2009

Development of unsteady algorithms for pressure-based unstructured solver for two-dimensional incompressible flows

Angela Dwi Lestari
Iowa State University

Development of unsteady algorithms for pressure-based unstructured solver for two-dimensional incompressible flows

by

Angela Lestari

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Aerospace Engineering

Program of Study Committee:
R. Ganesh Rajagopalan, Major Professor
Z.J. Wang
Mani Mina

Iowa State University

Ames, Iowa

2009

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

a_{nb}	coefficients in the discretization equations
A, B, C	constants for the shape function of ϕ
C_d	drag coefficient
C_l	lift coefficient
C_p	pressure coefficient
c	airfoil chord
dA	incremental area of a control volume
dl	incremental length of the control volume face
d^u, d^v	source term coefficients for the u and v momentum equations
f_i, g_i	functions representing the contribution of information from grid point i to the flux at the control volume face
F^u, F^v	u and v momentum flux leaving the domain around a boundary point
$\vec{J}$	total momentum flux of the general variable ϕ
$\vec{J}^u, \vec{J}^v$	total momentum flux for the variables u and v
LHS	left–hand side of the governing equation
$\dot{m}$	mass flux leaving the domain around a boundary point
Pe_Δ	element Peclet number
p	pressure
p'	pressure correction
p_∞	freestream pressure
q_∞	freestream dynamic pressure

r, s, t	points on the face of the control volume
R	mass terms in the pressure equation
RHS	right–hand side of the governing equation
Re	Reynolds number
S^u, S^v	u and v momentum source terms for a triangle
U, V	velocity components in the local coordinate system
u, v	velocity components in the global coordinate system
u', v'	velocity components used to correct the nodal velocity field
$\hat{u}, \hat{v}$	pseudo velocity components
$\tilde{u}, \tilde{v}$	artificial velocity components used in the equal–order formulation
$\tilde{u}', \tilde{v}'$	velocities used to correct the artificial velocity field
U_{avg}	average velocity for a triangle
u_{avg}, v_{avg}	velocity components of the average velocity for a triangle
V_∞	freestream velocity
$\vec{V}$	velocity vector
X, Y	coordinates in the local coordinate system
x, y	coordinates in the global coordinate system
ΔX	characteristic length of a triangle
X_{min}, X_{max}	minimum and maximum X components for a triangle in the local coordinate system
Z	variable for the exponential and power–law functions used in the shape function for ϕ

Greek Symbols

α, β, γ	constants in the pressure interpolation function
α	angle of attack
μ	fluid molecular viscosity

ρ	fluid density
θ_r	angle of the local coordinate system with respect to the global coordinate system
Δ	determinant of the system of equations used to calculate the constants of the shape function
$\bar{\Delta}$	determinant of the system of equations used to calculate the constants of the pressure interpolation function
ϕ	general, transport variable used to represent u and v

ACKNOWLEDGEMENTS

Many thanks and deep gratitude goes to my major professor, Dr. R. Ganesh Rajagopalan, for his guidance and support throughout this research. I would also like to thank my committee members: Dr. Z.J. Wang and Dr. Mani Mina, for their support. Many thanks are also due to Janani Murallidharan, Sayan Ghosh and Sunny Park for their help through discussion and moral support, and LIFE members for their prayers and encouragement. I would like to thank Bleu'au for his help and timely presence. I would also like to acknowledge Sukra Helitek for the experience and knowledge gained during my tenure, financial support, and computing resources. Many thanks are due to the Department of Aerospace Engineering at Iowa State University and its staff for administrative help and computing resources. Finally, I would like to express my deepest gratitude to my parents and family, without whose support and encouragement, this would not be possible. And to God, who makes all things possible, I am ever grateful.

CHAPTER 1. INTRODUCTION

1.1 Background and Previous Research

Recent progress in computer power has allowed the use of CFD (computational fluid dynamics) to solve various real-life engineering problems. Experiments are generally time-consuming and relatively expensive. Thus, the use of CFD can reduce the time and cost, and complement the results obtained from experiments. Since many engineering problems are time dependent, time-accurate simulations are mandatory for these problems.

In the recent decade, unstructured grids have been popular due to two main reasons. First, unstructured grids are easy to generate compared to structured grids, especially for complex geometries. Although structured orthogonal grids offer the best accuracy, especially for viscous flow problems, they are generally very difficult and time-consuming to generate. The grid generation process usually cannot be automated, whereas for unstructured grid, it can be automated more easily, reducing the time and manpower. Another advantage of unstructured grids is the capability of local refinement. In structured grids, when the mesh density of a region is refined, it will affect the whole computational domain, creating unnecessary grid points and reducing the efficiency of the simulation. On the contrary, for unstructured grids, refinement will be local to the region of interest and will not affect the whole computational domain. Even though the memory requirements and run time for an unstructured grid are generally more expensive compared with a structured grid with the same number of points, the savings in grid generation time outweigh the disadvantages.

In general, the numerical algorithms used to solve the Navier-Stokes equations can be divided into density-based and pressure-based [1, 2]. The density-based algorithms are suitable for solving high Mach number flows; however, for low Mach number, due to the nature of the mathematical equations they become unstable and the convergence rate is significantly decreased. Several methods have been attempted to extend the density-based approaches to handle low Mach number flows, such as preconditioning [3, 4, 5], and pseudo-compressibility [6, 7]. However, they are not efficient to solve incompressible flows. Therefore, special algorithms have been developed for solving the incompressible Navier-Stokes equations which are also called pressure-based algorithms.

Unfortunately, the pressure-based algorithms have their own drawbacks. For compressible flows, the pressure is related to density through the equation of state. However, in the incompressible flow equations, no explicit equation for pressure is directly available. The pressure has to be indirectly specified via the continuity equation. Two approaches are used to handle this problem, namely direct approach and segregated approach. In the direct or coupled approach [8, 9, 10], the whole set of discretized momentum and continuity equations are solved simultaneously, resulting in a stronger functional relation between pressure and velocity. The disadvantage of this approach is that it requires a large amount of computer memory to store all the coefficients, making it not pragmatic to solve practical engineering problems [11]. Moreover, the coefficients have to be repeatedly calculated; thus the process is uneconomical.

In the segregated approach, the pressure and velocity are solved sequentially. The pressure is determined from a given velocity field. The problem is how to improve the pressure field so that the velocity field satisfies the continuity equation. Another option is to eliminate the pressure from the governing equations as is done in the vorticity-stream function method. In the vorticity-stream function approach, the pressure is removed from the momentum equation through cross differentiation, and the velocity is replaced by the vorticity and stream function. A drawback of this method is that the boundary conditions at the walls are difficult to specify. Moreover, the pressure field is usually a desired output, so an extra Poisson equation needs to be solved to obtain the pressure. The biggest limitation, however, is that the method cannot be easily extended to 3-D.

The pressure and velocity components are generally called the primitive variables, and the formulations to directly solve pressure and velocity are referred to as primitive variables formulation. The pressure-based method with primitive variables and segregated approach are generally preferred to solve the incompressible equations. The fractional step method [12], penalty method [13, 14, 15, 16], artificial compressibility method [6, 17, 18, 19], and pressure-correction method are included in this category. The fractional step or projection method was introduced by Chorin [20] and Temam [21], but was reformulated and popularized by Kim and Moin [12]. The basic idea of this method is to eliminate the the singular matrix that arises from the low Mach number limit and replace it with some proper submatrices by using a factorization technique. However, the method often introduces a spurious numerical boundary layer which leads to substantial time differencing errors. The penalty-function formulation eliminates the pressure from the momentum equations using a "penalty" parameter and solves the modified momentum equations for the velocity components. The artificial compressibility method adds a pseudo-time derivative of pressure to the continuity equation, and the equations are iterated within a given time level to obtain a divergent-free velocity field. However, this method is only suitable for steady state problems. The most popular formulation was first proposed by Harlow and Welch [22] for their Marker and Cell (MAC) scheme. They introduced staggered mesh, and segregated approach for solving the primitive variables. Patankar and Spalding [23] extended this method with the SIMPLE algorithm (Semi-Implicit Method for Pressure-Linked Equations). In this procedure, they introduced the concept of pressure-correction. Basically, a pressure correction is introduced to improve the guessed velocity so that it satisfies the continuity equation better. Extensions and improvements to the SIMPLE algorithm have been developed such as SIMPLER, SIMPLEC, SIMPLEX, etc., that are called the SIMPLE family of algorithms. Besides the SIMPLE family, other pressure correction methods have been developed that include PRIME [24], PISO [25], and CLEAR [26, 27]. Most of these methods have been reviewed by Moukalled and Darwish [28].

For finite element and finite volume methods, the location of the variables and the choice of the control volume of integration in the spatial discretization are important. In general, the arrangement of the variables can be categorized as collocated and staggered. On a collocated arrangement, the pressure and velocity are located at the same point, while on a staggered grid, they are placed at different physical locations. Thus, the control volume for velocity and pressure may not be the same, depending on the variable arrangement. An example would be to have the pressure at the cell center and the velocity at the cell face. Although a collocated grid arrangement is much simpler, it gives rise to the decoupling of pressure from velocity resulting in spurious pressure, which is also referred to as checkerboard pressure. Rhie and Chow proposed a solution by adding artificial diffusion terms in the mass conservation equation. This method works by adding a corrective term to the mass fluxes at the control volume faces, which is the basis of the pressure-weighted interpolation method (PWIM). Miller and Schmidt [29], Majumdar [30], Lambropoulos [31] also further extended the method to improve the approximation of the cell face velocity. Another remedy to the spurious pressure problem is to use a staggered mesh. As mentioned earlier, the concept of staggered mesh was introduced by Harlow and Welch [22] for their Marker and Cell (MAC) scheme for structured grids. This scheme, in addition to eliminating the problem with checkerboard pressure, has good conservation properties without the need for artificial boundary condition.

The numerical methods discussed so far are mostly formulated for structured grids. As unstructured grids have become more popular, efforts to extend the methods to unstructured grids have been attempted. Among them are by Davidson [32], Lien [33] and Chen [34] which uses SIMPLE-like method on a collocated arrangement with the Rhie and Chow interpolation. A semi-staggered mesh has been used by Thomadakis and Leschziner [35]. They used different control volumes for momentum and pressure correction, but for some cases, the semi-staggered grid still results in spurious pressure oscillations. Unfortunately, the extension of staggered mesh to unstructured grids is not as straightforward due to geometry complexity. Physical staggering of grids have been done by several researchers [36, 37, 38, 39, 40, 41, 42, 43] for triangular unstructured grids. Wenneker and Segal [42, 43] developed a method that can be used for both incompressible and compressible flows on triangular grids. Later, the method was further extended by Vidovic [44] to be superlinearly convergent in space. For staggered grids, the velocity field needs to be reconstructed from the staggered data with sufficient accuracy. He proposed a new linear reconstruction of staggered velocity fields with special treatment of divergence. To overcome the unphysical oscillations near discontinuities commonly encountered when using high order methods, he proposed to combine an upwind-biased finite volume scheme and flux limiting approach, formulated for unstructured staggered schemes.

Since the staggering concept is not easily extended to unstructured grids, artificial staggering has been used to prevent spurious pressure field. Artificial staggering can be done by calculating pressure at fewer grid points than velocity as in unequal-order method by Baliga [45]. Although this solves the checkerboard pressure problem, the accuracy of the solution is significantly decreased when there are regions of high pressure gradient in the flow because the pressure is calculated at a much coarser grid. To improve this, Prakash [46, 47] developed an equal-order method where the pressure and velocity are computed at each grid point. A different kind of artificial staggering was used by using an artificial velocity in place of the nodal velocity. The equal-order velocity and pressure formulation is the basis of the current research.

1.1.1 Time Integration Methods

Time advancement methods can be divided into three categories: explicit, implicit and semi-implicit. Most early simulation methods use the first-order accurate forward Euler method for time advancement. However, since it is an explicit method, it has a limitation in the size of the time step due to stability constraints. The first implicit time advancement method was the backward Euler method. It is unconditionally stable, but it is only first-order accurate. Higher order implicit methods were later developed such as mid-point rule and Crank-Nicolson method [48]. The semi-implicit methods are a combination of implicit and explicit algorithms. Included in this category is the predictor-corrector method by MacCormack [49]. Unfortunately, the semi-implicit methods also have a time step size restriction.

For higher order accuracy in time, Runge-Kutta method provides an alternative. The Runge-Kutta family of time integration methods can be divided into explicit and implicit methods. The classical Runge-Kutta scheme was explicit and was first developed around 1900 by Runge and his successor Kutta [50, 51]. The explicit scheme has been mostly used in compressible flow algorithms [52, 53, 54, 55]. Watterson [56] developed a cell vertex procedure for compressible flow with the explicit time marching Runge-Kutta method. The implicit Runge-Kutta method was first proposed by Kuntzmann and Butcher [57]. The implicit method does not suffer from time step size limitation; however, its derivation is difficult and it requires extensive calculations which makes it computationally inefficient.

Despite the fact that Runge-Kutta methods are more efficient and stable compared to higher order methods such as the multi-step and multi-point methods [58, 59], the application of Runge-Kutta scheme for incompressible flow solvers, in particular, its usage for SIMPLE family of algorithms has not been popular. Previous work includes the implementation of a novel explicit four-stage Runge-Kutta scheme for a 2-D Cartesian flow solver by Purohit [60]. The Runge-Kutta time integration was used in conjunction with SIMPLER algorithm without the use of pressure correction equation. The current work is an extension of the same concept to triangular unstructured grids. Recently, a method has been developed, known as SIMPLE DIRK method, in which the second-order implicit Runge-Kutta scheme has been applied to SIMPLE algorithm for solving incompressible flows [61]. However, their work uses the pressure correction equation for updating the velocity.

1.2 Current Work

The present work is based on the original SIMPLER algorithm developed by Patankar [23]. The equal-order method developed by Prakash and implemented by Maresca [62] for solving steady, two-dimensional incompressible flow using triangular unstructured grid has been extended to unsteady flow. First, the general Fully-Implicit and Crank-Nicolson time integration scheme is implemented. In addition, the explicit four-stage Runge-Kutta scheme was implemented as another option of time integration method. In particular, the Runge-Kutta scheme is used to update the velocity components from the momentum conservation equation without a pressure correction equation.

CHAPTER 2. THEORETICAL FORMULATION

The scope of the current work is limited to laminar, incompressible flow. The governing equations for the flow is given in this chapter. Spatial integration and discretization will be performed assuming steady flow for mathematical simplicity. The formulation in this chapter follows the work of Maresca [62]. The time-accurate formulation will be discussed in Chapter 3.

2.1 Governing Equations

The governing equations for fluid motion are the Navier-Stokes equations. As the flow is assumed to be a Newtonian incompressible flow, the mass and momentum conservation equations are sufficient to model the flow.

2.1.1 Conservation of Mass

The mass conservation equation, or commonly referred to as the continuity equation for a general fluid flow, in divergence form, can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{V}\right) = 0 \tag{2.1}$$

For an incompressible fluid, density ρ is constant, and the equation reduces to:

$$\nabla \cdot \left(\rho \vec{V}\right) = 0 \tag{2.2}$$

It can be expressed in the 2-D Cartesian coordinate system as:

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \tag{2.3}$$

where u and v are the velocity coordinates in the x and y directions, respectively.

2.1.2 Conservation of Momentum

The momentum conservation equation is derived using the Newton's second law applied to a fluid passing through an infinitesimal control volume. It is written in divergence form as:

$$\frac{\partial}{\partial t}\left(\rho\vec{V}\right)+\nabla\cdot\left(\rho\vec{V}\vec{V}\right)=\rho\vec{f}+\nabla\cdot\Pi_{ij} \tag{2.4}$$

where ρ is the fluid density, $\vec{V}$ is the velocity vector, and $\rho\vec{f}$ and $\nabla\cdot\Pi_{ij}$ are the body forces and the surface forces per unit volume, respectively. The surface forces are due to the normal stresses and shear stresses which are represented by the components of the stress tensor Π_{ij}.

For a Newtonian fluid, the shear stress on a particular fluid element is linearly proportional to the rate of deformation. The stress tensor Π_{ij} in tensor notation is given by:

$$\Pi_{ij}=-p\delta_{ij}+\tau_{ij} \tag{2.5}$$

where δ_{ij} is the Kronecker delta function ($\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ if $i\neq j$), and τ_{ij} represents the viscous stress tensor and is given by:

$$\tau_{ij}=\mu\left[\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right] \qquad i,j,k=1,2 \tag{2.6}$$

For incompressible flow, Equation 2.4 will reduce to:

$$\frac{\partial}{\partial t}\left(\rho\vec{V}\right)+\nabla\cdot\rho\vec{V}\vec{V}=-\nabla p+\nabla\cdot\left(\mu\nabla\vec{V}\right)+\vec{S} \tag{2.7}$$

where $\vec{S}$ is the source term due to any external forces.

Expanding Equation 2.7 in 2-D Cartesian coordinates, we obtain:

$$\frac{\partial}{\partial t}(\rho u)+\frac{\partial}{\partial x}(\rho uu)+\frac{\partial}{\partial y}(\rho vu) \quad = \quad -\frac{\partial p}{\partial x}+S^u+\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right) \tag{2.8}$$

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho vv) = -\frac{\partial p}{\partial y} + S^v + \frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right) \tag{2.9}$$

where S^u and S^v are the source terms in the u and v momentum equations, respectively.

Combining like terms and defining the components of total flux vector in the u-momentum equation ($\vec{J^u}$) as:

$$\begin{aligned} J_x^u &= \rho uu - \mu\frac{\partial u}{\partial x} \\ J_y^u &= \rho vu - \mu\frac{\partial u}{\partial y} \end{aligned} \tag{2.10}$$

we can simplify Equation 2.8 to be:

$$\frac{\partial J_x^u}{\partial x} + \frac{\partial J_y^u}{\partial y} = -\frac{\partial p}{\partial x} + S^u \tag{2.11}$$

Similarly for v-momentum equation, the total flux vector ($\vec{J^v}$) has the following components:

$$\begin{aligned} J_x^v &= \rho uv - \mu\frac{\partial v}{\partial x} \\ J_y^v &= \rho vv - \mu\frac{\partial v}{\partial y} \end{aligned} \tag{2.12}$$

and Equation 2.9 becomes:

$$\frac{\partial J_x^v}{\partial x} + \frac{\partial J_y^v}{\partial y} = -\frac{\partial p}{\partial y} + S^v \tag{2.13}$$

2.2 Spatial Discretization

For a finite volume method, the computational domain is subdivided into control volumes, and the conservations laws are evaluated in integral form on each control volume. This research uses a vertex centered method with triangular elements. First, the domain is subdivided into triangular cells and then median dual control volumes are constructed by making a straight line from the center of the triangles to the midpoint of its faces. This results in non-overlapping control volumes for the vertices where Eulerian field variables are stored. The node-centered procedure developed by Prakash [46] and Baliga [63] is used in this work. A typical control volume for vertex P is depicted in Figure 2.1, whose neighboring points are A, B, C, D, E, F and G. The control volume is the shaded region and the dashed lines represent its boundary.

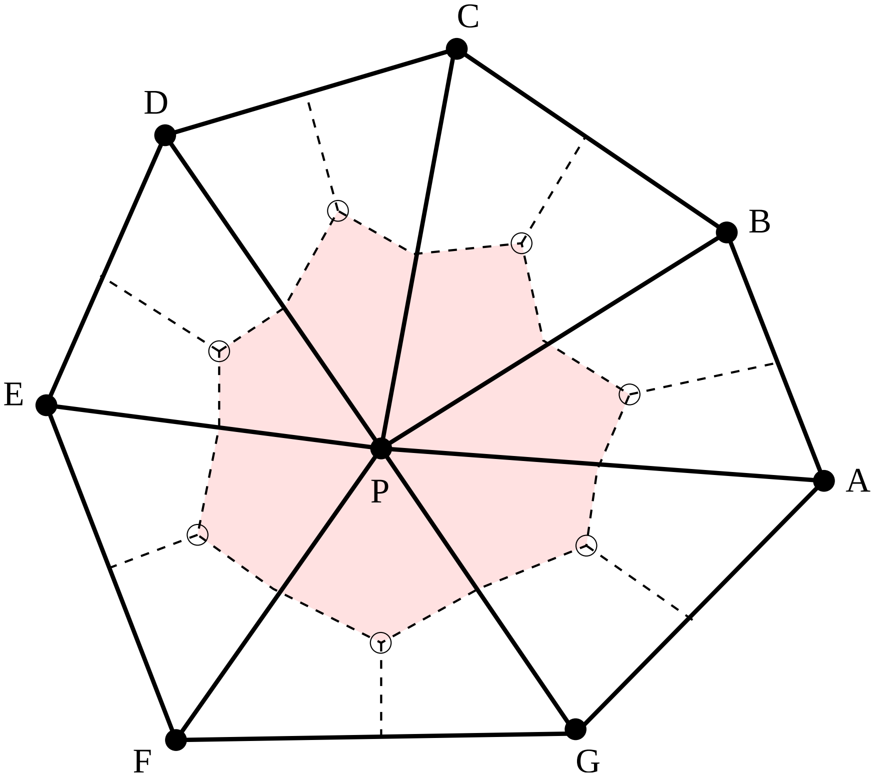


Figure 2.1 Median dual control volume for point P.

2.3 Variable Interpolation Function and Flux Calculation

For the flux calculation at the control volume faces, a profile assumption concerning the variables inside the triangular elements is required. In this research, the following assumptions are made:

- The density (ρ) and viscosity (μ) are constant for a triangular element. This ensures the continuity of flux of any general variable ϕ at the control volume surface.

- The momentum source terms for the u-momentum equation (S^u) and the v-momentum equation (S^v) are constant over a triangle.

- The velocity ($\vec{V}$) varies linearly within a triangle. This assumption will become particularly important when developing the shape function for the general variable ϕ.
- The pressure varies linearly in the x and y-directions. This means that the pressure gradients are constant within a triangular element.

For flux calculation, interpolations of the general variable ϕ or its gradient are required within a triangular element to obtain the values at the control volume faces. The choice of the interpolation function has a significant influence on the accuracy of the method. In addition, it must be able to correctly model the physics of the flow with reasonable computational effort and time. For a 1-D convection-diffusion problem without any source terms, the exact solution is an exponential function. However, since computation of an exponential function is rather expensive, the power law scheme is often used as an approximation to the exponential function [23].

For structured grids where the control volume faces are aligned with the direction of the coordinate system, the implementation of the scheme is straightforward. The drawback is that the numerical accuracy is greatly reduced for higher Reynolds number flows for the case when the velocity vector is not in the direction of the coordinate system. The difficulty with unstructured grids is that the control volume faces may be oriented in random directions, in general not aligned with the velocity. To handle this, Baliga proposed to use a local coordinate system for each triangular element, where one of the coordinate direction is aligned with the average velocity vector inside the triangle.

A schematic of a triangular element is depicted in Figure 2.2 with its vertices denoted as 1, 2 and 3. The global coordinate system is defined by x and y axes. The velocity vector is denoted by $\vec{V}$. The velocity components in these two directions are u and v, respectively. A local coordinate system of X and Y axes is defined such that its origin is located at the triangle's centroid and its X axis coordinate orientation is along the direction of the average velocity vector, $\vec{U}_{avg}$.

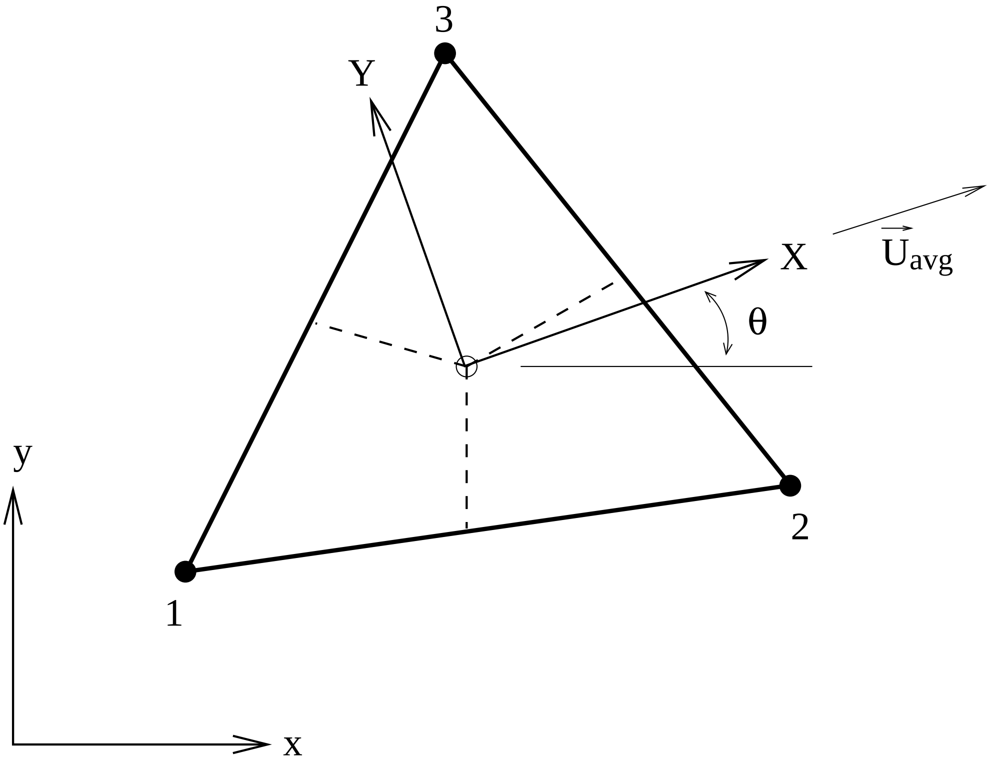


Figure 2.2 Global and local coordinate system for a triangular element.

The average velocity vector at the centroid is given by:

$$U_{avg} = \sqrt{u_{avg}^2 + v_{avg}^2} \tag{2.14}$$

where:

$$u_{avg} = \frac{1}{3}(u_1 + u_2 + u_3) \tag{2.15}$$

$$v_{avg} = \frac{1}{3}(v_1 + v_2 + v_3) \tag{2.16}$$

The angle between the average velocity vector and the global coordinate system x axis can be obtained by:

$$\theta = \cos^{-1}\left(\frac{u_{avg}}{U_{avg}}\right) \tag{2.17}$$

2.3.1 Interpolation Function for General Variable ϕ

Baliga [63] developed an interpolation function for the triangle unstructured mesh which incorporated the exact solution to the 1-D convection-diffusion problem but accounted for the two-dimensionality of the flow. This shape function is exponential in the X-direction and linear in the Y-direction. For a general variable ϕ, the shape function becomes:

$$\phi = AZ + BY + C \tag{2.18}$$

where A, B and C are constants to be determined from the values of ϕ at vertices of the triangle, Z is an exponential function in the local X direction and is defined as:

$$Z = \frac{\mu}{\rho U_{avg}} \left[exp\left(\frac{Pe_{\Delta}\left(X - X_{max}\right)}{X_{max} - X_{min}} \right) - 1 \right] \tag{2.19}$$

The exponential function is computationally expensive, and a “power-law” scheme for unstructured triangular mesh has been developed by Baliga [63] similar to the “power-law” scheme discussed by Patankar for structured grids [23]. The Z function defined in the power law scheme is:

$$Z = \frac{X - X_{max}}{Pe + \llbracket 0, \left(1 - 0.1\left|Pe\right|\right)^5 \rrbracket} \tag{2.20}$$

In the above equations, Pe is the element Peclet number, which can be calculated as:

$$Pe_{\Delta} = \frac{\rho U_{avg} \Delta X}{\mu} \tag{2.21}$$

The characteristic length, ΔX, is:

$$\Delta X = X_{max} - X_{min} \tag{2.22}$$

where:

$$X_{max} \quad = \quad max(X_1, X_2, X_3) \tag{2.23}$$

$$X_{min} = min(X_1, X_2, X_3) \tag{2.24}$$

The constants A, B, and C of Equation 2.18 are calculated from the boundary requirements of $\phi = \phi_1$, $\phi = \phi_2$, and $\phi = \phi_3$ at their respective vertices. Details of the derivation can be found in Appendix A. The constants are a function of the variable at the vertices of the triangle:

$$A = L_i\phi_i \tag{2.25}$$

$$B = M_i\phi_i \tag{2.26}$$

$$C = N_i\phi_i \tag{2.27}$$

where $a_ib_i = \sum_{i=1}^{3} a_ib_i$ as used in tensor notation, and the index i corresponds to indices 1, 2, and 3. The final form of the interpolation function is then given as:

$$\phi = (L_iZ + M_iY + N_i)\,\phi_i \tag{2.28}$$

2.3.2 Momentum Flux Computation

The flux vector $\vec{J}$ can be written in the local coordinate system as:

$$\vec{J} = J_X\hat{e}_X + J_Y\hat{e}_Y \tag{2.29}$$

where $\hat{e}_X$ and $\hat{e}_Y$ are the unit vector in the local coordinate system, and the components are as follows:

$$J_X = \rho U\phi - \mu\frac{\partial\phi}{\partial X} \tag{2.30}$$

$$J_Y = \rho V\phi - \mu\frac{\partial\phi}{\partial Y} \tag{2.31}$$

Now that we have the spatial profile of the general variable ϕ (Equation 2.28), we can find its derivatives, and substitute them in Equations 2.30 and 2.31. After substituting and rearranging the variables, we obtain:

$$J_X = (\rho f_i - \mu L_i)\,\phi_i \tag{2.32}$$

$$J_Y = (\rho g_i - \mu M_i)\,\phi_i \tag{2.33}$$

where:

$$f_i = (U - U_{avg})\,L_i Z + U\,(M_i Y + N_i) \tag{2.34}$$

$$g_i = V\,(L_i Z + M_i Y + N_i) \tag{2.35}$$

2.3.3 Interpolation Function for Pressure

Pressure is assumed to vary linearly within a triangular element in x and y-direction of the global coordinate system. The equation for pressure can be written as:

$$p = -\,(\alpha x + \beta y + \gamma) \tag{2.36}$$

where the coefficients are determined by forcing $p = p_1$, $p = p_2$, and $p = p_3$ at their respective grid points (Figure 2.3). If the pressure field within a triangle is a linear function, the derivatives will be constants. Thus, the derivatives with respect to x and y are given as:

$$\begin{aligned} -\frac{\partial p}{\partial x} &= \alpha \\ &= \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3 \end{aligned} \tag{2.37}$$

$$\begin{aligned} -\frac{\partial p}{\partial y} &= \beta \\ &= \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3 \end{aligned} \tag{2.38}$$

where the coefficients $\bar{L}'s$ and $\bar{M}'s$ are derived in Appendix A.

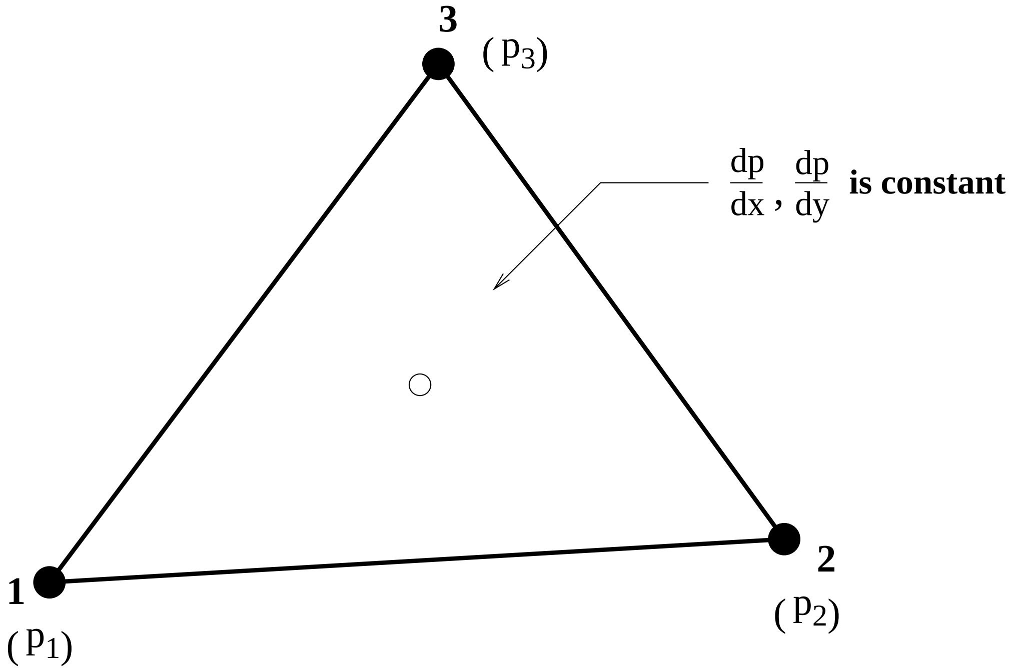


Figure 2.3   Pressure variation within an element.

2.4 Integration and Discretization of the Momentum Equations

The u and v momentum equations (derived earlier in Equations 2.11 and 2.13) are:

$$\frac{\partial J_x^u}{\partial x} + \frac{\partial J_y^u}{\partial y} = -\frac{\partial p}{\partial x} + S^u \tag{2.39}$$

$$\frac{\partial J_x^v}{\partial x} + \frac{\partial J_y^v}{\partial y} = -\frac{\partial p}{\partial y} + S^v \tag{2.40}$$

For the u momentum equation, the left hand side (LHS) of the equation can be written as:

$$\frac{\partial J_x^u}{\partial x} + \frac{\partial J_y^u}{\partial y} = \nabla \cdot \vec{J}^u \tag{2.41}$$

We can rewrite the LHS in the local coordinate system as:

$$\nabla \cdot \vec{J}^u = \frac{\partial J_X^u}{\partial X} + \frac{\partial J_Y^u}{\partial Y} \tag{2.42}$$

where the divergence operation in the local coordinate system is given in Appendix B.

Now the momentum conservation equations becomes:

$$\int\int\left(\frac{\partial J_X^u}{\partial X}+\frac{\partial J_Y^u}{\partial Y}\right)dA=\int\int\left(S^u-\frac{\partial p}{\partial x}\right)dA \tag{2.43}$$

$$\int\int\left(\frac{\partial J_X^v}{\partial X}+\frac{\partial J_Y^v}{\partial Y}\right)dA=\int\int\left(S^v-\frac{\partial p}{\partial y}\right)dA \tag{2.44}$$

Note that the LHS of the equations are expressed in the local coordinate system, aligned with the local flow which reduces false diffusion. However, the terms on the right hand side (RHS) of the equation which are the pressure gradient and source term, are in the global coordinate system.

The Gauss' theorem for a vector $\vec{B}$ is defined as:

$$\int\limits_{CV}\nabla\cdot\vec{B}\;dV=\oint\limits_{CS}\vec{B}\cdot d\vec{A}=\oint\limits_{CS}\vec{B}\cdot\hat{n}dA \tag{2.45}$$

where $\hat{n}$ is the normal vector pointing out of the enclosed area dA.

Using Gauss' theorem the equation can be transformed from surface integral to a closed line integral:

$$\oint\left(\vec{J^u}\cdot\hat{n}\right)dl=\int\int\left(S^u-\frac{\partial p}{\partial x}\right)dA \tag{2.46}$$

where the $\hat{n}$ is normal to the control volume face $r{-}s{-}t$ pointing away from point 1 (Figure 2.4), and is computed by:

$$\hat{n}=\left(\frac{Y_t-Y_r}{dl}\right)\hat{e}_X-\left(\frac{X_t-X_r}{dl}\right)\hat{e}_Y \tag{2.47}$$

2.4.1 Integration of the LHS

The left hand side (LHS) of Equation 2.46 requires the integration of the flux terms across the face $r-s-t$ with length dl:

$$\textbf{LHS}=\oint\left(\vec{J^u}\cdot\hat{n}\right)dl \tag{2.48}$$

If the vectors in LHS of the equation are composed in the local coordinate system, we can write the flux vector as:

$$\vec{J}^u = J_X^u \hat{e}_X + J_Y^u \hat{e}_Y \tag{2.49}$$

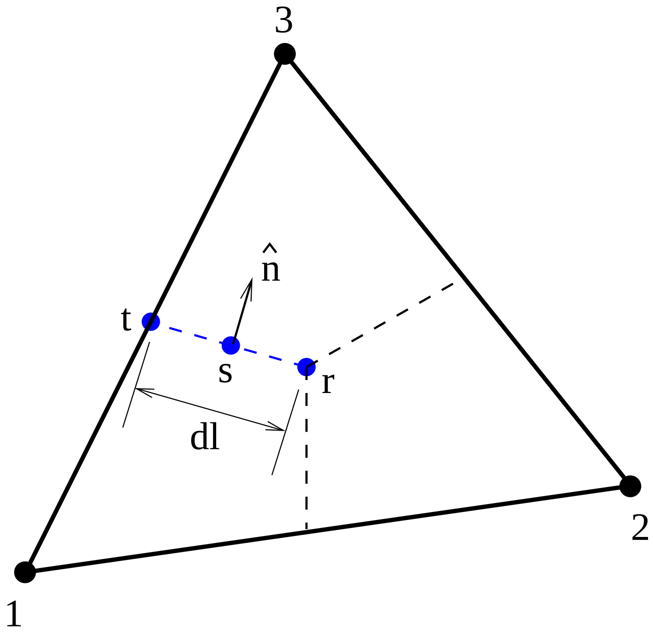


Figure 2.4 Flux integration.

Using Simpson's quadrature rule, the integral of Equation 2.48 is calculated based on information from three locations – r, s, and t:

$$\mathbf{LHS} = \frac{dl}{6}\left(\left[\vec{J}^u \cdot \hat{n}\right]_r + 4\left[\vec{J}^u \cdot \hat{n}\right]_s + \left[\vec{J}^u \cdot \hat{n}\right]_t\right) \tag{2.50}$$

Expanding the above equation results in an expression for the fluxes at the three points:

$$\mathbf{LHS} = \frac{dl}{6}\left(\left[J_{Xr}^u + 4J_{Xs}^u + J_{Xt}^u\right]\left(\frac{Y_t - Y_r}{dl}\right)\right.$$
$$\left. - \left[J_{Yr}^u + 4J_{Ys}^u + J_{Yt}^u\right]\left(\frac{X_t - X_r}{dl}\right)\right) \tag{2.51}$$

However, the fluxes at the interface are functions of values at grid points 1, 2, and 3. Taking guidance from Equations 2.30 and 2.31 and rearranging Equation 2.51, the flux of u across the interface is obtained as:

$$\mathbf{LHS} = a_1 u_1 + a_2 u_2 + a_3 u_3 = a_i u_i \tag{2.52}$$

For the grid point P, it follows naturally that the total flux leaving the control volume surrounding the node can be calculated collectively by visiting each edge and applying the same procedure discussed above. Looking at Figure 2.5, edge by edge, one is able to collectively compute the flux across the control volume for faces on both the left and right sides of the edge. For node P, the **LHS** of the equation is:

$$\mathbf{LHS} = a_P^u u_P - \sum a_{nb}^u u_{nb} \tag{2.53}$$

where nb represents the terms from the neighboring points (2's and 3's) of P and the a_P is composed of the sum of the a_1 coefficients from all the triangles surrounding point P such as shown in Figure 2.5.

2.4.2 Integration of the RHS

The right hand side (RHS) of the u-momentum equation (Equation 2.43) is:

$$\mathbf{RHS} = \int\int \left(S^u - \frac{\partial p}{\partial x} \right) dA \tag{2.54}$$

As previously stated, the source term S^u and pressure gradients are assumed to be constant for any particular triangle or element (Figure 2.6). Substituting pressure gradient from Equation 2.38 and assuming S^u is constant, integration of Equation 2.54 yields:

$$\mathbf{RHS} = A\left(S^u + \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3 \right) \tag{2.55}$$

where A is the area of the control volume shown in Figure 2.6.

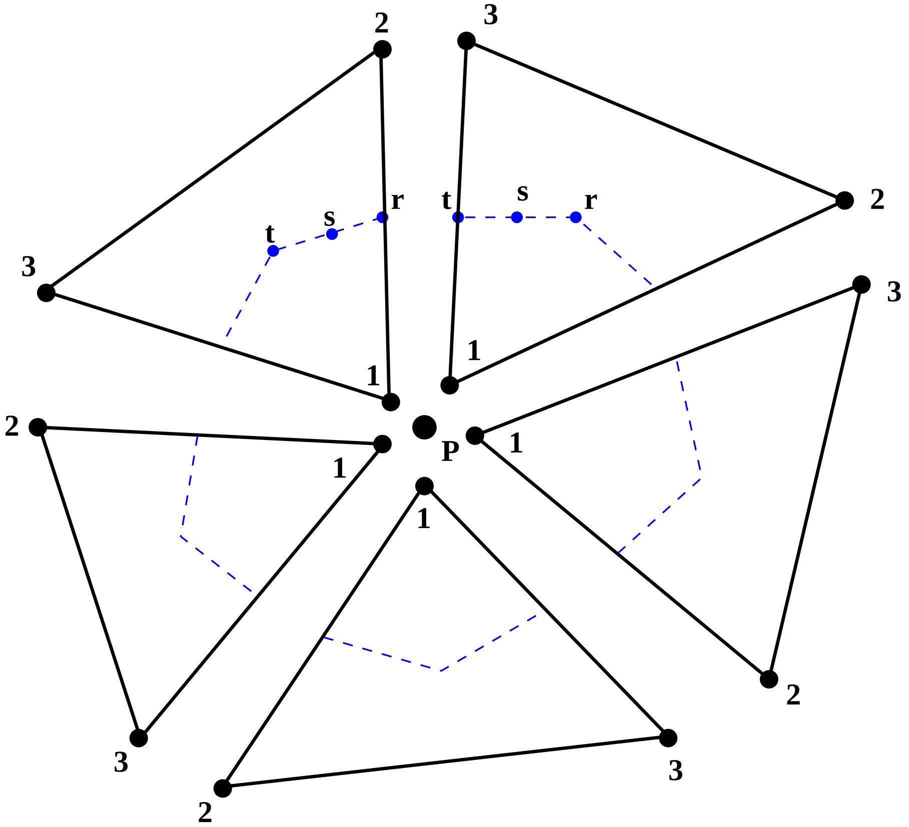


Figure 2.5 Total flux integration for grid point P.

2.4.3 The Total Discretized Equation

Combining Equations 2.53 and 2.55, the total discretized U–momentum equation for node P is obtained in the form of:

$$a_P^u u_P = \sum a_{nb}^u u_{nb} + \sum_{m=1}^{M} \left(A \left[S^u + \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3 \right] \right)_m \tag{2.56}$$

where M is the total number of triangles surrounding grid point P. Likewise, the V–momentum equation can be written as follows:

$$a_P^v v_P = \sum a_{nb}^v v_{nb} + \sum_{m=1}^{M} \left(A \left[S^v + \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3 \right] \right)_m \tag{2.57}$$

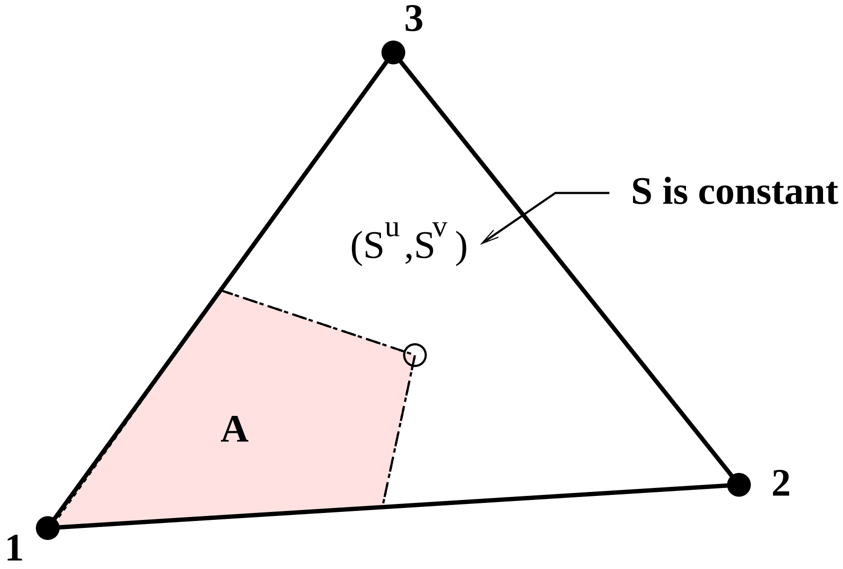


Figure 2.6 Integration of the source and pressure terms.

2.4.4 Boundary Condition

For boundary points such as shown in Figure 2.7, the discretized equations are slightly modified to take into account the known quantity from the boundary edge. Two boundary conditions are possible:

- given ϕ (u or v),
- given flux F^ϕ

To handle these boundary conditions, the discretization equation for the boundary point becomes:

$$a_P^u \phi_P = \sum a_{nb}\phi_{nb} + \sum_{m=1}^{M} \left(A_P \ S_{total}^{\phi} \right)_m - F_P^{\phi} + \dot{m}_P \phi_P \tag{2.58}$$

where F_P^ϕ is the total flux of ϕ, $\dot{m}_P$ is the total mass flux leaving the domain through section $i - P - a$ (Figure 2.7), and M is the total number of triangles surrounding point P. When the value of ϕ is known at point P (given ϕ_P), then Equation 2.58 is used to determine F_P^ϕ:

$$F_P^{\phi} = \sum a_{nb}\phi_{nb} - a_P^u \phi_P + \sum_{m=1}^{M} \left(A_P \ S_{total}^{\phi} \right)_m + \dot{m}_P \phi_P \tag{2.59}$$

If the flux of ϕ is given (known F_P^{ϕ}), then ϕ_P is obtained by solving Equation 2.58 and treating it as a regular point.

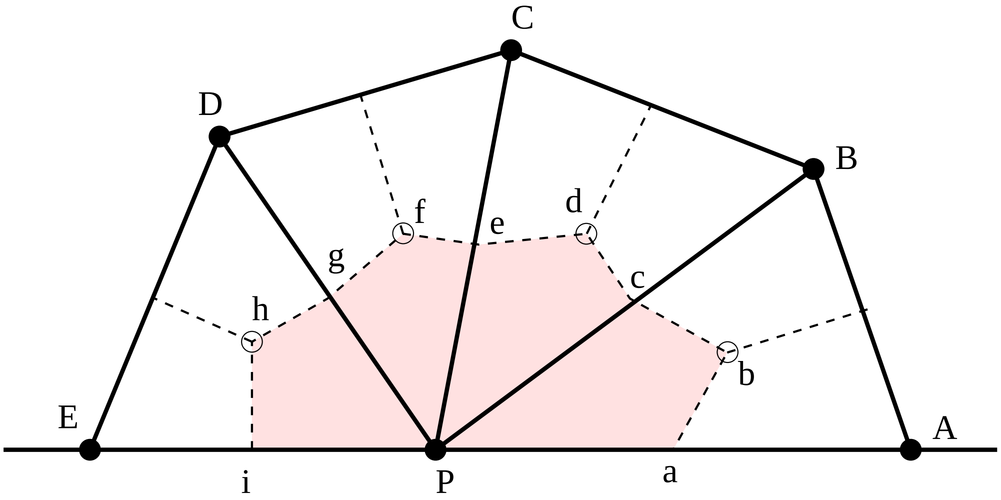


Figure 2.7 Boundary point P.

2.5 Equal-Order Velocity-Pressure Interpolation Method

The equal-order method developed by Prakash and Patankar [47] is implemented. The difficulties of an equal order method for incompressible flow is that there is no explicit equation of pressure, and only gradients of pressure appear in the momentum equation. The pressure is obtained indirectly from the continuity equation. When the pressure and velocity have the same interpolation function, spurious pressure distribution may occur. The basic idea behind the equal-order method is that the velocity field used to solve the pressure from the continuity equation should be dependent on the pressure difference between adjacent grid points. Thus, the solution of continuity equation does not allow a checkerboard pressure field.

2.5.1 Definition of pseudo-velocity ($\hat{u}$, $\hat{v}$) and source term coefficient (d^u, d^v)

The pseudo velocities, $\hat{u}$ and $\hat{v}$, and the coefficients, d^u and d^v, are central to the definition of the artificial velocity field. From Equations 2.56 and 2.57, the u and v velocity can be solved from:

$$u_P = \frac{\sum a^u_{nb} u_{nb}}{a^u_P} + \frac{A_P}{a^u_P} \left(S^u + L_1 p_1 + L_2 p_2 + L_3 p_3 \right) \tag{2.60}$$

$$v_P = \frac{\sum a^v_{nb} v_{nb}}{a^v_P} + \frac{A_P}{a^v_P} \left(S^v + M_1 p_1 + M_2 p_2 + M_3 p_3 \right) \tag{2.61}$$

If the pseudo velocities are defined by:

$$\hat{u}_P = \frac{\sum a^u_{nb} u_{nb}}{a^u_P} \tag{2.62}$$

$$\hat{v}_P = \frac{\sum a^v_{nb} v_{nb}}{a^v_P} \tag{2.63}$$

where, nb is a summation of grid points immediately surrounding grid point P. We can define the coefficients of the source term, d^u and d^v, for a node P as:

$$d^u_P = \frac{A_P}{a^u_P} \tag{2.64}$$

$$d^v_P = \frac{A_P}{a^v_P} \tag{2.65}$$

where A_P is the area of the control volume surrounding grid point P. It is assumed that $\hat{u}$, $\hat{v}$, d^u, and d^v vary linearly within the triangular element.

2.5.2 Definition of Artificial Velocity $\vec{\tilde{u}}$

The artificial velocity field used in the equal-order method is denoted as $\vec{\tilde{u}}$. It is derived from the solution of the discretized momentum equations and is defined element by element. For a typical element shown in Figure 2.8, $\vec{\tilde{u}}$ is defined as:

$$\vec{\tilde{u}} = \tilde{u}\hat{i} + \tilde{v}\hat{j} \tag{2.66}$$

where the components of the velocity is defined as:

$$\tilde{u} = \hat{u} + d^u \left[S^u - \frac{\partial p}{\partial x} \right]_e \tag{2.67}$$

$$\tilde{v} = \hat{v} + d^v \left[S^v - \frac{\partial p}{\partial y} \right]_e \tag{2.68}$$

and the pseudo-velocities $(\hat{u}, \hat{v})$ and the source term coefficients (d^u, d^v) are given in Equations 2.62 through 2.65. The subscript $_e$ indicates that the pressure gradient and source term are taken only from the given element. This is different from the discretized momentum equation (Equations 2.60 and 2.61) where the pressure gradient and source term are composed of the contributions from all the surrounding cells.

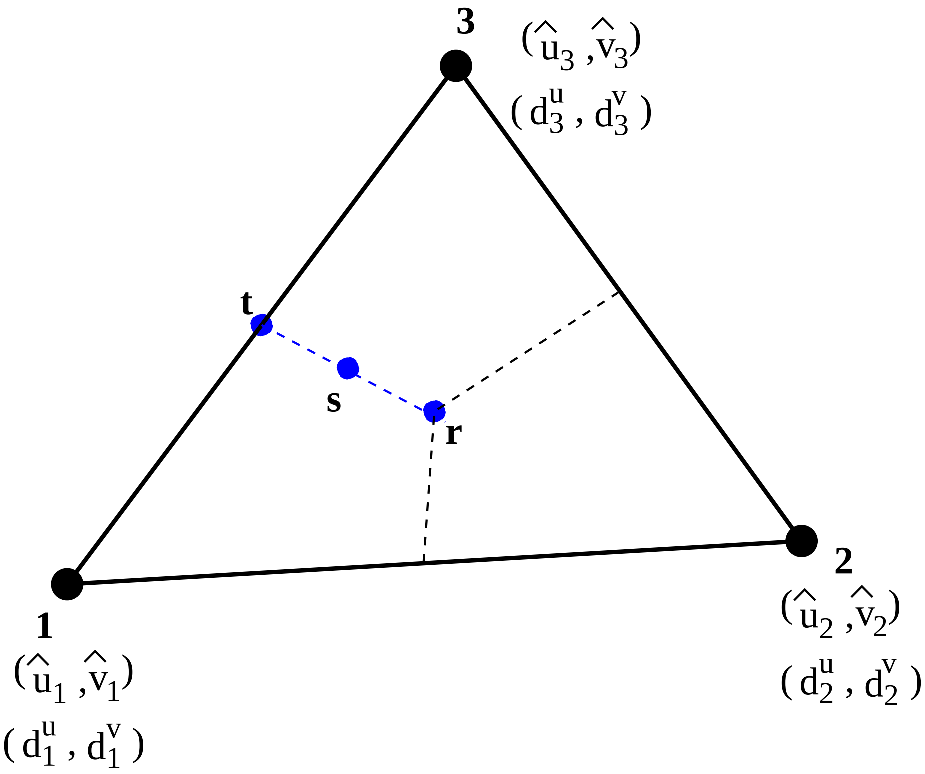


Figure 2.8 Artificial velocity field on a triangular element.

Unlike the nodal velocity which do not sense any difference between a uniform and a checkerboard pressure field, the artificial velocity field is dependent on and is driven by the pressure difference between adjacent grid points. Therefore, a checkerboard pressure field is not an acceptable solution to the continuity equation. Using the above definitions, the $\vec{\tilde{u}}$ field can be expressed at the control volume faces in a straightforward manner, by linearly interpolating from grid points at the vertices of the triangles. The derivation of the artificial velocity for the control volume faces will be given later in this chapter.

2.5.3 Equal-Order Formulation for Solving the Governing Equations

The discretization of the governing equation has been completed in the previous sections. In the equal-order method, the continuity equation was applied to the artificial velocity $\vec{\tilde{u}}$ field and not the nodal velocity field. Since the mass conservation equation is solved using the artificial velocity field to obtain pressure, the coefficients of the momentum equations must be calculated using a mass conserving velocity field for the method to be conservative. Therefore, the artificial velocity $\vec{\tilde{u}}$ field should be used in place of the nodal velocities when calculating the coefficient a_i in Equation 2.52. When finding the shape function and local coordinate system orientation, the artificial velocity should be used as follows:

$$\tilde{U}_{avg} = \sqrt{\tilde{u}_{avg}^2 + \tilde{v}_{avg}^2} \tag{2.69}$$

where:

$$\tilde{u}_{avg} = \frac{1}{3}\left(\tilde{u}_1 + \tilde{u}_2 + \tilde{u}_3\right) \tag{2.70}$$

$$\tilde{v}_{avg} = \frac{1}{3}\left(\tilde{v}_1 + \tilde{v}_2 + \tilde{v}_3\right) \tag{2.71}$$

Similarly, $\vec{\tilde{u}}$ should be used to compute the flux at the control volume faces in Equation 2.35:

$$f_i = \left(\tilde{U} - \tilde{U}_{avg}\right) L_i Z + \tilde{U}\left(M_i Y + N_i\right) \tag{2.72}$$

$$g_i = \tilde{V}\left(L_i Z + M_i Y + N_i\right) \tag{2.73}$$

where $\tilde{U}$ and $\tilde{V}$ are components of the $\vec{\tilde{u}}$ field in the local coordinate system.

2.5.4 Interpolation of the Artificial Velocity $\vec{\tilde{u}}$ at Control Volume Faces

In Figure 2.8, the velocity component $\tilde{u}$ is defined at the triangle vertices 1, 2, and 3 as:

$$\tilde{u}_1 = \hat{u}_1 + d_1^u \left[S^u + (\bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3)\right]_e \tag{2.74}$$

$$\tilde{u}_2 = \hat{u}_2 + d_2^u \left[S^u + (\bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3)\right]_e \tag{2.75}$$

$$\tilde{u}_3 = \hat{u}_3 + d_3^u \left[S^u + (\bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3)\right]_e \tag{2.76}$$

Similarly for the component $\tilde{v}$, it is defined as:

$$\tilde{v}_1 = \hat{v}_1 + d_1^v \left[S^v + (\bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3)\right]_e \tag{2.77}$$

$$\tilde{v}_2 = \hat{v}_2 + d_2^v \left[S^v + (\bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3)\right]_e \tag{2.78}$$

$$\tilde{v}_3 = \hat{v}_3 + d_3^v \left[S^v + (\bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3)\right]_e \tag{2.79}$$

As stated earlier, assumption has been made that the pseudo velocities ($\hat{u}$ and $\hat{v}$) and the coefficients of the source term (d^u and d^v) have a linear profile within a triangular element, while the source terms S^u, S^v, and the pressure gradient are constants for each element or triangle. This means that the artificial velocity also varies linearly inside a triangle. In other words, the artificial velocity components, $\tilde{u}$ and $\tilde{v}$, can be computed at the control volume face $r - s - t$ by linearly interpolating $\hat{u}$, $\hat{v}$, d^u, and d^v, while holding the pressure gradient and source terms constant.

Artificial Velocity at Point r

Referring to Figure 2.8, we can obtain the artificial velocity component in X-direction at the centroid r from:

$$\tilde{u}_r = \hat{u}_r + d_r^u \left[S^u + \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3\right] \tag{2.80}$$

where the pseudo-velocity and source term coefficient are obtained by:

$$\hat{u}_r = \frac{1}{3}\left(\hat{u}_1 + \hat{u}_2 + \hat{u}_3\right) \tag{2.81}$$

$$d_r^u = \frac{1}{3}\left(d_1^u + d_2^u + d_3^u\right) \tag{2.82}$$

Similarly,

$$\tilde{v}_r = \hat{v}_r + d_r^v\left[S^v + \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3\right] \tag{2.83}$$

where:

$$\hat{v}_r = \frac{1}{3}\left(\hat{v}_1 + \hat{v}_2 + \hat{v}_3\right) \tag{2.84}$$

$$d_r^v = \frac{1}{3}\left(d_1^v + d_2^v + d_3^v\right) \tag{2.85}$$

Artificial Velocity at Point t

Since the mid point of the triangle element edge (point t) is located halfway between point 1 and 3, the pseudo velocity and source term coefficient can be obtained by averaging the values at grid points 1 and 3:

$$\hat{u}_t = \frac{1}{2}\left(\hat{u}_1 + \hat{u}_3\right) \tag{2.86}$$

$$d_t^u = \frac{1}{2}\left(d_1^u + d_3^u\right) \tag{2.87}$$

$$\hat{v}_t = \frac{1}{2}\left(\hat{v}_1 + \hat{v}_3\right) \tag{2.88}$$

$$d_t^v = \frac{1}{2}\left(d_1^v + d_3^v\right) \tag{2.89}$$

The artificial velocity at the point t can be simplified to:

$$\tilde{u}_t = \hat{u}_t + d_t^u\left[S^u + \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3\right] \tag{2.90}$$

$$\tilde{v}_t = \hat{v}_t + d_t^v\left[S^v + \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3\right] \tag{2.91}$$

Artificial Velocity at Point s

The velocity at point s is computed by linearly interpolating between points r and t. Since point s is halfway between the two points, Equations 2.80 and 2.90 can be averaged to obtain

$\tilde{u}_s$:

$$\tilde{u}_s = \frac{1}{12}\left(5\hat{u}_1 + 2\hat{u}_2 + 5\hat{u}_3\right) + \frac{1}{12}\left(5d_1^u + 2d_2^u + 5d_3^u\right)\left[S^u + \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3\right] \quad (2.92)$$

Similarly for $\tilde{v}_s$, Equations 2.83 and 2.91 can be averaged to yield:

$$\tilde{v}_s = \frac{1}{12}\left(5\hat{v}_1 + 2\hat{v}_2 + 5\hat{v}_3\right) + \frac{1}{12}\left(5d_1^v + 2d_2^v + 5d_3^v\right)\left[S^v + \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3\right] \quad (2.93)$$

We can cast the formula in terms of pseudo velocity and source term coefficients:

$$\begin{aligned} \tilde{u}_s &= \hat{u}_s + d_s^u\left[S^u + \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3\right] && (2.94) \\ \tilde{v}_s &= \hat{v}_s + d_s^v\left[S^v + \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3\right] && (2.95) \end{aligned}$$

where the pseudo velocity components and source term coefficients are:

$$\begin{aligned} \hat{u}_s &= \frac{1}{12}\left(5\hat{u}_1 + 2\hat{u}_2 + 5\hat{u}_3\right) && (2.96) \\ d_s^u &= \frac{1}{12}\left(5d_1^u + 2d_2^u + 5d_3^u\right) && (2.97) \\ \hat{v}_s &= \frac{1}{12}\left(5\hat{v}_1 + 2\hat{v}_2 + 5\hat{v}_3\right) && (2.98) \\ d_s^v &= \frac{1}{12}\left(5d_1^v + 2d_2^v + 5d_3^v\right) && (2.99) \end{aligned}$$

Since the artificial velocity $\vec{\tilde{u}}$ field is used when solving the continuity equation to obtain the pressure, it can be considered as the mass conserving velocity field. In order to attain a conservative formulation, the coefficients of the momentum equations need to be generated using a conservative velocity field. Thus, the coefficients of the discretized momentum coefficients in Equations 2.56 and 2.57 are calculated using these values instead of the nodal velocity.

2.6 Pressure Equation

The key idea behind the equal-order method is to use the artificial velocity $\vec{\tilde{u}}$ field to compute the pressure from the discretized mass conservation equation:

$$\oint \rho \vec{\tilde{u}} \cdot \hat{n} dl = 0 \tag{2.100}$$

By applying a simple, one point integration to the mass conservation equation for the control volume face $r - s - t$ in Figure 2.8, we obtain:

$$\int_r^t \rho \vec{\tilde{u}} \cdot \hat{n} dl = \rho \left[\tilde{u}_s \left(y_t - y_r\right) - \tilde{v}_s \left(x_t - x_r\right)\right] \tag{2.101}$$

Substituting the velocity components $\tilde{u}_s$ and $\tilde{v}_s$ from Equations 2.94 and 2.95 results in:

$$\begin{aligned} &\rho \left[\tilde{u}_s \left(y_t - y_r\right) - \tilde{v}_s \left(x_t - x_r\right)\right] = \\ &\rho \left[\hat{u}_s + d_s^u \left(S^u + \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3\right)\right] \left(y_t - y_r\right) - \\ &\rho \left[\hat{v}_s + d_s^v \left(S^v + \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3\right)\right] \left(x_t - x_r\right) \end{aligned} \tag{2.102}$$

Collecting the terms which contain pressure and separating the other terms, we can rewrite the equation as:

$$\rho \left[\tilde{u}_s \left(y_t - y_r\right) - \tilde{v}_s \left(x_t - x_r\right)\right] = a_i^p p_i + r^p \tag{2.103}$$

where

$$a_i^p = \rho \left[\left(y_t - y_r\right) d_s^u \bar{L}_i - \left(x_t - x_r\right) d_s^v \bar{M}_i\right] \tag{2.104}$$

and

$$r^p = \rho \left[\left(y_t - y_r\right) \left(\hat{u}_s + d_s^u S^u\right) - \left(x_t - x_r\right) \left(\hat{v}_s + d_s^v S^v\right)\right] \tag{2.105}$$

Note that i represents the vertices 1, 2 and 3 of the triangle in which the face $r - s - t$ is located. If the above process is applied to every face of the control volume boundary, the total discretized equation for pressure becomes:

$$a_P^p \; p_P = \sum a_{nb}^p \; p_{nb} + b^p \tag{2.106}$$

where the subscript nb denotes all the neighbor grid points surrounding control volume for point p, and the pressure source term is:

$$b^p = \sum r^p \tag{2.107}$$

2.6.1 Boundary Condition for Pressure Equation

For a boundary point P, as in Figure 2.7, the discretized pressure equation needs to be modified to include the mass flow leaving the boundary face. This is done by subtracting the mass flux from the right hand side of the Equation 2.106. The discretization equation for pressure for a boundary point can be written as:

$$a_P^p p_P = \sum a_{nb}^p p_{nb} + b^p - \dot{m}_P \tag{2.108}$$

where $\dot{m}_P$ is the mass flow leaving the control volume through the boundary face $i - P - a$ (Figure 2.7).

2.7 Pressure Correction Equation

The momentum and pressure equations discretized so far are adequate to solve for the velocity components and pressure. However, the convergence can be accelerated by deriving a pressure correction equation which will correct the velocities so that they satisfy the mass conservation equation at the end of each iteration. Especially in the equal-order method, since the mass conservation equation is solved using the artificial velocity, the solution of the mass conservation does not directly affect the values of the nodal velocity. Therefore, the convergence of this algorithm is found to be rather slow. To accelerate convergence, Patankar popularized the use of pressure correction equation in SIMPLE and SIMPLER algorithm. A similar concept is adopted here for the equal-order method.

Suppose that the pressure field p^* is used to calculate the velocity components u^* and v^*. Using u^* and v^* to get $\hat{u}^*$ and $\hat{v}^*$, the artificial velocity field $\vec{\tilde{u}}$ is given as:

$$\tilde{u}^* = \hat{u}^* + d^u \left[S^u - \frac{\partial p^*}{\partial x} \right] \tag{2.109}$$

$$\tilde{v}^* = \hat{v}^* + d^v \left[S^v - \frac{\partial p^*}{\partial y} \right] \tag{2.110}$$

The star subscript indicates that the velocity field is a guessed values obtained from solving the momentum equations which does not satisfy the mass conservation (continuity) equation. Suppose we have a correction to pressure (p'):

$$p = p^* + p' \tag{2.111}$$

With this pressure, the artificial velocity will satisfy the mass conservation:

$$\tilde{u} = \hat{u}^* + d^u \left[S^u - \frac{\partial}{\partial x} \left(p^* + p' \right) \right] \tag{2.112}$$

$$\tilde{v} = \hat{v}^* + d^v \left[S^v - \frac{\partial}{\partial y} \left(p^* + p' \right) \right] \tag{2.113}$$

We can write the new velocity as the guessed velocity plus a correction term:

$$\tilde{u} = \tilde{u}^* + \tilde{u}' \tag{2.114}$$

$$\tilde{v} = \tilde{v}^* + \tilde{v}' \tag{2.115}$$

where the velocity correction $\tilde{u}'$ and $\tilde{v}'$ are given by:

$$\tilde{u}' = -d^u \left(\frac{\partial p'}{\partial x} \right) \tag{2.116}$$

$$\tilde{v}' = -d^v \left(\frac{\partial p'}{\partial y} \right) \tag{2.117}$$

If we compare these equations with the ones used in the pressure equation, the difference is only in the source terms. The source terms used in the pressure solution are replaced with:

$$S^u_{p'} = S^u - \frac{\partial p^*}{\partial x} \tag{2.118}$$

$$S^v_{p'} = S^v - \frac{\partial p^*}{\partial y} \tag{2.119}$$

and $\hat{u}$ and $\hat{v}$ are replaced with $\hat{u}^*$ and $\hat{v}^*$ for the p' computation. Therefore, the system of equations for the pressure correction procedure is:

$$a^p_P \; p' = \sum a^p_{nb} \; p'_{nb} + b^{pp} \tag{2.120}$$

For a boundary node where the mass flux is known, the equation becomes:

$$a^p_P \; p' = \sum a^p_{nb} \; p'_{nb} + b^{pp} - \dot{m}_P \tag{2.121}$$

where a^p_P and a^p_{nb} are the same coefficients from the pressure p^* equation. Once p' is obtained, the velocity correction terms $\tilde{u}'$ and $\tilde{v}'$ is calculated using Equations 2.116 and 2.117. Then, the nodal velocities are corrected using the p' field obtained. Prakash [46] developed the nodal velocity correction formulas based on the $\tilde{u}'$ and $\tilde{v}'$ equations. They are as follows:

$$u'_P = -\frac{d^u_P}{3A_P} \sum_{i=1}^{k} A_i \left(\frac{\partial p'}{\partial x} \right)_i \tag{2.122}$$

$$v'_P = -\frac{d^v_P}{3A_P} \sum_{i=1}^{k} A_i \left(\frac{\partial p'}{\partial y} \right)_i \tag{2.123}$$

where d^u_P and d^v_P are the coefficients d^u and d^v at grid point P, and the summation is made over the triangles which contribute to the control volume around P. Finally, u^*_P and v^*_P are corrected using Equations 2.122-2.123:

$$u_P = u^*_P + u'_P \tag{2.124}$$

$$v_P = v_P^* + v_P' \tag{2.125}$$

Note that the boundary velocities are not corrected because it will change the specified boundary condition for the given problem.

CHAPTER 3. NUMERICAL PROCEDURES FOR UNSTEADY SIMULATION

The steady SIMPLER algorithm [23] is the starting point of the algorithms to be developed here for unsteady flows. Three different time integration schemes are explored in this research, namely (a) Fully-Implicit; (b) Crank-Nicolson; and (c) Runge-Kutta. Of these schemes, the first two follow the traditional SIMPLER algorithm for the pressure-velocity coupling wherein a pressure-correction equation is used to correct the velocity components. The current research uses a different method for computing the velocity components. The explicit algorithm developed in this thesis for unstructured meshes, identified henceforth as “RK-SIMPLER”, follows the traditional SIMPLER algorithm for updating the pressure by solving the mass conservation equation, but uses the Runge-Kutta four-stage algorithm for updating the velocity components. Since no approximation is assumed in the momentum equation, the pressure correction used to correct the velocity components is no longer necessary. The Fully Implicit and the Crank-Nicolson schemes also require each time step to be subdivided and the velocities to be corrected at every sub-iteration, while in Runge-Kutta the velocity is updated only once while advancing to the next time level. The details of the current modified SIMPLER algorithm is described in this section.

3.1 Time Integration Method

3.1.1 Fully Implicit and Crank Nicolson Schemes

If ϕ is the variable to be integrated, an assumption is necessary about how it will vary with time from t to $t + \Delta t$. Many assumptions are possible; some of them may be generalized by proposing the following stencil:

$$\int_t^{t+\Delta t} \phi \; dt = \left[\alpha\phi + (1-\alpha)\phi^0\right] \Delta t \tag{3.1}$$

where α is the weighting factor. In particular, $\alpha = 0.5$ leads to the Crank-Nicolson scheme and $\alpha = 1$ to the Fully Implicit scheme.

3.1.2 General Convection-Diffusion Equation

The two-dimensional form of the General Convection-Diffusion equation can be written as:

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} = S \tag{3.2}$$

The term S represents the source term while J_x and J_y are the total (convection plus diffusion) fluxes defined by:

$$J_x \equiv \rho u\phi - \Gamma\frac{\partial\phi}{\partial x} \tag{3.3}$$

$$J_y \equiv \rho v\phi - \Gamma\frac{\partial\phi}{\partial y} \tag{3.4}$$

The terms u and v denote the velocity components in the x and y directions.

Integration of this equation over the control volume and time can be written as:

$$\int\int\int_t \frac{\partial(\rho\phi)}{\partial t} \; dt \; dA + \int_t\int\int \left(\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y}\right) \; dA \; dt = \int_t\int\int S \; dA \; dt \tag{3.5}$$

The first term can be expanded to:

$$\int\int\int_t \frac{\partial(\rho\phi)}{\partial t} \; dt \; dA = \left[(\rho\phi) - (\rho\phi)^0\right] \Delta A \tag{3.6}$$

The second term is:

$$\int_t\int\int \left(\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y}\right) \; dA \; dt = \int_t\int\int \left(\nabla \cdot \vec{J}\right) dA \; dt \tag{3.7}$$

Using Gauss-Divergence theorem:

$$\begin{aligned} \int_t \int \int \left(\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} \right) \, dA \, dt &= \int_t \oint \left(\vec{J} \cdot \hat{n} \right) \, dl \, dt \qquad (3.8) \\ &= \left[\alpha J_n^{int} + (1-\alpha) J_n^{int^0} \right] \, \Delta t \end{aligned}$$

$$(3.9)$$

where J_n^{int} is the integrated flux in Equation 2.52. Then, the second term can be computed as:

$$\int_t \int \int \left(\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} \right) \, dA \, dt = \left[\alpha \Sigma a_i \phi_i + (1-\alpha) \left(\Sigma a_i \phi_i \right)^0 \right] \, \Delta t \qquad (3.10)$$

The last term is:

$$\int_t \int \int S \, dA \, dt = \left[\alpha (S_C - S_P \phi_P) + (1-\alpha)(S_C^0 - S_P^0 \phi_P^0) \right] \, \Delta A \, \Delta t \qquad (3.11)$$

The source term S has been linearized as $S = S_C - S_P \phi_P$, where S_C denotes the constant part of the linearized source term and S_P the variable part.

The discretized form of this equation is obtained upon integration over the control volume and is given by the following equation.

$$\begin{aligned} &[(\rho\phi)_P - (\rho\phi)_P^0] \frac{\Delta A}{\Delta t} + \alpha \Sigma a_i \phi_i + (1-\alpha) \left(\Sigma a_i \phi_i \right)^0 = \\ &\qquad \left[\alpha (S_C - S_P \ \phi_P) + (1-\alpha)(S_C^0 - S_P^0 \phi_P^0) \right] \Delta A \qquad (3.12) \end{aligned}$$

For the unsteady term, ρ and ϕ are assumed to prevail over the entire control volume. The 'old' values (i.e., the values at the beginning of the time step) are denoted by ρ^0 and ϕ^0. This equation can be represented in the following simplified form:

$$\begin{aligned} \frac{\rho \, \Delta A}{\Delta t} \phi_P + \alpha \ a_P \ \phi_P + \alpha \ S_P \ \phi_P &= \alpha (\Sigma a_{nb} \ \phi_{nb}) + \alpha S_C \Delta A + \qquad (3.13) \\ &\quad (1-\alpha)(\Sigma a_{nb} \phi_{nb})^0 - (1-\alpha)(a_P \phi_P)^0 + \\ &\quad (1-\alpha)(S_C^0 - S_P^0 \phi^0) \Delta A + \frac{\rho \, \Delta A}{\Delta t} \phi_P^0 \end{aligned}$$

After rearrangement, the equation can be written as:

$$a_P\phi_P = \alpha(\Sigma a_{nb}\ \phi_{nb}) + b_{total} \tag{3.14}$$

where

$$\begin{aligned}
a_P &= \frac{\rho\ \Delta A}{\Delta t} + \alpha\ a_P + \alpha\ S_P\Delta A && (3.15)\\
b_{total} &= b + b_o && (3.16)\\
b &= \alpha S_C\Delta A && (3.17)\\
b_o &= (1-\alpha)[(\Sigma a_{nb}\phi_{nb})^0] \\
&\quad -(1-\alpha)[a_P - S_P^0\Delta A]\phi_P^0 \\
&\quad +\frac{\rho\Delta A}{\Delta t}\phi_P^0 + (1-\alpha)S_C^0\Delta A && (3.18)
\end{aligned}$$

All the terms with superscript 0 refer to values computed at the previous time step.

3.1.3 Time-Accurate Pseudo and Artificial Velocity

As discussed in Chapter 2, the artificial velocity components are defined as:

$$\begin{aligned}
\tilde{u} &= \hat{u} + d^u(S^u - \frac{\partial p}{\partial x}) && (3.19)\\
\tilde{v} &= \hat{v} + d^v(S^v - \frac{\partial p}{\partial y}) && (3.20)
\end{aligned}$$

From Equation 3.14, we can write the unsteady u-momentum equation as:

$$a_P^u u_P = \alpha(\Sigma a_{nb}^u\ u_{nb}) + b_{total}^u \tag{3.21}$$

If the pressure source is treated implicitly, the source term consisting of the unsteady source terms b^u and b_o^u becomes:

$$b^u = [\alpha S_C^u + (L_1p_1 + L_2p_2 + L_3p_3)]\,\Delta A \tag{3.22}$$

$$
\begin{aligned}
b_o^u &= (1-\alpha)[(\Sigma a_{nb}^u u_{nb})^0] \\
&\quad -(1-\alpha)[a_P^u - (S_P^u)^0\, A_P]u_P^0 \\
&\quad +\frac{\rho A_P}{\Delta t}u_P^0 + (1-\alpha)\,(S_C^u)^0\, A_P
\end{aligned}
$$

The term b^u has the pressure term, so we can rewrite the equation as:

$$
\begin{aligned}
a_P^u u_P &= \alpha(\Sigma a_{nb}^u\ u_{nb}) + b_o^u + [\alpha S_C^u + (L_1 p_1 + L_2 p_2 + L_3 p_3)]\, A_P && (3.23) \\
u_P &= \frac{\alpha(\Sigma a_{nb}^u\ u_{nb}) + b_o^u}{a_P^u} + \frac{A_P}{a_P^u}\,[\alpha S_C^u + (L_1 p_1 + L_2 p_2 + L_3 p_3)] && (3.24)
\end{aligned}
$$

Similar to Equations 2.62, now we can define the pseudo velocities as:

$$\hat{u} = \frac{\alpha(\Sigma a_{nb}^u\ u_{nb}) + b_o^u}{a_P^u} \tag{3.25}$$

and the source term coefficient as:

$$d^u = \frac{A_P}{a_P^u} \tag{3.26}$$

Even though Equation 3.26 looks similar to Equation 2.64, it is important to note that the grid point coefficient contains the unsteady terms as given in Equation 3.15:

On applying the same procedure to the v-momentum equation, we get:

$$
\begin{aligned}
\hat{v} &= \frac{\alpha(\Sigma a_{nb}^v\ v_{nb}) + b_o^v}{a_P^v} && (3.27) \\
d^v &= \frac{A_P}{a_P^v} && (3.28)
\end{aligned}
$$

where

$$a_P^v = \frac{\rho\ A_P}{\Delta t} + \alpha\ a_P^v + \alpha\ S_P^v A_P \tag{3.29}$$

The same assumptions about the variation of pseudo and artificial velocity are made. Thus, the interpolation of the artificial field $\vec{\tilde{u}}$ can follow the discussion of Section 2.5.4.

3.1.4 Pressure Equation

Starting from the integrated continuity equation for the control volume face $r - s - t$ (Figure 2.8), we have:

$$\int_r^t \rho\vec{\tilde{u}} \cdot \hat{n} dl = \rho \left[\tilde{u}_s \left(y_t - y_r\right) - \tilde{v}_s \left(x_t - x_r\right)\right] \tag{3.30}$$

Similar to Section 2.6, total integration of the continuity equation over the control volume faces yields:

$$a_P^p \, p_P = \sum a_{nb}^p \, p_{nb} + b^p \tag{3.31}$$

where:

$$a_i^p = \rho \left[\left(y_t - y_r\right) d_s^u \bar{L}_i - \left(x_t - x_r\right) d_s^v \bar{M}_i\right] \tag{3.32}$$

$$b^p = \rho \left[\left(y_t - y_r\right)\left(\hat{u}_s + d_s^u S^u\right) - \left(x_t - x_r\right)\left(\hat{v}_s + d_s^v S^v\right)\right] \tag{3.33}$$

The difference from the formulation in Chapter 2 is that the pseudo velocities are now dependent on the unsteady source terms from the previous time instant, b_o^u and b_o^v as computed in Equations 3.25 through 3.29 replacing Equations 2.62 through 2.65.

3.1.5 Explicit Runge-Kutta Method for Momentum Equations

The spatial discretization of the integral form of the Navier Stokes equations transforms this system of partial differential equations into a coupled set of ordinary differential equations. To integrate an equation of the type:

$$\frac{d\phi_i}{dt} = \frac{R_i(\phi_i)}{V_i} \tag{3.34}$$

the following explicit, four-stage Runge-Kutta scheme is adopted:

$$\phi^{(0)} \quad = \quad \phi^n \tag{3.35}$$

$$\phi^{(1)} \quad = \quad \phi^{(0)} + \frac{1}{4}\frac{\Delta t}{V_i} R_i(\phi^{(0)}) \tag{3.36}$$

$$\begin{aligned}
\phi^{(2)} &= \phi^{(0)} + \frac{1}{3}\frac{\Delta t}{V_i}R_i(\phi^{(1)}) \qquad (3.37)\\
\phi^{(3)} &= \phi^{(0)} + \frac{1}{2}\frac{\Delta t}{V_i}R_i(\phi^{(2)}) \qquad (3.38)\\
\phi^{(4)} &= \phi^{(0)} + \frac{\Delta t}{V_i}R_i(\phi^{(3)}) \qquad (3.39)\\
\phi^{n+1} &= \phi^{(4)} \qquad (3.40)
\end{aligned}$$

By carrying out only spatial integrations on the momentum equations, they can be transformed to the form:

$$\begin{aligned}
\frac{du}{dt} &= \frac{R_u^i}{\rho\Delta A} \qquad (3.41)\\
\frac{dv}{dt} &= \frac{R_v^i}{\rho\Delta A} \qquad (3.42)
\end{aligned}$$

The conservation form of the momentum equation is:

$$\frac{d(\rho u)}{dt}\Delta A + \oint \vec{J}\cdot\hat{n}dl = \int\int\left[S^u - \frac{\partial p}{\partial x}\right]dA \qquad (3.43)$$

Following the spatial discretization in Chapter 2 for the second term on the LHS and the term on the RHS, the equation can now be expressed as follows:

$$\frac{d(\rho u)}{dt}\Delta A = -\sum a_i u_i + \left[(S_C^u - S_P^u u_P) + (\bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3)\right]\Delta A \qquad (3.44)$$

On rearranging the terms and rewrite the flux term as a function of the velocity component at point P and its neighbors (Equation 2.53), the following equations are obtained:

$$\begin{aligned}
\frac{d(\rho u)}{dt}\Delta A &= \Sigma a_{nb}u_{nb} - a_P u_P - S_P^u\ \Delta A u_P + S_C^u\ \Delta A\\
&\quad +(\bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3)\Delta A \qquad (3.45)\\
\frac{d(\rho u)}{dt}\Delta A &= \Sigma a_{nb}u_{nb} - (a_P + S_P^u\ \Delta A)u_P + S_C^u\ \Delta A\\
&\quad +(\bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3)\Delta A \qquad (3.46)
\end{aligned}$$

Letting

$$a_P^u = a_P + S_P^u \Delta A \tag{3.47}$$

$$b_u = S_C^u\ \Delta A + (L_1 p_1 + L_2 p_2 + L_3 p_3)\Delta A \tag{3.48}$$

and assuming incompressible flow (density, ρ, is constant), the above equation is written as:

$$\frac{du}{dt} = \frac{(\Sigma a_{nb} u_{nb} - a_P^u u_P + b_u)}{\rho \Delta A} \tag{3.49}$$

If we define the residual function as:

$$R_u^i = \Sigma a_{nb} u_{nb} - a_P^u u_P + b_u \tag{3.50}$$

then the above equation reduces to

$$\frac{du}{dt} = \frac{R_u^i}{\rho \Delta A} \tag{3.51}$$

Similarly the $v-$ momentum equation can be cast to the form required as shown below:

$$\frac{dv}{dt} = \frac{(\Sigma a_{nb} v_{nb} - a_P^v v_P + b_v)}{\rho \Delta A} \tag{3.52}$$

$$a_P^v = a_P + S_P^v \Delta A \tag{3.53}$$

$$b_v = S_C^v\ \Delta A + (M_1 p_1 + M_2 p_2 + M_3 p_3)\Delta A \tag{3.54}$$

$$R_v^i = \Sigma a_{nb} v_{nb} - a_P^v v_P + b_v \tag{3.55}$$

$$\frac{dv}{dt} = \frac{R_v^i}{\rho \Delta A} \tag{3.56}$$

3.2 Solution Procedure

3.2.1 SIMPLER Algorithm

The SIMPLER algorithm is generally used in conjunction with a Fully-Implicit or Crank-Nicolson time integration as described in Section 3.1. The stepwise procedure for applying SIMPLER algorithm for the aforementioned discretized equations is as indicated below:

1. First, guess the initial values of u, v, and pressure, p, at all grid points within the domain.

2. Using the definitions of $\tilde{u}$ and $\tilde{v}$, set up the coefficients of the momentum equations.

3. Calculate the unsteady terms and modify the momentum coefficients accordingly to include these terms (Equations 3.15 through 3.18).

4. With the existing nodal velocities and the momentum coefficients generated in Step 2, calculate $\hat{u}$, $\hat{v}$, d^u, and d^v using Equations 3.25 to 3.29.

5. Calculate the coefficients and the source term for the pressure equation using Equations 3.32 and 3.33, and solve for the pressure field, p^*.

6. With the pressure field p^* and the momentum coefficients calculated in Step 2, solve the discretized momentum equations to get the nodal velocities (u^* and v^*). Additionally, update the fluxes, F^u and F^v, at the boundary points.

7. Using u^* and v^*, recalculate the pseudo velocities $\hat{u}^*$ and $\hat{v}^*$ to obtain the source term for the pressure correction equation.

8. The pressure correction equation is solved to obtain p', which in turn used to calculate $\tilde{u}'$, $\tilde{v}'$, u', and v'.

9. Correct the velocity components (u and v) using p' to be used for the next iteration.

10. Return to Step 3 and repeat until convergence.

11. Advance to the next time level.

Note that the pressure field is never corrected. Because the equations are non-linear, an iterative procedure is used. The pressure equation is not exact as in structured SIMPLER and hence, the pressure and momentum equations may need to be underrelaxed. However, the pressure correction equation should not be relaxed. The purpose of the correction procedure is to correct the velocity field such that it will conserve mass; consequently, a fairly converged solution of the correction equation is needed in order to ensure that the $\vec{\tilde{u}}$ field sufficiently satisfies continuity during each iteration.

3.2.2 Runge-Kutta SIMPLER Algorithm

To facilitate the implementation of the Runge-Kutta scheme in the SIMPLER algorithm, a few modifications have to be made to the procedure. The modified algorithm is as follows:

1. Initially, guess the values of u, v, and pressure, p, at all grid points within the domain.

2. Using the definitions of $\tilde{u}$ and $\tilde{v}$, set up the coefficients of the momentum equations.

3. Calculate the unsteady terms and modify the momentum coefficients accordingly to include these terms (Equations 3.15 through 3.18).

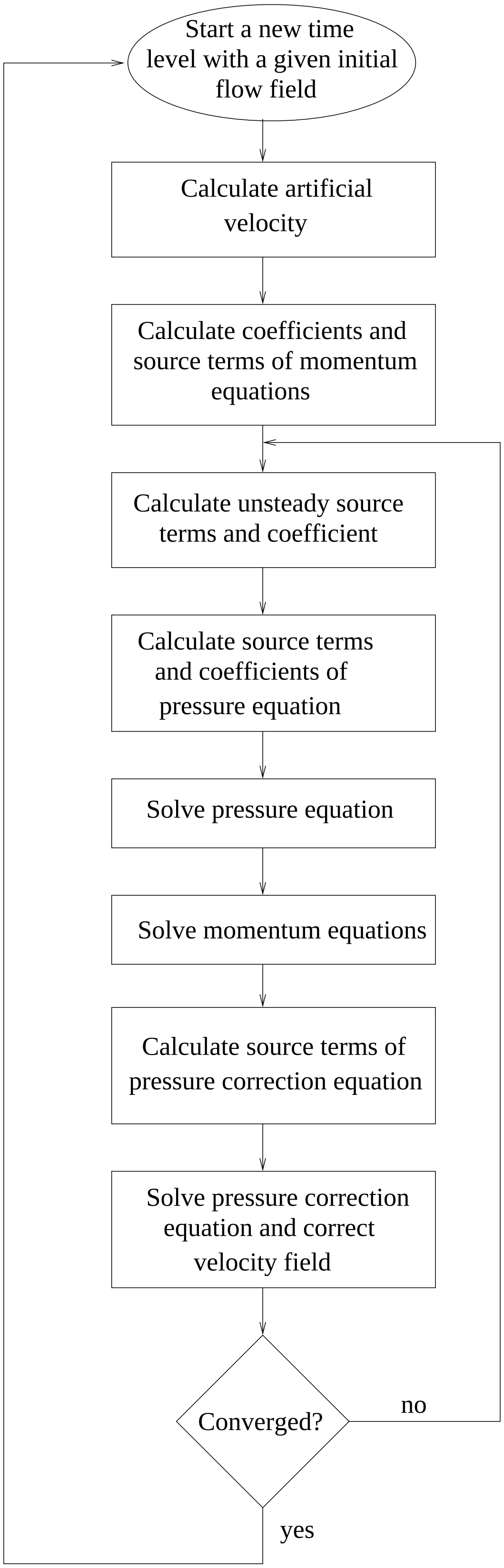
4. With the modified nodal velocities and the momentum coefficients in Step 3, calculate $\hat{u}$, $\hat{v}$, d^u, and d^v using Equations 3.25 to 3.29.

5. Calculate the coefficients and the source term for the pressure equation using Equations 3.32 and 3.33, and solve for the pressure field, p^*.

6. With the pressure field as the source, update the velocity components (u and v) using four-stage Runge-Kutta algorithm. In this step, the original velocity coefficients and source terms without the unsteady terms (Step 2) are used.

7. Advance to the next time level.

Note that in the Runge-Kutta algorithm, no pressure-correction equation is used to correct the velocities. The flowchart for the two algorithms are shown in Figures 3.1 and 3.2.



Start a new time level with a given initial flow field
Calculate artificial velocity
Calculate coefficients and source terms of momentum equations
Calculate unsteady source terms and coefficient
Calculate source terms and coefficients of pressure equation
Solve pressure equation
Solve momentum equations
Calculate source terms of pressure correction equation
Solve pressure correction equation and correct velocity field
Converged?
no
yes

Figure 3.1 SIMPLER algorithm using Fully-Implicit or Crank-Nicolson time integration method.

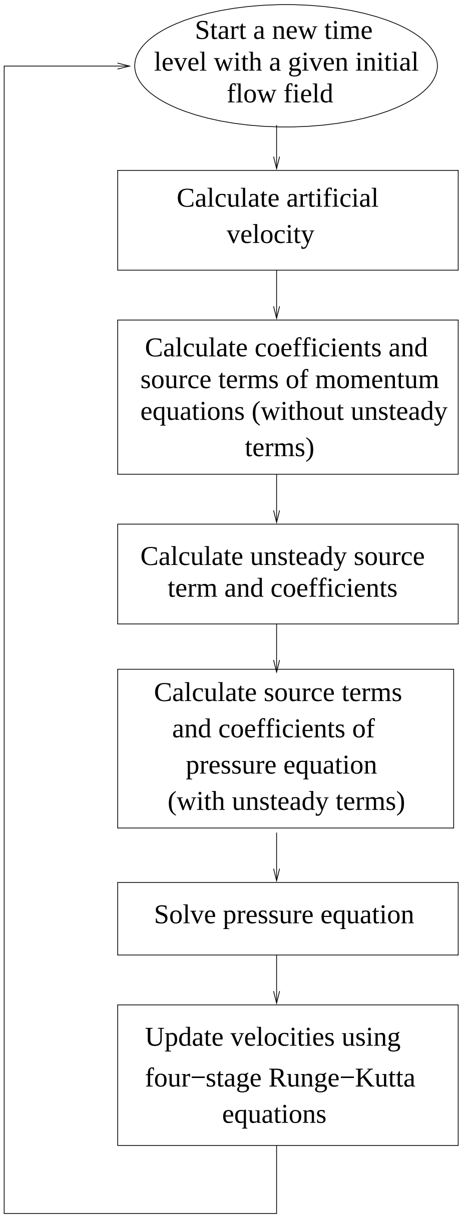

Start a new time level with a given initial flow field
Calculate artificial velocity
Calculate coefficients and source terms of momentum equations (without unsteady terms)
Calculate unsteady source term and coefficients
Calculate source terms and coefficients of pressure equation (with unsteady terms)
Solve pressure equation
Update velocities using four−stage Runge−Kutta equations

Figure 3.2 SIMPLER algorithm using 4-stage Runge-Kutta time integration method.

CHAPTER 4. RESULTS AND DISCUSSION

4.1 Lid Driven Cavity

The standard test case of a flow field within a two dimensional, lid driven square cavity is used to validate the solver. The configuration of the cavity is shown in Figure 4.1. The lid which moves at a constant speed, drives the flow inside the cavity due to the transport of shear stress by the molecular viscosity. This problem has a well defined geometry and boundary condition. In addition, the balance of convection and diffusion associated with recirculating regions present in the flow makes this problem an excellent case for testing numerical schemes.

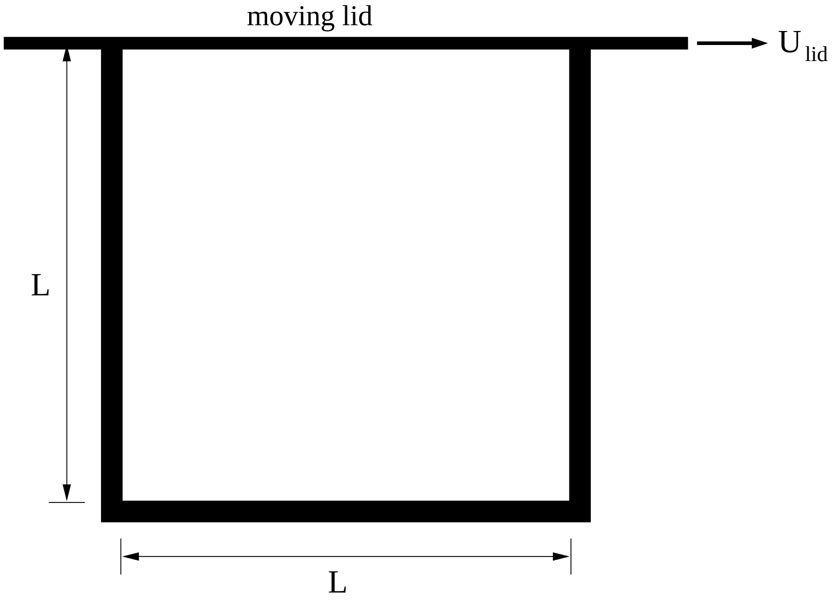


Figure 4.1 Schematic of lid driven Cartesian cavity.

To simulate the motion of the lid, the velocity at the top boundary of the cavity is set at 1 ft/s. The cavity is a square with unit dimensions in width and height. The four boundaries are viscous walls where a no-slip condition is applied. The flow is characterized by the Reynolds number, which is defined by:

$$Re = \frac{\rho \, U_{lid} \, L}{\mu} \tag{4.1}$$

where ρ is the density of the fluid, U_{lid} is the speed of the moving lid, L is the length and height of the cavity, and μ is the viscosity of the fluid. For the given geometry, three values of Reynolds number, 100, 400 and 1000, were simulated and compared with benchmark results from Ghia [64].

Two computational grids were used in the numerical simulations, as shown in Figures 4.2 and 4.3. The unstructured grids are generated from Cartesian structured grids, where each quadrilateral cell is subdivided into two triangles. The grids have a non-uniform distribution with higher grid density near the wall boundaries. The coarser of the two structured meshes has 52×52 grid points in the height and width direction, while the finer one has 129×129 grid points. The time step size was taken to be $0.01s$ for the cases using Fully-Implicit and Crank-Nicolson scheme, while it was reduced to $0.001s$ for the Runge-Kutta scheme.

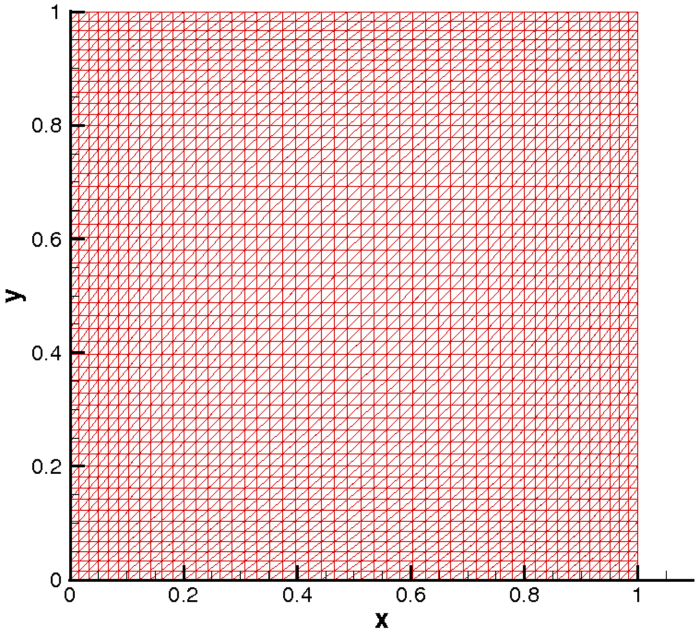


Figure 4.2 The computational unstructured grid triangulated from 52×52 grid.

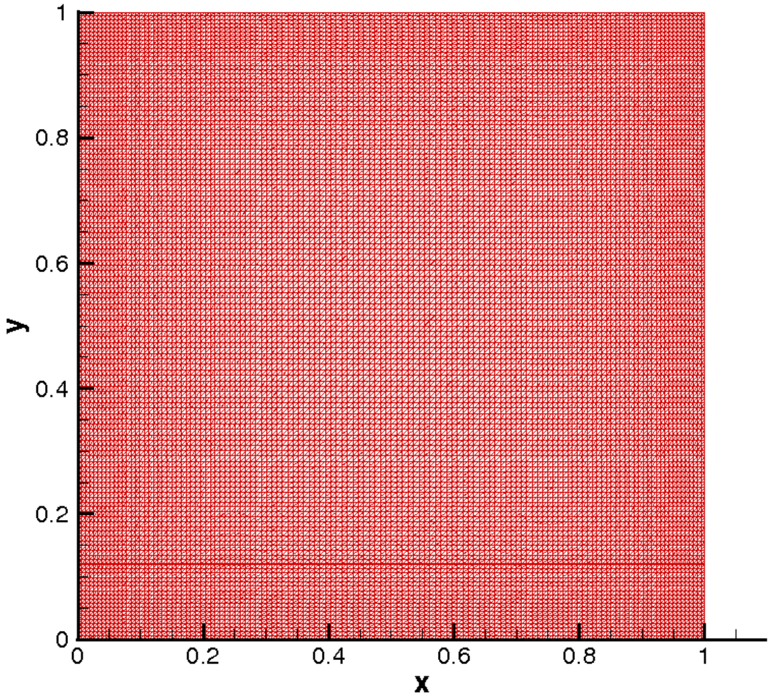


Figure 4.3 The computational unstructured grid triangulated from 129×129 grid.

4.1.1 Calculations for Re=100

For the Reynolds number of 100, the coarser grid (52×52, Figure 4.2) was used. The mass convergence history is given in Figure 4.4. It is shown that for all the unsteady algorithms, the mass residual has converged to a reasonable level. The Fully-Implicit scheme shows the best convergence level compared to the others, while the Runge-Kutta scheme gives the lowest level of convergence. For all the schemes, oscillations of the mass residual are seen at a short initial period of the flow development. Figure 4.5 shows the u-velocity profile on the vertical centerline of the unit cavity, while the v-velocity profile on the horizontal centerline is plotted in Figure 4.6. The results obtained from unsteady simulations using Fully-Implicit, Crank-

Nicolson and Runge-Kutta show an excellent agreement with the reference [64]. This shows that for the current Reynolds number, the grid resolution is sufficient to predict the flow field. The transient history of the u-velocity component at the center of the cavity is plotted in Figure 4.7. The results match well with previous computation by Pletcher [65].

The streamlines from the current analysis can be seen in Figures 4.8 through 4.10, respectively. The streamline profiles for all the schemes match well. The main vortex is slightly off center, closer to the upper right corner. Two secondary vortices are observed at the left and right lower corners of the cavity. The size of the right vortex is slightly larger than that of the left one.

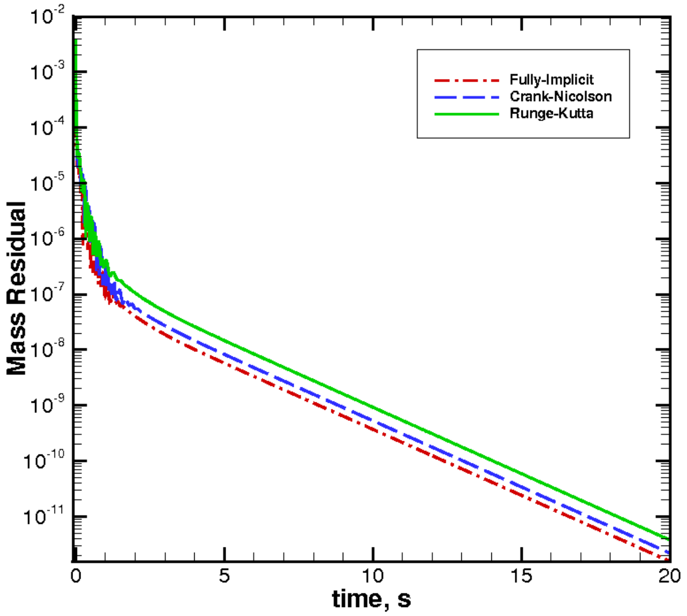


Figure 4.4 Mass residual history for Re=100 (52×52 grid).

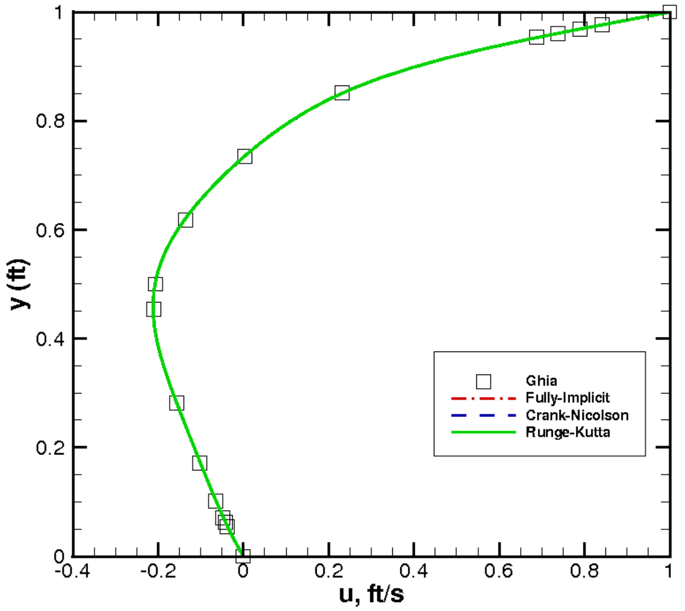


Figure 4.5 Vertical centerline u-velocity for Re=100 (52×52 grid).

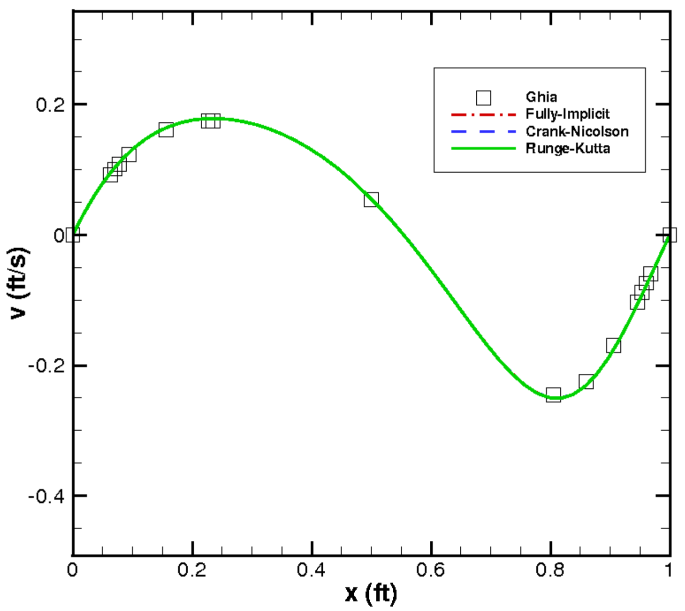


Figure 4.6 Horizontal centerline v-velocity for Re=100, 52×52 grid.

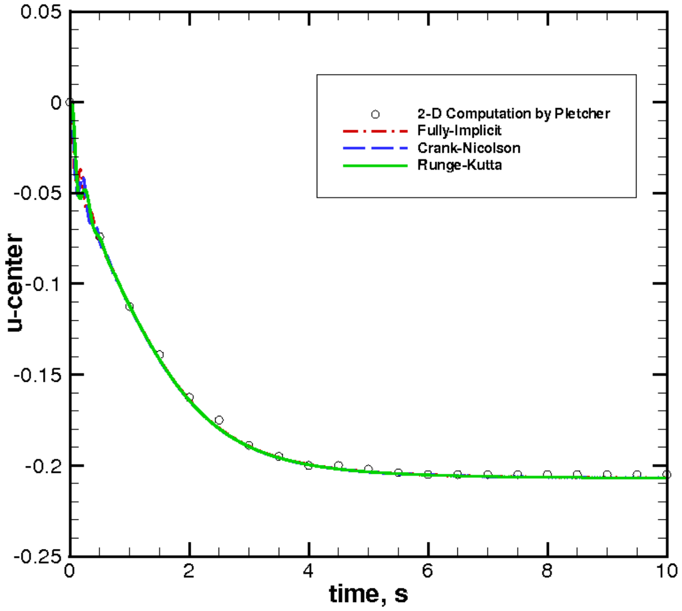


Figure 4.7 Time history of u-velocity at the center for Re=100.

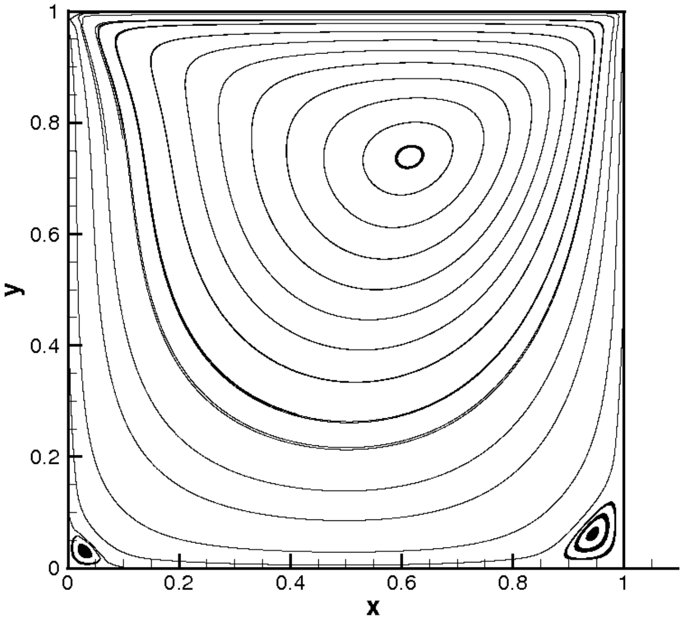


Figure 4.8 Streamlines for Re=100, 52x52 grid, Fully-Implicit.

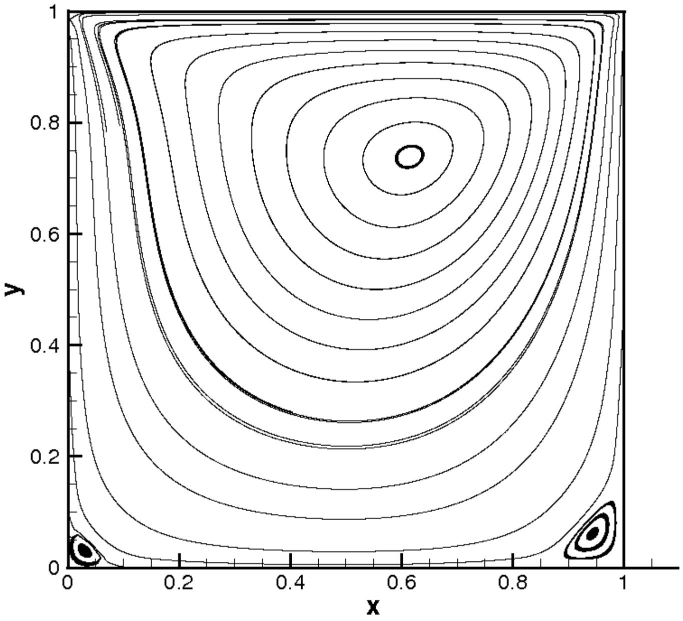


Figure 4.9 Streamlines for Re=100, 52x52 grid, Crank-Nicolson.

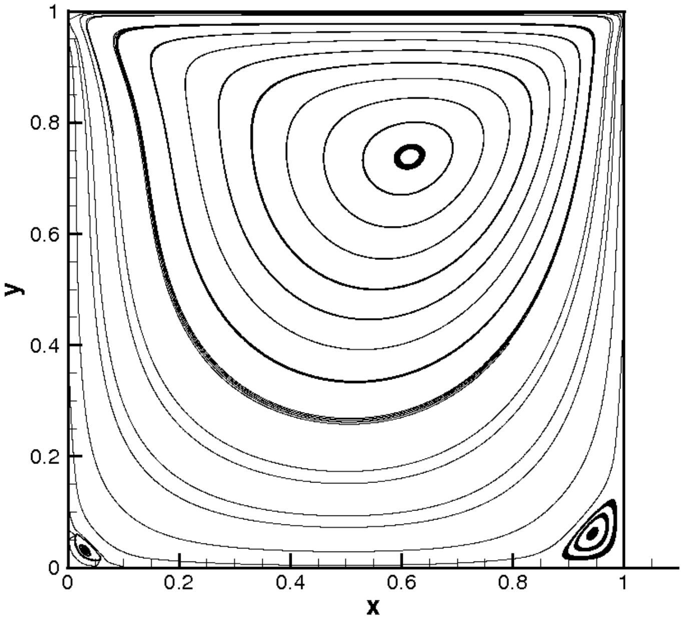


Figure 4.10 Streamlines for Re=100, 52x52 grid, Runge-Kutta.

4.1.2 Calculations for Re=400

Both meshes in Figures 4.2 and 4.3 were used to simulate $Re = 400$. Figure 4.11 gives the mass convergence history for this Reynolds number using the coarser grid. It is observed that Runge-Kutta has slightly lower convergence rate compared to the other two algorithms. Similar to the previous case, oscillations of the mass residual at the initial period of the flow development is again observed here. The u-velocity and v-velocity profile on the vertical and horizontal centerline of the cavity for a Reynolds number of 400 obtained using this grid are shown in Figures 4.12 and 4.13. Qualitatively the current results show similar profile with the results from experiment, but the discrepancies are quite significant. One can infer that for this Reynolds number, the grid resolution is not sufficient to simulate the flow accurately. It is also observed that Fully-Implicit, Crank-Nicolson and Runge-Kutta generate similar results. The transient history of the u-velocity component at the center of the cavity is shown in Figure 4.14. The Fully-Implicit scheme and Crank-Nicolson scheme show an excellent agreement. However, a small discrepancy is observed for the Runge-Kutta scheme. This might be due to the different time step size used in the Runge-Kutta scheme and the other two schemes. Comparison with the previous computation by Pletcher gives a reasonable agreement qualitatively and quantitatively.

Figure 4.15 gives the mass convergence history for this Reynolds number using the fine 129×129 grid. From the figure, the mass residual has larger oscillations at the beginning of the simulation compared to the result from the coarser grid. In Figures 4.16 and 4.17, the u-velocity and v-velocity profile on the horizontal and vertical centerline are plotted for the simulations using the refined mesh. As expected, the more refined mesh gives a more accurate profile than the coarser mesh. Figure 4.18 gives the transient history of the u-velocity component at the center of the cavity. Compared to the results from the coarser grid, the magnitude of the u-velocity component at the center reaches a higher maximum value before it decreases and reaches a converged value. This is consistent with the change in the transient behavior of the flow at the center of the cavity with grid refinement as observed by Wirogo [66].

The streamlines for the reference case is shown in Figure 4.19. Figures 4.20 through 4.22

show the streamlines for the flow field computed by the current analysis using the 52×52 grid, while Figures 4.23 through 4.25 depict the streamlines computed using the 129×129 grid. A good comparison was observed for all the schemes. The main circular flow is still centered about a point which is slightly above and to the right of the center of the cavity. However, on comparison with the Re=100 case, the center of the main vortex has moved slightly downward and towards the left. Similar to the previous case, the circular flow from the primary vortex creates two secondary vortices in both the lower right and lower left corners. The rotation of the flow around the secondary vortices is in the counter-clockwise direction. The size of these vortices are larger than observed in Re=100. The growth in the size of the vortices is evident for the vortex at the lower right corner of the cavity which is bigger than the vortex on the left.

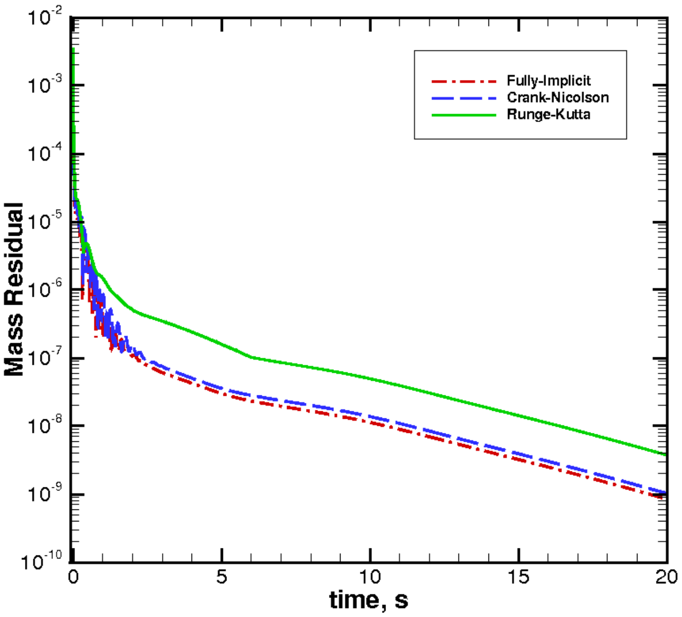


Figure 4.11 Mass residual history for Re=400 (52×52 grid).

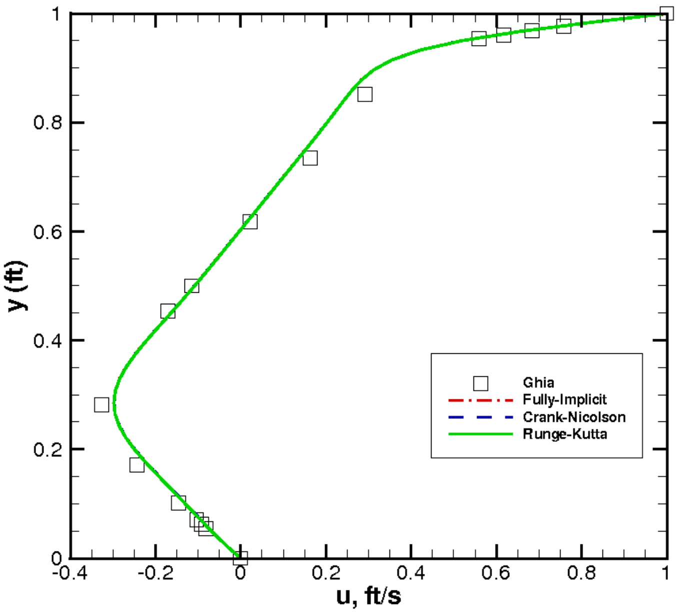


Figure 4.12 Vertical centerline u-velocity for Re=400 (52×52 grid).

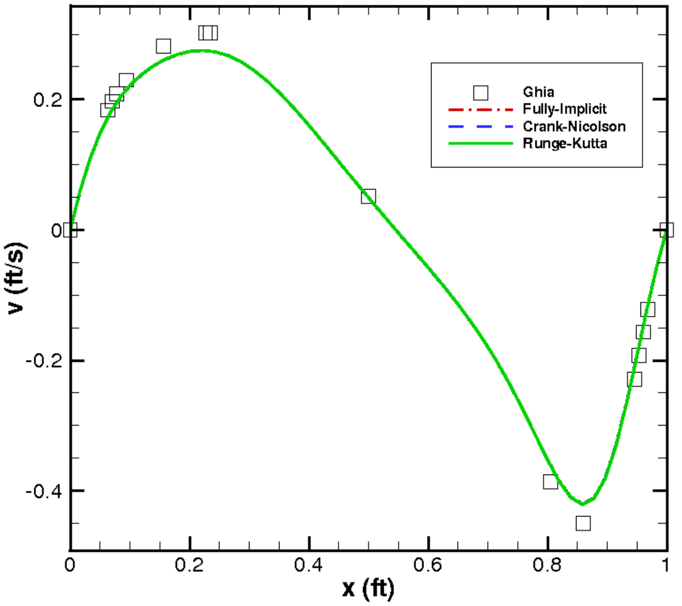


Figure 4.13 Horizontal centerline v-velocity for Re=400 (52×52 grid).

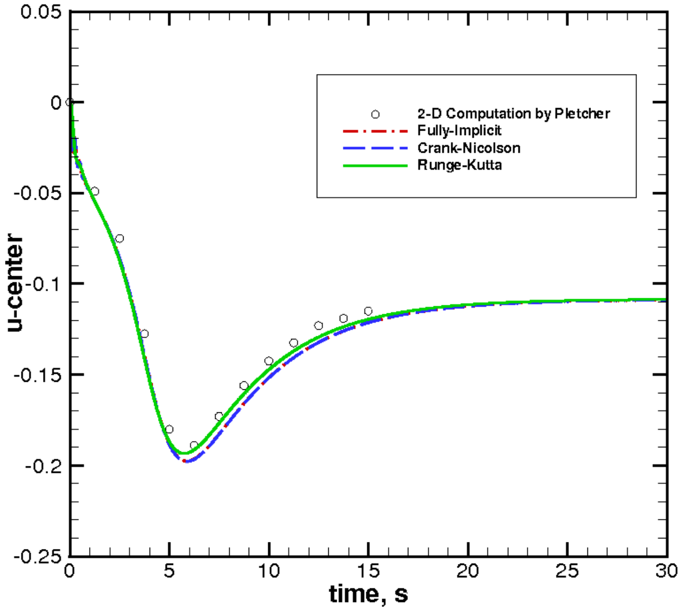


Figure 4.14 Time history of u-velocity at the center for Re=400 (52×52 grid).

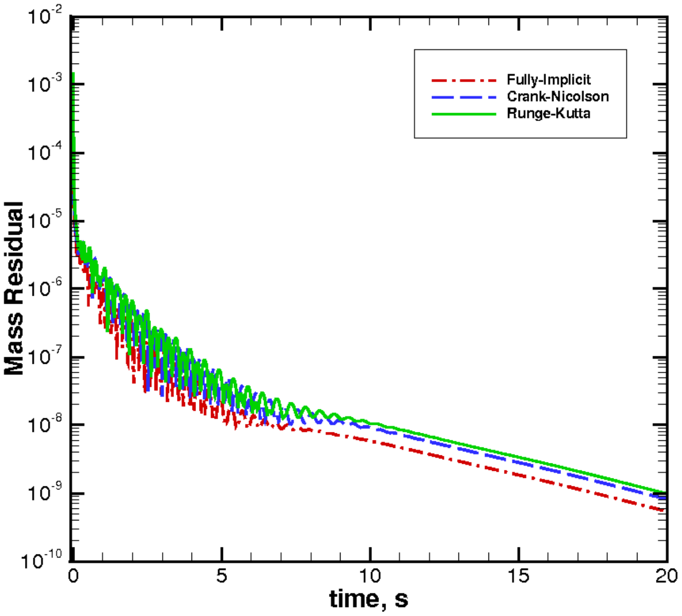


Figure 4.15 Mass residual history for Re=400 (129×129 grid).

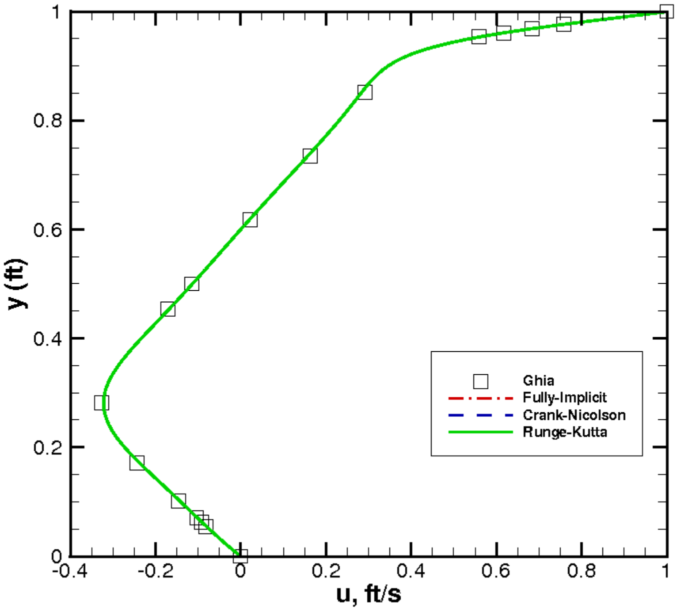


Figure 4.16 Vertical centerline u-velocity for Re=400 (129×129 grid).

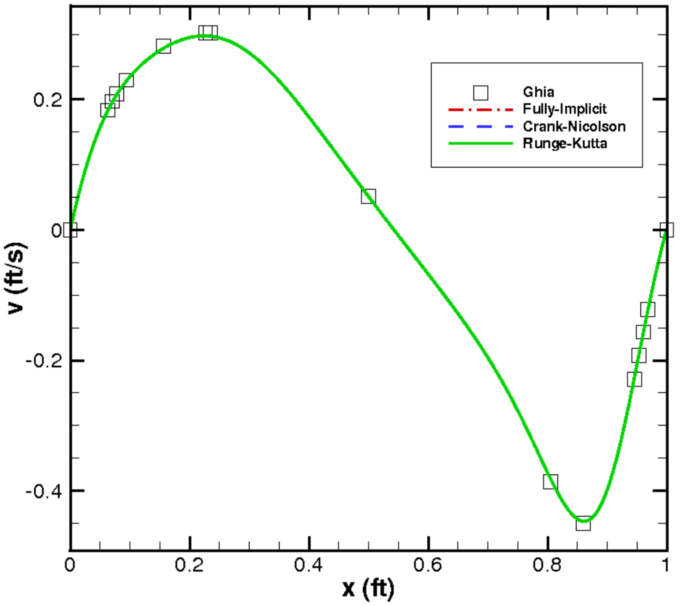


Figure 4.17 Horizontal centerline v-velocity for Re=400 (129×129 grid).

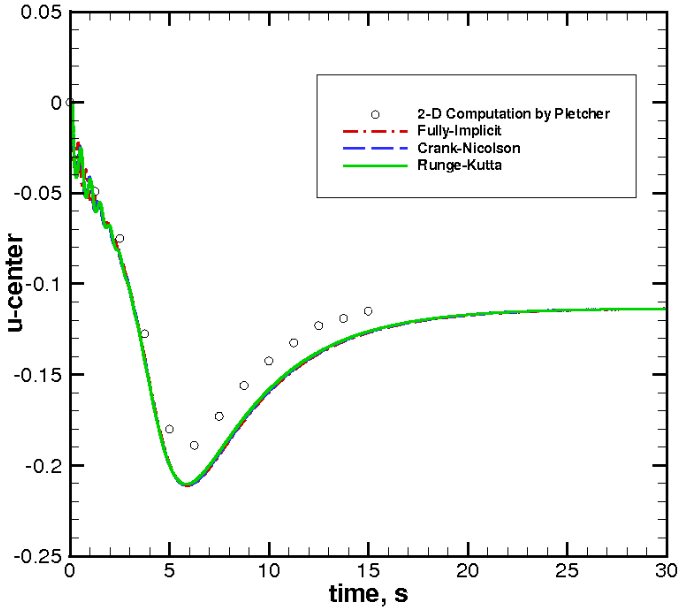


Figure 4.18 Time history of u-velocity at the center for Re=400 (129×129 grid).

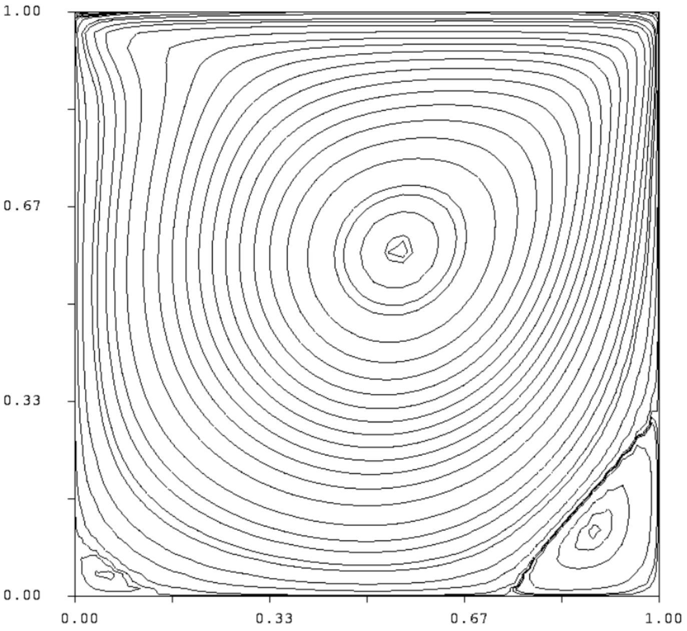


Figure 4.19 Streamlines for Re=400, Flux Corrected Method (FCM).

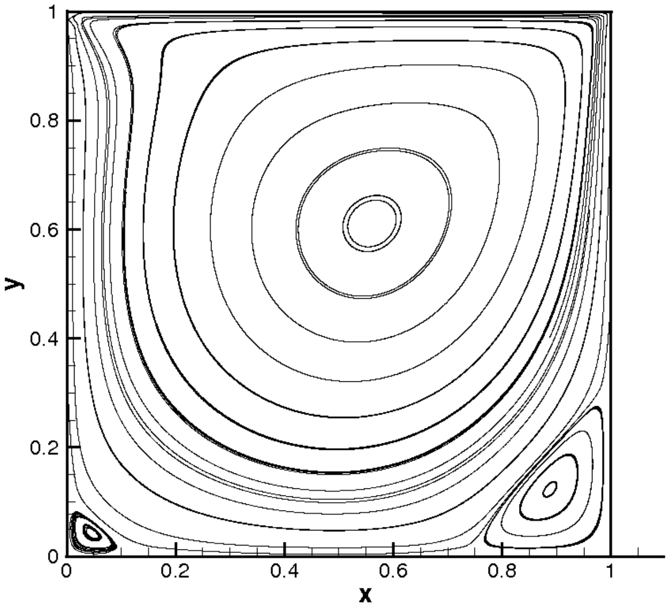


Figure 4.20 Streamlines for Re=400, 52×52 grid, Fully-Implicit.

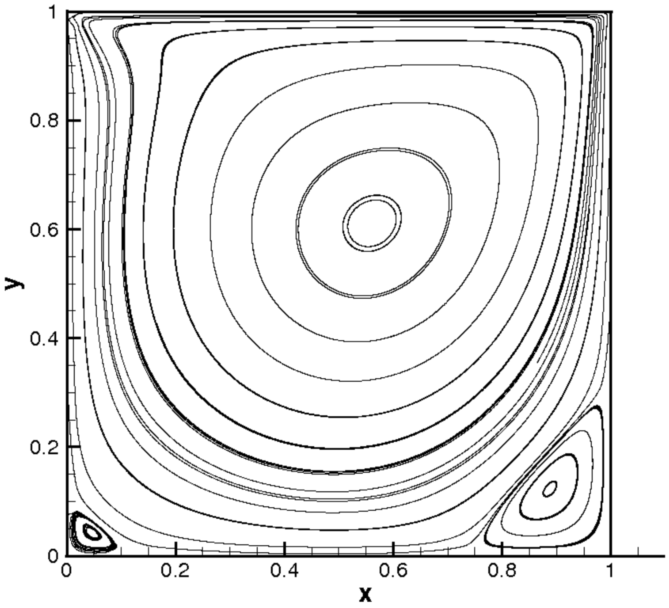


Figure 4.21 Streamlines for Re=400, 52×52 grid, Crank-Nicolson.

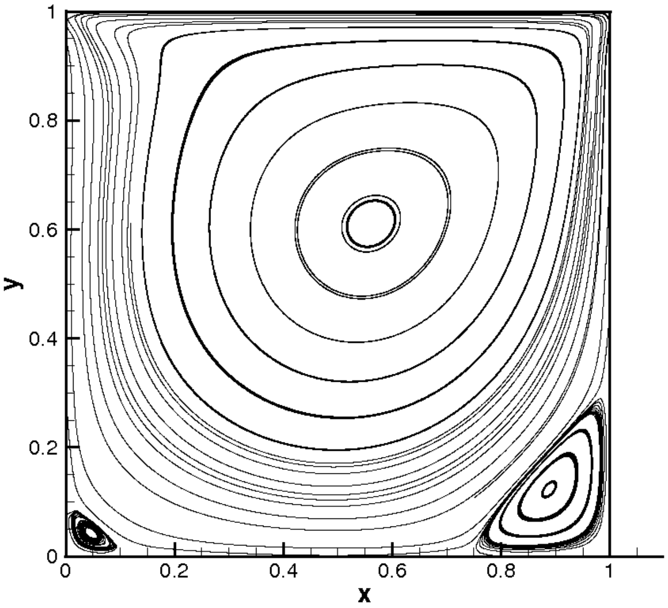


Figure 4.22 Streamlines for Re=400, 52×52 grid, Runge-Kutta.

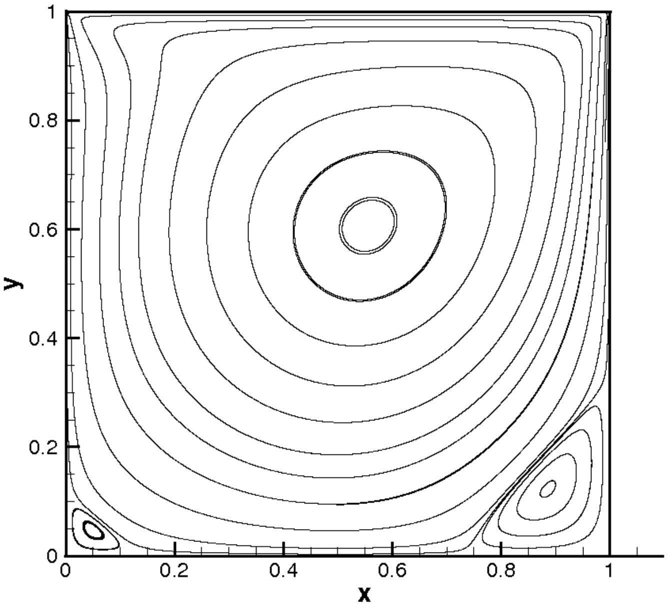


Figure 4.23 Streamlines for Re=400, 129×129 grid, Fully-Implicit.

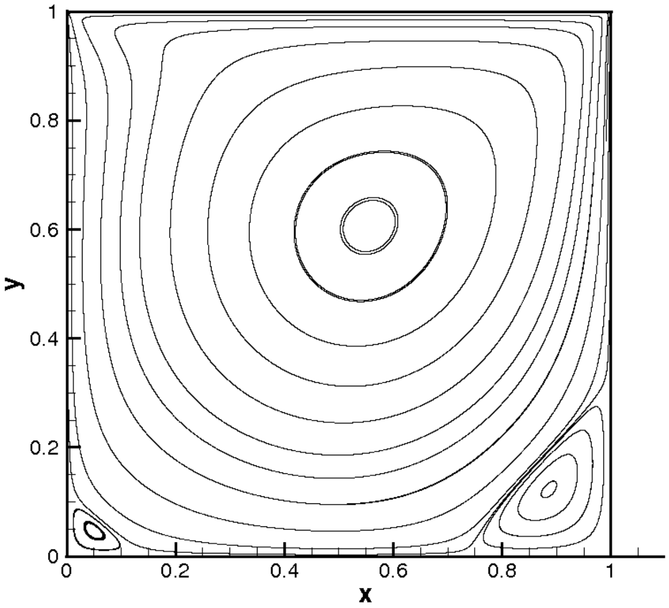


Figure 4.24 Streamlines for Re=400, 129×129 grid, Crank-Nicolson.

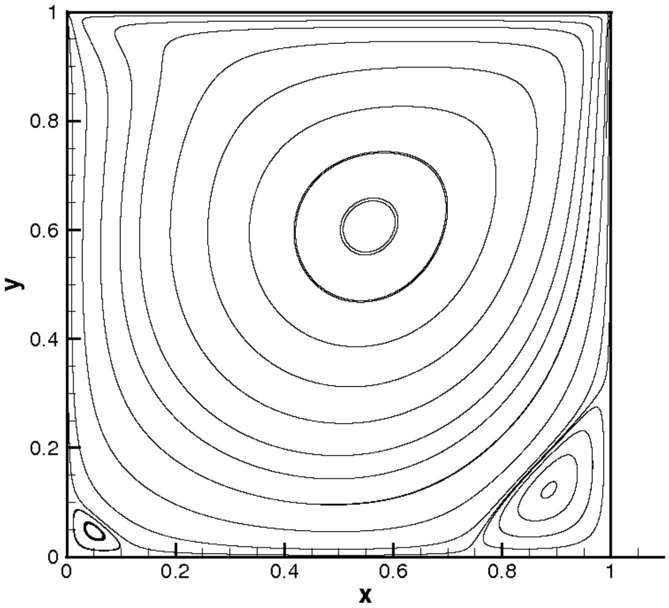


Figure 4.25 Streamlines for Re=400, 129×129 grid, Runge-Kutta.

4.1.3 Calculations for Re=1000

Considering the accuracy of the results obtained for Re=400, the simulations of the cavity flow with a Reynolds number of 1000 was carried out using the fine grid (129×129). The mass convergence history is shown in Figure 4.26. This plot shows a decrease in the convergence rate compared to the case of Re=400. In addition, larger oscillations at the initial period when the vortex is developing are observed. The u and v velocity profiles on the cavity centerlines are presented in Figures 4.27 and 4.28 For this case, the predicted u and v velocities on the vertical and horizontal centerlines are less accurate; however, the results are still reasonably close to the reference values. The plot of the u-velocity component at the center of cavity versus time is given in Figure 4.29. The results from the current algorithms are compared with the reference [66]. A good agreement is observed for all the schemes, which concludes that the algorithms developed in the current research is capable to follow the high transient gradient between $t = 5s$ and $t = 8s$.

Figures 4.31 through 4.33 shows the streamline profile for this Reynolds number using Fully-Implicit, Crank-Nicolson, and Runge-Kutta, respectively. The streamlines in the cavity for the case of Re=1000 exhibit the same structure as that of the Re=400 case. Some small changes are apparent in the location and size of the vortices. The vortex in the lower left has increased in size and the center of rotation has moved up and towards the right. The vortex in the lower right has stayed approximately the same size. The center has moved slightly lower than what was observed in the previous case.

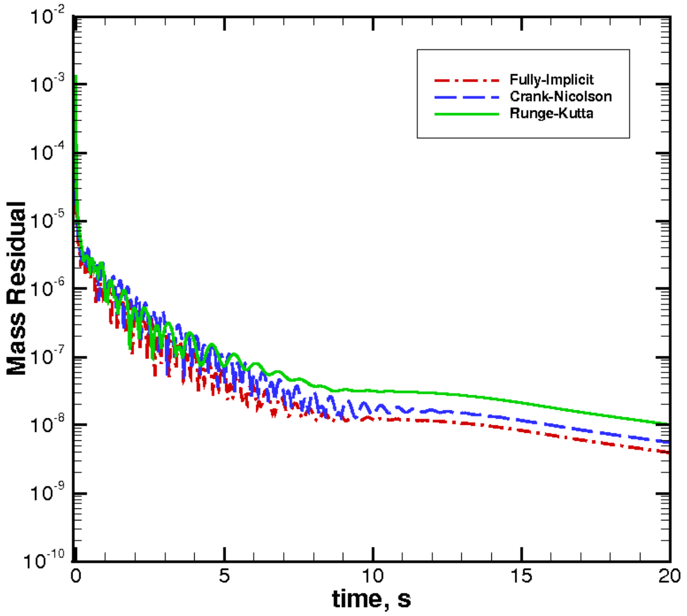


Figure 4.26 Mass residual history for Re=1000 (129×129 grid).

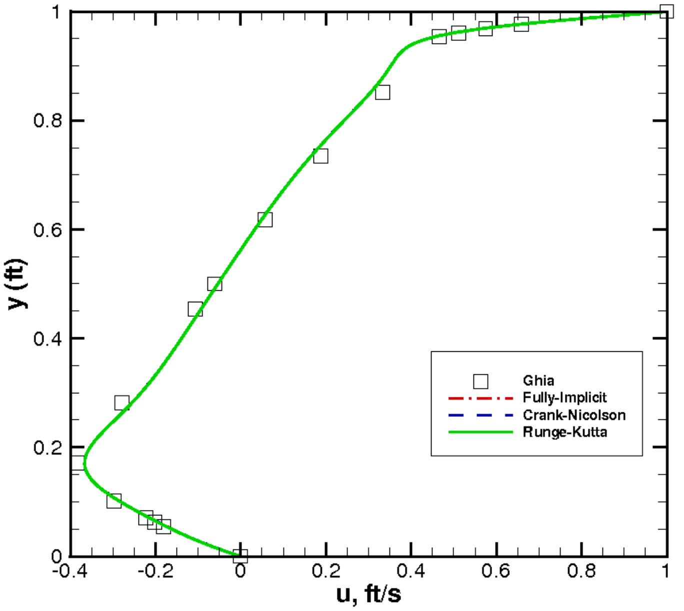


Figure 4.27 Vertical centerline u-velocity for Re=1000 (129×129 grid).

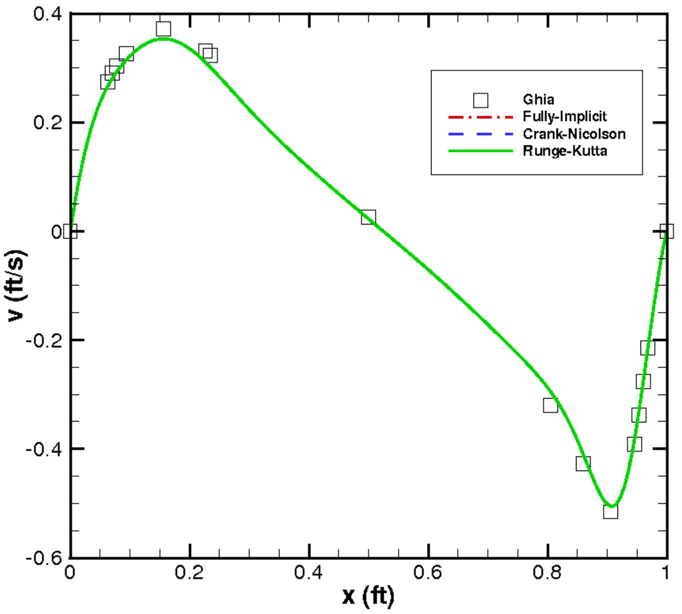


Figure 4.28 Horizontal centerline v-velocity for Re=1000 (129×129 grid).

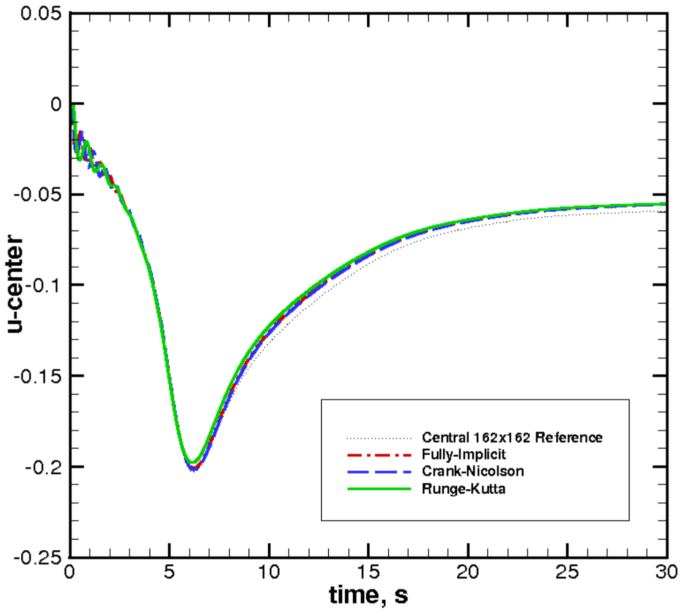


Figure 4.29 Time history of u-velocity at the center for Re=1000 (129×129 grid).

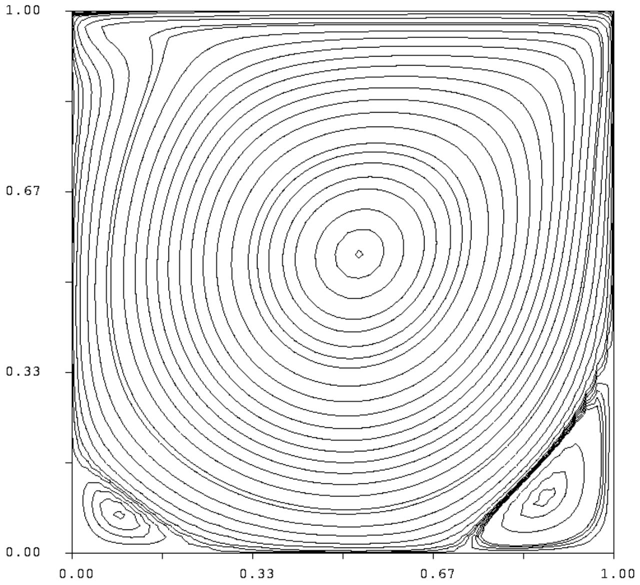


Figure 4.30 Streamlines for Re=1000, Flux Corrected Method (FCM).

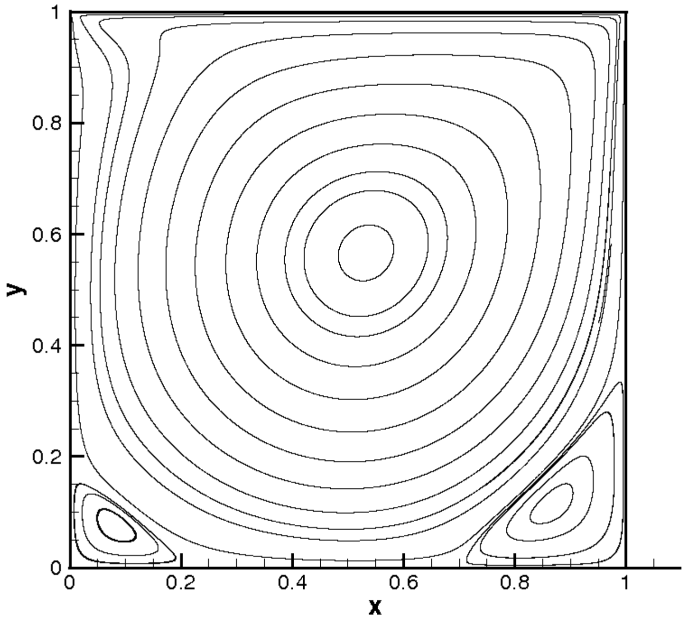


Figure 4.31 Streamlines for Re=1000, 129×129 grid, Fully-Implicit.

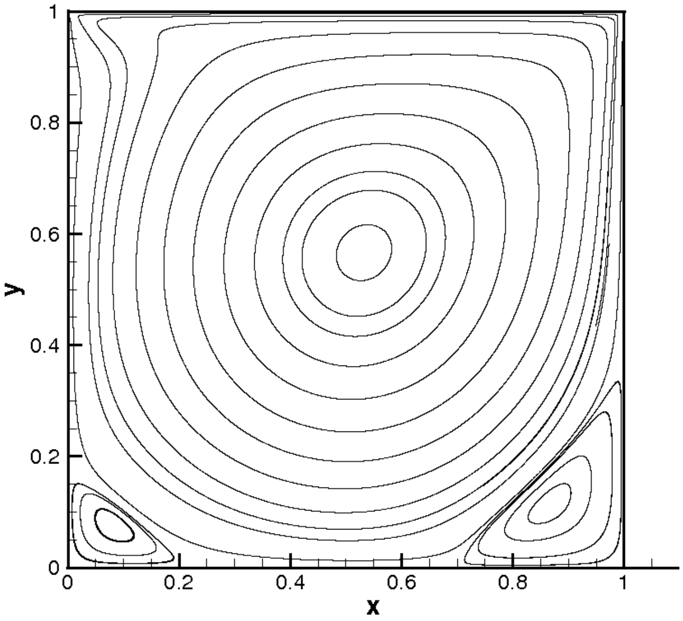


Figure 4.32 Streamlines for Re=1000, 129×129 grid, Crank-Nicolson.

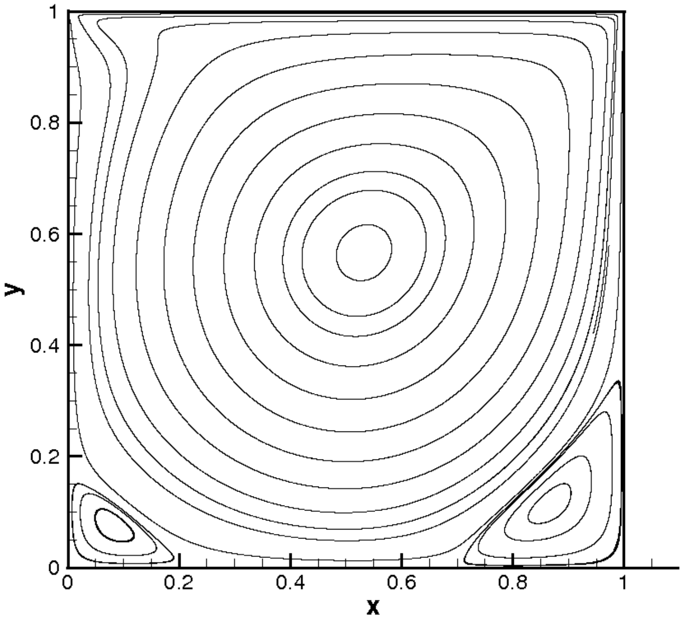


Figure 4.33 Streamlines for Re=1000, 129×129 grid, Runge-Kutta.

4.2 Backward-Facing Step Channel Flow

The second validation case is simulation of flow through a channel with a step. This case has been recognized as a good validation case due to the special characteristic exhibited by the flow due to the interaction between the reverse flow behind the step and the shear layer from the step edge. Another reason is that for a given expansion ratio and inlet profile, the reattachment point of the separated region behind the step is only dependent on Reynolds number.

Figure 4.34 shows a schematic of the step channel. The length of the channel, L, is 20 ft. The inlet of the channel has a height of 0.5319 ft while the overall height of the channel, H, is 1.0319 ft. The expansion ratio, η, which is defined as the ratio of the height of the channel to the inlet height is 1.94, to match with the experimental setup by Armaly [67]. Most previous computational results used an expansion ratio of 2. However, Wirogo [66] has shown that a slight difference in the expansion ratio of η=1.94 and η=2 produces negligible differences in the computed primary reattachment length, x_1.

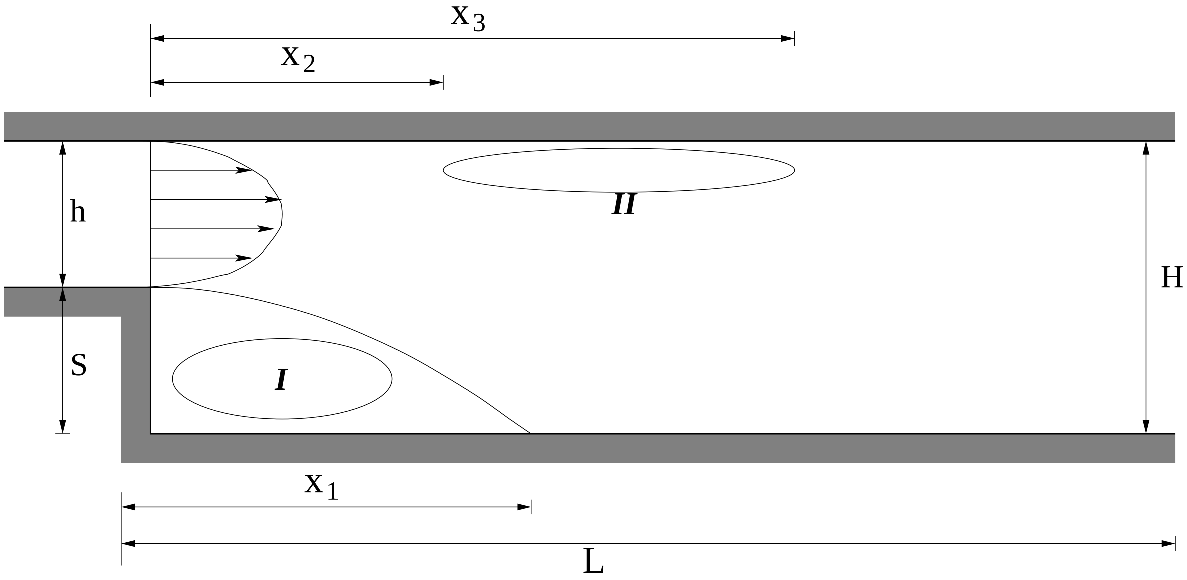


Figure 4.34 Schematic of the step channel.

The profile of the velocity entering the channel is set to model the profile that would exist if the flow was traveling through a channel with the diameter of the inlet. It is a parabolic profile for the horizontal velocity component u:

$$u = c(y - 0.5)(y - 1.0319) \tag{4.2}$$

where the constant c is determined to give a chosen average velocity along the inlet. For the current research, the average velocity of the inflow across the inlet is set to 1 ft/s.

Five cases were tested for Reynolds numbers of 100, 200, 400, 600 and 800. The Reynolds number for this problem is given by:

$$Re = \frac{\rho \, U_{avg} \, H}{\mu} \tag{4.3}$$

where U_{avg} is the average inlet velocity, ρ is the fluid density, and μ is the dynamic viscosity of the fluid. Previous investigations show that Regions I and II (in Figure 4.34) vary in size and are a function of the Reynolds number.

The computational grid used for this case has 7041 nodes and 20,432 faces. It is clustered near the inlet and the recirculating Region I of the channel. The grid is shown in Figures 4.35 and 4.36. For cases using Fully-Implicit and Crank-Nicolson scheme, a time step size of $0.01s$ was used, while for Runge-Kutta, the time step size was $0.001s$.

Figure 4.37 is a plot of the length of Region I scaled by the step height for different Reynolds numbers. Results are compared with the experimental results from Armaly [67] and other CFD methods such as the second-order accurate finite difference Fractional Step Method used by Kim and Moin [12]. At lower, fully laminar Reynolds numbers, the current results match the experimental data very well. However, at higher Reynolds number, the results deviate slightly from the experimental data. This may be due to the effects of three-dimensionality in the real experiment which has been discussed and shown by Armaly in the spanwise velocity plot [67].

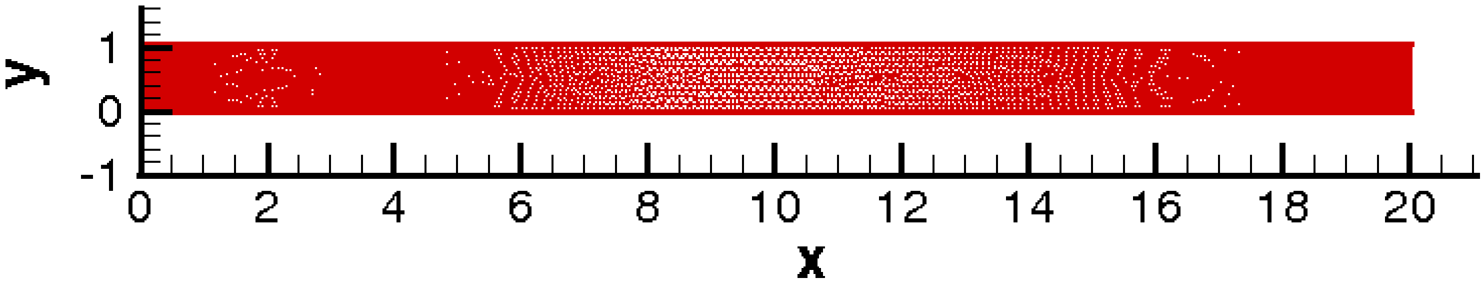


Figure 4.35 Computational grid for the backward step channel.

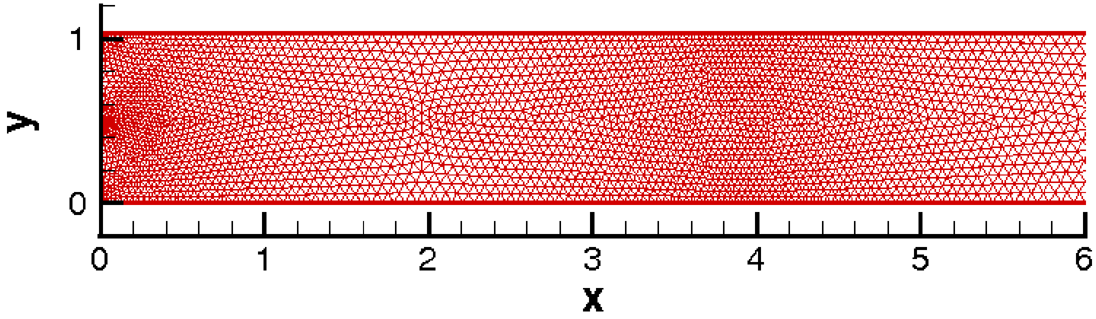


Figure 4.36 Close up of the grid at channel entrance.

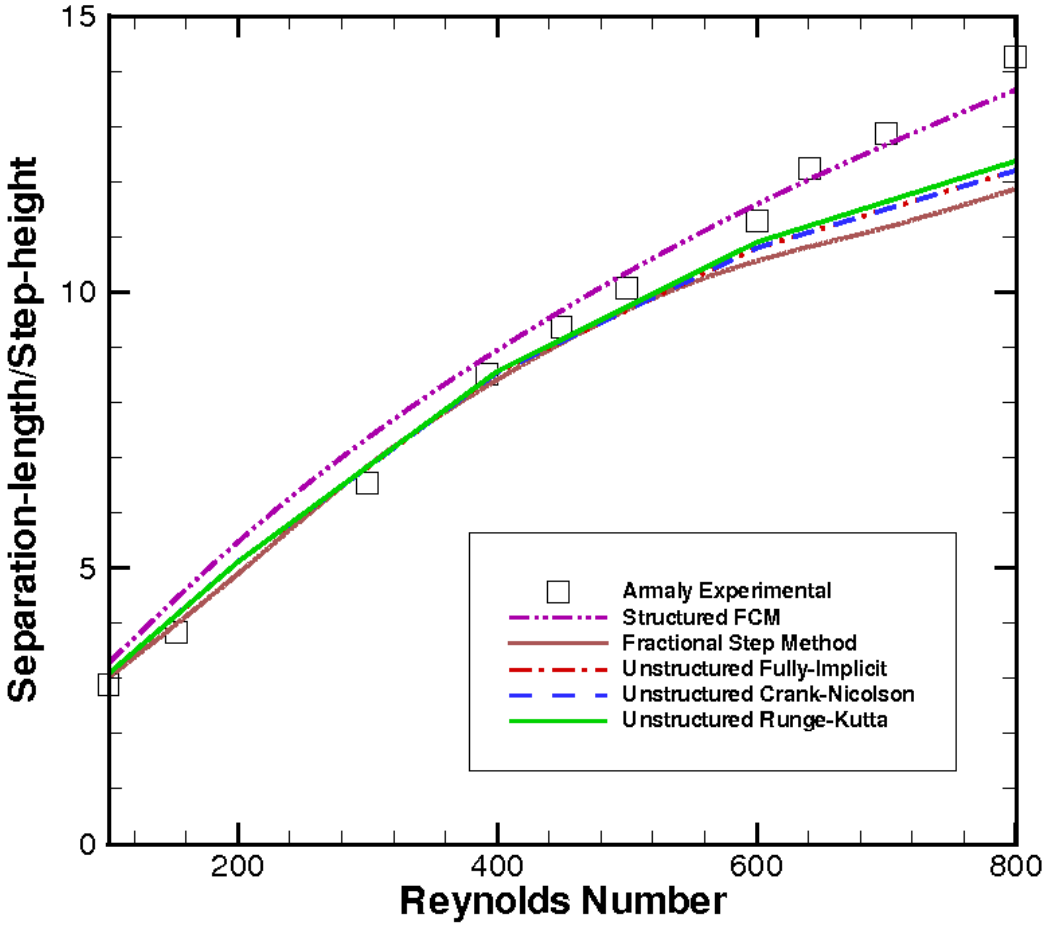


Figure 4.37 Comparison of separation length for different Reynolds numbers.

Figure 4.38 shows the plots of the streamlines within the channel for all of the Reynolds numbers tested using Fully-Implicit scheme. In these plots, Regions I and II can clearly be identified. Figure 4.38(a) shows a plot of the streamlines for the case when the Reynolds number of the flow is 100. Region I is small and Region II does not appear to exist. Region I rotates in a clockwise direction. In Figure 4.38(b), when the Reynolds number has been increased to 200, the size of Region I has increased significantly over that seen in the $Re = 100$ case. The center of rotation for Region I has also shifted downstream. Region II does not exist for this Reynolds number. Figure 4.38(c) shows the streamlines for $Re = 400$. The size of Region I seems to demonstrate a linear growth with increasing Reynolds number up to this point. This can also be verified in Figure 4.37. Although for this flow Region II still does not exist, some evidence of the emerging region can be seen. The streamlines on the upper wall directly over the end of Region I appear to grow indicating that a vortex is beginning to develop.

The Reynolds number is raised to 600 in Figure 4.38(d). Region I has grown slightly and Region II is now fully developed. The Region II vortex rotates counter-clockwise, the opposite direction of Region I. Region II is centered slightly downstream of the end of Region I. Figure 4.38(e) shows the streamlines for the channel flow when the Reynolds number is 800. The size of Region I has grown larger, as expected from the experimental results. In addition, Region II has also increased in size. The same sets of plots for the channel flow simulation using Crank-Nicolson and Runge-Kutta scheme are shown in Figures 4.39 and 4.40. These plots reveals similar flow characteristics with the increasing Reynolds number.

4.2.1 Unsteady Vortex Development at Re=800

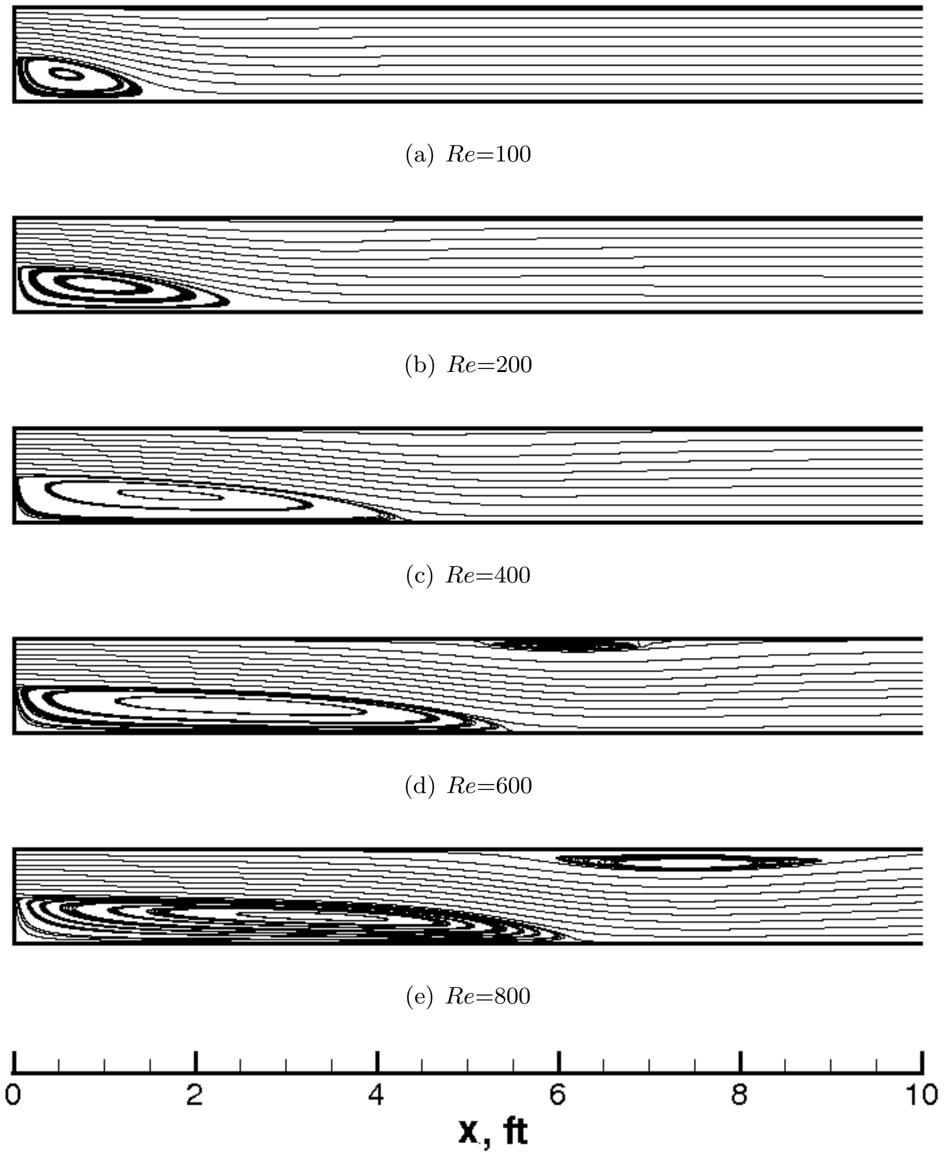
It has long been debated whether the flow for Reynolds number of 800 has a steady state. Gresho et al. found that for a sufficiently refined grid and simulation time, the numerical solution indeed reaches a steady state. This behavior is captured by the current results as shown in the streamline plots of the channel flow at different time instances.

The flow development for the Fully-Implicit scheme can be seen in Figures 4.41 through 4.43.

At $t = 1s$, the primary vortex forms just behind the step. After the initial development of the primary vortex, it moves downstream and a small eddy forms just behind it. As this primary vortex travels further downstream, its strength decreases and it progressively becomes smaller in size and eventually disappears. It is also observed that small eddies periodically form at the upper wall. The formation of the eddies occurs until about $t = 40s$ after which the flow starts to stabilize. From this point on, only two main vortices are present in the flow. This primary and secondary vortex grow in size until $t = 200s$ when the flow attains steady state.

A similar set of plots for the results using the Crank-Nicolson scheme is shown in Figure 4.44 and 4.45. At $t = 1s$, the formation of the primary vortex takes place behind the step. This vortex is convected downstream and a small eddy forms behind it. In addition to these vortices,a secondary eddy starts to form at the upper wall. As the core flow hits the bottom wall and gets deflected, a vortex develops at the bottom wall just downstream of the upper-wall eddy. In these plots, the shedding of the primary and secondary vortices continue until $t = 40s$. Subsequent plots indicate that the primary vortex grows in size whereas the secondary vortex decreases. This phenomenon continues till steady state is reached.

The results obtained for the Runge-Kutta time scheme are shown in Figures 4.47 and 4.48. Though the general nature of the flow development is consistent for the three schemes wherein they converge to an approximately steady state solution, the pattern of their transient development is slightly different. For this case, the Runge-Kutta and Crank-Nicolson are able to capture the transient behavior better than the Fully-Implicit scheme. This can be attributed to the fact that the first-order Fully-Implicit scheme is only first order accurate in time and is more dissipative.



(a) *Re*=100

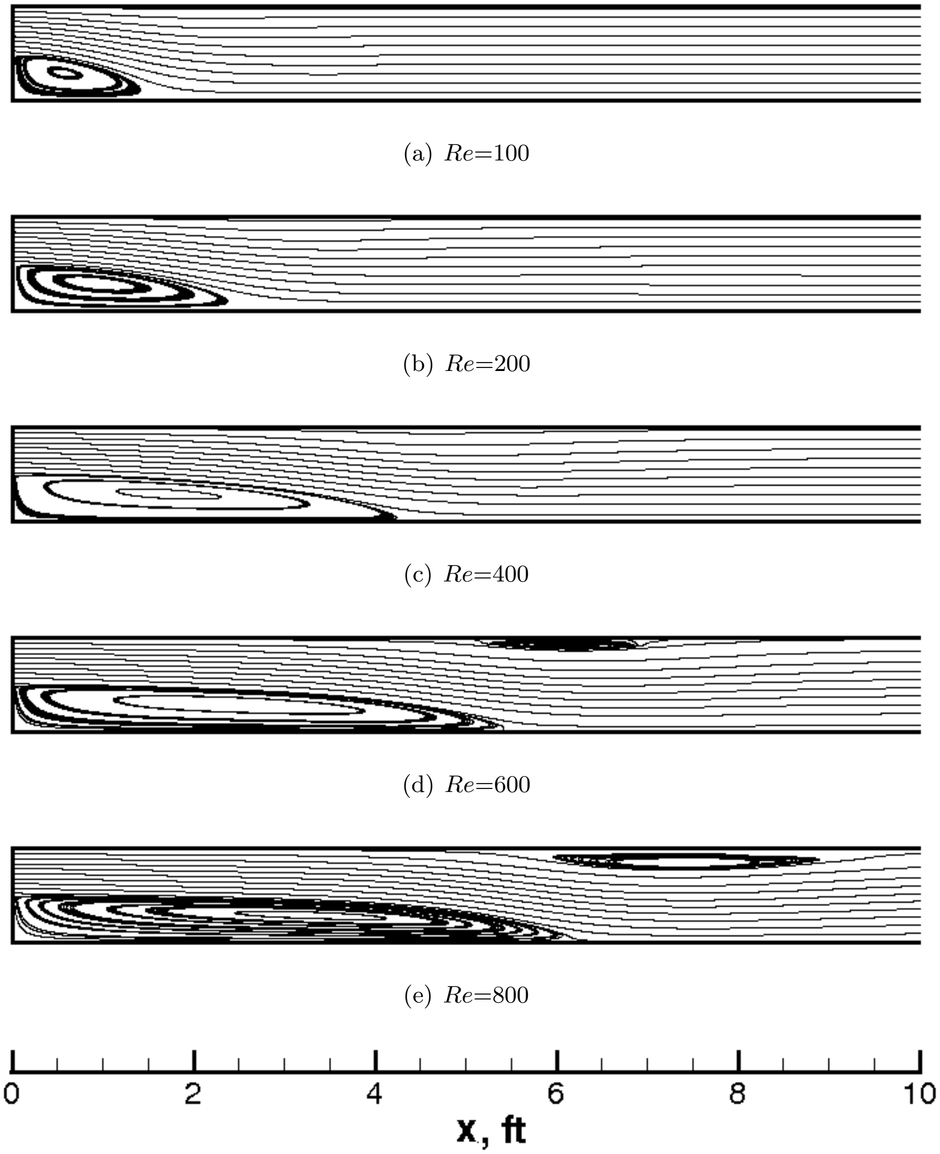
(b) *Re*=200

(c) *Re*=400

(d) *Re*=600

(e) *Re*=800

Figure 4.38 Streamline contour plot for *Re*=100, 200, 400, 600, and 800 using Fully-Implicit time integration.



(a) Re=100

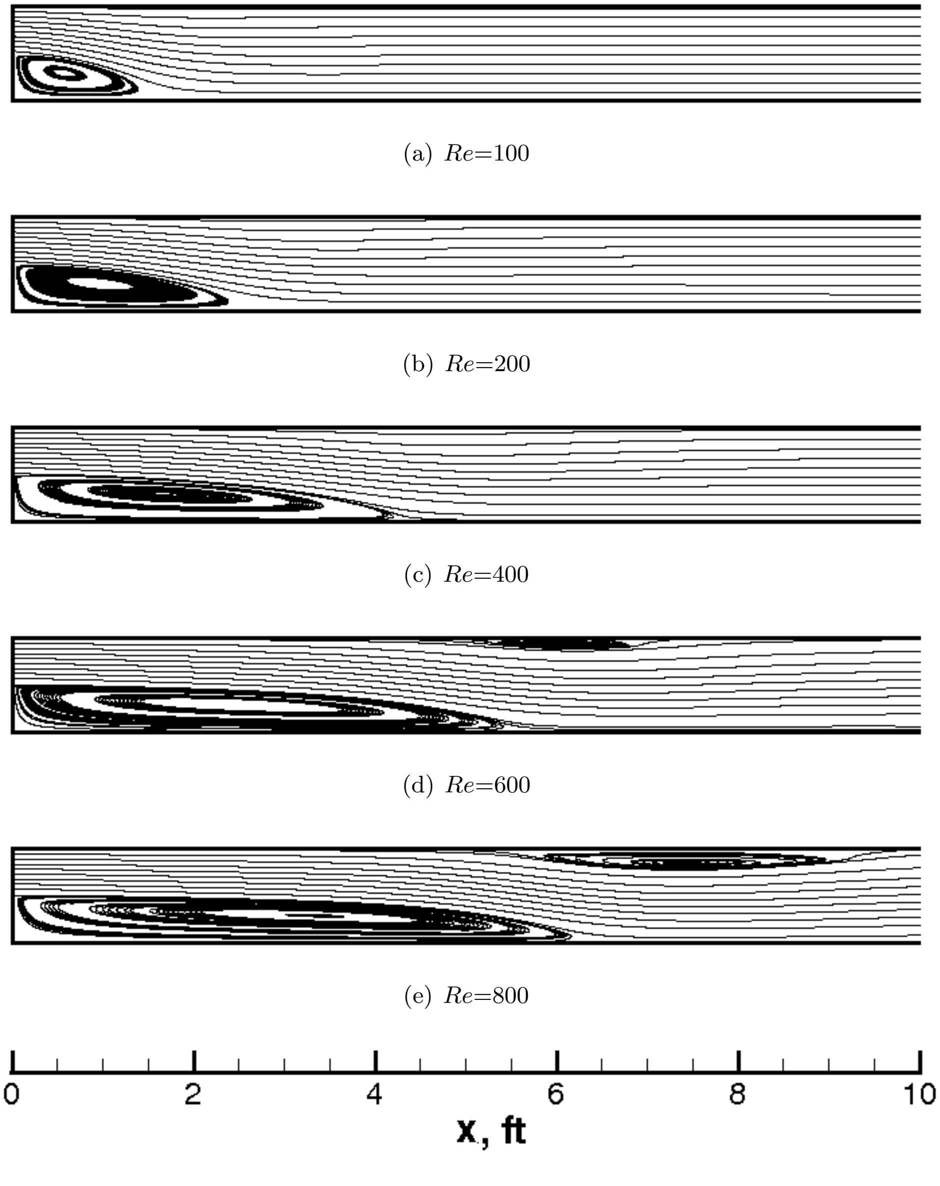
(b) Re=200

(c) Re=400

(d) Re=600

(e) Re=800

Figure 4.39 Streamline contour plot for Re=100, 200, 400, 600, and 800 using Crank-Nicolson time integration.



(a) Re=100

(b) Re=200

(c) Re=400

(d) Re=600

(e) Re=800

Figure 4.40 Streamline contour plot for Re=100, 200, 400, 600, and 800 using Runge-Kutta time integration.

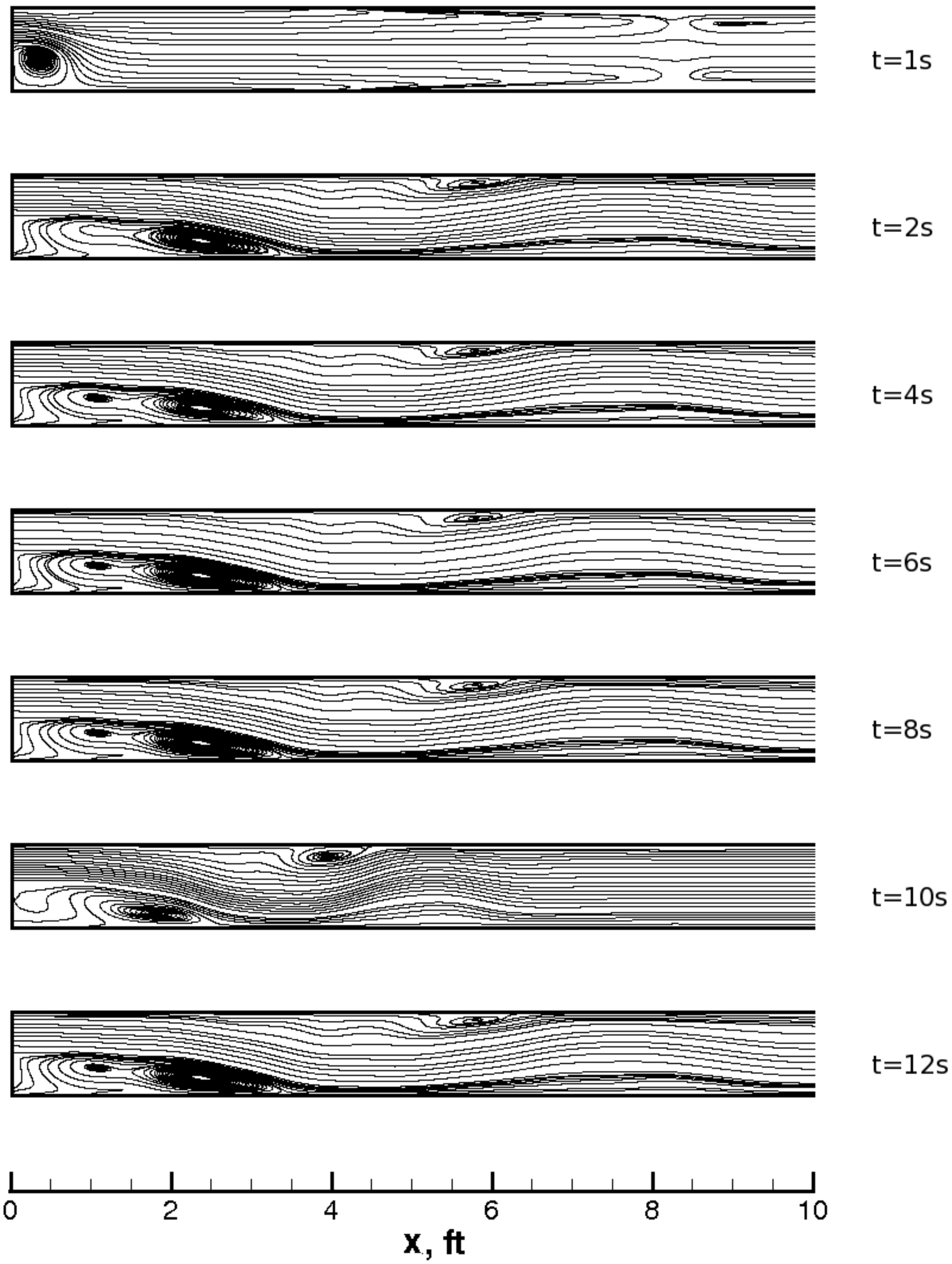


Figure 4.41 Unsteady development for *Re*=800 using Fully-Implicit time integration (part 1).

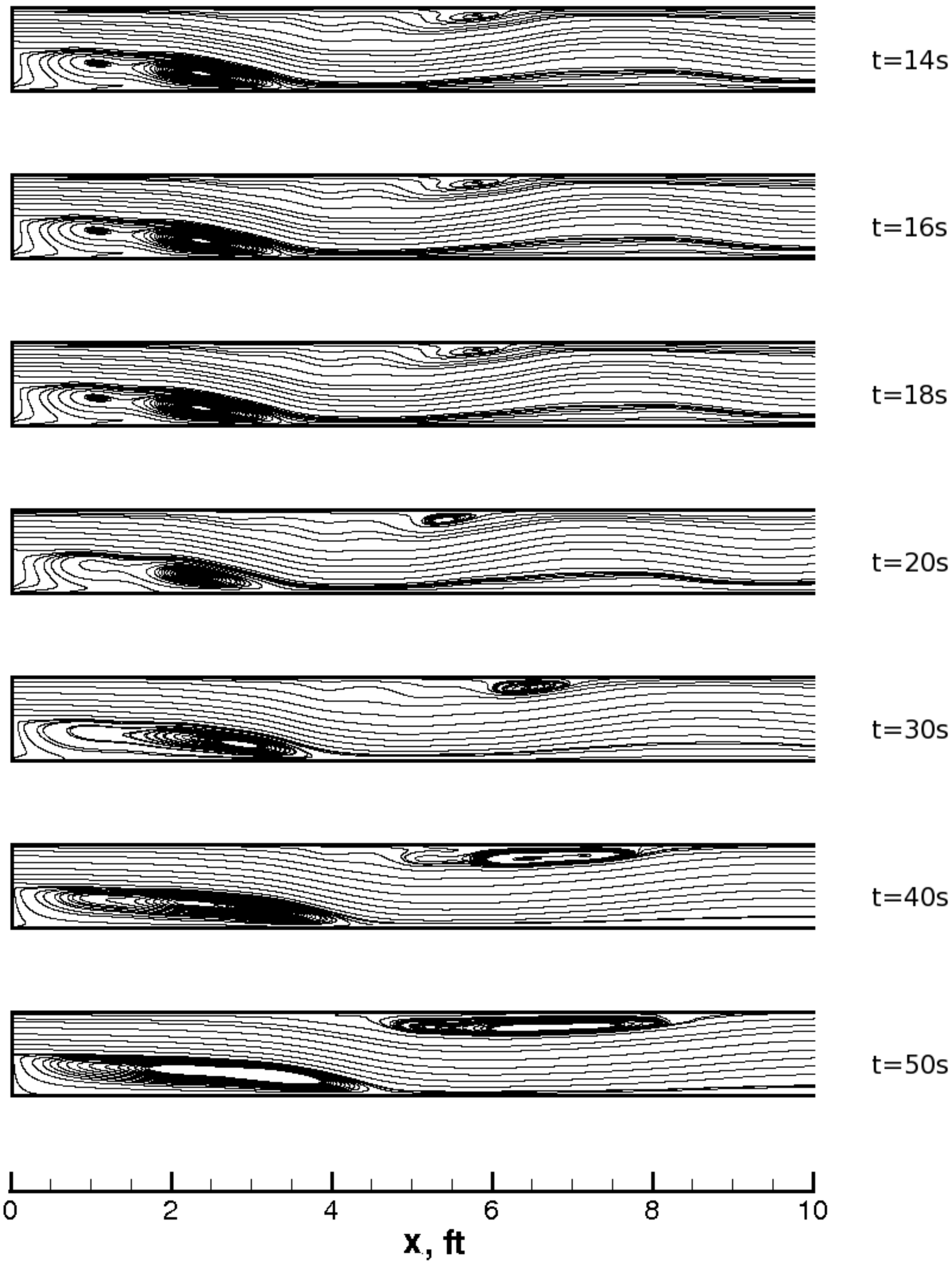


Figure 4.42 Unsteady development for Re=800 using Fully-Implicit time integration (part 2).

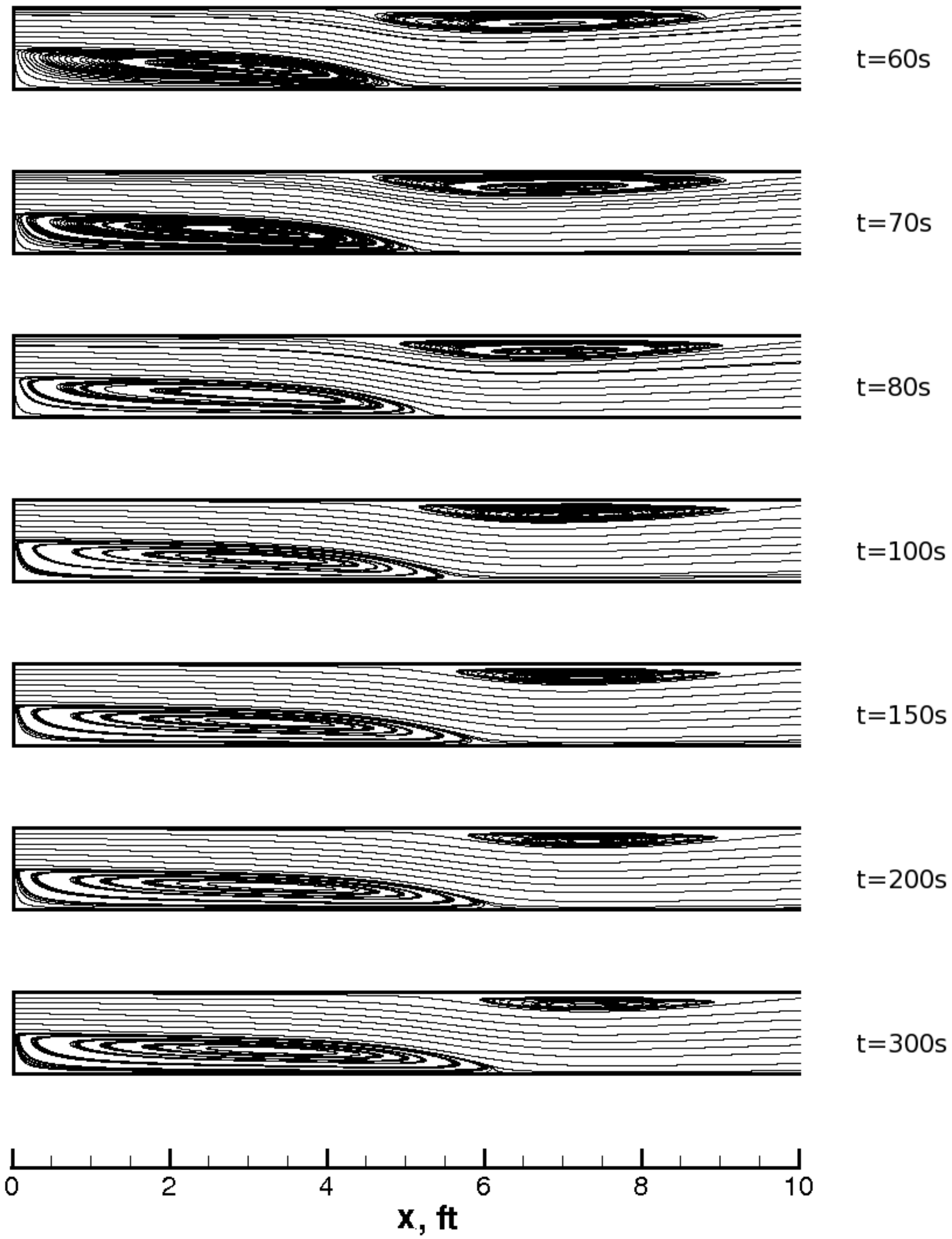


Figure 4.43 Unsteady development for *Re*=800 using Fully-Implicit time integration (part 3).

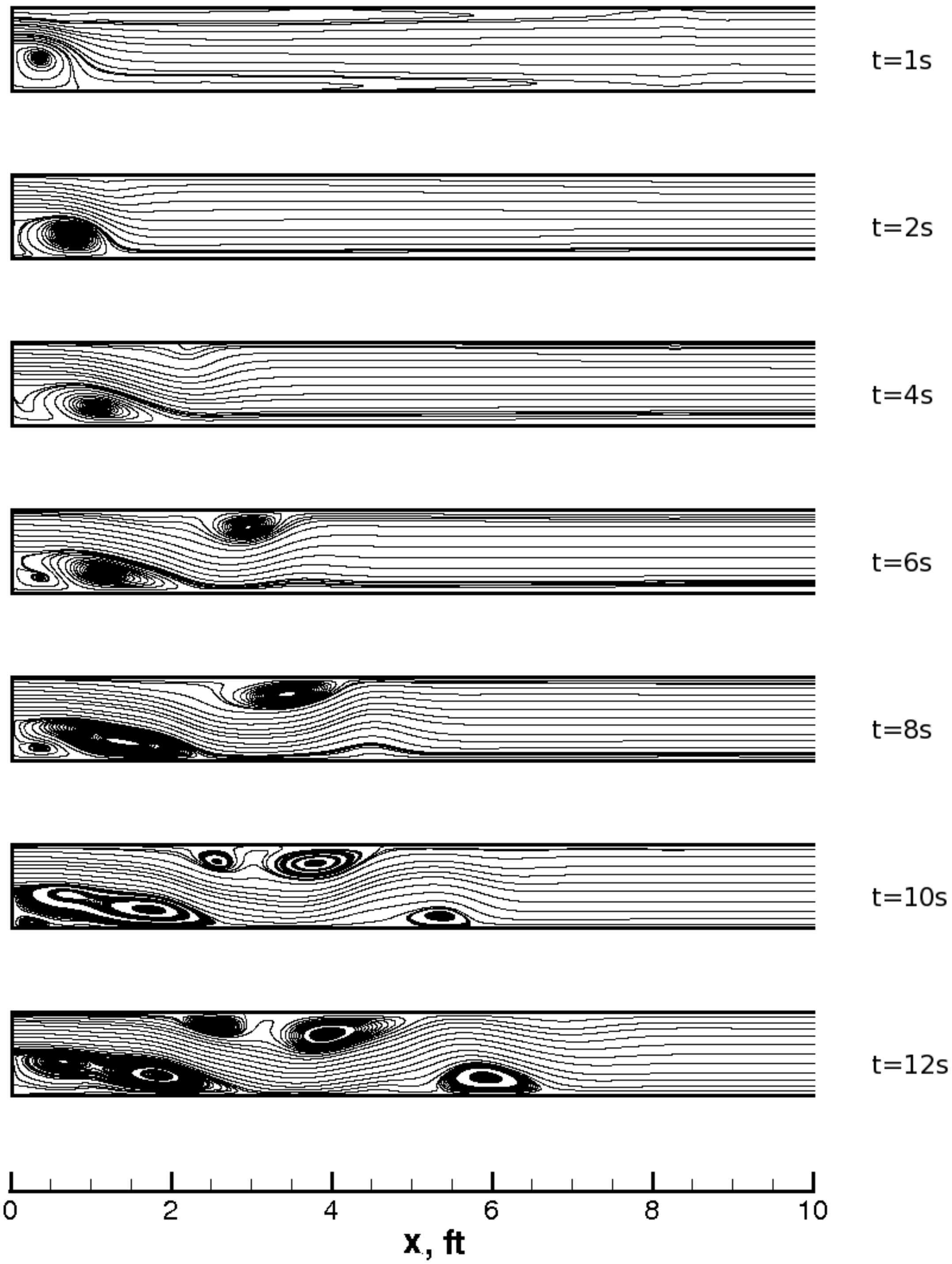


Figure 4.44 Unsteady development for *Re*=800 using Crank-Nicolson time integration (part 1).

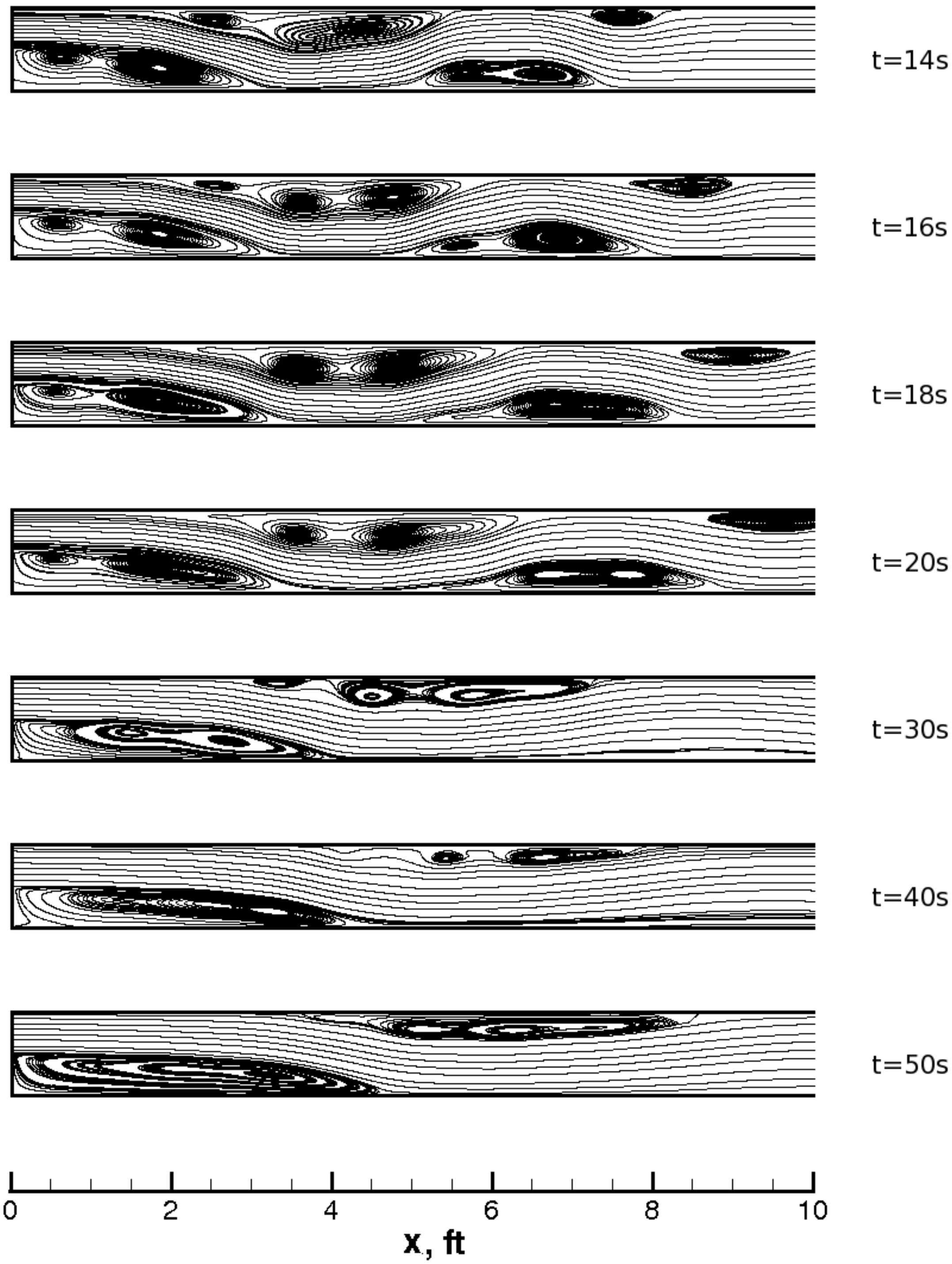


Figure 4.45 Unsteady development for *Re*=800 using Crank-Nicolson time integration (part 2).

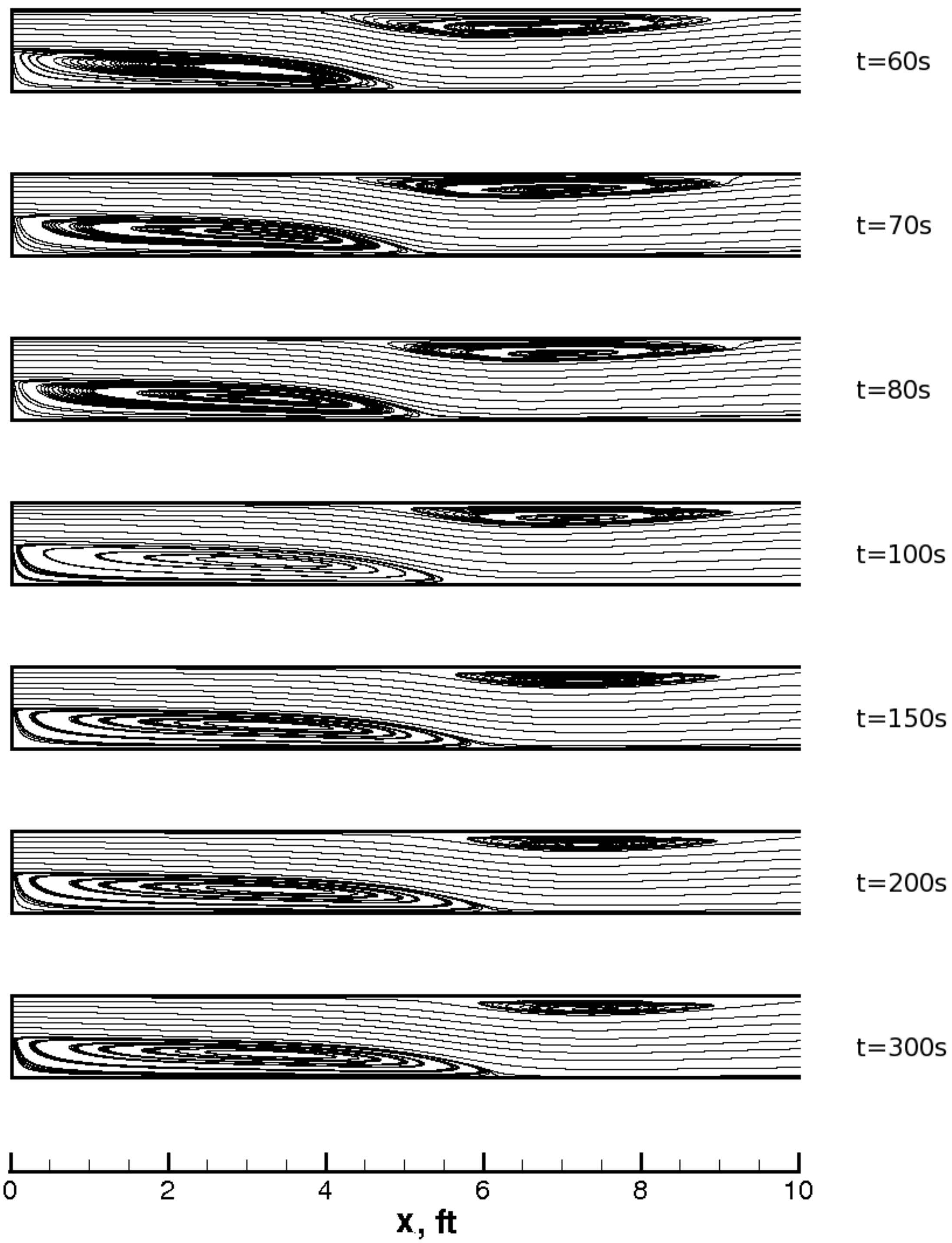


Figure 4.46 Unsteady development for Re=800 using Crank-Nicolson time integration (part 3).

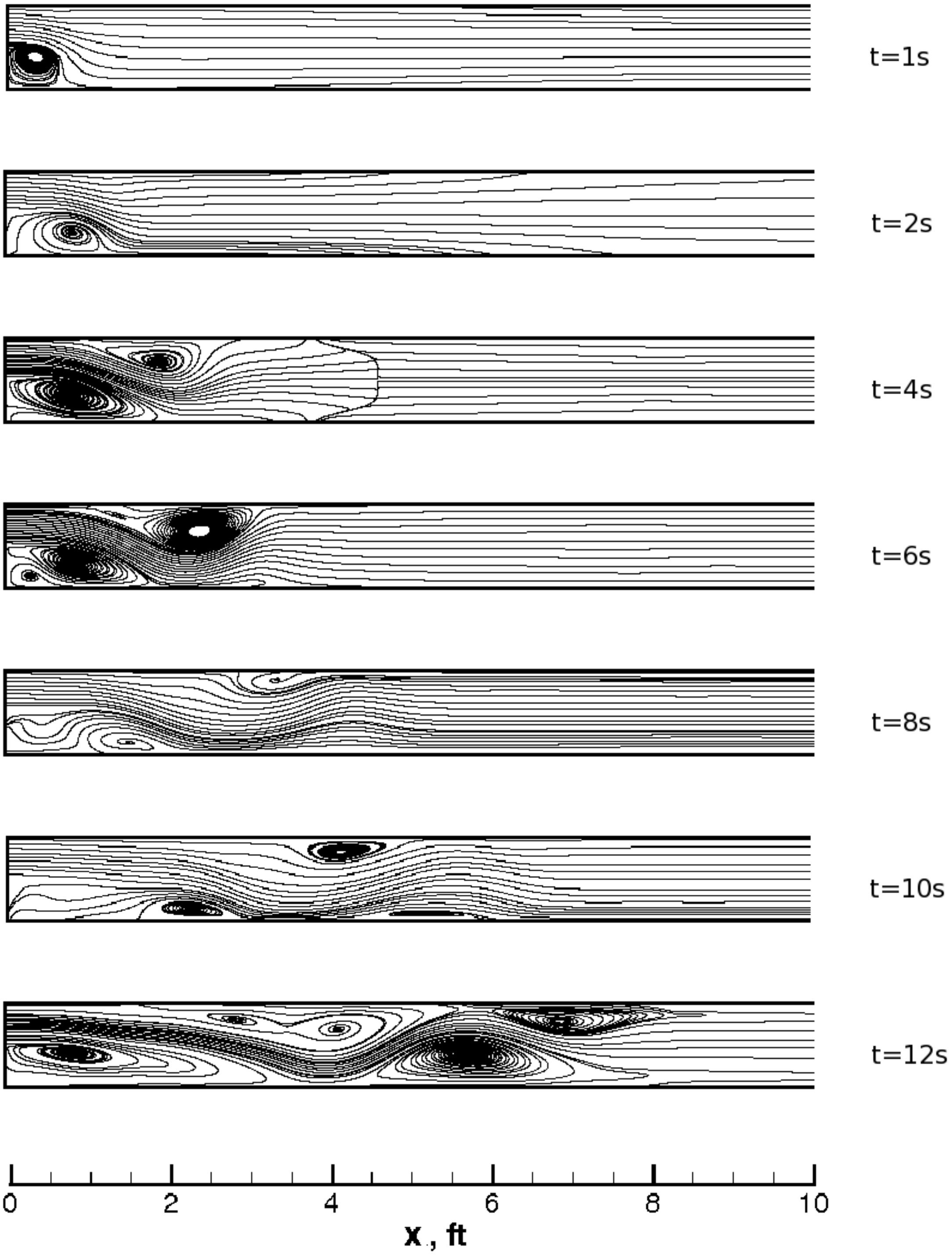


Figure 4.47 Unsteady development for *Re*=800 using Runge-Kutta time integration (part 1).

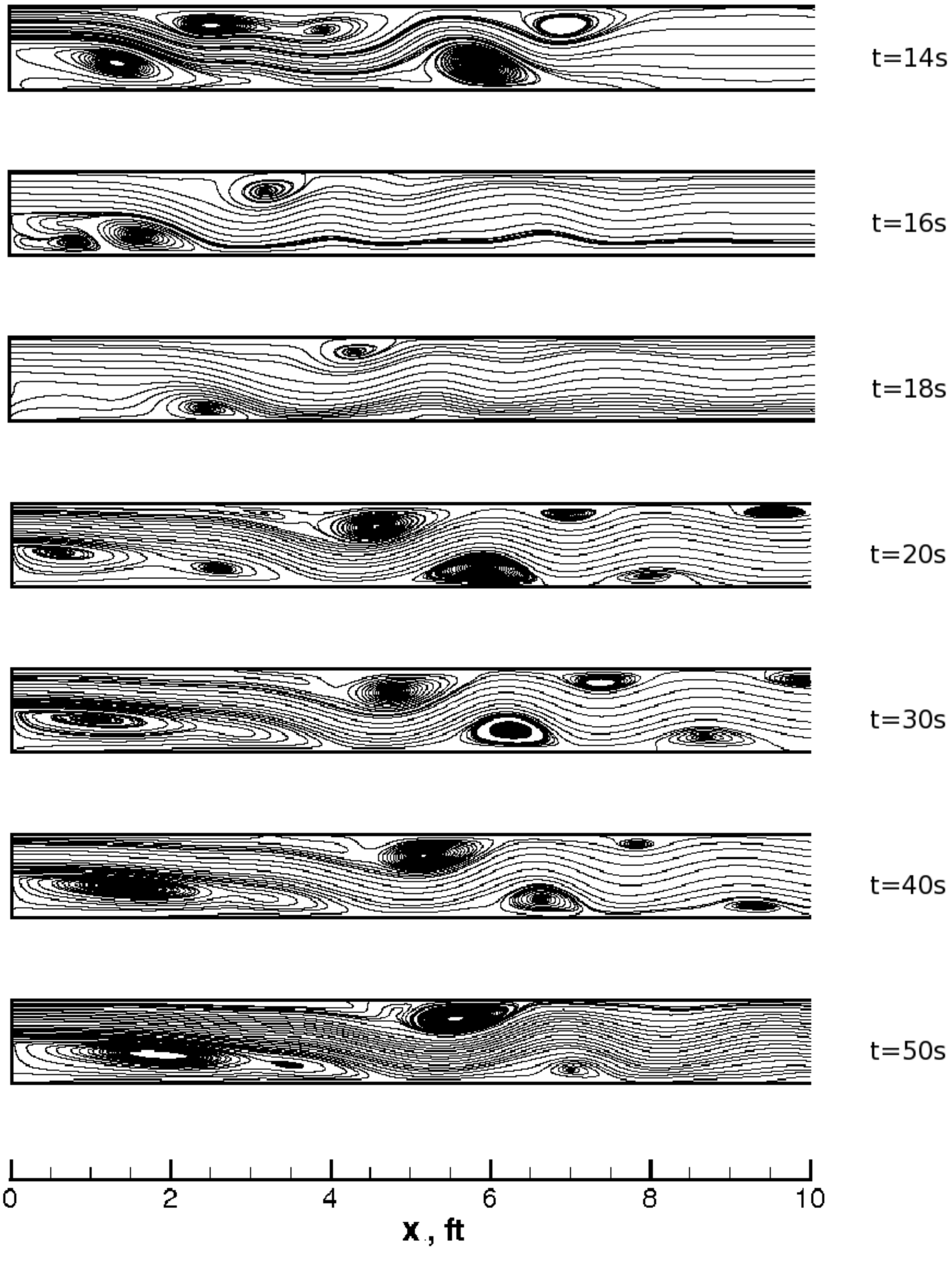


Figure 4.48 Unsteady development for Re=800 using Runge-Kutta time integration (part 2).

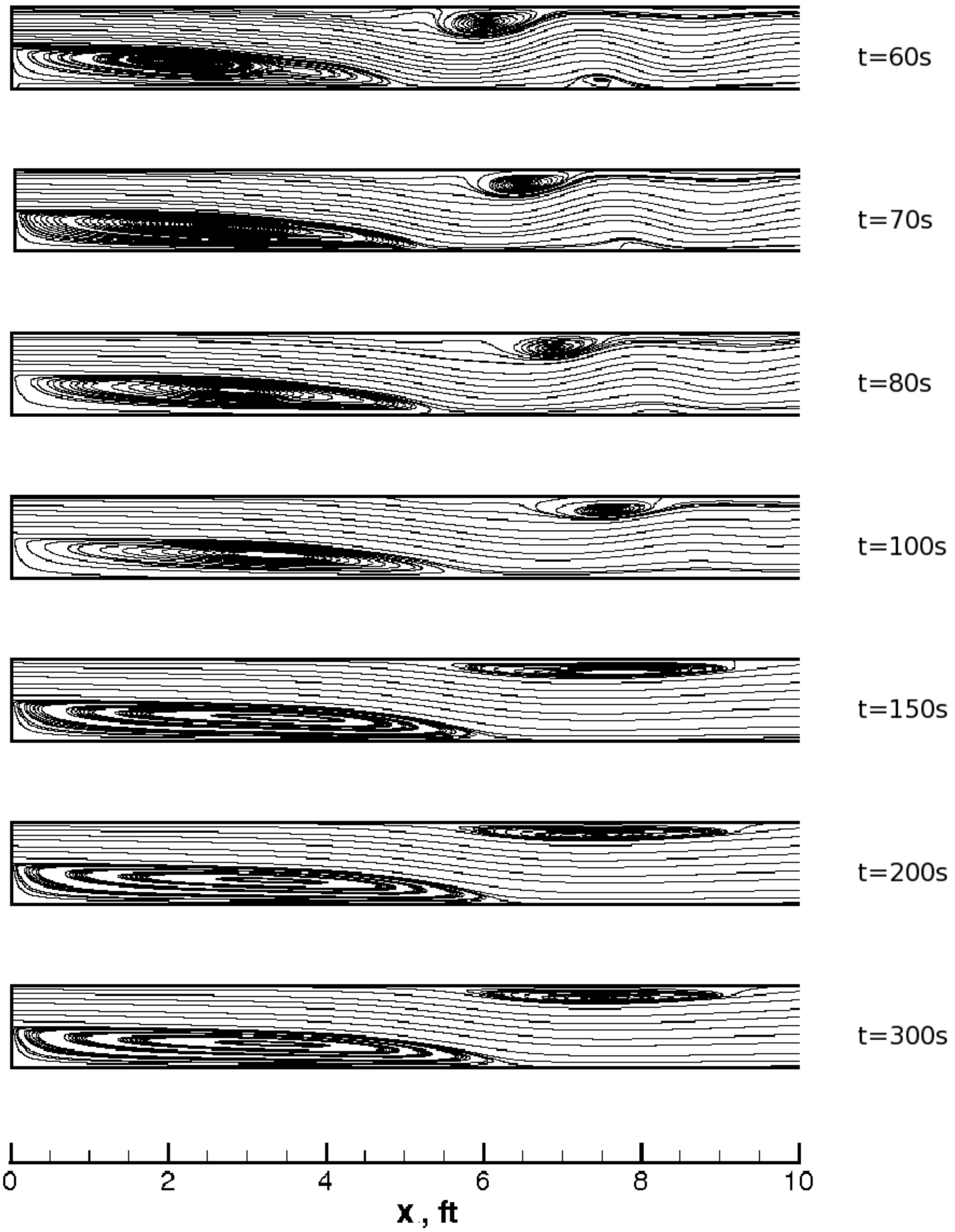


Figure 4.49 Unsteady development for *Re*=800 using Runge-Kutta time integration (part 3).

4.3 Flow Over a Vertical Flat Plate

The new algorithm is applied to simulate the unsteady flow over a flat plate at a 90 degree angle to the flow as shown in Figure 4.50. At the inlet boundary, a uniform flow U_∞ with a magnitude of 59.33 ft/s is assumed. The velocity at the outlet boundary is adjusted to satisfy the overall mass conservation. The Reynolds number of the flow is 17,800, where the Reynolds number for this flow is defined to be $Re = (U_\infty H)/\nu$. The height H of the plate is taken to be 0.03 ft and the thickness of the plate is 0.002 ft, which give an aspect ratio of 15. The The schematic of the flat plate configuration is presented in Figure 4.50. Another important parameter to be defined is the Strouhal number:

$$Sr = \frac{Hf}{U_\infty}$$

where f is the frequency of the shedding.

The computational grid used in the simulation has 16,504 nodes, 32,702 triangles and 49,206 faces, as shown in Figure 4.51. The grid near the plate is shown in Figure 4.52. The simulation is performed for 0.1 s of real time. The time step size was set to 10^-5 s for the simulations using the Fully-Implicit and the Crank-Nicolson schemes, while for the Runge-Kutta scheme a time step size of 10^-6 s was used.

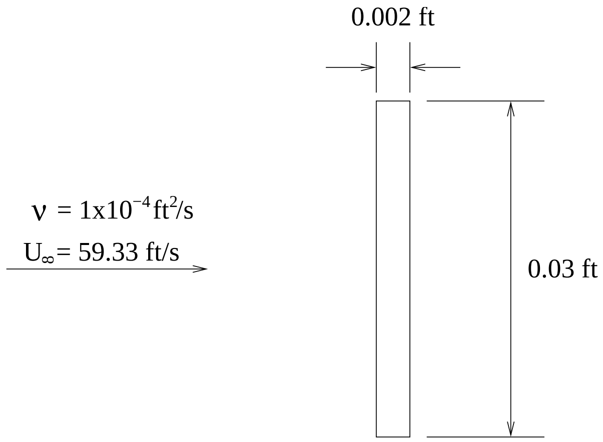


Figure 4.50 Schematic of flat plate at 90° to the flow.

The drag and lift coefficient history for the simulation using the Fully-Implicit scheme was plotted and is presented in Figure 4.53. For the given time step, the number of sub-iterations within each time step is varied until the change in the solution is negligible. Comparing the solution from the simulations using 10 sub-iterations and 30 sub-iterations in Figure 4.53(c), a discrepancy in the amplitude and an offset in the time axis are evident. When the number of sub-iterations is increased to 50, the solution matches reasonably well with that of 30 sub-iterations. This concludes that 30 sub-iterations would give a reasonable accuracy for the given problem. Similar plots for the simulation using Crank-Nicolson scheme are depicted in Figures 4.54 and 4.54(c). Similar behavior to the result from using Fully-Implicit scheme is observed here; thus; the same conclusion can be drawn. For the subsequent discussion, the results presented for both Fully-Implicit and Crank-Nicolson scheme are from the simulation using 50 sub-iterations.

A comparison of the drag and lift coefficient history using the Fully-Implicit, Crank-Nicolson and Runge-Kutta scheme is plotted in Figure 4.55. For all the schemes, the flow becomes periodic after about 0.02 second. The lift coefficient oscillates with a zero mean value. It is interesting to note that the drag oscillates at twice the frequency of the lift or the vortex shedding frequency. For the drag coefficient, one shedding cycle consists of two consecutive troughs and crests. In Figure 4.56, one cycle spans from point A to point E. The mean drag coefficient and the Strouhal number for the results of the flat plate is given in Table 4.1. There is a variation of about 4 percent observed for the mean drag coefficient, and about 8 percent for the Strouhal number. The variation in the drag amplitude and frequency might indicate that the time-step sizes taken for the current simulations are not small enough for the schemes to produce the same transient behavior.

Scheme	Mean Drag Coefficient (C_D)	Strouhal number (Sr)
Fully-Implicit	2.49	0.11926
Crank-Nicolson	2.52	0.11870
Runge-Kutta	2.55	0.12654

Table 4.1 Mean drag coefficient and Strouhal number comparison for the different schemes.

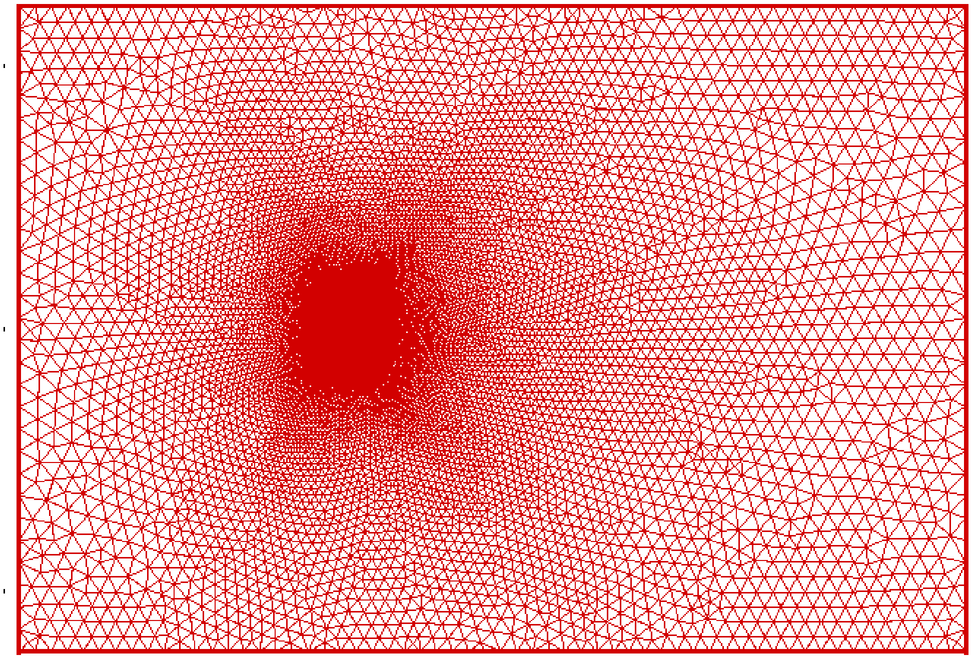


Figure 4.51 Computational grid for the flow over a vertical flat plate case.

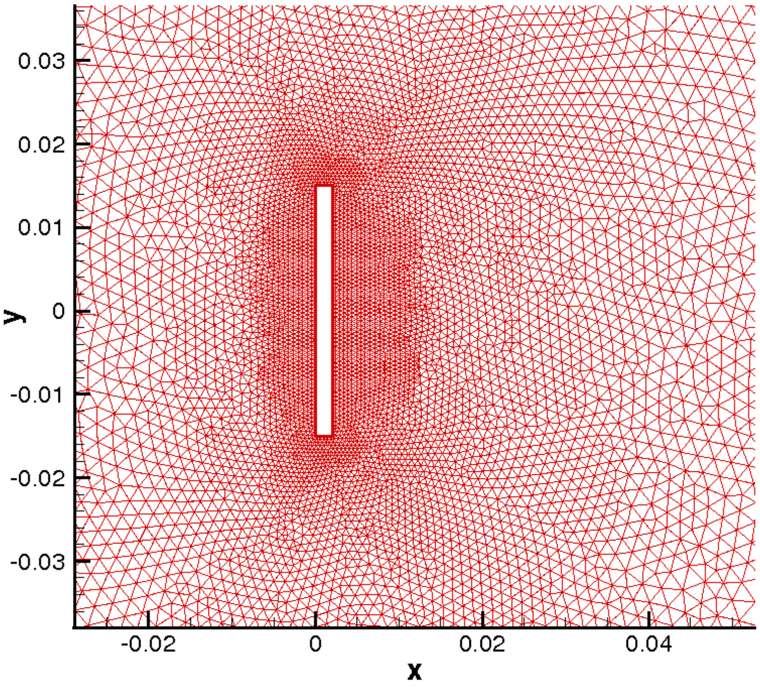
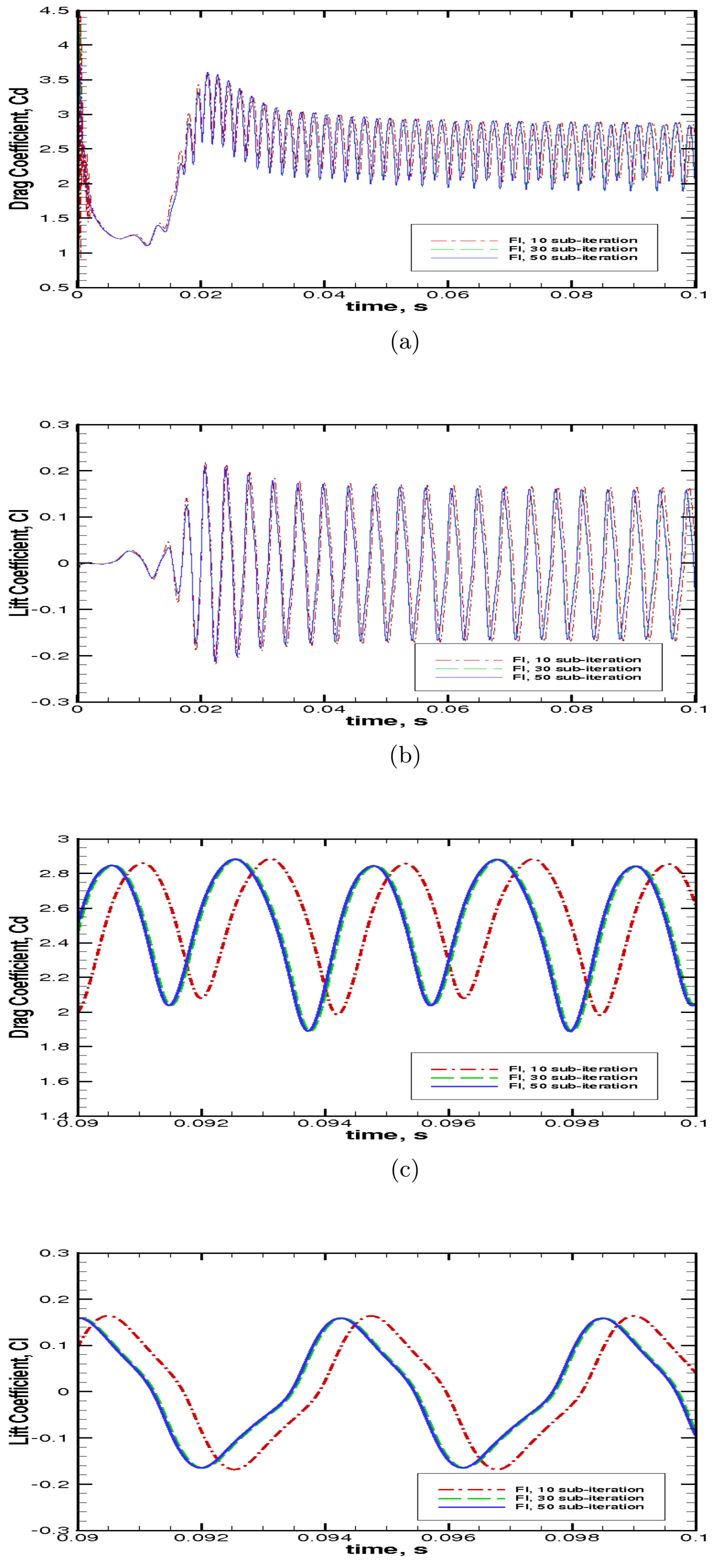


Figure 4.52 Grid around the flat plate.



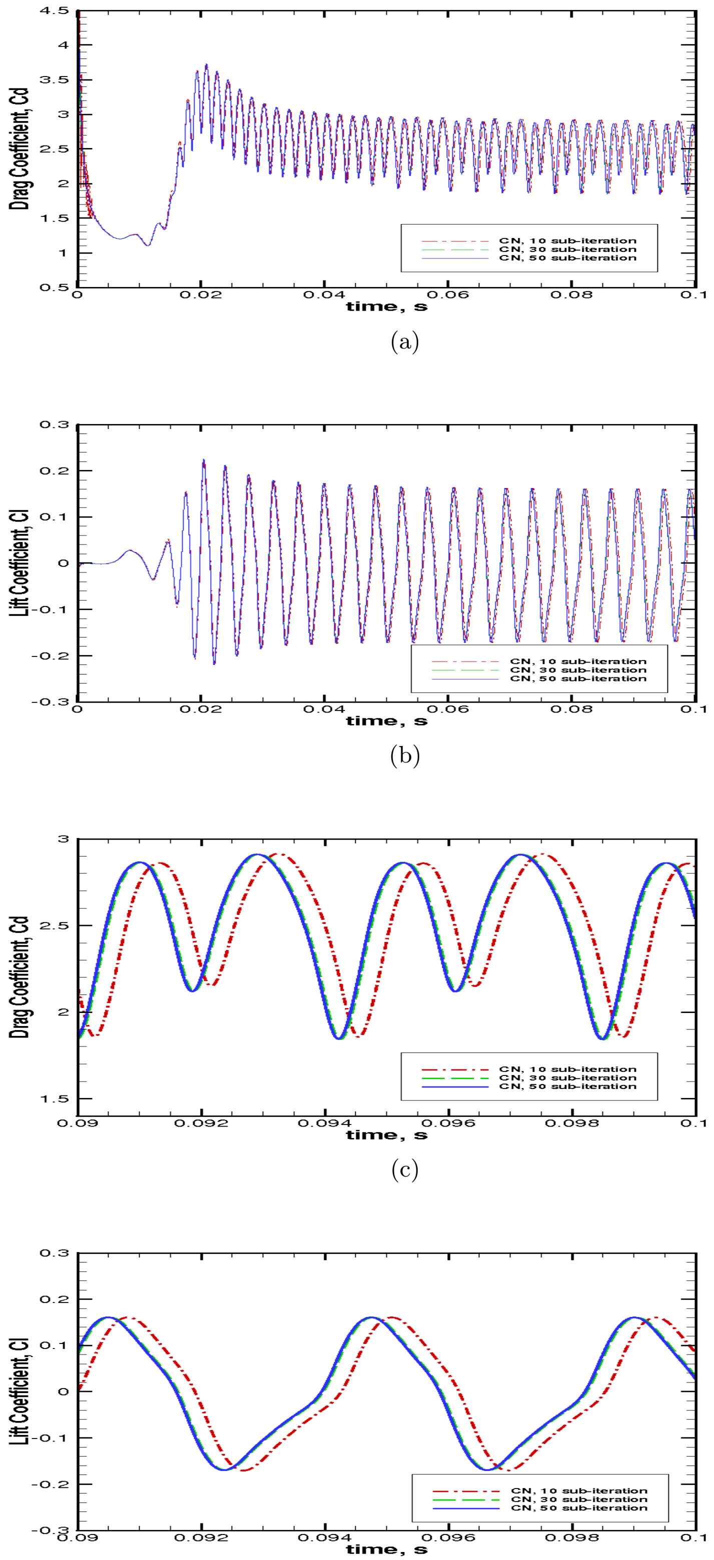
(a)

(b)

(c)

(d)

Figure 4.53 Drag and lift coefficient history using Fully-Implicit scheme.



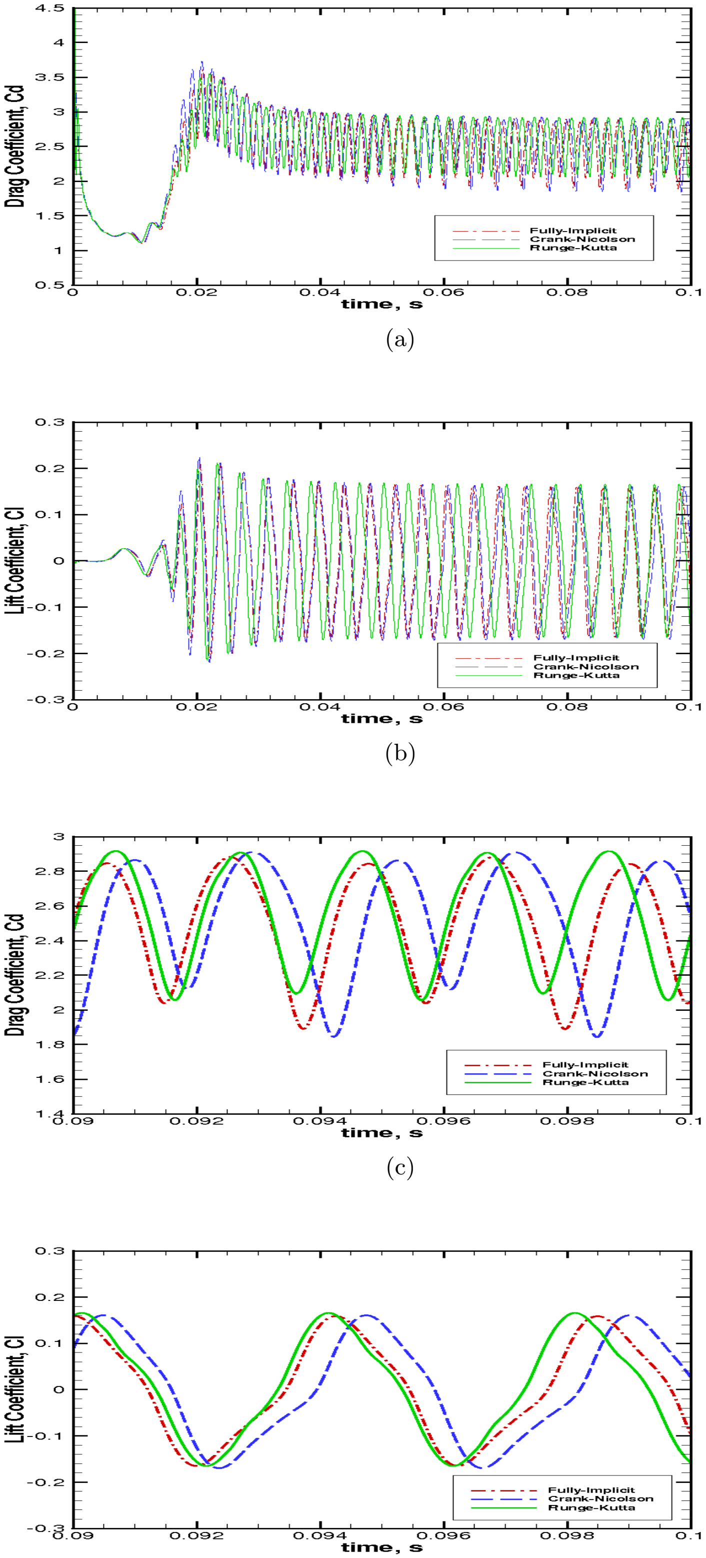
(a)

(b)

(c)

(d)

Figure 4.54 Drag and lift coefficient history using Crank-Nicolson scheme.



(a)

(b)

(c)

(d)

Figure 4.55 Comparison of the drag and lift coefficient history.

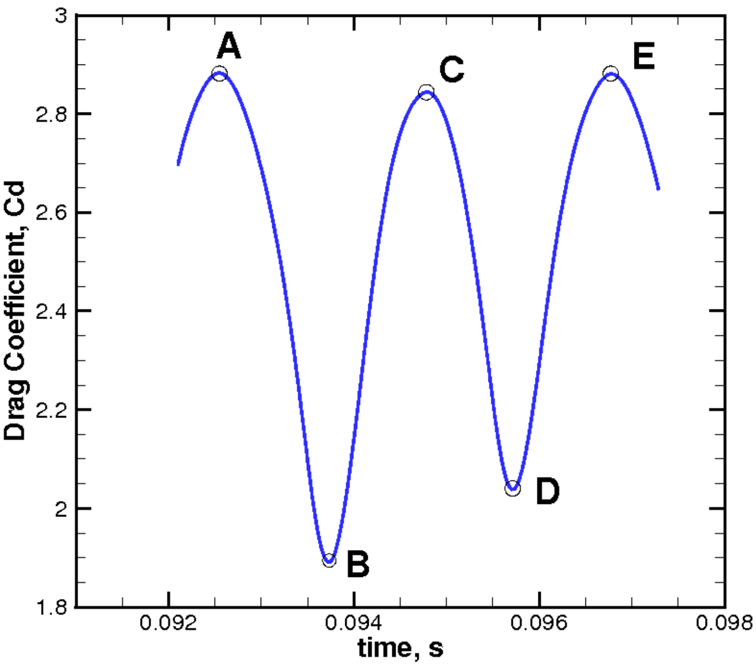
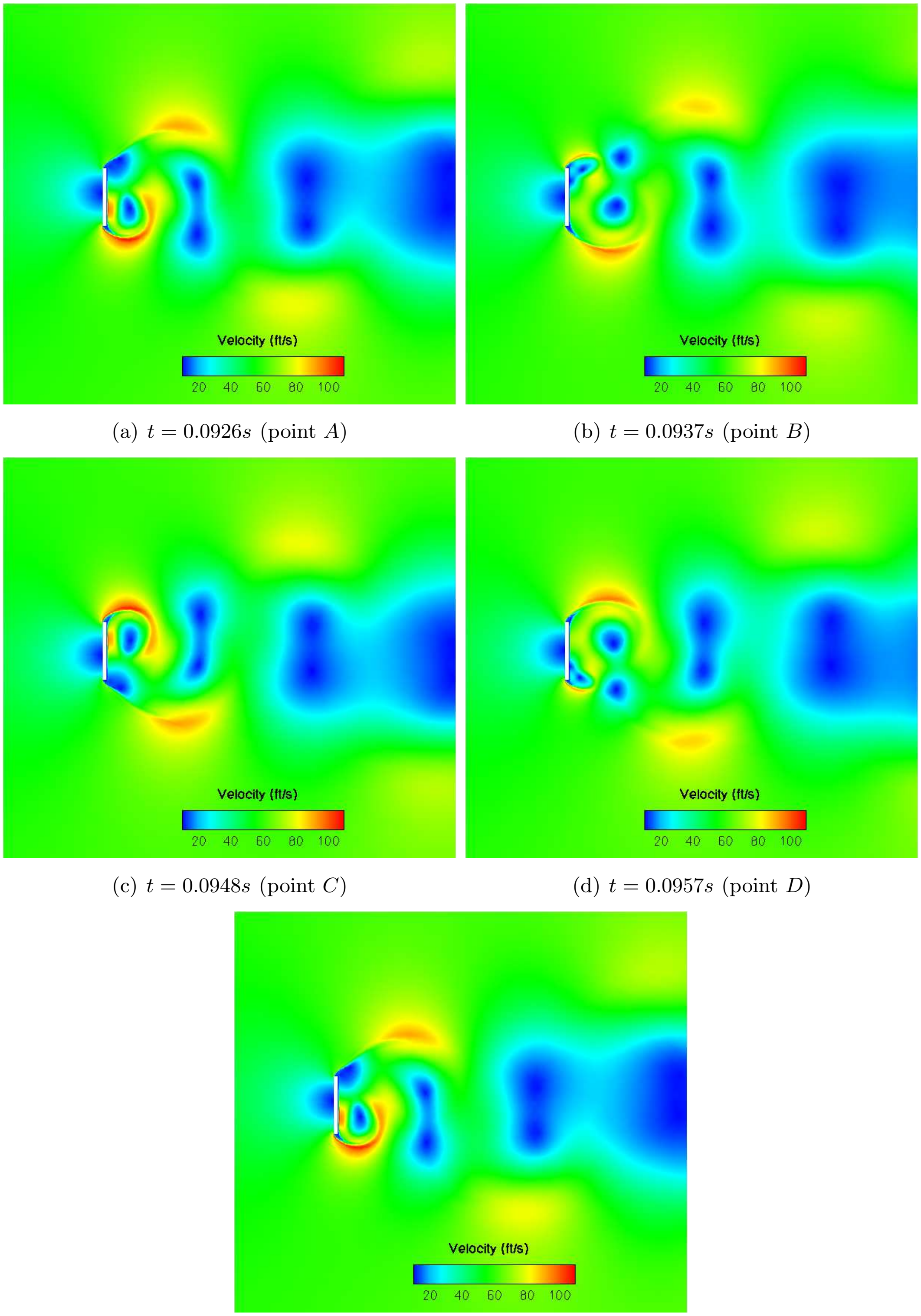


Figure 4.56 One pressure cycle.

The velocity and pressure field at points A through point E is shown in Figures 4.57 and 4.58. From Figure 4.57, it can be seen that at point A and B, the vortex shedding occurs at the bottom of the plate, while at point C and D the vortex shedding occurs at the top. In Figure 4.58 corresponding to point A, a low pressure region is observed just behind the plate, corresponding to a peak in the drag coefficient. At point B, the low pressure region has detached from the plate and move downstream, resulting in a drop in the drag values. At point C and D, the same phenomenon is observed with the vortex shedding from the top of the plate. This pressure cycle is repetitive as observed in the figures where point E represents the same state as point A. The same sets of plots for Crank-Nicolson and Runge-Kutta schemes are shown in Figures 4.59 through 4.62. From these figures, it is seen that all the schemes show similar behavior.



(a) $t = 0.0926s$ (point A)

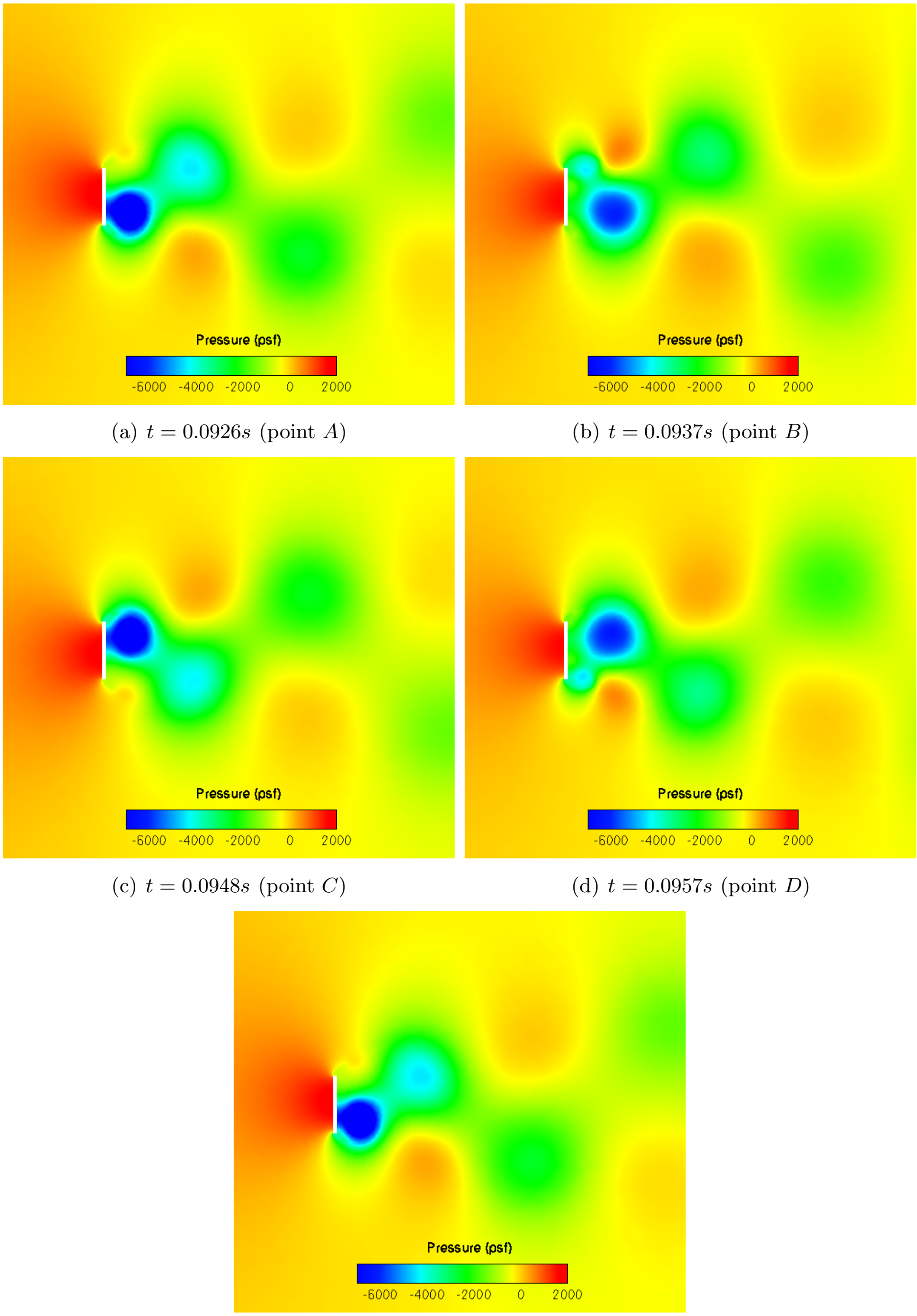
(b) $t = 0.0937s$ (point B)

(c) $t = 0.0948s$ (point C)

(d) $t = 0.0957s$ (point D)

(e) $t = 0.0968s$ (point E)

Figure 4.57 The velocity field for one pressure cycle (Fully-Implicit).



(a) $t = 0.0926s$ (point A)

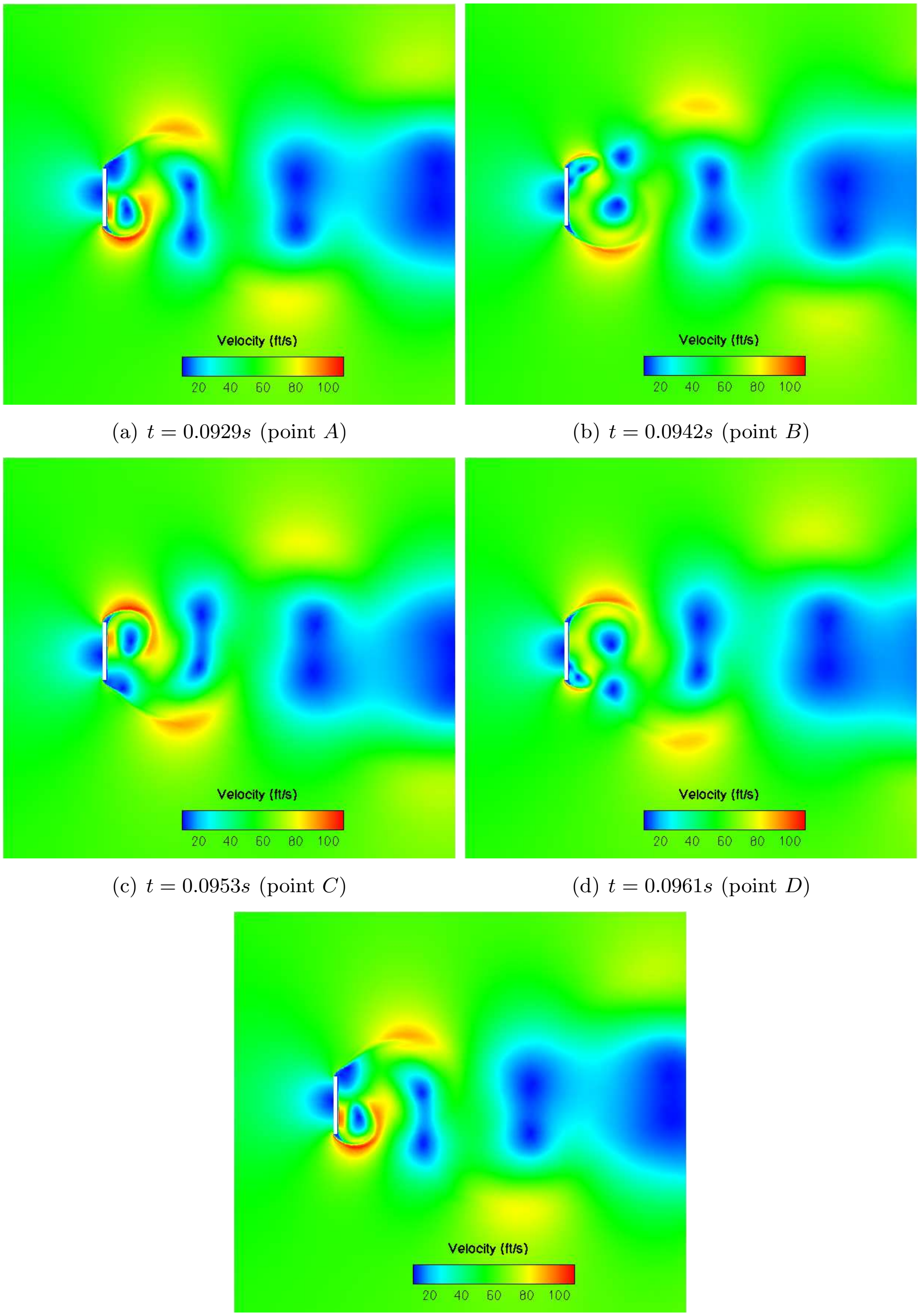
(b) $t = 0.0937s$ (point B)

(c) $t = 0.0948s$ (point C)

(d) $t = 0.0957s$ (point D)

(e) $t = 0.0968s$ (point E)

Figure 4.58 The pressure field for one pressure cycle (Fully-Implicit).



(a) $t = 0.0929s$ (point A)

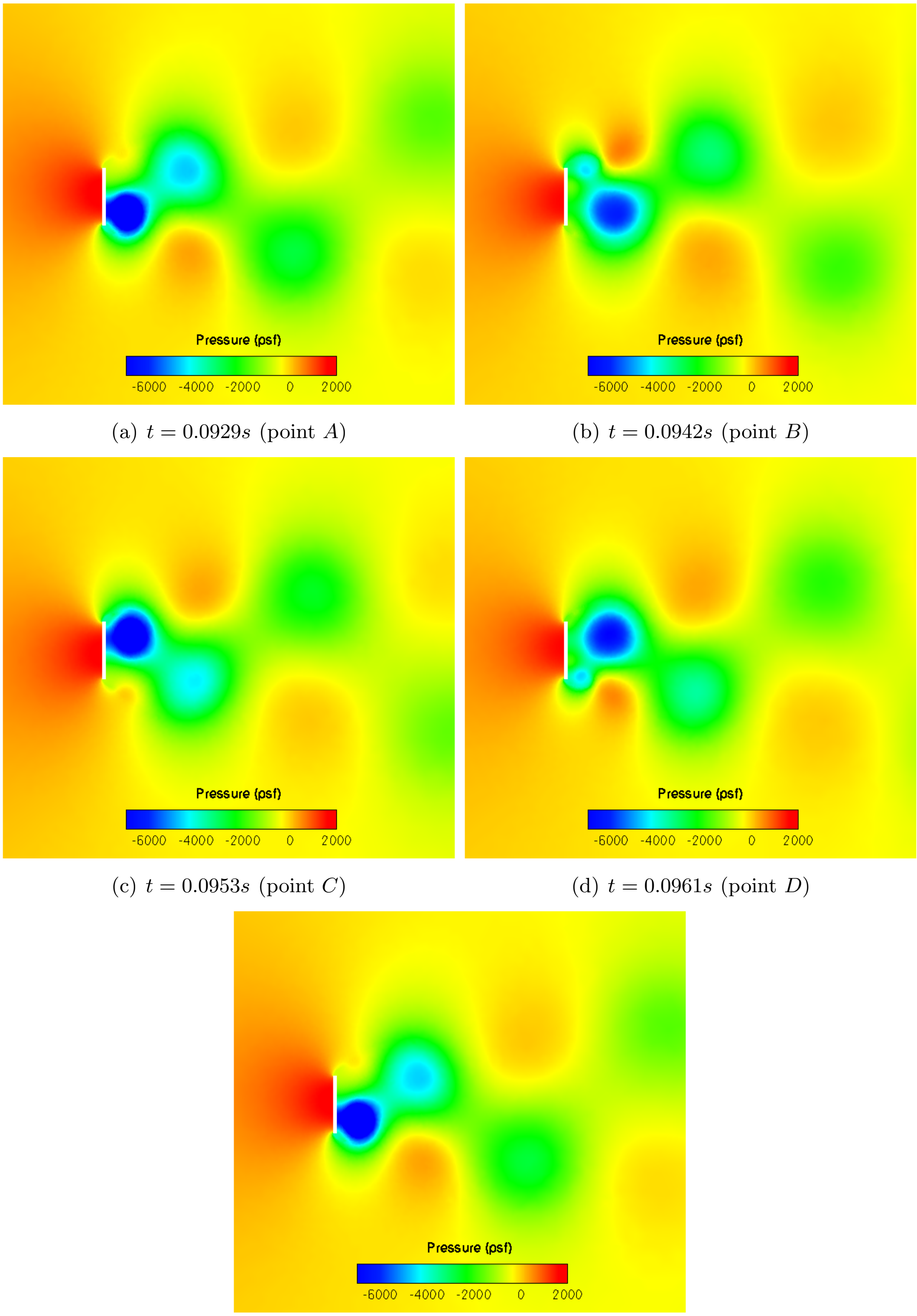
(b) $t = 0.0942s$ (point B)

(c) $t = 0.0953s$ (point C)

(d) $t = 0.0961s$ (point D)

(e) $t = 0.0927s$ (point E)

Figure 4.59 The velocity field for one pressure cycle (Crank-Nicolson).



(a) $t = 0.0929s$ (point A)

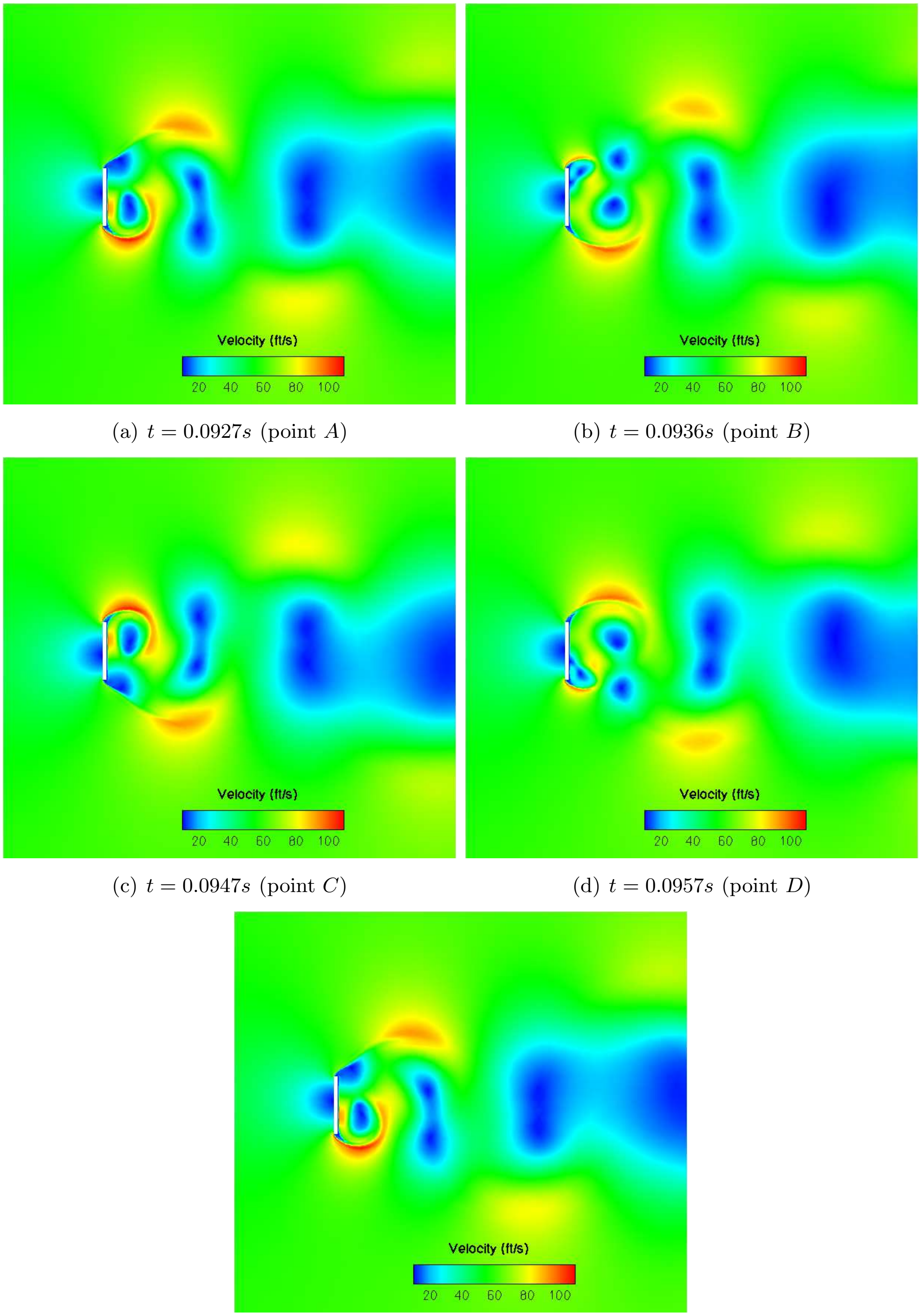
(b) $t = 0.0942s$ (point B)

(c) $t = 0.0953s$ (point C)

(d) $t = 0.0961s$ (point D)

(e) $t = 0.0927s$ (point E)

Figure 4.60 The pressure field for one pressure cycle (Crank-Nicolson).



(a) $t = 0.0927s$ (point A)

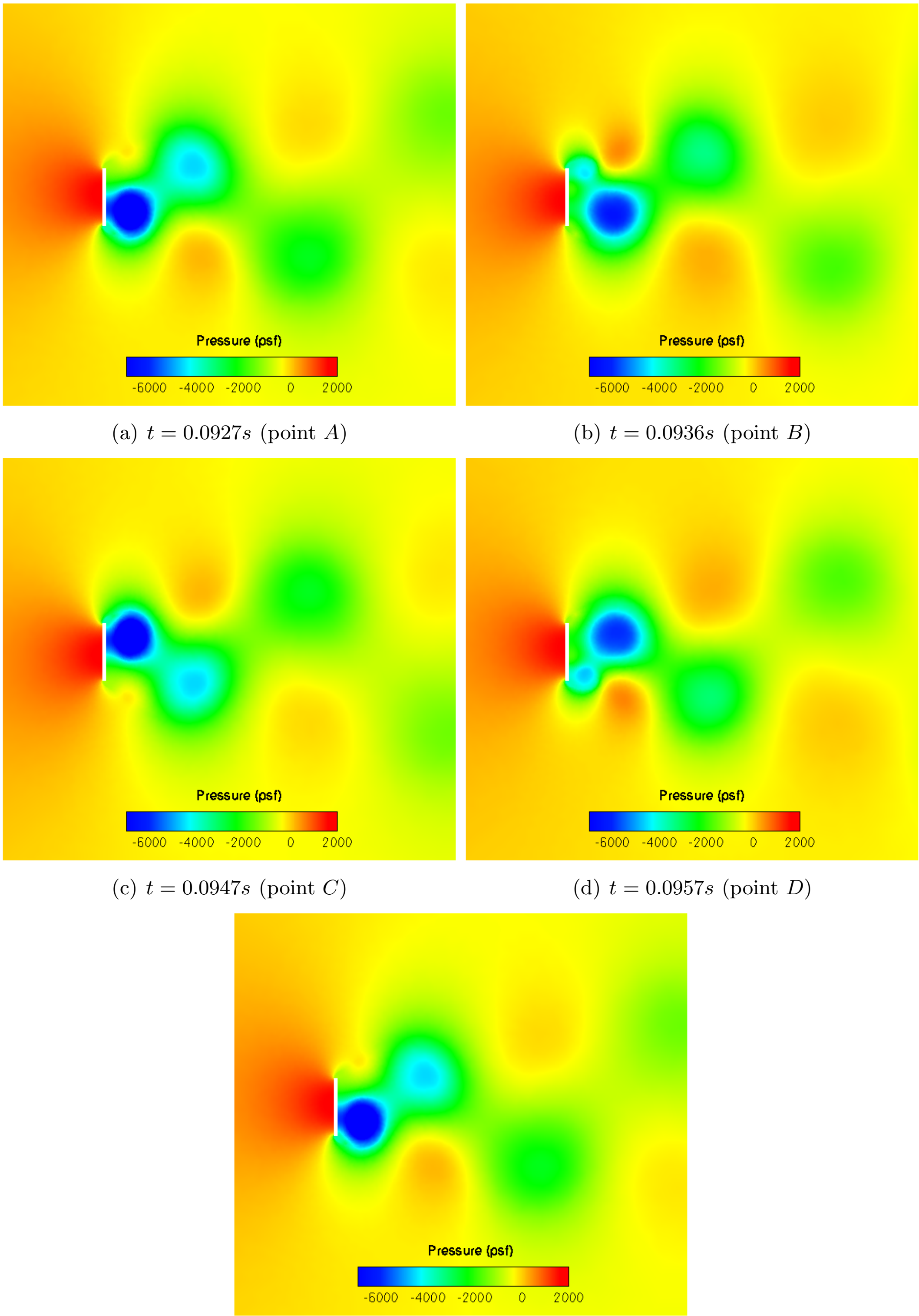
(b) $t = 0.0936s$ (point B)

(c) $t = 0.0947s$ (point C)

(d) $t = 0.0957s$ (point D)

(e) $t = 0.0967s$ (point E)

Figure 4.61 The velocity field for one pressure cycle (Runge-Kutta).



(a) $t = 0.0927s$ (point A)

(b) $t = 0.0936s$ (point B)

(c) $t = 0.0947s$ (point C)

(d) $t = 0.0957s$ (point D)

(e) $t = 0.0967s$ (point E)

Figure 4.62 The pressure field for one pressure cycle (Runge-Kutta).

CHAPTER 5. CONCLUDING REMARKS AND RECOMMENDATIONS

In the current research, the unsteady algorithms for incompressible Navier-Stokes flow on triangular unstructured grids have been developed and implemented. The time integration method uses a Fully-Implicit, Crank-Nicolson and an explicit four-step Runge-Kutta method. A vertex-centered discretization with median-dual control volume as proposed by Baliga is used. Also, the equal order velocity pressure interpolation method is chosen to avoid the checkerboard pressure oscillation commonly encountered in using a collocated grid for solving incompressible flows. The numerical algorithm used to solve the resulting equations is derived from the SIMPLER algorithm. The Fully-Implicit and Crank-Nicolson follow the traditional path of a pressure correction equation to update the velocities. The Runge-Kutta SIMPLER uses the four-stage Runge-Kutta to update the velocities directly without a pressure correction equation.

The resulting solver has been validated using the standard lid driven cavity problem and the backward facing step channel. A good agreement with existing experimental data is obtained for all cases. The algorithms are also capable of capturing the unsteady or transient behavior in vortex development such as observed in the step channel flow. The unsteady algorithms have been applied to simulate the unsteady flow over a vertical flat plate. It was found that Runge-Kutta scheme implemented in the current research displays general behaviors of an explicit scheme, such as sensitivity to the grid density and limitation on the allowable time step size. The Runge-Kutta scheme is also observed to perform well on low Reynolds number cases, but it becomes more unstable at higher Reynolds number, with the necessity of a finer grid density and lower time step. This is expected as the ODEs become stiff.

Although the results are promising, further efforts are necessary to make the algorithm robust and more general. A time comparison study of the implicit Crank-Nicolson and explicit Runge-Kutta scheme to obtain the difference and saving in execution time needs to be undertaken. Extension of the time-accurate algorithm to general arbitrary control volumes and three-dimensional flows would further enhance its usefulness.

APPENDIX A. Derivation of Coefficients of Momentum and Pressure Equations

Coefficients Of The Shape Function for General Variable ϕ

The shape function of a general variable ϕ as discussed in Chapter 2 is as follows:

$$\phi = AZ + BY + C \tag{A.1}$$

where Z is an exponential function in the local coordinate X and defined as:

$$Z = \frac{X - X_{max}}{Pe + [\![0, (1 - 0.1\,|Pe|)^5]\!]} \tag{A.2}$$

Using the values of the variable at the element vertices (Figure 2.4) as the boundary condition for solving the shape function coefficients, we have:

$$AZ_1 + BY_1 + C = \phi_1 \tag{A.3}$$

$$AZ_2 + BY_2 + C = \phi_2 \tag{A.4}$$

$$AZ_3 + BY_3 + C = \phi_3 \tag{A.5}$$

The above equations can be written in the matrix form, and solved using Cramer's rule for the coefficients A, B, and C to get:

$$\begin{bmatrix} Z_1 & Y_1 & 1 \\ Z_2 & Y_2 & 1 \\ Z_3 & Y_3 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \tag{A.6}$$

$$\begin{array}{lll} L_1 = \frac{(Y_2 - Y_3)}{\Delta}, & L_2 = \frac{(Y_3 - Y_1)}{\Delta}, & L_3 = \frac{(Y_1 - Y_2)}{\Delta}, \\ M_1 = \frac{(Z_3 - Z_2)}{\Delta}, & M_2 = \frac{(Z_1 - Z_3)}{\Delta}, & M_3 = \frac{(Z_2 - Z_1)}{\Delta}, \\ N_1 = \frac{(Z_2 Y_3 - Z_3 Y_2)}{\Delta}, & N_2 = \frac{(Z_3 Y_1 - Z_1 Y_3)}{\Delta}, & N_3 = \frac{(Z_1 Y_2 - Z_2 Y_1)}{\Delta} \end{array} \tag{A.7}$$

where the determinant Δ is:

$$\Delta = Z_1 (Y_2 - Y_3) + Z_2 (Y_3 - Y_1) + Z_3 (Y_1 - Y_2) \tag{A.8}$$

The coefficients can then be written as an interpolation function of the values of ϕ at the vertices:

$$A = L_1\phi_1 + L_2\phi_2 + L_3\phi_3 \tag{A.9}$$

$$B = M_1\phi_1 + M_2\phi_2 + M_3\phi_3 \tag{A.10}$$

$$C = N_1\phi_1 + N_2\phi_2 + N_3\phi_3 \tag{A.11}$$

Discretization Of The Flux Terms

The flux of the general variable ϕ expressed in terms of the total flux vector $\vec{J}$ is:

$$\vec{J} = \left(\rho U \phi - \mu \frac{\partial \phi}{\partial X}\right) \hat{\imath} + \left(\rho V \phi - \mu \frac{\partial \phi}{\partial Y}\right) \hat{\jmath} \tag{A.12}$$

Given the definition of ϕ derived through the shape function, the X and Y components of the flux vector are:

$$J_X = (\rho f_i - \mu L_i)\, \phi_i \tag{A.13}$$

$$J_Y = (\rho g_i - \mu M_i)\, \phi_i \tag{A.14}$$

where summation is implied over the neighboring grid points.

Expanding Equations A.13 and A.14, we obtain:

$$\begin{aligned} J_X &= \{\rho\left[\left(U - U_{avg}\right) L_1 Z + U\left(M_1 Y + N_1\right)\right] - \mu L_1\}\phi_1 \\ &+ \{\rho\left[\left(U - U_{avg}\right) L_2 Z + U\left(M_2 Y + N_2\right)\right] - \mu L_2\}\phi_2 \\ &+ \{\rho\left[\left(U - U_{avg}\right) L_3 Z + U\left(M_3 Y + N_3\right)\right] - \mu L_3\}\phi_3 \end{aligned} \tag{A.15}$$

$$\begin{aligned} J_Y &= \{\rho V\left(L_1 Z + M_1 Y + N_1\right) - \mu M_1\}\phi_1 \\ &+ \{\rho V\left(L_2 Z + M_2 Y + N_2\right) - \mu M_2\}\phi_2 \\ &+ \{\rho V\left(L_3 Z + M_3 Y + N_3\right) - \mu M_3\}\phi_3 \end{aligned} \tag{A.16}$$

The flux terms can then be expressed in terms vertices 1, 2, and 3 as:

$$J_X = f_1\phi_1 + f_2\phi_2 + f_3\phi_3 \tag{A.17}$$

$$J_Y = g_1\phi_1 + g_2\phi_2 + g_3\phi_3 \tag{A.18}$$

In Chapter 2, the **LHS** of the momentum equations is:

$$\mathbf{LHS} = \oint \vec{\mathbf{J}} \cdot \hat{\mathbf{n}} dl \tag{A.19}$$

This term can be integrated to obtain the flux across the control volume face $r - s - t$ in Figure 2.4. Using Simpson's quadrature rule, the integration yields:

$$\mathbf{LHS} = \frac{dl}{6}\left(\left[\vec{J} \cdot \hat{n}\right]_r + 4\left[\vec{J} \cdot \hat{n}\right]_s + \left[\vec{J} \cdot \hat{n}\right]_t\right) \tag{A.20}$$

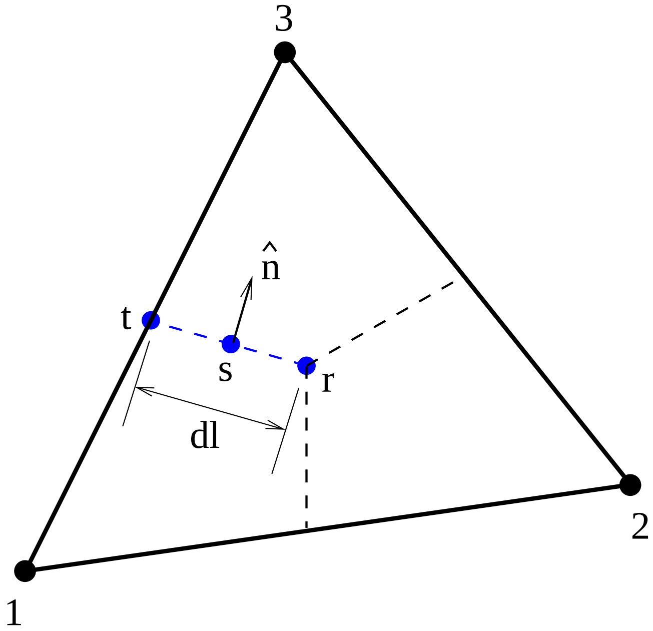


Figure A.1 Flux integration

Expanding Equation A.20:

$$\begin{aligned}
\mathbf{LHS} \quad = \quad & \frac{dl}{6}\left[\left(J_X\hat{e}_X + J_Y\hat{e}_Y\right)_r \cdot \left(n_X\hat{e}_X + n_Y\hat{e}_Y\right)\right. \\
& +4\left(J_X\hat{e}_X + J_Y\hat{e}_Y\right)_s \cdot \left(n_X\hat{e}_X + n_Y\hat{e}_Y\right) \\
& \left. +\left(J_X\hat{e}_X + J_Y\hat{e}_Y\right)_t \cdot \left(n_X\hat{e}_X + n_Y\hat{e}_Y\right)\right] \qquad \text{(A.21)}
\end{aligned}$$

where:

$$n_X = \left(\frac{Y_t - Y_r}{dl}\right) \qquad \text{(A.22)}$$

$$n_Y = -\left(\frac{X_t - X_r}{dl}\right) \qquad \text{(A.23)}$$

The terms are then collected and expressed in X and Y components as:

$$\begin{aligned}
\mathbf{LHS} = \frac{1}{6}&\left[\left(J_{Xr} + 4J_{Xs} + J_{Xt}\right)\left(Y_t - Y_r\right)\right. \\
&\left. -\left(J_{Yr} + 4J_{Ys} + J_{Yt}\right)\left(X_t - X_r\right)\right] \qquad \text{(A.24)}
\end{aligned}$$

Using Equations A.17 and A.18, the flux can be represented in terms of the variable at the vertices of the triangle. The X and Y components of the total flux vector $\vec{J}$ become functions of the grid points and can be written as:

$$J_X = f_1^{()}\phi_1 + f_2^{()}\phi_2 + f_3^{()}\phi_3 \tag{A.25}$$

$$J_Y = g_1^{()}\phi_1 + g_2^{()}\phi_2 + g_3^{()}\phi_3 \tag{A.26}$$

where the brackets () signify that the functions f and g are computed at either r, s, or t on the control volume edge. Substituting Equations A.25 and A.26 into Equation A.24, the expression becomes:

$$\begin{aligned}\mathbf{LHS} = \frac{1}{6}&\left(\left[f_1^r\phi_1 + f_2^r\phi_2 + f_3^r\phi_3\right](Y_t - Y_r)\right.\\ &+4\left[f_1^s\phi_1 + f_2^s\phi_2 + f_3^s\phi_3\right](Y_t - Y_r) + \left[f_1^t\phi_1 + f_2^t\phi_2 + f_3^t\phi_3\right](Y_t - Y_r)\\ &-\left[g_1^r\phi_1 + g_2^r\phi_2 + g_3^r\phi_3\right](X_t - X_r) - \left[g_1^s\phi_1 + g_2^s\phi_2 + g_3^s\phi_3\right](X_t - X_r)\\ &\left.-\left[g_1^t\phi_1 + g_2^t\phi_2 + g_3^t\phi_3\right](X_t - X_r)\right)\end{aligned} \tag{A.27}$$

The equation can now be regrouped according to information at the grid points as:

$$\mathbf{LHS} = a_1\phi_1 + a_2\phi_2 + a_3\phi_3 \tag{A.28}$$

where:

$$a_1 = \frac{1}{6}\left(\left(f_1^r + 4f_1^s + f_1^t\right)(Y_t - Y_r) - \left(g_1^r + 4g_1^s + g_1^t\right)(X_t - X_r)\right) \tag{A.29}$$

$$a_2 = \frac{1}{6}\left(\left(f_2^r + 4f_2^s + f_2^t\right)(Y_t - Y_r) - \left(g_2^r + 4g_2^s + g_2^t\right)(X_t - X_r)\right) \tag{A.30}$$

$$a_3 = \frac{1}{6}\left(\left(f_3^r + 4f_3^s + f_3^t\right)(Y_t - Y_r) - \left(g_3^r + 4g_3^s + g_3^t\right)(X_t - X_r)\right) \tag{A.31}$$

Coefficients Of The Pressure Equation

Pressure is assumed to vary linearly within any particular element or triangle. The procedure for the development of the coefficients is identical to the one for the shape function except that a linear profile is assumed in the form of:

$$p = -\left(\alpha x + \beta y + \gamma\right) \tag{A.32}$$

for node P. Setting up a system of equations in terms of pressure at the grid points, the following is obtained:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{A.33}$$

Using Cramer's rule again, the coefficients can be written as:

$$\begin{aligned} \alpha &= \bar{L}_1 p_1 + \bar{L}_2 p_2 + \bar{L}_3 p_3 && \text{(A.34)} \\ \beta &= \bar{M}_1 p_1 + \bar{M}_2 p_2 + \bar{M}_3 p_3 && \text{(A.35)} \\ \gamma &= \bar{N}_1 p_1 + \bar{N}_2 p_2 + \bar{N}_3 p_3 && \text{(A.36)} \end{aligned}$$

such that:

$$\begin{matrix} \bar{L}_1 = \frac{(y_3 - y_2)}{\bar{\Delta}}, & \bar{L}_2 = \frac{(y_1 - y_3)}{\bar{\Delta}}, & \bar{L}_3 = \frac{(y_2 - y_1)}{\bar{\Delta}}, \\ \bar{M}_1 = \frac{(x_2 - x_3)}{\bar{\Delta}}, & \bar{M}_2 = \frac{(x_3 - x_1)}{\bar{\Delta}}, & \bar{M}_3 = \frac{(x_1 - x_2)}{\bar{\Delta}}, \\ \bar{N}_1 = \frac{(x_3 y_2 - x_2 y_3)}{\bar{\Delta}}, & \bar{N}_2 = \frac{(x_1 y_3 - x_3 y_1)}{\bar{\Delta}}, & \bar{N}_3 = \frac{(x_2 y_1 - x_1 y_2)}{\bar{\Delta}} \end{matrix} \tag{A.37}$$

and:

$$\bar{\Delta} = x_1\left(y_2 - y_3\right) + x_2\left(y_3 - y_1\right) + x_3\left(y_1 - y_2\right) \tag{A.38}$$

APPENDIX B. Divergence in the Local Coordinate System

Figure B.1 shows a diagram of the global and local coordinate systems, where θ is the angle between the Cartesian x-axis and the local X-axis. The coordinate transformation from the Cartesian system (x, y) to the local coordinate system (X, Y) is given by:

$$\begin{aligned} X &= x\ cos\ \theta + y\ sin\ \theta \\ Y &= -x\ sin\ \theta + y\ cos\ \theta \end{aligned}$$

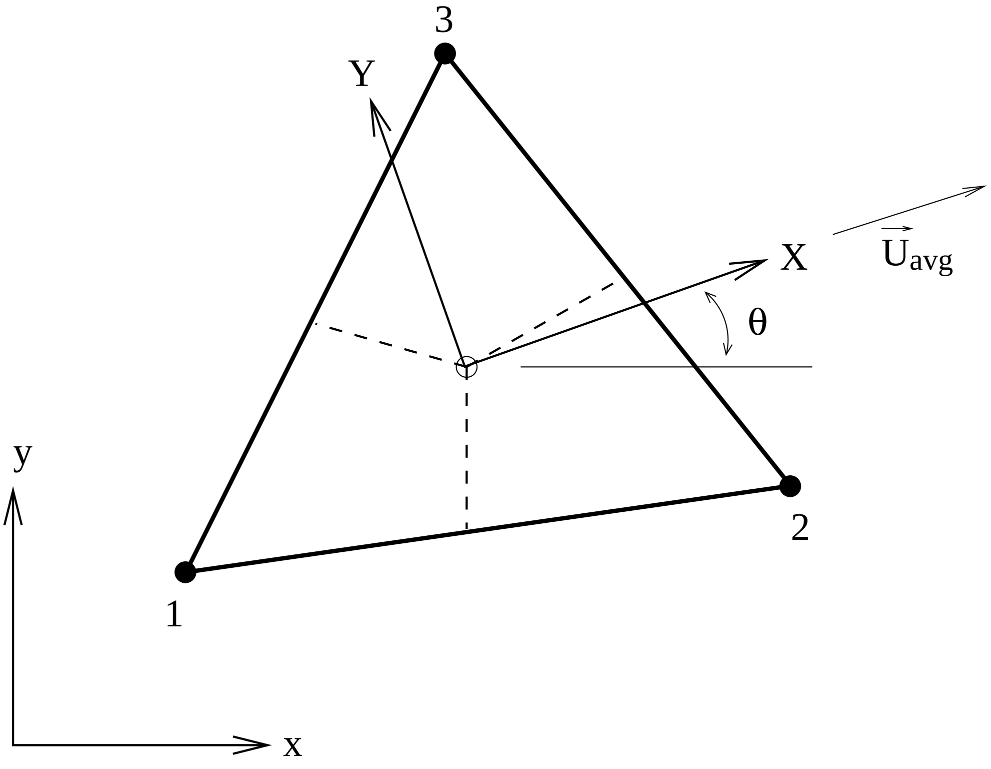


Figure B.1 Global and local coordinate system for a triangular element.

The unit vector of the local coordinate system can be written as:

$$\begin{aligned}\hat{e}_X &= cos\ \theta\ \hat{i} + sin\ \theta\ \hat{j} \\ \hat{e}_Y &= -sin\ \theta\ \hat{i} + cos\ \theta\ \hat{j}\end{aligned}$$

The divergence in the local coordinate system (X, Y) is given by:

$$\nabla = \hat{i}\frac{\partial}{\partial x} + \hat{j}\frac{\partial}{\partial y} \tag{B.1}$$

$$= \hat{i}\left(\frac{\partial}{\partial X}\frac{\partial X}{\partial x} + \frac{\partial}{\partial Y}\frac{\partial Y}{\partial x}\right) + \hat{j}\left(\frac{\partial}{\partial X}\frac{\partial X}{\partial y} + \frac{\partial}{\partial Y}\frac{\partial Y}{\partial y}\right) \tag{B.2}$$

$$= \hat{i}\left(\frac{\partial}{\partial X}\ cos\ \theta - \frac{\partial}{\partial Y}\ sin\ \theta\right) + \hat{j}\left(\frac{\partial}{\partial X}\ sin\ \theta + \frac{\partial}{\partial Y}\ cos\ \theta\right) \tag{B.3}$$

$$= \frac{\partial}{\partial X}\left(cos\ \theta\ \hat{i} + sin\ \theta\ \hat{j}\right) + \frac{\partial}{\partial Y}\left(-sin\ \theta\ \hat{i} + cos\ \theta\ \hat{j}\right) \tag{B.4}$$

$$= \hat{e}_X\frac{\partial}{\partial X} + \hat{e}_Y\frac{\partial}{\partial Y} \tag{B.5}$$

BIBLIOGRAPHY

[1] W. Shyy and R. Mittal. Solution methods for the incompressible Navier-Stokes equations. In *R.W. Johnson (Ed.), Handbook of Fluid Dynamics*, pages 31.1–31.33. CRC Press, Boca Raton, 1998.

[2] W.Q. Tao. *Recent Advances in Computational Heat Transfer*. Science Press, Beijing, 2000.

[3] J.M. Weiss and W.A. Smith. Solution of unsteady, low mach number flow using a preconditioned multi-stage scheme on an unstructured mesh. *AIAA Computational Fluid Dynamics Conference,* 11^{th}*, Orlando, FL*, July 6-9, 1993.

[4] J.M. Weiss and W.A. Smith. Preconditioning applied to variable and constant density flows. *AIAA Journal*, 33:2050–2057, 1995.

[5] J.M. Weiss, J.P. Maruszewski, and W.A. Smith. Implicit solution of preconditioned Navier-Stokes equations using algebraic multigrid. *AIAA Journal*, 37:29–36, 1999.

[6] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2:12–26, 1967.

[7] D. Kwak, J.L.C. Chang, and S.R. Chakravarthy. A three–dimensional incompressible Navier–Stokes flow solver using primitive variables. *AIAA Journal*, 24:390–396, 1985.

[8] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers and Fluids*, 1:73–100, 1973.

[9] D.K. Gartling. *Finite Element Analysis of Viscous, Incompressible Fluid Flow.* PhD thesis, University of Texas, Austin, Texas, 1975.

[10] T.J. Chung. *Finite Element Analysis in Fluid Dynamics.* McGraw-Hill, New York, 1978.

[11] M.E. Braaten. *Development and Evaluation of Iterative and Direct Methods for the Solution of Equations Governing Recirculating Flows.* PhD thesis, University of Minnesota, Minneapolis, MN, 1985.

[12] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, 59:308–323, 1985.

[13] T.J.R. Hughes, W.K. Liu, and A. Brooks. Finite element analysis of incompressible viscous flows by the penalty function formulation. *Journal of Computational Physics*, 30:1–60, 1979.

[14] J.C. Heinrich and R.S. Marshall. Viscous incompressible flow by a penalty function finite element method. *Computers and Fluids*, 9:73–83, 1981.

[15] J.N. Reddy. Penalty finite element methods in mechanics. (AMD-vol.51), November 1982.

[16] M.E. Braaten and W. Shyy. Comparison of iterative and direct method for viscous flow calculations in body-fitted coordinates. *International Journal for Numerical Methods in Fluids*, 6:325–349, 1986.

[17] S.L. Lee and R.Y. Tzong. Artificial pressure for pressure-linked equation. *International Journal for Heat Mass Transfer*, 35:2705–2716, 1992.

[18] Y. Sato, T. Hino, and M. Hinatsu. Unsteady flow simulation around a moving body by an unstructured Navier-Stokes solver. Proceedings of the Sixth Numerical Towing Tank Simulation, Rome, Italy, 2003.

[19] C.H. Sheng, D.L. Whitfield, and W.K. Anderson. A multiblock approach for calculating incompressible fluid flows on unstructured grids. *AIAA Journal*, 37(2):169–176, 1999.

[20] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comput.*, 22:745–762, 1968.

[21] R. Temam. Sur l'approximation de la solution des equations de Navier-Stokes par la methode des pas fractionnaires. *Rational Mechanics and Analysis*, 33(5):377–385, 1969.

[22] F. H. Harlow and J. E. Welch. Numerical calculation of three–dimensional time dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 1:2182, 1965.

[23] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing Corp, New York, 1980.

[24] C.R. Maliska and G.D. Raithby. Calculating 3-D fluid flows using non-orthogonal grid. pages 656–666. Proceedings of Third Int. Conf. on Numerical Methods in Laminar and Turbulent Flows, Seattle, WA, 1983.

[25] R.I. Issa. Solution of the implicit discretized fluid flow equations by operator splitting. Technical report, Mech. Eng. Rep. FS/82/15, Imperial College, London, UK, 1982.

[26] W.Q. Tao, Z.G. Qu, and Y.L. He. A novel segregated algorithm for incompressible fluid flow and heat transfer problems – clear (coupled and linked equations algorithm revised) part i: Mathematical formulation and solution procedure. *Numerical Heat Transfer, Part B*, 45:1–17, 2004.

[27] W.Q. Tao, Z.G. Qu, and Y.L. He. A novel segregated algorithm for incompressible fluid flow and heat transfer problems – clear (coupled and linked equations algorithm revised) part ii: Application examples. *Numerical Heat Transfer, Part B*, 45:19–48, 2004.

[28] F. Moukalled and M. Darwish. A unified formulation of the seggregated class of algorithm for fluid flow at all speeds. *Numerical Heat Transfer, Part B*, 37:103–139, 2000.

[29] T.F. Miller and F.W. Schmidt. Use of pressure weighted interpolation method for solution of incompressible Navier-Stokes equations on non-staggered grids. *Numerical Heat Transfer*, 14:213–233, 1988.

[30] S. Majumdar. Role of underrelaxation in momentum interpolation for calculation of flow with non-staggered grids. *Numerical Heat Transfer*, 13:125–132, 1988.

[31] N. Lambropoulos, E.S. Politis, K.C. Giannakoglou, and K.D. Papailiou. Co-located pressure-correction formulations on unstructured 2-d grids. *Computational Mechanics*, 27:258–264, 2001.

[32] L. Davidson. A pressure correction method for unstructured meshes with arbitrary control volumes. *International Journal for Numerical Methods in Fluids*, 22:265–281, 1996.

[33] F.S. Lien. A pressure-based unstructured grid method for all-speed flows. *International Journal for Numerical Methods in Fluids*, 33:355–374, 2000.

[34] Hua Chen, George P.-G. Huang, and Raymond P. LeBeau. A cell-centered pressure based method for two/three dimensional unstructured incompressible Navier-Stokes solver. AIAA 2005-880. 43rd AIAA Aerospace Sciences Meeting and Exhibit, 10-13 January 2005.

[35] M. Thomadakis and M.A. Leschziner. A pressure-correction method for the solution of incompressible viscous flows on unstructured grids. *International Journal for Numerical Methods in Fluids*, 22:581–601, 1996.

[36] C. Hall, J. Cavendish, and W. Frey. The dual-variable method for solving fluid flow difference equations on delaunay triangulations. *Computers and Fluids*, 20:145–164, 1991.

[37] C. Hall, T. Porsching, and P. Hu. Covolume-dual variable method for thermally expandable flow on unstructured triangular grids. *Journal of Computational Fluid Dynamics*, 2:111–139, 1994.

[38] Nicolaides R., T. Porsching, and C. Hall. Covolume methods in computational fluid dynamics. In *M. Hafez, K. Oshima (Eds.), Computational Fluid Dynamics Review*, pages 279–299. Wiley, Chichester, UK, 1995.

[39] Nicolaides R. The covolume approach to computing incompressible flows. In *M. Gunzburger, R. Nicolaides (Eds.), Incompressible Computational Fluid Dynamics*, pages 295–333. Cambridge University Press, Cambridge, UK, 1993.

[40] B. Perot. Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics*, 159:58–89, 2000.

[41] S. Rida, F. McKenty, F. Meng, and M. Reggio. A staggered control volume scheme for unstructured triangular grids. *International Journal for Numerical Methods in Fluids*, 25:697–717, 1997.

[42] I. Wenneker, A. Segal, and P. Wesseling. A mach-uniform unstructured staggered grid method. *International Journal for Numerical Methods in Fluids*, 40:1209–1235, 2002.

[43] I. Wenneker, A. Segal, and P. Wesseling. Conservation properties of a new unstructured staggered scheme. *Computers and Fluids*, 32:139–147, 2003.

[44] D. Vidovic, A. Segal, and P. Wesseling. A superlinearly convergent finite volume method for the incompressible Navier-Stokes equations on staggered unstructured grids. *Journal of Computational Physics*, 198:159–177, 2004.

[45] B. R. Baliga and S. V. Patankar. A control volume finite-element method for two-dimensional fluid flow and heat transfer. *Numerical Heat Transfer*, 6:245–261, 1983.

[46] Prakash C. *A Finite Element Method for Predicting Flow through Ducts with Arbitrary Cross Sections*. PhD thesis, University of Minnesota, Minneapolis, MN, 1981.

[47] C. Prakash and S.V. Patankar. A control volume based finite element method for solving the Navier-Stokes equations using equal-order velocity-pressure formulation. *Numerical Heat Transfer*, 8:259–280, 1985.

[48] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. volume 43, pages 50–67. Proceedings of Cambridge Philosophical Society, 1947.

[49] R.W. MacCormack. The effect of viscosity in hypervelocity impact cratering. *AIAA Paper No.69-354*, 1969.

[50] J.C. Butcher. A history of Runge-Kutta methods. *Applied Numerical Mathematics*, 20:247–260, 1996.

[51] J.C. Butcher and G. Wanner. Runge-Kutta methods: Some historical notes. *Applied Numerical Mathematics*, 22:113–151, 1996.

[52] E. Fehlberg. Classical fifth, sixth, seventh, and eighth order Runge-Kutta formulas with stepsize control. Technical Report NASA Tech. Rep. TR R-287, Marshall Space Flight Center, Atlanta, GA, 1968.

[53] E. Fehlberg. Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems. Technical Report NASA Tech. Rep. TR R-315, 1969.

[54] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA Paper 81-1259*, June 1981.

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

[56] J.K. Watterson. A pressure-based flow solver for the three-dimensional Navier-Stokes equations on unstructured and adaptive meshes. *AIAA Paper 94-2358*, 1994.

[57] J.C. Butcher. Implicit Runge-Kutta processes. *Mathematics of Computation*, 18(85):50–64, 1964.

[58] C. Hirsch. *Numerical Computation of Internal and External Flows Vol. 1 & 2*. John Wiley & Sons, Chichester, October 1992.

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

[60] Prabir Purohit. Development of an explicit time accurate scheme for incompressible flows. Master's thesis, Iowa State University, Ames, 2001.

[61] Muhammad Ijaz. *Implicit Runge-Kutta Methods to Simulate Unsteady Incompressible Flows*. PhD thesis, Texas A&M University, December 2007.

[62] M.J. Maresca. A pressure-based incompressible unstructured solver for general two-dimensional flows. Master's thesis, Iowa State University, Ames, 1995.

[63] B. R. Baliga and S. V. Patankar. A new finite-element formulation for convection-diffusion problems. *Numerical Heat Transfer*, 3:393–409, 1980.

[64] 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*, 48:387–411, 1982.

[65] R.H. Pletcher and K.-H. Chen. On solving the compressible Navier-Stokes equations for unsteady flows at very low mach numbers. Technical report, AIAA-93-3368, 1993.

[66] Sutikno Wirogo. *Flux Corrected Method: An Accurate Approach to Fluid Flow Modeling.* PhD thesis, Iowa State University, Ames, 1997.

[67] B. F. Armaly, F. Durst, J.C.R. Pereira, and B. Schonung. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics*, 127:473–496, 1983.