IMPLICIT RUNGE-KUTTA METHODS TO SIMULATE UNSTEADY INCOMPRESSIBLE FLOWS

A Dissertation

by

MUHAMMAD IJAZ

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2007

Major Subject: Mechanical Engineering

IMPLICIT RUNGE-KUTTA METHODS TO SIMULATE UNSTEADY INCOMPRESSIBLE FLOWS

A Dissertation

by

MUHAMMAD IJAZ

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	N. K. Anand
Committee Members,	Sai C. Lau
	Obdulia Ley
	Hamn-Ching Chen
Head of Department,	Dennis L. O'Neal

December 2007

Major Subject: Mechanical Engineering

ABSTRACT

Implicit Runge-Kutta Methods to Simulate Unsteady

Incompressible Flows. (December 2007)

Muhammad Ijaz, B.Sc., University of Engineering & Technology, Lahore, Pakistan;

M.Eng., Texas A&M University, College Station

Chair of Advisory Committee: Dr. N. K. Anand

A numerical method (SIMPLE DIRK Method) for unsteady incompressible viscous flow simulation is presented. The proposed method can be used to achieve arbitrarily high order of accuracy in time-discretization which is otherwise limited to second order in majority of the currently used simulation techniques. A special class of implicit Runge-Kutta methods is used for time discretization in conjunction with finite volume based SIMPLE algorithm. The algorithm was tested by solving for velocity field in a lid-driven square cavity. In the test case calculations, power law scheme was used in spatial discretization and time discretization was performed using a second-order implicit Runge-Kutta method. Time evolution of velocity profile along the cavity centerline was obtained from the proposed method and compared with that obtained from a commercial computational fluid dynamics software program, FLUENT 6.2.16. Also, steady state solution from the present method was compared with the numerical solution of Ghia, Ghia, and Shin and that of Erturk, Corke, and Goökçöl. Good agreement of the solution of the proposed method with the solutions of FLUENT; Ghia, Ghia, and Shin; and Erturk, Corke, and Goökçöl establishes the feasibility of the proposed method.

DEDICATION

To the Creator of the Universe, my parents, and my grandfather

to whom every success in my life is attributed

ACKNOWLEDGEMENTS

I would like to thank the chair of my committee, Dr. Anand, for his guidance and support throughout the course of this research. He gave me such authority and freedom of thought in my research that kept my vision unlimited. I also want to extend my gratitude to my committee members for serving on my committee. Thanks to Dr. Griffin who gave me the opportunity to work on experimental research and hence add diversity to my research experience.

Thanks also to Dr. Lalk, Dr. Schneider, Dr. Cohen, Mr. Bollfrass, Dr. McDermott, and Dr. Burger who provided me with their encouragement during my teaching as an instructor at Texas A&M University. I would also like to thank Dr. Anand, Dr. Caton, Dr. Lau, Dr. O'Neal, and my students for giving me the opportunity to thrive as a teacher while working on my research. Thanks to my colleagues who engaged me in interesting discussions and gave me a flavor of mentoring.

Thanks to the Center for Teaching Excellence personnel who helped me demonstrate and enhance my leadership and gave me the opportunity to mentor junior graduate teaching assistants. Thanks to my friends and the department staff for making my time at Texas A&M University a great experience.

Thanks to my wife and children for their patience. Their love gave me the strength and incentive to overcome challenges.

Lastly, I do not find words to thank my parents and describe what they have done for me. They always wanted to see me at the top.

NOMENCLATURE

A	weights used in *stage calculations* in a Runge-Kutta method
a	coefficients in the discretized form of momentum/pressure-correction equations
B	weights used in update solution in a Runge-Kutta method
$\mathbf{B}$	matrix of weights used in update solution in a Runge-Kutta method
b	source term in momentum equations
c	deferred correction term in the expression for time-derivative of velocity
$\vec{F}$	body force vector
f	interpolation factor
f	time-derivative
H	momentum term used in momentum interpolation
h	time-step size
i, j	indices associated with grid points
n	index associated with time step
$\vec{P}$	surface force vector
P	order of accuracy in a Runge-Kutta method
p	pressure, N/m^2
q	number of stages in a Runge-Kutta method
RMS	root mean square
Re	Reynolds number
res	residual

r, s	stage indices used in Runge-Kutta methods
$\vec{\mathrm{S}}$	surface vector
S	surface
S	source term
t	time, s
u	component of velocity in x-direction, m/s
$\vec{V}$	velocity vector, m/s
V	volume, m^3
v	component of velocity in y-direction, m/s
x, y	x- and y-coordinates, m

Greek Symbols

α	under-relaxation factor
Γ	general diffusion coefficient
Δ	geometric lengths of CVs, m
δ	diffusion length, m
μ	dynamic viscosity, $N.s/m^2$
ν	kinematic viscosity (μ / ρ), m^2/s
ρ	density, kg/m^3
τ	parameters used to define quadrature points in Runge-Kutta methods
$\boldsymbol{\tau}$	matrix of parameters used to define quadrature points in Runge-Kutta methods
φ	general variable representing u or v

Mathematical Symbols

∇	gradient operator
∇^2	Laplacian operator

Superscripts

$'$	correction
$*$	incorrect or guessed value
l	associated with preceding iteration
p	associated with pressure
pc	associated with pressure correction
u	associated with u-velocity
v	associated with v-velocity

Subscripts

i	association with i^{th} node
j	association with j^{th} node
max	maximum
n	time-step index
W, E, S, N	west, east, south, and north nodes relative to a node under consideration
w, e, s, n	west, east, south, and north faces of a node under consideration

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

1. INTRODUCTION

In nature and in engineering applications, heat transfer, phase changes and chemical reactions are mostly time dependent. Scientists and engineers are frequently confronted with the challenge of accurate prediction of time-dependent flow, thermal, and/or species fields in the areas like environmental engineering, meteorology, health, bio-medical, aeronautics, astronautics, energy exploration, power production, industry, and defense applications.

Accuracy of predictability is important for economy, safety, efficiency, and environment-friendliness in design of equipment and machinery, especially during process start up, control, and shutdown. Accuracy of predictability is also important in the simulation of processes which are difficult to be realized in a laboratory. Also, accurate techniques are essential for Direct Numerical Simulation (DNS) of flows. Until recently, time-dependent numerical simulations have been modestly accurate in time advancement due to the past limitations on computing capabilities and storage of data and memory. Majority of the current simulation methods are limited to second order accuracy in time. Moreover, usual simulation methods rely on explicit time discretization methods. For explicit methods numerical stability of solution has been an issue which is generally guaranteed only with very small time step sizes. Major commercial Computational Fluid Dynamics (CFD) software programs provide options for implicit time advancing, but the accuracy is limited to second order in time.

This dissertation follows the style and format of *Numerical Heat Transfer*.

The purpose of the current research is to propose a stable simulation method that can be used to achieve arbitrarily high order of accuracy in time advancement in simulation of time-dependent incompressible flow and heat transfer. The strategy is to combine the state-of-the-art mathematical tools with proven flow simulation algorithms to develop simulation techniques with higher-order accuracy. A special class of implicit Runge-Kutta methods is used in conjunction with SIMPLE algorithm [1]. The proposed method is called SIMPLE DIRK method. This method was initially presented for staggered grid approach at an international conference [2]. A journal paper on this method has also been published [3]. A FORTRAN code was developed to implement the method. As a test case a lid-driven square cavity flow was simulated with the developed code. The results were compared with the solution of a commercial CFD software program, FLUENT [4] for the same test case. Steady state solution was compared with the solutions of Ghia et al. [5] and Erturk et al. [6]. The method was also extended to co-located variables or non-staggered grid approach [7, 8]. The results of the non-staggered grid method were compared with the results of the staggered grid method.

Good agreement of the results of the developed code with the results of FLUENT [4], Ghia et al. [5], and Erturk et al. [6] establishes feasibility of the proposed method and prospects for its extension to complex geometry and more complex flows involving chemical reaction, radiation, and multiple phases.

This dissertation is arranged in eight sections. Section 2 presents a literature review of the work done on temporal discretization of incompressible flow equations, and establishes the need for the application of higher-order implicit methods to incompressible

flow simulation. In Section 3, the author's choice of a special class of implicit Runge-Kutta methods is justified on the basis of three criteria. Section 4 gives a detailed description of the method presented herein. At first the model equations are presented. Then the numerical method is discussed separately for both the staggered grid and the non-staggered grid approaches. In Section 5, a test case is described. This section explains the strategy adopted for the validation of the presented method. In Section 6, results of the presented method are compared with the results of FLUENT [4] and published numerical solutions, and conclusions are drawn from the presented discussion. Section 7 presents a summary of this dissertation work. In Section 8, some recommendations for possible future work are presented.

2. LITERATURE REVIEW ON TEMPORAL DISCRETIZATION

As will be described in Section 4, spatial discretization of momentum conservation equations converts them into first order ordinary differential equations (ODEs) in time. Depending on how the unavailability of evolution of pressure is handled, the available formulations can be identified as either vorticity-based or primitive variables-based. Based on the method used for time advancement in the solution of the spatially discretized form of momentum conservation equations, the available numerical methods can be categorized as explicit, implicit, or partially implicit.

In almost all the early flow simulation methods, time advancement was performed by explicit first-order finite difference or forward Euler method. The earliest attempts to solve flow problems numerically were made by using finite difference methods with primitive variables. Harlow [9] proposed the Particle-In-Cell (PIC) method for transient, compressible flow which had a combination of Lagrangian and Eulerian approaches. However PIC method was intensive in memory and computational effort. Gentry et al. [10] developed a variation of PIC method, called the Fluid-In-Cell (FLIC) method, which used finite differencing in Eulerian approach. Stability was a concern in FLIC method due to improper velocity and pressure coupling. Fromm and Harlow [11] developed vorticity-stream function formulation for transient, incompressible flows which is still being used for flow simulation in two-dimensional domains. The Marker-And-Cell (MAC) method of Harlow and Welch [12] was the earliest successful simulation method for unsteady incompressible flows by using primitive variables at staggered locations. The pseudo or artificial compressibility method developed by Chorin [13] modifies the continuity

equation for incompressible flows with an additional term. The artificial compressibility method provides an evolution equation for pressure and thus a mechanism to march in time. But this method ensures a solenoidal velocity field only at steady state and thus is not suitable for transient simulations. Later, Chorin [14, 15] used the Helmholtz-Hodge decomposition theorem and proposed a method for velocity-pressure coupling in incompressible flows, called projection or fractional-step method. Denaro [16] has presented a detailed discussion on the application of the Helmholtz-Hodge decomposition in projection methods for incompressible flows. Majority of modern flow simulation methods are variations of the projection method or fractional-step method. However, as pointed out by Orszag et al. [17], pseudo or spurious numerical boundary layer effect encountered in fractional-step methods can induce substantial time differencing errors. Researchers are still trying to deal with the effect of spurious numerical boundary layer (Dagan [18]).

Alternating Direction Implicit (ADI) method, proposed by Peaceman and Rachford [19], paved the way for implicit time advancement. ADI method was later adopted for hyperbolic differential equations by Lees [20]. Steger and Kutler [21] were among the first researchers to present an implicit method for time advancement in incompressible flows. Earlier fully implicit methods were backward Euler methods which were unconditionally stable, but only first-order accurate in time. Examples of higher-order implicit methods are mid-point rule and second-order implicit Crank-Nicholson method [22].

Combinations of explicit and implicit methods were also developed. Many different applications of predictor-corrector method, originally proposed by MacCormack

[23], are examples of combination explicit-implicit methods. Combination explicit-implicit methods suffer from constraints on time-step size due to stability conditions.

It may be noted that pressure-velocity coupling remained an area of interest in all incompressible flow simulations. In recent simulation methods, higher order of accuracy and stability are major areas of interest in addition to pressure-velocity coupling.

To achieve higher order of accuracy, multi-point methods, such as Adams-Bashforth methods, were used which need information at more than one instant in time at which data has already been computed. However, multipoint methods rely on some other method to generate enough data to start time marching. These methods often suffer from instability and generate non-physical solutions (Ferziger and Peric [24]).

Runge-Kutta (RK) methods offer an alternative to multi-point methods for higher order of accuracy in time. In RK methods, the value of the dependent variable at the end of any time step is calculated from its value at the beginning of the time step. For higher order of accuracy, the values of the dependent variable and/or its derivative are calculated at intermediate time instants within a single time step. For a desired order of accuracy, RK methods are more stable when compared with multi-point methods of same accuracy (Ferziger and Peric [24]). The classical explicit RK methods can be used to achieve high-order accuracy, but they are restricted by stability constraints on time-step size. Especially for unsteady incompressible flow simulations at high Reynolds numbers, which involve solution of stiff ODEs, explicit RK methods are not suitable. Explicit RK methods have been developed and employed for time advancement in compressible flow simulation by many researchers such as Fehlberg [25, 26], Jameson et al. [27], and Cebeci et al. [28].

The projection method of Chorin [14] was used by many researchers to develop methods with higher-order accuracy in time. Kim and Moin [29] developed an explicit-implicit projection method using a second order explicit Adams-Bashforth method for the convective terms and a second order implicit Crank-Nicolson method for the viscous term. Kan [30] and Bell et al. [31] also developed projection methods of second order accuracy in time similar to the one proposed by Kim and Moin [29]. These projection methods, though used by many, were later studied and criticized by many authors. Perot [32] argued that pressure calculation is only first order accurate in time. Strikwerda and Lee [33] confirmed Perot's argument. There have been some recent improvements in the accuracy of projection methods. Brown et al. [34] and Liu et al. [35] have presented projection methods with second order accuracy in time. Rai and Moin [36] presented a method, which is second order accurate in time, for direct simulation of incompressible fully developed turbulent channel flow using an explicit Runge-Kutta method for the convective terms and an implicit Crank-Nicholson method for the viscous terms. Based on third-order accurate Runge-Kutta methods, semi-implicit schemes were proposed by Spalart et al. [37], Verzicco and Orlandi [38], and Nikitin [39].

Besides the projection methods discussed in the above paragraph, a method of velocity and pressure coupling was proposed by Caretto et al. [40]. This method, which is called SIMPLE, is documented and discussed in detail by Patankar [1]. Many variations of this method with some improvements have been developed [1, 41-52]. However, there has been little attention to the possibility of using higher order time discretization in conjunction with SIMPLE family of methods. The author has found no work, in particular,

on the use of implicit RK methods. The current work focused on using implicit RK methods in conjunction with SIMPLE. The factors that led the author to choose implicit RK methods for temporal discretization are discussed in Section 3. The reason for choosing SIMPLE instead of its later variations is that although the variations of SIMPLE were proposed with claimed improvements yet these methods have demerits that are still being discussed by many researchers [41-52].

3. CHOICE OF TEMPORAL DISCRETIZATION METHOD FOR THE CURRENT WORK

Choice of temporal discretization method for the solution of incompressible flows is dictated by several considerations:

First, an explicit evolution equation for pressure is not available; instead, implementation of the continuity equation provides an implicit form of pressure evolution. In other words, pressure field has to evolve in time so that the continuity equation is satisfied at all instants in time. This is in contrast with compressible flows where the continuity equation contains time rate of change of density which can be related to pressure through some equation of state.

Second, the momentum conservation equations for incompressible flows are stiff differential equations which are susceptible to numerical instability, especially at higher Reynolds numbers. Stiffness may be defined in several ways. There are many mathematical representations of stiffness in the literature. In simple words, a stiff ODE requires much smaller time-step size to obtain a stable solution using an explicit method than that required for a desired accuracy using an implicit method. Thus the time-step size in explicit methods is dictated by stability rather than accuracy. Hoffman [53] has provided several simple definitions of stiffness. A stiff ODE contains some transient terms that decay faster than others. From the viewpoint of computational effort, an ODE is called stiff if the feasible step size is too large to give a stable solution. Often, for a stable solution the required step size is so small that the round-off errors dominate the solution.

The stiffness of an ODE can be mild or severe. Moreover, an ODE may be stiffer in certain part of the solution domain than the rest of the domain.

Third, there should be room for adaptation to arbitrarily high order of accuracy. High order of accuracy is desirable because higher order methods are more efficient [54].

The above considerations lead us to look for some implicit method for simultaneous iterative solution of momentum and mass conservation equations. Implicit methods can be derived to be unconditionally stable. Therefore, the step size is not limited by stability. This makes implicit methods a suitable choice for stiff problems (Dekker and Verwer [55]). Moreover, for higher order of accuracy, Runge-Kutta methods are one-step alternative to multipoint methods. Therefore, implicit Runge-Kutta methods were adopted for time discretization in the current work. For higher order simulations, implicit RK methods are preferable over their explicit counterparts because, beyond order 4, explicit RK methods require more stages than the required order (Butcher [54]). Since formally been proposed by Butcher [56] and some others, implicit Runge-Kutta methods have gone through years of development. The relatively large computational effort associated with implicit Runge-Kutta methods is less of an issue due to the advancement in computing hardware technology. Interested readers may refer to Appendix A for an account of Runge-Kutta (RK) methods.

In the current work, the author chose to adopt Explicit first stage, Single diagonal coefficient, Diagonally Implicit, Runge-Kutta (ESDIRK) methods in conjunction with SIMPLE algorithm. The current method was named as SIMPLE DIRK method. Several desirable characteristics of ESDIRK methods are discussed by Kennedy and Carpenter

[57] and Butcher [54]. Bijl et al. [58, 59] and Carpenter et al. [60] simulated unsteady compressible flow using ESDIRK methods in a pseudo-time sub-iteration algorithm. Isono and Zingg [61] performed similar simulation with a Newton-Krylov Algorithm. A special class of ESDIRK methods, called stiffly accurate ESDIRK methods, was chosen by these researchers. These methods were identified by Prothero and Robinson [62] and explained in detail by Hairer and Wanner [63]. We also chose to adapt stiffly accurate RK methods in the current simulation method. The reason for this choice is explained as follows. As will be explained later in this thesis, velocity and pressure fields in the proposed method are calculated simultaneously and implicitly during *stage calculations* in every time-step while satisfying both momentum and continuity equations. However, the *update solution* in every time-step is explicit in nature and does not guarantee a divergence-free velocity field. Moreover, the pressure field, corresponding to the velocity field obtained from the *update solution*, is not calculated simultaneously. The pressure field is required for calculations in the subsequent time-step and therefore needs to be calculated from the velocity field by some method such as solution of pressure Poisson's equation. Stiffly accurate RK methods eliminate the need for *update solution* which is required in other RK methods in every time-step. The last stage calculations in any time-step n yield a velocity field which is equal to the velocity field at the end of that time-step. Since this velocity field results from simultaneous solution of momentum and continuity equations, the corresponding pressure field is also calculated.

4. NUMERICAL METHOD

The method presented herein is an implicit formulation in time used in conjunction with finite volume based SIMPLE algorithm. The method can be used for arbitrarily high order of accuracy in time. Moreover, this method can be extended to three-dimensional domains with curvilinear coordinates, but for the sake of simplicity, we will limit our discussion to two-dimensional domains with Cartesian coordinates. Details of finite volume approach and SIMPLE algorithm can be found in many publications such as Patankar [1]. However, time discretization method presented in the current work is different from the usual time advancement methods presented in the literature.

The method is presented for both the staggered grid and the co-located variables approach.

4.1. Model Equations

The flow of a fluid is modeled by the law of conservation of mass and the law of motion. Continuity equation as derived from the law of conservation of mass is given below:

$$\frac{\partial \rho}{\partial t} + \nabla . \left(\rho \vec{V} \right) = 0 \tag{4.1}$$

From Newton's Second Law, equations of motion are derived which can be written in the following vector form (Schlichting [64]):

$$\rho \frac{D\vec{V}}{Dt} = \vec{F} + \vec{P} \tag{4.2}$$

In the current work, a fluid with constant thermo-physical properties is considered. It is assumed that the fluid is isotropic, i.e. the stress components and the rate of strain are related by the same relationship in all directions. The fluid is also assumed to be Newtonian, i.e. the stress components and the rate of strain are linearly related (Stoke's law). It is further assumed that the flow is incompressible with negligible viscous dissipation. With these assumptions, the model equations take the following form:

Mass conservation or continuity:

$$\nabla.\vec{V} = 0 \tag{4.3}$$

A velocity field that satisfies Eq. (4.3) is called divergence-free or solenoidal velocity field.

Momentum Conservation:

$$\frac{\partial \vec{V}}{\partial t} + \vec{V}.(\nabla \vec{V}) = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \vec{V} \tag{4.4}$$

Components of Eq. (4.4) in *x*- and *y*-direction are given below.

$$\rho \frac{\partial u}{\partial t} = -\rho \nabla.(u\vec{V} - \nu \nabla u) - \nabla p \tag{4.5}$$

$$\rho \frac{\partial v}{\partial t} = -\rho \nabla .(v\vec{V} - \nu \nabla v) - \nabla p \tag{4.6}$$

The numerical method used in the current work is a finite volume method in which the model equations are solved in their integral form. The method was used with staggered as well as non-staggered grid approach. The following sections explain the method for these two approaches.

4.2. Staggered Grid Approach

The concept of staggered-grid was introduced by Harlow and Welch [12]. Staggered grid approach ensures proper discretization of the pressure gradient terms in momentum conservation equation. This approach eliminates the possibility of emergence of non-physical pressure field in the solution. Staggered grid approach is very suitable for simple rectangular geometries. In the proceeding subsections, spatial and temporal discretization of mass conservation and momentum conservation equations is explained for staggered grid approach. Subsequently, simultaneous solution of the discretized equations is discussed.

4.2.1. Spatial Discretization

In the staggered grid approach pressure is calculated at the geometric center of control volume (CV) and velocities are calculated at the CV faces. Mass conservation equation (Eq. (4.3)) is integrated over control volumes enclosing main grid points; a typical CV is shown in Figure 4.1. Whereas, *x*- and *y*-components of momentum conservation equation (Eq. (4.4)) are integrated over the control volumes of the staggered grids shown in

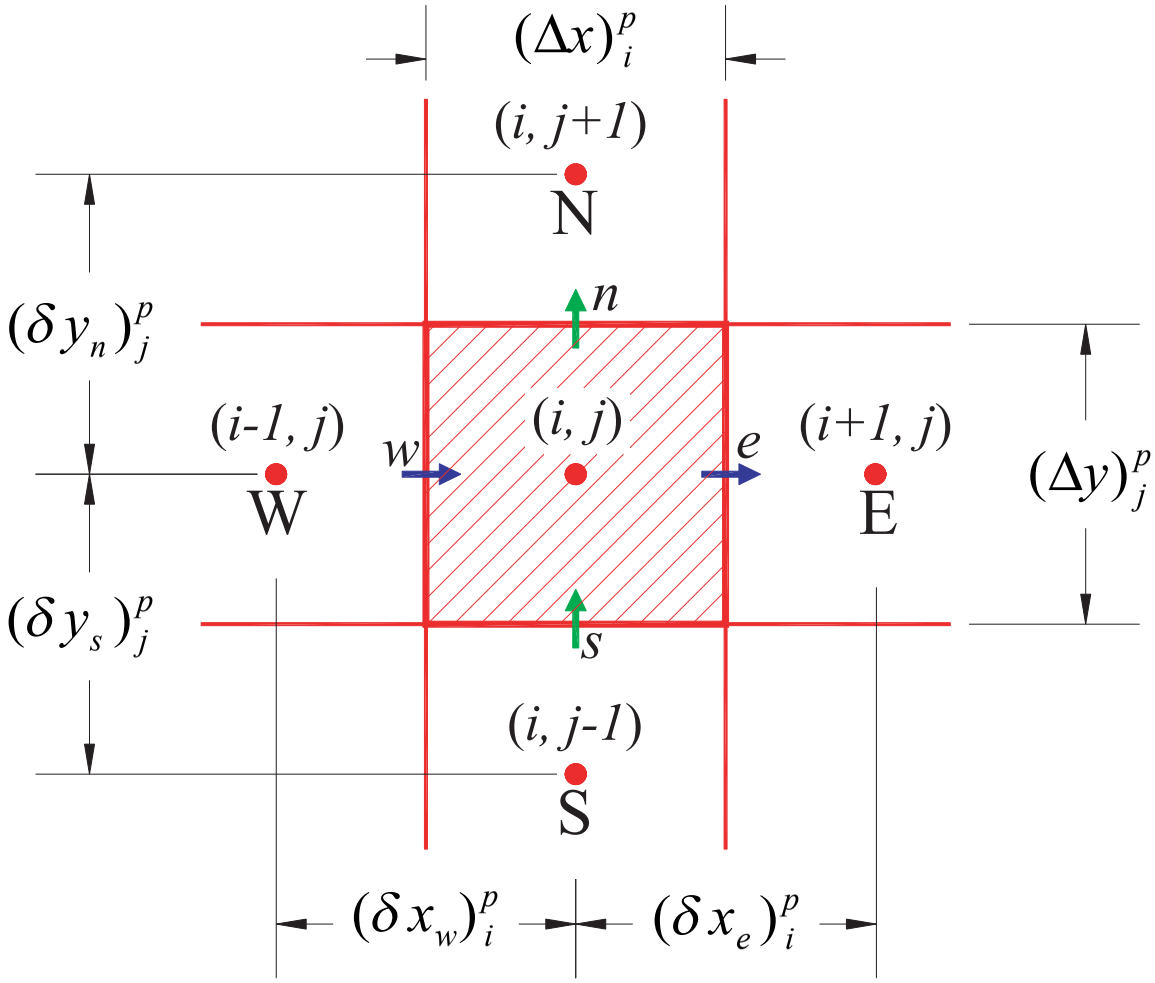


Figure 4.1 Typical *p*-Control Volume in Staggered-Grid Approach

Figures 4.2 and 4.3, respectively.

4.2.1a. Mass Conservation Equation

Integrating Eq. (4.3) over a typical *p*-CV, shown in Figure 4.1, and applyingdivergence theorem one gets,

$$\int_{\mathrm{V}} \nabla \vec{V}.d\mathrm{V} = \int_{\mathrm{S}} \vec{V}.d\vec{\mathrm{S}} \tag{4.7}$$

We assume that at any point on the face of the *p*-CV, velocity remains constant and equal to its value at the center of the face. With this assumption, discretization of Eq. (4.7) gives:

$$\left(u_{i,j} - u_{i-1,j}\right)\left(\Delta y\right)_j^p + \left(v_{i,j} - v_{i,j-1}\right)\left(\Delta x\right)_i^p = 0 \tag{4.8}$$

Eq. (4.8) is discretized form of mass conservation equation, Eq. (4.3), for a *p*-grid node (i, j) in staggered-grid approach.

4.2.1b. Momentum Conservation Equations

Integrating Eq. (4.5) over a *u*-CV, shown in Figure 4.2, and applying divergence theorem:

$$\rho\int_{\mathrm{V}} \frac{\partial u}{\partial t} d\mathrm{V} = -\underbrace{\rho\int_{\mathrm{S}} u\vec{V}.d\vec{\mathrm{S}}}_{\text{Convection Term}} + \underbrace{\int_{\mathrm{S}} \mu\nabla u.d\vec{\mathrm{S}}}_{\text{Diffusion Term}} - \underbrace{\int_{\mathrm{V}} \nabla p d\mathrm{V}}_{\text{Source Term}} \tag{4.9}$$

Similarly, for *v*-velocity, integrating Eq. (4.6) over a *v*-CV, shown in Figure 4.3,

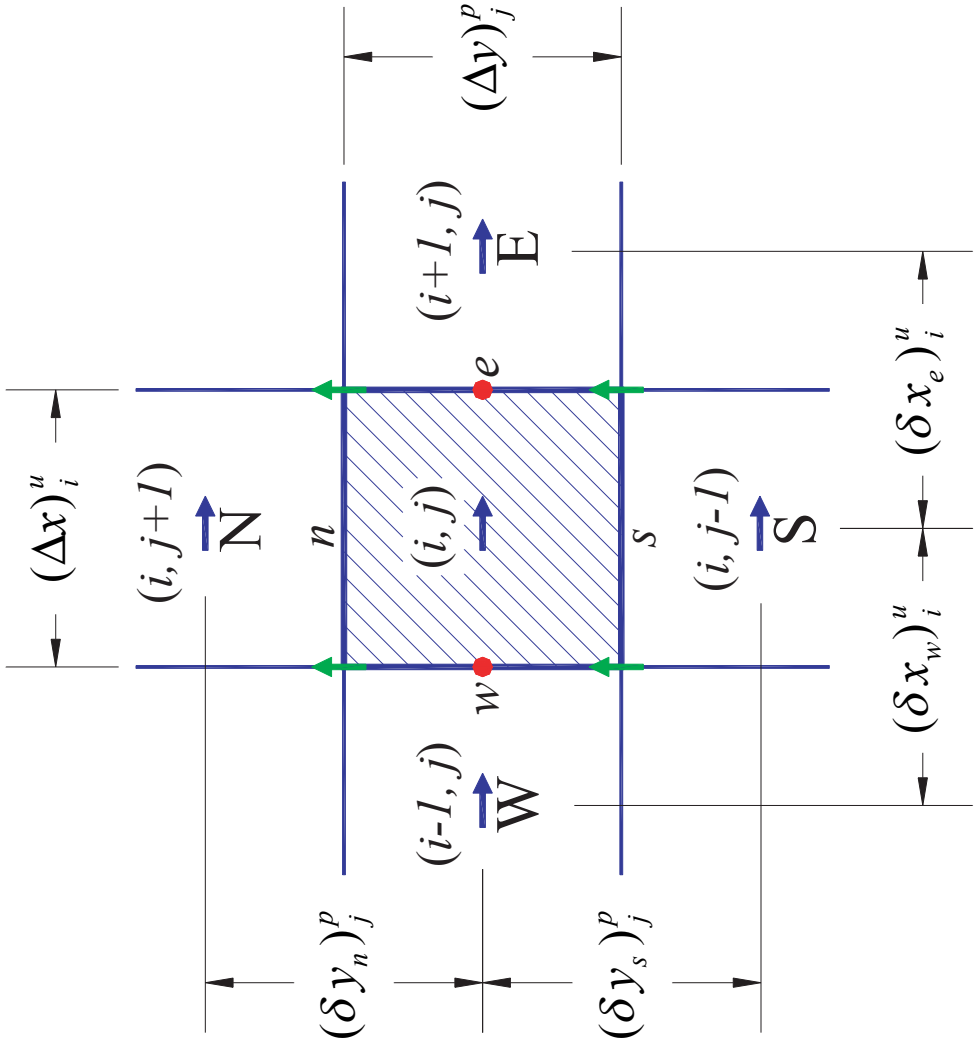


Figure 4.2 Typical u-Control Volume in Staggered-Grid Approach

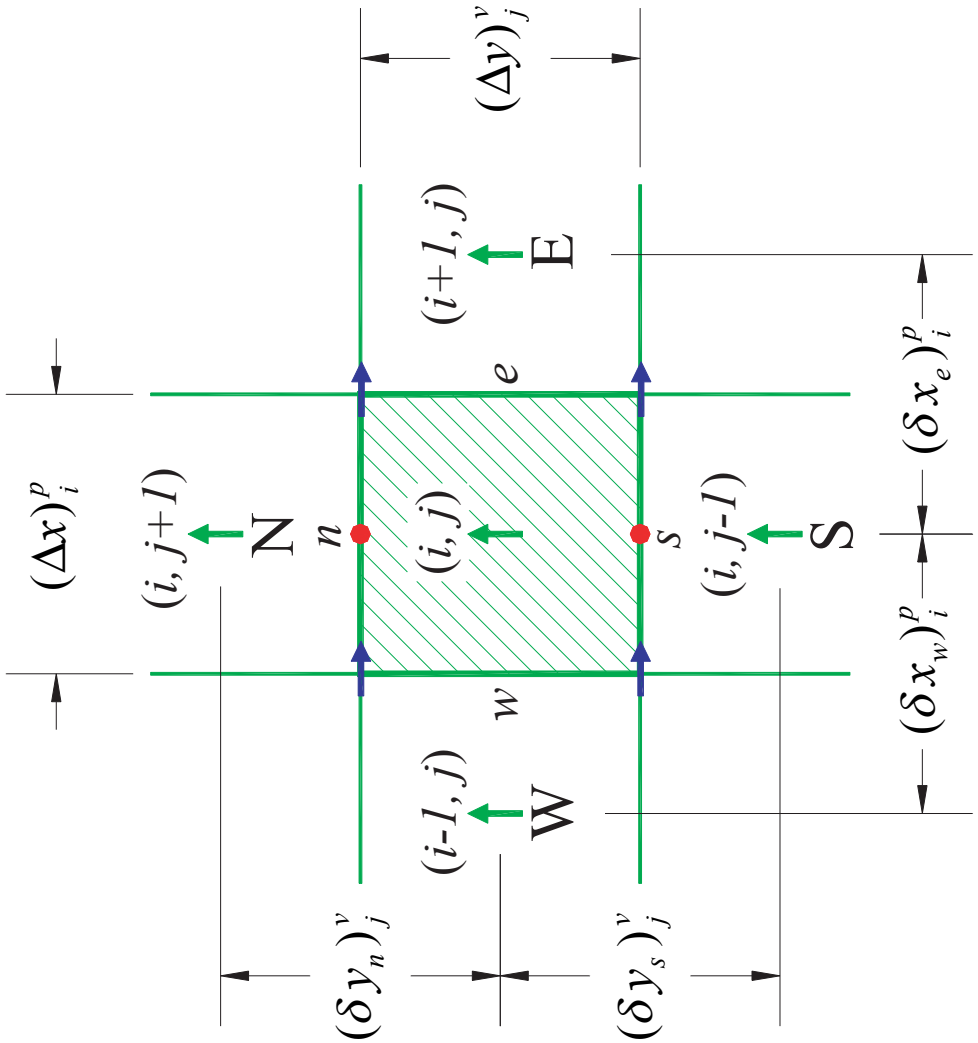


Figure 4.3 Typical *v*-Control Volume in Staggered-Grid Approach

and applying divergence theorem:

$$\rho\int_{\mathrm{V}}\frac{\partial v}{\partial t}d\mathrm{V} = -\underbrace{\rho\int_{\mathrm{S}}v\vec{V}.d\vec{\mathrm{S}}}_{\text{Convection Term}} + \underbrace{\int_{\mathrm{S}}\mu\nabla v.d\vec{\mathrm{S}}}_{\text{Diffusion Term}} - \underbrace{\int_{\mathrm{V}}\nabla p d\mathrm{V}}_{\text{Source Term}} \tag{4.10}$$

Eqs. (4.9) and (4.10) can be discretized with any spatial discretization scheme of one's choice yielding different orders of accuracy. Regardless of the spatial discretization scheme used, the discretized equations can be written in the following form.

For a *u*-grid node (*i, j*):

$$\begin{aligned}\left(\partial u/\partial t\right)_{i,j} &= f^{u}\left(t,u,v,p\right)_{i,j}\\ &= \frac{1}{\rho\left(\Delta x\right)_{i}^{u}\left(\Delta y\right)_{j}^{p}}\begin{Bmatrix}-\left(a_{W_{i,j}}^{u}+a_{E_{i,j}}^{u}+a_{S_{i,j}}^{u}+a_{N_{i,j}}^{u}\right)u_{i,j}\\ +a_{W_{i,j}}^{u}u_{i-1,j}+a_{E_{i,j}}^{u}u_{i+1,j}+a_{S_{i,j}}^{u}u_{i,j-1}+a_{N_{i,j}}^{u}u_{i,j+1}\\ +\left(p_{i,j}-p_{i+1,j}\right)\left(\Delta y\right)_{j}^{p}+c_{i,j}^{u}\end{Bmatrix}\end{aligned} \tag{4.11}$$

Similarly, for a *v*-grid node (*i, j*):

$$\begin{aligned}\left(\partial v/\partial t\right)_{i,j} &= f^{v}\left(t,u,v,p\right)_{i,j}\\ &= \frac{1}{\rho\left(\Delta x\right)_{i}^{p}\left(\Delta y\right)_{j}^{v}}\begin{Bmatrix}-\left(a_{W_{i,j}}^{v}+a_{E_{i,j}}^{v}+a_{S_{i,j}}^{v}+a_{N_{i,j}}^{v}\right)v_{i,j}\\ +a_{W_{i,j}}^{v}v_{i-1,j}+a_{E_{i,j}}^{v}v_{i+1,j}+a_{S_{i,j}}^{v}v_{i,j-1}+a_{N_{i,j}}^{v}v_{i,j+1}\\ +\left(p_{i,j}-p_{i,j+1}\right)\left(\Delta y\right)_{j}^{v}+c_{i,j}^{v}\end{Bmatrix}\end{aligned} \tag{4.12}$$

In Eqs. (4.11) and (4.12), $p_{i,j}$ refers to nodes (i, j) of the main grid (*p*-grid), shown

in Figure 4.1. $u_{i,j}$ and $v_{i,j}$ refer to nodes (i, j) of the staggered grids (u- and v-grid respectively), shown in Figures 4.2 and 4.3. Coefficients a_W, a_E, a_S, and a_N are dependent on velocity field, thermo-physical properties, and grid size. Formulae for these coefficients are based on the discretization scheme chosen. The term c arises when higher-order discretization schemes are used in conjunction with deferred-correction technique introduced by Khosla and Rubin [65]. Appendix B gives expressions for the term c and the coefficients a_W, a_E, a_S, and a_N for power law scheme of Patankar [1] and QUICK scheme of Leonard [66].

Eqs. (4.11) and (4.12) are ordinary differential equations in time. These are evolution equations for u- and v-velocity fields. At any instant in time, if velocity and pressure fields are known, time-derivative of u- and v-velocity can be calculated using these equations.

4.2.2. Temporal Discretization

In the current work, ESDIRK methods are used for temporal discretization (refer to Section 3). Referring to Appendix A, in a stiffly accurate ESDIRK method, u-velocity at n^{th} time-step at any grid node is calculated from Eqs. (A.16) through (A.18) which transform to the following three equations:

$$u_{n,1} = u_{n,0} \tag{4.13}$$

$$u_{n,r} = u_{n,0} + h\sum_{s=1}^{r} A_{rs} f^{u}\left(t,u,v,p\right)_{n,s}, \qquad r = 2,\ q \tag{4.14}$$

$$u_{n+1} = u_{n,q} \tag{4.15}$$

In Eqs. (4.13) through (4.15), the indices n, r refer to r^{th} stage of n^{th} time step and should not be confused with indices of the grid points. Parameters A_{rs} in Eq. (4.14) are the weights used in *stage calculations*. These parameters are taken from the Butcher array of the chosen RK method.

Re-writing Eq. (4.14) with a little re-arrangement:

$$u_{n,r} = u_{n,0} + h\sum_{s=1}^{r-1} A_{rs} f_{n,s}^{u} + hA_{rr} f_{n,r}^{u}, \qquad r = 2,\ q \tag{4.16}$$

At any r^{th} stage, the first two terms on the right hand side of Eq. (4.16) are explicit terms. The first term is known from the previous time step. The second term can be calculated, using Eq. (4.11), from the values of u calculated in the preceding stages. However, the third term is an implicit term because $f_{n,r}$ is dependent on $u_{n,r}$. Inserting Eq. (4.11) into the third term on the right hand side of Eq. (4.16); for n^{th} time step at r^{th} stage, we get:

$$\begin{aligned}\left(u_{i,j}\right)_r = \left(u_{i,j}\right)_0 &+ h\sum_{s=1}^{r-1} A_{rs}\left(f_{i,j}^{u}\right)_s \\ &+ \frac{hA_{rr}}{\rho\left(\Delta x\right)_i^u\left(\Delta y\right)_j^p}\Big\{-\left(a_{W_{i,j}}^{u} + a_{E_{i,j}}^{u} + a_{S_{i,j}}^{u} + a_{N_{i,j}}^{u}\right)u_{i,j} \\ &\qquad + a_{W_{i,j}}^{u} u_{i-1,j} + a_{E_{i,j}}^{u} u_{i+1,j} + a_{S_{i,j}}^{u} u_{i,j-1} + a_{N_{i,j}}^{u} u_{i,j+1} \\ &\qquad + \left(p_{i,j} - p_{i+1,j}\right)\left(\Delta y\right)_j^p + c_{i,j}^{u}\Big\}_r, \quad r = 2,\ q\end{aligned} \tag{4.17}$$

where subscript *n* is omitted for clarity.

Re-arranging terms and omitting subscript *r* from *u*-velocity terms:

$$\begin{aligned}\left(a^u_{W_{i,j}} + a^u_{E_{i,j}} + a^u_{S_{i,j}} + a^u_{N_{i,j}} + \rho\left(\Delta x\right)^u_i \left(\Delta y\right)^p_j / hA_{rr}\right)u_{i,j} \\ = a^u_{W_{i,j}} u_{i-1,j} + a^u_{E_{i,j}} u_{i+1,j} + a^u_{S_{i,j}} u_{i,j-1} + a^u_{N_{i,j}} u_{i,j+1} \\ + \left(p_{i,j} - p_{i+1,j}\right)\left(\Delta y\right)^p_j \\ + c^u_{i,j} + \left\{\left(u_{i,j}\right)_0 + h\sum_{s=1}^{r-1} A_{rs}\left(f^u_{i,j}\right)_s\right\}\frac{\rho\left(\Delta x\right)^u_i \left(\Delta y\right)^p_j}{hA_{rr}}, \quad r = 2,\, q\end{aligned} \tag{4.18}$$

Re-writing Eq. (4.18) in a condensed form:

$$a^u_{i,j} u_{i,j} = a^u_{W_{i,j}} u_{i-1,j} + a^u_{E_{i,j}} u_{i+1,j} + a^u_{S_{i,j}} u_{i,j-1} + a^u_{N_{i,j}} u_{i,j+1} + \left(p_{i,j} - p_{i+1,j}\right)\left(\Delta y\right)^p_j + b^u_{i,j} \tag{4.19}$$

Corresponding equation for *v*-velocity is:

$$a^v_{i,j} v_{i,j} = a^v_{W_{i,j}} v_{i-1,j} + a^v_{E_{i,j}} v_{i+1,j} + a^v_{S_{i,j}} v_{i,j-1} + a^v_{N_{i,j}} v_{i,j+1} + \left(p_{i,j} - p_{i+1,j}\right)\left(\Delta x\right)^p_i + b^v_{i,j} \tag{4.20}$$

where

$$a^u_{i,j} = a^u_{W_{i,j}} + a^u_{E_{i,j}} + a^u_{S_{i,j}} + a^u_{N_{i,j}} + \rho\left(\Delta x\right)^u_i \left(\Delta y\right)^p_j / hA_{rr}, \qquad r = 2,\, q \tag{4.21}$$

$$b^u_{i,j} = c^u_{i,j} + \left\{\left(u_{i,j}\right)_0 + h\sum_{s=1}^{r-1} A_{rs}\left(f^u_{i,j}\right)_s\right\}\rho\left(\Delta x\right)^u_i \left(\Delta y\right)^p_j / hA_{rr}, \qquad r = 2,\, q \tag{4.22}$$

$$a^v_{i,j} = a^v_{W_{i,j}} + a^v_{E_{i,j}} + a^v_{S_{i,j}} + a^v_{N_{i,j}} + \rho\left(\Delta x\right)^p_i \left(\Delta y\right)^v_j / hA_{rr}, \qquad r = 2,\, q \tag{4.23}$$

$$b^v_{i,j} = c^v_{i,j} + \left\{ \left(v_{i,j} \right)_0 + h \sum_{s=1}^{r-1} A_{rs} \left(f^v_{i,j} \right)_s \right\} \rho \left(\Delta x \right)^p_i \left(\Delta y \right)^v_j / hA_{rr}, \qquad r = 2, \, q \tag{4.24}$$

Eqs. (4.19) and (4.20) are the discretized forms of Eqs. (4.5) and (4.6), respectively, when staggered grid approach is used. Coefficients a_W, a_E, a_S, and a_N are evaluated based on the spatial discretization scheme of choice. Since coefficients of Eqs. (4.19) and (4.20) are dependent on the dependent variables u and v, these equations are non-linear equations, and therefore, require an iterative method for their solution. The solution method is explained in the next subsection.

4.2.3. Simultaneous Solution of Mass Conservation and Momentum Conservation Equations

At any n^{th} time step, Eqs. (4.8), (4.19), and (4.20) are required to be solved simultaneously at every r^{th} stage of a DIRK method. Therefore, in a q-stage DIRK method, q number of iterative solutions is required in every time step. However, in an ESDIRK method, the first stage velocity field is explicitly given by Eq. (4.13) and similar equation for v-velocity. Therefore, in a q-stage ESDIRK method, q–1 (one less than q) number of iterative solutions is required in every time step. Time advancement is shown schematically in Figure 4.4. SIMPLE algorithm (Patankar [1]) is used for simultaneous solution of Eqs. (4.8), (4.19), and (4.20). In this algorithm, velocity and pressure fields are required to be corrected in every iteration. The corrections are calculated from a pressure correction equation which is derived in Section 4.2.3b. Moreover, in order to ensure

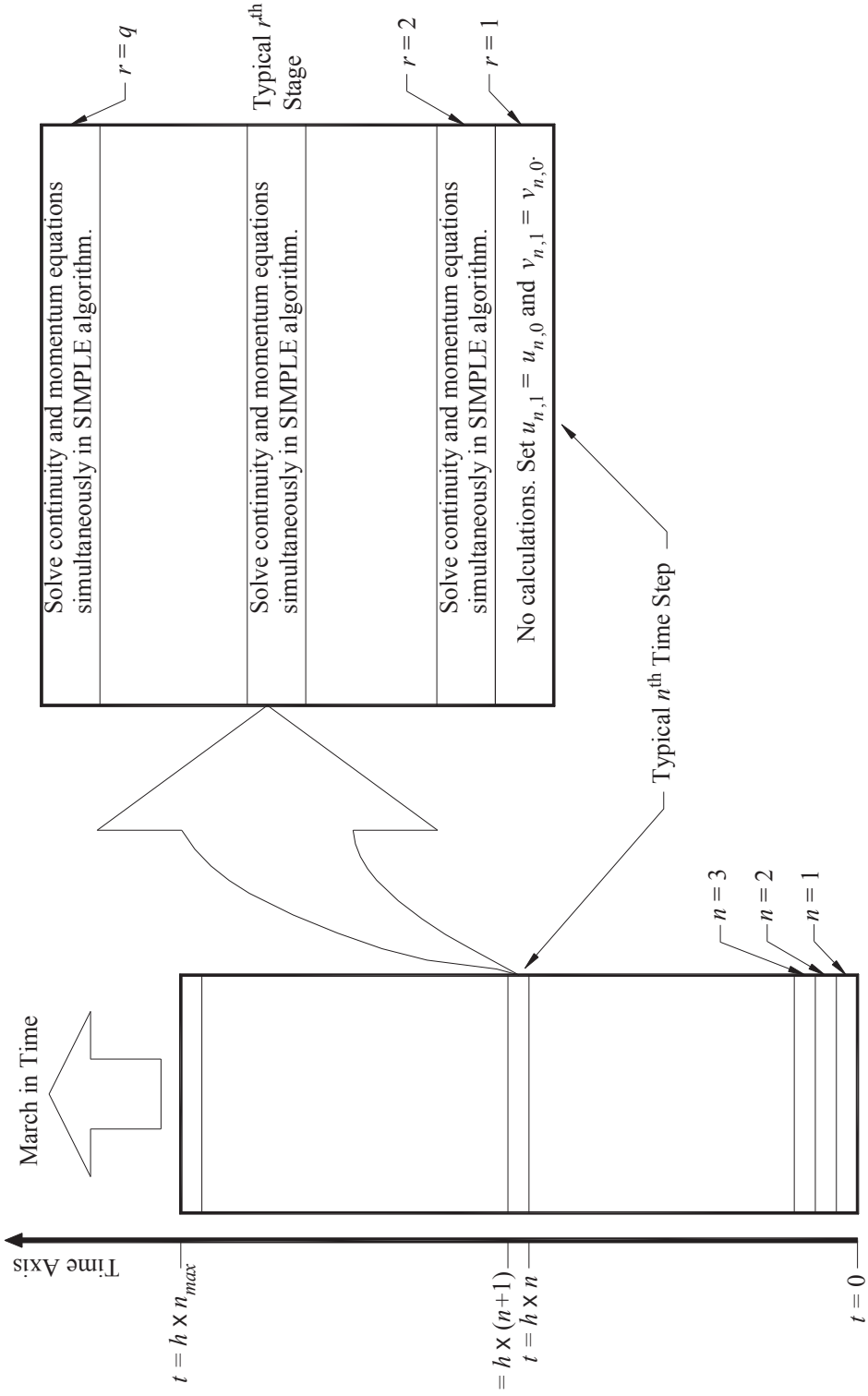


Figure 4.4 Schematic Representation of Time Advancement

convergence, the iterative solution of momentum equations is under-relaxed, as explained in Section 4.2.3c. Convergence of solution is monitored in every iteration by comparing relative residuals for mass conservation, *u*-velocity, and *v*-velocity with some specified values; relative residuals are defined in Section 4.2.3d.

The proceeding subsection explains how the velocities are calculated at the faces of control volumes. These velocities are required to calculate coefficients in the discretized equations Eqs. (4.19) and (4.20).

4.2.3a. Evaluation of CV Face Velocities

Since the discretized equations, Eqs. (4.19) and (4.20), were obtained from the integral form of model governing equations, formulae derived for the coefficients a_W, a_E, a_S, and a_N using any general discretization scheme involve velocities at the CV faces. In the staggered-grid approach, the velocities at the faces of *u*- and *v*-CVs are calculated from the velocities at the *u*- and *v*-grid nodes by some interpolation method, e.g. linear and quadratic interpolation. The interpolation method is chosen based on the order of accuracy desired. Appendix B gives formulae for *u*-velocity at the east face of a *u*-CV when QUICK scheme of Leonard [66] is used. Different formulae will arise if a different spatial discretization scheme is used.

4.2.3b. Correcting Velocity and Pressure Fields by Enforcing Mass Conservation

During the iterative solution process before convergence is reached, the velocity fields calculated from momentum equations do not satisfy continuity equation, Eq. (4.8). Corrections are applied to the calculated velocity fields after every iteration until convergence is achieved. Let p^* be a guessed or incorrect pressure field which is used in

the solution of Eqs. (4.19) and (4.20); the resulting velocity field can be denoted as u^* and v^*. Then corrections required in velocity and pressure fields are:

$$u' = u - u^* \tag{4.25}$$

$$v' = v - v^* \tag{4.26}$$

$$p' = p - p^* \tag{4.27}$$

From Eq. (4.19):

$$a^u_{i,j} u^*_{i,j} = a^u_{W_{i,j}} u^*_{i-1,j} + a^u_{E_{i,j}} u^*_{i+1,j} + a^u_{S_{i,j}} u^*_{i,j-1} + a^u_{N_{i,j}} u^*_{i,j+1} + \left(p^*_{i,j} - p^*_{i+1,j} \right) \left(\Delta y \right)^p_j + b^u_{i,j} \tag{4.28}$$

Subtracting Eq. (4.28) from Eq. (4.19):

$$a^u_{i,j} u'_{i,j} = a^u_{W_{i,j}} u'_{i-1,j} + a^u_{E_{i,j}} u'_{i+1,j} + a^u_{S_{i,j}} u'_{i,j-1} + a^u_{N_{i,j}} u'_{i,j+1} + \left(p'_{i,j} - p'_{i+1,j} \right) \left(\Delta y \right)^p_j \tag{4.29}$$

Omission of the first four terms in Eq. (4.29) and a little re-arrangement result in the following equation (justification for this omission is explained by Patankar [1]):

$$u'_{i,j} = \frac{\left(\Delta y \right)^p_j}{a^u_{i,j}} \left(p'_{i,j} - p'_{i+1,j} \right) \tag{4.30}$$

Similar equation is obtained for v':

$$v'_{i,j} = \frac{(\Delta x)_i^p}{a_{i,j}^v}\left(p'_{i,j} - p'_{i,j+1}\right) \tag{4.31}$$

Substituting Eqs. (4.30) and (4.31) into Eqs. (4.25) and (4.26):

$$u_{i,j} = u^*_{i,j} + \frac{(\Delta y)_j^p}{a_{i,j}^u}\left(p'_{i,j} - p'_{i+1,j}\right) \tag{4.32}$$

$$v_{i,j} = v^*_{i,j} + \frac{(\Delta x)_i^p}{a_{i,j}^v}\left(p'_{i,j} - p'_{i,j+1}\right) \tag{4.33}$$

Now substituting Eqs. (4.32) and (4.33) into Eq. (4.8) and re-arranging terms, we get the pressure correction equation:

$$a_{i,j}^{pc} p'_{i,j} = a_{W_{i,j}}^{pc} p'_{i-1,j} + a_{E_{i,j}}^{pc} p'_{i+1,j} + a_{S_{i,j}}^{pc} p'_{i,j-1} + a_{N_{i,j}}^{pc} p'_{i,j+1} + S_{i,j}^{pc} \tag{4.34}$$

where

$$a_{W_{i,j}}^{pc} = \rho\left(\frac{(\Delta y)_j^p}{a_{i-1,j}^u}\right)(\Delta y)_j^p \tag{4.35}$$

$$a_{E_{i,j}}^{pc} = \rho\left(\frac{(\Delta y)_j^p}{a_{i,j}^u}\right)(\Delta y)_j^p \tag{4.36}$$

$$a_{S_{i,j}}^{pc} = \rho\left(\frac{(\Delta x)_i^p}{a_{i,j-1}^v}\right)(\Delta x)_i^p \tag{4.37}$$

$$a_{N_{i,j}}^{pc} = \rho\left(\frac{(\Delta x)_i^p}{a_{i,j}^v}\right)(\Delta x)_i^p \tag{4.38}$$

$$a_{i,j}^{pc} = a_{W_{i,j}}^{pc} + a_{E_{i,j}}^{pc} + a_{S_{i,j}}^{pc} + a_{N_{i,j}}^{pc} \tag{4.39}$$

$$S_{i,j}^{pc} = \rho\left(u_{i-1,j}^* - u_{i,j}^*\right)(\Delta y)_j^p + \rho\left(v_{i,j-1}^* - v_{i,j}^*\right)(\Delta x)_i^u \tag{4.40}$$

4.2.3c. Under-relaxation

Since Eqs. (4.19) and (4.20) are non-linear, their solution needs to be under-relaxed to ensure convergence. Let u^l represent u-velocity at the preceding iteration; incorporating under-relaxation into Eq. (4.19):

$$u_{i,j} = \frac{\alpha^u}{a_{i,j}^u}\left\{a_{W_{i,j}}^u u_{i-1,j} + a_{E_{i,j}}^u u_{i+1,j} + a_{S_{i,j}}^u u_{i,j-1} + a_{N_{i,j}}^u u_{i,j+1} + \left(p_{i,j} - p_{i+1,j}\right)(\Delta y)_j^p + b_{i,j}^u\right\} + \left(1-\alpha^u\right)u_{i,j}^l \tag{4.41}$$

Re-arranging Eq. (4.41):

$$a_{i,j}^u u_{i,j} = a_{W_{i,j}}^u u_{i-1,j} + a_{E_{i,j}}^u u_{i+1,j} + a_{S_{i,j}}^u u_{i,j-1} + a_{N_{i,j}}^u u_{i,j+1} + \left(p_{i,j} - p_{i+1,j}\right)(\Delta y)_j^p + S_{i,j}^u \tag{4.42}$$

Corresponding equation for v-velocity is:

$$a^v_{i,j} v_{i,j} = a^v_{W_{i,j}} v_{i-1,j} + a^v_{E_{i,j}} v_{i+1,j} + a^v_{S_{i,j}} v_{i,j-1} + a^v_{N_{i,j}} v_{i,j+1} + \left(p_{i,j} - p_{i+1,j}\right)\left(\Delta x\right)^p_i + S^v_{i,j} \tag{4.43}$$

S^u and S^v in Eqs. (4.42) and (4.43) are given by:

$$S^u_{i,j} = b^u_{i,j} + \left(\frac{1}{\alpha^u} - 1\right) a^u_{i,j} u^l_{i,j} \tag{4.44}$$

$$S^v_{i,j} = b^v_{i,j} + \left(\frac{1}{\alpha^v} - 1\right) a^v_{i,j} v^l_{i,j} \tag{4.45}$$

where b^u and b^v are calculated from Eqs. (4.22) and (4.24), respectively.

Pressure correction is also under-relaxed as given below:

$$p = p^* + \alpha^p p' \tag{4.46}$$

4.2.3d. Convergence Criteria

During the simultaneous iterative solution of Eqs. (4.8), (4.19), and (4.20), the solution is checked for convergence during every iteration by evaluating relative residuals and comparing them with some chosen values. The relative residuals for mass and momentum conservation equations are calculated as given below.

Relative residual for mass conservation:

$$res_{\text{mass conservation}} = \frac{\sum_{p\text{-CVs}} \left| \rho\left(u_{i,j} - u_{i-1,j}\right)\left(\Delta y\right)^p_j + \rho\left(v_{i,j} - v_{i,j-1}\right)\left(\Delta x\right)^u_i \right|}{\rho u_c l_c} \tag{4.47}$$

where u_c and l_c are some characteristic values of velocity and length. Choice for

these characteristic values depends on nature of the problem under consideration; for example in case of lid-driven square cavity flow, these values can be lid speed and cavity height.

Relative residual for *u*-velocity:

$$res_{u-\text{velocity}} = \frac{1}{\sum_{u\text{-CVs}} \left| a^u_{i,j} u_{i,j} \right|} \sum_{u\text{-CVs}} \left| a^u_{i,j} u_{i,j} - \left(a^u_{W_{i,j}} u_{i-1,j} + a^u_{E_{i,j}} u_{i+1,j} + a^u_{S_{i,j}} u_{i,j-1} + a^u_{N_{i,j}} u_{i,j+1} \right.\right.$$
$$\left.\left. \left(p_{i,j} - p_{i+1,j} \right) \left(\Delta y \right)^p_j + S^u_{i,j} \right) \right| \quad (4.48)$$

Relative residual for *v*-velocity:

$$res_{v-\text{velocity}} = \frac{1}{\sum_{v\text{-CVs}} \left| a^v_{i,j} v_{i,j} \right|} \sum_{v\text{-CVs}} \left| a^v_{i,j} v_{i,j} - \left(a^v_{W_{i,j}} v_{i-1,j} + a^v_{E_{i,j}} v_{i+1,j} + a^v_{S_{i,j}} v_{i,j-1} + a^v_{N_{i,j}} v_{i,j+1} \right.\right.$$
$$\left.\left. + \left(p_{i,j} - p_{i+1,j} \right) \left(\Delta x \right)^p_i + S^v_{i,j} \right) \right| \quad (4.49)$$

4.2.3e. Algorithm

1. Assign initial values to velocity and pressure fields. These initial fields are also taken as initial guess for the subsequent iterative solution.
2. Set boundary conditions.
3. Set $n = 1$.
4. Set $u_{n,0}$ and $v_{n,0}$ equal to initial velocity field.

n^{th} Time-Step ($n = 1, n_{max}$):

1^{st} Stage ($r = 1$):

5. Set $u_{n,1} = u_{n,0}$ and $v_{n,1} = v_{n,0}$ (Eq. (4.13) and the corresponding equation for *v*-

velocity).

r^{th} Stage ($r = 2$, q):

6. Calculate coefficients a_W, a_E, a_S, and a_N using the velocity fields $u_{n,r\text{-}1}$ and $v_{n,r\text{-}1}$ with a spatial discretization scheme of choice.
7. Calculate time-derivatives $f^u_{n,s}$ and $f^v_{n,s}$ (for $s = 1, r\text{-}1$) from Eqs. (4.11) and (4.12) respectively.
8. Calculate b^u and b^v from Eqs. (4.22) and (4.24) respectively.

Iteration for u, v, and p:

9. Solve Eqs. (4.19) and (4.20) with some solution algorithm such as line-by-line procedure which is a combination of Tri-diagonal Matrix Algorithm (TDMA) and Gauss-Seidel scheme.
10. Calculate pressure correction from Eq. (4.34).
11. Correct u- and v- velocity fields using Eqs. (4.32) and (4.33).
12. Correct the pressure field using Eq. (4.46).
13. Calculate coefficients a_W, a_E, a_S, and a_N using the velocity fields calculated in step 11.
14. Calculate residuals from Eqs. (4.47) through (4.49). Check for convergence by comparing the residuals with some chosen values.
15. If solution is converged, go to step 17.
16. If solution is not converged, go to step 9.
17. Check the value of r.
18. If $r = q$, go to step 20.

19. If $r < q$, switch to next stage, i.e., set $r = r + 1$. Go to step 6.
20. Check the value of n.
21. If $n < n_{max}$, switch to the next time-step, i.e., set $n = n + 1$. Set $u_{n,0}$ and $v_{n,0}$ equal to the velocity fields calculated in step 11. Go to step 5.
22. If $n = n_{max}$, stop the program.

The above solution algorithm is shown as a flow chart in Figure 4.5.

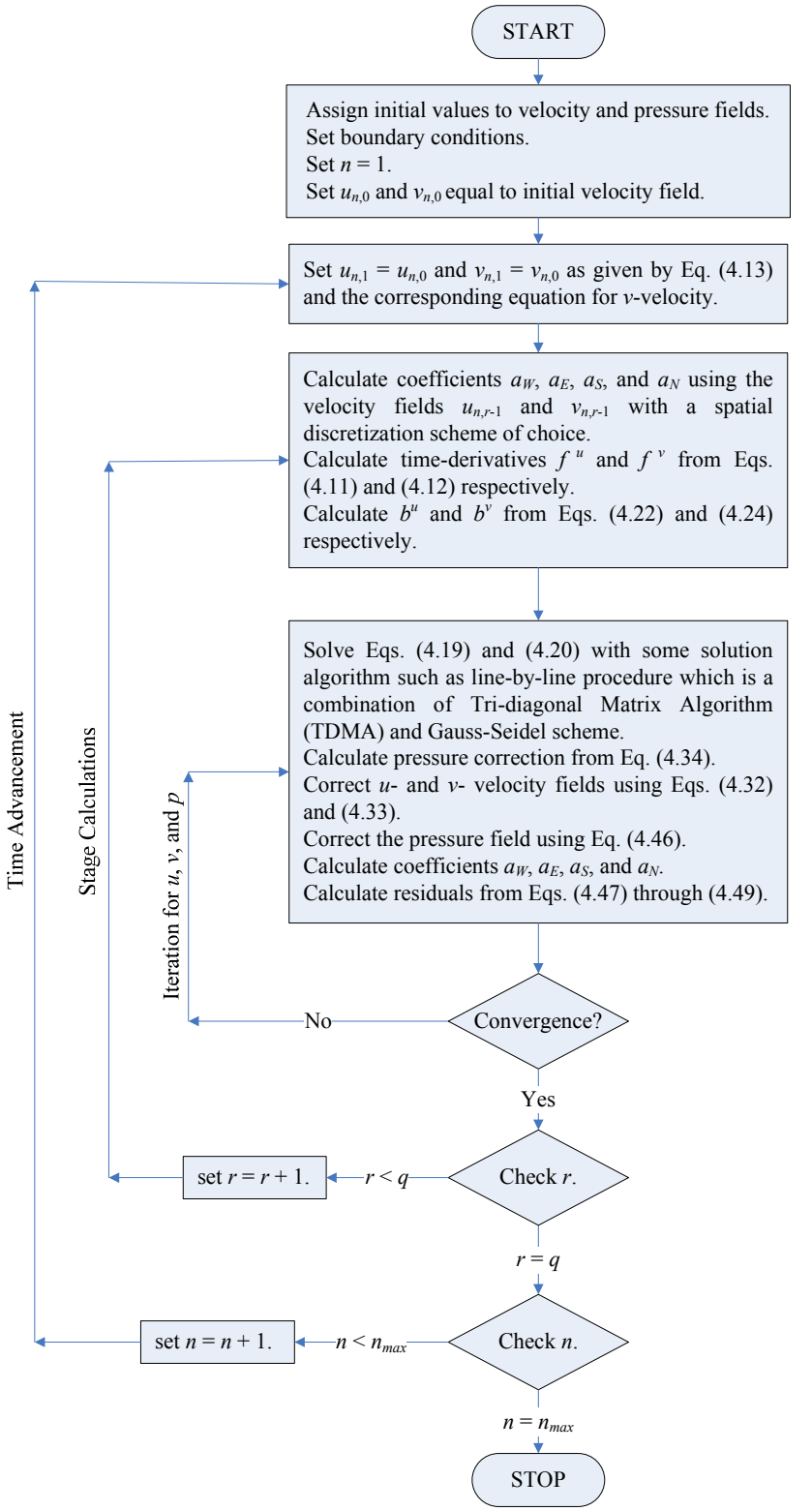


Figure 4.5 Solution Algorithm for Staggered Grid Method

4.3. Non-staggered Grid Approach (or Co-located Variables Approach)

Due to larger memory and computational effort requirement associated with the staggered grid approach, it has always been desirable to use a non-staggered grid wherein all variables can be calculated and stored at the nodes of a single grid. Moreover, the staggered grid approach is not suitable for use with complex geometries that involve internal bluff regions and require unstructured grids. Rhie and Chow [67] proposed a method to overcome the difficulties involved in the use of a non-staggered grid. The method was later improved by other researchers.

The above considerations motivated the current author to extend the proposed SIMPLE DIRK method to non-staggered grid approach. The proceeding subsections explain spatial and temporal discretization of mass conservation and momentum conservation equations for non-staggered grid approach. Subsequently, simultaneous solution of the discretized equations is discussed.

4.3.1. Spatial Discretization

In the co-located variables approach, pressure and velocities are calculated at the geometric center of control volumes (CVs). Both the mass conservation equation (Eq. (4.3)) and the momentum conservation equation (Eq. (4.4)) are integrated over the same CVs; a typical CV is shown in Figure 4.6.

4.3.1a. Mass Conservation Equation

Assuming that at any point on the face of the CV, velocity remains constant and equal to its value at the center of the face, discretization of Eq. (4.7) gives:

$$\left(u_e - u_w\right)_{i,j}\left(\Delta y\right)_j + \left(v_n - v_s\right)_{i,j}\left(\Delta x\right)_i = 0 \tag{4.50}$$

Eq. (4.50) is discretized form of mass conservation equation, Eq. (4.3), for a non-staggered grid node (i, j).

4.3.1b. Momentum Conservation Equations

Integrating Eqs. (4.5) and (4.6) over a CV, shown in Figure 4.6, and applying divergence theorem, we get equations similar to Eqs. (4.9) and (4.10):

$$\rho\int_{\mathrm{V}}\frac{\partial u}{\partial t}d\mathrm{V} = -\underbrace{\rho\int_{\mathrm{S}}u\vec{V}.d\vec{\mathrm{S}}}_{\text{Convection Term}} + \underbrace{\int_{\mathrm{S}}\mu\nabla u.d\vec{\mathrm{S}}}_{\text{Diffusion Term}} - \underbrace{\int_{\mathrm{V}}\nabla p d\mathrm{V}}_{\text{Source Term}} \tag{4.51}$$

$$\rho\int_{\mathrm{V}}\frac{\partial v}{\partial t}d\mathrm{V} = -\underbrace{\rho\int_{\mathrm{S}}v\vec{V}.d\vec{\mathrm{S}}}_{\text{Convection Term}} + \underbrace{\int_{\mathrm{S}}\mu\nabla v.d\vec{\mathrm{S}}}_{\text{Diffusion Term}} - \underbrace{\int_{\mathrm{V}}\nabla p d\mathrm{V}}_{\text{Source Term}} \tag{4.52}$$

For any node (i, j) in non-staggered grid approach, spatial discretization of Eq. (4.51) and (4.52) results in evolution equations for u- and v-velocity fields:

$$\begin{aligned}\left(\partial u/\partial t\right)_{i,j} &= f^u\left(t,u,v,p\right)_{i,j}\\ &= \frac{1}{\rho\left(\Delta x\right)_i\left(\Delta y\right)_j}\begin{Bmatrix}-\left(a_{W_{i,j}} + a_{E_{i,j}} + a_{S_{i,j}} + a_{N_{i,j}}\right)u_{i,j}\\ +a_{W_{i,j}}u_{i-1,j} + a_{E_{i,j}}u_{i+1,j} + a_{S_{i,j}}u_{i,j-1} + a_{N_{i,j}}u_{i,j+1}\\ \left(p_w - p_e\right)_{i,j}\left(\Delta y\right)_j + c^u_{i,j}\end{Bmatrix}\end{aligned} \tag{4.53}$$

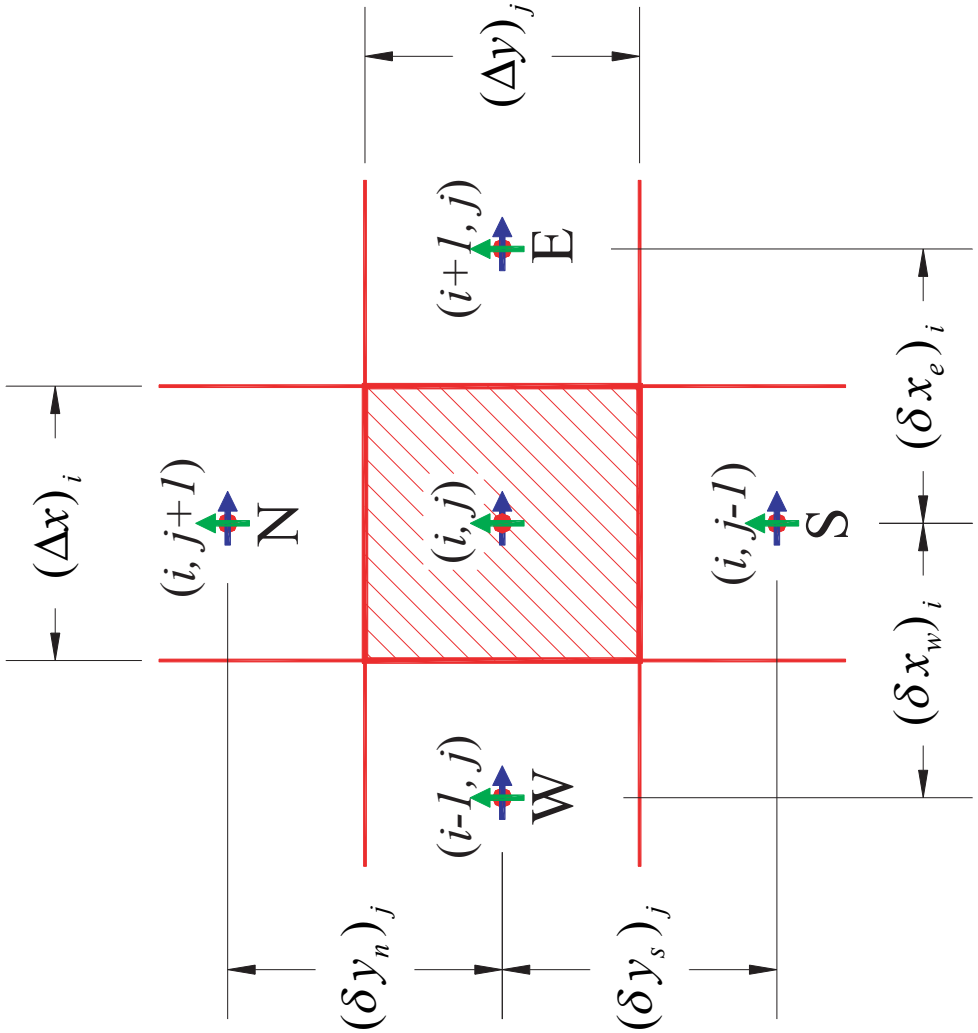


Figure 4.6 Typical Control Volume in Non-staggered Grid Approach

$$\left(\partial v/\partial t\right)_{i,j} = f^{v}\left(t,u,v,p\right)_{i,j}$$
$$= \frac{1}{\rho\left(\Delta x\right)_i\left(\Delta y\right)_j}\begin{Bmatrix} -\left(a_{W_{i,j}} + a_{E_{i,j}} + a_{S_{i,j}} + a_{N_{i,j}}\right)v_{i,j} \\ +a_{W_{i,j}}v_{i-1,j} + a_{E_{i,j}}v_{i+1,j} + a_{S_{i,j}}v_{i,j-1} + a_{N_{i,j}}v_{i,j+1} \\ \left(p_s - p_n\right)_{i,j}\left(\Delta x\right)_i + c^{v}_{i,j} \end{Bmatrix} \qquad (4.54)$$

In Eqs. (4.53) and (4.54), $p_{i,j}$, $u_{i,j}$, and $v_{i,j}$ refer to node (i, j) of the non-staggered grid, shown in Figure 4.6. Formulae for the velocity-field-dependent coefficients a_W, a_E, a_S, and a_N are based on the discretization scheme chosen. In the non-staggered grid approach, these coefficients are identical for both the evolution equations, Eqs. (4.53) and (4.54). Given in Appendix B are expressions for the deferred-correction term c and the coefficients a_W, a_E, a_S, and a_N for power law scheme of Patankar [1] and QUICK scheme of Leonard [66].

4.3.2. Temporal Discretization

In Section 4.2.2, Eq. (4.16) was written for a u-grid node. Now we consider Eq. (4.16) for a typical node of a non-staggered grid. Inserting Eq. (4.53) into the third term on the right hand side of Eq. (4.16); for n^{th} time step at r^{th} stage, we get:

$$\left(u_{i,j}\right)_r = \left(u_{i,j}\right)_0 + h\sum_{s=1}^{r-1} A_{rs}\left(f^{u}_{i,j}\right)_s$$
$$+ \frac{hA_{rr}}{\rho\left(\Delta x\right)_i\left(\Delta y\right)_j}\Big\{-\left(a_{W_{i,j}} + a_{E_{i,j}} + a_{S_{i,j}} + a_{N_{i,j}}\right)u_{i,j}$$
$$+ a_{W_{i,j}}u_{i-1,j} + a_{E_{i,j}}u_{i+1,j} + a_{S_{i,j}}u_{i,j-1} + a_{N_{i,j}}u_{i,j+1}$$
$$+ \left(p_w - p_e\right)_{i,j}\left(\Delta y\right)_j + c^{u}_{i,j}\Big\}_r, \quad r = 2,\, q \qquad (4.55)$$

where subscript *n* is omitted for clarity.

Re-arranging terms and omitting subscript *r* from *u*-velocity terms:

$$\begin{aligned}\left(a_{W_{i,j}}+a_{E_{i,j}}+a_{S_{i,j}}+a_{N_{i,j}}+\rho(\Delta x)_i(\Delta y)_j/hA_{rr}\right)u_{i,j}&\\ =a_{W_{i,j}}u_{i-1,j}+a_{E_{i,j}}u_{i+1,j}+a_{S_{i,j}}u_{i,j-1}+a_{N_{i,j}}u_{i,j+1}&\\ +(p_w-p_e)_{i,j}(\Delta y)_j+c^u_{i,j}&\\ +\left\{(u_{i,j})_0+h\sum_{s=1}^{r-1}A_{rs}(f^u_{i,j})_s\right\}\frac{\rho(\Delta x)_i(\Delta y)_j}{hA_{rr}},\quad r=2,\,q&\end{aligned}\tag{4.56}$$

Re-writing Eq. (4.56) in a condensed form, we get the discretized form of *x*-component of momentum conservation equation, Eq. (4.5):

$$a_{i,j}u_{i,j}=a_{W_{i,j}}u_{i-1,j}+a_{E_{i,j}}u_{i+1,j}+a_{S_{i,j}}u_{i,j-1}+a_{N_{i,j}}u_{i,j+1}+(p_w-p_e)_{i,j}(\Delta y)_j+b^u_{i,j}\tag{4.57}$$

where

$$a_{i,j}=a_{W_{i,j}}+a_{E_{i,j}}+a_{S_{i,j}}+a_{N_{i,j}}+\rho(\Delta x)_i(\Delta y)_j/hA_{rr},\qquad r=2,\,q\tag{4.58}$$

$$b^u_{i,j}=c^u_{i,j}+\left\{(u_{i,j})_0+h\sum_{s=1}^{r-1}A_{rs}(f^u_{i,j})_s\right\}\rho(\Delta x)_i(\Delta y)_j/hA_{rr},\qquad r=2,\,q\tag{4.59}$$

An equation similar to Eq. (4.57) is obtained from discretization of *y*-component of momentum conservation equation, Eq. (4.6):

$$a_{i,j}v_{i,j} = a_{W_{i,j}}v_{i-1,j} + a_{E_{i,j}}v_{i+1,j} + a_{S_{i,j}}v_{i,j-1} + a_{N_{i,j}}v_{i,j+1} + \left(p_s - p_n\right)_{i,j}\left(\Delta x\right)_i + b^v_{i,j} \qquad (4.60)$$

where $a_{i,j}$ are given by Eq. (4.58); $b^v_{i,j}$ are calculated as below:

$$b^v_{i,j} = c^v_{i,j} + \left\{\left(v_{i,j}\right)_0 + h\sum_{s=1}^{r-1} A_{rs}\left(f^v_{i,j}\right)_s\right\}\rho\left(\Delta x\right)_i\left(\Delta y\right)_j / hA_{rr}, \qquad r = 2,\, q \qquad (4.61)$$

Eqs. (4.57) and (4.60) are the discretized forms of Eqs. (4.5) and (4.6), respectively, when non-staggered grid approach is used. The iterative solution method required for these non-linear equations is explained in the following subsection.

4.3.3. Simultaneous Solution of Mass Conservation and Momentum Conservation Equations

In any n^{th} time step, Eqs. (4.50), (4.57), and (4.60) are required to be solved simultaneously at every r^{th} stage of an ESDIRK method. Like staggered grid approach, SIMPLE algorithm (Patankar [1]) is used for simultaneous solution of Eqs. (4.50), (4.57), and (4.60). q–1 (one less than q) number of iterative solutions is required in every time step in a q-stage ESDIRK method. Time advancement is shown schematically in Figure 4.4.

4.3.3a. Evaluation of CV Face Velocities

Since the discretized equations, Eqs. (4.57) and (4.60), were obtained from the integral form of model governing equations, formulae derived for the coefficients a_W, a_E, a_S, and a_N using any discretization scheme involve velocities at the CV faces. In the non-

staggered grid approach, the velocities at the faces of *u*- and *v*-CVs are calculated from an interpolation method similar to the one proposed by Rhie and Chow [67]. This method has been called momentum interpolation method. Later, Majumdar [68] and Choi [69] pointed out some problems in the original method and proposed improvements. The interpolation method in the current work is described below.

Incorporating under-relaxation into Eq. (4.57) and using Eq. (4.59), for a node (i, j) we get:

$$u_{i,j} = (1-\alpha)\left(u_{i,j}\right)_l + \frac{\alpha}{a_{i,j}}\left[H^u_{i,j} + \left(p_w - p_e\right)_{i,j}\left(\Delta y\right)_j + c^u_{i,j} + \left\{ \begin{matrix} \left(u_{i,j}\right)_0 \\ +h\sum_{s=1}^{r-1} A_{rs}\left(f^u_{i,j}\right)_s \end{matrix} \right\} \frac{\rho\left(\Delta x\right)_i\left(\Delta y\right)_j}{hA_{rr}} \right], \qquad r = 2,\ q \tag{4.62}$$

where

$$H^u_{i,j} = a_{W_{i,j}} u_{i-1,j} + a_{E_{i,j}} u_{i+1,j} + a_{S_{i,j}} u_{i,j-1} + a_{N_{i,j}} u_{i,j+1} \tag{4.63}$$

Similarly for the node $(i+1, j)$:

$$u_{i+1,j} = (1-\alpha)\left(u_{i+1,j}\right)_l + \frac{\alpha}{a_{i+1,j}}\left[\begin{array}{l} H^u_{i+1,j} + \left(p_w - p_e\right)_{i+1,j}\left(\Delta y\right)_j + c^u_{i+1,j} \\ + \left\{\left(u_{i+1,j}\right)_0 + h\sum_{s=1}^{r-1} A_{rs}\left(f^u_{i+1,j}\right)_s\right\}\dfrac{\rho\left(\Delta x\right)_{i+1}\left(\Delta y\right)_j}{hA_{rr}} \end{array}\right], \quad r = 2,\ q \tag{4.64}$$

where

$$H^u_{i+1,j} = a_{W_{i+1,j}} u_{i+1,j} + a_{E_{i+1,j}} u_{i+2,j} + a_{S_{i+1,j}} u_{i+1,j-1} + a_{N_{i,j}} u_{i+1,j+1} \tag{4.65}$$

u-velocity at the east face of the CV corresponding to the node (i, j) can be obtained by interpolating selected terms in the above equations. Consider an imaginary *u*-CV enclosing the geometric center of the east face of the CV corresponding to the node (i, j). Geometric centers of the west and east faces of this imaginary *u*-CV coincide with the nodes (i, j) and (i+1, j) respectively. Therefore the term ($p_w - p_e$) becomes ($p_{i,j} - p_{i+1,j}$). Other terms are interpolated or taken from either the previous iteration or the preceding time-step.

Based on the above discussion, *u*-velocity at the east face of the CV corresponding to the node (i, j) is written as:

$$u_{e_{i,j}} = (1-\alpha)\left(u_{e_{i,j}}\right)_l + \frac{\alpha}{\overline{a}_{e_{i,j}}}\Bigg[\overline{H}^u_{e_{i,j}} + \left(p_{i,j} - p_{i+1,j}\right)\left(\Delta y\right)_j + \overline{c}^u_{e_{i,j}} + \left\{\left(u_{e_{i,j}}\right)_0 + h\sum_{s=1}^{r-1} A_{rs}\left(\overline{f}^u_{e_{i,j}}\right)_s\right\}\frac{\rho\left(\delta x_e\right)_i\left(\Delta y\right)_j}{hA_{rr}}\Bigg], \quad r = 2,\ q \tag{4.66}$$

Equations similar to Eq. (4.66) can be written also for $u_{w_{i,j}}$, $v_{n_{i,j}}$, and $v_{s_{i,j}}$.

The terms with an over-bar in Eq. (4.66) are interpolated quantities at the east face of the CV. Interpolation can be based on the usual linear interpolation or some higher order accurate interpolation method. Examples of higher order momentum interpolation methods are Quadratic Momentum Interpolation Method proposed by Papageorgakopoulos et al. [70] and fourth-order momentum interpolation method presented by Yu et al. [71]. In case of linear interpolation the terms with over-bars in Eq. (4.66) are given by the following equations:

$$\frac{1}{\overline{a}_{i,j}} = \mathrm{f}_{e_i}\frac{1}{a_{i,j}} + \left(1-\mathrm{f}_{e_i}\right)\frac{1}{a_{i+1,j}} \tag{4.67}$$

$$\overline{H}^u_{e_{i,j}} = \mathrm{f}_{e_i} H^u_{i,j} + \left(1-\mathrm{f}_{e_i}\right) H^u_{i+1,j} \tag{4.68}$$

$$\overline{c}^u_{e_{i,j}} = \mathrm{f}_{e_i} c^u_{i,j} + \left(1-\mathrm{f}_{e_i}\right) c^u_{i+1,j} \tag{4.69}$$

$$\overline{f}^u_{e_{i,j}} = \mathrm{f}_{e_i} f^u_{i,j} + \left(1-\mathrm{f}_{e_i}\right) f^u_{i+1,j} \tag{4.70}$$

where interpolation factor f_e is given by:

$$\mathrm{f}_{e_i} = \frac{\Delta x_{i+1}/2}{\Delta x_i/2 + \Delta x_{i+1}/2} = \frac{\Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}} \tag{4.71}$$

4.3.3b. Correcting Velocity and Pressure Fields by Enforcing Mass Conservation

In SIMPLE algorithm, velocity and pressure fields are required to be corrected in every iteration. The corrections are calculated from a pressure correction equation which is derived below.

Consider an imaginary *u*-CV enclosing the geometric center of the east face of the CV corresponding to the node (*i*, *j*). Geometric centers of the west and east faces of this imaginary *u*-CV coincide with the nodes (*i*, *j*) and (*i*+1, *j*) respectively. An equation can be derived similar to Eq. (4.57) for u_e, *u*-velocity at the geometric center of the east face of the CV corresponding to the node (*i*, *j*). Such equation can be written as:

$$\begin{aligned} a_{e_{i,j}} u_{e_{i,j}} = {} & \left(a_{e_W}\right)_{i,j} u_{e_{i-1,j}} + \left(a_{e_E}\right)_{i,j} u_{e_{i+1,j}} + \left(a_{e_S}\right)_{i,j} u_{e_{i,j-1}} + \left(a_{e_N}\right)_{i,j} u_{e_{i,j+1}} \\ & + \left(p_{i,j} - p_{i+1,j}\right)\left(\Delta y\right)_j + b^u_{e_{i,j}} \end{aligned} \tag{4.72}$$

Recalling the definitions of p^*, u^*, and v^* given by Eqs. (4.25) through (4.27), if Eq. (4.72) is solved with a guessed or incorrect pressure field p^*, the resultant velocity field can be expressed as u^* and v^*. With these pressure and velocity fields, Eq. (4.72) becomes:

$$\begin{aligned} a_{e_{i,j}} u^*_{e_{i,j}} = {} & \left(a_{e_W}\right)_{i,j} u^*_{e_{i-1,j}} + \left(a_{e_E}\right)_{i,j} u^*_{e_{i+1,j}} + \left(a_{e_S}\right)_{i,j} u^*_{e_{i,j-1}} + \left(a_{e_N}\right)_{i,j} u^*_{e_{i,j+1}} \\ & + \left(p^*_{i,j} - p^*_{i+1,j}\right)\left(\Delta y\right)_j + b^u_{e_{i,j}} \end{aligned} \tag{4.73}$$

Subtracting Eq. (4.73) from Eq. (4.72):

$$a_{e_{i,j}} u'_{e_{i,j}} = \left(a_{e_W}\right)_{i,j} u'_{e_{i-1,j}} + \left(a_{e_E}\right)_{i,j} u'_{e_{i+1,j}} + \left(a_{e_S}\right)_{i,j} u'_{e_{i,j-1}} + \left(a_{e_N}\right)_{i,j} u'_{e_{i,j+1}} + \left(p'_{i,j} - p'_{i+1,j}\right)\left(\Delta y\right)_j \tag{4.74}$$

We are looking for an approximate correction that can be applied to velocity field during the iterative solution procedure. Since we are only interested in an approximation for u'_e, we drop the first four terms on the right hand side of Eq. (4.74) to get an explicit relationship between u'_e and p':

$$u'_{e_{i,j}} = \frac{\left(\Delta y\right)_j}{a_{e_{i,j}}}\left(p'_{i,j} - p'_{i+1,j}\right) \tag{4.75}$$

On the right hand side of Eq. (4.75), all quantities are known except for $a_{e_{i,j}}$. The coefficient $a_{e_{i,j}}$ is approximated by interpolation from the coefficients of the neighboring nodes of the actual grid. The interpolated value of the coefficient *a* at the east face of a CV enclosing a node (i, j) is denoted as $\overline{a}_{e_{i,j}}$. With this approximation, Eq. (4.75) becomes:

$$u'_{e_{i,j}} = \frac{\left(\Delta y\right)_j}{\overline{a}_{e_{i,j}}}\left(p'_{i,j} - p'_{i+1,j}\right) \tag{4.76}$$

Similarly, u'_w, v'_s, and v'_n are given as:

$$u'_{w_{i,j}} = \frac{(\Delta y)_j}{\overline{a}_{w_{i,j}}}\left(p'_{i-1,j} - p'_{i,j}\right) \tag{4.77}$$

$$v'_{s_{i,j}} = \frac{(\Delta x)_i}{\overline{a}_{s_{i,j}}}\left(p'_{i,j-1} - p'_{i,j}\right) \tag{4.78}$$

$$v'_{n_{i,j}} = \frac{(\Delta x)_i}{\overline{a}_{n_{i,j}}}\left(p'_{i,j} - p'_{i,j+1}\right) \tag{4.79}$$

Inserting Eq. (4.76) into Eq. (4.25), we get the following expression for u_e:

$$u_{e_{i,j}} = u^*_{e_{i,j}} + \frac{(\Delta y)_j}{\overline{a}_{e_{i,j}}}\left(p'_{i,j} - p'_{i+1,j}\right) \tag{4.80}$$

Expressions for u_w, v_s, and v_n are obtained by a similar procedure and are given below:

$$u_{w_{i,j}} = u^*_{w_{i,j}} + \frac{(\Delta y)_j}{\overline{a}_{w_{i,j}}}\left(p'_{i-1,j} - p'_{i,j}\right) \tag{4.81}$$

$$v_{s_{i,j}} = v^*_{s_{i,j}} + \frac{(\Delta x)_i}{\overline{a}_{s_{i,j}}}\left(p'_{i,j-1} - p'_{i,j}\right) \tag{4.82}$$

$$v_{n_{i,j}} = v^*_{n_{i,j}} + \frac{(\Delta x)_i}{\overline{a}_{n_{i,j}}}\left(p'_{i,j} - p'_{i,j+1}\right) \tag{4.83}$$

Now, a similar procedure is followed with Eqs. (4.57) and (4.60) to get the following expressions for nodal velocities:

$$u_{i,j} = u^*_{i,j} + \frac{(\Delta y)_j}{a_{w_{i,j}}}\left(p'_w - p'_e\right)_{i,j} \tag{4.84}$$

$$v_{i,j} = v^*_{i,j} + \frac{(\Delta x)_i}{a_{i,j}}\left(p'_s - p'_n\right)_{i,j} \tag{4.85}$$

Now, substituting Eqs. (4.80) through (4.83) into Eq. (4.50) and performing some re-arrangement, we get the following equation for pressure correction p':

$$a^{pc}_{i,j} p'_{i,j} = a^{pc}_{W_{i,j}} p'_{i-1,j} + a^{pc}_{E_{i,j}} p'_{i+1,j} + a^{pc}_{S_{i,j}} p'_{i,j-1} + a^{pc}_{N_{i,j}} p'_{i,j+1} + S^{pc}_{i,j} \tag{4.86}$$

where

$$a^{pc}_{W_{i,j}} = \rho\left(\frac{(\Delta y)_j}{\overline{a}_{w_{i,j}}}\right)(\Delta y)_j \tag{4.87}$$

$$a^{pc}_{E_{i,j}} = \rho\left(\frac{(\Delta y)_j}{\overline{a}_{e_{i,j}}}\right)(\Delta y)_j \tag{4.88}$$

$$a_{S_{i,j}}^{pc} = \rho\left(\frac{(\Delta x)_i}{\overline{a}_{s_{i,j}}}\right)(\Delta x)_i \tag{4.89}$$

$$a_{N_{i,j}}^{pc} = \rho\left(\frac{(\Delta x)_i}{\overline{a}_{n_{i,j}}}\right)(\Delta x)_i \tag{4.90}$$

$$S_{i,j}^{pc} = \rho\left(u_w^* - u_e^*\right)(\Delta y)_j + \rho\left(v_s^* - v_n^*\right)(\Delta x)_i \tag{4.91}$$

In every iteration, Eq. (4.86) is solved for p'. The calculated corrections are applied to pressure and velocity fields after every iteration.

4.3.3c. Under-relaxation

Solution of Eqs. (4.57) and (4.60) is under-relaxed to ensure convergence. Let u^l represent u-velocity at the preceding iteration; incorporating under-relaxation into Eq. (4.57):

$$u_{i,j} = \frac{\alpha^u}{a_{i,j}}\left\{a_{W_{i,j}}u_{i-1,j} + a_{E_{i,j}}u_{i+1,j} + a_{S_{i,j}}u_{i,j-1} + a_{N_{i,j}}u_{i,j+1} + \left(p_w - p_e\right)_{i,j}(\Delta y)_j + b_{i,j}^u\right\} + \left(1-\alpha^u\right)u_{i,j}^l \tag{4.92}$$

Re-arranging Eq. (4.92):

$$a_{i,j}u_{i,j} = a_{W_{i,j}}u_{i-1,j} + a_{E_{i,j}}u_{i+1,j} + a_{S_{i,j}}u_{i,j-1} + a_{N_{i,j}}u_{i,j+1} + \left(p_w - p_e\right)_{i,j}(\Delta y)_j + S_{i,j}^u \tag{4.93}$$

Corresponding equation for v-velocity is:

$$a_{i,j}v_{i,j} = a_{W_{i,j}}v_{i-1,j} + a_{E_{i,j}}v_{i+1,j} + a_{S_{i,j}}v_{i,j-1} + a_{N_{i,j}}v_{i,j+1} + \left(p_s - p_n\right)_{i,j}\left(\Delta x\right)_i + S^v_{i,j} \tag{4.94}$$

S^u and S^v in Eqs. (4.93) and (4.94) are given by:

$$S^u_{i,j} = b^u_{i,j} + \left(\frac{1}{\alpha^u} - 1\right)a_{i,j}u^l_{i,j} \tag{4.95}$$

$$S^v_{i,j} = b^v_{i,j} + \left(\frac{1}{\alpha^v} - 1\right)a_{i,j}v^l_{i,j} \tag{4.96}$$

where b^u and b^v are calculated from Eqs. (4.59) and (4.61), respectively.

Pressure correction is under-relaxed using Eq. (4.34).

4.3.3d. Convergence Criteria

The relative residuals for mass and momentum conservation equations are calculated as given below.

Relative residual for mass conservation:

$$res_{\text{mass conservation}} = \frac{\sum_{\text{CVs}}\left|\rho\left(u_w - u_e\right)_{i,j}\left(\Delta y\right)_j + \rho\left(v_s - v_n\right)_{i,j}\left(\Delta x\right)_i\right|}{\rho u_c l_c} \tag{4.97}$$

where u_c and l_c are some characteristic values of velocity and length.

Relative residual for u-velocity:

$$res_{u-\text{velocity}} = \frac{1}{\sum_{\text{CVs}} \left| a_{i,j} u_{i,j} \right|} \sum_{\text{CVs}} \left| a_{i,j} u_{i,j} - \left(a_{W_{i,j}} u_{i-1,j} + a_{E_{i,j}} u_{i+1,j} + a_{S_{i,j}} u_{i,j-1} + a_{N_{i,j}} u_{i,j+1} + \left(p_w - p_e \right)_{i,j} \left(\Delta y \right)_j + S^u_{i,j} \right) \right| \quad (4.98)$$

Relative residual for *v*-velocity:

$$res_{v-\text{velocity}} = \frac{1}{\sum_{\text{CVs}} \left| a_{i,j} v_{i,j} \right|} \sum_{\text{CVs}} \left| a_{i,j} v_{i,j} - \left(a_{W_{i,j}} v_{i-1,j} + a_{E_{i,j}} v_{i+1,j} + a_{S_{i,j}} v_{i,j-1} + a_{N_{i,j}} v_{i,j+1} + \left(p_s - p_n \right)_{i,j} \left(\Delta x \right)_i + S^v_{i,j} \right) \right| \quad (4.99)$$

4.3.3e. Algorithm

1. Assign initial values to velocity and pressure fields. These initial fields are also taken as initial guess for the subsequent iterative solution.
2. Set boundary conditions.
3. Set $n = 1$.
4. Set $u_{n,0}$ and $v_{n,0}$ equal to initial velocity field.

n^{th} Time-Step ($n = 1, n_{max}$):

1^{st} Stage ($r = 1$):

5. Set $u_{n,1} = u_{n,0}$ and $v_{n,1} = v_{n,0}$ (Eq. (4.13) and the corresponding equation for *v*-velocity).

r^{th} Stage ($r = 2, q$):

6. Calculate coefficients a_W, a_E, a_S, and a_N using the velocity fields $u_{n,r\text{-}1}$ and $v_{n,r\text{-}1}$

with a spatial discretization scheme of choice.

7. Calculate time-derivatives $f_{n,s}^{u}$ and $f_{n,s}^{v}$ (for s = 1, r-1) from Eqs. (4.53) and (4.54) respectively.
8. Calculate b^{u} and b^{v} from Eqs. (4.59) and (4.61) respectively.

Iteration for *u*, *v*, and *p*:

9. Solve Eqs. (4.57) and (4.60) with some solution algorithm such as line-by-line procedure which is a combination of Tri-diagonal Matrix Algorithm (TDMA) and Gauss-Seidel scheme.
10. Solve pressure correction equation, Eq. (4.34).
11. Apply pressure corrections to nodal pressures using Eq. (4.46).
12. Calculate pressures at CV faces by using some interpolation.
13. Calculate pressure corrections at CV faces by using some interpolation.
14. Calculate corrected face velocities using Eqs. (4.80) through (4.83).
15. Calculate coefficients of Eqs. (4.57) and (4.60) using the corrected face velocities calculated in step 14.
16. Correct u- and v- velocity fields using Eqs. (4.84) and (4.85).
17. Calculate residuals from Eqs. (4.97) through (4.99). Check for convergence by comparing the residuals with some chosen values.
18. If solution is converged, go to step 20.
19. If solution is not converged, go to step 9.
20. Check the value of r.
21. If $r = q$, go to step 23.

22. If $r < q$, switch to next stage, i.e., set $r = r + 1$. Go to step 6.

23. Check the value of n.

24. If $n < n_{max}$, switch to the next time-step, i.e., set $n = n + 1$. Set $u_{n,0}$ and $v_{n,0}$ equal to the velocity fields calculated in step 16. Go to step 5.

25. If $n = n_{max}$, stop the program.

The above solution algorithm is shown as a flow chart in Figure 4.7.

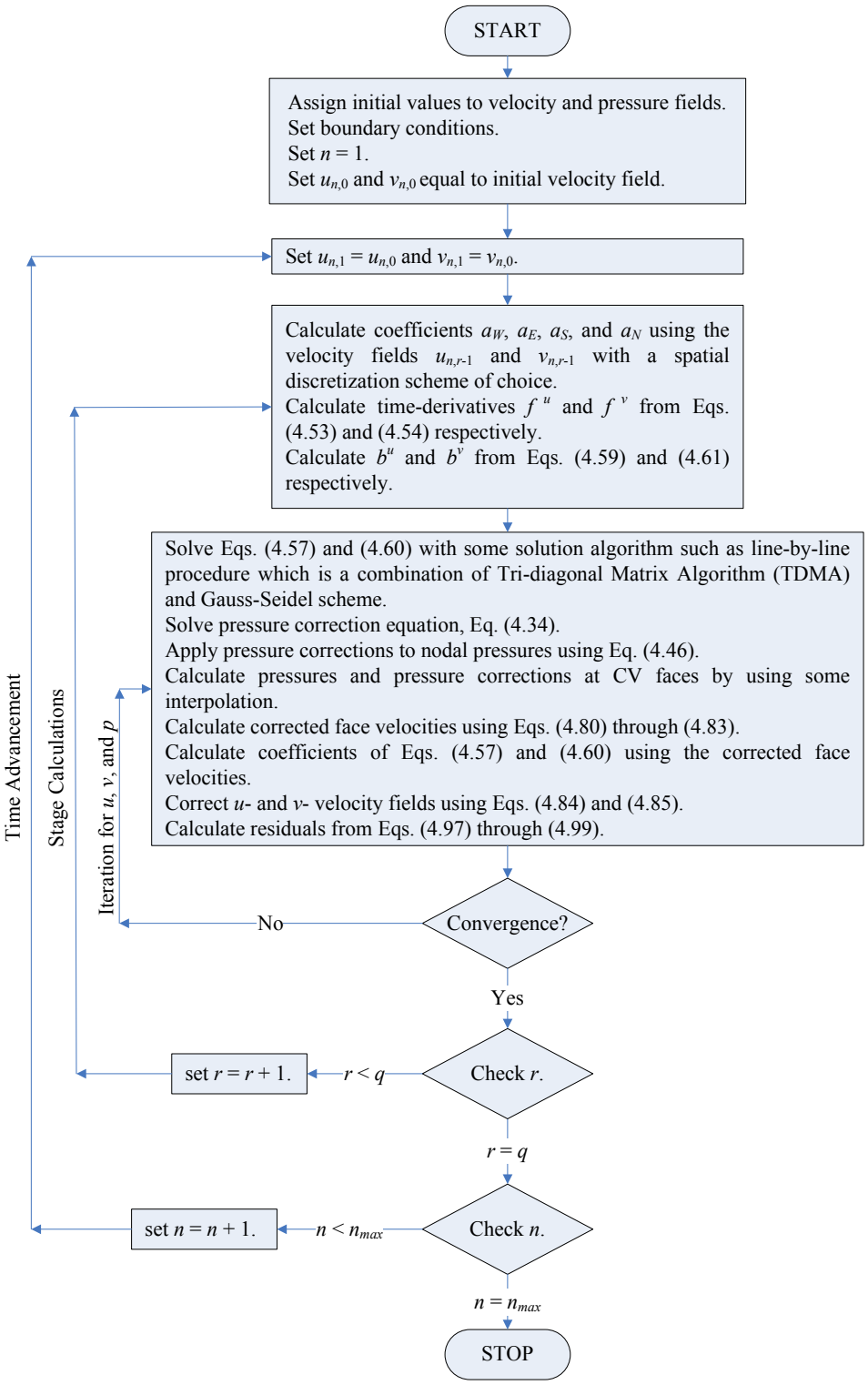


Figure 4.7 Solution Algorithm for Staggered Grid Method

5. VALIDATION

In order to validate the proposed SIMPLE DIRK method, a FORTRAN code was developed for each of the staggered and the non-staggered grid approaches. Power-law scheme of Patankar [1] was used in spatial discretization. Temporal discretization was performed with a two-stage second-order stiffly-accurate ESDIRK method. The Butcher array for the used ESDIRK method is given below:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

In the staggered grid method, CV face velocities were calculated by linear interpolation. In case of non-staggered grid method, CV face velocities were calculated by linear momentum interpolation as given by Eqs. (4.66) through (4.71). Under-relaxation factors for the momentum and pressure correction equations were based on the following relationships as proposed by Ferziger and Peric [24]:

$$\alpha_p = 1 - \alpha_{u,} \qquad \alpha_p = 1 - \alpha_v \tag{5.1}$$

A value of 0.3 was used for α_p and 0.7 for both α_u and α_v. The simultaneous solution of momentum and mass conservation equations was considered to be converged when the values of residuals of momentum conservation equations became less than 10^{-6}

and that of continuity equation reached below 10^{-5}.

5.1. Test Case

Flow field in a lid-driven 1 m × 1 m square cavity was solved by the proposed method for air with constant thermo-physical properties. The values used for absolute viscosity and density were 1.843×10^{5} $N.s/m^2$ and 1.177 kg/m^3 respectively. Calculations were performed for Reynolds number of 400 and 1,000; where Reynolds number was based on cavity height and the lid velocity.

5.2. Grid Dependence Study

A grid dependence study was performed before a grid size was chosen for the code validation runs. The time step size for grid dependence study was 10 seconds.

The procedure for this study is as follows. The staggered grid code was run for a flow time of 30 seconds with a grid size of 11×11 and Root-Mean-Square (RMS) value of u-velocity along the vertical centerline of the cavity was calculated. The code was run again for the same flow time but with the refined grid size of 25×25 and RMS value of u-velocity along the vertical centerline of the cavity was calculated. Then the absolute value of marginal relative percent change in the RMS value of u-velocity was calculated from the following formula:

$$\text{Marginal Relative \% Change} = \frac{\left| u_{\text{RMS}_{\text{fine grid}}} - u_{\text{RMS}_{\text{coarse grid}}} \right|}{u_{\text{RMS}_{\text{coarse grid}}} \times \text{Increase in Number of Grid Points}} \times 100 \quad (5.2)$$

The above procedure was repeated with grid sizes of 40×40, 51×51, 60×60, 68×68, 75×75, and 81×81. The results are plotted in Figure 5.1. This figure also shows the results

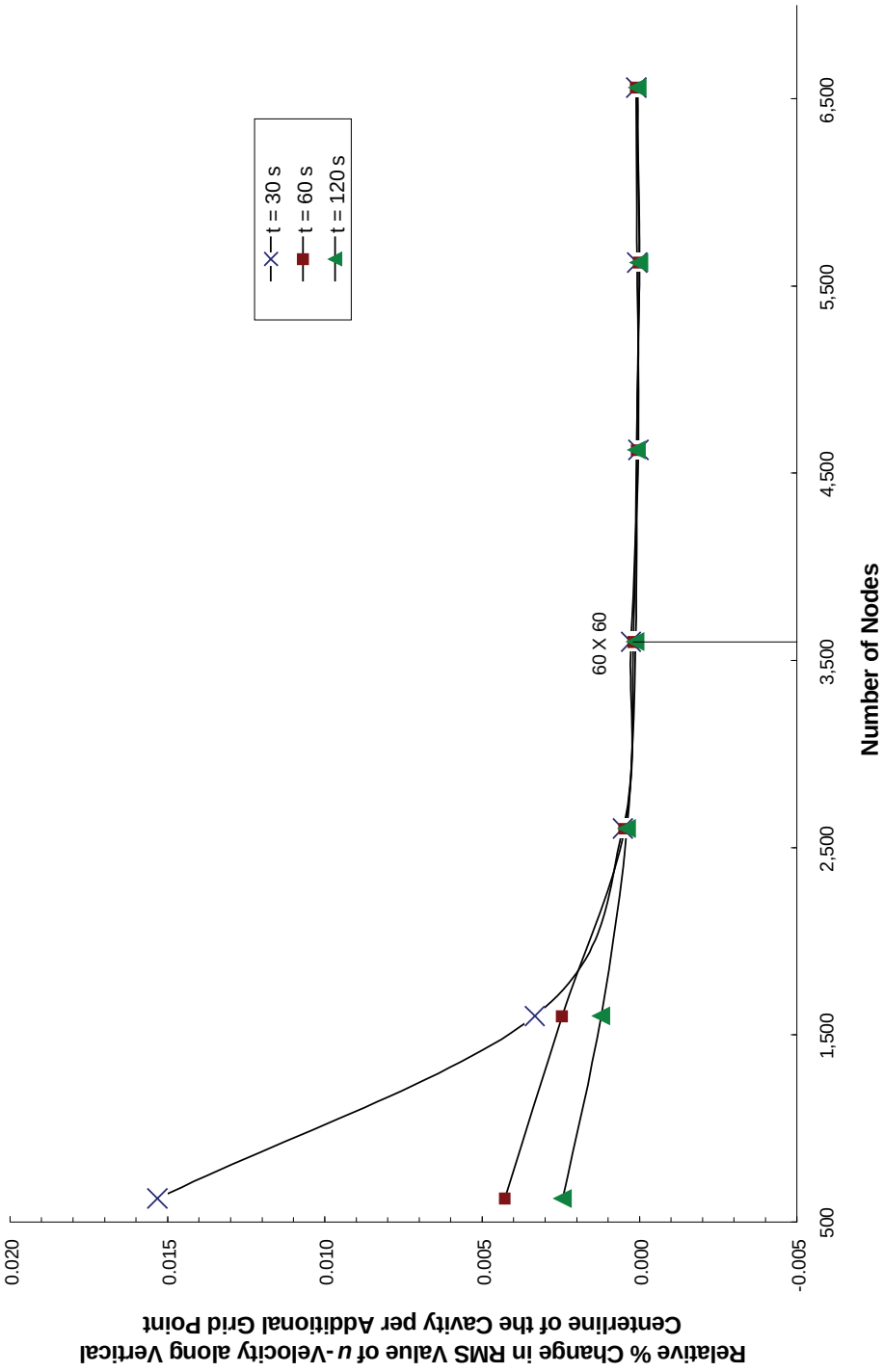


Figure 5.1 Grid Dependence Study

obtained with the flow time of 60 seconds and 120 seconds. Although the marginal relative percent change in RMS value of *u*-velocity along the vertical centerline of the cavity decreases when grid is refined beyond 60×60 internal nodes, yet the change is so small that it was decided to perform calculations with the 60×60 grid.

5.3. Code Validation Runs

The staggered grid code was validated by comparison with the results of commercial software program FLUENT [4], Ghia et al. [5], and Erturk et al. [6]. Then the non-staggered grid code was validated by comparison with the results of the staggered grid code.

5.3.1. Comparison of the Staggered Grid Code with FLUENT for Re = 400

For the comparison of the staggered grid code with the results of commercial software program FLUENT [4], simulation was performed for *Re* = 400 and a flow time of 3,600 seconds. The time step size for this simulation was 1 second. The same grid density, time step size, and flow time were used in the simulation with the staggered grid code and FLUENT. In the solution by FLUENT, second order implicit time advancing was chosen. A higher-order scheme (QUICK) was used for spatial discretization in FLUENT compared with power law scheme used in the test solution by the staggered grid code. The results are discussed in Section 6.

5.3.2. Comparison of the Staggered Grid Code with the Results of Erturk et al. [6] for Re = 1,000

The staggered grid code was also run for *Re* = 1,000 and a flow time of 3,000

seconds. The results were compared with the steady state solutions of Ghia et al. [5] and Erturk et al. [6]. The time step size for this simulation was 10 seconds. The same grid density was used in the simulation with the staggered grid code as used by Ghia et al. [5] and Erturk et al. [6]. The results are discussed in the next section.

5.3.3. Comparison of the Non-staggered Grid Code with the Staggered Grid Code for Re = 400

In order to validate the non-staggered grid method the above problem was again solved with both the staggered grid and the non-staggered grid codes for a flow time of 4,000 seconds, with a time step size of 10 seconds, and grid density of 60×60 internal nodes. A discussion on the results is presented in Section 6.

6. RESULTS AND DISCUSSION

First, the results from the staggered grid code were compared with those from FLUENT [4] and the numerical solution of Ghia et al. [5] and Erturk et al. [6]. *Second*, the results from the non-staggered grid code were compared with those from the staggered grid code.

Normalized u-velocity profile along a vertical line through the center of the cavity as obtained from the staggered grid code was compared with that calculated from FLUENT [4]. The results at various instants in time are presented in Figure 6.1. In Figures 6.2 and 6.3, u- and v-velocity contours obtained from the staggered grid code at t = 200 s are compared with those obtained from FLUENT [4]. Figures 6.4 and 6.5 present similar comparison at t = 400 s. It is evident from Figures 6.1 through 6.5 that the solution obtained from the code is in good agreement with the solution of FLUENT [4]. In Figure 6.6, the u-velocity profile obtained from the staggered grid code at 3,600 seconds is compared with the steady state numerical solution of Ghia et al. [5]. The staggered grid code was also run for Re = 1,000 and a flow time of 3,000 seconds. Normalized u- and v-velocity profiles along a vertical line through the center of the cavity are compared with those of Ghia et al. [5] and Erturk et al. [6] in Figures 6.7 and 6.8. The results from the code agree well with the results of Ghia et al. [5].] and Erturk et al. [6].

In order to investigate how the difference in the solution of the staggered grid code and the results of FLUENT varies as the simulation proceeds in time, normalized u-velocity data was extracted for 50 equally spaced points on the vertical centerline of the cavity from the solutions of the staggered grid code and FLUENT [4] at various instants in

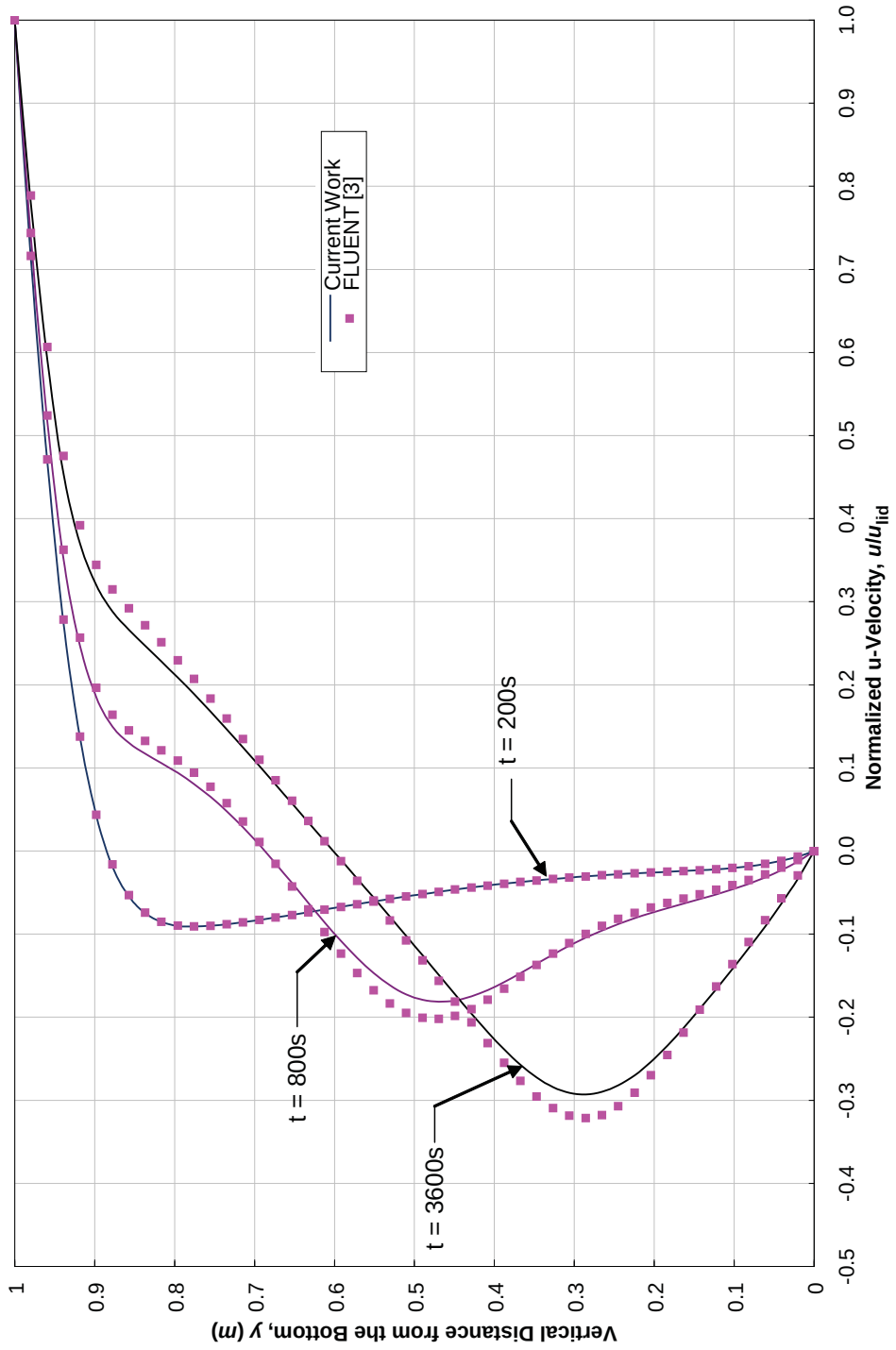
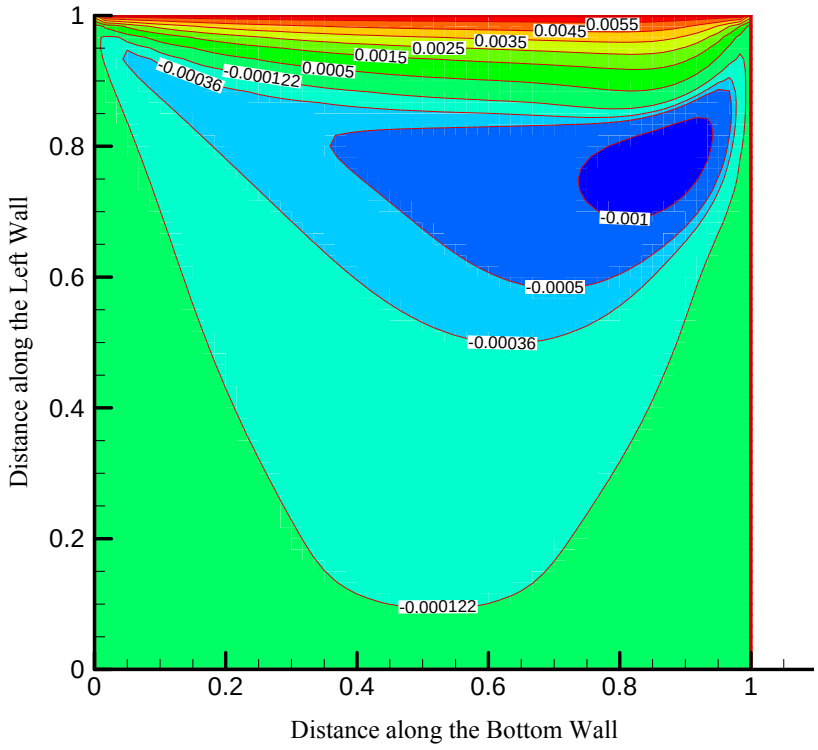


Figure 6.1 Time Evolution of Normalized u-Velocity Profile along Vertical Centerline ($x = 0.5\ m$) of a Square Cavity for $Re = 400$

a) FLUENT



b) Staggered Grid Code

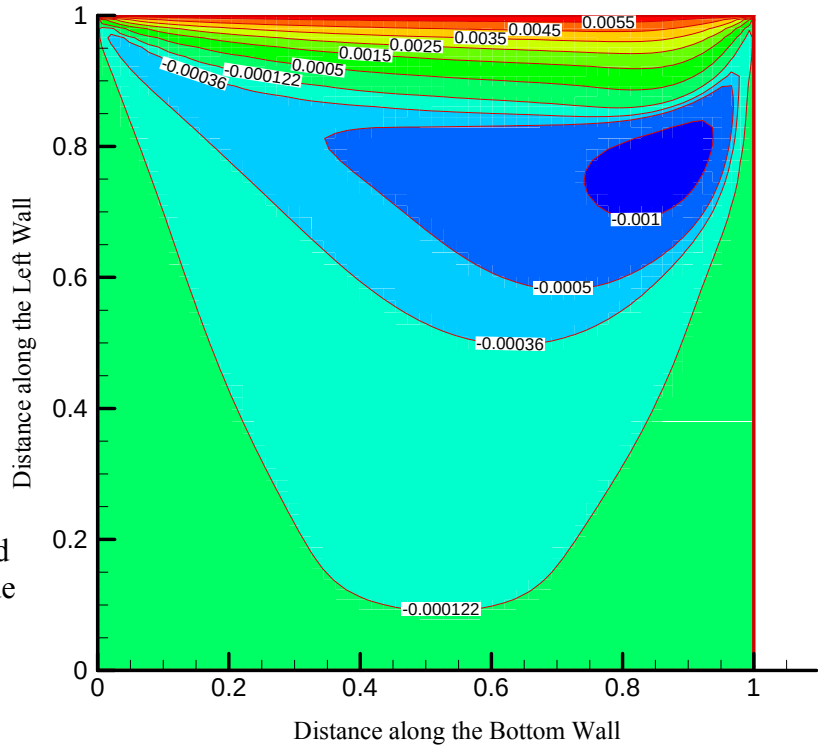
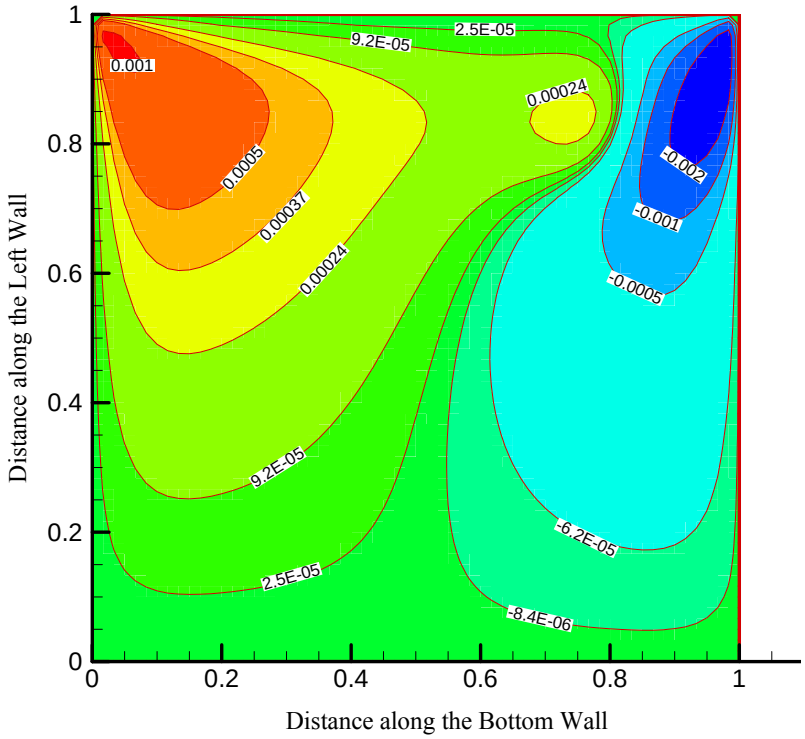


Figure 6.2 *u*-Velocity Contours at $t = 200\ s$ for $Re = 400$

a) FLUENT



b) Staggered Grid Code

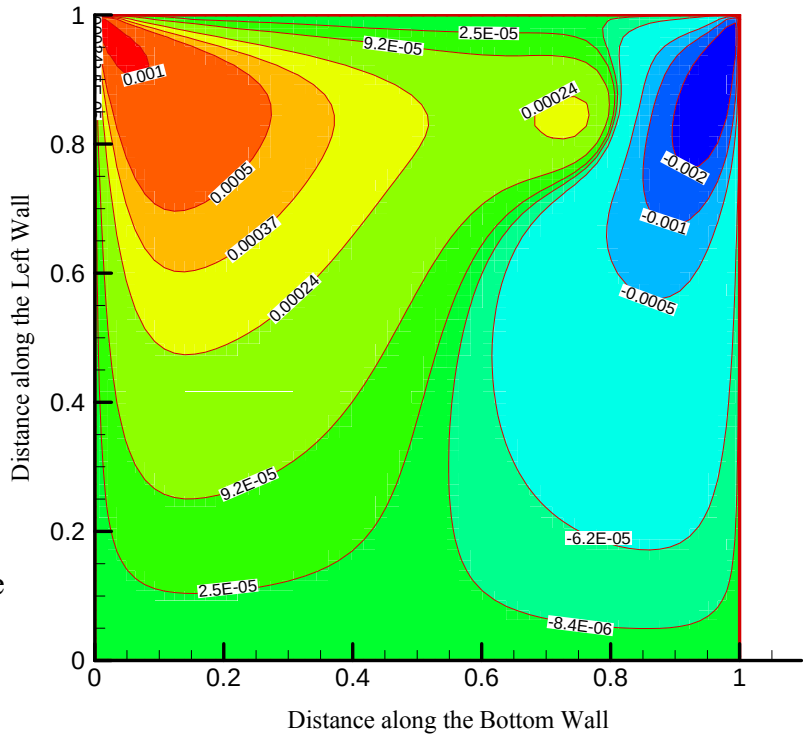
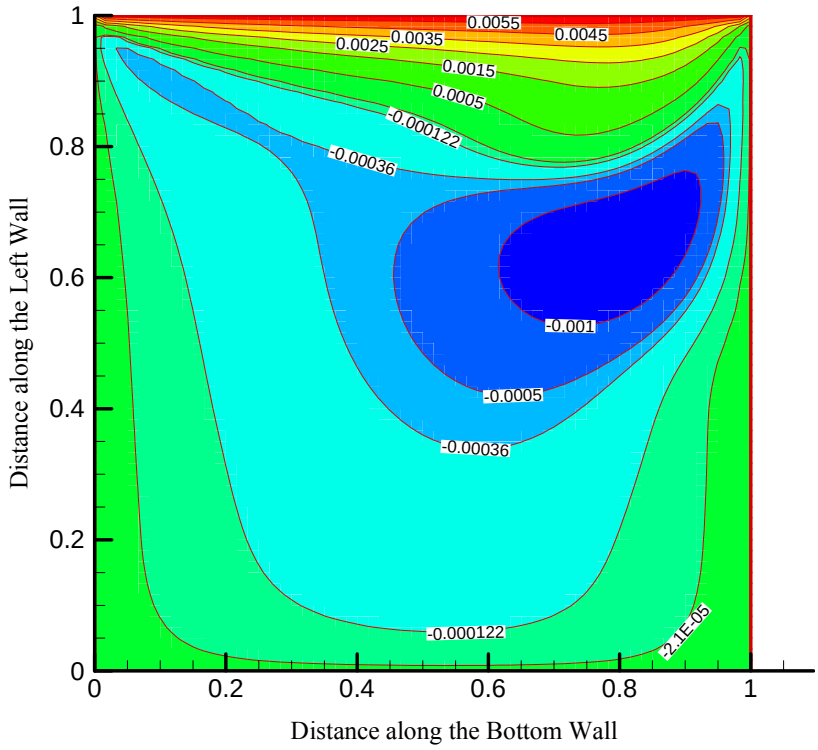


Figure 6.3 *v*-Velocity Contours at $t = 200\ s$ for $Re = 400$

a) FLUENT



b) Staggered Grid Code

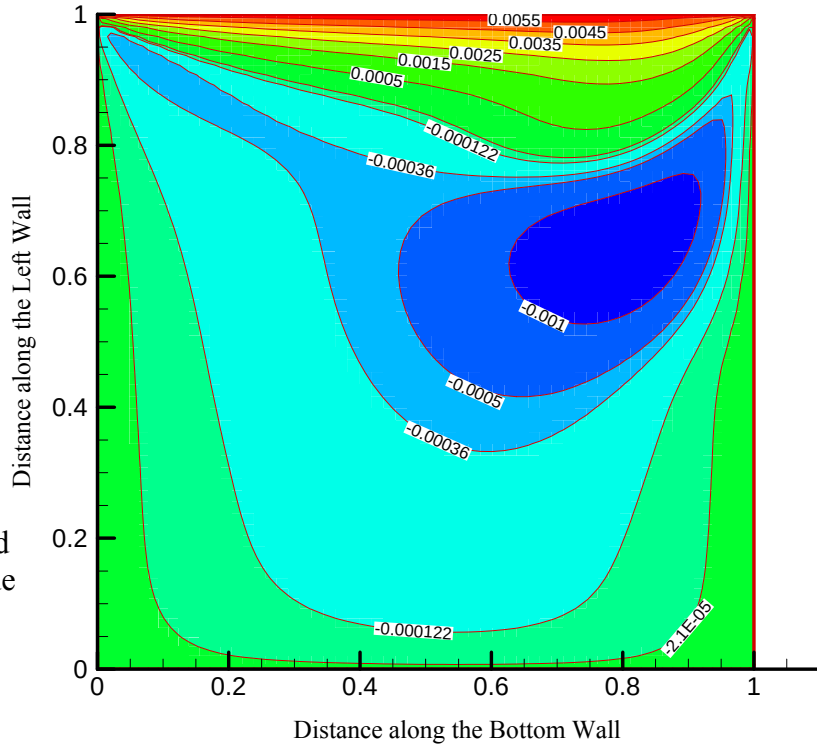
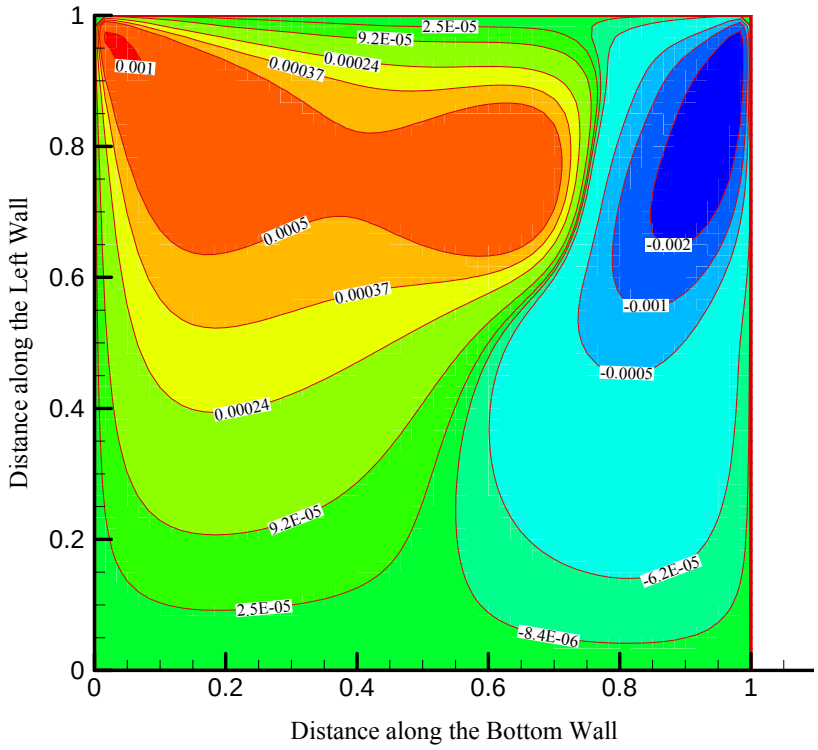


Figure 6.4 *u*-Velocity Contours at $t = 400\ s$ for $Re = 400$

a) FLUENT



b) Staggered Grid Code

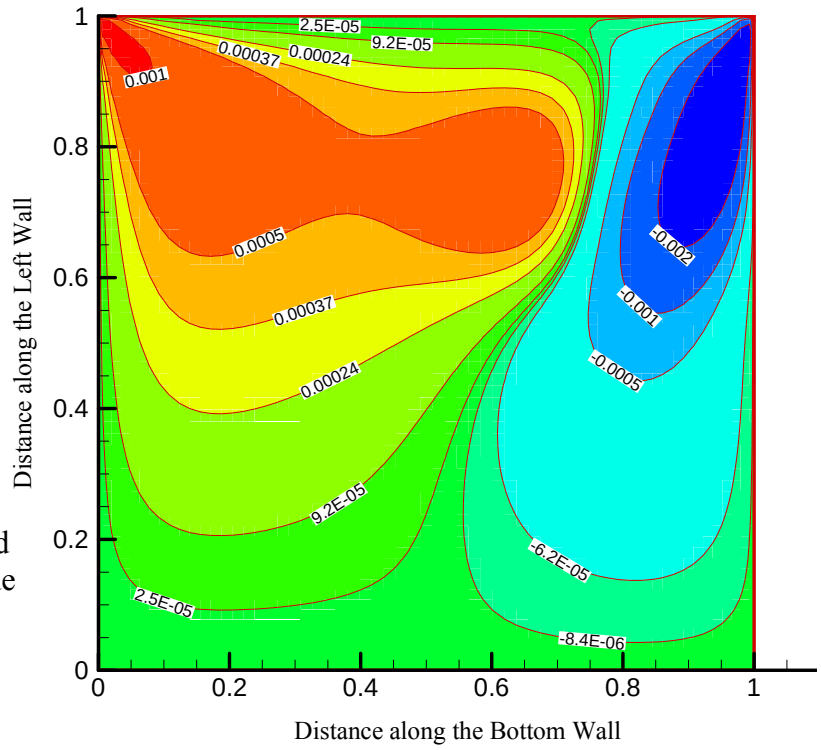


Figure 6.5 *v*-Velocity Contours at $t = 400\ s$ for $Re = 400$

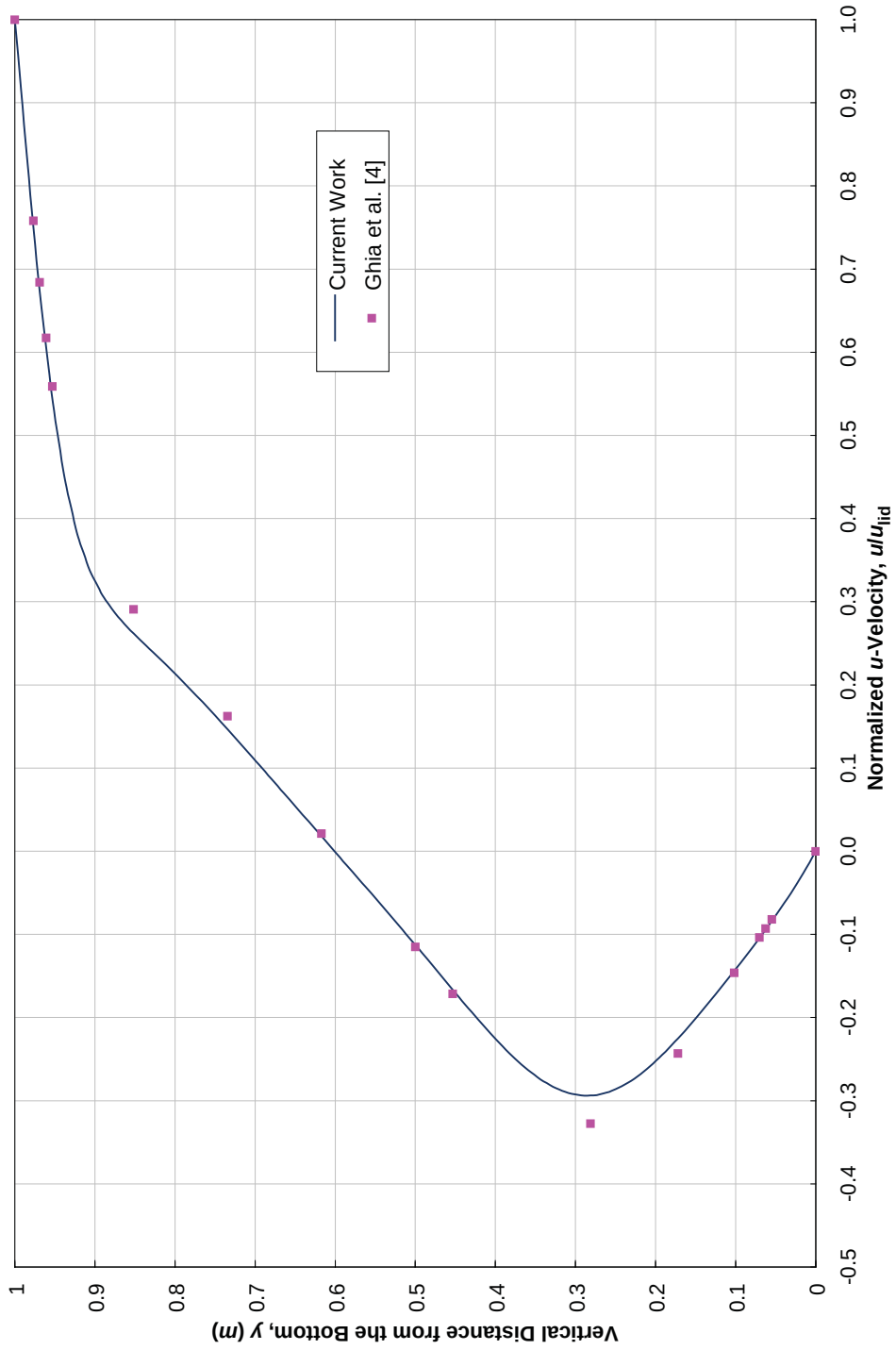


Figure 6.6 Normalized u-Velocity Profile on Vertical Centerline ($x = 0.5\ m$) of a Square Cavity for $Re = 400$ at $t = 3{,}600\ s$ (Steady State)

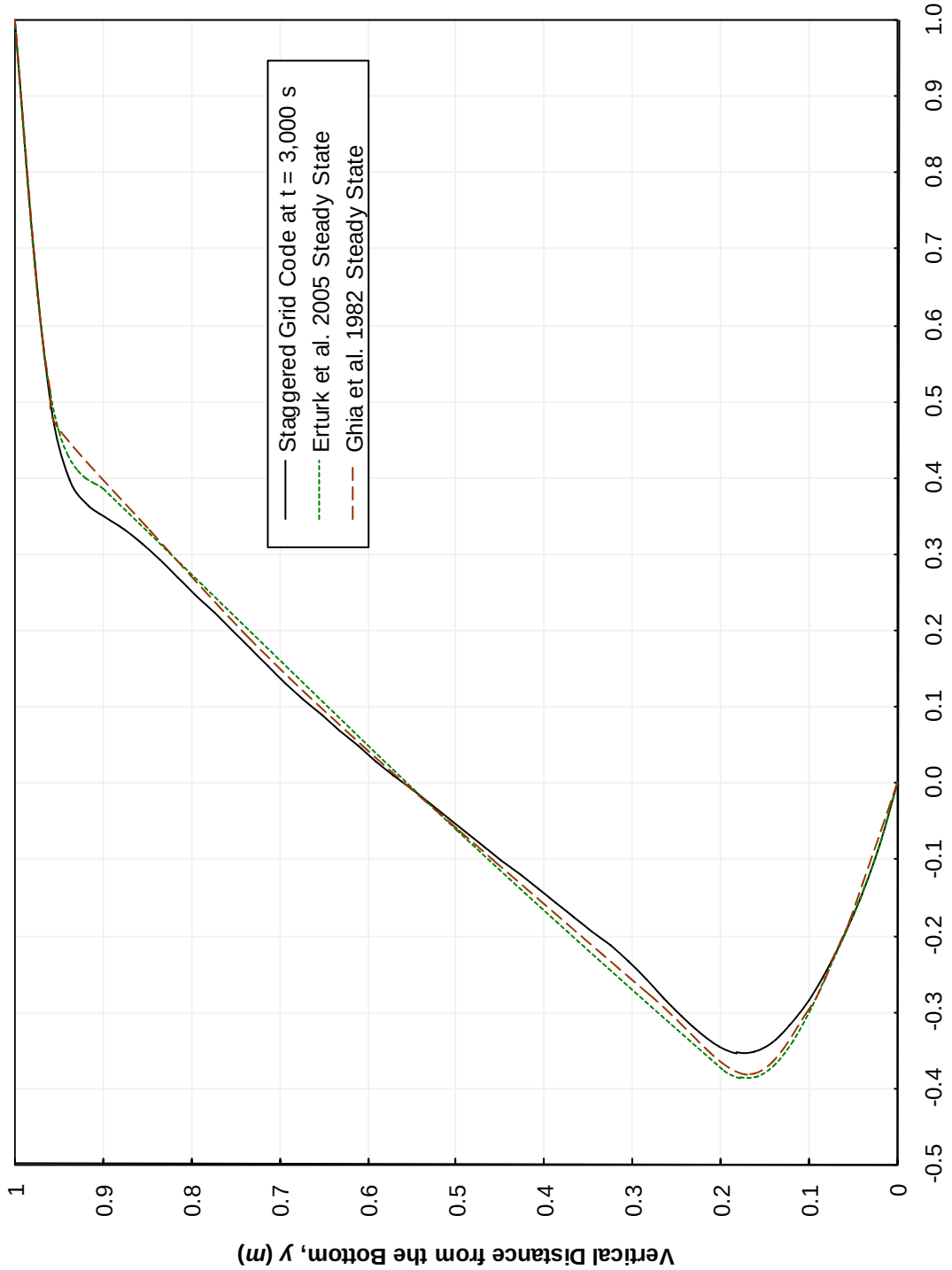


Figure 6.7 Normalized u-Velocity Profile on Vertical Centerline ($x = 0.5\ m$) of a Square Cavity for $Re = 1{,}000$

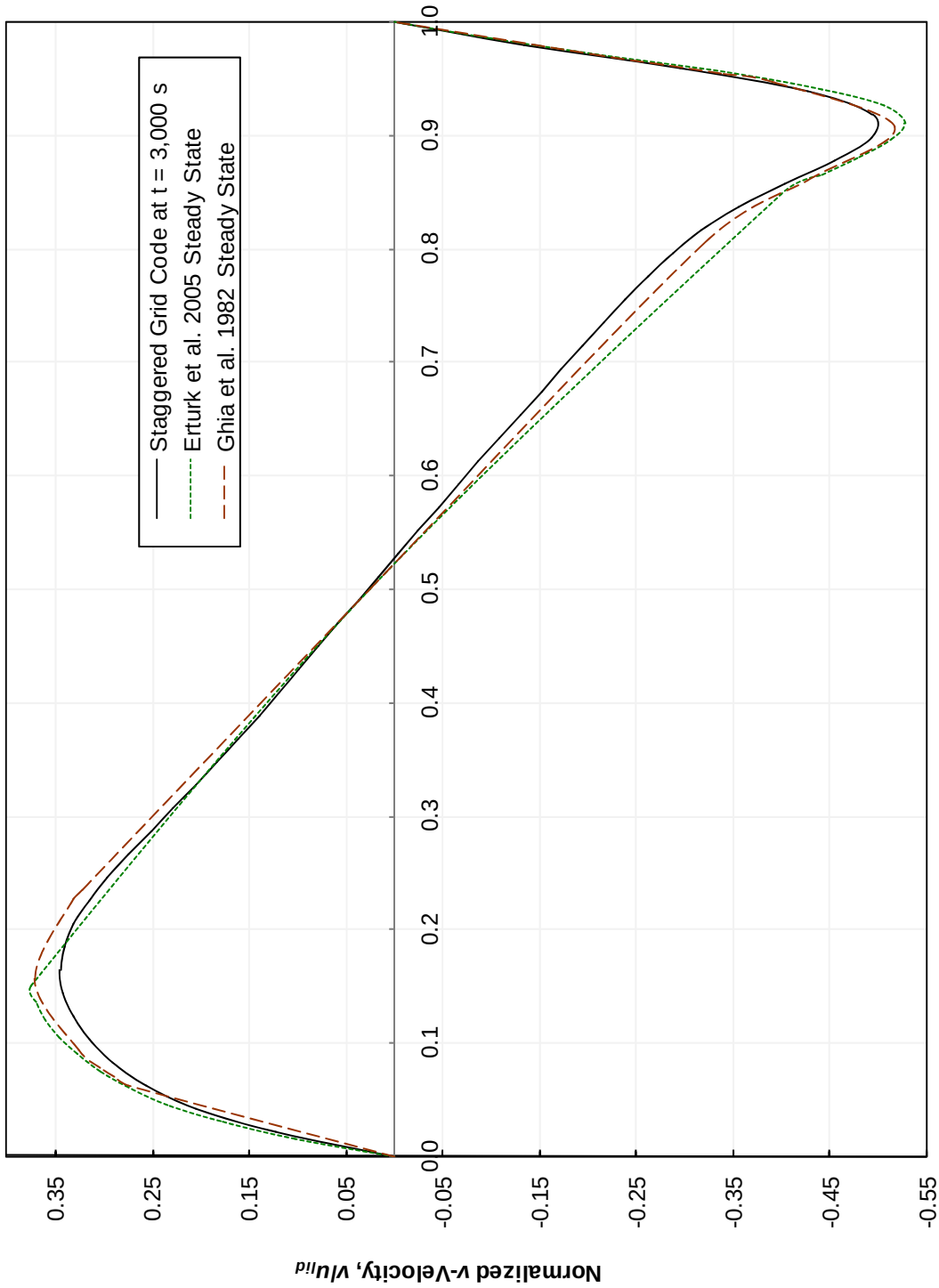


Figure 6.8 Normalized v-Velocity Profile on Horizontal Centerline ($y = 0.5\ m$) of a Square Cavity for $Re = 1{,}000$

time. Absolute difference in the two solutions was calculated for these 50 points and maximum value of absolute difference determined. RMS value of the differences was also calculated. Figure 6.9 shows variation of maximum absolute difference and RMS value of the difference of the two solutions with time. At the beginning, when time-derivatives of velocity are large in this flow, the difference in solutions grows quickly. At later times, growth of difference in solutions almost levels out. This can be explained as follows. Accuracy of time-derivative of velocity is dependent on the accuracy of spatial discretization scheme as is evident from Eqs. (4.11), (4.12), (4.53) and (4.54). Use of lower-order spatial discretization introduces errors in calculation of time-derivative of velocity. Eqs. (4.14) and (4.15) show that velocity field at the end of every time-step depends on the values of time-derivatives obtained from *stage calculations*. The errors in time-derivative of velocity, therefore, affect the accuracy of velocity field. When velocity field changes rapidly with time, i.e. time-derivatives are large, the effect of errors in time-derivatives on the calculated velocity field is more significant. During time spans when time-derivatives are small, the effect of errors in time-derivatives introduced due to use of low-order spatial discretization on the calculated velocity field are less significant. This is the reason for growth of difference in solutions in Figure 6.9 to diminish at later times in the cavity flow.

Another conclusion that follows the above discussion is that the use of higher-order spatial discretization is necessary if higher-order accuracy in time is desired. Moreover, as shown by Eq. (4.14), a smaller time-step size h will lessen the effect of errors introduced by use of low-order spatial discretization. However, a time-step size dependence study is

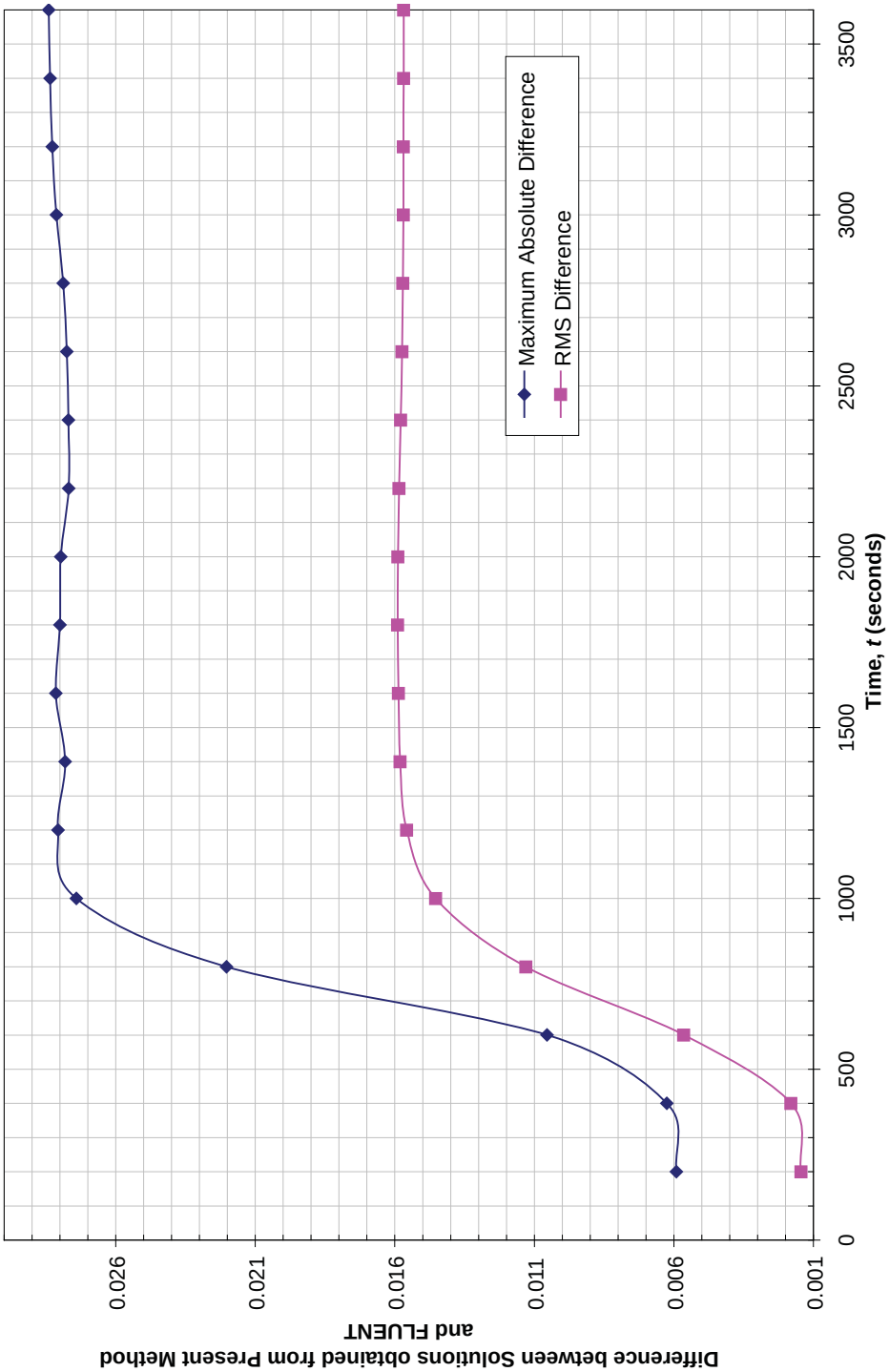


Figure 6.9 Difference in Solutions from the staggered Grid Code and from FLUENT for Normalized u-Velocity on Vertical Centerline ($x = 0.5\ m$) of a Square Cavity for $Re = 400$

required in order to find out optimum time-step size.

Since growth of difference in solutions almost stops after 1,200 seconds of flow time (Figure 6.9), the solution will not become unstable. This observation gives a fair indication that the proposed method is stable. However, more rigorous analysis and testing are required to establish the stability of this method.

After comparison of the results of the staggered grid code, a comparison was made between the results of the staggered and non-staggered grid codes. Normalized *u*- and *v*-velocity profiles are plotted along the cavity vertical and horizontal centerlines, respectively, in Figures 6.10 and 6.11 as obtained from both the staggered grid and non-staggered grid codes at various instants in time. The solutions obtained from the two methods are in excellent agreement with each other. To take a quantitative look at the comparison of the two results, the absolute differences were calculated as percents of the lid velocity using the following equation:

$$\text{Absolute\% Difference} = \frac{\left| u_{\text{staggered grid}} - u_{\text{non-staggered grid}} \right|}{u_{lid}} \times 100 \tag{6.1}$$

The maximum % absolute difference is shown in Figure 6.12 at four instants in time. Similar percent differences were also calculated for *v*-velocity at the horizontal centerline of the cavity. The maximum values of these percent differences are also shown in Figure 6.12. At t = 1,000 s, the maximum % difference in the values of *v*-velocity at the horizontal centerline of the cavity is only 0.68. Figure 6.13 presents a comparison of the two methods in terms of number of iterations required for convergence of solution at

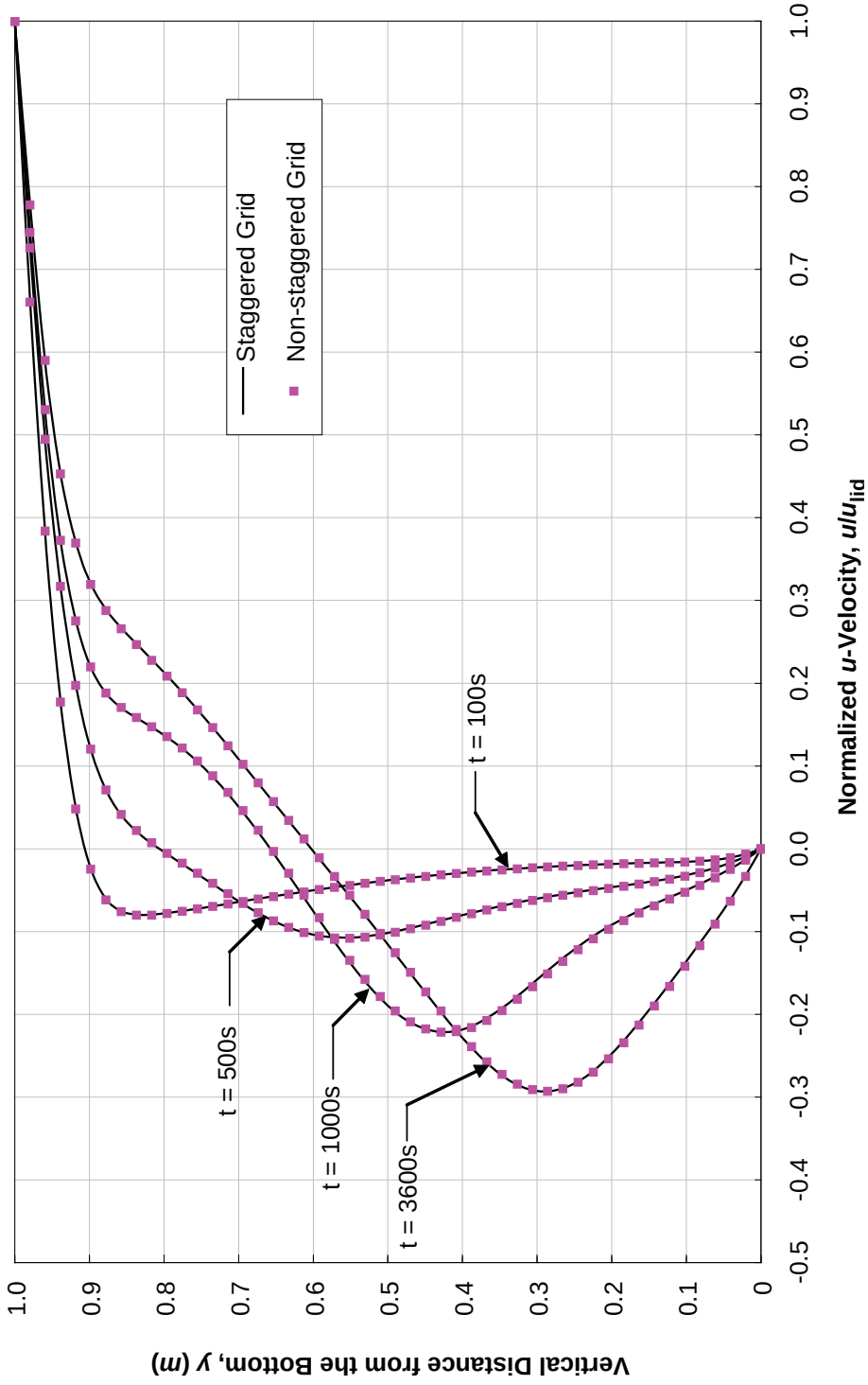


Figure 6.10 Time Evolution of Normalized u-Velocity Profile along Vertical Centerline ($x = 0.5\ m$) of a Square Cavity for $Re = 400$

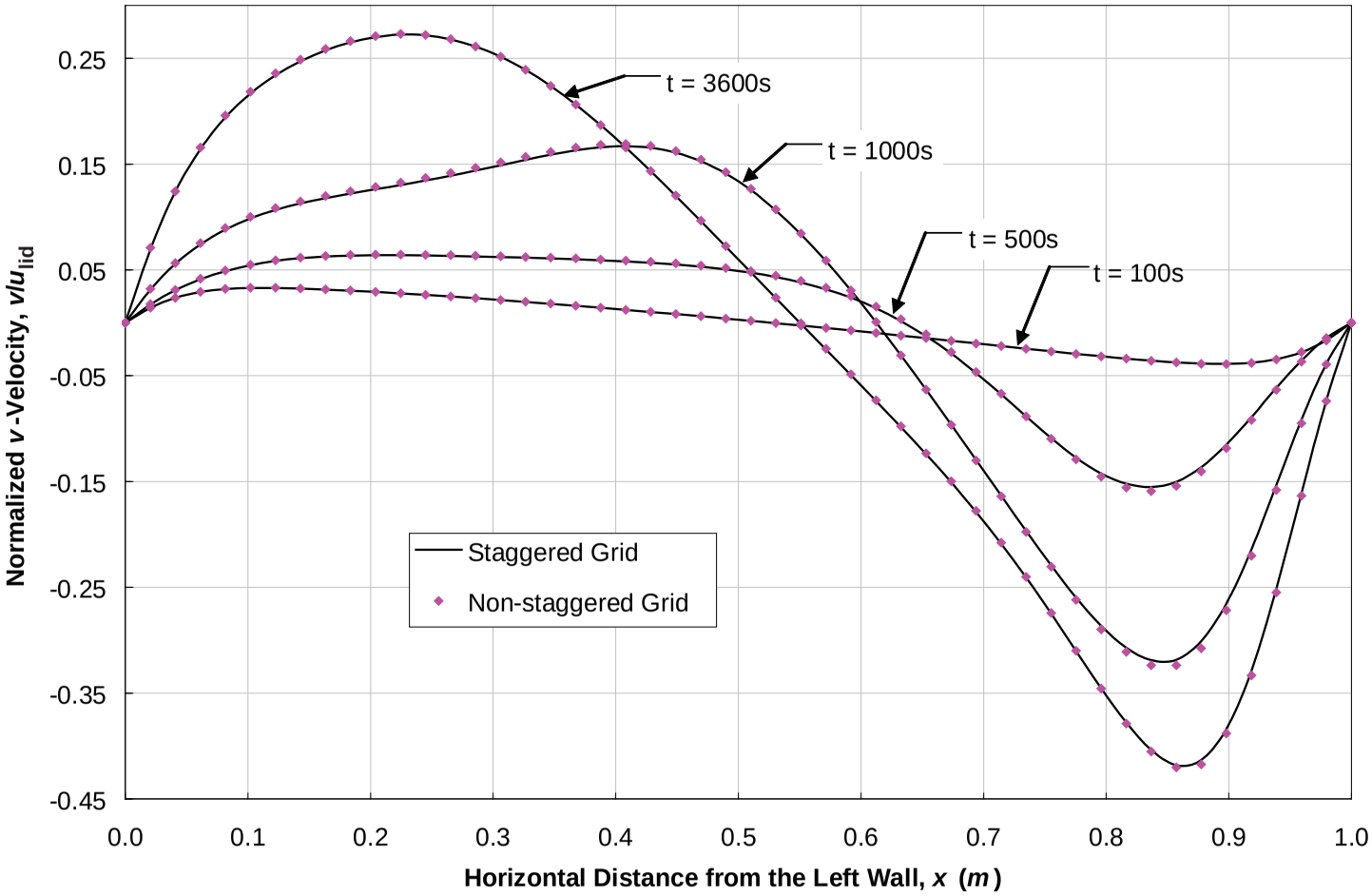


Figure 6.11 Time Evolution of Normalized v-Velocity Profile along Horizontal Centerline ($y = 0.5\ m$) of a Square Cavity for $Re = 400$

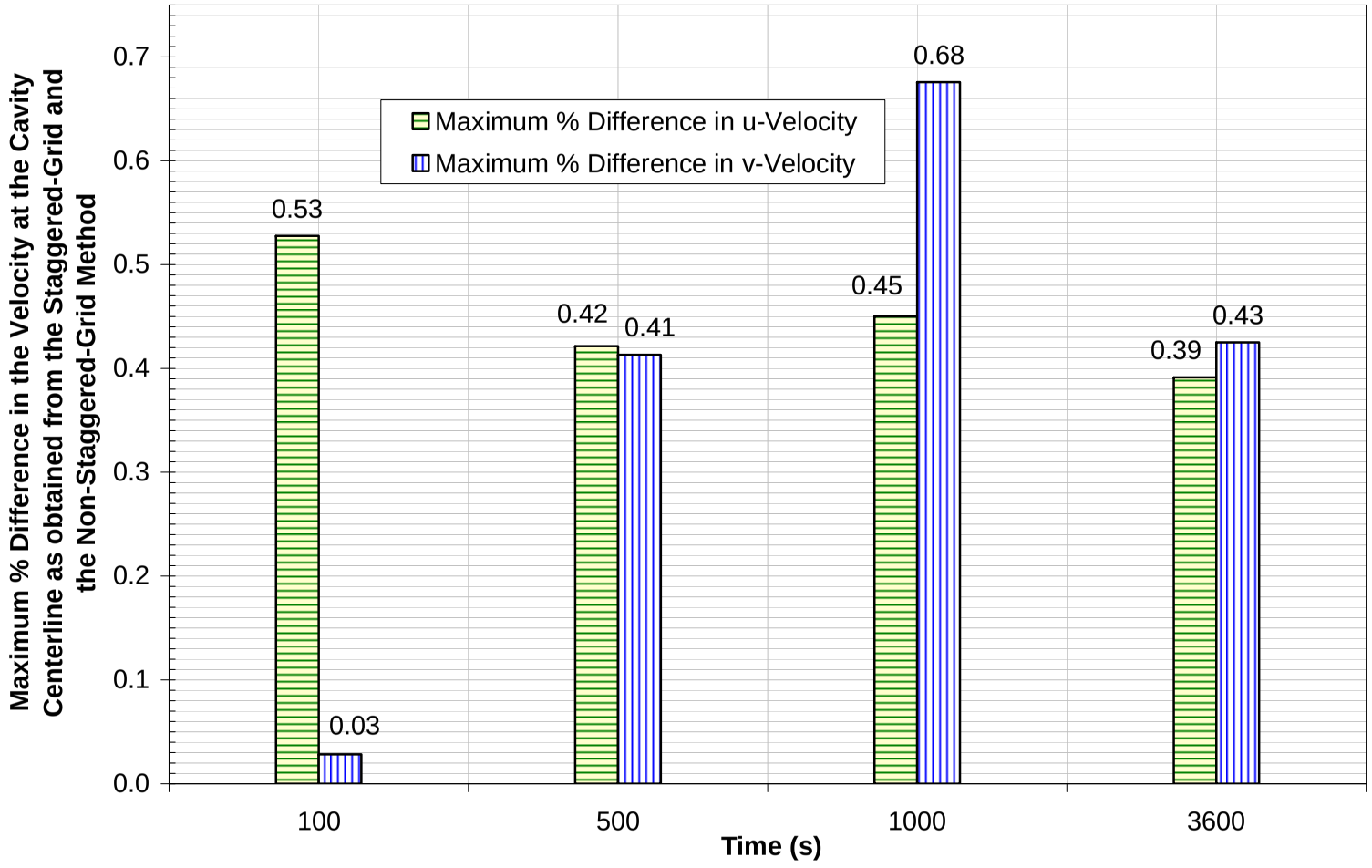


Figure 6.12 Maximum Difference in Values of *u*- and *v*-Velocity as obtained from the Staggered-Grid and the Non-Staggered-Grid Methods at Selected Time Instants as Percent of the Lid Velocity

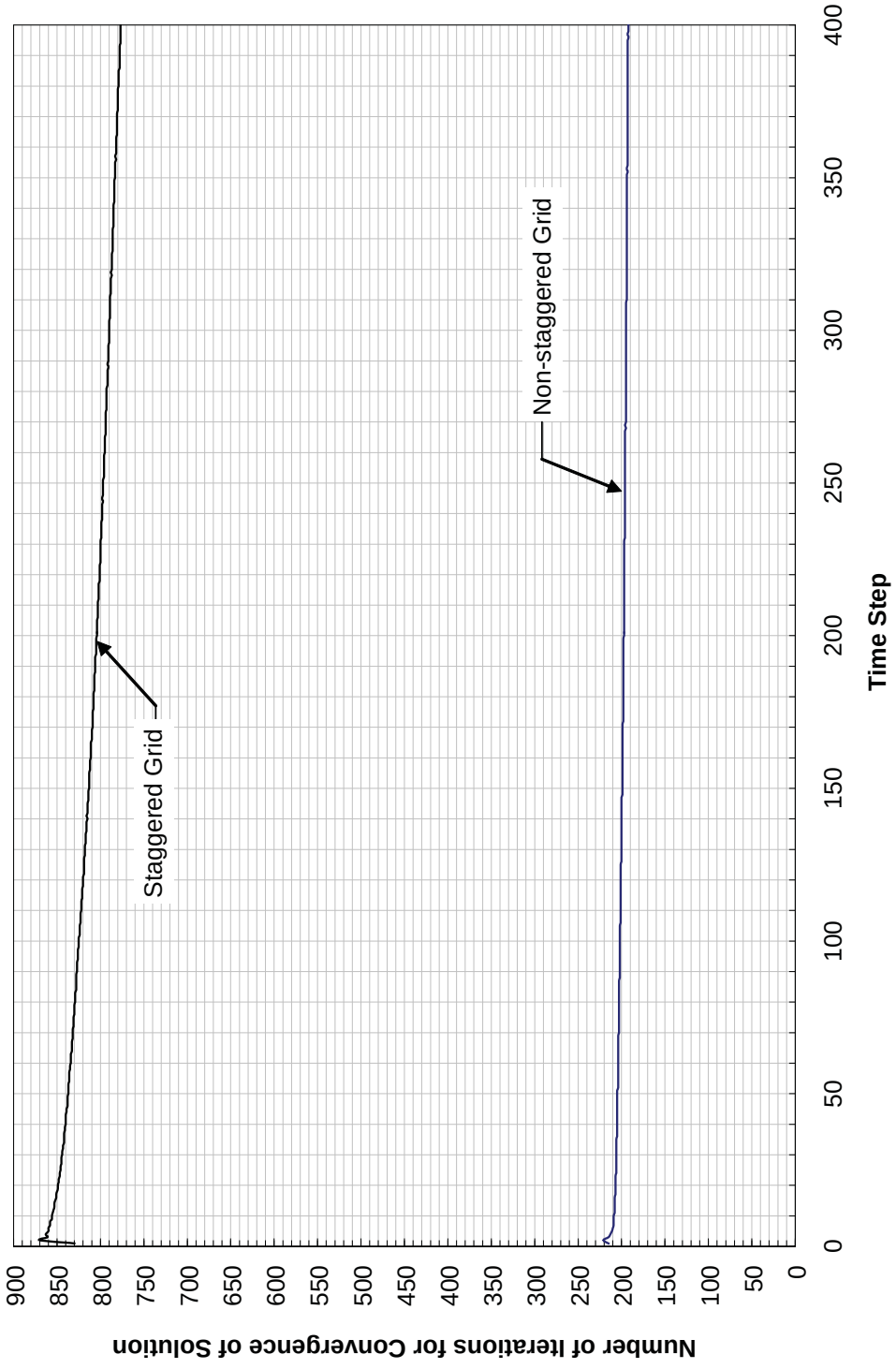


Figure 6.13 Number of Iterations for Convergence of Solution at Each Time Step

each time step. Similar comparison is shown in terms of CPU time in Figure 6.14. The CPU time data shown in Figure 6.14 were obtained when the code was run on a personal computer with Intel Core 2 Duo processor. It is clear from Figures 6.13 and 6.14 that the non-staggered grid SIMPLE DIRK method converges faster than the staggered-grid method. Figure 6.13 shows that the number of iterations required in a typical time step to meet the convergence criteria chosen in the current solution is about 800 with the staggered-grid method as opposed to about 200 with the non-staggered-grid method. Figure 6.14 shows that the CPU time required in a typical time step to meet the convergence criteria chosen in the current solution is about 23 seconds with the staggered-grid method as opposed to about 11 seconds with the non-staggered-grid method. One obvious reason for the superior behavior of the non-staggered-grid method is that the coefficients of both *x*- and *y*-momentum equations are identical and, therefore, required to be calculated only once during every iteration. Figures 6.15 and 6.16 show, for staggered grid and non-staggered grid method respectively, the values of residuals of *u*-velocity, *v*-velocity, and continuity at every iteration for a typical time step. Logarithmic values of all the residuals decrease linearly after certain number of iterations. In another study (results are not shown here) when time step size was increased to large values, this linear variation was replaced by oscillations, but the solution still proceeded toward convergence. However, in comparison to the staggered grid method, the non-staggered grid method offers faster convergence.

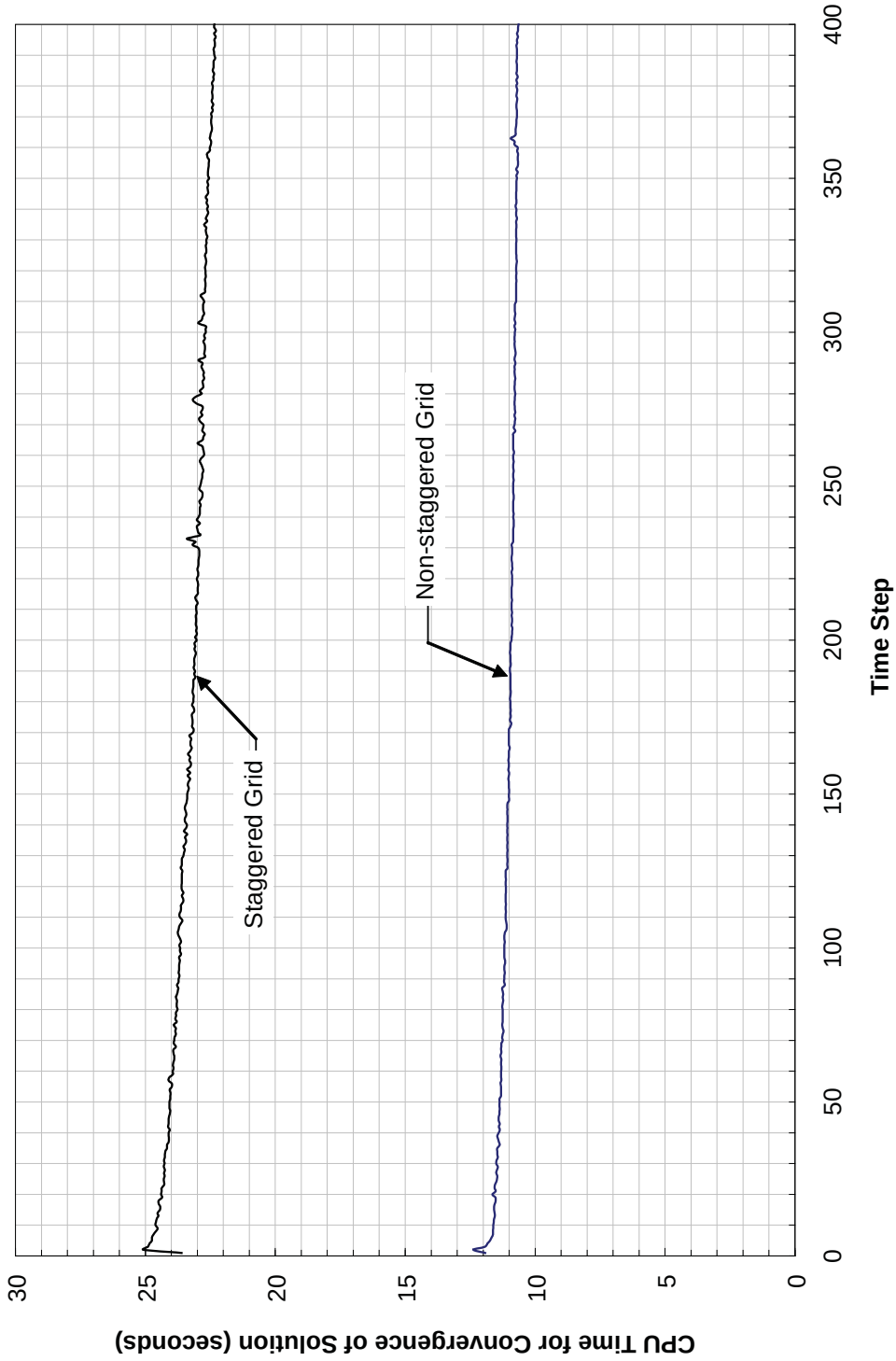


Figure 6.14 CPU Time for Convergence of Solution at Each Time Step

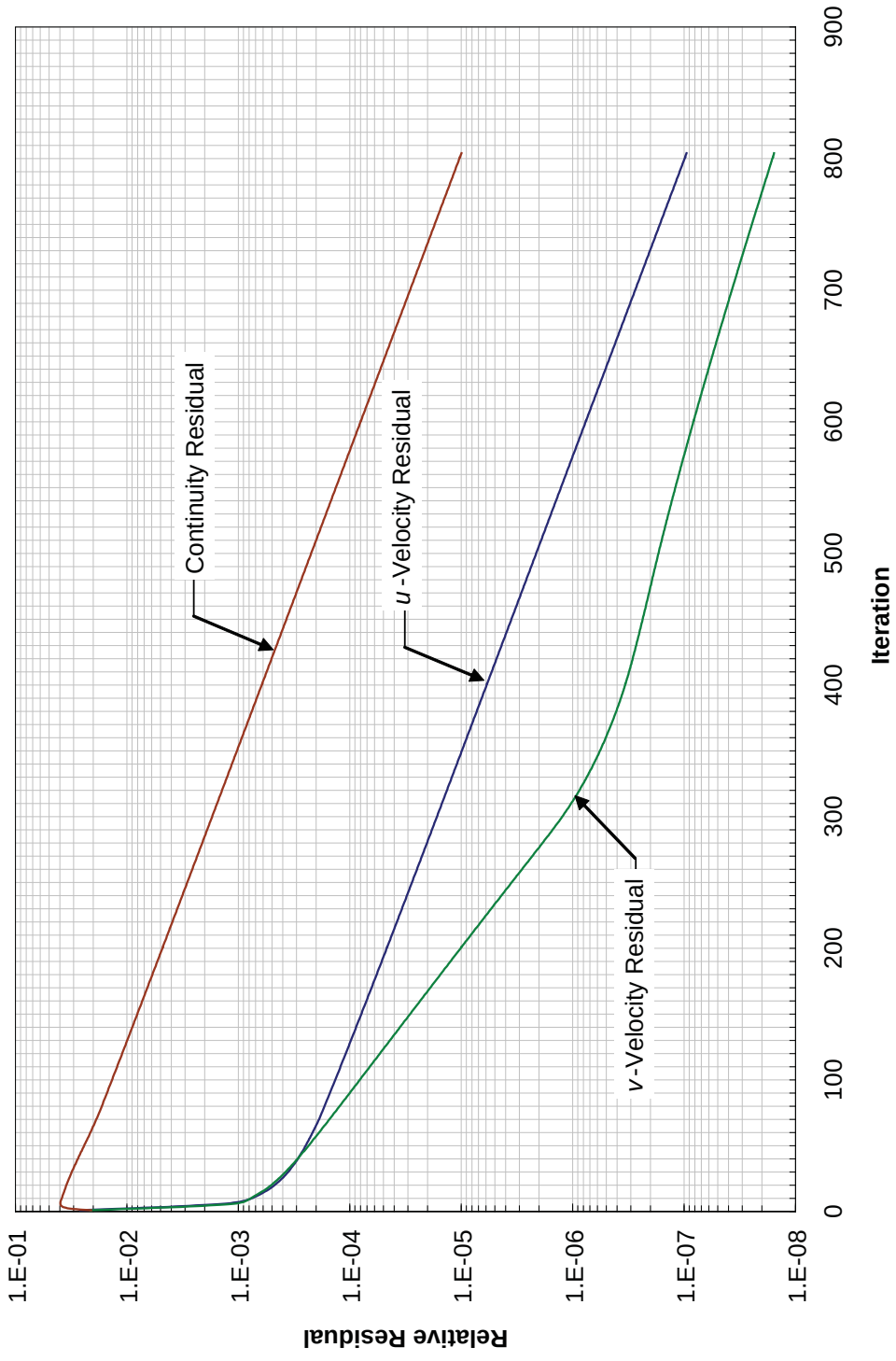


Figure 6.15 Residual for u- and v-Velocity, and Continuity at Each Iteration at the 200th Time Step in the Solution of Square Cavity Flow with Staggered Grid SIMPLE DIRK Method

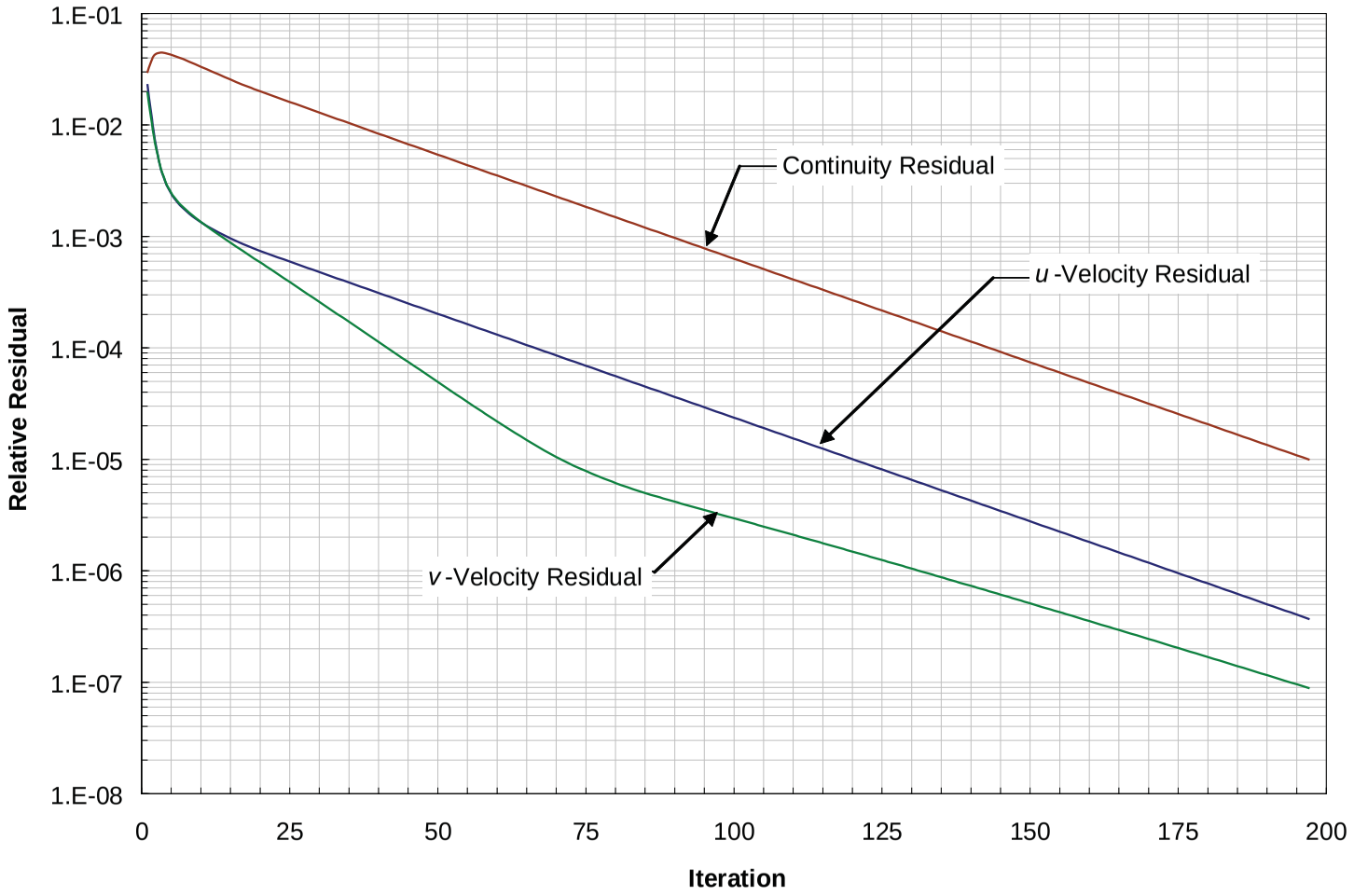


Figure 6.16 Residual for u- and v-Velocity, and Continuity at Each Iteration at the 200th Time Step in the Solution of Square Cavity Flow with Non-staggered Grid SIMPLE DIRK Method

7. SUMMARY

A numerical method (SIMPLE DIRK Method) is presented for unsteady incompressible flow simulation. This method uses implicit Runge-Kutta methods in conjunction with finite volume method. The method is presented for both staggered and non-staggered approaches. A FORTRAN code was developed for each of these two approaches. The staggered grid code was validated by comparison of its results with those obtained from FLUENT [4] and published by Ghia et al. [5] and Erturk et al. [6]. Non-staggered grid code was validated by comparison with the staggered grid code.

Good agreement of the results of the two codes with the solution of FLUENT [4] and the results of Ghia et al. [5] and Erturk et al. [6] establishes that the proposed method is feasible and has prospects for extension to higher-order RK methods with higher-order spatial discretization. For higher-order accuracy in time, use of higher-order spatial discretization is necessary. Moreover, a smaller time-step size h will produce higher accuracy. The method was observed to be stable. The non-staggered-grid (co-located variables) SIMPLE DIRK method produced results that are nearly equivalent to the ones obtained from the staggered-grid SIMPLE DIRK method. However, the non-staggered grid method exhibited better convergence behavior with less CPU time requirement for the same level of convergence.

8. RECOMMENDATIONS FOR FUTURE WORK

Studies should be initiated to investigate the effect of convergence criteria, under-relaxation factors, and time step size on the results of the presented method. For DNS applications, the proposed method should be used with higher order ESDIRK methods for time discretization in conjunction with higher order spatial discretization schemes. Using higher order discretization, DNS data can be generated for code validations and investigation of physical laws. Appendix C gives formulation for a method with four stage Runge-Kutta method. Application of the presented method to heat transfer problems and multi-phase flows involving chemical reaction and/or radiation should be explored. Extendibility of the presented non-staggered grid method to complex domains with internal regions should be worked out.

REFERENCES

1. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.

2. M. Ijaz and N. K. Anand, Simulation of Unsteady Incompressible Viscous Flow using Higher Order Implicit Runge-Kutta Methods – Staggered Grid, HT 2007-32486, *ASME-JSME Thermal Engineering Conference and Summer Heat Transfer Conference*, Vancouver, BC, Canada, July 8-12, 2007.

3. M. Ijaz and N. K. Anand, Simulation of Unsteady Incompressible Viscous Flow using Higher Order Implicit Runge-Kutta Methods – Staggered Grid, *Numerical Heat Transfer*, Part B, vol. 52, pp. 495–512, 2007.

4. FLUENT, Software Package, Ver. 6.2.16, Fluent Inc., 10 Cavendish Court, Lebanon, NH 03766.

5. U. Ghia, K. N. Ghia, and C. T. Shin, High-Re Solutions for Incompressible Flow using the Navier-Stokes Equations and a Multigrid Method, *Journal of Computational Physics*, vol. 48, pp. 387–411, 1982.

6. E. Erturk, T. C. Corke, and C. Goökçöl, Numerical Solutions of 2-D Steady Incompressible Driven Cavity Flow at High Reynolds Numbers, *International Journal for Numerical Methods in Fluids*, vol. 48, pp. 747–774, 2005.

7. M. Ijaz and N. K. Anand, Co-located Variables Approach in SIMPLE DIRK Method for Simulation of Unsteady Incompressible Viscous Flows, *Numerical Heat Transfer* (to be submitted).

8. M. Ijaz and N. K. Anand, Co-located Variables Approach in SIMPLE DIRK Method for Simulation of Unsteady Incompressible Viscous Flows, US-34, *8th ISHMT-ASME Heat and Mass Transfer Conference*, Hyderabad, India, January 3-5, 2008.

9. F. H. Harlow, Hydrodynamic Problems Involving Large Fluid Distortion, *Journal of Association of Computing Machinery*, vol. 4, pp. 137–142, 1957.

10. R. A. Gentry, R. E. Martin and B. J. Daly, An Eulerian Differencing Method for Unsteady Compressible Flow Problems, *Journal of Computational Physics*, vol. 1, pp. 87–118, 1966.

11. J. E. Fromm and F. H. Harlow, Numerical Solution of the Problem of Vortex Street Development, *Physics of Fluids*, vol. 6, pp. 975–982, 1963.

12. F. H. Harlow and J. E. Welch, Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluids with Free Surface, *Physics of Fluids*, vol. 8, pp. 2182–2189, 1965.

13. A. J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, *Journal of Computational Physics*, vol. 2, pp. 12–26, 1967.

14. A. J. Chorin, Numerical Solution of the Navier-Stokes Equations, *Mathematics of Computation*, vol. 22, No. 104, pp. 745–762, 1968.

15. A. J. Chorin, On the Convergence of Discrete Approximation to the Navier–Stokes Equations, *Mathematics of Computation*, vol. 23, No. 106, pp. 341–353, 1969.

16. F. M. Denaro, On the Application of the Helmholtz–Hodge Decomposition in Projection Methods for Incompressible Flows with General Boundary Conditions, *International Journal for Numerical Methods in Fluids*, vol. 43, pp. 43–69, 2003.

17. S. A. Orszag, M. Israeli, and M. O. Deville, Boundary Conditions for Incompressible Flows, *Journal of Scientific Computing*, vol. 1, No. 1, pp. 75–111, 1986.

18. A. Dagan, Numerical Consistency and Spurious Boundary Layer in the Projection Method, *Computers & Fluids*, vol. 32, pp. 1213–1232, 2003.

19. D. W. Peaceman and H. H. Rachford, Jr., The Numerical Solution of Parabolic and Elliptic Differential Equations, *SIAM Journal on Applied Mathematics*, vol. 3, No. 1, pp. 28–41, 1955.

20. M. Lees, Alternating Direction Methods for Hyperbolic Differential Equations, *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, No. 4, pp. 610–616, 1962.

21. J. L. Steger and P. Kutler, Implicit Finite-Difference Procedures for the Computation of Vortex Wakes, *AIAA Journal*, vol. 15, No. 4, pp. 581–590, 1977.

22. J. Crank and P. Nicholson, A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-conduction Type, *Proceedings of the Cambridge Philosophical Society*, vol. 43, pp. 50-67, 1947.

23. R. W. MacCormack, The Effect of Viscosity in Hypervelocity Impact Cratering, *AIAA Hypervelocity Impact Conference*, Cincinnati, Ohio, *AIAA Paper*, No. 69-354, 1969.

24. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, Germany, 1980.

25. E. Fehlberg, Classical Fifth, Sixth, Seventh, and Eighth Order Runge-Kutta Formulas with Stepsize Control, NASA Tech. Rep. TR R-287, Marshall Space Flight Center, Atlanta, GA, 1968.

26. E. Fehlberg, Low-Order Classical Runge-Kutta Formulas with Stepsize Control and their Application to some Heat Transfer Problems, NASA Tech. Rep. TR R-315, Marshall Space Flight Center, Atlanta, GA, 1969.

27. A. Jameson, W. Schmidt, and E. Turkel, Numerical Solution of the Euler Equations by Finite volume Methods Using Runge-Kutta Time-Stepping Schemes, *AIAA 14th Fluid and Plasma Dynamics Conference*, June 23-25, 1981.

28. T. Cebeci, J. P. Shao, F. Kafyeke, and E. Laurendeau, *Computational Fluid Dynamics for Engineers*, Springer, Berlin, Germany, 2005.

29. J. Kim and P. Moin, Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations, *Journal of Computational Physics*, vol. 59, pp. 308–323, 1985.

30. J. V. Kan, A Second-Order Accurate Pressure-Correction Scheme for Viscous Incompressible Flow, *SIAM Journal on Scientific Computing*, vol. 7, pp. 870–891, 1986.

31. J. B. Bell, P. Colella, and H. M. Glaz, A Second Order Projection Method for the Incompressible Navier-Stokes Equations, *Journal of Computational Physics*, vol. 85, pp. 257–283, 1989.

32. J. B. Perot, An Analysis of the Fractional Step Method, *Journal of Computational Physics*, vol. 108, pp. 51–58, 1993.

33. J. C. Strikwerda and Y. S. Lee, The Accuracy of the Fractional Step Method, *SIAM Journal on Numerical Analysis*, vol. 37, No. 1, pp. 37–47, 1999.

34. D. L. Brown, R. Cortez, M. L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, *Journal of Computational Physics*, vol. 168, pp. 464–499, 2001.

35. M. Liu, Y. Ren, and H. Zhang, A Class of Fully Second Order Accurate Projection Methods for Solving the Incompressible Navier-Stokes Equations, *Journal of Computational Physics*, vol. 200, pp. 325–346, 2004.

36. M. M. Rai and P. Moin, Direct simulations of turbulent Flow using Finite-Difference Schemes, *Journal of Computational Physics*, vol. 96, pp. 15–53, 1991.

37. P. R. Spalart, R. D. Moser, and M. Rogers, Spectral Methods for the Navier-Stokes Equations with one Infinite and two Periodic Directions, *Journal of Computational Physics*, vol. 96, pp. 297–324, 1991.

38. R. Verzicco and P. Orlandi, A Finite-Difference Scheme for the Three-Dimensional Incompressible Flows in Cylindrical Coordinates, *Journal of Computational Physics*, vol. 123, pp. 402–414, 1996.

39. N. Nikitin, Third-Order-Accurate Semi-Implicit Runge-Kutta Scheme for Incompressible Navier-Stokes Equations, *International Journal for Numerical Methods in Fluids*, vol. 51, pp. 221–233, 2006.

40. L. S. Carreto, A. D. Gosman, S. V. Patankar, and D. B. Spalding, Two Calculation Procedures for Steady, Three-Dimensional Flows with Recirculation, *Third International Conference on Numerical Methods in Fluid Dynamics*, Paris, 1972.

41. S. V. Patankar, A Calculation Procedure for Two-dimensional Elliptic Situations, *Numerical Heat Transfer*, vol. 4, pp. 409–425, 1981.

42. J. P. Van Doormal and G. D. Raithby, Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numerical Heat Transfer*, vol. 7, pp. 147–163, 1984.

43. D. S. Jang, R. Jetli, and S. Acharya, Comparison of the PISO, SIMPLER, and SIMPLEC Algorithms for the Treatment of the Pressure-Velocity Coupling in Steady Flow Problems, *Numerical Heat Transfer*, vol. 19, pp. 209–228, 1986.

44. A. U. Chatwani and A. Turan, Improved Pressure-Velocity Coupling Algorithm based on Global Residual Norm, *Numerical Heat Transfer*, Part B, vol. 20, pp. 115–123, 1991.

45. R.-H.Yen and C.-H. Liu, Enhancement of the SIMPLE Algorithm by an Additional Explicit Corrector Step, *Numerical Heat Transfer*, Part B, vol. 24, pp. 127–141, 1993.

46. E. Blosch and W. Shyy, The Role of Mass Conservation in Pressure-Based Algorithms, *Numerical Heat Transfer*, Part B, vol. 24, pp. 415–429, 1993.

47. Y. Sheng, M. Shoukri, G. Sheng, and P. Wood, A Modification to the SIMPLE Method for Buoyancy-Driven Flows, *Numerical Heat Transfer*, Part B, vol. 33, pp. 65–78, 1998.

48. F. Moukalled and M. Darwish, A Unified Formulation of the Segregated Class of Algorithm for Fluid Flow at all Speeds, *Numerical Heat Transfer*, Part B, vol. 37, pp. 103–139, 2000.

49. M. M. Rahman and T. Siikonen, An Improved SIMPLE Method on a Collocated Grid, *Numerical Heat Transfer*, Part B, vol. 38, pp. 177–201, 2000.

50. B. Yu, H. Ozoe, and W. Q. Tao, A Modified Pressure-Correction Scheme for the SIMPLER Method, MSIMPLER, *Numerical Heat Transfer*, Part B, vol. 39, pp. 439–449, 2001.

51. W. Z. Shen, J. A. Michelsen, N. N. Sørensen, and J. Nørkær Sørensen, An Improved SIMPLEC Method on Collocated Grids for Steady and Unsteady Flow Computations, *Numerical Heat Transfer*, Part B, vol. 43, pp. 221–239, 2003.

52. Z. G. Qu, W. Q. Tao, and Y. L. He, An Improved Numerical Scheme for the SIMPLER Method on Nonorthogonal Curvilinear Coordinates: SIMPLERM, *Numerical Heat Transfer*, Part B, vol. 51, pp. 43–66, 2007.

53. J. D. Hoffman, *Numerical Methods for Engineers and Scientists*, Marcel Dekker, New York, 2001.

54. J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons Inc., England, 2003.

55. K. Dekker and J. G. Verwer, *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, North-Holland, Amsterdam, 1984.

56. J. C. Butcher, Implicit Runge-Kutta Processes, *Mathematics of Computation*, vol. 18, No. 85, pp. 50–64, 1964.

57. C. A. Kennedy and M. H. Carpenter, Additive Runge–Kutta Schemes for Convection–Diffusion–Reaction Equations, *Applied Numerical Mathematics*, vol. 44, pp. 139–181, 2003.

58. H. Bijl, M. H. Carpenter, and V. N. Vatsa, Time Integration Schemes for the Unsteady Navier-Stokes Equations, *15th AIAA Computational Fluid Dynamics Conference*, Anaheim, California, 11–14 June 2001.

59. H. Bijl, M. H. Carpenter, V. N. Vatsa, and C. A. Kennedy, Implicit Time Integration Schemes for the Unsteady Compressible Navier-Stokes Equations: Laminar Flow, *Journal of Computational Physics*, vol. 179, pp. 313–329, 2002.

60. M. H. Carpenter, S. A. Viken, and E. J. Nielsen, The Efficiency of High Order Temporal Schemes, *41st Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, *AIAA Paper*, No. 2003-0086, 6-9 January 2003.

61. S. Isono, and D. W. Zingg, A Runge-Kutta-Newton-Krylov Algorithm for Fourth-Order Implicit Time Marching Applied to Unsteady Flows, *AIAA Paper*, No. 2004-0433, 2004.

62. A. Prothero and A. Robinson, On the Stability and Accuracy of One-Step Methods for Solving Stiff Systems of Ordinary Differential Equations, *Mathematics of Computation*, vol. 28, No. 125, pp. 145–162, 1974.

63. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*, Springer-Verlag, Berlin, Germany, 1991.

64. H. Schlichting, *Boundary-Layer Theory*, McGraw-Hill, New York, 7th Edition, 1979.

65. P. K. Khosla and S. G. Rubin, A Diagonally Dominant Second-Order Accurate Implicit Scheme, *Computers and Fluids*, vol. 2, pp. 207–209, 1974.

66. B. P. Leonard, A Stable and Accurate Convective Modelling Procedure based on Quadratic Upstream Interpolation, *Computer Methods in Applied Mechanics and Engineering*, vol. 19, pp. 59–98, 1979.

67. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA Journal*, vol. 21, No. 11, pp. 1525–1532, 1983.

68. S. Majumdar, Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with Nonstaggered Grids, *Numerical Heat Transfer*, vol. 13, pp. 125–132, 1988.

69. S. K. Choi, Note on the Use of Momentum Interpolation Method for Unsteady Flows, *Numerical Heat Transfer*, Part A, vol. 36, pp. 545–550, 1999.

70. J. Papageorgakopoulos, G. Arampatzis, D. Assimacopoulos, and N. C. Markatos, Enhancement of the Momentum Interpolation Method on Non-Staggered Grids, *International Journal for Numerical Methods in Fluids*, vol. 33, pp. 1–22, 2000.

71. Bo Yu, Wen-Quan Tao, Jin-Jia Wei, Yasuo Kawaguchi, Toshio Tagawa, and Hiroyuki Ozoe, Discussion on Momentum Interpolation Method for Collocated Grids of Incompressible Flow, *Numerical Heat Transfer*, Part B, vol. 42, pp. 141–166, 2002.

72. J. C. Butcher, A History of Runge-Kutta Methods, *Applied Numerical Mathematics*, vol. 20, pp. 247–260, 1996.

73. J. C. Butcher and G. Wanner, Runge-Kutta Methods: Some Historical Notes, *Applied Numerical Mathematics*, vol. 22, pp. 113–151, 1996.

74. R. Alexander, Diagonally Implicit Runge-Kutta Methods for Stiff O.D.E.'s, *SIAM Journal on Numerical Analysis*, vol. 14, No. 6, pp. 1006–1021, 1977.

75. T. Hayase, J. A. C. Humphrey, and R. Greif, A Consistently Formulated QUICK Scheme for Fast and Stable Convergence using Finite-Volume Iterative Calculation Procedures, *Journal of Computational Physics*, vol. 98, pp. 108–118, 1992.

76. Jin-Jia Wei, Bo Yu, Wen-Quan Tao, Yasuo Kawaguchi, and Huang-sheng Wang, A New High-Order-Accurate and Bounded Scheme for Incompressible Flow, *Numerical Heat Transfer*, Part B, vol. 43, pp. 19–41, 2003.

APPENDIX A: RUNGE-KUTTA METHODS

As documented by Butcher [72] and Butcher and Wanner [73], Runge and his successors Heun, Kutta, and Nyström laid the foundation of Runge-Kutta (RK) methods during late 19^{th} and early 20^{th} century. Classical RK methods were explicit. It was during 1960s when Kuntzmann and Butcher [56] proposed implicit RK methods. Fully implicit RK methods were, however, difficult to derive and computationally inefficient. During seventies, many researchers (such as Alt, Kurdi, Nørsett, Crouzeix, and Alexander [74]) worked towards improvement in efficiency of implicit RK methods. The work of Alexander [74] is very frequently referenced. The historical works of Runge, Heun, Kutta, Nyström, Kuntzmann, Alt, Kurdi, Nørsett, Crouzeix are originally referenced by Butcher [72] and/or Butcher and Wanner [73]. The current author has not reviewed their work. For interested readers, a list of references as quoted by Butcher [72] and/or Butcher and Wanner [73] is presented in Appendix D.

At this point, it is appropriate to define *Stage* and *Order of accuracy* of RK methods. *Stage* is defined as the number of times the dependent variable or its time-derivative is calculated during every time step. *Order of accuracy* of an RK method is the level of accuracy determined by neglecting certain order terms in Taylor series expansion during the derivation of the method. In case of explicit RK methods, for a given order of accuracy, P, the required number of stages, q, may be equal to or more than the order of accuracy, i.e. $P \leq q$. But a q-stage implicit RK method can be derived for order P such that $P > q$ (Butcher [54]). Parameters A_{rs} in Eq. (A.6) are determined based on the required order, number of stages, and stability considerations.

In the following sections various types of RK methods are summarized. The following discussion is derived from the work of Alexander [74] and Dekker and Verwer [55].

A1. General Form of Runge-Kutta Methods

The purpose of RK methods is to find out approximate solution of the initial value problem:

$$\frac{d\varphi}{dt} = f\left(t, \varphi\right), \qquad \varphi\left(0\right) = \varphi_0 \tag{A.1}$$

Let h be the size of a typical n^{th} time step:

$$t_{n+1} = t_{n,0} + h \tag{A.2}$$

$\varphi(t_{n,0})$ and $\varphi(t_{n+1})$ are values of the dependent variable φ at the beginning and at the end of n^{th} time step, respectively. The concept of RK methods is to calculate $\varphi(t_{n+1})$ from $\varphi(t_{n,0})$ by approximating the integral in the following formula:

$$\varphi\left(t_{n+1}\right) = \varphi\left(t_{n,0}\right) + \int_{t_n}^{t_{n+1}} f\left(t, \varphi\left(t\right)\right) dt \tag{A.3}$$

The indices r and s, used in the following discussion, should not be confused with the indices of the grid points. Let there be q number of quadrature points defined by:

$$t_{n,r} = t_{n,0} + \tau_r h, \qquad r = 1, q \tag{A.4}$$

If b_r are the weights at quadrature points $t_{n,r}$, the following quadrature formula is used to approximate the integral in Eq. (A.3):

$$\varphi\left(t_{n+1}\right) = \varphi\left(t_{n,0}\right) + h\sum_{r=1}^{q} b_r f\left(t_{n,r}, \varphi\left(t_{n,r}\right)\right) \tag{A.5}$$

Let $\varphi_{n,0}$, $\varphi_{n,r}$, and φ_{n+1} be the approximations of $\varphi(t_{n,0})$, $\varphi(t_{n,r})$, and $\varphi(t_{n+1})$, respectively. The values $\varphi_{n,r}$ are calculated at the quadrature points defined by Eqs. (A.4) using the following quadrature formula:

$$\varphi_{n,r} = \varphi_{n,0} + h\sum_{s=1}^{q} A_{rs} f\left(t_{n,s}, \varphi_{n,s}\right), \qquad r = 1, q \tag{A.6}$$

Eqs. (A.6) are, in general, a set of q implicit equations. Solution of Eqs. (A.6) is called *stage calculations*. The values of $\varphi_{n,r}$ obtained from stage calculations are used in Eq. (A.5) to obtain *update solution* φ_{n+1}:

$$\varphi_{n+1} = \varphi_{n,0} + h\sum_{r=1}^{q} B_r f\left(t_{n,r}, \varphi_{n,r}\right) \tag{A.7}$$

Eqs. (A.6) and (A.7) define general form of Runge-Kutta methods. In every time step, q number of values of the dependent variable φ are calculated from Eqs. (A.6); therefore, the method is called q-stage method. τ_i, B_i, and A_{ij} are the parameters.

Calculation of these parameters is based on the required order, number of stages, and stability consideration. A condensed form of presentation for a Runge-Kutta method is called Butcher array:

$$\begin{array}{c|c} \tau & \mathrm{A} \\ \hline & \mathrm{B}^{\mathrm{T}} \end{array} = \begin{array}{c|ccc} \tau_1 & A_{11} & \cdots & A_{1q} \\ \vdots & \vdots & \cdots & \vdots \\ \tau_q & A_{q1} & \cdots & A_{qq} \\ \hline & B_1 & \cdots & B_q \end{array} \tag{A.8}$$

A2. Explicit Runge-Kutta Methods

If $A_{rs} = 0$ for $s \geq r$ in Butcher array, $\varphi_{n,r}$ in Eq. (A.6) can be calculated explicitly from the preceding values of $\varphi_{n,s}$. For explicit RK methods, Eqs. (A.6) and (A.7) take the following form:

$$\varphi_{n,1} = \varphi_{n,0} \tag{A.9}$$

$$\varphi_{n,r} = \varphi_{n,0} + h\sum_{s=1}^{r-1} A_{rs} f\left(t_{n,s}, \varphi_{n,s}\right), \qquad r = 2,\, q \tag{A.10}$$

$$\varphi_{n+1} = \varphi_{n,0} + h\sum_{r=1}^{q} B_r f\left(t_{n,r}, \varphi_{n,r}\right) \tag{A.11}$$

A3. Diagonally Implicit Runge-Kutta (DIRK) Methods

Runge-Kutta methods, for which $A_{ij} = 0$ for all $j > i$ in Butcher array, are called Diagonally Implicit Runge-Kutta (DIRK) methods. In DIRK methods, at every stage of

calculation in a time step, dependent variable depends on its value at that stage as well as at all previous stages. Thus a dependent variable φ at i^{th} stage of n^{th} time step is calculated from:

$$\varphi_{n,r} = \varphi_{n,0} + h\sum_{s=1}^{r} A_{rs} f\left(t_{n,s}, \varphi_{n,s}\right), \qquad r = 1, q \tag{A.12}$$

Update solution is calculated from Eq. (A.7).

A4. Singly Diagonally Implicit Runge-Kutta (SDIRK) Methods

DIRK methods for which all the diagonal elements of matrix **A** are equal to a single number, are called Singly Diagonally Implicit Runge-Kutta (SDIRK) methods. Bijl et al. [58, 59] used a more explanatory name for these methods, i.e. Single diagonal coefficient, Diagonally Implicit, Runge-Kutta methods.

A5. Explicit first stage, Single diagonal coefficient, Diagonally Implicit, Runge-Kutta (ESDIRK) Methods

ESDIRK methods are characterized by the first explicit stage. For these methods $A_{11} = 0$, so that:

$$\varphi_{n,1} = \varphi_{n,0} \tag{A.13}$$

$$\varphi_{n,r} = \varphi_{n,0} + h\sum_{s=1}^{r} A_{rs} f\left(t_{n,s}, \varphi_{n,s}\right), \qquad r = 2, q \tag{A.14}$$

A6. Stiffly Accurate Runge-Kutta Methods

An RK method is *stiffly accurate* if the last stage approximation $\varphi_{n,q}$ is equal to the update solution φ_{n+1}. This class of RK methods requires that:

$$A_{qs} = B_s, \qquad s = 1, q \tag{A.15}$$

These methods can be expressed as:

$$\varphi_{n,1} = \varphi_{n,0} \tag{A.16}$$

$$\varphi_{n,r} = \varphi_{n,0} + h\sum_{s=1}^{r} A_{rs} f\left(t_{n,s}, \varphi_{n,s}\right), \qquad r = 2,\ q-1 \tag{A.17}$$

$$\varphi_{n+1} = \varphi_{n,q} = \varphi_{n,0} + h\sum_{s=1}^{q} A_{qs} f\left(t_{n,s}, \varphi_{n,s}\right) \tag{A.18}$$

APPENDIX B: TWO EXAMPLES OF EVALUATION OF DEFERRED-CORRECTION TERM AND COEFFICIENTS IN THE DISCRETIZED MOMENTUM EQUATIONS

B1. Power Law Scheme

B1.1. Staggered-Grid Approach

When power law scheme of Patankar [1] is used in spatial discretization of momentum equations, the deferred correction terms in Eqs. (4.11) and (4.12) are zero:

$$c^u_{i,j} = c^v_{i,j} = 0 \tag{B.1}$$

In the following equations the operator $[\![\,]\!]$ is used to return the maximum of the enclosed operands. The coefficients a_W, a_E, a_S, and a_N in Eqs. (4.11) and (4.12) are evaluated from the following equations when power law scheme is used:

$$a^u_{W_{i,j}} = \frac{\mu(\Delta y)^p_j}{(\delta x_w)^u_i}\left[\!\left[0, \left[1 - 0.1\left|\frac{(u_w)^u_{i,j}(\delta x_w)^u_i}{\nu}\right|\right]^5\right]\!\right] + \left[\!\left[0, \rho(u_w)^u_{i,j}(\Delta y)^p_j\right]\!\right] \tag{B.2}$$

$$a^u_{E_{i,j}} = \frac{\mu(\Delta y)^p_j}{(\delta x_e)^u_i}\left[\!\left[0, \left[1 - 0.1\left|\frac{(u_e)^u_{i,j}(\delta x_e)^u_i}{\nu}\right|\right]^5\right]\!\right] + \left[\!\left[0, \rho(u_e)^u_{i,j}(\Delta y)^p_j\right]\!\right] \tag{B.3}$$

$$a^u_{S_{i,j}} = \frac{\mu(\Delta x)^u_i}{(\delta y_s)^p_j}\left[\left[0, \left[1-0.1\left|\frac{(v_s)^u_{i,j}(\delta y_s)^p_j}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(v_s)^u_{i,j}(\Delta x)^u_i\right]\right] \quad \text{(B.4)}$$

$$a^u_{N_{i,j}} = \frac{\mu(\Delta x)^u_i}{(\delta y_n)^p_j}\left[\left[0, \left[1-0.1\left|\frac{(v_n)^u_{i,j}(\delta y_n)^p_j}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(v_n)^u_{i,j}(\Delta x)^u_i\right]\right] \quad \text{(B.5)}$$

$$a^v_{W_{i,j}} = \frac{\mu(\Delta y)^v_j}{(\delta x_w)^p_i}\left[\left[0, \left[1-0.1\left|\frac{(u_w)^v_{i,j}(\delta x_w)^p_i}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(u_w)^v_{i,j}(\Delta y)^v_j\right]\right] \quad \text{(B.6)}$$

$$a^v_{E_{i,j}} = \frac{\mu(\Delta y)^v_j}{(\delta x_e)^p_i}\left[\left[0, \left[1-0.1\left|\frac{(u_e)^v_{i,j}(\delta x_e)^p_i}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(u_e)^v_{i,j}(\Delta y)^v_j\right]\right] \quad \text{(B.7)}$$

$$a^v_{S_{i,j}} = \frac{\mu(\Delta x)^p_i}{(\delta y_s)^v_j}\left[\left[0, \left[1-0.1\left|\frac{(v_s)^v_{i,j}(\delta y_s)^v_j}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(v_s)^v_{i,j}(\Delta x)^p_i\right]\right] \quad \text{(B.8)}$$

$$a^v_{N_{i,j}} = \frac{\mu(\Delta x)^p_i}{(\delta y_n)^v_j}\left[\left[0, \left[1-0.1\left|\frac{(v_n)^v_{i,j}(\delta y_n)^v_j}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(v_n)^v_{i,j}(\Delta x)^p_i\right]\right] \quad \text{(B.9)}$$

CV face velocities used in Eqs. (B.2)

through (B.9) are usually calculated by linear interpolation.

B1.2. Non-Staggered Grid Approach

When power law scheme of Patankar [1] is used in spatial discretization of

momentum equations, the deferred correction terms in Eqs. (4.53) and (4.54) are zero:

$$c_{i,j}^{u} = c_{i,j}^{v} = 0 \tag{B.10}$$

The coefficients a_W, a_E, a_S, and a_N, that are identical for both the evolution equations, Eqs. (4.53) and (4.54), are evaluated from the following equations when power law scheme is used:

$$a_{W_{i,j}} = \frac{\mu(\Delta y)_j}{(\delta x_w)_i}\left[\left[0, \left[1 - 0.1\left|\frac{(u_w)_{i,j}(\delta x_w)_i}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(u_w)_{i,j}(\Delta y)_j\right]\right] \tag{B.11}$$

$$a_{E_{i,j}} = \frac{\mu(\Delta y)_j}{(\delta x_e)_i}\left[\left[0, \left[1 - 0.1\left|\frac{(u_e)_{i,j}(\delta x_e)_i}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(u_e)_{i,j}(\Delta y)_j\right]\right] \tag{B.12}$$

$$a_{S_{i,j}} = \frac{\mu(\Delta x)_i}{(\delta y_s)_j}\left[\left[0, \left[1 - 0.1\left|\frac{(v_s)_{i,j}(\delta y_s)_j}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(v_s)_{i,j}(\Delta x)_i\right]\right] \tag{B.13}$$

$$a_{N_{i,j}} = \frac{\mu(\Delta x)_i}{(\delta y_n)_j}\left[\left[0, \left[1 - 0.1\left|\frac{(v_n)_{i,j}(\delta y_n)_j}{\nu}\right|\right]^5\right]\right] + \left[\left[0, \rho(v_n)_{i,j}(\Delta x)_i\right]\right] \tag{B.14}$$

CV face velocities used in Eqs. (B.11) through (B.14) are usually calculated by momentum interpolation.

B2. QUICK Scheme

B2.1. Staggered Grid Approach

Consistent formulation of QUICK scheme was provided by Hayase et al. [75] which was used by Yu et al. [71] and Wei et al. [76] in a general formulation. Following the work of these authors, deferred correction term *c* in Eqs. (4.11) and (4.12) is calculated as below:

$$\begin{aligned} c^u_{i,j} = &+\left[\!\left[0,\rho\left(u_w\right)^u_{i,j}\left(\Delta y\right)^p_j\right]\!\right]\left(\left(u_w\right)^u_{i,j}-u_{i-1,j}\right)-\left[\!\left[0,\rho\left(-u_w\right)^u_{i,j}\left(\Delta y\right)^p_j\right]\!\right]\left(\left(u_w\right)^u_{i,j}-u_{i,j}\right)\\ &+\left[\!\left[0,\rho\left(-u_e\right)^u_{i,j}\left(\Delta y\right)^p_j\right]\!\right]\left(\left(u_e\right)^u_{i,j}-u_{i+1,j}\right)-\left[\!\left[0,\rho\left(u_e\right)^u_{i,j}\left(\Delta y\right)^p_j\right]\!\right]\left(\left(u_e\right)^u_{i,j}-u_{i,j}\right)\\ &+\left[\!\left[0,\rho\left(v_s\right)^u_{i,j}\left(\Delta x\right)^u_i\right]\!\right]\left(\left(v_s\right)^u_{i,j}-v_{i,j-1}\right)-\left[\!\left[0,\rho\left(-v_s\right)^u_{i,j}\left(\Delta x\right)^u_i\right]\!\right]\left(\left(v_s\right)^u_{i,j}-v_{i,j}\right)\\ &+\left[\!\left[0,\rho\left(-v_n\right)^u_{i,j}\left(\Delta x\right)^u_i\right]\!\right]\left(\left(v_n\right)^u_{i,j}-v_{i,j+1}\right)-\left[\!\left[0,\rho\left(v_n\right)^u_{i,j}\left(\Delta x\right)^u_i\right]\!\right]\left(\left(v_n\right)^u_{i,j}-v_{i,j}\right) \end{aligned} \tag{B.15}$$

$$\begin{aligned} c^v_{i,j} = &+\left[\!\left[0,\rho\left(u_w\right)^v_{i,j}\left(\Delta y\right)^v_j\right]\!\right]\left(\left(v_w\right)^v_{i,j}-v_{i-1,j}\right)-\left[\!\left[0,\rho\left(-u_w\right)^v_{i,j}\left(\Delta y\right)^v_j\right]\!\right]\left(\left(v_w\right)^v_{i,j}-v_{i,j}\right)\\ &+\left[\!\left[0,\rho\left(-u_e\right)^v_{i,j}\left(\Delta y\right)^v_j\right]\!\right]\left(\left(v_e\right)^v_{i,j}-v_{i+1,j}\right)-\left[\!\left[0,\rho\left(u_e\right)^v_{i,j}\left(\Delta y\right)^v_j\right]\!\right]\left(\left(v_e\right)^v_{i,j}-v_{i,j}\right)\\ &+\left[\!\left[0,\rho\left(v_s\right)^v_{i,j}\left(\Delta x\right)^p_i\right]\!\right]\left(\left(v_s\right)^v_{i,j}-v_{i,j-1}\right)-\left[\!\left[0,\rho\left(-v_s\right)^v_{i,j}\left(\Delta x\right)^p_i\right]\!\right]\left(\left(v_s\right)^v_{i,j}-v_{i,j}\right)\\ &+\left[\!\left[0,\rho\left(-v_n\right)^v_{i,j}\left(\Delta x\right)^p_i\right]\!\right]\left(\left(v_n\right)^v_{i,j}-v_{i,j+1}\right)-\left[\!\left[0,\rho\left(v_n\right)^v_{i,j}\left(\Delta x\right)^p_i\right]\!\right]\left(\left(v_n\right)^v_{i,j}-v_{i,j}\right) \end{aligned} \tag{B.16}$$

When QUICK scheme is used, the coefficients in Eqs. (4.11) and (4.12) are given as:

$$a^u_{W_{i,j}} = \frac{\mu\left(\Delta y\right)^p_j}{\left(\delta x_w\right)^u_i}+\left[\!\left[0,\rho\left(u_w\right)^u_{i,j}\left(\Delta y\right)^p_j\right]\!\right] \tag{B.17}$$

$$a^u_{E_{i,j}} = \frac{\mu(\Delta y)^p_j}{(\delta x_e)^u_i} + \left[\!\left[0, \rho(u_e)^u_{i,j}(\Delta y)^p_j\right]\!\right] \tag{B.18}$$

$$a^u_{S_{i,j}} = \frac{\mu(\Delta x)^u_i}{(\delta y_s)^u_j} + \left[\!\left[0, \rho(v_s)^u_{i,j}(\Delta x)^u_i\right]\!\right] \tag{B.19}$$

$$a^u_{N_{i,j}} = \frac{\mu(\Delta x)^u_i}{(\delta y_n)^u_j} + \left[\!\left[0, \rho(v_n)^u_{i,j}(\Delta x)^u_i\right]\!\right] \tag{B.20}$$

CV face velocities used in Eqs. (B.17) through (B.20) are calculated by interpolation. QUICK interpolation formulae for *u*-velocity at east face of a *u*-CV are given below:

For $(u_e)^u_{i,j} \geq 0$:

$$\begin{aligned}(u_e)^u_{i,j} = & \left\{\frac{-(\Delta x)^p_i (\Delta x)^p_{i+1}}{\left[(\Delta x)^p_{i-1} + (\Delta x)^p_i\right]\left[(\Delta x)^p_{i-1} + 2(\Delta x)^p_i + (\Delta x)^p_{i+1}\right]}\right\} u_{i-1,j} \\ & + \left\{\frac{(\Delta x)^p_{i+1}\left[(\Delta x)^p_{i-1} + 2(\Delta x)^p_i\right]}{\left[(\Delta x)^p_{i-1} + (\Delta x)^p_i\right]\left[(\Delta x)^p_i + (\Delta x)^p_{i+1}\right]}\right\} u_{i,j} \\ & + \left\{\frac{(\Delta x)^p_i\left[(\Delta x)^p_{i-1} + 2(\Delta x)^p_i\right]}{\left[(\Delta x)^p_{i-1} + 2(\Delta x)^p_i + (\Delta x)^p_{i+1}\right]\left[(\Delta x)^p_i + (\Delta x)^p_{i+1}\right]}\right\} u_{i+1,j}\end{aligned} \tag{B.21}$$

For $(u_e)^u_{i,j} < 0$:

$$
\begin{aligned}
\left(u_e\right)_{i,j}^{u} = & \left\{\frac{\left(\Delta x\right)_{i+1}^{p}\left[2\left(\Delta x\right)_{i+1}^{p}+\left(\Delta x\right)_{i+2}^{p}\right]}{\left[\left(\Delta x\right)_{i}^{p}+2\left(\Delta x\right)_{i+1}^{p}+\left(\Delta x\right)_{i+2}^{p}\right]\left[\left(\Delta x\right)_{i}^{p}+\left(\Delta x\right)_{i+1}^{p}\right]}\right\}u_{i,j} \\
& +\left\{\frac{\left(\Delta x\right)_{i}^{p}\left[2\left(\Delta x\right)_{i+1}^{p}+\left(\Delta x\right)_{i+2}^{p}\right]}{\left[\left(\Delta x\right)_{i}^{p}+\left(\Delta x\right)_{i+1}^{p}\right]\left[\left(\Delta x\right)_{i+1}^{p}+\left(\Delta x\right)_{i+2}^{p}\right]}\right\}u_{i+1,j} \\
& +\left\{\frac{-\left(\Delta x\right)_{i}^{p}\left(\Delta x\right)_{i+1}^{p}}{\left[\left(\Delta x\right)_{i+1}^{p}+\left(\Delta x\right)_{i+2}^{p}\right]\left[\left(\Delta x\right)_{i}^{p}+2\left(\Delta x\right)_{i+1}^{p}+\left(\Delta x\right)_{i+2}^{p}\right]}\right\}u_{i+2,j}
\end{aligned}
\tag{B.22}
$$

Velocity field used in Eqs. (B.17) through (B.22) is taken from preceding iteration; use of superscript *l* is avoided for the sake of simplicity. *u*-velocity at the west face of a CV and *v*-velocity at the south and north faces are calculated from equations similar to Eqs. (B.21) and (B.22).

B2.2. Non-staggered Grid Approach

Deferred correction term *c* in Eqs. (4.53) and (4.54) is calculated as below:

$$
\begin{aligned}
c_{i,j}^{u} = & +\left[\left[0,\rho\left(u_w\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(u_w\right)_{i,j}-u_{i-1,j}\right)-\left[\left[0,\rho\left(-u_w\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(u_w\right)_{i,j}-u_{i,j}\right) \\
& +\left[\left[0,\rho\left(-u_e\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(u_e\right)_{i,j}-u_{i+1,j}\right)-\left[\left[0,\rho\left(u_e\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(u_e\right)_{i,j}-u_{i,j}\right) \\
& +\left[\left[0,\rho\left(v_s\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_s\right)_{i,j}-v_{i,j-1}\right)-\left[\left[0,\rho\left(-v_s\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_s\right)_{i,j}-v_{i,j}\right) \\
& +\left[\left[0,\rho\left(-v_n\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_n\right)_{i,j}-v_{i,j+1}\right)-\left[\left[0,\rho\left(v_n\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_n\right)_{i,j}-v_{i,j}\right)
\end{aligned}
\tag{B.23}
$$

$$
\begin{aligned}
c_{i,j}^{v} = & +\left[\left[0,\rho\left(u_w\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(v_w\right)_{i,j}-v_{i-1,j}\right)-\left[\left[0,\rho\left(-u_w\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(v_w\right)_{i,j}-v_{i,j}\right) \\
& +\left[\left[0,\rho\left(-u_e\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(v_e\right)_{i,j}-v_{i+1,j}\right)-\left[\left[0,\rho\left(u_e\right)_{i,j}\left(\Delta y\right)_j\right]\right]\left(\left(v_e\right)_{i,j}-v_{i,j}\right) \\
& +\left[\left[0,\rho\left(v_s\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_s\right)_{i,j}-v_{i,j-1}\right)-\left[\left[0,\rho\left(-v_s\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_s\right)_{i,j}-v_{i,j}\right) \\
& +\left[\left[0,\rho\left(-v_n\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_n\right)_{i,j}-v_{i,j+1}\right)-\left[\left[0,\rho\left(v_n\right)_{i,j}\left(\Delta x\right)_i\right]\right]\left(\left(v_n\right)_{i,j}-v_{i,j}\right)
\end{aligned}
\tag{B.24}
$$

When QUICK scheme is used, the coefficients in Eqs. (4.53) and (4.54) are given as:

$$a_{W_{i,j}} = \frac{\mu(\Delta y)_j}{(\delta x_w)_i} + \left[\!\left[0, \rho(u_w)_{i,j} (\Delta y)_j \right]\!\right] \tag{B.25}$$

$$a_{E_{i,j}} = \frac{\mu(\Delta y)_j}{(\delta x_e)_i} + \left[\!\left[0, \rho(u_e)_{i,j} (\Delta y)_j \right]\!\right] \tag{B.26}$$

$$a_{S_{i,j}} = \frac{\mu(\Delta x)_i}{(\delta y_s)_j} + \left[\!\left[0, \rho(v_s)_{i,j} (\Delta x)_i \right]\!\right] \tag{B.27}$$

$$a_{N_{i,j}} = \frac{\mu(\Delta x)_i}{(\delta y_n)_j} + \left[\!\left[0, \rho(v_n)_{i,j} (\Delta x)_i \right]\!\right] \tag{B.28}$$

CV face velocities used in Eqs. (B.23) through (B.28) are calculated by quadratic interpolation as explained by Papageorgakopoulos et al. [70].

APPENDIX C: FORMULATION FOR SIMPLE DIRK METHOD USING A FOUR-STAGE ESDIRK METHOD

The method presented in this dissertation can be used with ESDIRK method of any order P with any number of stages q. The number of stages and the parameters A_{rs} in the Butcher array will depend on the chosen method. In this dissertation, the feasibility of the presented method was tested with one of the simplest ESDIRK methods which involved calculations for only one stage. In the following subsections, formulation is presented for a four-stage ESDIRK method as an example considering a typical n^{th} time step. This formulation will be helpful for future work on the use of higher-order ESDIRK methods in the presented simulation method.

C1. Staggered Grid Approach

Eqs. (4.8), (4.19), and (4.20) are solved with SIMPLE algorithm in every stage except for the first stage. No calculations are required in the first stage because $u_{n,1}$ and $v_{n,1}$ are set equal to $u_{n,0}$ and $v_{n,0}$ respectively as given by Eq. (4.13) and corresponding equation for v-velocity. Given below are the source terms $b^u_{i,j}$ and $b^v_{i,j}$ and the coefficients $a^u_{i,j}$ and $a^v_{i,j}$ to be used in the solution of Eqs. (4.19) and (4.20) in stage 2 through 4. The deferred correction terms c^u and c^v and the coefficients a_W, a_E, a_S, and a_N are calculated from the current available velocity field using some spatial discretization scheme of choice.

Stage 2 (r = 2):

$$b_{i,j}^{u} = c_{i,j}^{u} + \frac{\rho(\Delta x)_i^{u}(\Delta y)_j^{p}}{hA_{22}}\left\{\left(u_{i,j}\right)_0 + hA_{21}\left(f_{i,j}^{u}\right)_1\right\} \tag{C.1}$$

$$b_{i,j}^{v} = c_{i,j}^{v} + \frac{\rho(\Delta x)_i^{p}(\Delta y)_j^{v}}{hA_{22}}\left\{\left(v_{i,j}\right)_0 + hA_{21}\left(f_{i,j}^{v}\right)_1\right\} \tag{C.2}$$

$$a_{i,j}^{u} = a_{W_{i,j}}^{u} + a_{E_{i,j}}^{u} + a_{S_{i,j}}^{u} + a_{N_{i,j}}^{u} + \rho(\Delta x)_i^{u}(\Delta y)_j^{p} / hA_{22} \tag{C.3}$$

$$a_{i,j}^{v} = a_{W_{i,j}}^{v} + a_{E_{i,j}}^{v} + a_{S_{i,j}}^{v} + a_{N_{i,j}}^{v} + \rho(\Delta x)_i^{p}(\Delta y)_j^{v} / hA_{22} \tag{C.4}$$

Stage 3 (r = 3):

$$b_{i,j}^{u} = c_{i,j}^{u} + \frac{\rho(\Delta x)_i^{u}(\Delta y)_j^{p}}{hA_{33}}\left\{\left(u_{i,j}\right)_0 + hA_{31}\left(f_{i,j}^{u}\right)_1 + hA_{32}\left(f_{i,j}^{u}\right)_2\right\} \tag{C.5}$$

$$b_{i,j}^{v} = c_{i,j}^{v} + \frac{\rho(\Delta x)_i^{p}(\Delta y)_j^{v}}{hA_{33}}\left\{\left(v_{i,j}\right)_0 + hA_{31}\left(f_{i,j}^{v}\right)_1 + hA_{32}\left(f_{i,j}^{v}\right)_2\right\} \tag{C.6}$$

$$a_{i,j}^{u} = a_{W_{i,j}}^{u} + a_{E_{i,j}}^{u} + a_{S_{i,j}}^{u} + a_{N_{i,j}}^{u} + \rho(\Delta x)_i^{u}(\Delta y)_j^{p} / hA_{33} \tag{C.7}$$

$$a_{i,j}^{v} = a_{W_{i,j}}^{v} + a_{E_{i,j}}^{v} + a_{S_{i,j}}^{v} + a_{N_{i,j}}^{v} + \rho(\Delta x)_i^{p}(\Delta y)_j^{v} / hA_{33} \tag{C.8}$$

Stage 4 (r = 4):

$$b_{i,j}^{u} = c_{i,j}^{u} + \frac{\rho(\Delta x)_i^{u}(\Delta y)_j^{p}}{hA_{44}}\left\{\left(u_{i,j}\right)_0 + hA_{41}\left(f_{i,j}^{u}\right)_1 + hA_{42}\left(f_{i,j}^{u}\right)_2 + hA_{43}\left(f_{i,j}^{u}\right)_3\right\} \tag{C.9}$$

$$b_{i,j}^{v} = c_{i,j}^{v} + \frac{\rho(\Delta x)_i^{p}(\Delta y)_j^{v}}{hA_{44}}\left\{\left(v_{i,j}\right)_0 + hA_{41}\left(f_{i,j}^{v}\right)_1 + hA_{42}\left(f_{i,j}^{v}\right)_2 + hA_{43}\left(f_{i,j}^{v}\right)_3\right\} \tag{C.10}$$

$$a_{i,j}^{u} = a_{W_{i,j}}^{u} + a_{E_{i,j}}^{u} + a_{S_{i,j}}^{u} + a_{N_{i,j}}^{u} + \rho(\Delta x)_i^{u}(\Delta y)_j^{p} / hA_{44} \tag{C.11}$$

$$a_{i,j}^{v} = a_{W_{i,j}}^{v} + a_{E_{i,j}}^{v} + a_{S_{i,j}}^{v} + a_{N_{i,j}}^{v} + \rho(\Delta x)_i^{p}(\Delta y)_j^{v} / hA_{44} \tag{C.12}$$

C2. Non-staggered Grid Approach

Similar to the staggered grid method, Eqs. (4.50), (4.57), and (4.60) are solved with SIMPLE algorithm in every stage except for the first stage. No calculations are required in the first stage because $u_{n,1}$ and $v_{n,1}$ are set equal to $u_{n,0}$ and $v_{n,0}$ respectively (Eq. (4.13) and corresponding equation for v-velocity). Given below are the source terms $b_{i,j}^{u}$ and $b_{i,j}^{v}$ and the coefficients $a_{i,j}$ to be used in the solution of Eqs. (4.57) and (4.60) in stage 2 through 4. The deferred correction terms c^{u} and c^{v} and the coefficients a_W, a_E, a_S, and a_N are calculated from the current available velocity field using some spatial discretization scheme of choice.

Stage 2 (r = 2):

$$b_{i,j}^{u} = c_{i,j}^{u} + \frac{\rho(\Delta x)_i (\Delta y)_j}{hA_{22}} \left\{ \left(u_{i,j}\right)_0 + hA_{21} \left(f_{i,j}^{u}\right)_1 \right\} \tag{C.13}$$

$$b_{i,j}^{v} = c_{i,j}^{v} + \frac{\rho(\Delta x)_i (\Delta y)_j}{hA_{22}} \left\{ \left(v_{i,j}\right)_0 + hA_{21} \left(f_{i,j}^{v}\right)_1 \right\} \tag{C.14}$$

$$a_{i,j} = a_{W_{i,j}} + a_{E_{i,j}} + a_{S_{i,j}} + a_{N_{i,j}} + \rho(\Delta x)_i (\Delta y)_j / hA_{22} \tag{C.15}$$

Stage 3 (r = 3):

$$b_{i,j}^{u} = c_{i,j}^{u} + \frac{\rho(\Delta x)_i (\Delta y)_j}{hA_{33}} \left\{ \left(u_{i,j}\right)_0 + hA_{31} \left(f_{i,j}^{u}\right)_1 + hA_{32} \left(f_{i,j}^{u}\right)_2 \right\} \tag{C.16}$$

$$b_{i,j}^{v} = c_{i,j}^{v} + \frac{\rho(\Delta x)_i (\Delta y)_j}{hA_{33}} \left\{ \left(v_{i,j}\right)_0 + hA_{31} \left(f_{i,j}^{v}\right)_1 + hA_{32} \left(f_{i,j}^{v}\right)_2 \right\} \tag{C.17}$$

$$a_{i,j} = a_{W_{i,j}} + a_{E_{i,j}} + a_{S_{i,j}} + a_{N_{i,j}} + \rho(\Delta x)_i (\Delta y)_j / hA_{33} \tag{C.18}$$

Stage 4 (r = 4):

$$b_{i,j}^{u} = c_{i,j}^{u} + \frac{\rho(\Delta x)_i (\Delta y)_j}{hA_{44}} \left\{ \left(u_{i,j}\right)_0 + hA_{41} \left(f_{i,j}^{u}\right)_1 + hA_{42} \left(f_{i,j}^{u}\right)_2 + hA_{43} \left(f_{i,j}^{u}\right)_3 \right\} \tag{C.19}$$

$$b_{i,j}^{v} = c_{i,j}^{v} + \frac{\rho(\Delta x)_i (\Delta y)_j}{hA_{44}} \left\{ \left(v_{i,j}\right)_0 + hA_{41} \left(f_{i,j}^{v}\right)_1 + hA_{42} \left(f_{i,j}^{v}\right)_2 + hA_{43} \left(f_{i,j}^{v}\right)_3 \right\} \tag{C.20}$$

$$a_{i,j} = a_{W_{i,j}} + a_{E_{i,j}} + a_{S_{i,j}} + a_{N_{i,j}} + \rho\left(\Delta x\right)_i \left(\Delta y\right)_j / hA_{44} \tag{C.21}$$

APPENDIX D: LIST OF REFERENCES AS QUOTED BY BUTCHER [72] AND/OR BUTCHER AND WANNER [73]

i) C. Runge, Über die numerisehe Auflösing yon Differentialgleichungen, *Math. Ann.*, vol. 46, pp. 167–178, 1895.

ii) K. Heun, Neue Methoden zur approximativen Integration der Differentialgleichungen eincr unabhängigen Veränderlichen, *Z. Math. Phys.*, vol. 45, pp. 23–38, 1900.

iii) W. Kutta, Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, *Z. Math. Phys.*, vol. 46, pp. 435–453, 1901.

iv) E. J. Nyström, Über die numerisehe Integration yon Differentialgleiehungen, *Acta Soc. Sci. Fennicæ*, vol. 50, No. 13, p. 55, 1925.

v) J. Kuntzmann, Neure Entwicklungen der Methoden yon Runge und Kutta, *Z. Angew. Math. Mech.*, pp. T28–T31, 1961.

vi) R. Alt, Méthodes A-stables pour l'intégration de systèmes différentielles ma1 conditionnés, Thèse présentée a l'Université Paris VI, Paris, 1971.

vii) M. A. Kurdi, Stable High Order Methods for Time Discretization of Stiff Differential Equations, Thesis, University of California, 1974.

viii) S. P. Nørsett, Semi-explicit Runge-Kutta Methods, Department of Mathematics, Report No. 6/74, University of Trondheim, 1974.

ix) M. Crouzeix, Sur l'approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge Kutta, Thèse présentée a l'Université Paris VI, Paris, 1975.

VITA

Muhammad Ijaz received his Bachelor of Science degree in Mechanical Engineering from The University of Engineering and Technology, Lahore, Pakistan in 1995. He worked in chemical industry for four years as a Mechanical/Project Engineer before he entered the graduate program at Texas A&M University in August 2000. Mr. Ijaz received his Master of Engineering and Doctor of Philosophy degrees in August 2003 and December 2007 respectively both in Mechanical Engineering. Throughout his academic career at Texas A&M University, he worked as either Instructor or Teaching Assistant in the areas of Thermal/Fluid Science and Engineering Design. He also served several student organizations. He is a Fellow of Graduate Teaching Academy of Texas A&M University and served for two consecutive years as a Mentor in Teaching Assistants Training and Evaluation Program administered by Center for Teaching Excellence. He was selected for membership in The Honor Society of Phi Kappa Phi in 2002. Mr. Ijaz is a member of American Society of Mechanical Engineers (ASME), American Society for Engineering Education (ASEE), and American Institute of Aeronautics and Astronautics (AIAA). He is currently a faculty member in the Department of Mechanical Engineering at Texas A&M University. His research interests, in general, include improvement in understanding of physics of fluid flow and heat transfer and enhancement of predictability in engineering design. In particular, he is interested in developing higher-order accurate simulation methods for transient multi-phase flow and heat transfer.

Mr. Ijaz may be reached at Texas A&M University, Department of Mechanical Engineering, 3123 TAMU, College Station TX 77843-3123, E-mail: ijazansari@tamu.edu.