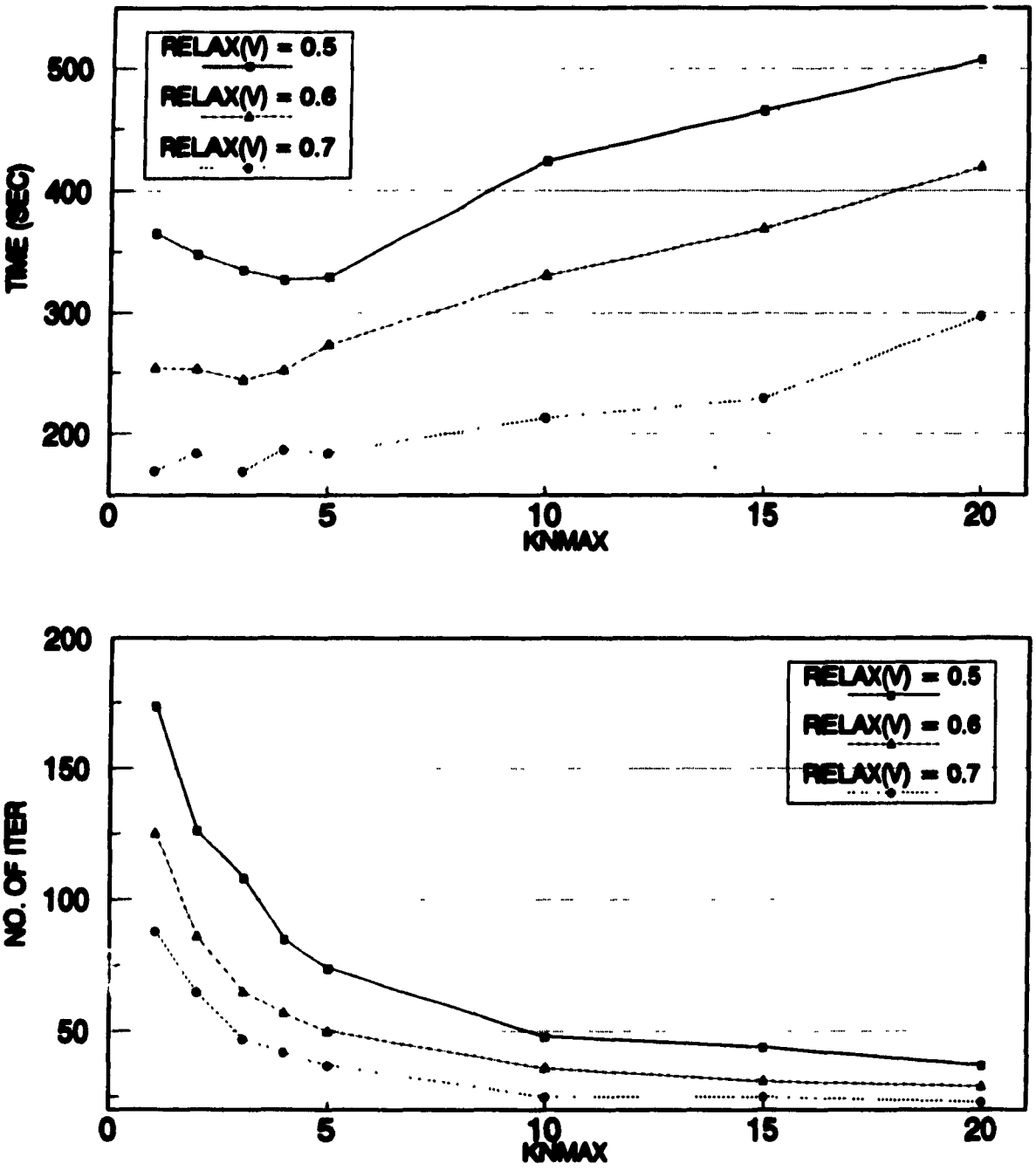


Figure 5.5: Lid-Driven Flow in A Square Cavity: Convergence Behaviour of ESSA: ALPHAO = 1.0; RELAX(V) = 0.5, 0.6, and 0.7

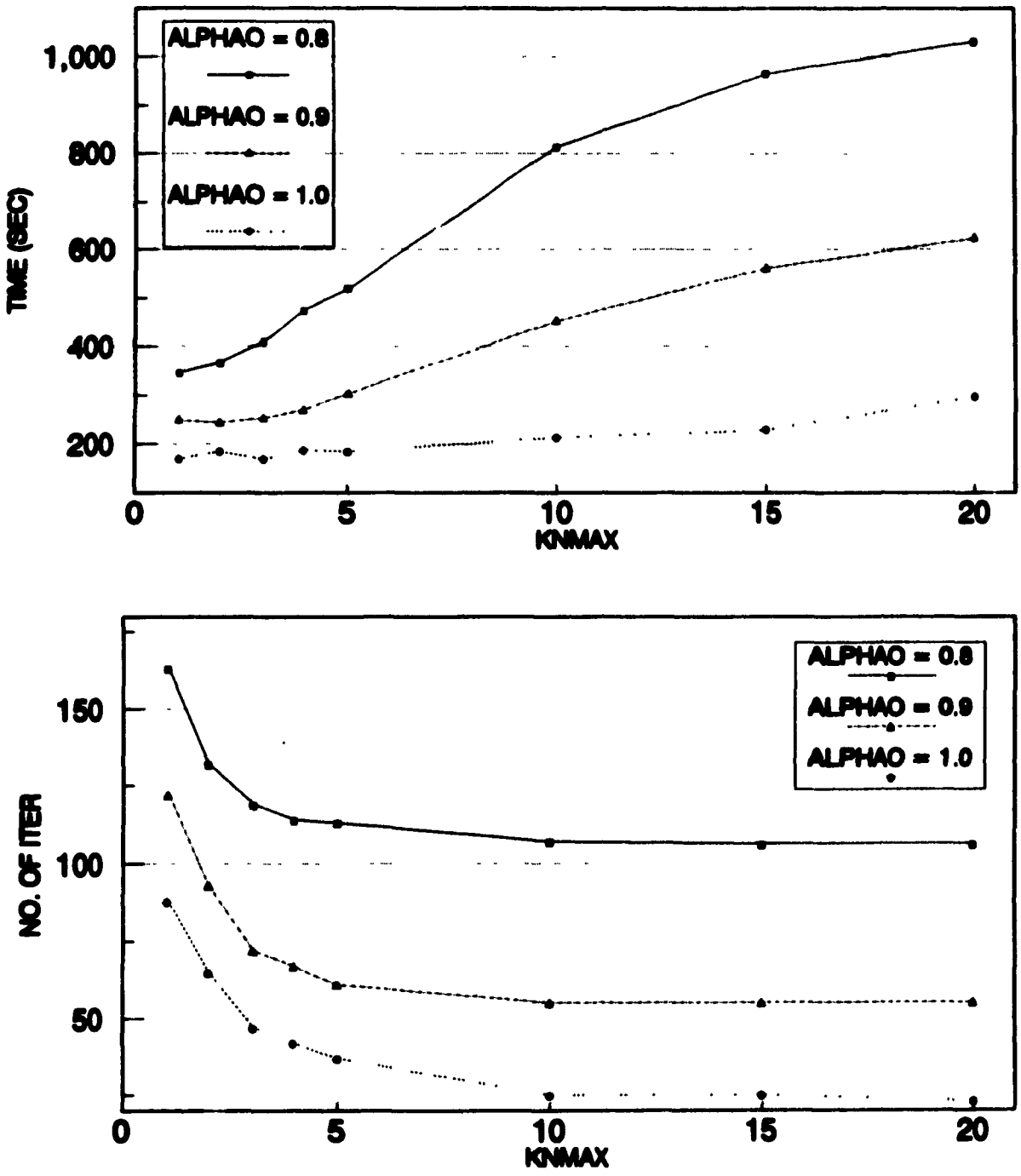


Figure 5.6: Lid-Driven Flow in A Square Cavity: Convergence Behaviour of ESSA: RELAX(V) = 0.7; ALPHAO = 0.8, 0.9, and 1.0

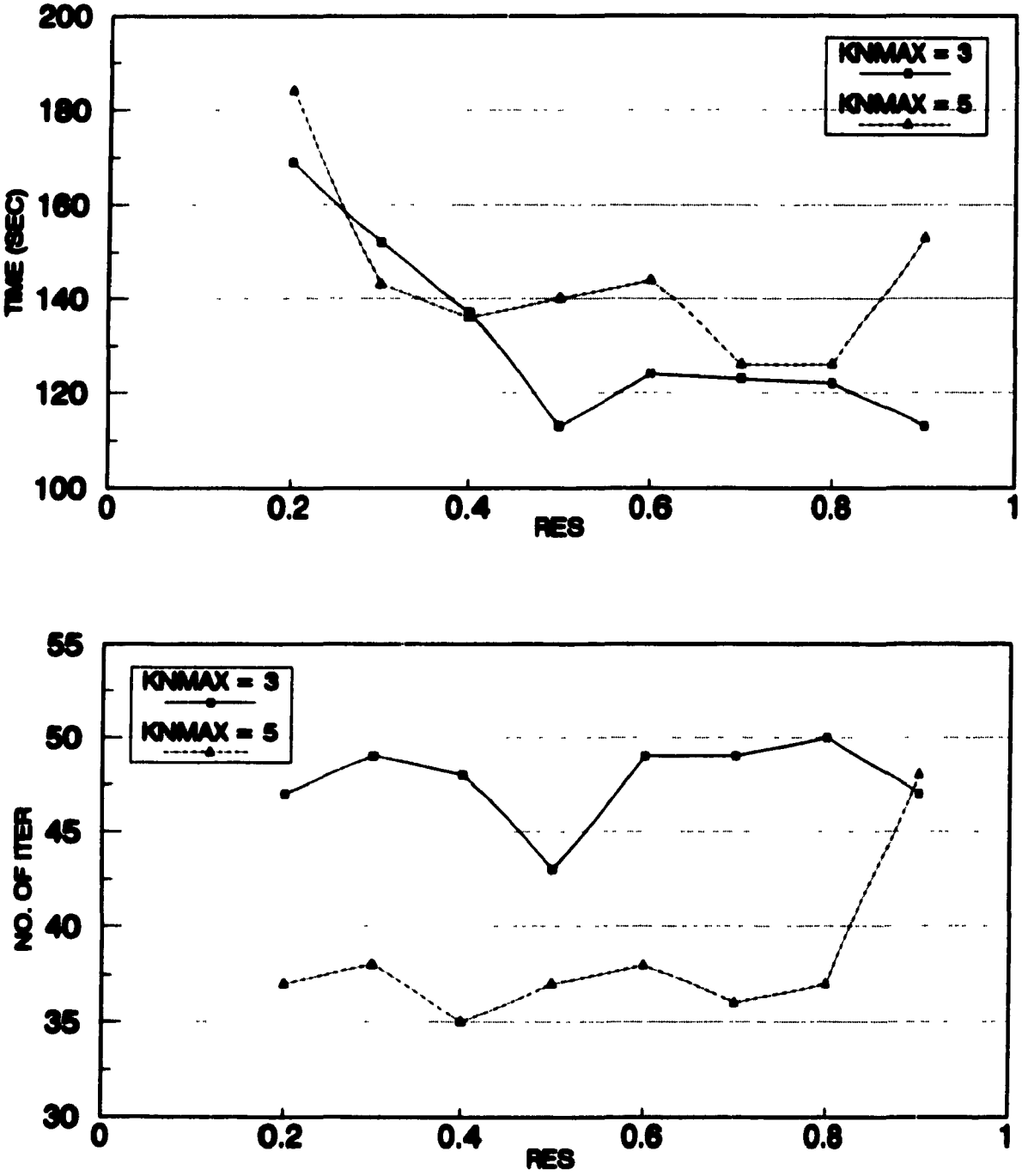


Figure 5.7: Lid-Driven Flow in A Square Cavity: Effect of RES on the Convergence of ESSA: ALPHAO = 1.0; RELAX(V) = 0.7

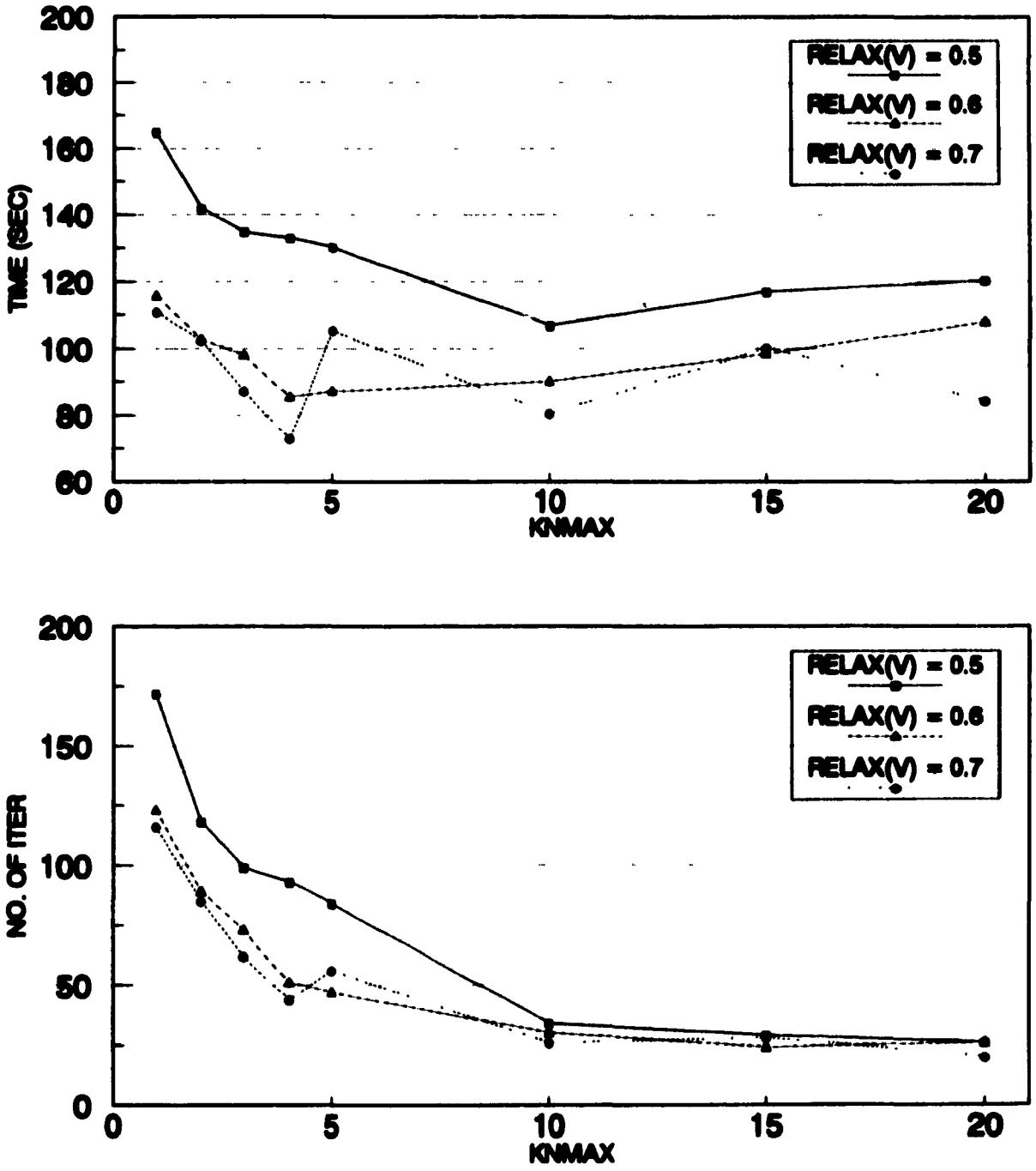


Figure 5.8: Lid-Driven Flow in A Square Cavity: Performance of ESSA with One Sweep in Line-by-Line TDMA Solver: ALPHAO = 1.0

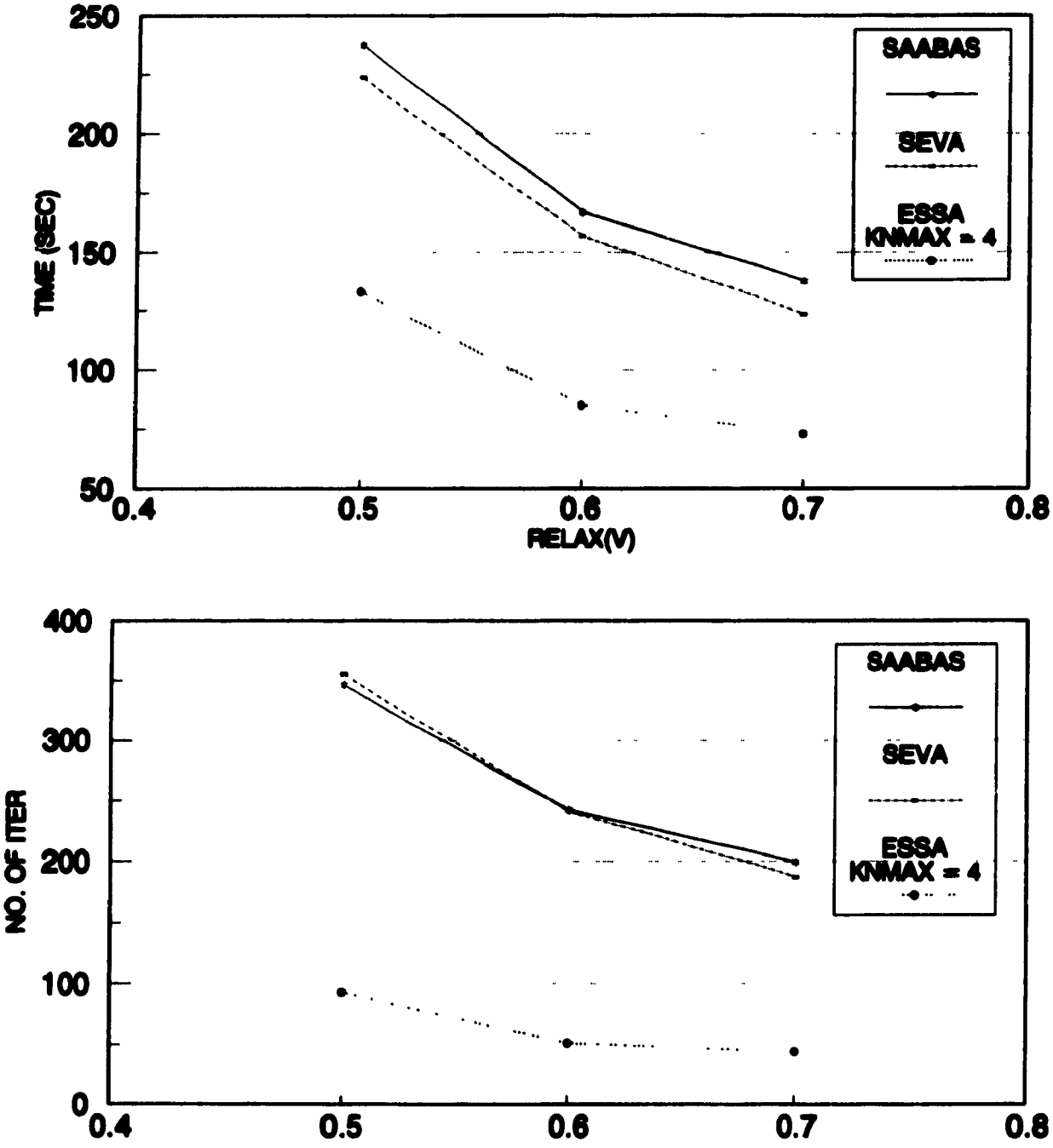


Figure 5.9: Lid-Driven Flow in A Square Cavity: Comparison of the Convergence Behaviour of the Saabas, SEVA, and ESSA schemes on a 31 x 31 Node Grid

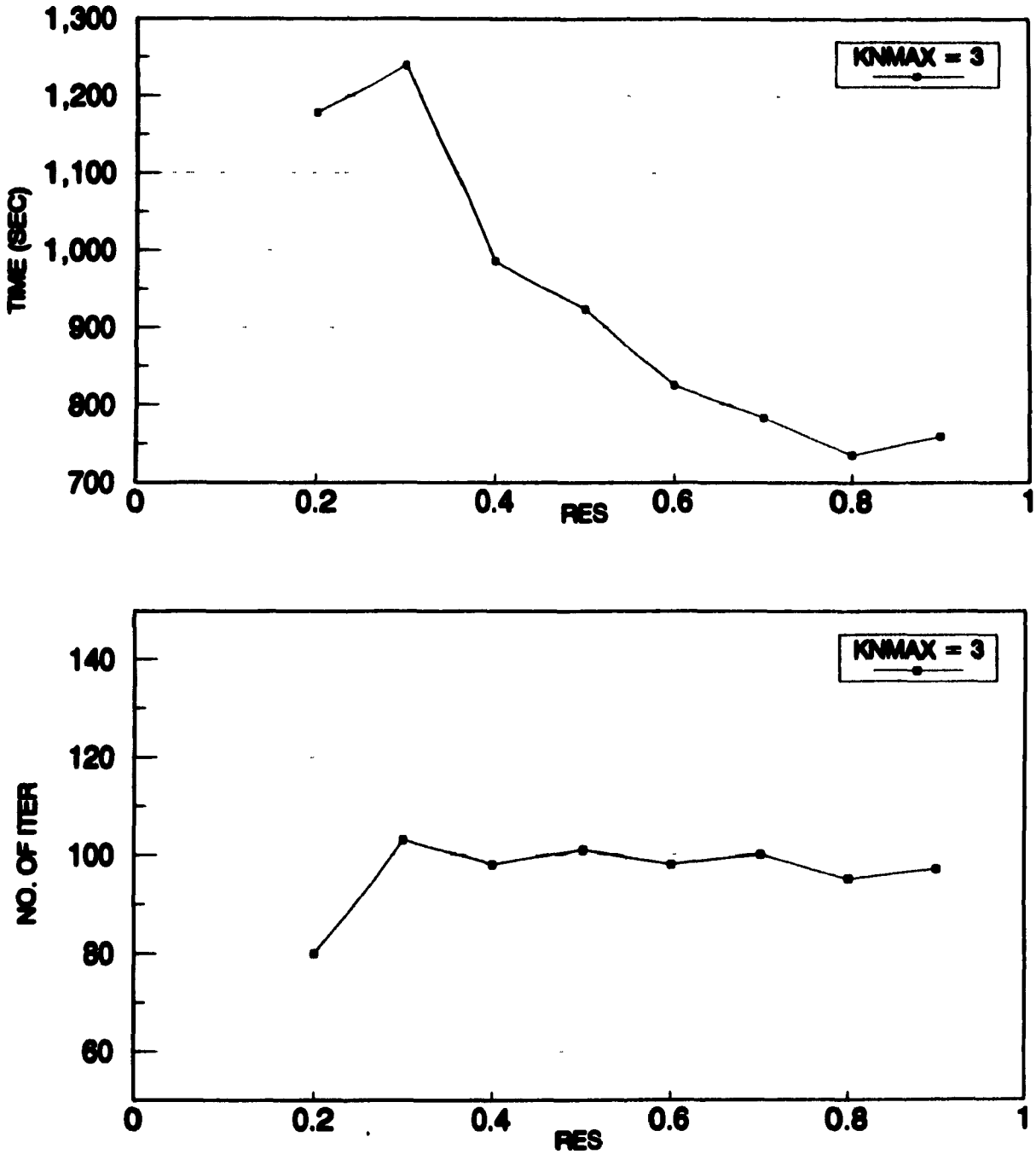


Figure 5.10: Lid-Driven Flow in A Square Cavity: Effect of RES on the Convergence Behaviour of ESSA on a 51 x 51 Node Grid: ALPHAO = 1.0; RELAX(V) = 0.7

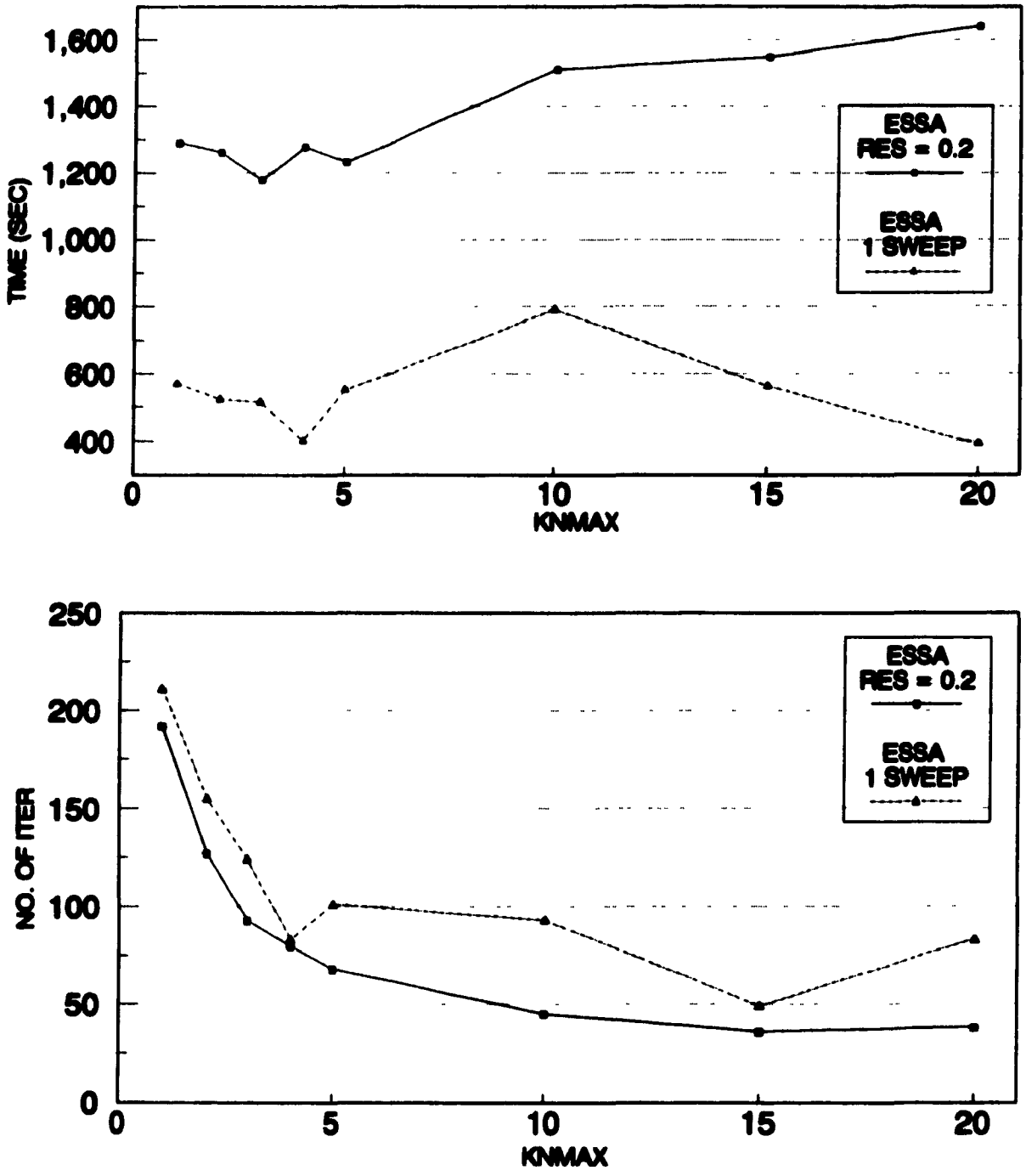


Figure 5.11: Lid-Driven Flow in A Square Cavity: Convergence Behaviour of ESSA on a 51 x 51 Node Grid: ALPHAO = 1.0; RELAX(V) = 0.7

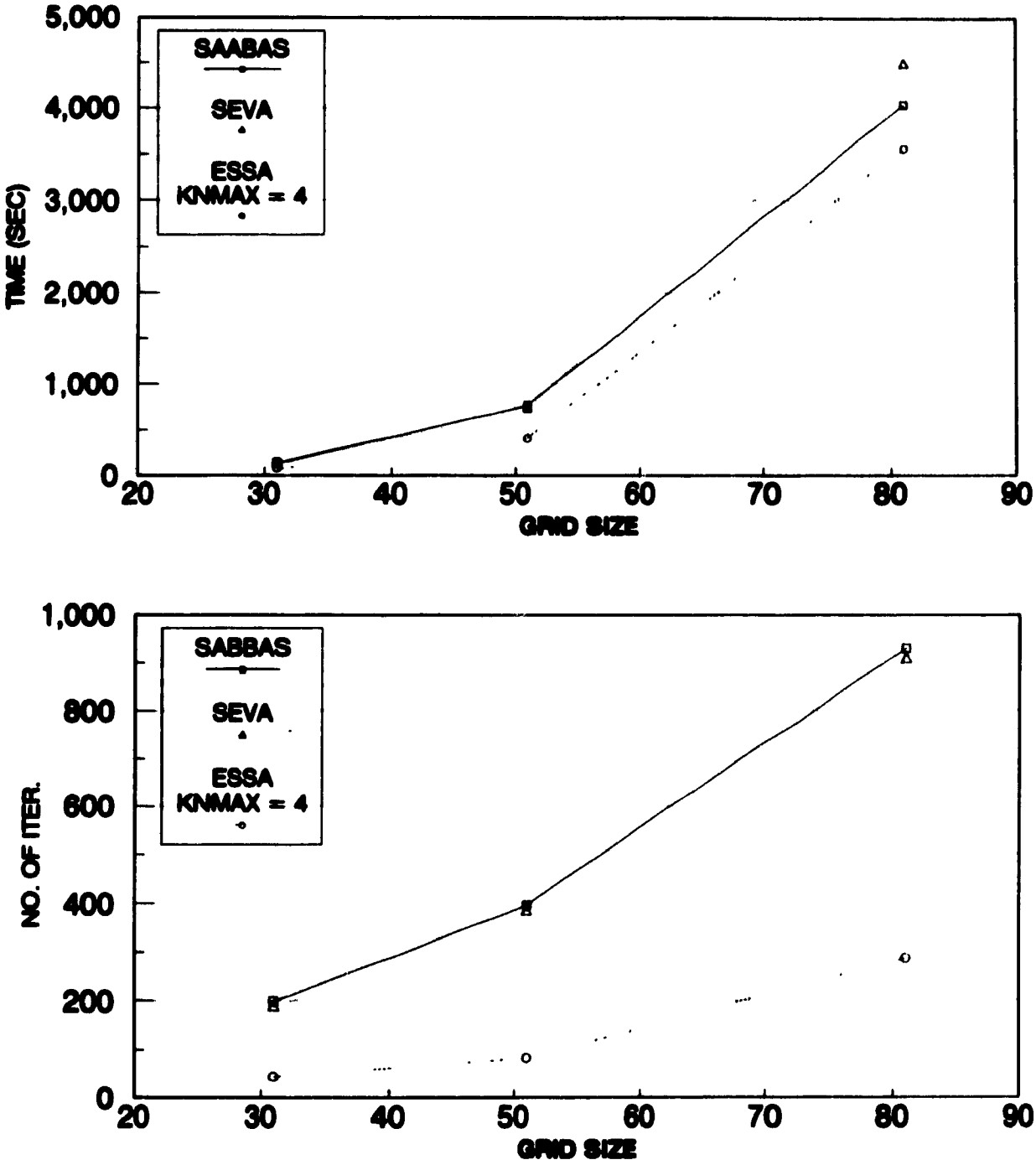
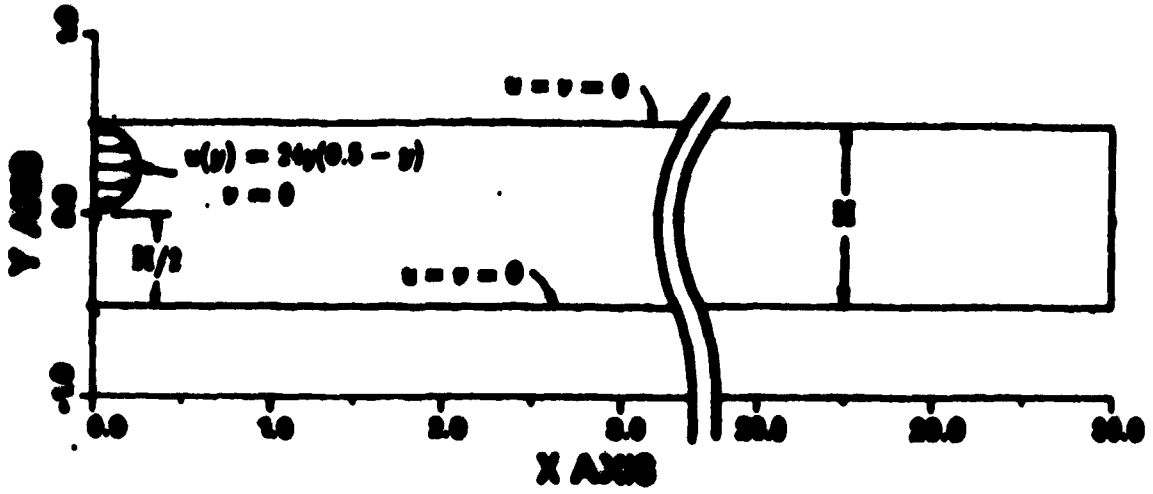
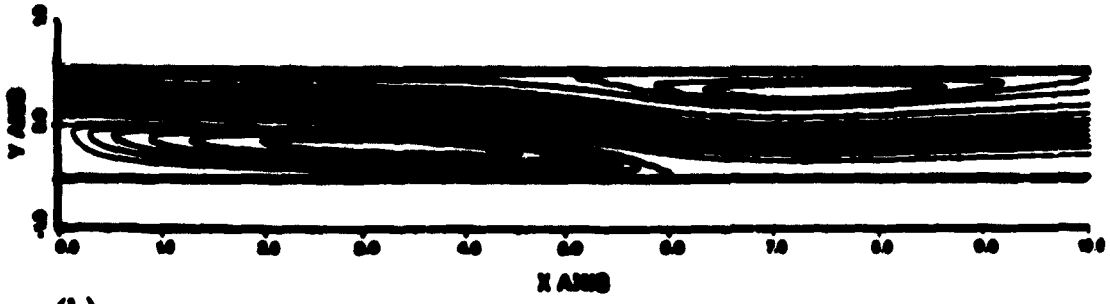


Figure 5.12: Lid-Driven Flow in A Square Cavity: Convergence Behaviour of Saabas, SEVA, and ESSA Schemes for Different Grid Sizes



(a)



(b)

Figure 5.13: **Schematic Representation of Flow Over a Backward-Facing Step: (a) Geometry; and (b) Flow Pattern**

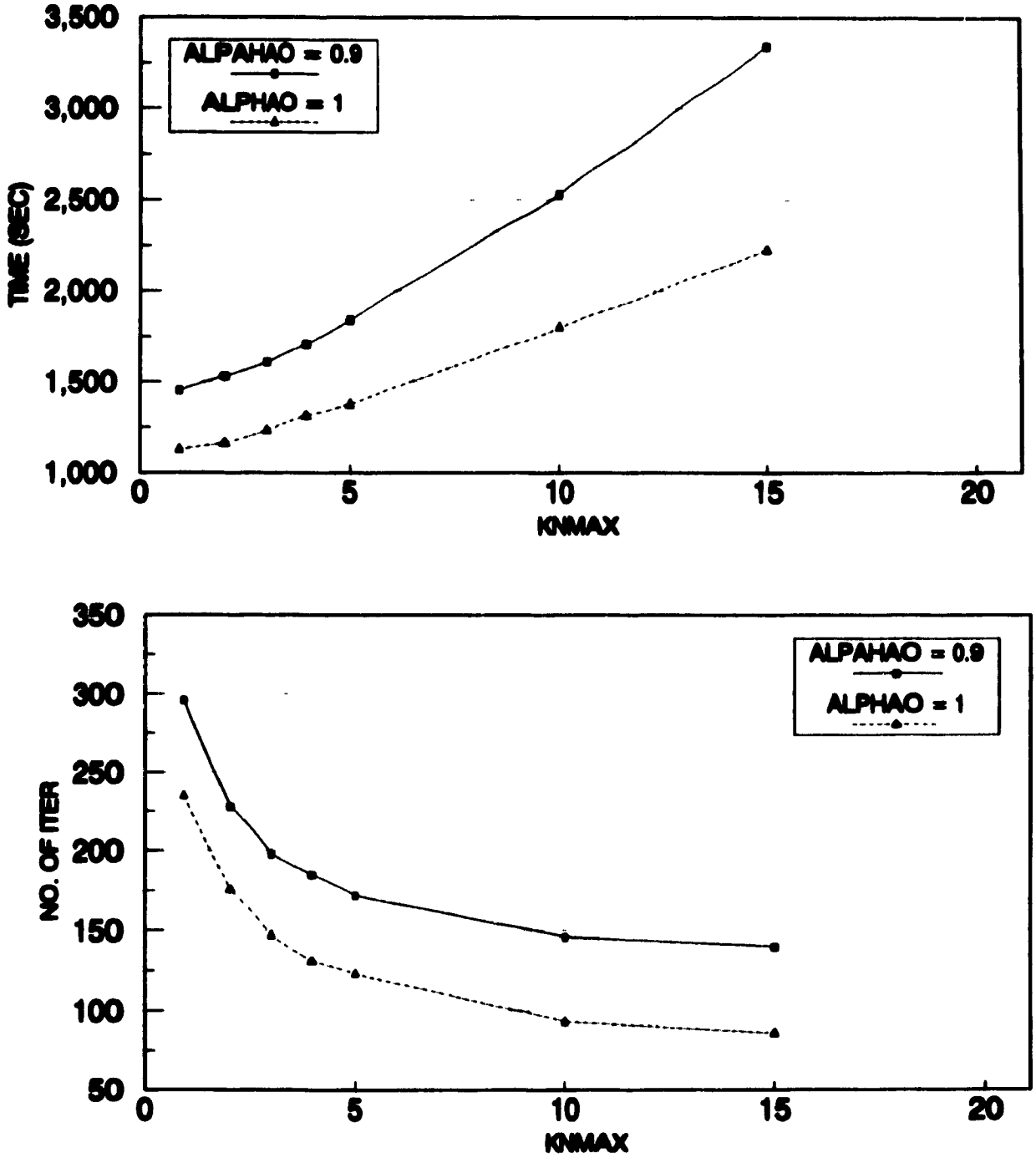


Figure 5.14: Flow Over a Backward-Facing Step: Convergence Behaviour of ESSA: RES = 0.2; RELAX(V) = 0.6

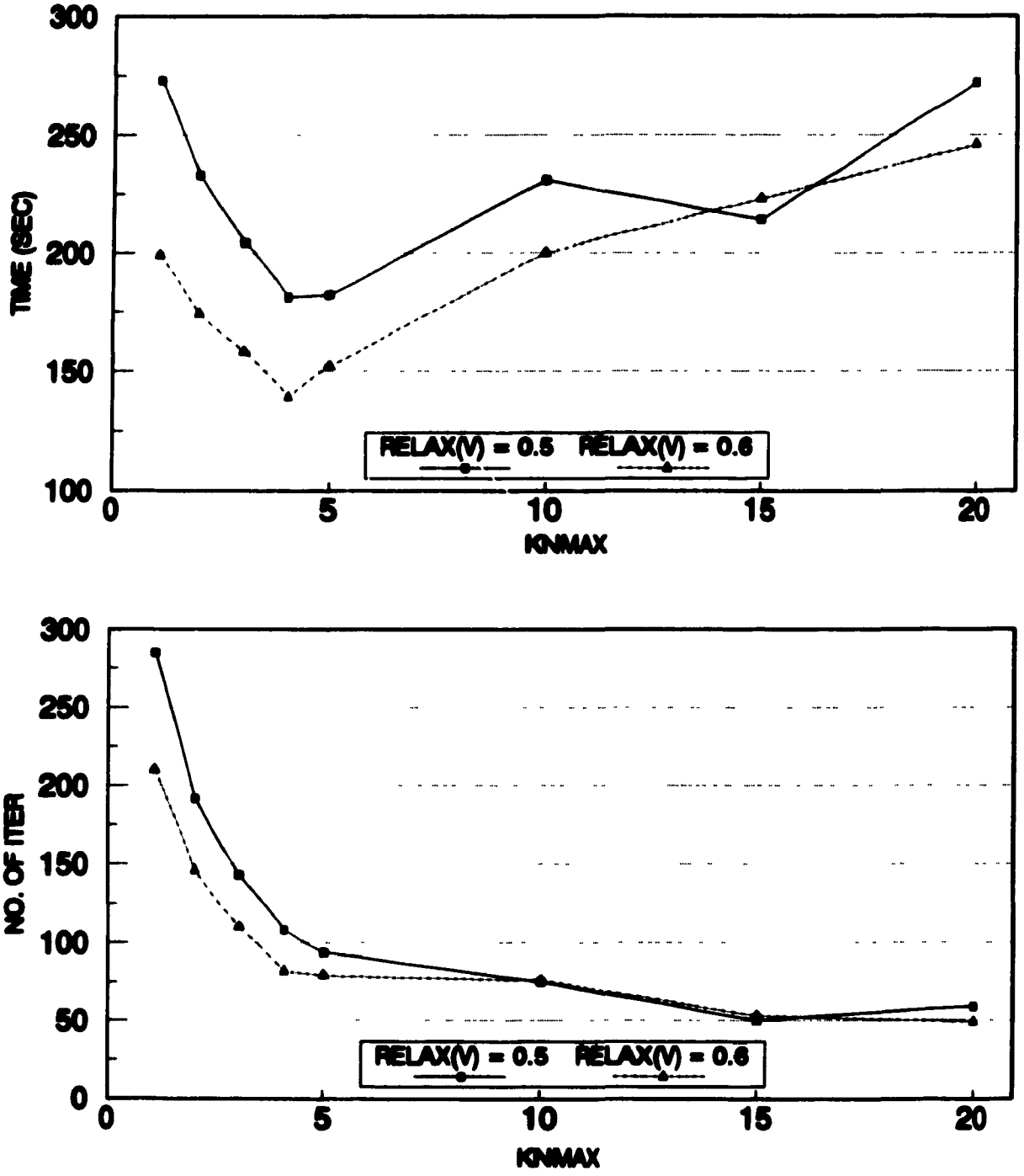


Figure 5.15: Flow Over a Backward-Facing Step: Convergence Behaviour of ESSA with One Sweep in TDMA Solver: ALPHAO = 1.

GRID SIZE	Saabas (t_s)	ESSA (t_E)	t_E / t_s
31 x 31	274.90	139.62	0.51
51 x 51	1451.96	701.8	0.48
81 x 81	7737.55	3791.4	0.49

(a) EXECUTION TIMES

GRID SIZE	SAABAS (N_s)	ESSA (N_E)	N_E / N_s
31 x 31	399	82	0.21
51 x 51	735	125	0.17
81 x 81	1443	260	0.18

(b) NUMBER OF ITERATIONS

Table 5.3: **Flow Over a Backward-Facing Step: Comparison of the Convergence Behaviour of the Saabas and ESSA Schemes for 3 Different Grid Sizes: RELAX(V) = 0.6; ALPHAO = 1**

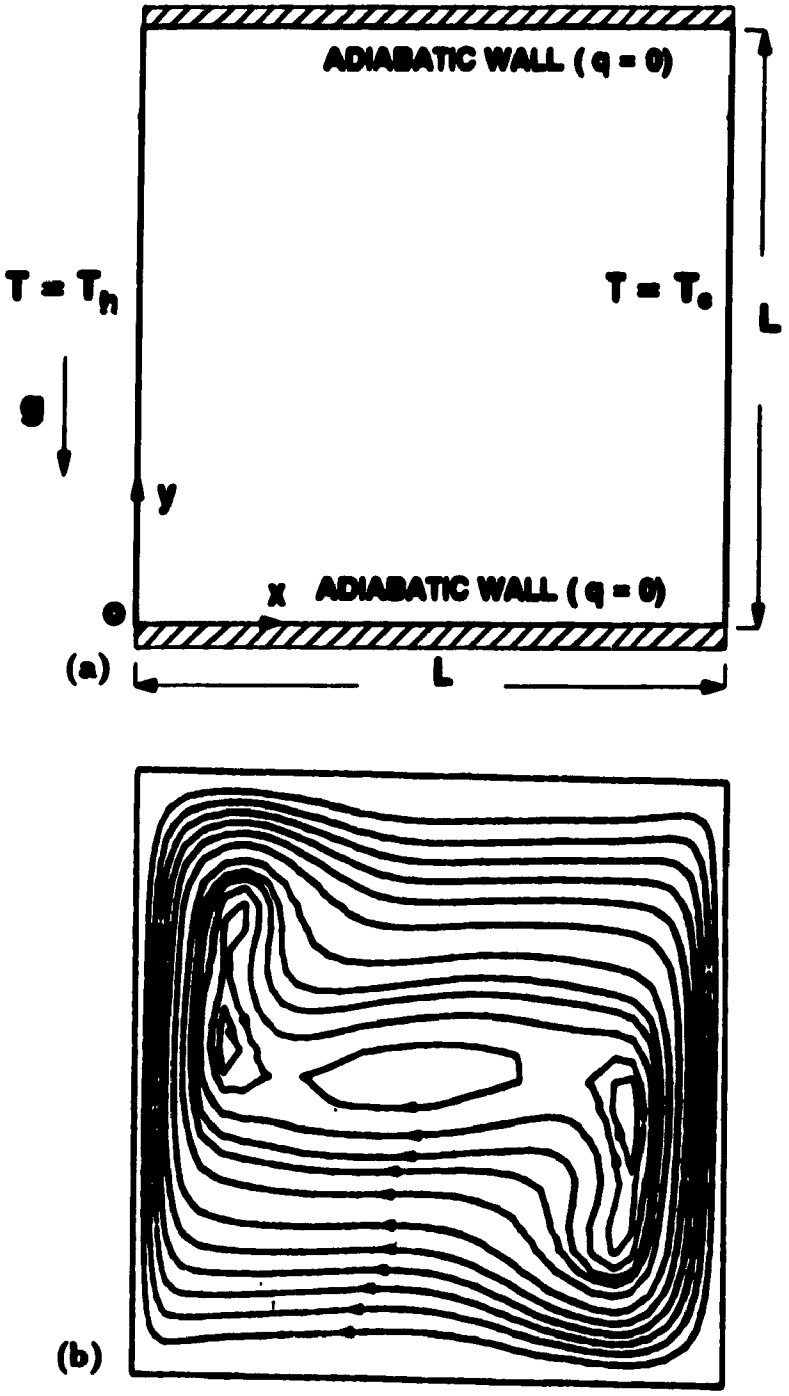


Figure 5.16: **Schematic Representation of Natural Convection in a Square Enclosure: (a) Geometry; and (b) Flow Pattern**

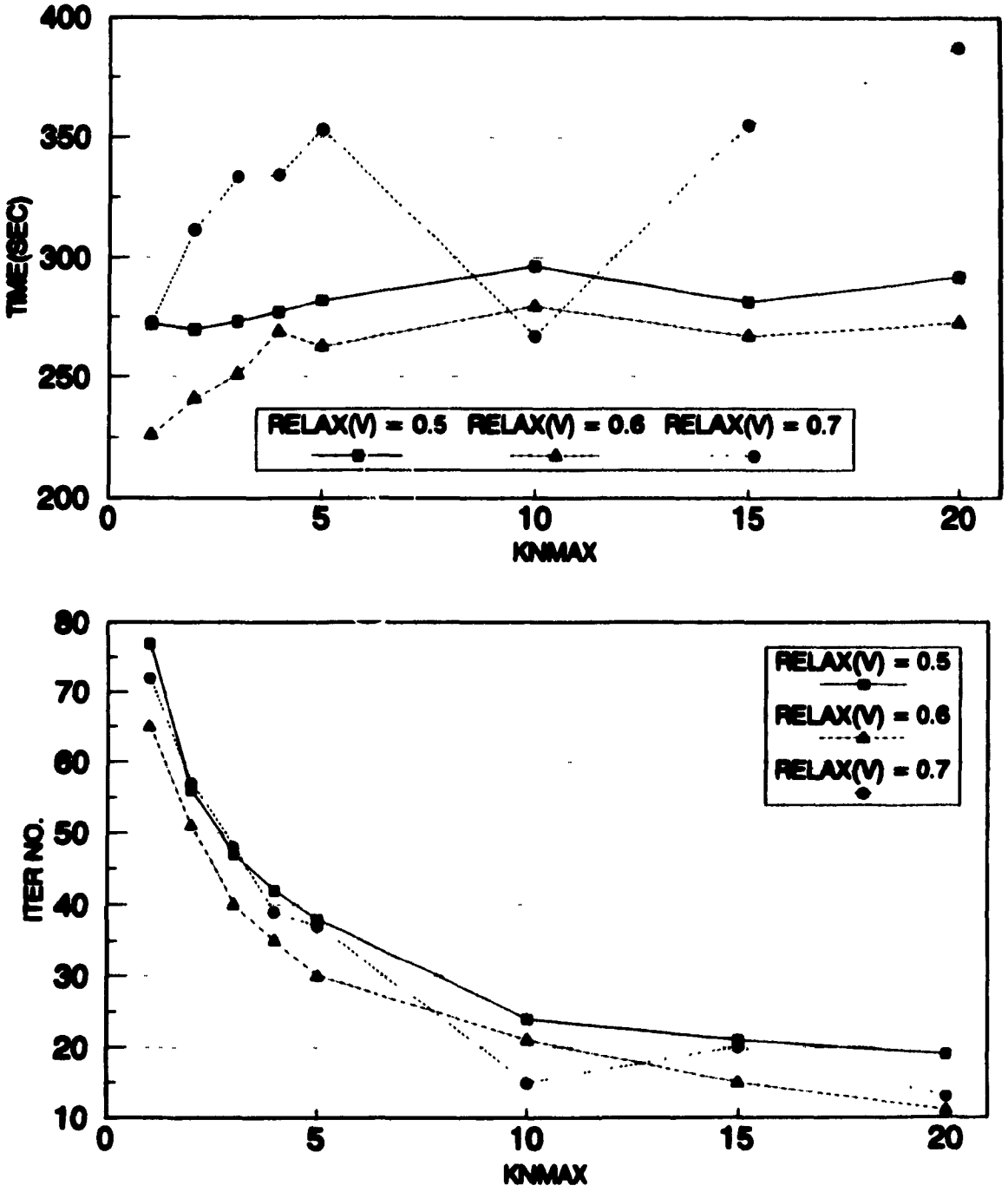


Figure 5.17: Natural Convection in a Square Enclosure: Convergence Behaviour of ESSA: ALPHAO = 1.0; RELAX(T) = 0.9

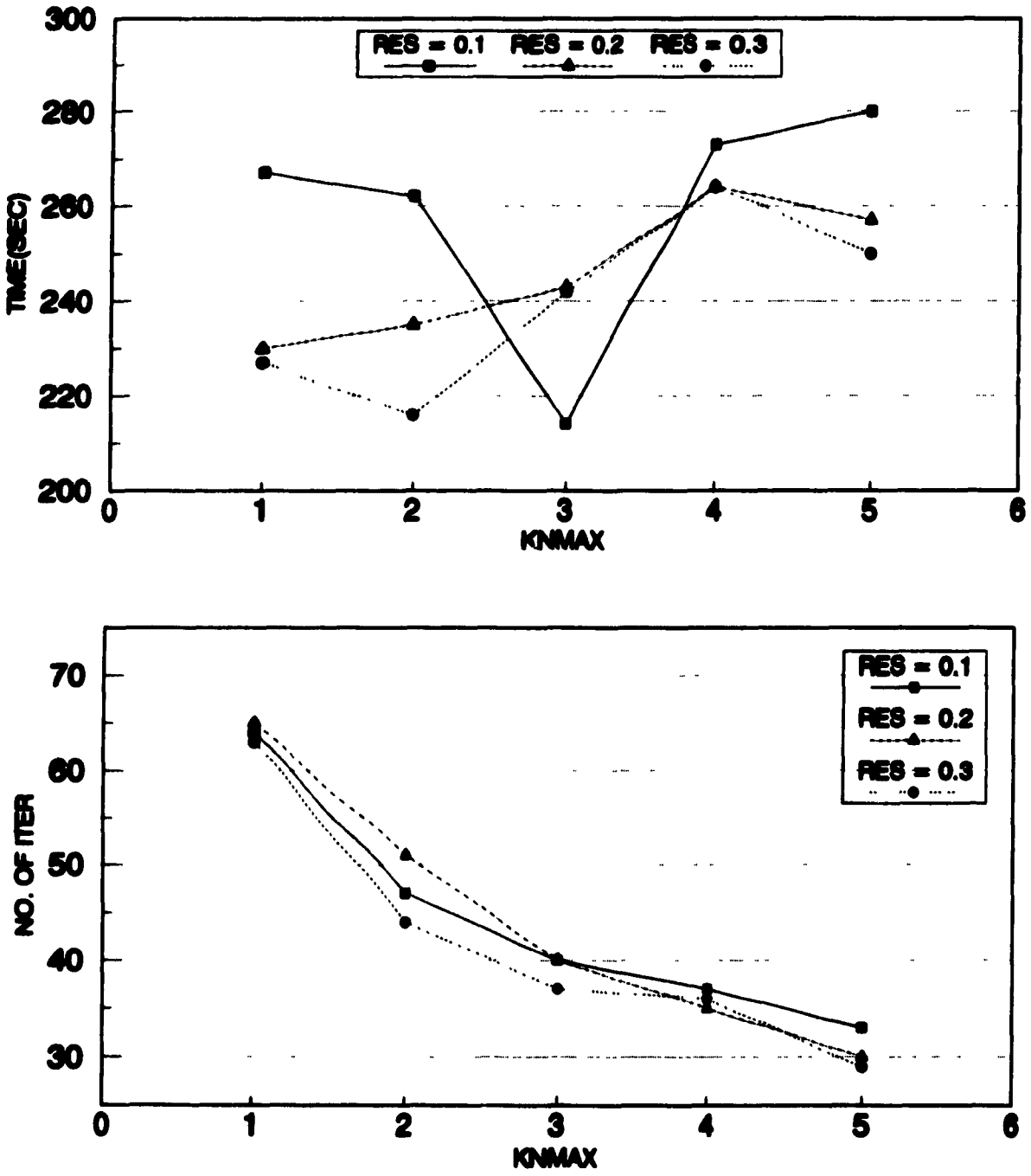


Figure 5.18: Natural Convection in a Square Enclosure: Effect of RES on the Convergence Behaviour of ESSA

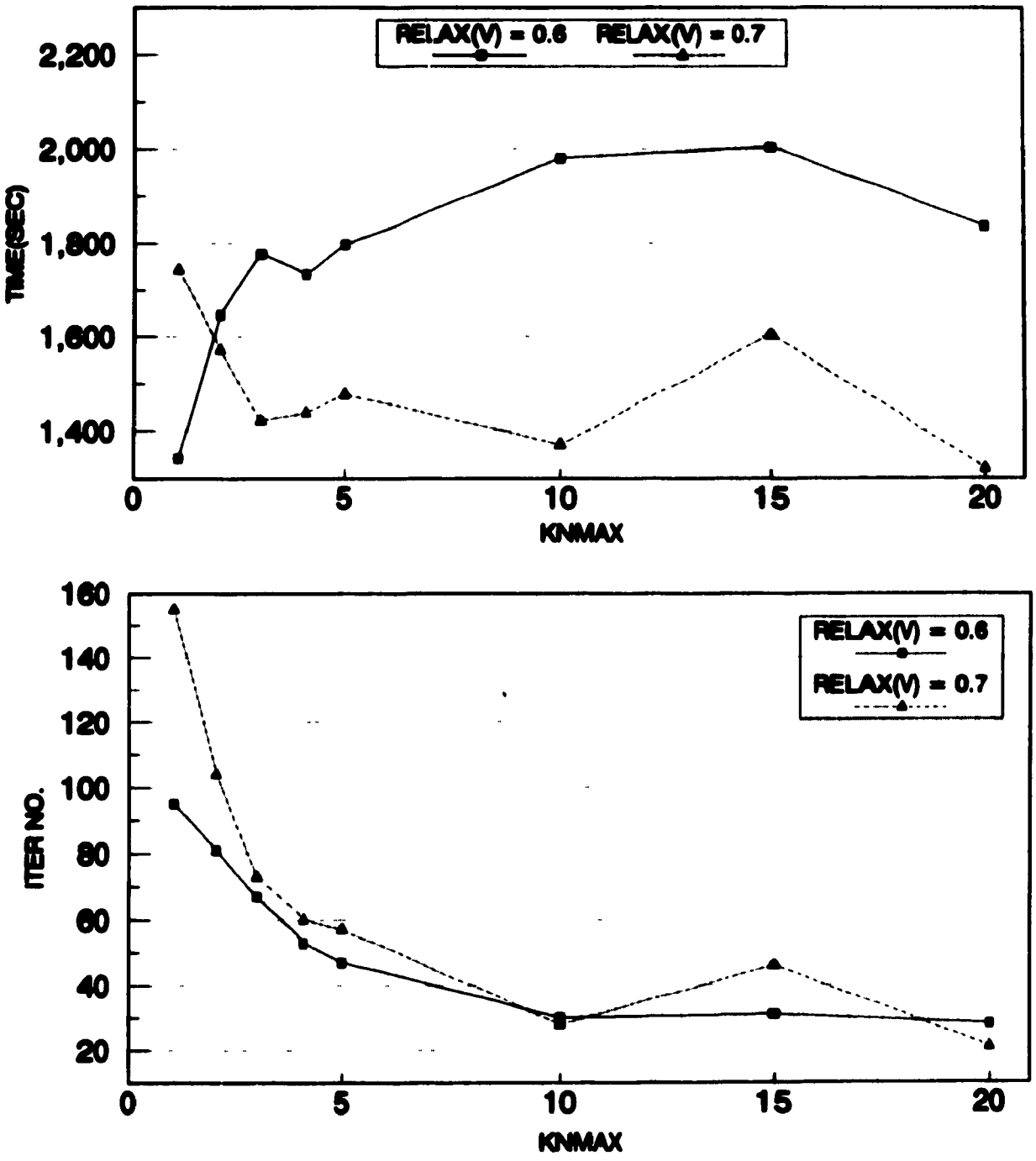


Figure 5.19: Natural Convection in a Square Enclosure: Convergence Behaviour of ESSA on a 51 x 51 Node Grid: ALPHAO = 1.0; RELAX(T) = 0.9

GRID SIZE	SAABAS (t_S)	ESSA (t_E)	t_E / t_S
31 x 31	616.92	214.14	0.35
51 x 51	3756.35	1341.44	0.36
81 x 81	6929.89	5008.87	0.72

(a) EXECUTION TIMES

GRID SIZE	SAABAS (N_S)	ESSA (N_E)	N_E / N_S
31 x 31	192	40	0.21
51 x 51	417	95	0.23
81 x 81	571	70	0.12

(b) NUMBER OF ITERATIONS

Table 5.4: **Natural Convection in a Square Enclosure: Comparison of the Convergence Behaviour of the Saabas and ESSA Schemes for 3 Different Grid Sizes: RELAX(T) = 0.9; ALPHAO = 1**

Chapter 6

A COMPARATIVE EVALUATION OF THE MAW SCHEME

6.1 Introduction

As was discussed in chapter 2, the proper interpolation of the scalar dependent variable, ϕ, in the convection-diffusion terms is still a challenging task in computational fluid dynamics. A MAss Weighted difference scheme (MAW) was proposed by Saabas [100] in the context of CVFEMs for fluid flow, and he showed that it produces acceptable accuracy. But, this scheme has not been compared with other available schemes for interpolation of the scalar dependent variable.

In this chapter, an evaluation of the MAW scheme is conducted by comparing its results with those obtained using the following five schemes:

1. Central Difference Scheme (CDS)

2. Upwind Difference Scheme (UDS)

3. Skew Upstream Difference Scheme (SUDS)

4. Linear Skew Difference scheme (LSD)

5. Quadratic Upstream Interpolation for Convective Kinematics (QUICK)

All of these schemes have already been introduced and discussed in chapter 2.

The well-established test problem of steady, two-dimensional, laminar, incompressible lid-driven fluid flow in a square cavity was used to compare these schemes. To keep the thesis from becoming too long, only the results for tests with uniform grids of 21 x 21, 31 x 31, and 51 x 51 nodes are reported. Three important issues were chosen to be studied in this comparison. These issues are:

1. the accuracy of the results;

2. the execution time required to achieve converged solutions; and

3. the number of iterations needed to obtain converged solutions.

These issues [1] are affected in fluid flow problems by the solution algorithm, under-relaxation parameters used in the solution of the discretized equations, discretization of the domain, and other factors. To be consistent, in each test, for all schemes, the same solution algorithm (ESSA), the same under-relaxation parameters, and the same kind of domain discretization were used.

The problem statement and formulation of this test problem have already been discussed in chapter 5. In this chapter, only the numerical details for this problem and the corresponding results will be discussed.

[1] Items 2 and 3 will be called execution time and number of iterations from now on for simplicity.

6.2 Lid-Driven Flow in a Square Cavity

Numerical Details

Uniform grids of 21 x 21, 31 x 31, and 51 x 51 nodes were used to perform tests with this problem. For each grid, tests were performed for the following values of Reynolds number: Re = 100, 400, and 1000, with each of the six schemes. The ESSA scheme was used as the solution algorithm with KNMAX = 3. No under-relaxation was performed in the outer loop of the ESSA scheme (ALPHAO = 1), and in the inner loop, the under-relaxation parameter in the discretized velocity equations (RELAX(V)) was set equal to 0.7. The convergence criteria for these tests were the same as the convergence criteria which were defined for this problem in chapter 5.

As initial conditions in each test, all dependent variables were set equal to zero at all nodes in the domain, except values of the u-velocity component at the top nodes of the domain (the sliding lid) which were made equal to 1. In some cases, these initial guesses did not produce converged solutions: in such cases only, to check for the accuracy of the results obtained with each scheme, another test was performed in which a solution obtained with the MAW scheme was used as the initial guess.

Results

The u-velocity profiles along the vertical line located at $x^* = 0.5$, obtained using each of the six schemes, and the bench-mark solution of Ghia et al. [39] are plotted for flows with Re = 100, 400, and 1000, and for uniform grids of 21 x 21, 31 x 31, and 51 x 51 nodes in Figs. 6.1 to 6.3, respectively. These plots are used to evaluate the accuracy of these schemes for flows with low to high Reynolds number, with coarse to fine grids. Also, to study the effect of grid refinement on the accuracy of the results obtained with these schemes, for each Reynolds number, the results obtained with each scheme for grids of 21 x 21, 31 x 31, and 51 x 51 nodes are plotted along with

the corresponding bench-mark solution in Figs 6.4 to 6.6. In Fig. 6.6, the results obtained with only four schemes, QUICK, UDS, SUDS, and MAW are presented. The CDS and LSD schemes did not produce a converged solution for Re = 1000.

Tables 6.1 to 6.3 show details of the computational effort for tests with each of these schemes for the above cases. In each table, the computer time and number of iterations (t and N) are reported for the tests done with each of these schemes. Also, to perform a comparison with reference to the computational effort required with the MAW scheme, the ratios of time and number of iterations for tests with these schemes to the corresponding values for the MAW scheme ($t/t_M, N/N_M$) are also shown in these tables.

Discussion

In almost all cases, the results obtained with the QUICK scheme are more accurate than the results obtained with the other schemes. For Re = 1000, however, results obtained with QUICK exhibit wiggles. as shown in Figs. 6.3 and 6.6. At lower Reynolds number, Re = 100 and 400, the results obtained with the CDS scheme are almost similar to the results provided by the QUICK scheme. These two schemes are second-order accurate in terms of Taylor series expansions, and these results show that they are superior in terms of accuracy when compared with the other schemes, which are first order accurate in terms of Taylor series expansions. The CDS scheme, as is well known [77], does not provide converged solution at high Reynolds number, because it generates negative coefficients in the discretized equations. In all tests with Re = 1000, solutions with the CDS scheme diverged. The QUICK scheme also produces negative coefficients at Re = 1000. This causes wiggles in the solutions, but the ESSA converged for this case.

The results of tests with the LSD scheme also show over-shooting and under-shooting. Furthermore, even in tests with low Reynolds number, the LSD solu-

tions exhibit the lowest accuracy in comparison with the results obtained with other schemes. At higher Reynolds number, this scheme leads to divergence in the solution process. This is because of negative coefficients in the discretized equations.

The other three schemes, UDS, SUDS, and MAW, provide almost the same results. The false diffusion inherent in the UDS scheme at high-Reynolds number leads to the less accurate results in comparison with the other two schemes as shown in Fig. 6.3. The MAW scheme is prone to less false diffusion than the UDS scheme, hence it gives better accuracy for high Reynolds number recirculating flows, similar to that in this problem. The results obtained with the SUDS scheme at Re = 1000 with a grid of 21 x 21 nodes are less accurate than the corresponding results obtained with the MAW scheme, and they exhibit oscillations as shown in Figs. 6.3 and 6.6: This is because of negative coefficients in the discretized equations with the SUDS scheme. The MAW scheme provides better accuracy than the SUDS scheme with coarse grids for flows at high Reynolds number, since there are no negative coefficients in the discretized equations produced by the MAW scheme.

Regarding the computational effort for each scheme, the tests with the QUICK and LSD schemes require excessive execution time in comparison to the other schemes. With Re = 100 and 400, tests with the CDS scheme achieved converged solutions faster than those with the MAW scheme for grids of 51 x 51 nodes, but both these schemes require almost the same amount of time for grids of 31 x 31 and 21 x 21 nodes. The number of iterations at convergence for tests with the QUICK scheme are again considerably more than those for the other schemes. The tests with the LSD scheme also require more iterations to converge than the MAW scheme. Tests with the SUDS and UDS schemes converge to the final solution in almost the same number of iterations as the corresponding tests with the MAW scheme. The ratio (N/N_M) for tests with the UDS scheme are always more than 1 (up to 1.19), and this

ratio for the tests with the SUDS scheme is between 0.91 and 1.09.

6.3 Summary

The MAW scheme of Saabas [100] was compared with five other schemes, CDS, UDS, SUDS, LSD, and QUICK, in a simulation of steady, two-dimensional, incompressible, lid-driven fluid flow in a square cavity.

In summary, the results of this test problem show the following:

1. In terms of accuracy and computational effort, the MAW scheme has almost the same performance as UDS and SUDS.

2. The LSD scheme had the poorest performance among the six scheme tested.

3. The CDS and QUICK schemes, when they produce converged solutions, provide the best accuracy.

4. At the highest Reynolds number, Re = 1000, the tests with the CDS diverged, and tests with the QUICK scheme produced solutions with unphysically spatial oscillations (wiggles).

5. Of the six schemes tested, the QUICK scheme was the most expensive in terms of computer (CPU) time.

Thus the MAW scheme compares quite favourably with the five other schemes. It should also be noted that the MAW scheme was originally proposed in the context of control-volume finite element methods (CVFEMs), in the works of Schneider and Raw [103] and Saabas [100]. Thus it is very well suited for applications involving complex irregular-shaped geometries. The other five schemes tested in this work are all formulated in the context of FVMs based on rectilinear Cartesian grids [77,

89, 63, 53]. Their extension to unstructured non-orthogonal grids is by no means a straightforward task, if at all possible.

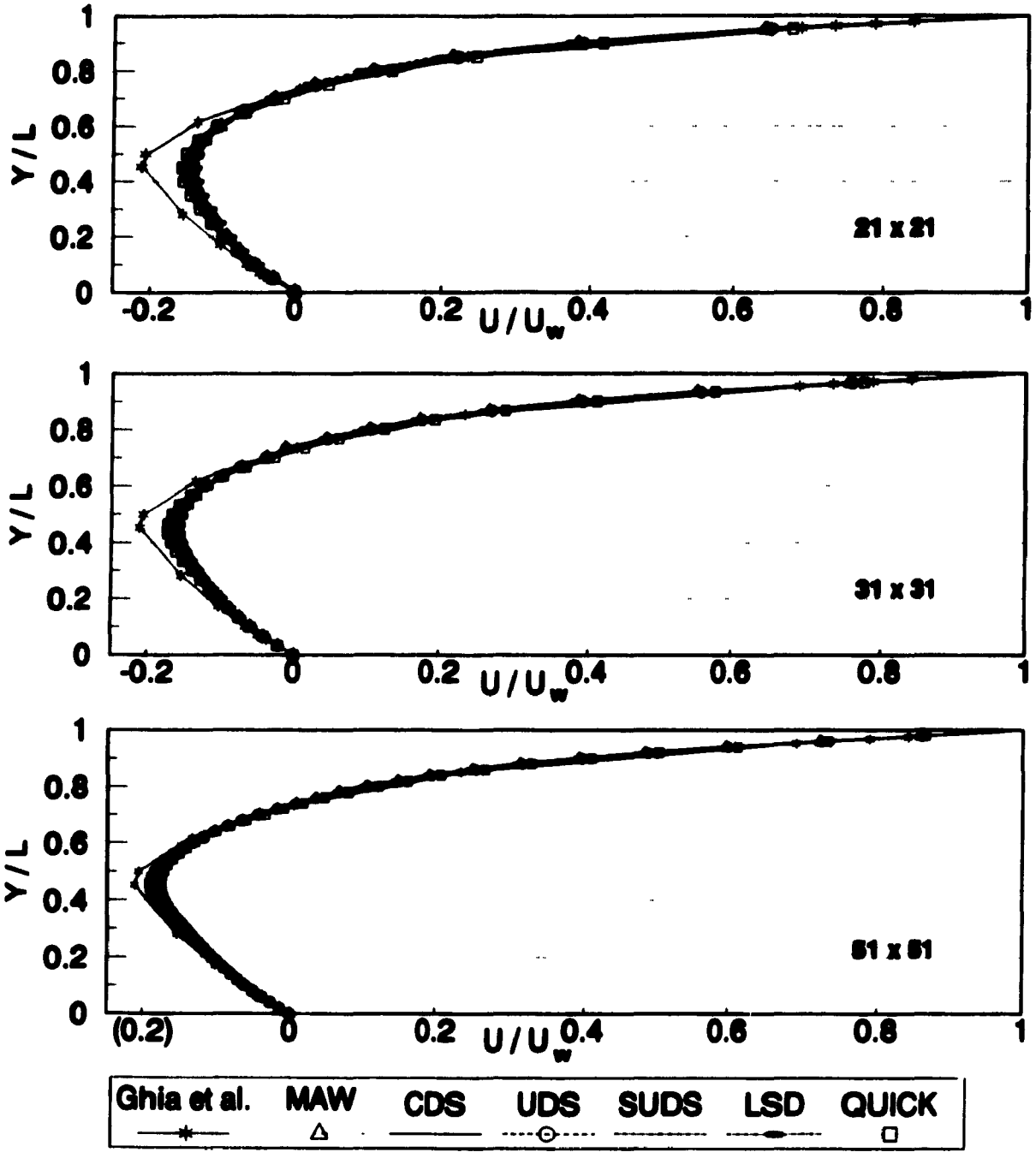


Figure 6.1: **Lid-Driven Flow in A Square Cavity: u-Velocity Profiles at x^* = 0.5: Re = 100**

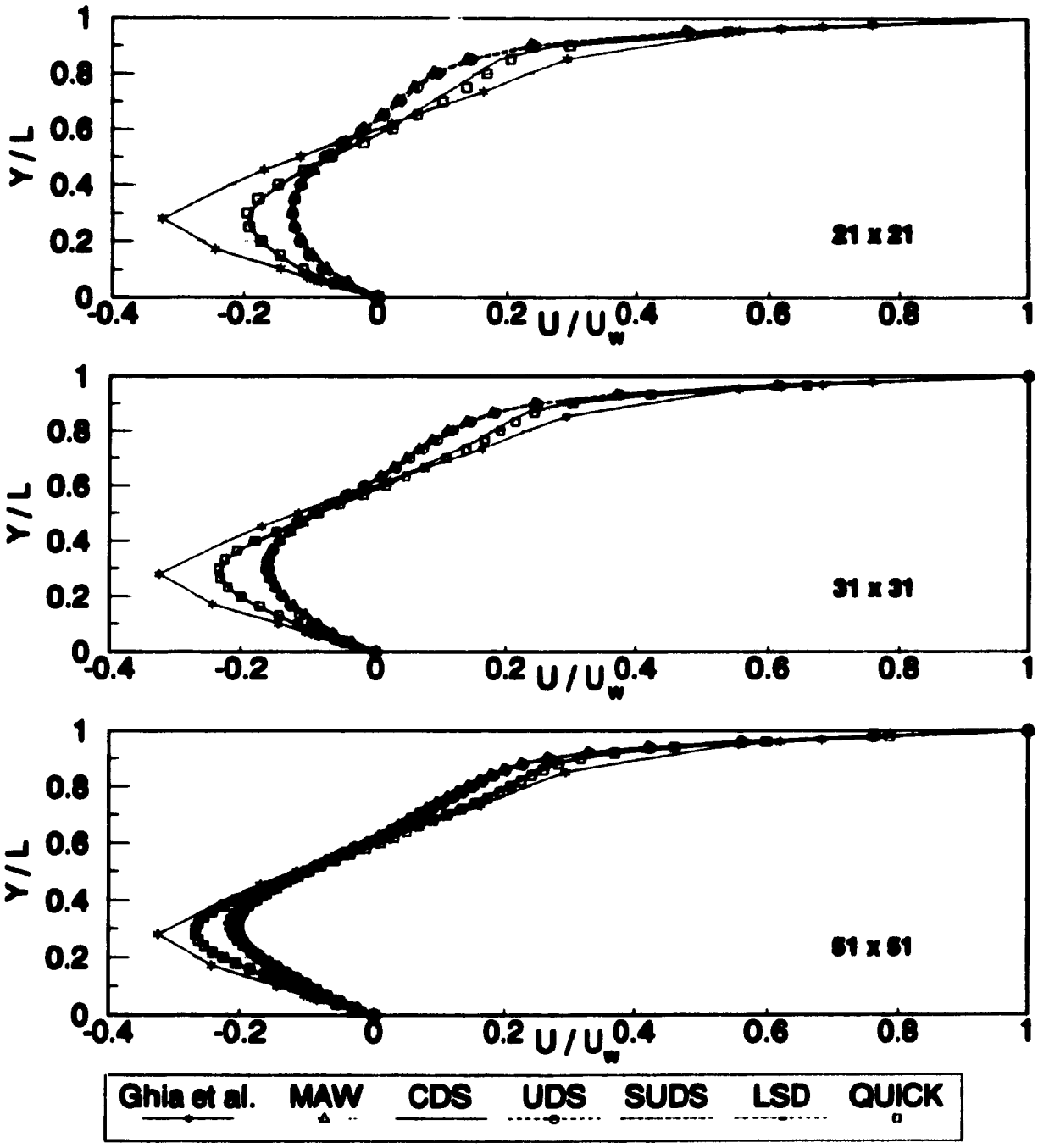


Figure 6.2: Lid-Driven Flow in A Square Cavity: u-Velocity Profiles at x^* = 0.5: Re = 400

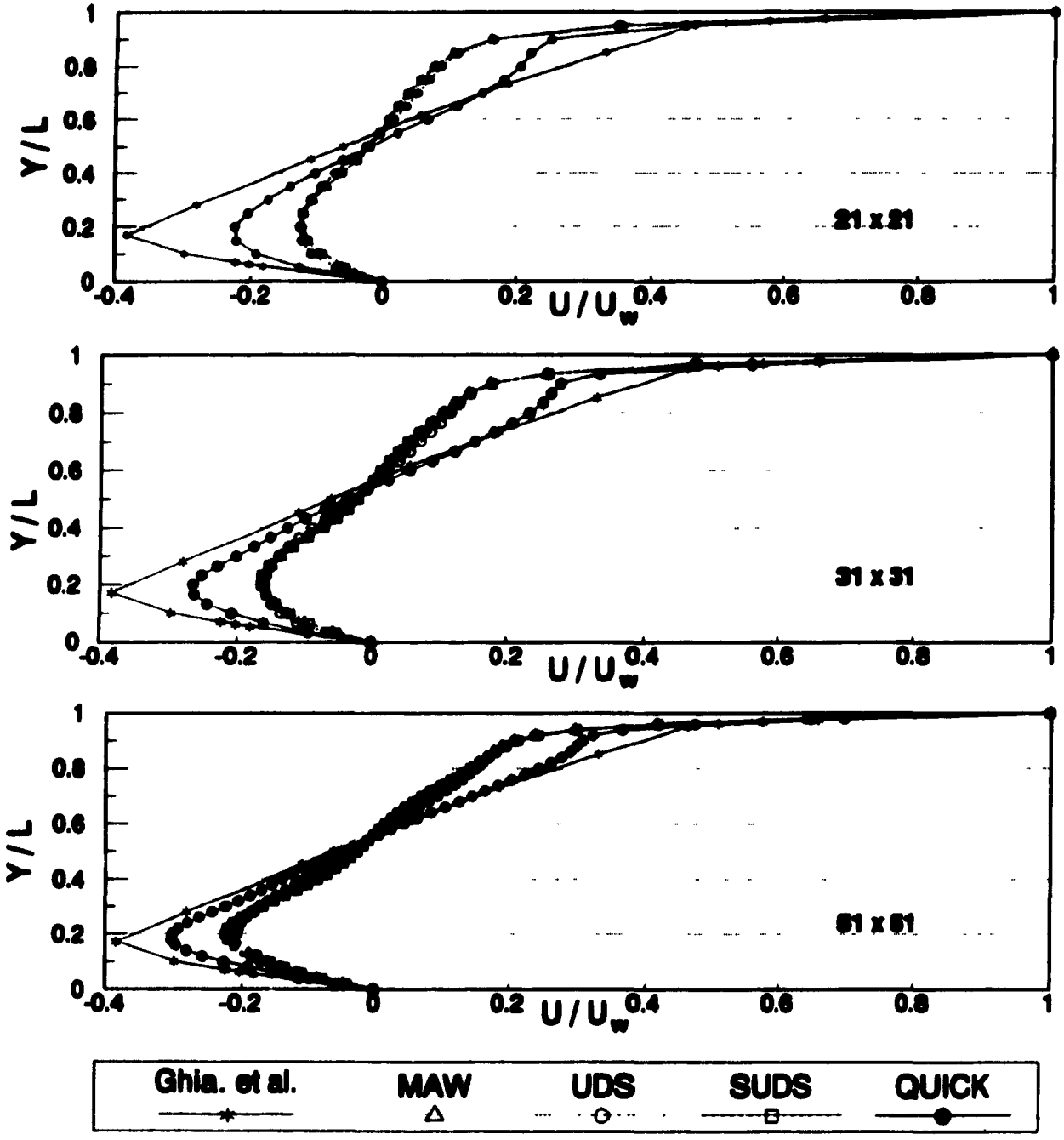


Figure 6.3: Lid-Driven Flow in A Square Cavity: u-Velocity Profiles at x^* = 0.5: Re = 1000

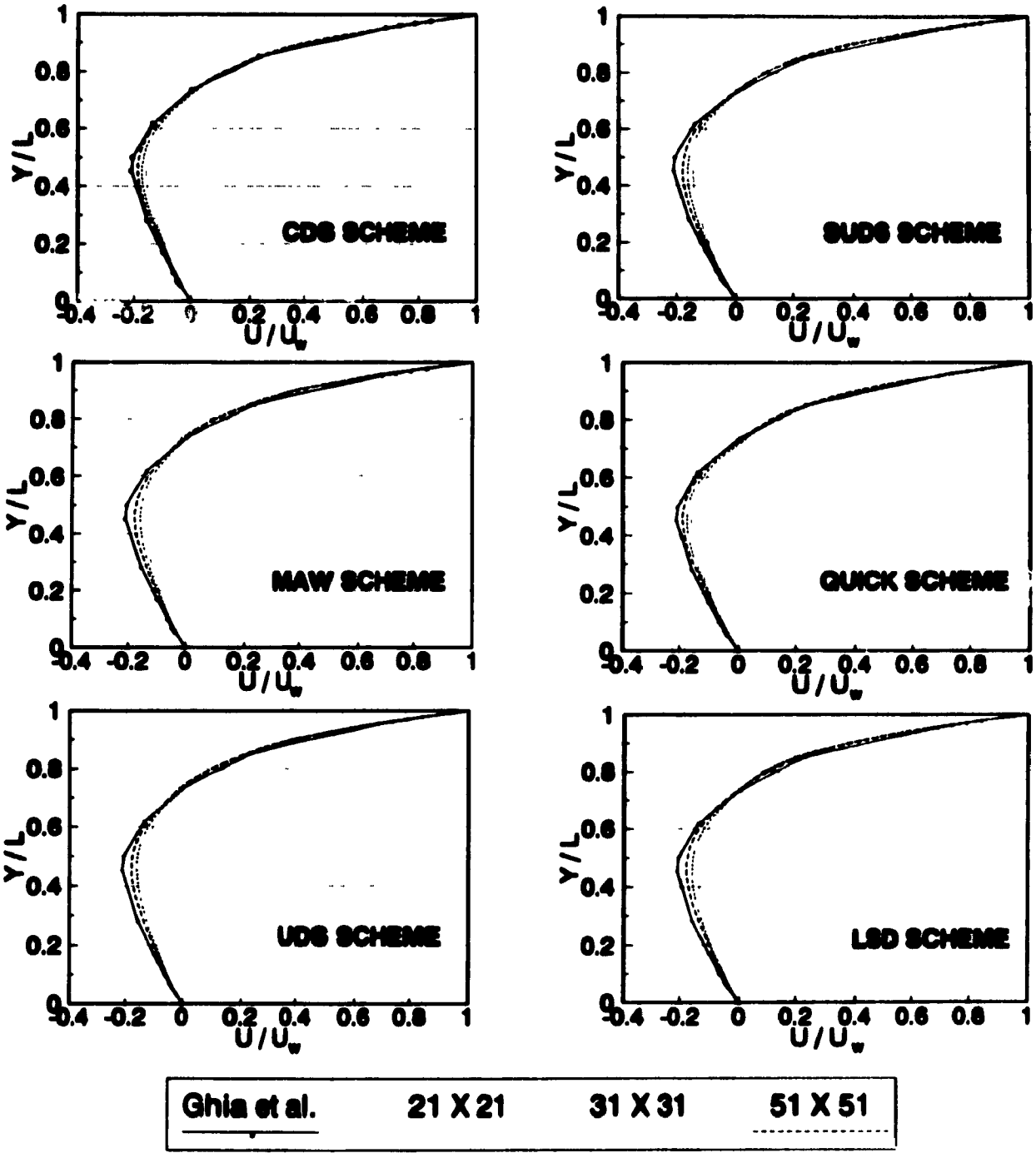


Figure 6.4: Lid-Driven Flow in A Square Cavity: u-Velocity Profiles at $x^* = 0.5$: Effect of Grid Refinement for Re = 100

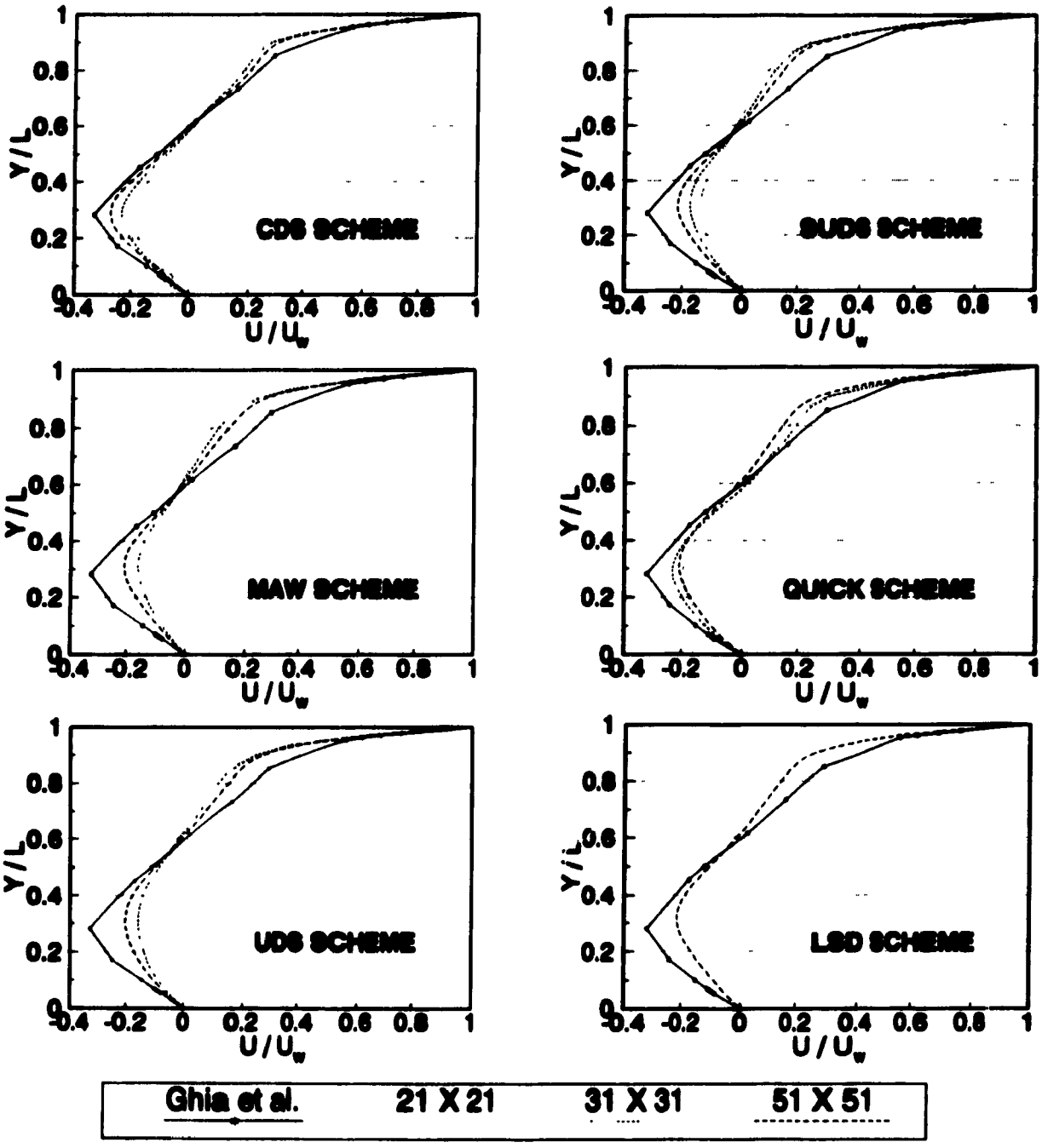


Figure 6.5: Lid-Driven Flow in A Square Cavity: u-Velocity Profiles at x^* = 0.5: Effect of Grid Refinement for Re = 400

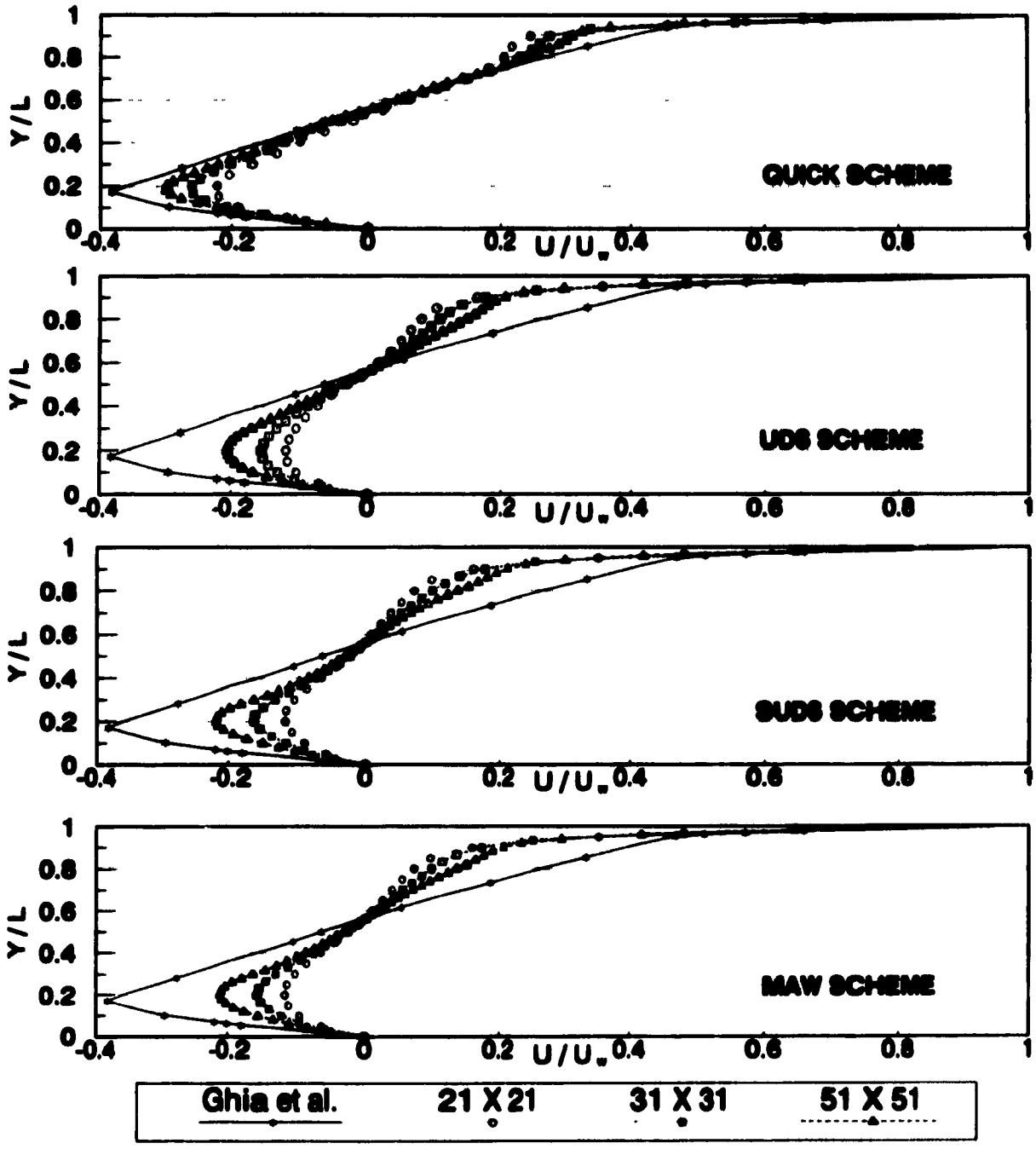


Figure 6.6: Lid-Driven Flow in A Square Cavity: u-Velocity Profiles at $x^* = 0.5$: Effect of Grid Refinement for Re = 1000

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	45.31	33	1.00	1.00
CDS	46.47	37	1.02	1.12
UDS	46.75	36	1.03	1.09
SUDS	42.51	32	0.94	0.97
QUICK	102.87	38	2.27	1.15
LSD	83.81	32	1.85	0.97

RE = 100 21 x 21 UNIFORM GRID

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	44.82	35	1.00	1.00
CDS	43.61	32	0.97	0.91
UDS	47.79	37	1.06	1.06
SUDS	41.86	37	0.93	1.06
QUICK	96.73	39	2.16	1.11
LSD	DIVERGED			

RE = 400 21 x 21 UNIFORM GRID

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	35.98	29	1.00	1.00
CDS	DIVERGED			
UDS	40.70	31	1.13	1.07
SUDS	48.51	28	1.35	0.96
QUICK	283.86	106	7.89	3.65
LSD	DIVERGED			

RE = 1000 21 x 21 UNIFORM GRID

Table 6.1: **Lid-Driven Flow in A Square Cavity: Details of Computational Effort for Tests with the Six Schemes: Uniform 21 x 21 Node Grid**

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	215.33	47	1.00	1.00
CDS	282.59	65	1.31	1.38
UDS	241.67	55	1.12	1.17
SUDS	217.45	48	1.01	1.02
QUICK	525.25	73	2.44	1.55
LSD	592.04	71	2.75	1.51

RE = 100 31 x 31 UNIFORM GRID

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	263.31	63	1.00	1.00
CDS	191.14	54	0.73	0.86
UDS	298.30	70	1.13	1.11
SUDS	262.38	69	1.00	1.09
QUICK	556.08	73	2.11	1.16
LSD	DIVERGED			

RE = 400 31 x 31 UNIFORM GRID

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	196.35	56	1.00	1.00
CDS	DIV	ERG		
UDS	252.93	62	1.30	1.11
SUDS	193.94	51	0.99	0.91
QUICK	648.78	104	3.30	1.86
LSD	DIVERGED			

RE = 1000 31 x 31 UNIFORM GRID

Table 6.2: Lid-Driven Flow in A Square Cavity: Details of Computational Effort for Tests with the Six Schemes: Uniform 31 x 31 Node Grid

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	2654.33	153	1.00	1.00
CDS	2413.81	154	0.91	1.01
UDS	2816.75	180	1.06	1.17
SUDS	2525.75	139	0.95	0.91
QUICK	3844.57	154	1.45	1.01
LSD	3563.79	115	1.34	0.75

RE = 100 51 x 51 UNIFORM GRID

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	2530.96	143	1.00	1.00
CDS	2035.76	120	0.80	0.84
UDS	2822.46	158	1.11	1.10
SUDS	2497.35	151	0.98	1.06
QUICK	3721.09	141	1.47	0.99
LSD	9520.34	314	3.76	2.19

RE = 400 51 x 51 UNIFORM GRID

SCHEME	TIME (SEC)	ITER #	t / t_M	N / N_M
MAW	1559.39	113	1.00	1.00
CDS	DIVERGED			
UDS	2083.55	135	1.33	1.19
SUDS	1756.34	116	1.13	1.03
QUICK	4963.62	191	3.18	1.69
LSD	DIVERGED			

RE = 1000 51 x 51 UNIFORM GRID

Table 6.3: Lid-Driven Flow in A Square Cavity: Details of Computational Effort for Tests with the Six Schemes: Uniform 51 x 51 Node Grid

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

This chapter is divided into two main parts. In the first part, the thesis and its contributions are reviewed. This part has two sections: in the first section, the ESSA and SEVA schemes will be reviewed; then the performance of the MAW scheme will be discussed. In the second part, some recommendations for extending the current work will be presented.

7.1 Contributions of the Thesis

7.1.1 Review of the Enhancements to Sequential Solution Algorithms

The main contribution of this work is the development of two new enhancements to sequential solution algorithms for solution of the u, v, and p discretization equations. These two new enhancements have been labelled as the Enhanced Sequential Solution Algorithm (ESSA), and the SEquential Variable Adjustment (SEVA) algorithm. Both

SEVA and ESSA were developed during efforts to enhance the performance of a iterative sequential solution algorithm proposed by Saabas [100].

In SEVA, in contrast to the Saabas scheme [100], the sets of u- and v-velocity discretization equations are solved sequentially, before the set of discretized pressure equations are solved. Hence, it is ***not*** necessary to store the coefficients of the u- and v-velocity discretization equations (except for the boundary nodes, and a_P and constant term for all nodes) as in the Saabas scheme [100]. This enhancement reduces the computer storage requirements of the Saabas scheme considerably. The results of tests with the SEVA scheme on the problem of lid-driven flow in a square cavity , presented in chapter 5, showed that it produces converged solutions in computer times not more than those needed in the corresponding tests with the Saabas scheme.

The ESSA scheme adds an *inner loop* to available sequential solution algorithms akin to SIMPLE [77, 122, 100]. In this work, this enhancement was incorporated in the Saabas scheme [100]. The ESSA scheme takes advantage of ideas contained in the direct and semi-direct solvers [32, 50, 68], to improve the rate of convergence of the Saabas scheme. In ESSA, the coefficients of the u, v, and p discretization equations are calculated and stored. In the outer loop, these discretization equations are solved as in the Saabas scheme. In the inner loop, repeated solutions of the u, v, and p discretization equations are done sequentially, without changing the coefficients in these equations. Once the inner loop iterations are completed, the coefficients are recalculated, and the complete procedure are repeated until the convergence criteria is satisfied.

The convergence behaviour of the Saabas, SEVA, and ESSA schemes were compared for three test problems: lid-driven flow in a square cavity, flow over a backward-facing step, and natural convection in a square enclosure. The results of these show that ESSA produces converged solutions significantly faster than the Saabas and

SEVA schemes. The computer times are decreased in the tests with the ESSA scheme up to 50% in comparison to the corresponding tests with the Saabas scheme. Also, the overall number of iterations exhibits a dramatic decrease: up to 70% reduction in the required number of iterations was achieved in the tests with the ESSA scheme in comparison to corresponding tests with the Saabas scheme.

It should also be noted that enhancements similar to ESSA can be easily added to sequential solution algorithms akin to SIMPLE, SIMPLER, and SIMPLEC, to improve the convergence rate of these algorithms. The ESSA scheme retains the advantages of SIMPLE-like algorithms, namely, simplicity, ease of implementation, generality, and robustness. Furthermore, in contrast to CELS [32, 50], the ESSA scheme does not require complex manipulations of coefficients in the discretized equations, and it can be easily extended to solve sets of discretization equations for additional dependent variables that may be coupled to the velocity components and pressure.

The ESSA scheme is also quite robust, and it can be successfully applied to problems with outflow boundaries.

7.1.2 Review of the Performance of the MAW scheme

The MAss Weighted skew upwind scheme (MAW) studied in this work is similar to that proposed by Saabas [100] in the context of CVFEM. The performance of this scheme was evaluated by comparing its results with those obtained using five other schemes: CDS, UDS, SUDS, QUICK, and LSD. All of these schemes were discussed in chapter 2, and the results of this comparison were presented in chapter 6. Three important issues were studied in this comparison: (i) accuracy of the results; (ii) execution time required to achieve converged solutions; and (iii) number of iterations to obtain converged solutions.

This comparative evaluation was done using the co-located FVM, described in

chapters 2 and 3, and ESSA for the solution of the discretized equations. Similar comparative evaluations in the literature, of CDS, UDS, SUDS, QUICK and LSD, were done with FVMs based on staggered grids for u, v, and p [43, 111, 79, 74]. The MAW scheme displayed almost the same performance as UDS and SUDS in terms of accuracy and computational effort. The CDS and QUICK schemes, when they produce converged solutions, provide the best accuracy, and LSD displayed the poorest performance among the six schemes tested. The QUICK scheme was the most expensive in terms of computer time. These results for the five schemes other than MAWs are similar to results obtained with FVMs based on staggered grids [43, 111, 79, 74].

7.2 Recommendations

Although the results obtained with the ESSA scheme are very encouraging, there are several areas that seems appropriate and interesting for future studies. In this work, ESSA was used to enhance the Saabas scheme [100], and it was utilized to solve problems involving steady, two-dimensional, laminar flow and heat transfer. These problems involved incompressible Newtonian fluids, and rectangular calculation domains that were discretized using structured line-by-line grids. Structured line-by-line co-located grids were used to solve these problems. It would be relatively easy and useful to incorporate ESSA in FVMs based on staggered and non-staggered grids. It can also be extended to co-located equal-order CVFEMs, such as the one proposed by Saabas [100]. It would also be interesting to apply the ESSA to complex flow problems similar those solved using staggered-grid FVMs over the last two decades.

The following specific extensions of the work in this thesis are recommended:

1. Incorporation of ESSA in iterative sequential solution algorithms akin to SIMPLE [77].

2. **Utilization of the proposed FVM and ESSA to solve three-dimensional problems that involve turbulent flow and heat transfer, multiphase flows, or combustion. Methods akin to SIMPLE converge rather slowly in such complex problems, so it would be interesting to study and record potential benefits that ESSA could provide.**

3. **Utilization of ESSA in FVMs based on: staggered-grid arrangements; curvilinear orthogonal grids; and staggered and non-staggered curvilinear nonorthogonal coordinate systems.**

4. **Incorporation and evaluation of ESSA in co-located equal-order CVFEMs for fluid flow and heat transfer [10, 100].**

The MAW scheme provides solutions with accuracy comparable to those of UDS [77] and SUDS [89]. However, when higher-order schemes such as CDS and QUICK provide converged solutions, their accuracy is superior to the results of the MAW scheme. Furthermore, Saabas [100] found that the flow-oriented upwind scheme (FLO) of Baliga and Patankar [7], when it gives converged solutions, is more accurate than the MAW scheme. It appears, therefore, that a hybrid scheme that switches smoothly from MAW to CDS in FVMs, and from MAW to FLO in CVFEMs, would be very useful. The development of such a hybrid scheme would be worthwhile.

Bibliography

[1] Achraya, S., Moulkalled, F.H., Improvements to Incompressible Flow Calculation on a Non-Staggered Grid, *Numerical Heat Transfer, Section B*, Vol. 15, No. 2. pp 131-152, 1989.

[2] Achraya, S. , Patankar, S.V., Use of an Adaptive Grid Procedure for Parabolic Flow Problems, *Int. J. of Heat Mass Transfer*, Vol. 28, pp. 1057-1066, 1989.

[3] Ames, W.F., Numerical Methods for Partial Differential Equations, 2nd Academic Press, NY, 1977.

[4] Aziz, K. Hellums, J.D., Numerical Solution of the Three-Dimensional Equations of Motion for Laminar Natural Convection, *Phys. of Fluids*, Vol. 10, p. 314, 1967.

[5] Baker, A.J., *Computation of Fluid Flow by the Finite Element Method*, Hemisphere Publishing Corporation, Washington, 1983.

[6] Baliga, B.R., *A Control-Volume Based Finite Element Method for Convective Heat and Mass Transfer*, Ph.D. Thesis, Dept. of Mech. Eng., University of Minnesota, 1978.

[7] Baliga, B.R., Patankar, S.V., A New Finite Element Formulation for Convection Diffusion Problems, *Numerical Heat Transfer*, Vol. 3, pp. 393-409, 1980.

[8] Baliga, B.R., Patankar, S.V., A Control Volume Finite Element Method for Two-Dimensional Fluid Flow and Heat Transfer, *Numerical Heat Transfer*, Vol. 6, pp. 245-261, 1983.

[9] Baliga, B.R., Patankar, S.V., Elliptic Systems: Finite Element Method II, in Minkowycz, W.J., Sparrow, E.M., Schneider, G.E., and Pletcher, R.H. (eds.), *Handbook of Numerical Heat Transfer*, Chap. 10, pp. 421-461, Wiley, New York, 1988.

[10] Baliga, B.R., Saabas, H.J., Control Volume Finite Element Method for Incompressible Fluid Flow, *3rd Portuguese Conf. on Computational Mechanics*, Vol 1, pp. c2.1-c2.20, Sept. 28-30, Coimbra, Portugal, 1992.

[11] Barakat, H.Z., Clark, J.A., Analytical and Experimental Study of Transient Laminar Natural Convection Flows in Partially Filled Containers, *3rd International Heat Transfer Conf.*, Vol. II, paper 57, p. 152, Chicago, 1966.

[12] Blottner, F.G., Investigation of Some Finite-Difference Techniques for Solving the Boundary Layer Equations, *Comp. Meth. App. Mech. and Eng.* , Vol. 6, pp. 1-30, 1975.

[13] Blue, J.L., Robust Methods for Solving Systems of Nonlinear Equations, *SIAM J. Sci. Stat. Comp.* ,Vol. 1. No. 1., pp. 22-33, 1980.

[14] Bozeman, J.D., Dalton, C., Numerical Study of Viscous Flow in a Cavity, *J. Comp. Phys.*, Vol. 12, pp. 348-363, 1973.

[15] Braaten, M.E., Development and Evaluation of Iterative and Direct Methods for the Solution of the Equations Governing Recirculating Flows PhD. thesis, Un ersity of Minnesota, 1985.

[16] Braga, W., On the Use of Some Weighted Upwind Scheme for Strongly Convective Flows, *Comp. Numerical Heat Transfer, part B*, Vol. 18, No. 1, pp. 43-60, 1990.

[17] Brandt, A., Multi-Level Adaptive Solutions to Boundary Value Problems, *Math. Comp.*, Vol. 31, pp. 333-390, 1980.

[18] Brooks, A.N., Hughes, T.J.R., Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows With Particular Emphasis on the Incompressible Navier-Stokes Equations, *Comp. Meth. in App. Mech. and Eng.*, Vol. 32, pp. 199-259, 1982.

[19] Burggraf, O.R., Analytical and Numerical Studies of the Structure of Steady Separated Flows, *J. Fluid Mech.*, Vol. 24, pp. 113-151, 1966.

[20] Caretto, L.S., Curr, R.M., Spalding, D.B., Two Numerical Methods for Three-Dimensional Boundary Layers, *Comp. Meth. in App. Mech. and Eng.*, Vol. 1, pp. 39-57, 1972.

[21] Chu g, T.J., *Finite Element Analysis of Fluid Dynamics*, McGraw Hill Book Co., Toronto 1978.

[22] Coelho, P.G., Pereira, J.C.F., Finite-Volume Computation of the Turbulent Flow Over a Hill Employing 2D and 3D Non-orthogonal Co-Located Grid Systems, *Int. J. of Num. Meth. in Fluids*, Vol. 14, pp. 423-441, 1992.

[23] Courant, R., Isaacson, E., Rees, M., On the Solution of Non-Linear Hyperbolic Differential Equations by Finite Differences, *Comm. Pure Appl. Math.*, Vol. 5, p. 243, 1952.

[24] De Vahl Davis, G., Natural Convection of Air in a Square Cavity: A Benchmark Numerical Solution, *Int. J. for Num. Meth. in Fluids*, Vol. 3, pp. 249-264, 1983.

[25] Demirdzic, I.A., *A Finite Volume Method for Computation of Fluid Flow in Complex Ducts*, Ph.D thesis submitted to the University of London, London, England, 1982.

[26] Demuren, A.O., False Diffusion in Three-Dimensional Flow Calculations, *Computers and Fluids*, Vol. 13, pp. 411-419, 1985.

[27] Dietrich, D.E., McDonald, B.E. Warn-Varnas, H.J. Optimized Block Implicit Relaxation, *J. Comp. Phys.*, Vol. 18, pp. 421-439, 1975.

[28] Finlayson, B.A., Scriven, L.E., The Method of Weighted Residuals - A review, *Appl. Mech., Rev.*, Vol. 19, pp. 735-748, 1966.

[29] Fromm, J.E., Harlow, F.H., Numerical solution of the Problem of Vortex Street Development, *Phys. of Fluids*, Vol. 6, p. 975, 1963.

[30] Forsythe G.E., Wasow, W., *Finite Difference Methods for Partial Differential Equations*, Wiley, NY, 1968.

[31] Galpin, P.F., *Solution of the Non-Linear Algebraic Equations in the Numerical Prediction of Incompressible Fluid Flow and Heat Transfer*, M.A.Sc thesis submitted to the University of Waterloo, Waterloo, Canada, 1985.

[32] Galpin, P.F., Van Doormaal J.P., Raithby, G.D., Solution of Incompressible Mass and Momentum Equations by Application of a Coupled Equation Line Solver, *Int. J. for Num. Meth. in Fluids*, Vol. 5, pp. 615-625, 1985.

[33] Galpin, P.F., Raithby, G.D., Van Doormaal J.P., Discussion of Upstream weighted Advection for Curved Grids, *Numerical Heat Transfer*, Vol. 9, pp. 241-246, 1986.

[34] Galpin, P.F., Raithby, G.D., Treatment of Nonlinearities in the Numerical Solution of the Incompressible Navier-Stokes Equations, *Int. J. for Num. Meth. in Fluids*, Vol. 6, pp. 409-426, 1986.

[35] Galpin, P.F., Raithby, G.D., Numerical Solution of Problems in Incompressible Fluid Flow: Treatment of the Temperature-Velocity Coupling, *Numerical Heat Transfer*, Vol. 10, No. 2, pp. 105-129, 1986.

[36] Galpin, P.F., Huget, R.G., Raithby, G.D., Fluid Flow Simulation in Complex Geometries, *CNS/ANS 2nd. Int. Conf. on Simulation Meth. in Nuclear Eng.* 1986.

[37] Gartling, D.K., A Test Problem for Outflow Boundary Conditions-Flow over a Backward-Facing Step, *Int. J. of Num. Meth. in Fluids*, Vol. 11, pp. 953-967, 1990.

[38] Gentry, R.A., Martin, R.E. Daly, B.J., An Eulerian Differencing Method for Unsteady Compressible Flow Problems, *J. of Comp. Phys.*, Vol. 1, p. 87, 1966.

[39] Ghia, U., Ghia, K.N., Shin, C.T., High Re Solutions for Incompressible Flow Using Navier-Stokes Equations, and a Multi-Grid Method, *J. Comp. Phys*, Vol. 48, pp. 387-411, 1982.

[40] Gosman, A.D. Pun, W.M., Runchal, A.K., Spalding, D.B. Wolfshtein, M., *Heat and Mass Transfer in Recirculating Flows*, Academic Press Ltd., New York, 1969.

[41] Gresho, P., Lee, R.L., Don't Suppress the Wiggles- They're Telling You Something, *Computers and Fluids*, Vol. 4, pp. 345-357, 1981.

[42] Hackman, L.P., Raithby, G.D., Strong, A.B., Numerical Predictions of Flows over Backward-Facing Steps, *Int. J. for Num. Meth. in Fluids*, Vol. 4, pp. 711-724, 1984.

[43] Han, T., Humphery, J.A.C., Launder, B.E., A Comparison of Hybrid and Quadratic-Upstream Differencing in High Reynolds Number Elliptic Flows, *Comp. Meth. in App. Mech. and Eng.*, Vol. 29, pp. 81-95, 1981.

[44] Harlow, F.H., Welch, J.E., Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Phys. of Fluids*, Vol. 8, p. 2182, 1965.

[45] Hassan, Y.A., Rice, J.G., Kim, J.H., A Stable Mass-Flow-Weighted Two-Dimensional Skew Upwind Scheme, *Numerical Heat Transfer*, Vol. 6, pp. 395-408, 1983.

[46] Hockney, R.W., The Potential Calculations and Some Applications, *Meth. Comp. Phys.*, Vol. 9, pp 135-211, 1970.

[47] Hookey, N.A., *Evaluation and Enhancements of Control Volume Finite-Element Methods for Two-Dimensional Fluid Flow and Heat Transfer*, M.Eng. thesis, McGill University, Montreal, Que., 1986.

[48] Hookey, N.A., Baliga, B.R., Prakash, C., Evaluation and Enhancements of Some Control Volume Finite-Element Methods - part 1: Convection-Diffusion Problems, *Numerical Heat Transfer*, Vol. 14, pp. 255-272, 1988.

[49] Hookey,N.A.,Baliga, B.R., Evaluation and Enhancements of Some Control Volume Finite-Element Methods: Part II - Fluid Flow Problems, *Numerical Heat Transfer*, Vol. 14, pp. 273-293, 1988.

[50] Hookey, N.A., *A CVFEM for Two Dimensional Viscous Compressible Fluid Flow*, Ph.D thesis, McGill University, 1989.

[51] Hsu, C., *A Curvilinear-Coordinate Method for Momentum, Heat and Mass Transfer in Domains of Irregular Geometry*, Ph.D. Thesis, Dept. of Mech. Eng., University of Minnesota, 1981.

[52] Huang, P.G., Launder, B.E., Leschziner, M.A., Discretization of Nonlinear Convection Processes: A Broad-Range Comparison of Four Schemes, *Comp. Meth. in App. Mech. and Eng.*, Vol. 48, pp. 1-24, 1985.

[53] Huget, R.G., *The Evaluation and Development of Approximation Schemes for the Finite Volume Method*, Ph.D thesis submitted to the University of Waterloo, Waterloo, Canada, 1985.

[54] Hutchinson, B.R., Raithby, G.D., A Multigrid Method Based on the Additive Correction Strategy, *Numerical Heat Transfer*, Vol. 9, pp. 511-537, 1986.

[55] Incropera, F.P. DeWitt, D.P., *Fundamentals of Heat and Mass Transfer*, John Wiley and Sons, New York, 1990.

[56] Karki, K.C., *A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries*, Ph.D. thesis, Univ. of Minnesota, Minneapolis, Minn., 1986.

[57] Karki, K.C., Patankar, S.V., Three-Dimensional Fluid Flow Calculations Using a Flux-Spline Method, *AIAA Journal*, Vol. 28. No. 4, pp. 631-634, 1990.

[58] Screiber, R., Keller, H.B., Driven Cavity Flows by Efficient Numerical Techniques, *J. Comp. Phys.*, Vol. 49, pp. 310-333, 1983.

[59] Kobayashi, M.H., Pereira, J.C.F., Calculation of Incompressible Laminar Flows on a Nonstaggered, Nonorthogonal Grid, *Num. Heat Transfer, part B*, Vol. 19, pp. 243-262, 1991.

[60] Latimer, B.R., Pollard, A., Comparison of Pressure-Velocity Coupling Solution Algorithms, *Numerical Heat Transfer*, Vol. 8, pp. 635-652, 1985.

[61] LeBail, R.C., Use of Fast Fourier Transforms for Solving Partial Differential Equations in Physics, *J. Comp. Phsy.*, Vol. 9, 1972.

[62] LeDain-Muir. B., Baliga, B.R., Solution of Three Dimensional Convection Problems Using Tetrahedral Elements and Flow Oriented Upwind Interpolation Functions, *Numerical Heat Trans*, Vol. 9, pp. 143-162, 1986.

[63] Leonard, B.P., A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, *Comp. Meth. in App. Mech. and Eng.*, Vol. 19, pp. 59-98, 1979.

[64] Leonard, B.P., Locally Modified QUICK Scheme for Highly Convective Two-Dimensional and Three-Dimensional Flows, *Procc. 5th. Int. Conf.* , Montreal, pp. 35-47, 1987.

[65] Leonard, B.P., Simple High-Accuracy Resolution Program for Convective Modelling of Discontinuities, *Int. J. for Num. Meth. in Fluids*, Vol. 8, pp. 1291-1318, 1988.

[66] Levenberg, K., A Method for the Solution of Certain Nonlinear Problems in Least Squares, *Quart. Appl. Math*, Vol. 2, 1944.

[67] Lillington, J.N., A Vector Upstream Differencing Scheme for Problems in Fluid Flow Involving Significant Source Terms in Steady-State Linear Systems, *Int. J. for Num. Meth. in Fluids*, Vol. 1, pp. 3-16, 1981.

[68] Macarthur, J.W., *Development and Implementation of Robust Direct Finite-Difference Method For the Solution of Strongly Coupled Elliptic Transport*, Ph.D. thesis, Univ. of Minnesota, Minneapolis, Minn., 1986.

[69] Marquardt, D.W., An Algorithm for Least Sequence Estimation of Nonlinear Parameters, *J. Soc. Indus. Appl. Math.*, Vol. 11, 1963.

[70] McCormick, S.F., Multilevel Adaptive Methods for Partial Differential Equations, *Frontiers in Applied Mathematics*, Vol. 6, Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[71] Majumdar, S., Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with Nonstaggered Grids, *Numerical Heat Transfer*, Vol. 13, pp. 125-132, 1988.

[72] Mallinson, G.D., De Vahl Davis, G., The Method of False Transients for the Solution of Coupled Elliptic Equations, *J. Comp. Phys.*, Vol. 12, pp. 435-461, 1973.

[73] Miller, T.F., Schmidt, F.W., Use of a Pressure-Weighted Interpolation Method for the Solution of The Incompressible Navier-Stokes Equations on a Nonstaggered Grid, *Numerical Heat Transfer*, Vol. 14, pp. 213-233, 1988.

[74] Mohamad, A.A., Viskanta, R., An Evaluation of Different Discretization Schemes for Convection of Low-Prandtl-Number Fluids in Cavities, *Numerical Heat Transfer, part B*, Vol. 16, pp. 179-192, 1989.

[75] Oden, J.T., Finite Element Analogue of Navier-Stokes Equation, *J. of Eng. Mech. Div., ASCE*, Vol. 96, No. EM4, pp. 529-534, 1970.

[76] Patankar, S.V., Spalding. D.B., A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three-Dimensional Parabolic Flows, *Int. J. of Heat Mass Transfer*, Vol. 15, pp. 1787-1806, 1972.

[77] Patankar, S.V. *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Co., 1980.

[78] Patankar, S.V., Recent Developments in Computational Heat Transfer, *Journal of Heat Transfer, (Trans. ASME)*, Vol. 110, pp. 1037-1045, 1988.

[79] Patel, M.K., Markatos, N.C., An Evaluation of Eight Discretization Schemes for Two-Dimensional Convection-Diffusion Equations, *Int. J. for Num, Meth. in Fluids*, Vol. 6, pp. 129-154, 1986.

[80] Patel, M., Cross, M., Markatos, N.C., Mace, A.C.H., An Evaluation of Eleven Discretization Schemes for Predicting Elliptic Flow and Heat Transfer in Supersonic Jets, *Int. J. Heat Mass Transfer*, Vol. 30, No. 9, pp.1907-1925, 1987.

[81] Peric, M., *A Finite Volume Method for Prediction of Three Dimensional Fluid Flow in Complex Ducts*, Ph.D thesis submitted to the University of London, London, England, 1985.

[82] Peric, M., Kessler, R., Scheuerer, G., Comparison of Finite-Volume Numerical Methods with Staggered and Collocated Grids, *Computers and Fluids*, Vol. 16, No. 4, pp. 389-403

[83] Pollard, A., Siu, A.L.W., The Calculation of Some Laminar Flows Using Various Discretized Schemes, *Comp. Meth. in App. Mech. and Eng.*, Vol. 35, pp. 293-313, 1982.

[84] Powell, M.J.D., A Hybrid Method for Nonlinear Equations, *Numerical Methods Nonlinear Algebraic Equations*, Academic Press, NY, 1970.

[85] Prakash, C., *A Finite Element Method for Prediction Flow Through Ducts with Arbitrary Cross Sections*, Ph.D. Thesis, Dept. of Mech. Eng., University of Minnesota, 1981.

[86] Prakash, C., Patankar, S.V., A Control Volume-Based Finite-Element Method for Solving the Navier Stokes Equations Using Equal-Order Velocity-Pressure Interpolation, *Numerical Heat Transfer*, Vol. 8, pp. 259-280, 1985.

[87] Prakash, C., An Improved Control Volume Finite-Element Method for Heat and Mass Transfer, and For Fluid Flow Using Equal-Order Velocity-Pressure Interpolation, *Numerical Heat Transfer*, Vol. 9, pp. 253-276, 1986.

[88] Prakash, C., Examination of the Upwind (Donor-Cell) Formulation in Control Volume Finite-Element Methods for Fluid Flow and Heat Transfer, *Numerical Heat Transfer*, Vol. 11, pp. 401-416, 1987.

[89] Raithby, G.D., Skew Upstream Differencing Schemes for Problems Involving Fluid Flow, *Comp. Meth. in App. Mech. and Eng.*, Vol. 9, pp. 153-164, 1976.

[90] Raithby, G.D., A Critical Evaluation of Upstream Differencing Applied to Problems Involving Fluid Flow, *Comp. Meth. in App. Mech. and Eng.*, Vol. 9, pp. 75-103, 1976.

[91] Raithby, G.D., Galpin, P.F., Van Doormaal, J.P., Prediction of Heat and Fluid Flow in Complex Geometries Using General Orthogonal Coordinates, *Numerical Heat Transfer*, Vol. 9, pp. 125-142, 1986

[92] Raw, M.J., *A New Control-Volume-Based Finite Element Procedure for the Numerical Solution of the Fluid Flow and Scalar Transport Equations*, Ph.D thesis submitted to the University of Waterloo, Waterloo, Canada, 1985.

[93] Reggio, M., Camarero, R., Numerical Solution Procedure for Viscous Incompressible Flows, *Numerical Heat Transfer*, Vol. 10, pp. 131-146, 1986.

[94] Reggio, M., Agouzoul, M., Camarero, R., Computation of Incompressible Turbulent Flows by an Opposed-Differencing Scheme, *Numerical Heat Transfer*, Vol. 12, pp. 307-320, 1987.

[95] Reid, J.K., On the Method of Conjugate Gradients for the Solution of Large Sparse Systems, *Large Space Sets of Linear Equations*, Academic Press, 1971.

[96] Rhie, C.M., Chow, W.L., Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA Journal*, Vol. 21, No. 11, pp. 1525-1532, 1983.

[97] Roache, P., *Computational Fluid Dynamics*, 1972

[98] Runchal, A.K., Wolfshtien, M., Numerical Integration Procedure for the Steady State Navier-Stokes Equations, *J. Mech. Eng. Sci.*, Vol. 11, pp 445, 1969.

[99] Runchal, A.K., Convergence and Accuracy of Three Finite difference Schemes for a Two-Dimensional Conduction and Convection Problem, *Int. J. for Num. Meth. in Fluids*, Vol. 4, pp 551-559, 1972.

[100] Saabas, H.J., *A CVFEM for Three-Dimensional Incompressible, Viscous Fluid Flow*, Ph.D thesis, McGill University, 1991.

[101] Schneider,G.E., Raithby, G.D., Yovanovich, M.M., Finite Element Solution for Solving the Incompressible Navier Stokes Equations using Equal Order Variable Interpolation *Num Ht Trans*, Vol. 1, pp 433-451, 1978.

[102] Schneider,G.E., Zedan, M., A Modified Strongly Implicit Procedure for the Numerical Solution of Field Problems, *Numerical Heat Transfer*, Vol. 4, pp. 1-19, 1981.

[103] Schneider,G.E., Raw, M.J., A Skewed Positive Influence Coefficient Upwinding Procedure for Control Volume Based Finite Element Convection Diffusion Computation, *Numerical Heat Transfer*, Vol. 9, p 1- 26, 1986.

[104] Schneider, G.E., A Novel Co-Located Finite Difference Scheme Procedure for the Numerical Computation of Fluid Flow, *Paper AIAA-86-1330, Presented at AIAA/ASME 4th Joint Thermophysics and Heat Transfer Conference*, June 2-4, 1986, Boston, Mass.

[105] Schneider, G.E., Raw, M.J., Control Volume Finite-Element Method for Heat Transfer and Fluid Flow Using Collocated Variables - 1. Computational Procedure, *Numerical Heat Transfer*, Vol. 11, pp. 363-390, 1987.

[106] Settari, A., K. Aziz, A Generalization of the Additive Correction Method for the Iterative Solution of Matrix Equations, *SIAM. J. Num. Anal.*, Vol. 10, pp. 506-521, 1973.

[107] Shih, T.M., Ren, A.L., Primitive-Variable Formulation Using Nonstaggered Grids, *Comp. Meth. in App. Mech. and Eng.* , Vol. 46, pp. 277-293, 1984.

[108] Shyy, W., A study of Finite Difference Approximation to Steady State Convection Dominated Flow Problems, *J. of Comp. Phys.*, Vol. 57, pp. 415-438, 1985.

[109] Shyy, W., Tong, S.S., Correa, S. Numerical Recirculating Flows Calculation Using a Body Fitted Coordinate System, *Numerical Heat Transfer*, Vol. 8, pp. 99-113, 1985.

[110] Shyy, W., Vu, T.C., On the Adoption of Velocity Variable and Grid System for Fluid Flow Computation in Curvilinear Coordinates, *J. of Comp. Phys.*, Vol. 92, pp. 82-105, 1991.

[111] Smith, R.E., Hutton, A., The Numerical Treatment of Advection: A Performance Comparison of Current Methods, *Num. Heat Transfer*, Vol. 5, pp. 439-461, 1982.

[112] Spalding, D.B., A Novel Finite Difference Formulation for Differential Expressions Involving Both First and Second Derivatives, *Int. J. for Num. Meth. in Eng.*, Vol. 4, pp. 551-559, 1972.

[113] Stone, H.L., Iterative Solution of Implicit Appropriations of Multidimensional Partial Differential Equations, *SIAM J. Num. Anal.*, Vol. 5, No. 3, pp. 530-558, 1968.

[114] Stubley, G.D., Raithby, G.D., Strong, A.B., Proposal for a New Discrete Method Based on an Assessment of Discretization Errors, *Numerical Heat Transfer*, Vol. 3, pp. 411-428, 1980.

[115] Stubley, G.D., Raithby, G.D., Strong, A.B., Woolner, K.A., Simulation of Convection and Diffusion Processes by Standard Finite Difference Schemes and By Influence Schemes, *Comp. Meth. in App. Mech. and Eng.*, Vol. 35, pp. 153-168, 1982.

[116] Taylor, C., Hood, P., A Numerical Solution of the Navier-Stokes Equations Using the Finite Element Method, *Computers and Physics*, Vol., 1, pp. 73-100, 1973.

[117] Thiart, G.D., Finite Difference Scheme for the Numerical Solution of Fluid Flow and Heat Transfer Problems on Nonstaggered Grids, *Numerical Heat Transfer, Part B*, Vol., 17, pp. 43-62, 1990.

[118] Thiart, G.D., Improved Finite-Difference Scheme for the Solution Convection-Diffusion Problems with the SIMPLEN Algorithm, *Numerical Heat Transfer, Part B*, Vol., 18, No. 1, pp. 81-96, 1990.

[119] Tsui, Y.Y., A study of Upstream-Weighted High-Order Differencing for Approximation to Flow Convection, *Int. J. of Num. Meth. in Fluids*, Vol. 13, pp. 167-199, 1991.

[120] Tzanos, C., Central Difference-Like Approximation for the Solution of the Convection-Diffusion Equation, *Numerical Heat Transfer, Part B*, Vol. 17, pp. 97-112, 1990.

[121] Van Doormaal, J.P., Raithby, G.D., Strong, A.B., Prediction of Natural Convection in Non-Rectangular Enclosures, *Numerical Heat Transfer*, Vol. 4, pp. 21-38, 1981.

[122] Van Doormaal, J.P., Raithby, G.D., Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numerical Heat Transfer*, Vol. 7, pp. 147-163, 1984.

[123] Van Doormaal, J.P., Raithby, G.D., An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, *ASME paper 85-HT-9, presented at the National Heat Transfer Conference*, Denver, Colorado, 1985.

[124] Van Doormaal, J.P., Turan, A., Raithby, G.D., Evaluation of New Techniques for the Calculation of Internal Recirculating Flows, *paper AIAA-87-0059*, AIAA 25th Aerospace Sciences Meeting, Jan. 12-15, 1987, Reno, Nevada.

[125] Vanka, S., Fully coupled Calculation of Fluid Flows with Limited Use of Computer Storage, *Argonne National Laboratory*, Report-ANL-83-87, 1983

[126] Varejao, L.M.C., ***Flux-Spline Method for Heat, Mass, and Momentum Transfer***, Ph.D. Thesis, Dept. of Mech. Eng., University of Minnesota, 1980.

[127] Winslow, A.M., Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh, ***Journal of Computational Physics***, Vol. 1, pp. 149-172, 1966.

[128] Wong,H.H., Raithby, G.D., Improved Finite Difference Methods Based on a Critical Evaluation of the Approximation Errors, ***Numerical Heat Transfer***, Vol. 2, pp. 139-163, 1979.

[129] Zienkiewicz, O.C, ***The Finite Element Method***, 3rd ed. , McGraw Hill Book Co., London, 1977.

[130] Zurigat, Y.H., Ghajar, A., Comparative Study of Weighted Upwind and Second Order Upwind Difference Schemes, ***Num. Heat Transfer, part B***, Vol. 18, No. 1, pp 61-80, 1990.

Appendix A

IMPLEMENTATION DETAILS OF THE MAW SCHEME

As was discussed in 2.4.3 for the MAW scheme, the values of ϕ at the integration-points are calculated via Eq. 2.39, which is repeated here for convenience:

$$[\phi_k] = [A^{-1}][B][\phi_{l,m}] \Longrightarrow [\phi_k] = [D][\phi_{l,m}] \quad where \quad [D] = [A^{-1}][B] \qquad 2.39$$

(A.1)

in which, [A] is the (4 x 4) coefficient matrix of the vector of the 4 integration-point values $[\phi_k]$, and [B] is the (4 x 4) coefficient matrix of the vector of the 4 nodal values $[\phi_{l,m}]$.

In this appendix, the elements of the matrices [A] and [B] will be presented, and then the methods used to invert for the matrix [A], as developed in this thesis, will be given.

A.1 Elements of the Matrices [A] and [B]

The matrices [A], and [B] are of the following form:

$$A = \begin{bmatrix} 1 & -f_1' & 0 & -f_1 \\ -f_2 & 1 & -f_2' & 0 \\ 0 & -f_3' & 1 & f_3 \\ -f_4 & 0 & -f_4' & 1 \end{bmatrix} \tag{A.2}$$

$$B = \begin{bmatrix} 1-f_1 & 1-f_1' & 0 & 0 \\ 0 & 1-f_2 & 1-f_2' & 0 \\ 0 & 0 & 1-f_3' & 1-f_3 \\ 1-f_4 & 0 & 0 & 1-f_4' \end{bmatrix} \tag{A.3}$$

where:

$$f_1 = min\left[max\left(-\frac{\dot{m}_4}{\dot{m}_1}, 0\right), 1\right] \tag{A.4}$$

$$f_1' = min\left[max\left(-\frac{\dot{m}_2}{\dot{m}_1}, 0\right), 1\right] \tag{A.5}$$

$$f_2 = min\left[max\left(-\frac{\dot{m}_1}{\dot{m}_2}, 0\right), 1\right] \tag{A.6}$$

$$f_2' = min\left[max\left(-\frac{\dot{m}_3}{\dot{m}_2}, 0\right), 1\right] \tag{A.7}$$

$$f_3 = min\left[max\left(-\frac{\dot{m}_4}{\dot{m}_3}, 0\right), 1\right] \tag{A.8}$$

$$f_3' = min\left[max\left(-\frac{\dot{m}_2}{\dot{m}_3}, 0\right), 1\right] \tag{A.9}$$

$$f_4 = min\left[max\left(-\frac{\dot{m}_1}{\dot{m}_4}, 0\right), 1\right] \tag{A.10}$$

$$f_4' = min\left[max\left(-\frac{\dot{m}_3}{\dot{m}_4}, 0\right), 1\right] \tag{A.11}$$

A.2 Inversion of the Matrix [A]

As was mentioned in section 2.4.3, a special pivoting technique was developed for inversion of the matrix [A] in this work. This pivoting technique, is incorporated in the following FORTRAN77 code:

```
*       SUBROUTINE MINV

THE PURPOSE OF THIS SUBROUTINE IS TO INVERT THE COEFFICIENT
MATRIX [A], WHICH IS A 4 x 4 MATRIX (N = 4).
AN SPECIAL PIVOTING STRATEGY IS USED TO INVERT MATRIX [A]

.       SUBROUTINE MINV(A,N)
.       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
.       DIMENSION A(4,4)

COLUMN 1, PIVOT ON ELEMENT 1,1
.       DO 10 J = 2, 4
.       DO 10 I = 2, 4
10        A(I,J) = A(I,J) - A(I,1) * A(1,J)
.       DO 20 J = 2, 4
20        A(1,J) = A(1,J)
.       DO 30 I = 2, 4
30        A(I,1) = -A(I,1)

COLUMN 2, PIVOT ON ELEMENT 2,2
.       DO 40 J = 3, 4
.       DO 40 I = 3, 4
40        A(I,J) = A(I,J) - A(I,2) * A(2,J) / A(2,2)
.       A(1,1) = A(1,1) - A(1,2) * A(2,1) / A(2,2)
.       DO 50 J = 3, 4
.       A(1,J) = A(1,J) - A(1,2) * A(2,J) / A(2,2)
```

```
50          A(2,J) = A(2,J) / A(2,2)
.           DO 60 I = 3, 4
.           A(I,1) = A(I,1) - A(I,2) * A(2,1) / A(2,2)
60          A(I,2) = -A(I,2) / A(2,2)
.           A(1,2) = -A(1,2) / A(2,2)
.           A(2,1) = A(2,1) / A(2,2)
.           A(2,2) = 1.0D+00 / A(2,2)
```

COLUMN 3, PIVOT ON ELEMENT 3,3

```
.           DO 70 J = 1, 2
.           DO 70 I = 1, 2
70          A(I,J) = A(I,J) - A(I,3) * A(3,J) / A(3,3)
.           A(4,4) = A(4,4) - A(3,4) * A(4,3) / A(3,3)
.           DO 80 J = 1, 2
.           A(4,J) = A(4,J) - A(4,3) * A(3,J) / A(3,3)
80          A(3,J) = A(3,J) / A(3,3)
.           DO 90 I = 1, 2
.           A(I,4) = A(I,4) - A(I,3) * A(3,4) / A(3,3)
90          A(I,3) = -A(I,3) / A(3,3)
.           A(4,3) = -A(4,3) / A(3,3)
.           A(3,4) = A(3,4) / A(3,3)
.           A(3,3) = 1.0D+00 / A(3,3)
```

COLUMN 4, PIVOT ON ELEMENT 4,4

```
.           DO 110 J = 1, 3
.           DO 110 I = 1, 3
```

```
110         A(I,J) = A(I,J) - A(I,4) * A(4,J) / A(4,4)
.         DO 120 J = 1, 3
120         A(4,J) = A(4,J) / A(4,4)
.         DO 130 I = 1, 3
130         A(I,4) = -A(I,4) / A(4,4)
.         A(4,4) = 1.0D+00 / A(4,4)

.         RETURN
.         END
```