DEVELOPMENT OF AN INCOMPRESSIBLE, LAMINAR FLOW SOLVER BASED ON LEAST SQUARES SPECTRAL ELEMENT METHOD WITH P-TYPE ADAPTIVE REFINEMENT CAPABILITIES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALTUĞ ÖZÇELİKKALE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JUNE 2010

Approval of the thesis:

DEVELOPMENT OF AN INCOMPRESSIBLE, LAMINAR FLOW SOLVER BASED ON LEAST SQUARES SPECTRAL ELEMENT METHOD WITH P-TYPE ADAPTIVE REFINEMENT CAPABILITIES

submitted by **ALTUĞ ÖZÇELİKKALE** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Suha Oral
Head of Department, **Mechanical Engineering** ________________

Asst. Prof. Dr. Cüneyt Sert
Supervisor, **Mechanical Engineering Dept., METU** ________________

Examining Committee Members:

Prof. Dr. Mehmet Haluk Aksel
Mechanical Engineering Dept., METU ________________

Asst. Prof. Dr. Cüneyt Sert
Mechanical Engineering Dept., METU ________________

Dr. Tahsin Çetinkaya
Mechanical Engineering Dept., METU ________________

Asst. Prof. Dr. İlker Tarı
Mechanical Engineering Dept., METU ________________

Assoc. Prof. Dr. Hakan Tarman
Engineering Sciences Dept., METU ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ALTUĞ ÖZÇELİKKALE

Signature :

ABSTRACT

DEVELOPMENT OF AN INCOMPRESSIBLE, LAMINAR FLOW SOLVER BASED ON LEAST SQUARES SPECTRAL ELEMENT METHOD WITH P-TYPE ADAPTIVE REFINEMENT CAPABILITIES

Özçelikkale, Altuğ
M.S., Department of Mechanical Engineering
Supervisor : Asst. Prof. Dr. Cüneyt Sert

June 2010, 104 pages

The aim of this thesis is to develop a flow solver that has the ability to obtain an accurate numerical solution fast and efficiently with minimum user intervention. In this study, a two-dimensional viscous, laminar, incompressible flow solver based on Least-Squares Spectral Element Method (LSSEM) is developed. The LSSEM flow solver can work on hp-type non-conforming grids and can perform p-type adaptive refinement. Several benchmark problems are solved in order to validate the solver and successful results are obtained. In particular, it is demonstrated that p-type adaptive refinement on hp-type non-conforming grids can be used to improve the quality of the solution. Moreover, it is found that mass conservation performance of LSSEM can be enhanced by using p-type adaptive refinement strategies while keeping computational costs reasonable.

Keywords: Least-Squares, Spectral Element Method, Non-Conforming Grid, Adaptive Refinement, Mass Conservation

ÖZ

P-TİPİ UYUMLU SIKLAŞTIRMA YETENEKLERİNE SAHİP, EN KÜÇÜK KARELER SPEKTRAL ELEMAN METODU TABANLI BİR SIKIŞTIRILAMAYAN, LAMİNER AKIŞ ÇÖZÜCÜSÜNÜN GELİŞTİRİLMESİ

Özçelikkale, Altuğ
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi : Yar. Doç. Dr. Cüneyt Sert

Haziran 2010, 104 sayfa

Bu tezin amacı sayısal çözümü doğru, hızlı ve verimli bir şekilde, en az kullanıcı müdahalesi ile elde etme yeteneğine sahip bir akış çözücüsü geliştirmektir. Bu çalışmada, iki boyutlu viskoz, laminer, sıkıştırılamayan akışlar için En Küçük Kareler Spektral Eleman Metoduna (LSSEM) dayalı bir akış çözücüsü geliştirilmiştir. LSSEM akış çözücüsü hp-tipi uyuşmaz ağlar üzerinde çalışabilmekte ve p-tipi uyumlu sıklaştırma yapabilmektedir. Çözücüyü doğrulamak için çeşitli test problemleri çözülmüş ve başarılı sonuçlar elde edilmiştir. Özel olarak, hp-tipi uyumsuz ağlar üzerinde p-tipi uyumlu sıklaştırmanın çözümün niteliğini geliştirmek için kullanılabileceği gösterilmiştir. Buna ek olarak p-tipi uyumlu sıklaştırma stratejilerinin hesaplama maliyetlerini makul düzeyde tutarken, LSSEM'in kütle korunumu performansını iyileştirdiği bulunmuştur.

Anahtar Kelimeler: En Küçük Kareler, Spektral Eleman Metodu, Uyuşmaz Ağ, Uyumlu Sıklaştırma, Kütle Korunumu

To my family...

ACKNOWLEDGMENTS

For his guidance and his unremitting encouragement during the research process, I would like to express my deepest gratitude to my thesis supervisor Asst. Prof. Dr. Cüneyt Sert. His continuous support and motivation has kept me going while dealing with all sorts of problems. I aspire to have his breadth of knowledge in engineering and enthusiasm in research.

I would like to express my genuine appreciation to the members of the examining committee: Prof. Dr. Mehmet Haluk Aksel, Dr. Tahsin Çetinkaya, Asst. Prof. Dr. İlker Tarıand Assoc. Prof. Dr. Hakan Tarman for their valuable comments and suggestions.

My appreciation extends to the Scientific and Technological Research Council of Turkey. I am grateful for the financial support they have given me during my graduate studies.

I owe my sincere gratitude to my mother, Gülsen Özçelikkale, my father, Levent Özçelikkale, and my sister, Ayça Özçelikkale Hünerli for their tolerance, patience and encouragement over the years. Their endless love and unwavering support have kept my spirits up whenever I was in need.

I am grateful to my dearest friend and companion, Zeynep Akgül, for being always by my side during all the hard times of writing this thesis. This thesis would not be possible without her love and compassion that have been the remedy of my despairs and the source of my hopes.

I am especially thankful to Ayça Özçelikkale Hünerli and Zeynep Akgül for their help in preparing the thesis for the final print.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES x

LIST OF FIGURES xi

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Related Work 3

1.3 The Present Study 10

1.4 Thesis Outline 11

2 MATHEMATICAL FORMULATION AND NUMERICAL METHOD 12

2.1 Governing Equations 12

2.1.1 Velocity-Pressure-Vorticity Formulation 13

2.1.2 Time Stepping Method for Time Dependent Problems 13

2.1.3 Linearization of Non-linear Convective Terms 15

2.2 Least Squares Method 17

2.2.1 Definitions 17

2.2.2 Least-Squares Formulation 21

2.2.3 Extension to Problems with Multiple Equations and Unknowns 22

2.3 Least-Squares Spectral Element Method 24

2.3.1 Domain Decomposition 24

2.3.2 Approximation of the Unknowns 26

2.3.3 Calculation of the Elemental System of Equations 29

2.3.4 Assembly of Global System of Equations from Elemental Systems 30

2.3.5 Solution of the Global System of Equations........................ 31

3 LSSEM FLOW SOLVER.. 32

3.1 Introduction 32

3.2 Implementation Highlights 33

3.2.1 Elemental Operations 33

3.2.2 Elemental Node Ordering 36

3.2.3 Constrained Approximation Method 38

3.2.4 Static Condensation... 43

3.2.5 Adaptive Refinement Criterion 46

4 VALIDATION OF LSSEM FLOW SOLVER 50

4.1 Kovasznay Flow 50

4.2 2D Lid Driven Cavity Flow... 55

4.3 Flow Past a Cylinder in a Channel 70

4.4 Unsteady Flow Past Cylinder 83

5 CONCLUSION... 94

REFERENCES 97

LIST OF TABLES

TABLES

Table 4.1 Kovaznay flow. Comparison of local posteriori error estimates and relative true errors. The average of the relative true errors, e_{rel}^{ave} is also provided for comparison with error estimate based on least-squares functional $\tilde{e}_{ls}$. All values are the maxima of the quantities across the domain.. 54

Table 4.2 Lid driven cavity flow. Specifications of the grids used in solution of Re=5000 problem. For adaptive solutions the given expansion orders p refer to the initial grid. The error lower bounds are kept small such that there is not any coarsening in the adaptive grids. 60

Table 4.3 Lid driven cavity flow: Re 5000 Solution. Vortex center coordinates and vorticity values at vortex centers.. 66

Table 4.4 Flow past a cylinder in a channel: Specifications of the grids used in the p-type adaptive refinement study 72

Table 4.5 Flow past a cylinder in a channel: Mass flow rates ($\dot{m}$) across several cross sections... 75

Table 4.6 Flow past a cylinder in a channel: The number of degree of freedoms per dependent variable (N_{DOF}) and wall clock times of the solutions. The mass flow rate at $x = 0$ is also provided for comparison. 82

Table 4.7 Unsteady flow past a cylinder. Specifications of the computational grids........ 84

Table 4.8 Unsteady flow past a cylinder. Strouhal numbers and amplitude of y-velocity at $x = 2\,m$, $y = 0$.. 90

LIST OF FIGURES

FIGURES

Figure 2.1 Domain decomposition from global domain Ω to elements Ω^e 25

Figure 2.2 An arbitrary quadrilateral element and the corresponding standard element... 26

Figure 2.3 One-dimensional Lagrange interpolation polynomials and the resulting two-dimensional basis function.. 29

Figure 3.1 Standard and Arbitrary Quadrilateral Elements.. 34

Figure 3.2 Elemental node ordering 37

Figure 3.3 A simple hp-type non-conforming grid 40

Figure 3.4 Constrained approximation at edges 0 and 3 of element *e* 41

Figure 3.5 Elemental node numbering before and after constrained approximation 42

Figure 3.6 Comparison of CPU time and memory uses with and without static condensation.. 45

Figure 4.1 Kovasznay flow. Exact solution and the computational grids used in this study.. 52

Figure 4.2 Kovasznay flow. The convergence trends for conforming, p-type non-conforming and h-type non-conforming grids for Kovasznay flow 53

Figure 4.3 Lid driven cavity flow. Problem setup... 57

Figure 4.4 Lid driven cavity problem. Computational grid, $NE = 334$......................... 57

Figure 4.5 Lid driven cavity problem. Streamlines of Re=100 and 1000 solutions 58

Figure 4.6 Lid driven cavity problem. Velocity profiles across centerlines for Re=100 and 1000 solutions 59

Figure 4.7 Lid driven cavity problem. Streamlines of *Re*=5000 solution. 61

Figure 4.8 Lid driven cavity problem. x-velocity contours of *Re*=5000 solution. 62

Figure 4.9 Lid driven cavity problem. y-velocity contours of *Re*=5000 solution. 62

Figure 4.10 Lid driven cavity problem. Pressure contours of *Re*=5000 solution.............. 63

Figure 4.11 Lid driven cavity problem. Vorticity contours of *Re*=5000 solution. 63

Figure 4.12 Lid driven cavity problem. Re=5000 solution. x-velocity profile across vertical centerline 64

Figure 4.13 Lid driven cavity problem. Re=5000 solution. y-velocity profile across horizontal centerline.. 65

Figure 4.14 Lid driven cavity flow. Distribution of error estimates. 68

Figure 4.15 Lid driven cavity flow. Adaptive grid refinements. 69

Figure 4.16 Flow past a cylinder in a channel: Problem setup....................................... 71

Figure 4.17 Flow past a cylinder in a channel: Computational grid. $NE = 32$ 71

Figure 4.18 Flow Past a Cylinder In a Channel: Contour plots of dependent variables. The contours are based on the solution with grid (a4) for which very good mass conservation is achieved. 73

Figure 4.19 Flow past a cylinder in a channel: Comparison of the streamlines near the cylinder for the non-adaptive and adaptive solutions.. 74

Figure 4.20 Flow past a cylinder in a channel: x-velocity profiles at $x = 0$ between the cylinder and lower channel wall for adaptive grid solutions.................................. 77

Figure 4.21 Flow past a cylinder in a channel: Centerline x-velocity (u_c) profiles along the upstream of the cylinder 78

Figure 4.22 Flow past a cylinder in a channel: Adaptive grids that result from different error upper bounds for $\tilde{e}_{ls}$. The numbers in the upper half of the domain indicate the expansion orders employed in individual elements. Note that adaptive refinement is generally symmetric with respect to horizontal centerline.................. 80

Figure 4.23 Flow past a cylinder in a channel: Adaptive grids that result from different error upper bounds for $\tilde{e}^{u}_{spectral}$. The numbers in the upper half of the domain indicate the expansion orders employed in individual elements. Note that adaptive refinement is generally symmetric with respect to horizontal centerline.................. 81

Figure 4.24 Unsteady flow past cylinder. Problem setup. 84

Figure 4.25 Unsteady flow past a cylinder. Instantaneous adaptive computational grid at $t = 195$ s. NE=174. 85

Figure 4.26 Unsteady flow past a cylinder. Instantaneous contours of dependent variables at $t = 190$ s 87

Figure 4.27 Unsteady flow past a cylinder. Time history plots of several flow variables at $x(= 2.3m, y = 0)$. 88

Figure 4.28 Unsteady flow past a cylinder. Time evolution of the streamlines and the grid resolution 92

Figure 4.28 (Cont.) Unsteady flow past a cylinder. Time evolution of the streamlines and the grid resolution 93

CHAPTER 1

INTRODUCTION

1.1 Motivation

Use of computational approaches to solve flow problems are becoming more common in non-engineering disciplines such as biology or medicine. The application areas of numerical flow solution techniques in these disciplines can be further extended by enabling people who have little or no knowledge in numerical methods use such techniques to solve their problems with ease and success. Such an idea can be realized by the development of numerical softwares that have the ability to obtain an accurate solution fast and efficiently with minimum user intervention. In this context, adaptive grid (mesh) refinement (AMR) strategies come forward as a way of automating the solution procedure. This thesis work constitutes the initial steps of the development of a robust biological flow solver that will obtain an accurate solution automatically using AMR strategies.

This study focuses on numerical solution of viscous, laminar, incompressible flows since flows concerning a wide range of applications including land vehicle aerodynamics and human cardiovascular system are within the viscous incompressible flow regime. Additionally, many low speed flow applications like blood flow in artery involve laminar flow. The most common numerical methods used in the solution of viscous incompressible flows are the finite difference, finite volume and the finite/spectral element methods. In finite difference method (FDM) [1], the derivatives in the governing differential equations are approximated by discrete differencing formulas. While FDM can result in very accurate solutions, see eg. [2, 3], its application is limited to simple geometries. In finite volume method (FVM) [4], the problem domain is partitioned into small regions called finite volumes. The governing equations are integrated over each finite volume analytically by the virtue of Gauss' Theorem and differencing formulas are applied to the resulting flux balance equations. FVMs can be applied to arbitrarily complex geometries. While FVMs are usually low-order accurate, they are widely used in commercial flow solver packages and are applied successfully to practical engineering problems.

Alternatives to FVMs are the finite and spectral element methods (FEM, SEM respectively) that share common aspects in principle. Like FVM, both FEM and SEM employ the discretization of the problem domain into a computational grid composed of small regions called *elements*. The approximate solution is represented by linear or high order polynomial functions over each element. The problem is solved by making the approximate solution satisfy a variational boundary value problem derived from the governing equations. In FEM, the approximate solution is represented by low order linear polynomials. On the contrary, in SEM, the approximate solution is represented by higher order polynomials (expansions). The main difference between the FEM and SEM is the way the quality of the solution is improved. In FEM, convergence to a grid-independent solution is achieved by using smaller elements i.e. geometric (h-type) refinement of the computational grid. On the other hand, in SEM, the polynomial order utilized over each element can also be increased, without dividing the elements, through a process referred as p-type refinement. As a result, SEM can employ both h- and p-type refinements to improve the quality of the solution. Both refinement strategies enrich the approximate solution by introducing additional degrees of freedoms to solve. However, it is known that for problems with smooth solutions, the error in the approximate solution decays much faster with p-type refinement when compared to h-type refinement [5]. Therefore the same accuracy can be obtained with SEM by using less number of degrees of freedom than FEM. In this study, a flow solver based on SEM is developed.

In FEM and SEM, the governing differential equations can be transformed into variational boundary value problems by different methods. One option for this purpose is the Galerkin method that corresponds to principle of minimum potential energy in the case of solid mechanics [6]. Galerkin method has received great interest from FEM community in solid mechanics problems and diffusion problems in fluid mechanics and heat transfer. However extension of Galerkin method to fluid dynamics problems with convection encounters some difficulties such as the necessity to use different orders of approximations for the velocity and pressure unknowns or instabilities in solution that arise when convective transport is important [7]. In solution of fluid dynamics problems, an alternative to Galerkin method is the *least-squares method* that presents a general and robust way to handle flow problems without encountering such difficulties [8]. The solver developed in this study uses a spectral element method based on least-squares variational formulation, also referred as *least-squares spectral element method* (LSSEM).

The present study concerns with the use of LSSEM on hp-type non-conforming grids [9]. In hp-type non-conforming grids an element may be adjacent to more than one element at the same edge (h-type non-conforming) and the order of approximation (expansion) may vary across elements (p-type non-conforming). hp-type non-conforming grids are suitable for AMR since the regions of the problem domain that require higher grid resolution can be refined either by h-type or p-type refinement locally without affecting other regions of the domain. Clever use of both h- and p-type refinements to obtain an optimal computational grid that solves the problem accurately with least amount of computational resources is not straightforward and is a subject of continuing research.

Next section summarizes the literature that is related to the current study.

1.2 Related Work

Least-squares (LS) method has been used as a way of finding solutions to overdetermined linear systems of equations since Gauss' and Legendre's pioneering works in early 19. century [10]. The application of least-squares method to partial differential equations was introduced by Bramble and Schatz [11]. In that work, a least-squares formulation and associated error estimates were presented for the solution of boundary value problems governed by even-degree high order elliptic partial differential equations. This advance has been practically important since such partial differential equations are encountered in engineering problems such as momentum, heat and mass diffusion, and Euler-Bernoulli Beam Theory. At that time, applications of least-squares method in developing finite element methods were still limited since least-squares method required higher regularity on the approximate solution than classical Galerkin method did. This regularity requirement results in impractical finite element base that are continuous and also have continuous first partial derivatives across element interfaces [12]. Later, it was shown by Lynn and Arya [13] that the high regularity requirement of least-squares method can be circumvented by transforming the governing set of differential equations into an equivalent set of first order differential equations. Following this idea, Zienkiewicz et al. [14] introduced a finite element method based on least-squares method to solve plane-elasticity and elastic beam problems. In the following years the approach of equivalent set of first order equations became the common practice to develop LSFEM and LSSEM in the least-squares community.

Among the variational methods, Galerkin method has been the most prominent one in developing FEM for problems of solid and fluid mechanics. The popularity of Galerkin method can be attributed to the fact that an arbitrary boundary value problem without a known equivalent variational form can still be formulated in Galerkin method and the Galerkin method has the best approximating properties in energy norm for self-adjoint equations that are observed in solid mechanics and diffusion problems[8, 15]. The applications of Galerkin FEM on such problems also result in linear system of equations with symmetric coefficient matrices which can be solved efficiently [16]. However, favorable properties of Galerkin method are lost when it is applied to non-self adjoint equations such as the Navier-Stokes equations that arise in incompressible flow problems. In particular, Galerkin method does not have the best approximating property for this type of problems [8]. Additionally, solution of incompressible Navier-Stokes equations with FEM based on a Galerkin mixed (velocity-pressure) method requires the use of different approximation orders for velocity and pressure to satisfy the mathematical stability constraint: Ladyzhenskaya-Babŭska-Brezzi (LBB) condition [17] in order to avoid spurious pressure solutions [18]. It also results in linear systems with non-symmetric coefficient matrices and zero-diagonal terms corresponding to pressure unknown hindering the stability of temporal solution procedures [19].

Least-squares method is offered by Jiang [8] as a better alternative to Galerkin method for the solution of fluid dynamics problems. Its major advantages over the Galerkin method can be listed as follows:

- In FEM and SEM based on least-squares method, the same approximation (expansion) order can be used for all approximated unknowns without violating the LBB stability condition. This feature of least-squares method enables the desired order of accuracy for all unknowns and also simplifies the implementation work considerably [20].

- Application of LSFEM and LSSEM, even on problems governed by non-self-adjoint partial differential equations, results in linear system of equations with symmetric and positive-definite matrices. Therefore, efficient iterative linear system solvers such as preconditioned conjugate gradient method can be used [21]. This convenience does not arise in the case of Galerkin and finite volume methods where resulting matrices are mostly non-symmetric and also are not positive-definite.

- The governing equations of fluid flow and heat transfer have different characteristics for different flow regimes. The equations are elliptic for slow viscous flow and hyperbolic for convection dominated problems. These different characteristics require special treatment in the case of Galerkin and finite volume methods. For instance, upwinding schemes are found necessary to account for the direction of information travel in convection dominated flow. Without upwinding schemes, both methods return unstable results, therefore are useless. In the case of least-squares method, no special treatments such as upwinding as well as artificial dissipation, staggered grid or operator splitting are necessary. In other words, least-squares method presents a unified approach to fluid flow and heat transfer problems [8]. In fact, that point explains the great interest the least-squares method has received recently from researchers working on computational fluid dynamics e.g. [22–25].

As mentioned before, the practical application of LSFEM and LSSEM requires a set of first order differential equations. For the incompressible Navier-Stokes problems, different sets of first order equations that are equivalent to the higher order problem can be derived by introducing different auxiliary variables. The most commonly used auxiliary variables are the vorticity vector ω, velocity gradient tensor $\nabla \mathbf{U}$ and viscous (deviatoric) stress tensor $\boldsymbol{\tau}$. The use of these additional variables result in the first-order equation sets referred as the velocity-pressure-vorticity ($\mathbf{U} - p - \omega$), velocity-pressure-velocity gradient ($\mathbf{U} - p - \nabla \mathbf{U}$), and velocity-pressure-stress ($\mathbf{U} - p - \boldsymbol{\tau}$) formulations respectively. These three formulations involve different number of unknowns and have different boundary conditions. They also exhibit different convergence performances for the iterative linear system solution [26]. A comparative study of the formulations was performed by Ding and Tsang [26]. In that study, CPU times and memory required by the formulations were compared and it was observed that the $\mathbf{U} - p - \omega$ formulation is a promising candidate for wide-range use since it results in the least memory use. It was also reported that $\mathbf{U} - p - \sigma$ formulation may be useful for non-Newtonian, viscoelastic fluids or turbulence problems where stress variables are of interest. However it has the disadvantage of too many additional variables to solve. Following this discussion, the present study uses the $\mathbf{U} - p - \omega$ formulation in reducing the governing equations into an equivalent set of first order differential equations.

A significant portion of the work on least-squares method has been devoted to developing LSFEM for the solution of incompressible Stokes and Navier-Stokes equations. The aspects of mathematical framework such as error estimates upon h-convergence, optimality of the methods and well-posed boundary conditions were investigated [18, 20–22, 27–30]. The reference books of Jiang [8] and Bochev and Gunzburger [31] can serve as a summary of the work done on those subjects. In the recent years, the focus of research on this area has shifted to development of p-type FEM [21, 32, 33] and SEM [23, 24] based on the least-squares method. SEM studies were conducted using both hierarchical basis expansions [12, 34, 35] and nodal basis expansions [12, 23, 36–42]. Additionally, Least-squares spectral collocation schemes are also introduced recently [25, 43–46]. While a different name: “collocation scheme” is given to the methods in these studies, it can be shown that LSFEM and LSSEM are equivalent to weighted least-squares collocation schemes when the integrals arising from the variational formulation are evaluated by numerical quadrature. In this case quadrature points turn out to the collocation points [8].

It can be observed that LSFEM comes with many advantageous features. However, it also has some disadvantages against the Galerkin method. Namely, introducing the additional dependent variables to obtain the equivalent set of first order equations increases the use of computational resources i.e. CPU time and memory. Nevertheless the effort to solve for the additional unknowns may not be totally in vain since the additional unknown, for instance vorticity, may be a quantity of physical interest.

Another disadvantage of the least-squares method is that LSFEM is known to result in solutions with severe mass conservation violations especially for the problems with inflows and outflows [47, 30]. For LSSEM, the situation is improved by use of high order expansions but the mass conservation problem still persists [38]. In contrary to the least-squares method, Galerkin mixed method is not associated with any mass conservation problems. The reason for the lack of mass conservation can be understood by comparing the way the continuity (divergence-free) constraint is imposed in Galerkin mixed and least-squares methods. In Galerkin mixed method, the continuity equation is included in the variational formulation as a constraint imposed by the method of Lagrange multipliers where pressure is the Lagrange multiplier [19]. Therefore, the continuity equation is enforced exactly in the Galerkin mixed method. On the other hand, in least-squares method, the continuity residual is minimized in a least-squares sense along with the residuals of other equations. Therefore the continuity equation is not enforced exactly in the least-squares method[38]. Recognizing the cause of the problem some methods and remedies to improve the mass conservation properties of LSFEM and LSSEM were presented in the literature.

First of all, a *restricted LSFEM* where the continuity equation is enforced by the method of Lagrange multipliers was suggested [47] to circumvent the mass conservation problem. While this method results in good mass conservation it also destroys the two favorable features of LSFEM i.e. symmetry and positive-definiteness of the coefficient matrix. Furthermore, it requires the solution of an extra linear system to determine of Lagrange multipliers [30].

Another approach suggested to improve mass conservation was to increase the *importance* of the continuity equation among the other equations by using a weighting factor (greater than unity) in continuity contribution to squared sum of residuals. This approach results in the *weighted LSFEM*[30, 48]. The mass conservation can be enhanced by increasing the weighting factor appropriately. However, such an improvement in mass conservation is achieved at the expense of momentum conservation. Moreover, the weighting factor increases the condition number of the resulting coefficient matrix hindering the convergence of iterative and accuracy of direct solution methods [38].

It was also reported that least-squares method, while having poor mass conservation properties, conserves momentum better than the Galerkin mixed method. Additionally, it was noted that mass conservation performance of LSSEM gets better as the expansion order is increased [38]. Both of these statements are valid. Nevertheless, it will be shown in the present study (in section 4.3) that mass conservation is still an important problem of LSSEM since it affects the formation of flow features such as recirculation regions behind bluff bodies.

Decoupling the solution of velocity and pressure fields through a consistent splitting scheme was also found to be effective in recovering good mass conservation properties [40]. In such an approach, first, the velocity field is solved by application of least-squares method to an advection-diffusion equation and a divergence-curl system successively. Pressure is obtained by substituting the solved velocity field in a Poisson equation and solving it by either Galerkin or least-squares method. This approach may be preferable to the original least-squares method for the cases where the derivations of well-posed initial and boundary conditions are straightforward.

Least-squares penalty formulations were also reported to have good mass conservation properties [42, 45]. In penalty methods, pressure is eliminated from the governing equations and the continuity equation is imposed in regularized form through a penalty parameter [19]. Success of such methods depends on the proper selection of the adjustable penalty parameter. Moreover the solution procedure suffers from high condition numbers of the coefficient matrix if a penalty parameter that is too large is selected. In the studies of Prabhakar and Reddy [42] an iterative penalty method is employed to keep the penalty parameter small and the condition number of the coefficient matrix manageable.

Apart from these studies, a new first order formulation for the incompressible Navier-Stokes equations was introduced to use with the original least-squares method [49, 50]. The new formulation involves a new variable that is the combination of pressure gradient and convective acceleration: $\mathbf{r} = \nabla p + Re\,(\mathbf{U} \cdot \nabla\mathbf{U})$ where Re is the Reynolds number. While the new formulation results in good mass conservation, application of boundary conditions associated with $\mathbf{r}$ is quite involved and requires further study [49].

Most recently, it has been shown that use of few and large elements with high expansion orders in LSSEM results in good mass conservation. This idea has also been verified in the present study. Nevertheless, the use of high expansion orders uniformly across the solution domain is computationally prohibitive for practical problems and should be avoided.

Another interest of this study is the hp-type non-conforming grids. In practice, hp-type non-conforming grids are obtained by constructing an underlying h-type non-conforming grid and use of a high order method like SEM with varying orders of expansion over elements. One popular class of h-type non-conforming grids are non-conforming Cartesian Grids (NCG). In NCGs, the computational grid is obtained by recursive sub-division of the domain into square (in 2D) or cubic (in 3D) regions. The level of subdivision may vary across different regions of the domain, resulting in a h-type non-conforming grid structure. NCGs are popular in recent AMR studies since they offer an elegant way to discretize domains with complex-moving boundaries[51]. Moreover, NCG has a flexible grid structure which allows for efficient implementation of AMR [52]. In other words, the practical aspect of using NCG is the ease of grid generation and adaptation. There is a great amount of work on h-type adaptive refinement on NCGs with low order methods like FVM or FDM. See eg. [53, 54] One focus of studies is on the *immersed boundary methods (IBM)* where the computational grid does not conform to the boundaries of immersed bodies. The affects of the boundaries are simulated by addition of singular body forces in the formulation. Then the usual FVM or FDM is applied [55, 56]. Since the grid is not required to conform with complex boundaries, grid generation and adaptation is straightforward. However, IBMs are not suitable for high order methods like SEM. Use of relatively large elements in SEM renders the application of boundary conditions complicated and existence of singular forces in formulation hinders the exponential convergence properties of SEM. Another approach for the use of NCGs is the cut-cell method [51] where the grid conforms with the boundaries of immersed bodies. For this purpose, the NCG generated across the domain is modified to form a conforming unstructured grid near the boundaries. The presence of unstructured grid complicates the grid generation and adaptation. Nevertheless, the cut-cell method can be used with high order methods.

Studies on hp-type non-conforming grids based on NCGs are relatively limited. Studies to devise a universal hp-type adaptive refinement strategy has been of particular importance in that context. Through the series of related studies by Demkowicz et al. [9], Oden et al. [57] and Rachowicz et al. [58], the constrained approximation method, which is also used in the present study, was introduced to construct the solution on a hp-type non-conforming grid [9]. Secondly, several error estimates to drive hp-type adaptive refinement were proposed and compared by solving model problems on simple domains. It was concluded that error estimates based on element residuals offer a good compromise between accuracy and computation cost [57]. Finally, an hp-type adaptive refinement strategy that aims to minimize the error with a fixed number of degrees of freedom was proposed. It was observed that the hp-type adaptive refinement strategy can lead to exponential rates of convergence even for problems where the solution has large gradients and the domain has geometric singularities [58]. Another study was by Henderson [59] where an error estimate based on the expansion coefficients of an high order expansion of the SEM solution was proposed to drive h-type adaptive refinement on an hp-type non-conforming grid composed of elements with fixed expansion order ($p = 7$ in specific).

Adaptive refinement studies with least-squares method are rare. One important study is by Jiang and Carey [60] which proposed an error estimate based on least-squares functional to drive h-type adaptive refinement with LSFEM. More recently in the work of Cai et al. [61], LSFEM is used together with a grid deformation method (also referred as r-type adaptive refinement) where a FEM grid composed of fixed number of elements is deformed to capture the interfaces and boundaries. The error estimation in that study was also based on elemental residuals.

1.3 The Present Study

In this study, a two-dimensional incompressible flow solver based on least-squares spectral element method (LSSEM) is developed. the focus of the study is on solutions with LSSEM using p-type adaptive refinement on hp-type non-conforming grids when available.

As stated in Section 1.2, the adaptive refinement studies in the literature focus on h-type adaptive refinement on non-conforming Cartesian grids with low order methods. p-type adaptive refinement studies are limited. In particular, adaptive refinement studies in least-squares literature are restricted to h-type refinement [60] and r-type refinement [61] with LSFEM. The adaptive refinement studies presented in this thesis involves p-type adaptive refinement with LSSEM and aims to address this gap in the least-squares literature.

The survey of the related work reveals that the poor mass conservation properties of least-squares method and the ways to improve it have received great interest recently. The remedies proposed in the literature involve modification of the original least-squares formulation and they cause some favorable features of the formulation to be lost. Recently it was shown that good mass conservation can be achieved by the original least-squares formulation using few and large elements with very high expansion orders ($p \propto 20$) [46]. Implementing such an approach on a conforming grid results in excessive and unnecessary refinement. Therefore it is computationally impractical. In the present study, it is shown that mass conservation performance of LSSEM can be enhanced by using p-type adaptive refinement strategies up to the same accuracy as the uniform refinement while keeping computational costs practical. For additional discussion on this subject, one can refer to Section 4.3.

1.4 Thesis Outline

Chapter 1 introduces the motivation of the thesis, the related work, the present study, and the outline of the thesis. Chapter 2 presents the governing equations solved in this study, and overviews the theory of the least-squares method and the least-squares spectral element method. Chapter 3 illustrates the features and implementation highlights of the LSSEM flow solver developed in this study. In Chapter 4, the LSSEM flow solver is validated by solving of several test problems and comparing the results with the literature. Chapter 5 summarizes the findings of the thesis and presents ideas for future work.

CHAPTER 2

MATHEMATICAL FORMULATION AND NUMERICAL METHOD

This chapter presents the governing equations of the laminar incompressible flows solved in this study, the formulation of least-squares method and the least-squares spectral element method.

2.1 Governing Equations

Conservation of momentum in viscous flows is governed by the Navier-Stokes equations. By incompressible and Newtonian fluid assumptions, the Navier-Stokes equations can be written in terms of velocity field and pressure as the unknowns (dependent variables). Navier-Stokes equations are solved together with conservation of mass (continuity) equation to determine the dependent variables. Then for the viscous, incompressible flow of a Newtonian fluid, the governing equation set is as follows:

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{(Continuity)} \qquad (2.1a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p - \nu \nabla^2 \mathbf{u} = \mathbf{f} \qquad \text{(Momentum)} \qquad (2.1b)$$

where $\mathbf{u}$ is the velocity field, p is pressure, $\mathbf{f}$ is the body force, ρ is density and ν is kinematic viscosity. $\mathbf{u}$ and p are the dependent variables that are functions of both space variables and time. $\mathbf{f}$, ρ and ν are assumed to be constant.

2.1.1 Velocity-Pressure-Vorticity Formulation

Eqn. set (2.1) is referred as the *primitive variable form* of the governing equations. It involves second order derivatives in the viscous diffusion term. As stated in section 1.2, direct application of least-squares method to eqn. set (2.1) is impractical [13]. Therefore the governing equations are reformulated into an equivalent set of first order differential equations by introducing an auxiliary unknown. It this study, vorticity vector $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is used for this purpose. By using the vector identity:

$$\nabla^2 \mathbf{u} = -\nabla \times (\nabla \times \mathbf{u}) + \nabla (\nabla \cdot \mathbf{u}) \tag{2.2}$$

and the conservation of mass, eqn. (2.1a), also referred as the *divergence-free constraint*, the governing equations become:

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{(Continuity)} \tag{2.3a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p + \nu \nabla \times \boldsymbol{\omega} = \mathbf{f} \qquad \text{(Momentum)} \tag{2.3b}$$

$$\boldsymbol{\omega} - \nabla \times \mathbf{u} = 0 \qquad \text{(Vorticity Definition)} \tag{2.3c}$$

Eqn. set (2.3) is known as the “velocity-pressure-vorticity” $(\mathbf{u} - p - \boldsymbol{\omega})$ formulation of the Navier-Stokes equations. Other equivalent sets of first order systems based on velocity gradient, viscous stress tensor and stream function are also available [26, 48]. As stated in section 1.2, velocity-pressure-vorticity formulation is selected due to it’s widespread use in least-squares community, low memory requirements in implementation and the fact that vorticity is an physical quantity of interest that should be calculated at some point in solution. It can be noted that eqn. set (2.3) involves only first order derivatives. Therefore, the application of least-squares method to Eqn. set (2.3) can be realized by practical C^0-continuous finite and spectral element bases [12].

2.1.2 Time Stepping Method for Time Dependent Problems

It should be noted that the solution of Eqn. set (2.3) depends on time as well as spatial variables. SEM and FEM solutions for time-dependent problems can be obtained in two alternative ways:

1. Space-Time Coupled Formulation: Both temporal and spatial variations of the unknowns are approximated by SEM (or FEM). In this approach, time appears as an addi-

tional dimension in the spectral element formulation. For instance if a two-dimensional problem is to be solved by a space-time coupled SEM formulation three-dimensional spectral elements are used to construct the approximate solution.

2. Space-Time Decoupled Formulation: The temporal and spatial variations of the unknowns are assumed to be separable. The most common practice with space-time decoupled formulations is to approximate the spatial variation of the unknowns using SEM or FEM while the time dependence is approximated by finite difference formulas. [19].

Several space-time coupled LSSEM formulations are reported in the literature [32, 33, 62]. The advantage of these formulations is the high accuracy in time. However space-time coupled formulations are not commonly used in practical applications since their application to three-dimensional problems are computationally prohibitive [62]. Therefore, in this study a space-time decoupled formulation is used. For time stepping, α-family time integration scheme [19] is selected for it's simplicity. In this study, the governing equations are first discretized in time using the time integration scheme. LSSEM is applied to the resulting semi-discrete equations.

In α-family time integration scheme, the integral of a function $f(t)$ over a time interval $(t, t+\Delta t)$ is approximated as:

$$\int_t^{t+\Delta t} f(t)\,dt = \Delta t\,\left[(1-\alpha) f(t) + \alpha f(t+\Delta t)\right] \tag{2.4}$$

$$= \Delta t\,\left[(1-\alpha) f_n + \alpha f_{n+1}\right] \tag{2.5}$$

where α is a parameter between 0 and 1. The subscripts n and $n+1$ are the time step indices that correspond to times t and $t+\Delta t$.

In the way to obtain a time stepping method for the governing equations, it is assumed that the solution of the eqn. set (2.3) is known at time t. Then eqn. (2.3b) is integrated over the time interval $(t, t+\Delta t)$. Note that the other equations in the eqn. set (2.3) need not be integrated in time since they do not involve any time dependence.

$$\underbrace{\int_t^{t+\Delta t} \frac{\partial \mathbf{u}}{\partial t}\,dt}_{\mathbf{u}_{n+1}-\mathbf{u}_n} + \int_t^{t+\Delta t} \left(\mathbf{u}\cdot\nabla\mathbf{u} + \frac{1}{\rho}\nabla p + \nu\nabla\times\omega\right) dt = \int_t^{t+\Delta t} \mathbf{f}\,dt \tag{2.6}$$

The first term on the right hand side of eqn. (2.6) can be integrated exactly by the Fundamental Law of Calculus. The rest of the integrals are approximated with the α-family time integration scheme, eqn. (2.5). At that point, the subscript $n + 1$ is dropped from the unknowns to be solved for brevity. The subscript n is kept to indicate that the associated variables are known from the solution of previous time step. Then eqn. (2.6) becomes:

$$\frac{\mathbf{u} - \mathbf{u}_n}{\Delta t} + (1 - \alpha)\left(\mathbf{u}_n \cdot \nabla \mathbf{u}_n + \frac{1}{\rho} \nabla p_n + \nu \nabla \times \omega_n\right) + \alpha \left(\mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p + \nu \nabla \times \omega\right) = (1 - \alpha)\, \mathbf{f}_n + \alpha\, \mathbf{f} \tag{2.7}$$

where all terms have been divided by Δt for convenience. Since body force $\mathbf{f}$ is assumed to be constant, right hand side of eqn. (2.7) reduces to “$(1 - \alpha)\, \mathbf{f}_n + \alpha\, \mathbf{f} = \mathbf{f}$”. The terms with subscript n are known from the previous time step. Therefore they can be moved to the right hand side. Then eqn. (2.7) becomes:

$$\frac{\mathbf{u}}{\Delta t} + \alpha \left(\mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p + \nu \nabla \times \omega\right) = \mathbf{f}^* \tag{2.8}$$

where

$$\mathbf{f}^* = \mathbf{f} + \frac{\mathbf{u}_n}{\Delta t} - (1 - \alpha)\left(\mathbf{u}_n \cdot \nabla \mathbf{u}_n + \frac{1}{\rho} \nabla p_n + \nu \nabla \times \omega_n\right) \tag{2.9}$$

It should be noted that all terms constituting the new right hand side vector $\mathbf{f}^*$ are constants or known from the previous time step.

2.1.3 Linearization of Non-linear Convective Terms

An additional approximation to the governing equations arise from the fact that the convective acceleration term $\mathbf{u} \cdot \nabla \mathbf{u}$ in the Navier-Stokes equations is nonlinear with respect to the velocity field. This non-linearity can be eliminated by using an iterative procedure based on the Newton’s method [63]. Newton’s method involves a first order variational approximation to the non-linear term with:

$$\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} \cong \mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k+1} + \mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k} - \mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k} \tag{2.10}$$

where superscript k is the non-linear iteration index. $\mathbf{u}^k$ are known from the previous non-linear iteration. $\mathbf{u}^{k+1}$ are calculated in the current non-linear iteration and substituted as known values in the next nonlinear iteration. Substituting eqn. (2.10) in eqn. (2.8) and rearranging the terms:

$$\frac{\mathbf{u}}{\Delta t} + \alpha \left(\mathbf{u}^k \cdot \nabla \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}^k + \frac{1}{\rho} \nabla p + \nu \nabla \times \omega\right) = \mathbf{f}^* \tag{2.11}$$

where

$$\mathbf{f}^* = \mathbf{f} + \frac{\mathbf{u}_n}{\Delta t} + \alpha\, \mathbf{u}^k \cdot \nabla \mathbf{u}^k - (1-\alpha)\left(\mathbf{u}_n \cdot \nabla \mathbf{u}_n + \frac{1}{\rho}\nabla p_n + \nu \nabla \times \boldsymbol{\omega}_n\right) \tag{2.12}$$

Again, the superscript $k+1$ is dropped from the unknowns for simplicity. The momentum equation (2.11) can be written together with the continuity eqn. (2.3a) and the definition of vorticity (2.3c) to obtain the final form of governing equations:

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{(Continuity)} \tag{2.13a}$$

$$\frac{\mathbf{u}}{\Delta t} + \alpha\left(\mathbf{u}^k \cdot \nabla \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}^k + \frac{1}{\rho}\nabla p + \nu \nabla \times \boldsymbol{\omega}\right) = \mathbf{f}^* \qquad \text{(Momentum)} \tag{2.13b}$$

$$\boldsymbol{\omega} - \nabla \times \mathbf{u} = 0 \qquad \text{(Vorticity Def.)} \tag{2.13c}$$

where

$$\mathbf{f}^* = \mathbf{f} + \frac{\mathbf{u}_n}{\Delta t} + \alpha\, \mathbf{u}^k \cdot \nabla \mathbf{u}^k - (1-\alpha)\left(\mathbf{u}_n \cdot \nabla \mathbf{u}_n + \frac{1}{\rho}\nabla p_n + \nu \nabla \times \boldsymbol{\omega}_n\right)$$

Both time stepping method and the non-linear convective terms result in iterative solution of the governing equations. At each time step, an iterative solution is performed to solve the non-linear system of equations. The inner iterations can be terminated when the some measure of the change in the velocity magnitude or the individual velocity components is less than a tolerance. In this study, the relative change in the Euclidean norm of the x-velocity unknown vector $\tilde{\mathbf{u}}$ is used as the non-linear error measure:

$$\frac{||\tilde{\mathbf{u}}^{k+1} - \tilde{\mathbf{u}}^k||_2}{||\tilde{\mathbf{u}}^{k+1}||_2} < \varepsilon_{nonlinear} \tag{2.14}$$

Common values for $\varepsilon_{nonlinear}$ is from 10^{-4} to 10^{-6} for which non-linear convergence is usually achieved in less than 5 iterations. Nevertheless, the actual convergence rate depends on the problem solved.

In this study, the governing equations are solved for two-dimensional problems based on Cartesian coordinates (x, y). For this case eqn. (2.13) reduces to:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.15a}$$

$$\frac{1}{\Delta t}u + \alpha\left(u^k\frac{\partial u}{\partial x} + v^k\frac{\partial u}{\partial y} + u\frac{\partial u^k}{\partial x} + v\frac{\partial u^k}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial \omega}{\partial y}\right) = f_x^* \tag{2.15b}$$

$$\frac{1}{\Delta t}v + \alpha\left(u^k\frac{\partial v}{\partial x} + v^k\frac{\partial v}{\partial y} + u\frac{\partial v^k}{\partial x} + v\frac{\partial v^k}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} - \nu\frac{\partial \omega}{\partial x}\right) = f_y^* \tag{2.15c}$$

$$\omega - \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} = 0 \tag{2.15d}$$

where

$$f_x^* = f_x + \frac{1}{\Delta t}u_n + \alpha\left(u^k\frac{\partial u^k}{\partial x} + v^k\frac{\partial u^k}{\partial y}\right) - (1-\alpha)\left(u_n\frac{\partial u_n}{\partial x} + v_n\frac{\partial u_n}{\partial y} + \frac{1}{\rho}\frac{\partial p_n}{\partial x} + \nu\frac{\partial \omega_n}{\partial y}\right) \tag{2.16a}$$

$$f_y^* = f_y + \frac{1}{\Delta t}v_n + \alpha\left(u^k\frac{\partial v^k}{\partial x} + v^k\frac{\partial v^k}{\partial y}\right) - (1-\alpha)\left(u_n\frac{\partial v_n}{\partial x} + v_n\frac{\partial v_n}{\partial y} + \frac{1}{\rho}\frac{\partial p_n}{\partial y} - \nu\frac{\partial \omega_n}{\partial x}\right) \tag{2.16b}$$

Here, u is the x-component of the velocity vector (x-velocity) and v is the y-component of the velocity vector (y-velocity). p is the pressure and ω is the z-component of vorticity vector. For two-dimensional flows, vorticity vector is in z-direction. So the subscript is dropped. Subscripts x and y denote the x- and y-components of the other vector quantities. Eqn. set (2.15) are solved for the four dependent variables: u, v, p and ω.

Eqn. set (2.15) is the final form of the governing equations. It is found convenient in implementation to keep the governing equations in dimensional form. In the next section, the least-squares method is applied to the eqn. set (2.15).

2.2 Least Squares Method

In this section the mathematical formulation of least-squares method is derived from variational principles.

2.2.1 Definitions

The derivation of least-squares formulation is illustrated using the following model boundary value problem (BVP):

$$\mathcal{L}u = f \quad \text{on } \Omega \tag{2.17a}$$

$$\mathcal{B}u = g \quad \text{on } \partial\Omega \tag{2.17b}$$

where

Ω	:	Problem domain
$\partial\Omega$	:	Boundary of the problem domain
$\mathcal{L}$	:	A 1st order, linear partial differential operator
u	:	A scalar valued unknown function
f	:	Right hand side (RHS) function
$\mathcal{B}$	:	An algebraic boundary operator
g	:	Boundary function

The problem domain and the boundary are composed of points $\mathbf{x} = \{x_1, x_2, \cdots, x_{N_{sd}}\}$ where N_{sd} is the number of spatial dimensions. The unknown and the RHS vector are functions of $\mathbf{x}$. In this problem, the differential operator involves first order derivatives only. Therefore, the boundary operator imposes boundary conditions that involve the unknown itself and not it's derivatives i.e. only essential boundary conditions. Stating that the boundary operator is algebraic signifies this point. The BVP (2.17) is assumed to be well-posed and therefore has a unique solution.

The discussion starts with the assumption that the RHS function f is square-integrable on Ω. In other words, f is a member of the space of square-integrable functions on Ω: $L^2(\Omega)$.

The space of real valued square-integrable functions is defined as:

$$L^2(\Omega) = \left\{ u : \Omega \rightarrow \mathbb{R} \,\middle|\, \int_\Omega u^2 d\Omega \; < \; \infty \right\} \tag{2.18}$$

$L^2(\Omega)$ is an inner product space. Therefore the orientation of two members $u, v \,\epsilon\, L^2(\Omega)$ can be measured by the inner product:

$$(u, v)_{L^2(\Omega)} = \int_\Omega u\, v \; d\Omega \tag{2.19}$$

and a measure of the size of a member is provided by the inner product induced norm:

$$||u||_{L^2(\Omega)} = \left(\int_\Omega u^2 \; d\Omega \right)^{1/2} \tag{2.20}$$

When it comes to the unknown function u it is observed that u must be square-integrable even when operated by the first order linear differential operator $\mathcal{L}$. It can be shown that u is a member of a subspace of $L^2(\Omega)$: $H^1(\Omega)$ which is one of the Hilbert spaces [64]. $H^1(\Omega)$ is defined as:

$$H^1(\Omega) = \left\{ u \,\epsilon\, L^2(\Omega) \,\middle|\, \frac{\partial u}{\partial x_i} \,\epsilon\, L^2(\Omega),\; i = 1, 2, \cdots, N_{sd} \right\} \tag{2.21}$$

This space is also an inner product space equipped with the following inner product and norm:

$$(u, v)_{H^1(\Omega)} = (u, v)_{L^2(\Omega)} + \sum_{i=1}^{N_{sd}} \left(\frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_i} \right)_{L^2(\Omega)} \tag{2.22a}$$

$$||u||_{H^1(\Omega)} = \left(||u||^2_{L^2(\Omega)} + \sum_{i=1}^{N_{sd}} ||\frac{\partial u}{\partial x_i}||^2_{L^2(\Omega)} \right)^{1/2} \tag{2.22b}$$

When a solution to the BVP (2.17) is to be found using a variational method like the Galerkin or least-squares methods, the solution is sought among the members of these function spaces. Both $L^2(\Omega)$ and $H^1(\Omega)$ are infinite dimensional spaces. In finite element and spectral element applications, approximate solution is sought in finite dimensional subspaces of $H^1(\Omega)$.

The solution of the BVP (2.17), u, can be decomposed into two components:

$$u = u_h + u_b \tag{2.23}$$

where u_h is the part of the solution that satisfies the homogeneous boundary conditions i.e. $\mathcal{B}u_h = 0$. On the other hand, u_b is the part that satisfies the original non-homogeneous boundary conditions in the boundary equation (2.17b). This procedure is referred as the *lifting of the solution* [15]. Lifting is used for convenience in deriving the variational formulations and applying boundary conditions in implementation. It can be noted that a suitable u_b can be found by the mere knowledge of the boundary conditions. Therefore, it is actually a known function. Then u_h is the new unknown of the problem. It is a member of the subspace of $H^1(\Omega)$ that satisfies the homogeneous boundary conditions: $H^1_0(\Omega) = \left\{ u \,\epsilon\, H^1\Omega \mid \mathcal{B}u = 0 \right\}$.

When an approximate solution $\tilde{u}$, apart from the exact solution, is substituted into the differential equation (2.17a), a *residual* R that is a non-zero function on Ω arises:

$$R = \mathcal{L}\tilde{u} - f \neq 0 \text{ on } \Omega \tag{2.24}$$

This residual function is of interest in the variational methods.

Finally, the concepts of *functional* and *variation of a function* are introduced briefly. A functional is a function whose argument is also a function [19]. For instance in:

$$F(x, u, \frac{du}{dx}) = x + 2u - , \frac{du}{dx} \tag{2.25}$$

F is a functional of the function u and it's derivative $\frac{du}{dx}$, which are treated as independent variables. u and $\frac{du}{dx}$ themselves are functions of x.

Another concept is the variation of a function. The variation of function u, denoted by δu, is an admissible change from u to another function $u+\delta u$ for the fixed values of the independent variable x [19]. The change is admissible in a way that, if u is prescribed as some values of x, for instance through a boundary condition, δu must be zero at these values of x since u can not be varied there. δu is otherwise arbitrary.

First variation of a functional is analogous to the differential of a function. It is the change in the value of the functional with respect to variations in it's function arguments:

$$\delta F(x, u, \frac{du}{dx}) = \frac{\partial F}{\partial u}\delta u + \frac{\partial F}{\partial\left(\frac{du}{dx}\right)}\delta\left(\frac{du}{dx}\right) \tag{2.26}$$

In eqn. (2.26) there is no term involving variation of x since it is kept constant through this process.

An extremum (local minimum or maximum) of a function u can be found by setting it's differential du to zero and solving for independent variable x. Similarly, an extremum of a functional can be found by setting it's variational to zero and solving for it's function argument u.

For additional information in calculus of variations, one can refer to Reddy [19].

2.2.2 Least-Squares Formulation

Least squares formulation is obtained by applying the least-squares method to the BVP (2.17). Given an arbitrary function $u \epsilon H^1(\Omega)$, the residual is given by eqn. (2.24):

$$R = \mathcal{L}u - f \neq 0 \tag{2.24}$$

An integral functional $\mathcal{I}$ can be defined by having the square of the residual and integrating over Ω:

$$\mathcal{I} = \mathcal{I}(u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \cdots, \frac{\partial u}{\partial x_{N_{sd}}}) = \int_\Omega R^2 \, d\Omega \tag{2.27}$$

The functional $\mathcal{I}$ is referred as the *least-squares functional*. It can also be defined as the square of the L^2-norm of the residual function. i.e. $\mathcal{I} = ||R||^2_{L^2(\Omega)}$.

In least-squares method, the solution u that makes the least-squares functional a minimum is sought. This is achieved by having the first variation of the least-squares functional and requiring it to be zero for u.

$$\begin{aligned} \delta\mathcal{I} &= \delta \int_\Omega R^2 \, d\Omega &= 0 \\ &= \delta \int_\Omega (\mathcal{L}u - f)^2 \, d\Omega &= 0 \\ &= \int_\Omega \delta \, (\mathcal{L}u - f)^2 \, d\Omega &= 0 \end{aligned} \tag{2.28}$$

The last step in eqn (2.28) is possible since variation and integration operations are interchangeable. Performing the variation operation,

$$2 \int_\Omega (\mathcal{L}u - f)(\delta\mathcal{L}u) \, d\Omega = 0 \tag{2.29}$$

The variation and differentiation operations are also interchangeable. Then $\delta(\mathcal{L}u) = \mathcal{L}(\delta u)$. Denoting the variation of u by v for convenience,

$$\int_\Omega (\mathcal{L}u - f) \, \mathcal{L}v \, d\Omega = 0 \tag{2.30}$$

It should be noted that, while $u \epsilon H^1(\Omega)$, $v \epsilon H^1_0(\Omega) \subset H^1(\Omega)$. Since the solution is specified at the boundary through the boundary condition (2.17b), it's variation is zero there. It is desirable to seek the solution in the same space as it's variation. Therefore the solution can be lifted using eqn. (2.23) Substituting eqn. (2.23) into (2.30) and rearranging the terms,

$$\int_\Omega \mathcal{L}v\mathcal{L}u_h \, d\Omega = \int_\Omega \mathcal{L}v(f - \mathcal{L}u_b) \, d\Omega \tag{2.31}$$

Then the least-squares formulation for the BVP (2.17) can be stated as:

$$
\begin{gathered}
\text{Find } u_h \,\epsilon\, H_0^1(\Omega) \text{ such that} \\
\int_\Omega \mathcal{L}v\,\mathcal{L}u_h\,d\Omega \;=\; \int_\Omega \mathcal{L}v\,(f - \mathcal{L}u_b)\,d\Omega \\
\text{For all } v \,\epsilon\, H_0^1(\Omega)
\end{gathered}
\tag{2.32}
$$

After finding u_h, u is obtained by eqn. (2.23).

Since $H_0^1(\Omega)$ is infinite dimensional, problem (2.32) requires simultaneous solution of infinitely many equations. In the spectral element method, $H_0^1(\Omega)$ is replaced by a finite dimensional subspace, say $H_0^{1,hp}(\Omega)$.Then the approximate solution u_h^{hp} is sought in that space.

2.2.3 Extension to Problems with Multiple Equations and Unknowns

The least-squares formulation in the previous section is derived for a BVP with a single equation and unknown. However the governing equations introduced in section 2.1 involves multiple equations and unknowns. In particular, there are four equations: the continuity equation, two components of momentum equation and definition of vorticity, and four unknowns: the x-velocity u, y-velocity v, pressure p and vorticity ω. For this case, the least-squares formulation can be derived in a similar manner.

If the governing equations (2.15) are written in operator matrix form:

$$
\underbrace{\begin{bmatrix}
\frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 & 0 \\
\frac{1}{\Delta t} + \alpha\left(\frac{\partial u^k}{\partial x} + u^o\frac{\partial}{\partial x} + v^k\frac{\partial}{\partial y}\right) & \alpha\frac{\partial u^k}{\partial y} & \frac{\alpha}{\rho}\frac{\partial}{\partial x} & \alpha v\frac{\partial}{\partial y} \\
\alpha\frac{\partial v^k}{\partial x} & \frac{1}{\Delta t} + \alpha\left(\frac{\partial v^k}{\partial y} + u^k\frac{\partial}{\partial x} + v^k\frac{\partial}{\partial y}\right) & \frac{\alpha}{\rho}\frac{\partial}{\partial y} & -\alpha v\frac{\partial}{\partial x} \\
\frac{\partial}{\partial y} & -\frac{\partial}{\partial x} & 0 & 1
\end{bmatrix}}_{\mathcal{L}}
\underbrace{\begin{pmatrix} u \\ v \\ p \\ \omega \end{pmatrix}}_{\mathbf{u}}
=
\underbrace{\begin{pmatrix} 0 \\ f_x^* \\ f_y^* \\ \omega \end{pmatrix}}_{\mathbf{f}}
\tag{2.33}
$$

Here $\mathcal{L}$ is an $N_{eqn}\times N_u$ differential matrix operator, the unknown vector, $\mathbf{u}$, and the right-hand-side vector, $\mathbf{f}$, are $N_u \times 1$ vectors where N_{eqn} is the number of equations and N_u is the number of unknowns (or dependent variables).

The vector valued L^2 and H^1 spaces are defined by the Cartesian product of corresponding scalar spaces i.e. $\mathbf{L^2}(\Omega) = (L^2(\Omega))^{N_u}$ and $\mathbf{H^1}(\Omega) = (H^1(\Omega))^{N_u}$.

Defining the residual as $\mathbf{R} = \mathcal{L}\mathbf{u} - \mathbf{f}$, the least-squares functional becomes:

$$\mathcal{I} = \int_{\Omega} \mathbf{R}^{T}\,\mathbf{R}\,d\Omega \tag{2.34}$$

The least-squares formulation for a BVP with multiple equations and unknowns can be obtained by carrying out the procedure outlined in the previous section. The resulting least-squares formulation is:

$$\begin{gathered}\text{Find } \mathbf{u_h} \,\epsilon\, \mathbf{H_0^1}(\Omega) \text{ such that}\\ \int_{\Omega} (\mathcal{L}\mathbf{v})^{T}\,\mathcal{L}\mathbf{u_h}\,d\Omega \;=\; \int_{\Omega} (\mathcal{L}\mathbf{v})^{T}\,(\mathbf{f} - \mathcal{L}\mathbf{u_b})\,d\Omega\\ \text{For all } \mathbf{v} \,\epsilon\, \mathbf{H_0^1}(\Omega)\end{gathered} \tag{2.35}$$

2.3 Least-Squares Spectral Element Method

Spectral element method (SEM) is an approximate solution technique which uses a combination of the sub-domain division approach of finite element methods (FEM) and high order approximation of spectral methods (SM)[65]. As in FEM, the domain is divided into several sub-domains (elements). While FEM utilizes linear (1^{st} order) polynomial approximations of unknowns over each element, SEM follows the approach of SM and approximates the unknowns using high-order expansion bases over each element. In SM, high-order expansion bases are in the form of either the Fourier series or orthogonal polynomial expansions. In the case of SEM, use of Fourier series is restricted to the problems with periodic solutions. Polynomial expansions are found more suitable otherwise. In this study, a polynomial spectral element due to Patera [65] is employed.

Spectral element methods have different names based on the underlying variational formulation. The least-squares spectral element method (LSSEM) is basically a spectral element method based on the least-squares formulation introduced in section 2.2.

Numerical solution with LSSEM is performed in the following steps:

- Domain Decomposition
- Approximation of the Unknowns
- Calculation of the Elemental System of Equations
- Assembly of Global System of Equations from Elemental Systems
- Solution of the Global System of Equations

2.3.1 Domain Decomposition

In both FEM and SEM, first, the problem domain is divided into sub-domains, referred as *elements*. Domain decomposition is an efficient way of handling complex domains and arbitrary boundary conditions [8]. Figure 2.1 illustrates the domain decomposition process.

The idea of domain decomposition emerges from the fact that an integral over a domain can be written as a summation of non-overlapping sub-domains. As a result, integrals that appear

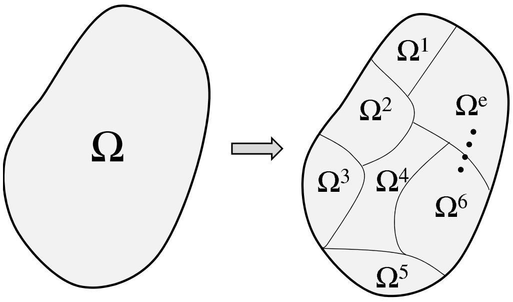


Figure 2.1: Domain decomposition from global domain Ω to elements Ω^e

in the variational formulations such as eqn. (2.31) can be written as follows:

$$\int_{\Omega} (\cdots)\, d\Omega = \int_{\Omega^1} (\cdots)^1\, d\Omega^1 + \int_{\Omega^2} (\cdots)^2\, d\Omega^2 + \cdots + \int_{\Omega^e} (\cdots)^e\, d\Omega^e \tag{2.36}$$

In general practice, two-dimensional domains are decomposed into triangular or quadrilateral elements or a combination of both. While, triangular elements provide better flexibility in decomposing the domain their use in spectral element studies are limited when compared to the quadrilateral elements. The problems with the triangular elements are mainly the necessity to use impractical coordinate transformations and the problem of finding basis expansions on triangular domains with good approximation properties that also allow for efficient implementation [66]. The drawbacks of triangular elements tend to be eliminated in recent studies [66–68] Nevertheless, quadrilateral elements are still associated with more efficient implementation due to tensor product basis functions. In this study, the problem domain is decomposed into quadrilateral elements.

Before presenting the approximation of unknowns over the elements, a word about the elemental coordinates is necessary. The efficient implementation of FEM/SEM depends on handling of the arbitrary coordinates of elements in a standard manner. For this purpose, every element is transformed onto a corresponding standard element and elemental operations are performed on the standard element. Figure 2.2 shows an arbitrary quadrilateral element on the right and the corresponding standard element on the left. The standard element is associated with coordinates ξ, η where $-1 < \xi, \eta < 1$. The coordinates of the physical domain: $\mathbf{x} = (x, y)$ can be defined as a function of standard domain coordinates i.e. $\mathbf{x}(\xi, \eta) = (x(\xi, \eta), y(\xi, \eta))$. As a result, transformation from an arbitrary element to the standard element and vice versa are available. The details of the standard element and coordinate transformations are presented in section 3.2.1.

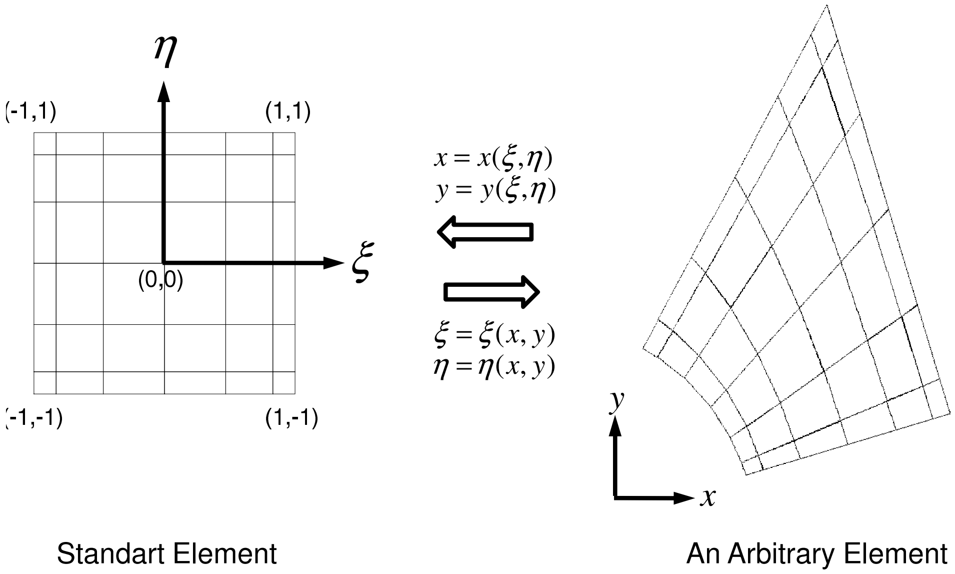


Figure 2.2: An arbitrary quadrilateral element and the corresponding standard element

2.3.2 Approximation of the Unknowns

After the domain decomposition, unknowns are approximated by basis expansions over each element.

An approximate solution of a single unknown over an element e, represented by u^e, can be written as:

$$u^e(\mathbf{x}) \approx \tilde{u}^e(\mathbf{x}) = \sum_{i=0}^{N^e_{DOF}-1} \hat{u}^e_i \phi_i(\mathbf{x}); \tag{2.37}$$

where $\hat{u}^e_i$ are the expansion coefficients that are to be found. They are also referred as the *elemental degrees of freedom (DOFs).* $\phi_i(\mathbf{x})$ are the basis functions and N^e_{DOF} is the number of elemental degrees of freedom[1]. When the basis function $\phi_i(\mathbf{x})$ is given as a function of physical domain coordinates, it is implied that the basis functions are actually defined on the standard element and transformed on the element e by the coordinate transformation $\mathbf{x} = \mathbf{x}(\xi, \eta)$.

Eqn. (2.37) can be written in vector form by defining the vectors: $\hat{\mathbf{u}}^e = \left(\hat{u}^e_0, \hat{u}^e_1, \cdots, \hat{u}^e_{N^e_{DOF}-1}\right)^T$ and $\boldsymbol{\phi}(\mathbf{x}) = \left(\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \cdots, \phi_{N^e_{DOF}-1}(\mathbf{x})\right)^T$ as:

$$\tilde{u}^e(\mathbf{x}) = \hat{\mathbf{u}}^{e\,T}\ \boldsymbol{\phi}(\mathbf{x}) \tag{2.38}$$

[1] The array indexing in the text follows the convention of C++ programming language where dummy indices start from 0.

The unknown vector $\mathbf{u} = (u, v, p, \omega)^T$ introduced in eqn. (2.33) involves multiple dependent variables. As stated in section 1.2, least-squares method allows for the use of equal order approximations for all of the unknowns of the problem. Therefore each unknown in $\mathbf{u}$ can be approximated by the same basis expansion. In matrix notation,

$$\tilde{\mathbf{u}}^e(\mathbf{x}) = \hat{\mathcal{U}}^{e\,T}\ \boldsymbol{\phi}(\mathbf{x}) \tag{2.39}$$

where

$$\hat{\mathcal{U}}^e = \begin{bmatrix} \hat{u}_0^e & \hat{v}_0^e & \hat{p}_0^e & \hat{\omega}_0^e \\ \hat{u}_1^e & \hat{v}_1^e & \hat{p}_1^e & \hat{\omega}_1^e \\ \vdots & \vdots & \vdots & \vdots \\ \hat{u}_{N_{DOF}^e-1}^e & \hat{v}_{N_{DOF}^e-1}^e & \hat{p}_{N_{DOF}^e-1}^e & \hat{\omega}_{N_{DOF}^e-1}^e \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\phi}(\mathbf{x}) = \begin{pmatrix} \phi_0(\mathbf{x}) \\ \phi_1(\mathbf{x}) \\ \vdots \\ \phi_{N_{DOF}^e-1}(\mathbf{x}) \end{pmatrix} \tag{2.40}$$

In eqn. (2.40) N_{DOF}^e stands for the number of elemental degrees of freedoms per dependent variable.

The expansion presented in eqn. (2.37) can be *modal* or *nodal* depending on the choice of basis functions.

Restricting the attention to one-dimensional problems for the time being, in modal expansions, the basis functions are polynomials with increasing order. This type of expansions are also called *hierarchical expansions* since a modal expansion of order p contains all the basis functions of a modal expansion of order $p - 1$ plus an additional polynomial of order p. Modal expansions are constructed by using an orthogonal set of polynomials as the basis functions. Examples to such orthogonal sets of polynomials are the Chebyshev and Legendre polynomials that are the special cases of the family of Jacobi polynomials [15].

In nodal expansions, the basis functions are polynomials of the same order. A nodal expansion of order p is constructed by selecting a set of $p + 1$ points $\{\xi_i\}_{i=0}^{p}$ which is known as the *collocation grid*. Then $p + 1$ polynomial functions $L_i(\xi)_{i=0}^{p}$ of order p can be defined by requiring that

$$L_i(\xi_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad i, j = 1, 2, \cdots, p + 1 \tag{2.41}$$

Eqn. (2.41) is referred as the *cardinality (kronecker delta)* property of the basis expansion. The resulting basis functions are essentially the Lagrange interpolating polynomials passing through the points of the collocation grid. For instance, $L_i(\xi)$ is the Lagrange interpolating polynomial that is 1 at the i^{th} collocation point and 0 at all other collocation points.

It is due to the cardinality property that the expansion coefficients $\hat{\mathbf{u}}^e$ in nodal expansions are same as the unknown values evaluated at the points of collocation grid. So the one-dimensional nodal expansion of the unknown $u(x)$ takes the form:

$$u(\xi) = \sum_{i=0}^{p} u_i L_i(\xi) \tag{2.42}$$

where u_i is the value of the unknown evaluated at the corresponding collocation point: $u_i = u(\xi_i)$.

Up to now, only one-dimensional basis functions are discussed. For two-dimensional problems, the set of basis functions can be constructed by the tensor products of the sets of one-dimensional basis functions defined for ξ and η separately.

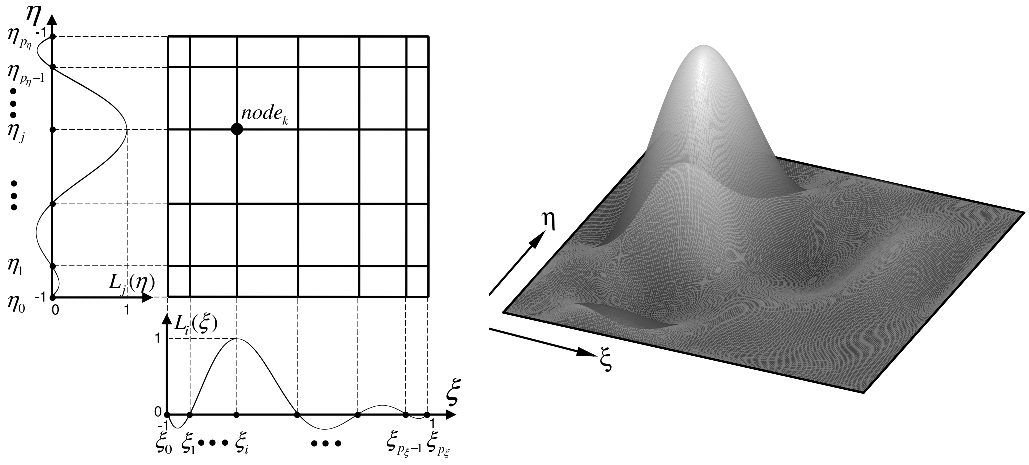
In this study, the unknowns are approximated by nodal expansions. The one-dimensional basis functions are the Lagrange interpolating polynomials that pass through the points of the Gauss-Lobatto-Legendre (GLL) grid. The points of the GLL grid $\{\xi_i\}_{i=0}^{p}$ are the roots of the polynomial $(1-\xi)(1+\xi)P_p^{'}(\xi)$, $\xi \,\epsilon\, [-1, 1]$ where $P_p(\xi)$ is the Legendre polynomial of order p. Then the one-dimensional Lagrange interpolating polynomial of order p that passes through the i^{th} point of the GLL grid $\{\xi_i\}_{i=0}^{p}$ can be given as:

$$L_i(\xi) = \frac{(1-\xi)(1+\xi)P_p^{'}(\xi)}{p(p+1)P_p(\xi_i)(\xi_i - \xi)}, \qquad 0 \le i \le p \tag{2.43}$$

Following the tensor product approach, the basis functions are defined as the product of one-dimensional Lagrange interpolation polynomials:

$$\phi_k(\mathbf{x}(\xi,\eta)) = L_i(\xi)\, L_j(\eta), \qquad \begin{array}{c} 0 \le i \le p_\xi \\ 0 \le j \le p_\eta \\ 0 \le k \le N_{DOF}^e - 1, \\ N_{DOF}^e = (p_\xi + 1)(p_\eta + 1) \end{array} \tag{2.44}$$

It can be noted that the two-dimensional basis function preserves the cardinality property of the Lagrange interpolating polynomials. Therefore the elemental DOFs can be associated with physical locations on the element called *nodes* or *nodal points*. In this study, the nodal points are essentially the set of points formed by the tensor product of one-dimensional GLL



(a) One-Dimensional Lagrange interpolating polynomials

(b) Two-Dimensional basis function

Figure 2.3: One-dimensional Lagrange interpolation polynomials and the resulting two-dimensional basis function

grids of appropriate orders. Then the elemental DOFs can be identified as the unknown values evaluated at the nodal points. In that case, the approximate solution is represented by:

$$\mathbf{u}^e(\mathbf{x}) = \sum_{k=0}^{N^e_{DOF}-1} \mathbf{u}^e_k\, \phi_k(\mathbf{x}) \qquad \text{OR} \qquad \mathbf{u}^e(\mathbf{x}) = \mathcal{U}^{eT} \boldsymbol{\phi}(\mathbf{x}) \tag{2.45}$$

where

$$\mathbf{u}^e_k = (u_k, v_k, p_k, \omega_k)^T \quad \text{and} \quad \mathcal{U}^e_{kn} = \left(\mathbf{u}^e_k\right)_n, \qquad k = 0, 1, \cdots, N^e_{DOF}, \quad n = 0, 1, \cdots, N_u$$

u_k, v_k, p_k, and ω_k are the unknown values evaluated at nodal point k. Figure 2.3a illustrates a pair of Lagrange interpolating polynomials, $L_i(\xi)$ and $L_j(\eta)$. Figure 2.3b presents the resulting two-dimensional basis function. The basis function is 1 at node k and 0 at all other nodes. Node k is the node that has the coordinates (ξ_i, η_j).

2.3.3 Calculation of the Elemental System of Equations

The integral in the least-squares formulation (2.35) can be written over each element as:

$$\int_{\Omega^e} \mathcal{L}(\mathbf{v}^e)^T\, \mathcal{L}(\mathbf{u}^e)\, d\Omega^e \;=\; \int_{\Omega^e} \mathcal{L}(\mathbf{v})^T\, (\mathbf{f})\, d\Omega^e \tag{2.46}$$

If the elemental unknowns $\mathbf{u}^e$, their variations $\mathbf{v}^e$ and the force vector $\mathbf{f}^e$ are approximated by the nodal basis expansions in eqn. (2.40), eqn. 2.46 becomes

$$\mathbf{V}^{eT} \left(\int_{\Omega^e} \mathcal{L}\boldsymbol{\phi}^T\, \mathcal{L}\boldsymbol{\phi}\, d\Omega^e \right) \mathbf{U}^e \;=\; \mathbf{V}^{eT} \int_{\Omega^e} \mathcal{L}\boldsymbol{\phi}^T\, (\mathbf{E}^e)\, d\Omega^e \tag{2.47}$$

where

$$\begin{aligned}
\mathcal{L}\boldsymbol{\phi} &= \left[\mathcal{L}(\phi_0), \mathcal{L}(\phi_1), \cdots, \mathcal{L}(\phi_{N^e_{DOF}-1})\right] && (N_{eqn} \times N^e_u) && (2.48)\\
\mathbf{U}^e &= \left((\mathbf{u}^e)^T_0, (\mathbf{u}^e)^T_1, \cdots, (\mathbf{u}^e)^T_{N^e_{DOF}-1}\right)^T && (N^e_u \times 1) && (2.49)\\
\mathbf{V}^e &= \left((\delta\mathbf{u}^e)^T_0, (\delta\mathbf{u}^e)^T_1, \cdots, (\delta\mathbf{u}^e)^T_{N^e_{DOF}-1}\right)^T && (N^e_u \times 1) && (2.50)\\
\mathbf{E}^e &= \left((\mathbf{f}^e)^T_0, (\mathbf{f}^e)^T_1, \cdots, (\mathbf{f}^e)^T_{N^e_{DOF}-1}\right)^T && (N^e_u \times 1) && (2.51)
\end{aligned}$$

$N^e_u = N_u \times N^e_{DOF}$ is the *total number of elemental DOFs*, that is the number of elemental DOFs used to approximate all of the dependent variables. $\mathbf{U}^e$ is referred as the *elemental solution vector*. It can be noted that the elemental variation vector $\mathbf{V}^e$ is present in both sides of the equation (2.47). Therefore it can be discarded. The resulting set of equations:

$$\mathbf{K}^e\,\mathbf{U}^e = \mathbf{F}^e \tag{2.52}$$

where

$$\begin{aligned}
\mathbf{K}^e &= \int_{\Omega^e} \mathcal{L}\boldsymbol{\phi}^T\,\mathcal{L}\boldsymbol{\phi}\, d\Omega^e && (N^e_u \times N^e_u) && (2.53)\\
\mathbf{F}^e &= \int_{\Omega^e} \mathcal{L}\boldsymbol{\phi}^T\,(\mathbf{E}^e)\, d\Omega^e && (N^e_u \times 1) && (2.54)
\end{aligned}$$

$\mathbf{K}^e$ and $\mathbf{F}^e$ are known as the *elemental stiffness matrix* and *elemental force vector* respectively.

2.3.4 Assembly of Global System of Equations from Elemental Systems

The next step is the summation of elemental integrals as illustrated in eqn. (2.36). The integral summation is implemented by an *direct stiffness summation* operation where the a global system is assembled from elemental system of equations. Since approximate solution is continuous across element interfaces the elemental unknowns at a common interface of two elements correspond to the same global unknown. Associated terms in elemental systems are summed together and enter the global system as single terms.

The global system of equations is:

$$\mathbf{K}\,\mathbf{U} = \mathbf{F} \tag{2.55}$$

where $\mathbf{K}$ is the *global stiffness matrix*, $\mathbf{U}$ is the *global solution vector* and $\mathbf{F}$ is the *global force vector*.

2.3.5 Solution of the Global System of Equations

The global system of equations (2.55) can be solved by both iterative methods [8] and direct methods [44]. In the case of LSSEM, the global stiffness matrix is symmetric and positive definite. This feature allows for use of efficient iterative linear system solvers such as preconditioned conjugate gradient method.

The assembly of the elemental systems into a global system is not necessarily performed physically. The system can be solved by a Jacobi preconditioned conjugate gradient method through an element-by-element (EBE) procedure where the global coefficient matrix is never constructed [60]. In the so-called *matrix-free method* [21], the memory requirements are further reduced by not keeping the elemental systems in the memory, instead recalculating them at each linear solver iteration. Such an approach is especially required when solving large three-dimensional problems where forming the global system or keeping the elemental systems in memory require excessive resources.

The design of the solver presented in this study is currently focused on solving two-dimensional problems. For two-dimensional problems, keeping the elemental systems in memory through the linear solution process is still feasible. Since the elemental systems are calculated only once for the linear solution, such an approach is especially efficient as far as computation time is concerned. Therefore In this study, the global system of equations are solved using a Jacobi-preconditioned conjugate gradient method in and element-by-element (EBE) procedure where all elemental systems are present in memory at the same time. The approach to keep the elemental systems in the memory is also motivated by static condensation that is introduced in section 3.2.4.

CHAPTER 3

LSSEM FLOW SOLVER

3.1 Introduction

A two-dimensional incompressible flow solver based on least-squares spectral element method is developed in this study. The solver, referred as *LSSEM flow solver*, has the following features:

- Solves time-dependent, incompressible flow problems governed by the Navier-Stokes equations for Newtonian fluids.
- Works on computational grids with both geometry based (h-type) and expansion based (p-type) non-conforming interfaces between elements.
- Performs p-type adaptive refinement through the transient solution.

Implementation highlights are:

- Written in C++ programming language [69] with an object-oriented approach.
- Has a file input and output interface.
- Gordon-Hall transfinite interpolation [70]is used to enable spectral elements with curved edges.
- Uses constrained approximation method [71, 72] in constructing an approximate solution that is continuous across non-conforming element interfaces.
- Performs elemental boundary-interior decomposition and static condensation [15] to reduce the CPU time and memory requirements.

- p-type adaptive refinement procedures are based on two alternative error estimates: Error estimate based on elemental least-squares functional [60] ($\tilde{e}_{ls}$) and error estimate based on spectral expansion coefficients [59] ($\tilde{e}_{spectral}$).

Implementation highlights are described in the following section.

3.2 Implementation Highlights

3.2.1 Elemental Operations

In both FEM and SEM, the problem domain is decomposed into elements of arbitrary shapes (certainly with some limitations) over which certain operations are performed. An example to the elemental operations is the evaluation of integrals that constitute the components of the elemental system introduced in section 2.3. A robust implementation that can handle the unique geometries and other arbitrary aspects of elements in a standard manner can be realized through the concept of *standard (master) element*. In particular, the elemental operations on arbitrary elements are transfered onto and performed on corresponding standard elements. Most of the element attributes like node coordinates and derivatives of basis functions are defined on standard elements. They are calculated for arbitrary elements using appropriate coordinate transformations.

A standard element and an arbitrary element that can be handled by the LSSEM flow solver are illustrated in figure 3.1. The standard element,represented by Ω^e_{st}, corresponding to an arbitrary element e is defined in two-dimensions as a square domain:

$$\Omega^e_{st} = \left\{(\xi,\eta)\,\epsilon\,\mathbb{R}^2\,|{-1} \le \xi \le 1,\, -1 \le \eta \le 1\right\} \tag{3.1}$$

(ξ,η) are the standard domain coordinates. The standard element and the associated arbitrary element e are presented in figure 3.1. The convention adopted in this study for the ordering of the corners and edges of a standard quadrilateral element is as follows: The first corner of the element, identified as corner 0, is located at $(\xi,\eta) = (-1,-1)$. The remaining corners are numbered in the order they are encountered during a counter-clock-wise (CCW) travel around the element. The first edge, edge 0, is identified as the edge between corner 0 and corner 1. the rest of the faces are, again, numbered in the order they are encountered during a CCW travel around the element. An elemental operation like the evaluation of an integral over the

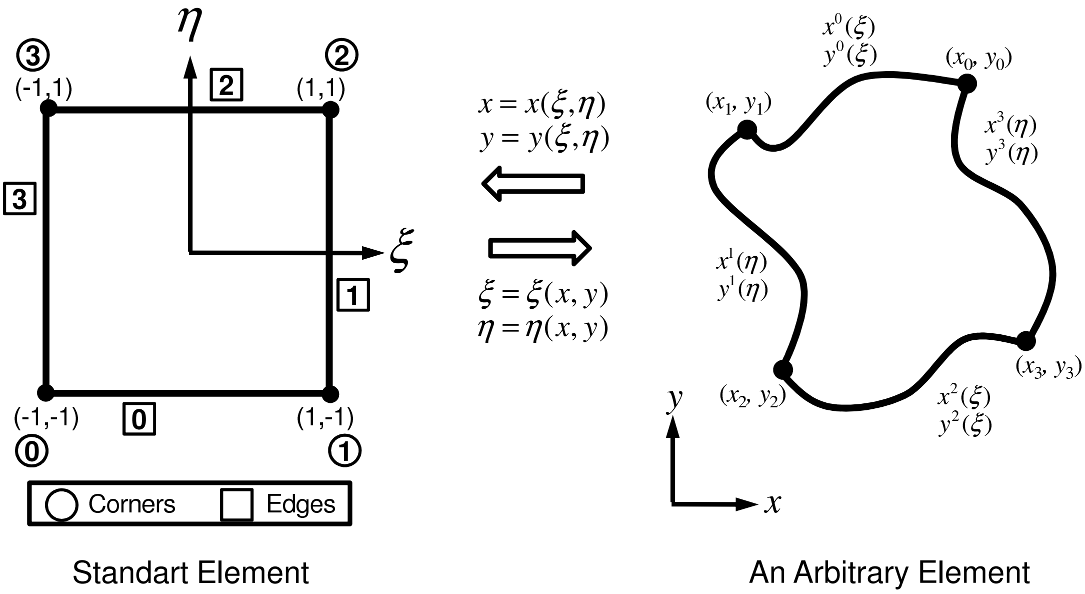


Figure 3.1: Standard and Arbitrary Quadrilateral Elements

arbitrary element can be transformed onto the standard element as follows:

$$I = \int_{\Omega^e} f(x,y)\, dxdy = \int_{\Omega^e_{st}} f^*(\xi,\eta)\, |\mathbf{J}|\, d\xi d\eta \tag{3.2}$$

Here, the integrand $f(x,y)$ is transformed into a new function $f^*(\xi,\eta)$ through appropriate coordinate transformations. The differential area element on the arbitrary element is also transformed onto the standard element by using the Jacobian of the coordinate transformation from standard domain to arbitrary domain. $|\mathbf{J}|$ is the determinant of the Jacobian matrix.

In this study, the coordinate transformation from standard domain to arbitrary domain is defined by using a bilinear mapping. For straight sided arbitrary elements, the bilinear mapping takes the form:

$$x = x(\xi,\eta) = x_0\frac{(1-\xi)(1-\eta)}{2} + x_1\frac{(1+\xi)(1-\eta)}{2} + x_2\frac{(1+\xi)(1+\eta)}{2} + x_3\frac{(1-\xi)(1+\eta)}{2} \tag{3.3}$$

$$y = y(\xi,\eta) = y_0\frac{(1-\xi)(1-\eta)}{2} + y_1\frac{(1+\xi)(1-\eta)}{2} + y_2\frac{(1+\xi)(1+\eta)}{2} + y_3\frac{(1-\xi)(1+\eta)}{2} \tag{3.4}$$

and the Jacobian of the transformation is:

$$\mathbf{J} = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{3.5}$$

The Jacobian matrix can be evaluated by differentiating the expressions in eqns. (3.3) and (3.4) accordingly.

For elements with curved edges, the bilinear mapping can be extended by Gordon-Hall transfinite interpolation introduced by Gordon and Hall [70], as:

$$\begin{aligned} x = x(\xi,\eta) = & \left(x^0(\xi) - x_1\frac{1+\xi}{2}\right)\frac{1-\eta}{2} + \left(x^1(\eta) - x_2\frac{1+\eta}{2}\right)\frac{1+\xi}{2} + \\ & \left(x^2(\xi) - x_3\frac{1-\xi}{2}\right)\frac{1+\eta}{2} + \left(x^3(\eta) - x_0\frac{1-\eta}{2}\right)\frac{1-\xi}{2} \end{aligned} \tag{3.6}$$

$$\begin{aligned} y = y(\xi,\eta) = & \left(y^0(\xi) - y_1\frac{1+\xi}{2}\right)\frac{1-\eta}{2} + \left(y^1(\eta) - y_2\frac{1+\eta}{2}\right)\frac{1+\xi}{2} + \\ & \left(y^2(\xi) - y_3\frac{1-\xi}{2}\right)\frac{1+\eta}{2} + \left(y^3(\eta) - y_0\frac{1-\eta}{2}\right)\frac{1-\xi}{2} \end{aligned} \tag{3.7}$$

where x^d and y^d are the parametric representations of the boundary curve at edge d in term of one of the standard domain coordinates. It can be noted that the standard domain coordinates vary in ξ only along edges 0 and 2 and, in η only along edges 1 and 3. The independent parameters to represent the boundary curve are selected accordingly. In practice the functions x^d and y^d are not available explicitly. Instead, the boundary curve is represented by line segments. In that case, the line segmented geometry is interpolated by the one-dimensional Lagrange interpolating polynomials. The order of interpolation may be same as or different than the expansion order of the element. Once the geometry is available in terms of Lagrange interpolating polynomials, eqns (3.6) and (3.7) can be evaluated.

After the integral (3.2) is transformed onto the standard element, it is must be integrated numerically. For this purpose, Gauss-Legendre numerical quadrature rules are employed. In Gauss-Legendre quadrature, the integral is evaluated by the sum of integrand's values at some particular points called the *quadrature points*, that are multiplied by corresponding weights. For a one-dimensional integral:

$$I = \int_{-1}^{1} f(\xi)\, d\xi = \sum_{k=0}^{q} w_k f(\xi_k) \tag{3.8}$$

q is the order of numerical quadrature. There are several quadrature rules that correspond to different sets of quadrature points and weights. In this study, the Gauss-Lobatto-Legendre (GLL) quadrature rule is used. In GLL quadrature, the quadrature points $\{\xi_k\}_{k=0}^{q}$ are the $q+1$ roots of the equation: $(1-\xi)(1+\xi)P_q^{'}(\xi)$, $\xi \,\epsilon\, [-1, 1]$ where $P_q(\xi)$ is the Legendre polynomial of order q. It is possible to select the weights such that the integral of a polynomial of order

$2q-1$ can be evaluated exactly. The weights that satisfy this requirement are available in the literature [15].

For a two-dimensional integral like the integral in eqn. (3.2) the numerical quadrature can be extended by the tensor product approach outlined in section 2.3. Then the integral can be written as:

$$\int_{\Omega_{st}^{e}} f^{*}(\xi,\eta)\,|\mathbf{J}|\,d\xi d\eta = \int_{\eta=-1}^{1}\int_{\xi=-1}^{1} f^{**}(\xi,\eta), d\xi\, d\eta = \sum_{l=0}^{q_\eta} w_l^{\eta} \sum_{k=0}^{q_\xi} w_k^{\xi} f^{**}(\xi_k,\eta_l) \tag{3.9}$$

where the determinant of Jacobian is lumped into f^{**}. It can be noted that the integrand f^{**} involves multiplication of the approximate solution expressions with the first derivatives of the approximate solutions as well as the determinant of Jacobian. Therefore the integrand is a polynomial of order greater than $2q-1$ in both space dimensions. As a result, GLL quadrature does not evaluate the integral exactly. However it can be shown that the error in numerical integration is in the same order of magnitude as the error in interpolation of the unknowns and therefore is not significant [15].

When the set of collocation points used to construct the Lagrange interpolating functions in section 2.3 is compared with the set of quadrature points introduced here, it is seen that they involve the same points. In other words, the quadrature points coincide with the nodal points. The motivation in selecting the GLL grid as the set of collocation points is due to clustered nature of the GLL grid which results in Lagrange interpolating polynomials with good approximation properties. In specific, the Lagrange interpolating polynomials defined on a GLL grid are free of unbounded oscillations that emerge as the approximation order is increased i.e. the Runge phenomenon [73]. The motivation behind selecting the quadrature points same as the nodal points is to improve the efficiency of implementation. In particular, the computational complexity of the elemental system calculation is reduced to $O(p^4)$ from $O(p^6)$ by employing coincident nodal and quadrature points.

3.2.2 Elemental Node Ordering

As stated in Section 2.3, the approximate solution is defined over each element as a basis expansion and the expansion coefficients are referred as the elemental degrees of freedom (DOFs).

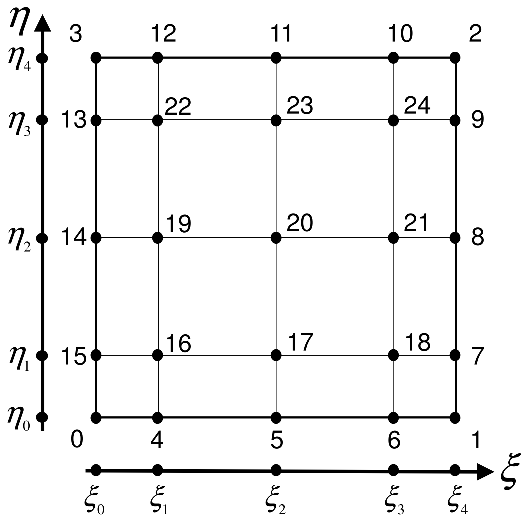


Figure 3.2: Elemental node ordering

In Section 2.3, the approximate solution over an element was written as:

$$\mathbf{u}^e(\mathbf{x}) = \sum_{k=0}^{N^e_{DOF}-1} \mathbf{u}^e_k\, \phi_i(\mathbf{x}) \qquad \text{OR} \qquad \mathbf{u}^e(\mathbf{x}) = \mathcal{U}^{eT} \boldsymbol{\phi}(\mathbf{x}) \tag{3.10}$$

where

$$\mathbf{u}^e_k = (u_k, v_k, p_k, \omega_k)^T \quad \text{and} \quad \mathcal{U}^e_{kn} = \left(\mathbf{u}^e_k\right)_n, \qquad k = 0, 1, \cdots, N^e_{DOF}, \quad n = 0, 1, \cdots, N_u$$

u_k, v_k, p_k, and ω_k are the unknown values evaluated at nodal point k. It can observed that an ordering of the elemental DOFs or the nodal points is assumed in eqn. (3.10). While this ordering can be arbitrary, an ordering that assigns the nodes on the elemental boundary and the nodes at the interior of the element to separate groups is preferred in this study. Because this type of ordering, referred as *boundary-interior decomposition* [15] is advantageous in implementing efficient linear system solution algorithms such as static condensation that will be discussed in section 3.2.4. In boundary-interior decomposition approach, the corner nodes are numbered first, followed by the edge nodes. The interior nodes are numbered last. For illustration one can consider the quadrilateral element presented in figure 3.2. The solution over the element is represented by expansion order 4 in both space dimensions. That is, $p_\xi = p_\eta = 4$. Then it is composed of $N^e_{DOF} = (p_\xi + 1)(p_\eta + 1) = 25$ elemental DOFs to calculate. These 25 elemental DOFs are the unknown values evaluated at 25 nodal points ordered as presented in figure 3.2. First the corner nodes are numbered starting from the node at corner 0. After all corner nodes are numbered, the edge nodes are numbered beginning from edge 0. Across each edge, the edge nodes are numbered in the order they are encountered during a CCW travel around the element. Finally when all edge nodes are also numbered, the interior nodes are numbered. The numbering of interior nodes among themselves is not important. In this case, the interior nodes are numbered in an η-major ordering following the collocation point ordering of the underlying GLL grids. The nodes and the associated elemental DOFs can be grouped as follows:

$$\left.\begin{array}{l} 0-3 : \text{Corner Nodes} \\ 4-15 : \text{Edge Nodes} \end{array}\right\} \quad \text{Boundary Nodes}$$
$$16-24 : \qquad\qquad \text{Interior Nodes}$$

It should be noted that the ordering achieved by the boundary-interior decomposition has a direct affect on the elemental solution vector $\mathbf{U}^e$ of the elemental system. The elemental solution vector is obtained by concatenating the rows of $\mathcal{U}^e$ in a single row vector in the order of elemental DOF numbering. Then the elemental solution vector can be decomposed into boundary and interior solution vectors, $\mathbf{U}_b^e$ and $\mathbf{U}_i^e$ respectively as:

$$\mathbf{U}^e = \begin{pmatrix} \mathbf{U}_b^e \\ \mathbf{U}_i^e \end{pmatrix} \qquad \text{where} \qquad \begin{array}{l} \mathbf{U}_b^e = \left((u, v, p, \omega)_0^T, (u, v, p, \omega)_1^T, \cdots, (u, v, p, \omega)_{15}^T\right)^T \\ \mathbf{U}_i^e = \left((u, v, p, \omega)_{16}^T, (u, v, p, \omega)_{17}^T, \cdots, (u, v, p, \omega)_{24}^T\right)^T \end{array} \tag{3.11}$$

The decomposition into boundary and interior components is possible also for the elemental stiffness matrix and elemental force vector as will be discussed in section 3.2.4.

3.2.3 Constrained Approximation Method

Use of a hp-type non-conforming grid with LSSEM leads to two types of non-conformities in element interfaces. h-type non-conforming interfaces occur where an element becomes neighbor to more than one element at a single edge. That is, the interfacing elements are geometrically non-conforming. h-type non-conformities are a result of using a non-conforming Cartesian grid. Another type of non-conformity is the p-type non-conformity that arises due to the use of high order expansions on individual elements. In p-type non-conforming interfaces two elements are geometrically conforming at the adjacent edges but they use different orders of expansions to approximate the unknowns on those edges. The solver developed in this study handles both types of non-conformities by the *constrained approximation method* [9, 71].

In constrained approximation method, one of the interfacing edges are declared as *active* and the nodal unknowns on active edge are used in global assembly process. The nodal unknowns on the other (*passive*) edge(s) are not a part of global solution. When they are needed for elemental operations, they are interpolated from the unknowns on the *active* edge. Degree of the interpolation is the same as the expansion order used on the *active* edge. In this study, activeness of edges at an h-type non-conformity are decided according to *long rule* where the edge which is longer is made active. Similarly, activeness of edges at a p-type non-conformity, are decided according to *minimum rule* where the edge with lower expansion order is set as active. For the details of constrained approximation method, one can refer to Sert and Beskok [74].

Figure 3.3 shows a simple hp-type non-conforming grid. One of the elements, labeled as *e*, is chosen to illustrate the constrained approximation method. The numbers encapsulated by squares are the edge labels. The element employs expansion order $p = 4$ in both space dimensions. It is adjacent to elements of polynomial orders $p = 3, 6, 4$ and 7 on edges 0, 1, 2 and 3 respectively. It can be observed that edge 0 features a p-type non-conforming interface while edges 1 and 3 feature h-type non-conforming interfaces. Edge 2 is a conforming interface. When the minimum and long rules are applied, it is observed that edge 0 of the element is passive since $p = 4 > 3$. Similarly edge 3 is passive regardless of the expansion order since it is adjacent to an element with longer edge. While edge 1 also features an h-type non-conformity, it is not passive since this time, the element *e* is the element with longer edge.

The constrained approximation procedure on the passive edges (edges 0 and 3) of element *e* is illustrated in figure 3.4. The procedure involves the interpolation of unknowns at constrained nodes, shown by squares, from the basis functions on the active edge of the adjacent element. It can be noted that the basis functions reduce to the corresponding one-dimensional Lagrange interpolants across the edges. Since the expansion orders of the elements adjacent to element *e* at edges 0 and 3 are $p_0 = 3$ and $p_3 = 7$, the approximate solutions over the active edges are represented by linear combinations of $p_0 + 1 = 4$ and $p_3 + 1 = 8$ Lagrange interpolants respectively. The local coordinates of the active edges are represented by ξ and η. The Lagrange interpolants pass through the 4-point and 8-point GLL grids defined on those local coordinates. Only $L_1(\xi)$ on edge 0 and $L_3(\eta)$ on edge 3 are shown in figure 3.4 for simplicity. The local coordinates of the passive edges are represented by $\bar{\xi}$ and $\bar{\eta}$. For

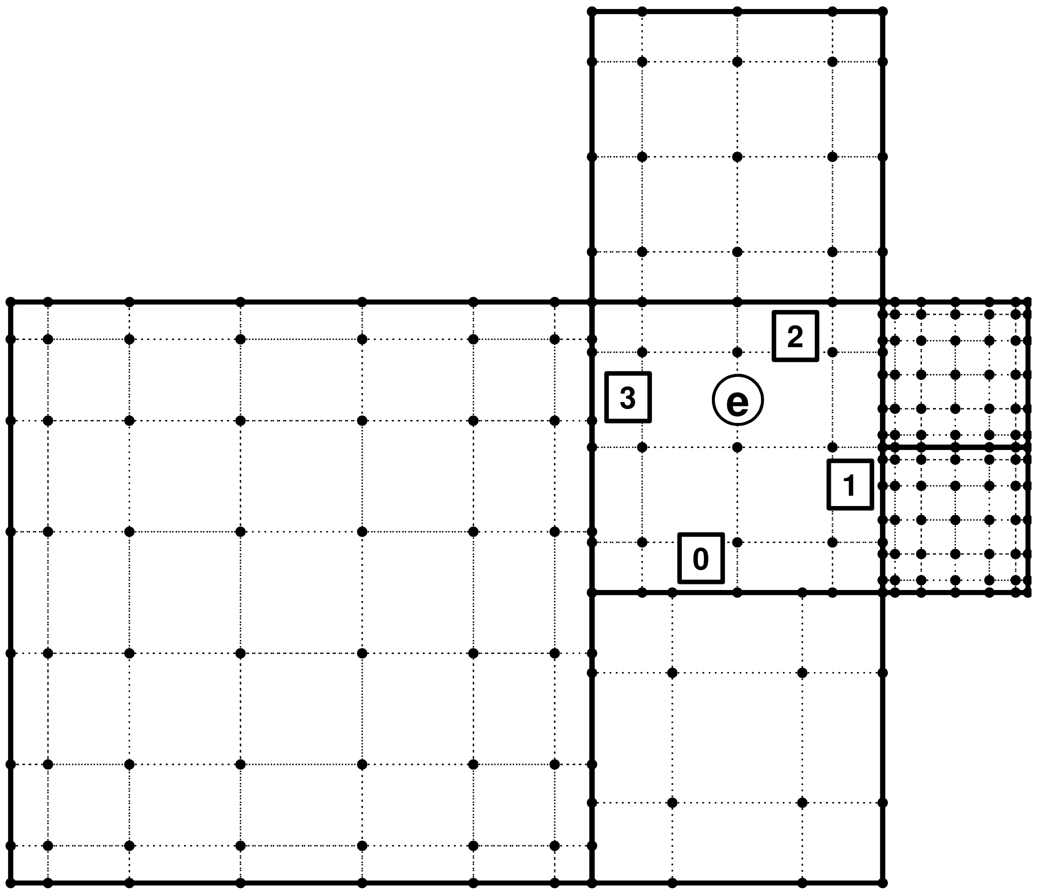


Figure 3.3: A simple hp-type non-conforming grid

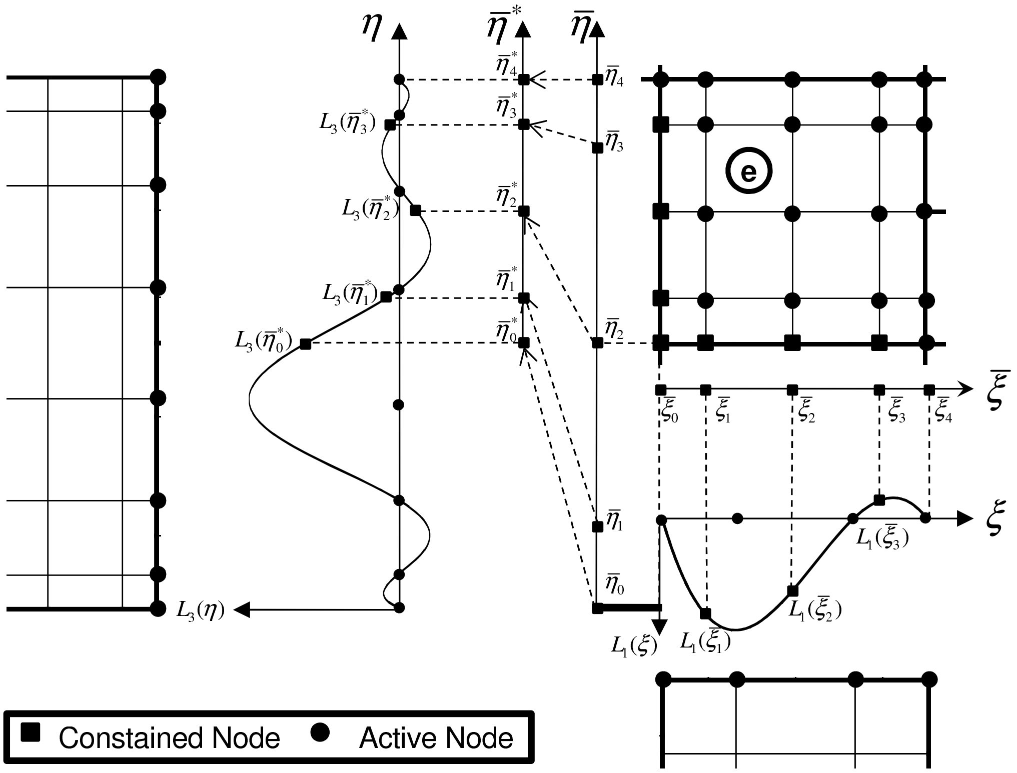


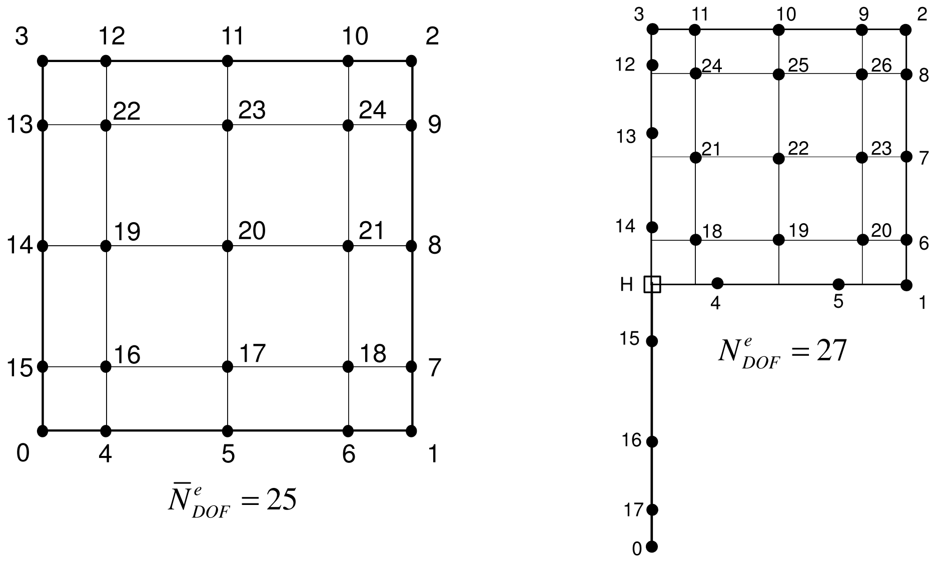
Figure 3.4: Constrained approximation at edges 0 and 3 of element *e*

constrained approximation, the Lagrange interpolants are evaluated at the local coordinates of the constrained nodes. For instance, the value of the unknown at the mid-point node of edge 0 is calculated by a linear combination of Lagrange interpolants: $L_0(\bar{\xi}_2), L_1(\bar{\xi}_2), L_2(\bar{\xi}_2)$ and $L_3(\bar{\xi}_2)$. For h-type non-conformities an additional step is needed. The local coordinate of passive edge 3 ($\bar{\eta}$) must be mapped to the upper half of the local coordinate of the active edge of the larger element (η). This mapping is performed as follows:

$$\bar{\eta}^* = \frac{1+\bar{\eta}}{2} \tag{3.12}$$

Then the Lagrange interpolants are evaluated at the corresponding $\bar{\eta}^*$ coordinates.

Figure 3.5 illustrates the element *e* before and after the constrained approximation. The elemental node numbering of the element without regard to constrained approximation was presented in figure 3.2 and is repeated in figure 3.5a. This state of the element can be regarded as the *unconstrained state* since the nodes that to be passive are still present in the node numbering. A new elemental node numbering involving the active nodes only is provided in figure 3.5b. This state can be referred as the *constrained state* for obvious reasons. At this point, the necessity of constrained approximation should be recognized. The unconstrained state repre-



(a) Unconstrained state

(b) Constrained state

Figure 3.5: Elemental node numbering before and after constrained approximation

sents the state of the element when it is considered regardless of it's surroundings. Therefore, the unconstrained state is suitable for the isolated elemental operations like calculation of the elemental system. In particular, the unconstrained state is in accordance with the underlying quadrature grid. However the unknown values only at active nodes are available from the global solution. Therefore, the aim of the constrained approximation is to represent the unknowns at the constrained nodes in terms of the unknowns at the active nodes.

As an example of constrained approximation, one can consider the approximation of one of the dependent variables, say x-velocity, at constrained nodes, for instance node 14 on edge 3 and node 6 on edge 0 (See figure 3.5a), in terms of x-velocities at active nodes along the corresponding edges as provided in figure 3.5b. The values of the x-velocity at constrained nodes have bars over them, following the general convention.

$$\overline{u}^e_{14} = u^e_0 L_0(\overline{\eta}^*_1) + u^e_{17} L_1(\overline{\eta}^*_1) + \cdots + u^e_{12} L_6(\overline{\eta}^*_1) + u^e_3 L_7(\overline{\eta}^*_1) \tag{3.13}$$

$$\overline{u}^e_6 = \overline{u}^e_H L_0(\overline{\xi}_3) + u^e_4 L_1(\overline{\xi}_3) + u^e_5 L_6(\overline{\xi}_3) + u^e_1 L_7(\overline{\xi}_3) \tag{3.14}$$

It can be seen in eqn. (3.13) that the unknown values on the right hand side are available from the global solution and the basis function values can be evaluated. So the unknowns at any constrained node on edge 3 can be calculated from the knowledge of unknowns at the constrained state. When it comes to the edge 0, it is observed that the right hand side of eqn. (3.14) involves an unknown value at a constrained node,$\overline{u}^e_H$, that is not present in the constrained state. This constrained node is a hanging node, indicated by H in figure 3.5b and it is a result of occurrence of h-type and p-type non-conforming interfaces at edges of the element connecting to the same corner. In this case it is still possible to apply the constrained approximation by interpolating the value at the hanging node from the unknowns on edge 3 first and then using it in eqn. (3.14).

$$\overline{u}^e_H = u^e_0 L_0(\overline{\eta}^*_0) + u^e_{17} L_1(\overline{\eta}^*_0) + \cdots + u^e_{12} L_6(\overline{\eta}^*_0) + u^e_3 L_7(\overline{\eta}^*_0) \tag{3.15}$$

The unconstrained and constrained states involve different number of elemental DOFs: $\overline{N}^e_{DOF} = 25$ and $N^e_{DOF} = 27$. Since the elemental node numberings are also different, two elemental solution vectors are introduced: $\overline{\mathbf{U}}^e$ that is the unconstrained elemental solution vector and $\mathbf{U}^e$ that is the constrained elemental solution vector. The constrained approximation of the unknowns at individual constrained nodes can be extended to elemental solution vectors by assembling the equations of the form of eqns. (3.13), (3.14), and (3.15) into a elemental conformity matrix $\mathbf{C}^e$ and representing the transformation from the constrained state to unconstrained state as follows:

$$\overline{\mathbf{U}}^e = \mathbf{C}^e \mathbf{U}^e \tag{3.16}$$

As stated before, the elemental system is calculated using the unconstrained state of the element. Since, the globals system of equations is solved for the constrained unknown vector $\mathbf{U}$, the elemental system must be modified before assembling it into global system:

$$\mathbf{K}^e \overline{\mathbf{U}}^e = \mathbf{F}^e \qquad \Longrightarrow \qquad \left[\mathbf{C}^{e^T} \mathbf{K}^e \mathbf{C}^e\right] \mathbf{U}^e = \mathbf{C}^{e^T} \mathbf{F}^e \tag{3.17}$$

After the modification above in eqn. (3.17), the solution of the linear system of equations is straightforward.

3.2.4 Static Condensation

The boundary-interior decomposition introduced in section 3.2.2 serves as a convention for elemental node ordering. It is also consistent with the fact that the boundary DOFs of an element are in interaction with the boundary DOFs of the surrounding elements while the interior DOFs of the element are isolated from the DOFs of the surrounding elements. Based on this observation, the elemental and global system of equations can be decomposed into two sets of equations that are decoupled from one another through a process called static condensation [15].

It was stated in section 3.2.2 that the boundary-interior decomposition approach enables decomposition of the elemental solution vector into boundary and interior DOF components as illustrated by eqn. (3.11). The same approach can be realized for the whole elemental system as follows:

$$\begin{bmatrix} \mathbf{K}_b^e & \mathbf{K}_c^e \\ \mathbf{K}_c^{eT} & \mathbf{K}_i^e \end{bmatrix} \begin{pmatrix} \mathbf{U}_b^e \\ \mathbf{U}_i^e \end{pmatrix} = \begin{pmatrix} \mathbf{F}_b^e \\ \mathbf{F}_i^e \end{pmatrix} \tag{3.18}$$

where subscript b stands for the boundary DOF contributions, i stands for the interior DOF contributions and c stands for the coupling between elemental boundary and interior DOFs. The solution of $\mathbf{U}_b^e$ and $\mathbf{U}_i^e$ can be decoupled by pre-multiplying eqn. (3.18) by the following matrix:

$$\begin{bmatrix} \mathbf{I} & \mathbf{K}_c^e \mathbf{K}_i^{e-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{3.19}$$

to arrive at:

$$\begin{bmatrix} \mathbf{K}_b^e - \mathbf{K}_c^e \mathbf{K}_i^{e-1} \mathbf{K}_c^{eT} & \mathbf{0} \\ \mathbf{K}_c^{eT} & \mathbf{K}_i^e \end{bmatrix} \begin{pmatrix} \mathbf{U}_b^e \\ \mathbf{U}_i^e \end{pmatrix} = \begin{pmatrix} \mathbf{F}_b^e - \mathbf{K}_c^e \mathbf{K}_i^{e-1} \mathbf{F}_i^e \\ \mathbf{F}_i^e \end{pmatrix} \tag{3.20}$$

Eqn. (3.20) can now be decomposed into two sets of equations. The first set is for the solution of elemental boundary DOFs that is to be solved first:

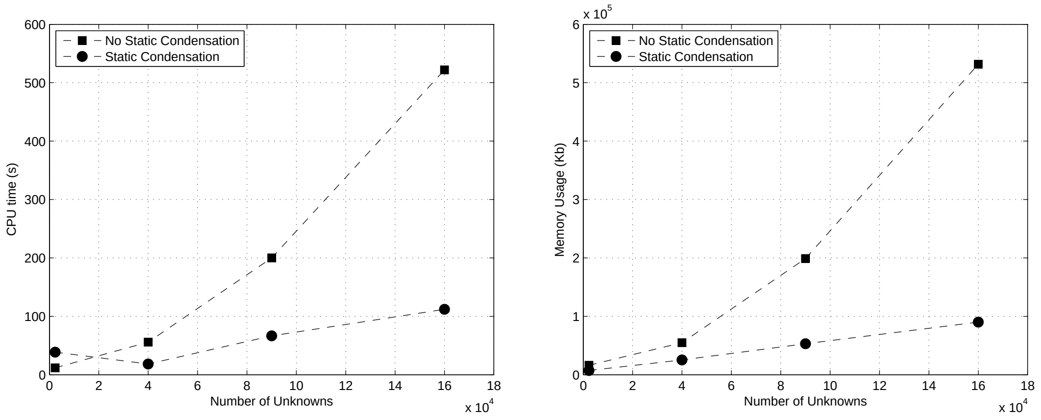
$$\left[\mathbf{K}_b^e - \mathbf{K}_c^e \mathbf{K}_i^{e-1} \mathbf{K}_c^{eT}\right] \mathbf{U}_b^e = \left(\mathbf{F}_b^e - \mathbf{K}_c^e \mathbf{K}_i^{e-1} \mathbf{F}_i^e\right) \tag{3.21}$$

After the solution of eqn. (3.21), a second set of equations are solved for the elemental interior DOFs:

$$\mathbf{K}_i^e \mathbf{U}_i^e = \left(\mathbf{F}_i^e - \mathbf{K}_c^{eT} \mathbf{U}_b^e\right) \tag{3.22}$$

where $\mathbf{U}_b^e$ is substituted to right hand side from the previously solved elemental boundary solution vector.

Static condensation essentially decomposes a system of equation of larger size into two systems of equations of smaller size. For illustration, the original elemental system of corresponding to the element illustrated in figure 3.5b is of size $N_u^e = N_u \times N_{DOF}^e = 4 \times 27 = 108$. Of all 27 DOFs 18 of them are boundary DOFs and 9 of them are interior DOFs leading to linear systems of equations of size $N_{bu}^e = 4 \times 18 = 72$ and $N_{iu}^e = 4 \times 9 = 36$ respectively. Here, N_{bu}^e is the number of elemental boundary unknowns and N_{iu}^e is the number of elemental interior unknowns. Since the computation time increases rapidly with the system size, (matrix-vector multiplication time is usually proportional to N^3) decomposing it into smaller



(a) CPU Time

(b) Memory Use

Figure 3.6: Comparison of CPU time and memory uses with and without static condensation

systems decreases the time spent for the linear system solution. Static condensation also reduces the memory requirements since only one of the smaller systems are kept in the memory at a time.

The above discussion is complemented with numerical experiments. Comparison of computational resources required to solve a benchmark problem with and without static condensation are presented in figure 3.6. The benchmark problem is the two-dimensional lid-driven cavity problem that will be introduced in section 4.2. The problem is solved with a steady-state LSSEM solver on grids with increasing resolution. Grid resolution is increased by both h-type and p-type refinements. The number of unknown degrees of freedoms is reported as a measure of computational load. CPU times and memory uses of various runs with static condensation and without static condensation are illustrated in figures 3.6a and 3.6b respectively. It is observed that static condensation results in lower values for both the CPU time and the memory use. The ratio of the computational resources required for solution without and with static condensation increases as the computational load increases, making static condensation an attractive approach for large-scale computations.

As stated in section 2.3.5, the LSSEM flow solver solves the linear system of equations with an EBE procedure where elemental systems of all elements are present in the memory at the same time. An alternative is the matrix-free method where the elemental systems are not kept in the memory, instead recalculated at each linear solver iteration. The matrix-free method is definitely more memory saving. However it is practical only if the calculation cost of the elemental system is small enough to enable recalculation at each linear solver iteration. When the steps to perform static condensation are examined, it is seen that static condensation involves CPU intensive tasks such as the inversion of the symmetric, positive-definite matrix $\mathbf{K}_i^e$ to arrive at the final system to be solved. As a result, static condensation is associated with great computational costs for the calculation of the elemental system. This aspect hinders the use of the static condensation with matrix-free method and motivates the approach of keeping the elemental systems in the memory through the linear system solution. It is important to note that for two-dimensional problems, the memory requirement is not a major issue and CPU time dominates the performance considerations. However memory requirements become excessive for three-dimensional problems. Therefore the matrix-free method, without static condensation, may be the only option for the solution of 3-dimensional problems with the EBE procedure.

3.2.5 Adaptive Refinement Criterion

In this study, two alternative posteriori error estimates are considered to drive the adaptive refinement procedures. Namely, the error estimate based on least-squares functional, referred as $\tilde{e}_{ls}$, and the error estimate based on spectral coefficients, referred as $\tilde{e}_{spectral}$.

It can be observed from eqn. (2.34) in chapter 2 that least-squares functional provides a measure of residuals of the governing equations. Therefore it can be used as an error indicator. An error estimate based on least-squares functional was employed by Jiang and Carey [60] to drive h-type refinement. The error estimate proposed in that study is:

$$\tilde{e} = \frac{\mathcal{J}^e}{A^e} \tag{3.23}$$

where

$$\tilde{e} = \mathcal{J}^e = \frac{1}{2}\mathcal{I}^e = \frac{1}{2}\int_{\Omega^e} \mathbf{R}^T\,\mathbf{R}\,d\Omega^e \tag{3.24}$$

The $1/2$ factor is due to different definitions of the least-squares functional. $\mathcal{I}^e$ is the elemental least-squares functional. It Jiang and Carey's work [60], the elemental least-squares functional is normalized by the element area. It should be noted that error estimate (3.23) is proportional to the square of the magnitude of the residual. It also is not normalized with respect to the magnitude of the unknowns over each element. Therefore use of this error estimate provides a measure of the true error (in fact, the square of the true error) while the relative error is of interest in engineering applications.

In the present study, the error estimate (3.23) is modified as follows to provide a measure of the residual, but not it's square and to account for the magnitude of the unknowns over each element.

$$\tilde{e}_{ls} = \frac{\sqrt{\mathcal{I}^e}}{||\tilde{\mathbf{u}}^e||_{\mathbf{H}^1(\Omega^e)}} = \frac{||\mathbf{R}^e||_{\mathbf{L}^2(\Omega^e)}}{||\tilde{\mathbf{u}}^e||_{\mathbf{H}^1(\Omega^e)}} \tag{3.25}$$

where $\mathbf{R}^e$ and $\tilde{\mathbf{u}}^e$ are the elemental residual and approximate solution vectors respectively. The elemental L^2- and H^1-norms are defined by eqns. (2.20) and (2.22).

Error estimate is a valid measure of the error since it can be shown that the H^1-norm of the error $||\mathbf{e}||_{\mathbf{H}^1(\Omega)} = ||\mathbf{u} - \tilde{\mathbf{u}}||_{\mathbf{H}^1(\Omega)}$ approaches zero as $||\mathbf{R}||_{\mathbf{L}^2(\Omega)}$ approaches zero [60].

The error estimate (3.25) is practical to calculate once the elemental system (2.52) and the approximate solution is available:

$$\tilde{e}_{ls} = \frac{\sqrt{\mathcal{I}^e}}{||\tilde{\mathbf{u}}^e||_{\mathbf{H}^1(\Omega^e)}} = \frac{1}{||\tilde{\mathbf{u}}^e||_{\mathbf{H}^1(\Omega^e)}} \sqrt{\mathbf{U}^{eT}\mathbf{K}^e\mathbf{U}^e - 2\mathbf{U}^{eT}\mathbf{F}^e + \int_{\Omega^e} \mathbf{f}^T\mathbf{f}\,d\Omega^e} \tag{3.26}$$

The second error estimate used in this study is based on spectral coefficients of the approximate solution's Legendre expansion. This error estimate is proposed by Henderson [59] to drive h-type adaptive refinement on a non-conforming Cartesian grid composed of high order elements. The error estimate is based on the fact that multiple polynomial expansions can be used to represent the approximate solution over each element. As stated in chapter 2 the present study employs a nodal expansion where the basis functions are Lagrange interpolating polynomials. The approximate solution can also be written as a hierarchical expansion where the basis functions are, for instance, Legendre polynomials. A favorable property of the hierarchical expansion is that, when the polynomial order p is sufficiently high ($p > 5$ in practice [59]) the expansion coefficient associated with the Legendre polynomial of the highest order can be used as a measure of the approximation error in the basis expansion. The approximate solution of one of the unknowns, let it be u, can be written over each element as:

$$u^e(\mathbf{x}) = \sum_{k=0}^{N^e_{DOF}-1} u^e_k \phi_k(\mathbf{x}) \qquad = \sum_{k=0}^{N^e_{DOF}-1} a_k \psi_k(\mathbf{x}) \tag{3.27}$$

$$= \underbrace{\sum_{i=0}^{p}\sum_{j=0}^{p} u^e_{ij} L_i(x) L_j(y)}_{\text{Nodal Expansion}} = \underbrace{\sum_{i=0}^{p}\sum_{j=0}^{p} a_{ij} P_i(x) P_j(y)}_{\text{Hierarchical Expansion}} \tag{3.28}$$

L is the one-dimensional Lagrange interpolating polynomial of order p and P_i is the Legendre polynomial of order i. The expansion coefficients a_{ip} and a_{pi}, $i = 0, 1, \cdots, p$ are associated with the Legendre polynomials of highest order. An equivalent expansion coefficient $\hat{a}_p$ can be obtained by lumping these expansion coefficients:

$$\hat{a}_p = |a_{pp}| + \sum_{i=0}^{p-1} |a_{ip}| + |a_{pi}| \tag{3.29}$$

The error estimate is defined by a suitable normalization of the equivalent expansion coefficient with respect to the magnitude of the unknown over the element.

$$\tilde{e}^u_{spectral} = \frac{\hat{a}_p}{||u^e||_{H^1(\Omega^e)}} \tag{3.30}$$

The spectral expansion coefficients a_{ij} can be calculated from the approximate solution by using a transformation from the nodal expansion to the hierarchical expansion. Each basis function of the nodal expansion can be written in terms of the basis functions of the hierarchical expansion:

$$\phi_k(\mathbf{x}) = \sum_{l=0}^{N^e_{DOF}-1} b_{kl}\psi_l(\mathbf{x}) \tag{3.31}$$

or in vector notation,

$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{B\Psi}(\mathbf{x}) \tag{3.32}$$

Also representing the nodal and hierarchical expansions in vector notation and substituting eqn. (3.32),

$$u^e(\mathbf{x}) = \mathbf{u}^{eT}\mathbf{\Phi}(\mathbf{x}) \quad = \mathbf{a}^T\mathbf{\Psi}(\mathbf{x}) \tag{3.33}$$

$$= \mathbf{u}^{eT}\mathbf{B\Psi}(\mathbf{x}) \;\; = \mathbf{a}^T\mathbf{\Psi}(\mathbf{x}) \tag{3.34}$$

Then the spectral expansion coefficient vector **a** can be obtained as:

$$\mathbf{a} = \mathbf{B}^T\,\mathbf{u}^e \tag{3.35}$$

In practice the components of the transformation matrix b_{kl} are made available in a data file and read in the beginning of the program execution. Calculation of the error estimate (3.30) is straightforward once the spectral expansion coefficients are available.

The adaptive refinement strategy adopted in this study is to keep the elemental errors between prescribed error bounds. Therefore p-type adaptive refinement is performed to obtain and maintain the following condition

$$\varepsilon_{min} \le \tilde{e} \le \varepsilon_{max} \tag{3.36}$$

where ε_{min} is the error lower bound and ε_{max} is the error upper bound. Typical values for the error bounds are between 10^{-5} and 10^{-2}. While the error bounds may give an idea of the order of magnitude of the accuracy desired, they are selected in this study by numerical experimentation. The p-type adaptive refinement is performed isotropically, i.e. by changing the expansion orders in both space dimensions by the same increments/decrements. The expansion order increment/decrement is 2 unless otherwise is specified.

There are several differences between the two error estimates. First of all, $\tilde{e}_{ls}$ is a measure of the error in the overall solution in regard to how well the approximate solution satisfies the governing equations. In contrary, $\tilde{e}_{spectral}$ measures the error in one of the dependent variables (u, v, p, ω) in regard to how well the approximate solution interpolates the exact solution. Therefore it is complemented by a superscript to indicate the dependent variable it is associated with. In this study, $\tilde{e}^{u}_{spectral}$, that is associated with the x-velocity is used. The error estimates associated with other dependent variables are also reported for comparison. Another difference is that, as $\tilde{e}_{ls}$ approaches zero, it can be understood that the approximate solution reaches to the exact solution that satisfies the governing equations exactly. This is not the case with $\tilde{e}_{spectral}$. As $\tilde{e}_{spectral}$ approaches zero, it should be understood that the solution approaches to a grid-independent solution upon p-type refinement. The grid independent solution is not necessarily the exact solution. Finally, $\tilde{e}_{ls}$ can be applied to solutions with low and high order elements. However $\tilde{e}_{spectral}$ requires the use of high order elements. The performance of the error estimates are compared though various benchmark solutions in chapter 4

CHAPTER 4

VALIDATION OF LSSEM FLOW SOLVER

4.1 Kovasznay Flow

In SEM solutions of smooth problems, the error associated with the numerical solution decreases exponentially as the expansion order is increased. The resulting convergence pattern is referred as *spectral (exponential) convergence* [5]. It is common practice to investigate SEm flow solvers' accuracy and the numerical stability with respect to expansion order to verify the spectral convergence characteristics. See, for instance [15, 34]. In this section, the spectral convergence characteristics of the LSSEM flow solver for various grid configurations are illustrated by solving the Kovasznay flow.

Kovasznay flow, presented by Kovasznay [75], is a laminar, incompressible flow problem whose steady-state solution is available in analytical form. The availability of the exact solution makes it possible to measure the true error associated with the numerical solution and investigate the convergence properties of the solver.

The exact solution is:

$$u = 1 - e^{\lambda x} cos\left(2\pi y\right) \tag{4.1a}$$

$$v = \frac{\lambda}{2\pi} e^{\lambda x} sin\left(2\pi y\right) \tag{4.1b}$$

$$p = p_0 + \frac{1}{2}\left(1 - e^{2\lambda x}\right) \tag{4.1c}$$

$$\omega = \left(\frac{\lambda^2}{2\pi} - 2\pi\right) e^{\lambda x} sin\left(2\pi y\right) \tag{4.1d}$$

where $\lambda = 1/2\nu - \left[\left(1/4\nu^2\right) + 4\pi^2\right]^{1/2}$, ν being the kinematic viscosity. Note that the pressure is specified only up to a constant reference pressure p_0. While the problem can be solved on an arbitrary domain for any Reynolds number, The solutions in this study are performed on a rectangular domain: $[-0.5, 1] \times [1, 1.5]$ for $Re = 1/\nu = 40$[1]. The streamlines and the x-velocity contours within the specified region are presented in figures 4.1a and 4.1b. The flow field involves a wake pattern similar to a flow behind an array of cylinders.

In order to study the convergence properties of the solver, solutions on conforming, p-type non-conforming and h-type non-conforming grids are performed as follows:

1. Conforming Grid: In this study, grid 1, illustrated in figure 4.1c is used. This grid has 8 elements. Same expansion order, p is used for all elements and in both space dimensions, resulting in conforming element interfaces.

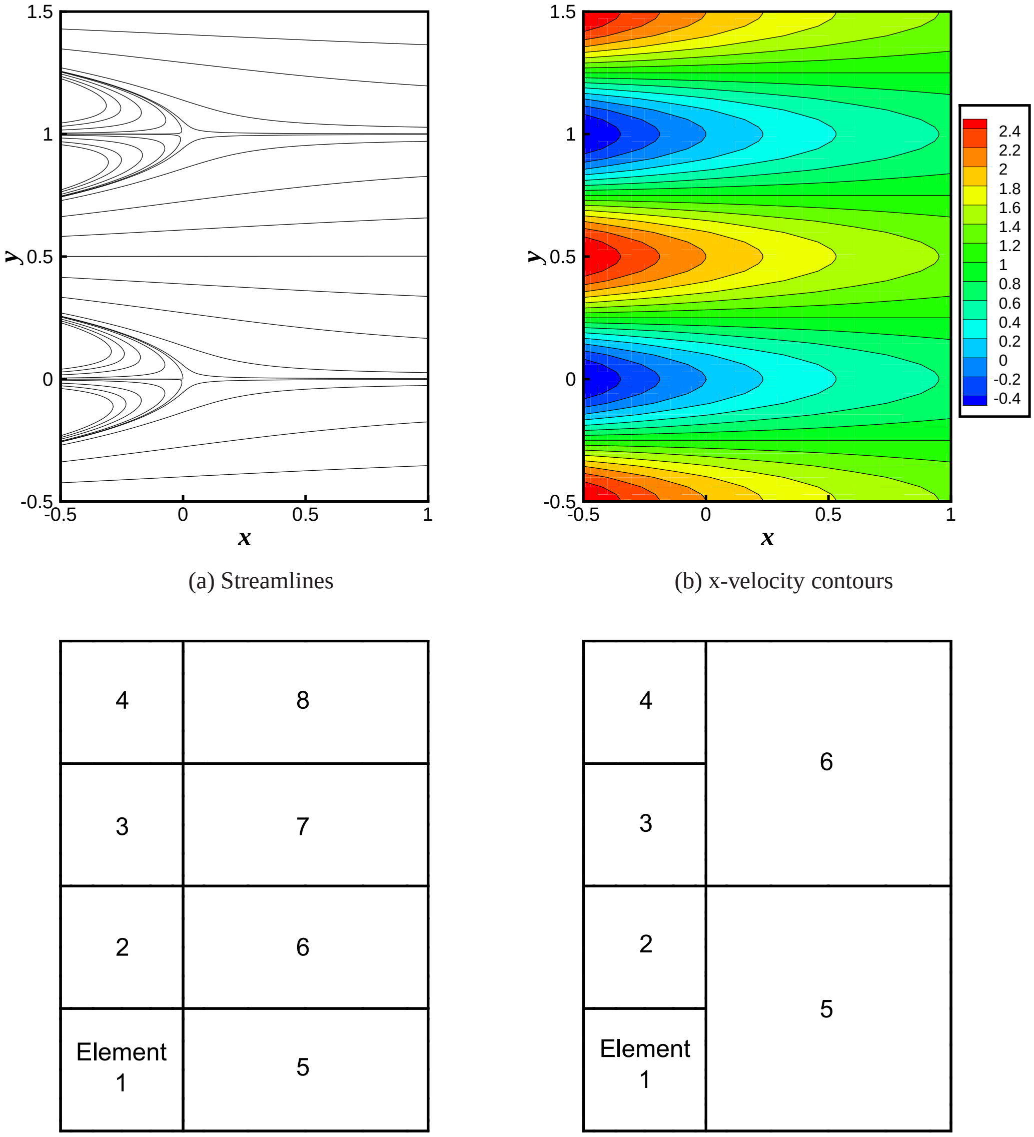
2. p-type Non-conforming Grid: Again, grid 1 is used. However this time, expansion orders of elements are altered such that elements 1, 4, 6 and 7 have expansion orders p while elements 2, 3, 5 and 8 have expansion orders $p - 2$. Same expansion order is used in both space dimensions. This setup results in p-type non-conforming interfaces.

3. h-type Non-conforming Grid: In this study, grid 2, illustrated in figure 4.1d is used. Here, 2 h-type non-conforming interfaces lying in y-direction are present. Elements on the left half of the domain have expansion orders p in both space dimensions. The two large elements on the right hand side also have expansion order p in the x-direction. $p + 4$ is used in y-direction to compensate for the large element size in that direction.

For all the grids described above, a series of solutions are performed for various expansion orders by starting with $p = 6$ and incrementing by 2 until $p = 14$. The steady-state solutions are obtained by marching in time starting from a stagnant flow field. A time interval of 30 seconds is found adequate to reach the steady state within the accuracy of spatial resolution for all expansion orders. Exact values of the velocity field are specified at the boundaries. The reference pressure is set as zero at the origin ($x = 0$, $y = 0$).

In this study the trends of errors with respect to increasing expansion order is investigated. However, for non-conforming grids, the expansion order varies from one element to other. So the expansion order of an individual element is not a representative value for the expansion

[1] In this study, the dimensional form of governing equations are used. For physical interpretation, the results presented here can be followed by appropriate SI units. For instance, velocity components have the unit m/s and pressure has the unit Pa. The focus of this study is on validating the numerical performance of the flow solver. Therefore the units are omitted whenever they are found irrelevant.



(a) Streamlines

(b) x-velocity contours

(c) Grid 1

(d) Grid 2

Figure 4.1: Kovasznay flow. Exact solution and the computational grids used in this study.

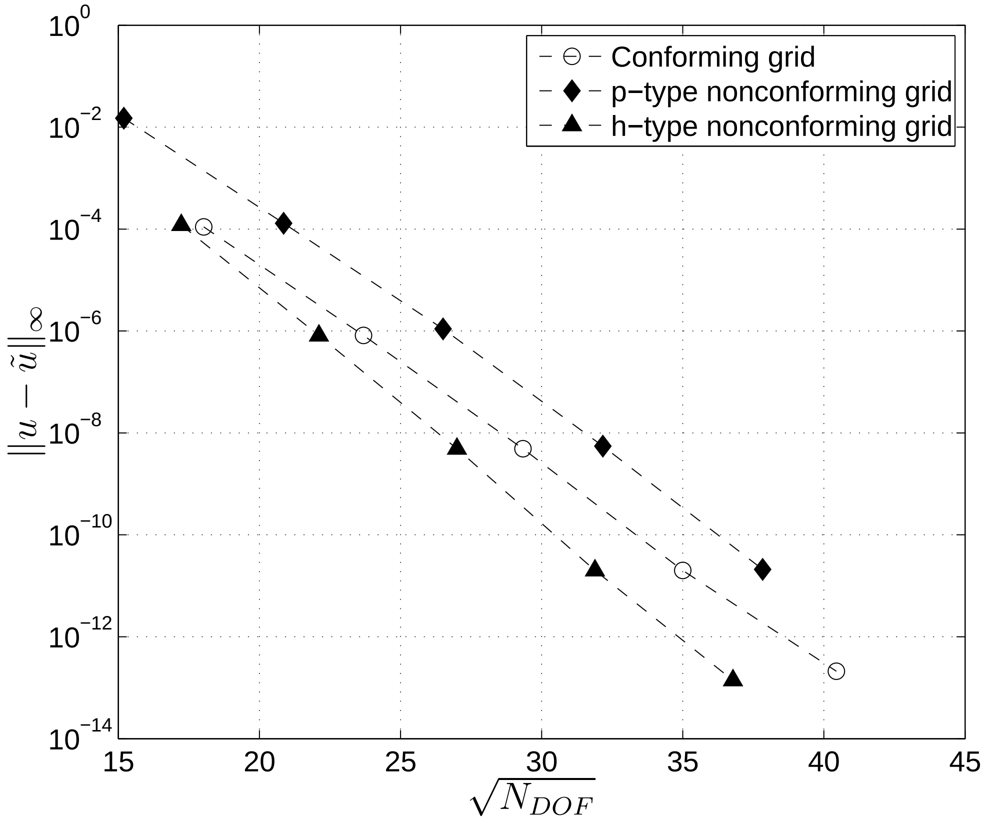


Figure 4.2: Kovasznay flow. The convergence trends for conforming, p-type non-conforming and h-type non-conforming grids for Kovasznay flow

order of the whole grid. Instead, the number of degrees of freedom used to approximate each of the dependent variables, N_{DOF} can be used as a measure of expansion order. Noting that $N_{DOF} \propto p^2$, $\sqrt{N_{DOF}}$ is used as a representative value for the expansion order.

In figure 4.2 presents a semi-log plot of maximum-norm error in x-velocity, $\|u - \tilde{u}\|_\infty$ against $\sqrt{N_{DOF}}$ is presented. Similar trends are also observed for other dependent variables but are not presented here for brevity. The results are consistent with the fact that an exponential decay is represented by a linear trend on a semi-log plot. The straight line patterns in figure 4.2 indicate that spectral convergence is achieved for all three grid configurations regardless of the presence of p- and h-type non-conformities.

The exact solution of the Kovasznay flow can also be used to investigate the accuracy of the posteriori error estimates used in LSSEM flow solver. Table 4.1 compares the local (elemental) error estimates based on the least-squares functional, $\tilde{e}_{ls}$ and spectral expansion coefficients, $\tilde{e}_{spectral}$ with the local relative true error, e_{rel} of the solution. The local relative true

Table 4.1: Kovaznay flow. Comparison of local posteriori error estimates and relative true errors. The average of the relative true errors, e_{rel}^{ave} is also provided for comparison with error estimate based on least-squares functional $\tilde{e}_{ls}$. All values are the maxima of the quantities across the domain.

			Posteriori Error Estimates			
p	N_{DOF}	$\tilde{e}_{ls}$	$\tilde{e}^{u}_{spectral}$	$\tilde{e}^{v}_{spectral}$	$\tilde{e}^{p}_{spectral}$	$\tilde{e}^{\omega}_{spectral}$
4	153	$3.0 \cdot 10^{-3}$	$2.9 \cdot 10^{-3}$	$3.2 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$
6	325	$7.6 \cdot 10^{-5}$	$2.5 \cdot 10^{-5}$	$8.4 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$	$4.6 \cdot 10^{-4}$
8	561	$9.8 \cdot 10^{-7}$	$3.1 \cdot 10^{-7}$	$1.0 \cdot 10^{-5}$	$4.3 \cdot 10^{-6}$	$6.7 \cdot 10^{-6}$
10	861	$3.8 \cdot 10^{-8}$	$2.0 \cdot 10^{-9}$	$8.0 \cdot 10^{-8}$	$2.7 \cdot 10^{-8}$	$5.8 \cdot 10^{-8}$

			Relative True Errors			
p	N_{DOF}	e^{ave}_{rel}	e^{u}_{rel}	e^{v}_{rel}	e^{p}_{rel}	e^{ω}_{rel}
4	153	$2.8 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$4.8 \cdot 10^{-2}$
6	325	$4.8 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$	$6.4 \cdot 10^{-4}$	$8.7 \cdot 10^{-4}$
8	561	$6.3 \cdot 10^{-6}$	$1.3 \cdot 10^{-6}$	$3.3 \cdot 10^{-6}$	$9.7 \cdot 10^{-6}$	$1.1 \cdot 10^{-5}$
10	861	$5.9 \cdot 10^{-8}$	$7.8 \cdot 10^{-9}$	$2.7 \cdot 10^{-8}$	$9.3 \cdot 10^{-8}$	$1.1 \cdot 10^{-7}$

error e_{rel} of each of the dependent variables is calculated by:

$$e^{u}_{rel} = \frac{||u - \tilde{u}||_{H^1,\Omega_e}}{||\tilde{u}||_{H^1,\Omega_e}} \tag{4.2}$$

where u is the dependent variable and $||\cdot||_{H^1,\Omega_e}$ is the elemental H^1 norm. $\tilde{e}_{ls}$, $\tilde{e}_{spectral}$ and e_{rel} are calculated for each of the elements and their maximum values are presented in table 4.1 for comparison. It is observed that for both kind of error estimates, the differences between the relative true errors and corresponding error estimates are within an order of magnitude. Moreover the difference decreases as the expansion order is increased. While the estimates differ from the true errors by an order of magnitude, they follow the same trend as the true errors as the expansion order is increased. Therefore, as far as this smooth problem is concerned, both $\tilde{e}_{ls}$ and $\tilde{e}_{spectral}$ can be used as qualitative error estimates for p-type adaptive refinement. Finally it is worth to mention that the distribution of the error estimates and the relative true errors across the domain are almost uniform. As a result, an p-type adaptive refinement procedure would result in uniform refinement.

4.2 2D Lid Driven Cavity Flow

A popular incompressible flow problem used to test flow solvers is the shear driven flow inside a 2-dimensional square cavity. In this study, lid driven cavity problem is solved for various Reynolds numbers to illustrate the use of h-type non-conforming grids and p-type adaptive refinement procedures with the LSSEM flow solver.

The problem setup and the main flow features are illustrated in figure 4.3. The flow is induced by an infinite lid pulled towards right at the top of the cavity. As the flow reaches steady-state, a large central vortex forms in the middle of the cavity. The stationary walls near the lower corners form local triangular wedge regions where some small corner vortices driven by the central vortex are formed. The size and the number of the corner vortices change with the Reynolds number based on the cavity dimension. In this study, three different Reynolds numbers are of interest: *Re*=100, 1000 and 5000.

It is important to observe that the boundary conditions presented in the problem setup make the x-velocity discontinuous at the top corners resulting in a singular (non-smooth) solution for the velocity field. These corner singularities destroy the favorable convergence characteristics of the high order methods and it is desirable to eliminate the corner singularities completely or confine their effects into small regions in SEM studies. It is possible to subtract the singular solutions at the corners from the overall solution and solve the remaining smooth solution numerically by spectral methods [76]. Botella and Peyret [76] follows this approach to present highly accurate results for the *Re*=1000 flow. Another approach to avoid corner singularities is to solve a regularized version of the problem where the velocity boundary condition across the top boundary is smoothed by using a steep but continuous variation of velocity near corners [77]. However the singularities can not be smoothed extensively without changing the physical nature of the problem by this way. In the current study, the effects of the singularities are restricted to small regions in the vicinity of the corners by utilizing the ability of LSSEM flow solver to work on h-type non-conforming grids.

The computational grid used in this study is an h-type nonconforming Cartesian grid composed of $NE = 334$ square elements. The computational grid is illustrated in figure 4.4. The grid resolution is increased near the boundaries to resolve the boundary layers. The lower corners are further refined to capture corner vortices properly. The upper corners are refined even further to confine the effects of the singularities to few small elements in the vicinity of corners.

The steady-state solution is obtained by time marching. The solution is declared to have reached steady-state when:

$$\frac{||\mathbf{V}^t - \mathbf{V}^{t-5}||_{L^2}}{||\mathbf{V}^t||_{L^2}} < 10^{-4} \tag{4.3}$$

where $\mathbf{V}^t$ is the velocity magnitude vector at the current time and $\mathbf{V}^{t-5}$ is the velocity magnitude vector 5 seconds before. With this approach, steady state solutions for *Re*=100 and 1000 are obtained from an initially stagnant flow field before $t = 25$ and $t = 100$ seconds respectively. Since the final steady state solution is not affected from the size of the time steps, large time steps such as $\Delta t = 1.0$ and 0.5 seconds are used respectively. Similarly steady state solution for *Re*=5000 is obtained by starting from the steady state *Re*=1000 solution, using $\Delta t = 0.5$ second and integrating over a time interval of $t = 125$ seconds.

While the focus of this study is on the more challenging *Re*=5000 problem, the solutions for *Re*=100 and 1000 are also presented for completeness. The *Re*=100 and 1000 problems are solved using elements with expansion orders $p = 4$ in both space dimensions. For visual validation, the streamlines of the resulting numerical solutions are presented in figure 4.5. At $Re = 100$, the central vortex is offset from the center of the cavity towards the downstream and the lid. There is a single small corner vortex at each lower corner, the one near the right corner being greater in size. As *Re* is increased to 1000, the central vortex moves closer to the center of the cavity and the corner vortices become greater in size. Two additional corner vortices in the vicinity of the lower corners form though these vortices can be observed only in high resolution benchmark solutions such as the work of Erturk et al. [3].

For quantitative validation of the solutions, it is found necessary to compare the results of the present study with the benchmark results available in the literature. The previous works chosen for comparison are as follows: Ghia et al. [2] performed multi-grid finite difference solutions on a 256×256 grid for a range of Reynolds numbers between 100 and 10000. Later Botella and Peyret [76] obtained highly-accurate *Re*=100 and 1000 solution using Chebyshev pseudospectral method upon singularity subtraction. More recently, Erturk et al. [3] presented finite difference solutions for Reynolds numbers between *Re*=1000 and 21000 using stream function-vorticity formulation on very fine grids (401×401 to 601×601). Figures 4.6a and 4.6b present the x-velocity and y-velocity profiles taken from the vertical and horizontal cen-

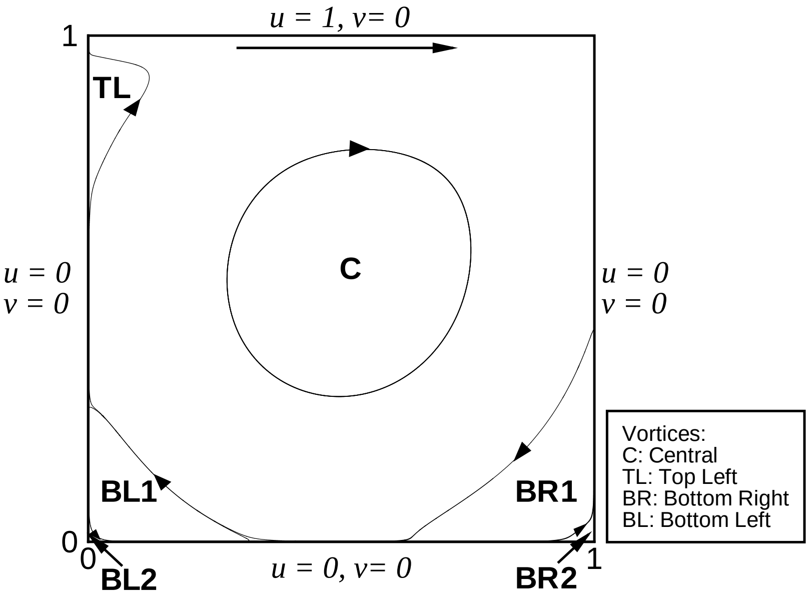


Figure 4.3: Lid driven cavity flow. Problem setup.

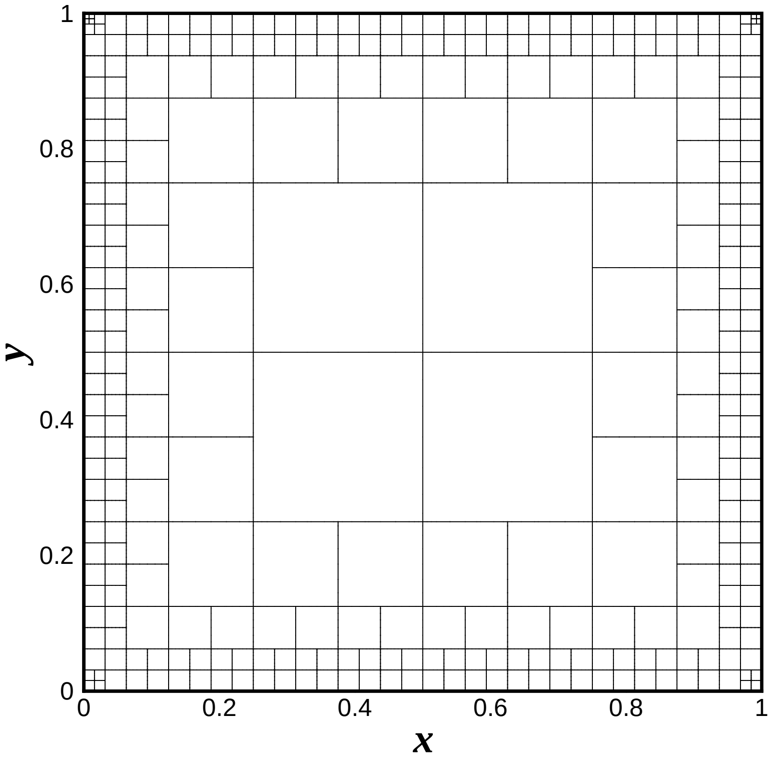
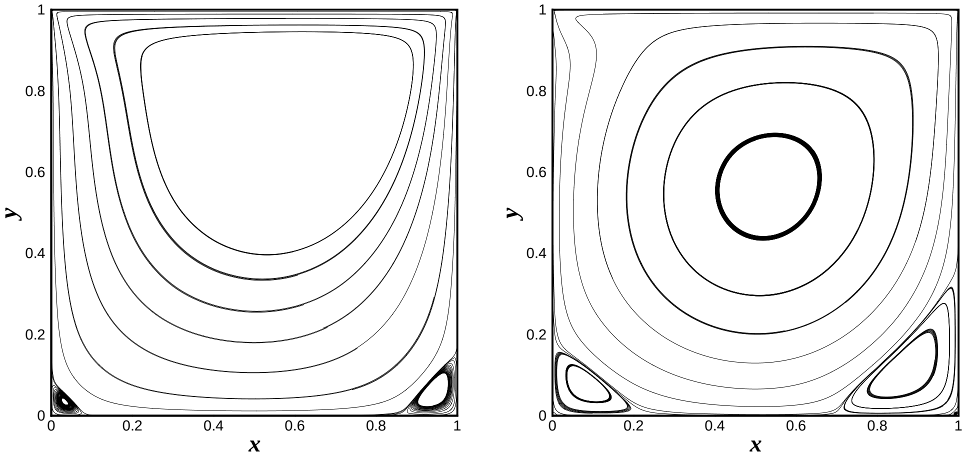


Figure 4.4: Lid driven cavity problem. Computational grid, $NE = 334$.



(a) *Re*=100

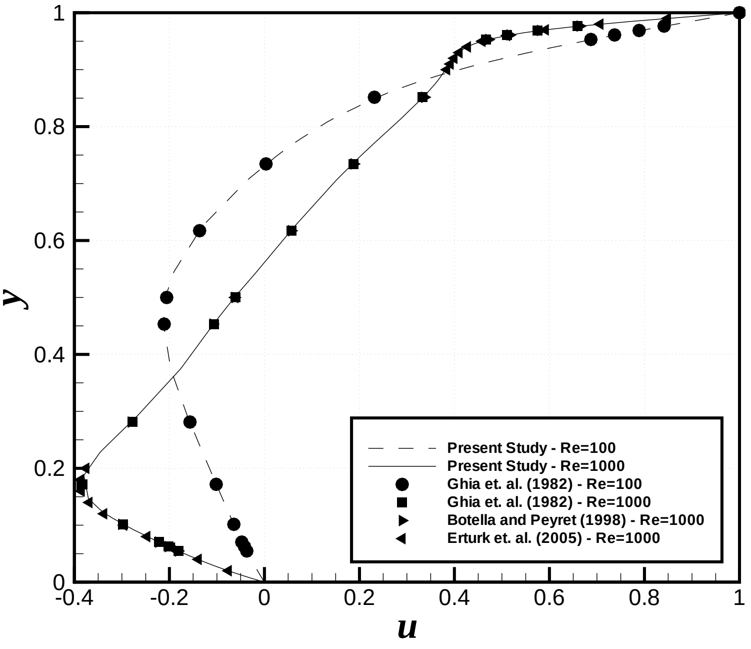
(b) *Re*=1000

Figure 4.5: Lid driven cavity problem. Streamlines of Re=100 and 1000 solutions

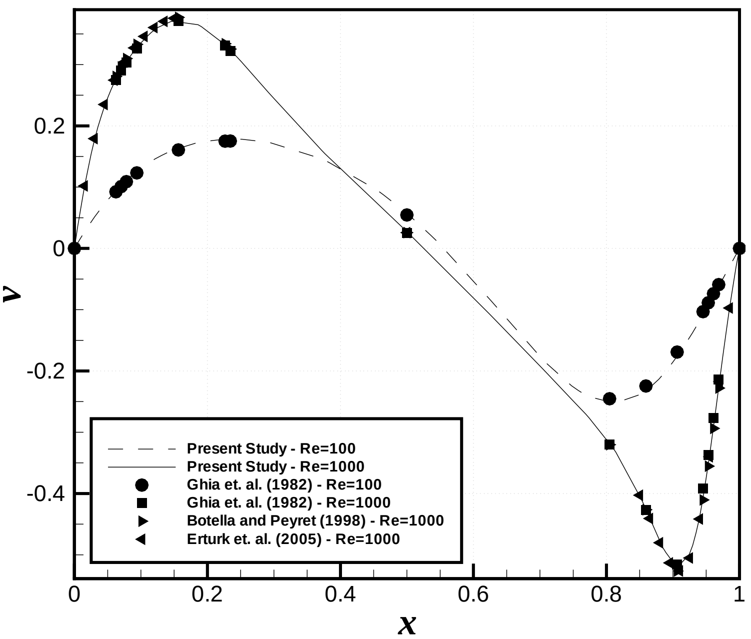
terline of the domain respectively together with the data available in [2], [76] and [3]. It is observed that for both *Re*=100 and 1000, x- and y-velocity profiles are in very good agreement with the literature. The h-type nonconforming Cartesian grid with p=4th order elements results in satisfactory results for this range of Reynolds number.

As *Re* is increased further it becomes challenging to resolve the flow features accurately. When the problem is solved for *Re*=5000 with the aforementioned grid composed of 4th order elements, discrepancy with respect to literature is observed in some regions of velocity profiles. Therefore a grid study is conducted to improve the results. The problem is solved on four grids: First grid is the previously used grid that has NE=334 elements with expansion orders p=4. Another grid is obtained by uniform p-type refinement of first grid. This grid has NE=334 elements with expansion orders p=8. Two additional grids are obtained by starting the solution with the first grid and solving the problem with p-type adaptive refinement procedures. One adaptive refinement procedure uses the error estimate based on least-squares functional while the other uses the error estimate based on spectral coefficients for x-velocity. Table 4.2 summarizes the specifications of the computational grids.

Figure 4.7 illustrates the streamlines of the solution obtained by the finest grid (Grid (4)). It is observed that as Reynolds number is increased from 1000 to 5000 the central vortex moves even closer to the center of the cavity. The second set of corner vortices becomes more pronounced and a new vortex near the upstream of the lid forms. The contour plots of x-velocity, y-velocity, pressure, and vorticity are also presented in figures 4.8, 4.9, 4.10, and



(a) x-velocity profile across the vertical centerline



(b) y-velocity profile across the horizontal centerline

Figure 4.6: Lid driven cavity problem. Velocity profiles across centerlines for Re=100 and 1000 solutions

Table 4.2: Lid driven cavity flow. Specifications of the grids used in solution of *Re*=5000 problem. For adaptive solutions the given expansion orders *p* refer to the initial grid. The error lower bounds are kept small such that there is not any coarsening in the adaptive grids.

Grid no	NE	p	Error Estimate	Error Upper Bound (ε_{max})	(p_{min}, p_{max})
1	334	4	Nonadaptive	-	-
2	334	4	$\tilde{e}_{ls}$	10^{-4}	$(4, 10)$
3	334	4	$\tilde{e}^{u}_{spectral}$	10^{-3}	$(4, 10)$
4	334	8	Nonadaptive	-	-

4.11 respectively.

x- and y-velocity profiles are presented together with the results of Ghia et al. [2] and Erturk et al. [3] in figures 4.12 and 4.13 respectively. When general trends are considered, all of the solutions agree well with the literature. However the solution of grid (1), that is the non-adaptive grid with *p*=4, is associated with significantly lower x-velocity magnitudes at the cusp region near the upper boundary layer when compared to the other solutions and literature. This solution also estimates the minimum x-velocity magnitude and location different than the other solutions. When y-velocity profiles are examined it is again observed that the solution with grid (1) deviates from the other solutions and benchmark data near maximum and minimum y-velocity regions. On the other hand, x- and y-velocity profiles obtained by grids (2), (3) and (4) agree well with the literature, especially with the benchmark solution of Erturk et al. [3]. It is worth to mention that velocity profile data for *Re*=5000 is scarce in the literature and the recent results of Erturk et al. [3] can be considered as the most accurate due to the very fine grid (601×601) utilized.

The vortex center coordinates and the vorticity values at vortex centers are also compared with the literature in table 4.3. It is seen that the results of the past studies differ from each other. Therefore, there is not any precise values for vortex center coordinates and vorticity at vortex centers. All four grids used in the present study capture the vortices properly and present values for vortex centers that are close to or within the range of results of past studies. Vorticity values at vortex centers get closer to the range of results of past studies with p-type adaptive and uniform refinement. All together, the solutions with the grids (2), (3) and (4) can be considered as accurate solutions of lid driven cavity flow at *Re*=5000.

Finally, the distribution of the error estimate values in the steady-state solution obtained by grid (1) and the adaptive grids, grid (2) and grid (3), are illustrated in figures 4.14 and 4.15 respectively. It is observed in figure 4.14a that the regions where $\tilde{e}_{ls}$ is greater than ε_{max} =

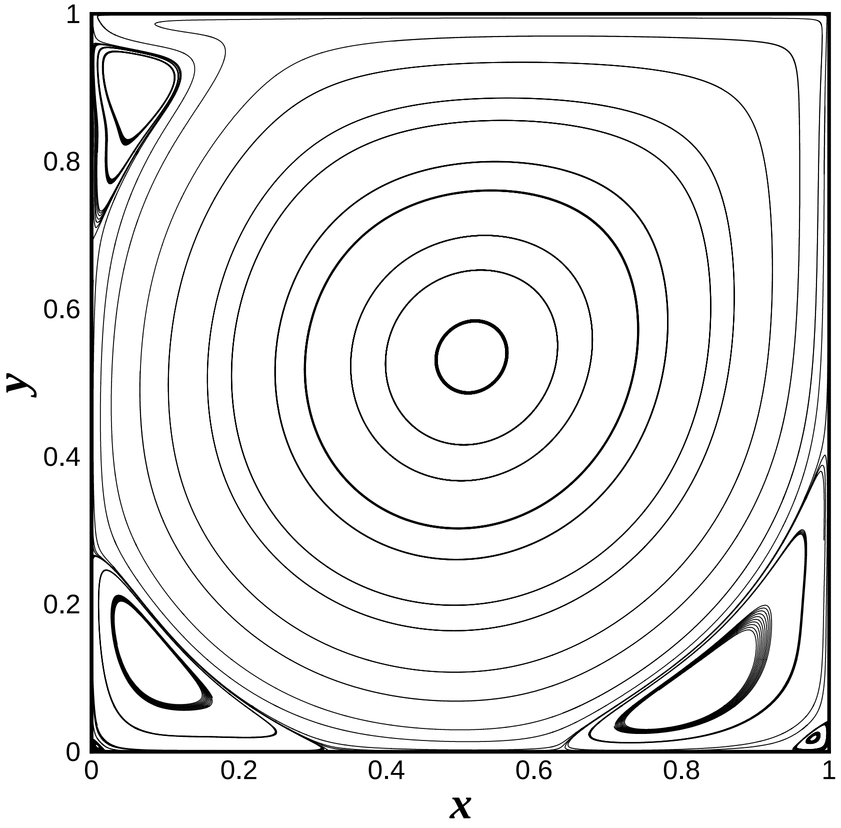


Figure 4.7: Lid driven cavity problem. Streamlines of *Re*=5000 solution.

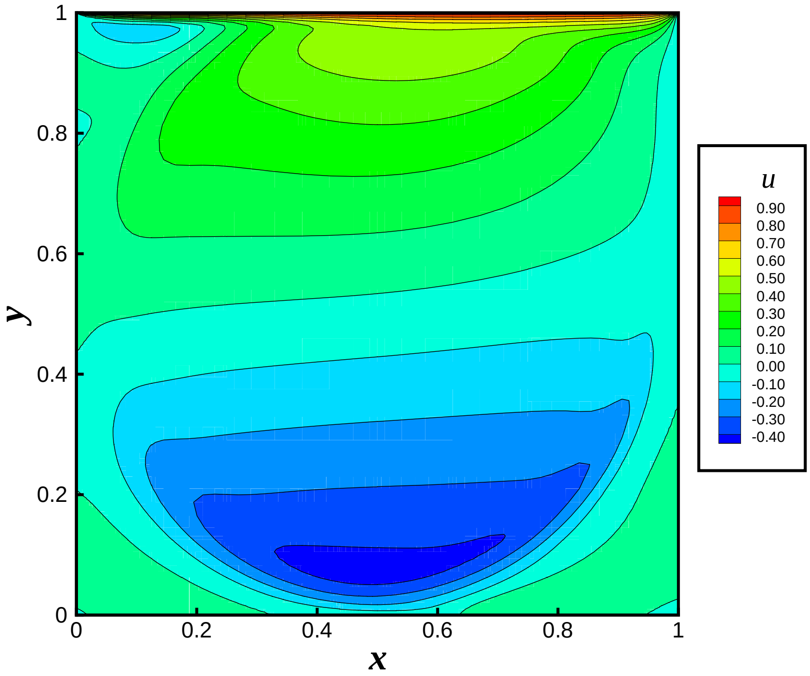


Figure 4.8: Lid driven cavity problem. x-velocity contours of *Re*=5000 solution.

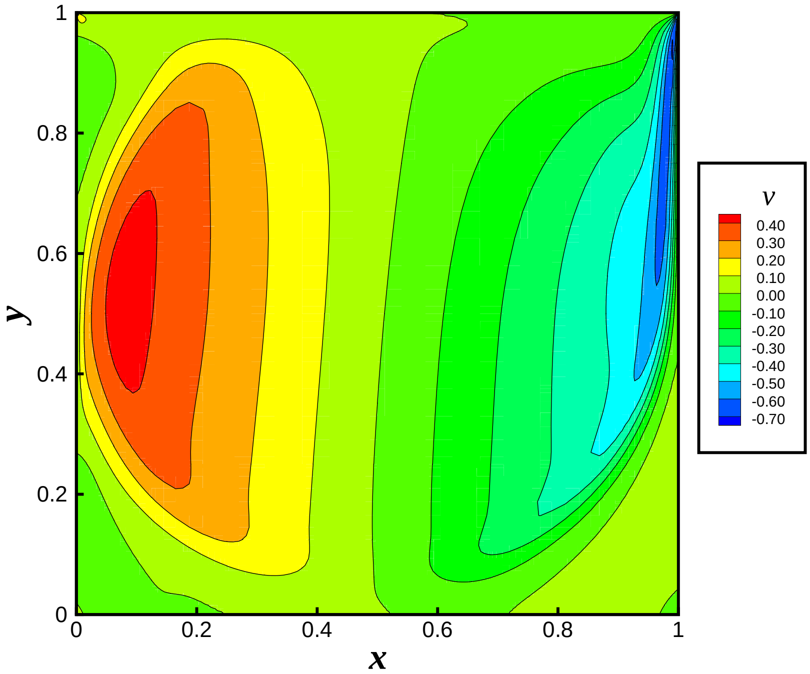


Figure 4.9: Lid driven cavity problem. y-velocity contours of *Re*=5000 solution.

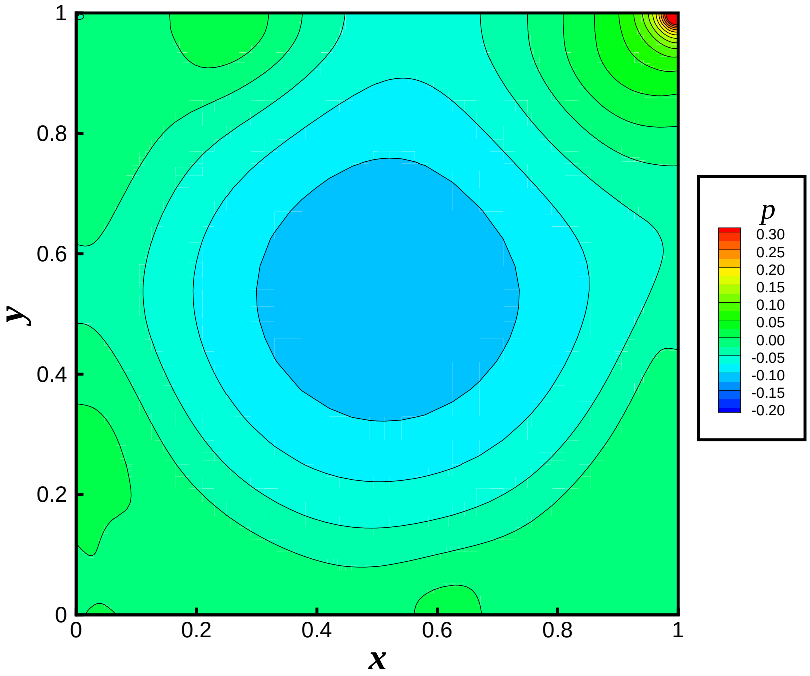


Figure 4.10: Lid driven cavity problem. Pressure contours of *Re*=5000 solution.

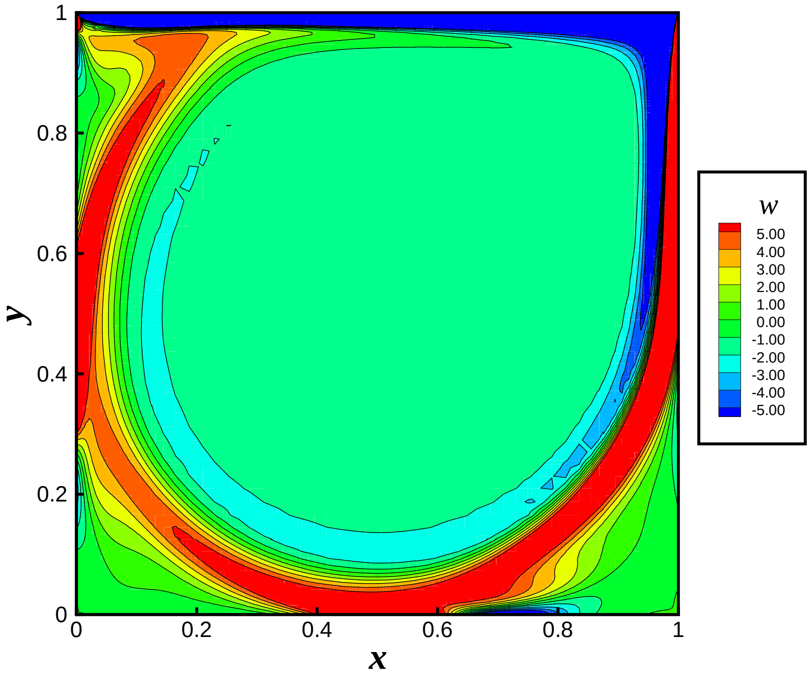
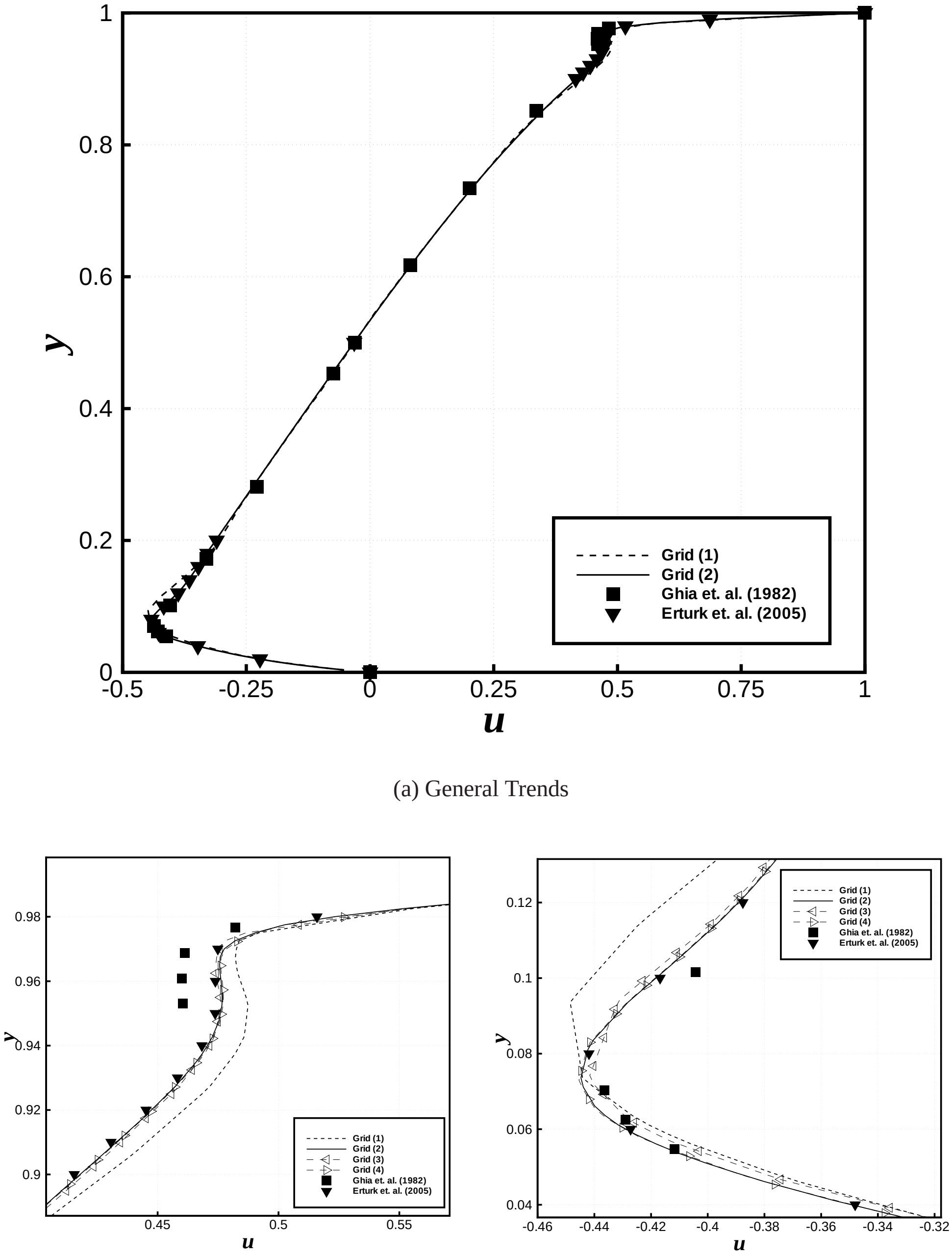


Figure 4.11: Lid driven cavity problem. Vorticity contours of *Re*=5000 solution.

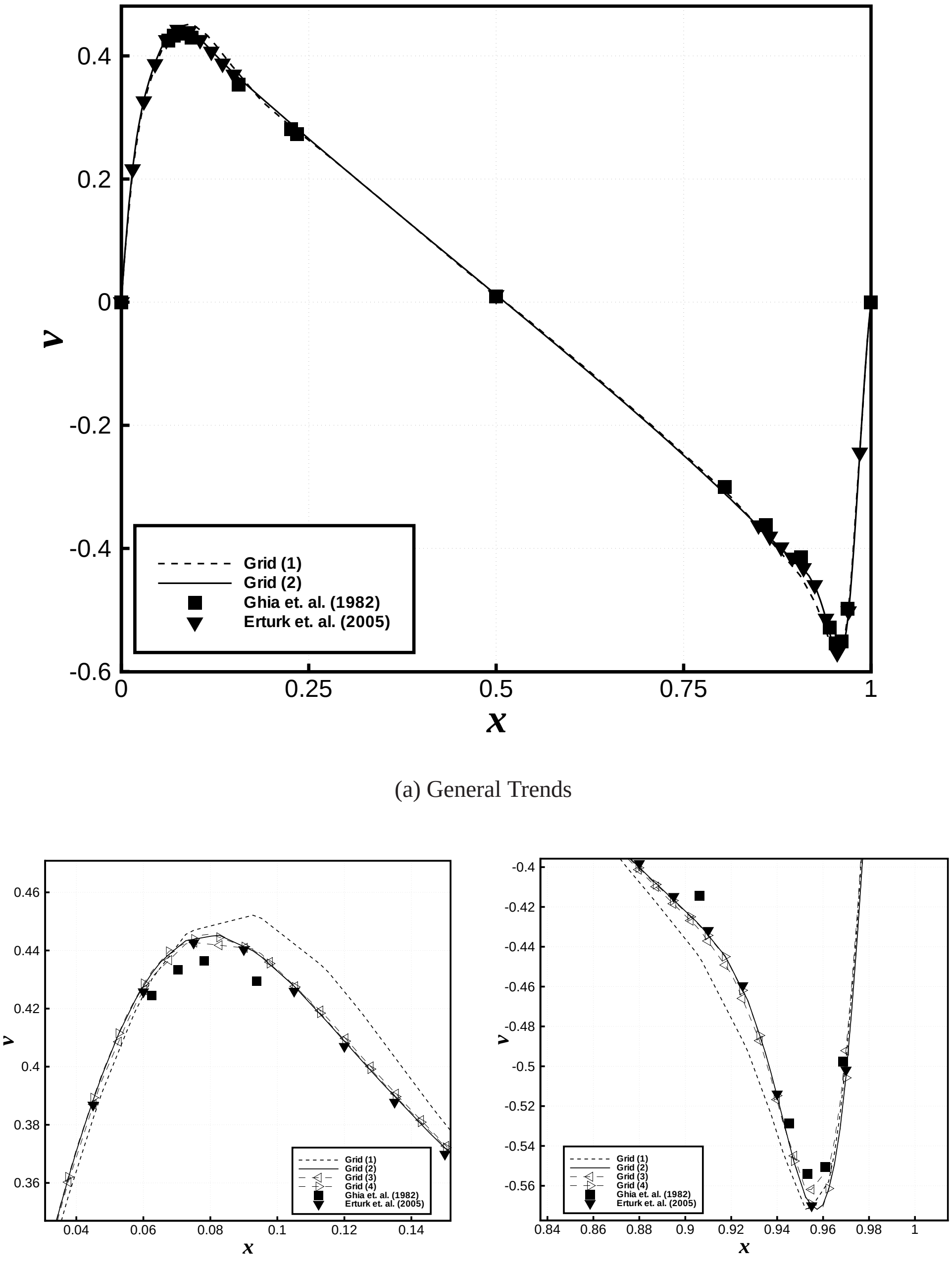


(a) General Trends

(b) Close up: Cusp Region

(c) Close up: Minimum velocity region

Figure 4.12: Lid driven cavity problem. Re=5000 solution. x-velocity profile across vertical centerline



(a) General Trends

(b) Close up: Cusp Region

(c) Close up: Minimum velocity region

Figure 4.13: Lid driven cavity problem. Re=5000 solution. y-velocity profile across horizontal centerline

Table 4.3: Lid driven cavity flow: Re 5000 Solution. Vortex center coordinates and vorticity values at vortex centers

	Top Left		Central	
	$(x,y)_c$	ω_c	$(x,y)_c$	ω_c
Present Study, Grid 1	(0.0755, 0.9064)	1.6256	(0.5161, 0.5366)	-1.8667
Present Study, Grid 2	(0.0629, 0.9118)	2.1876	(0.5147, 0.5357)	-1.9815
Present Study, Grid 3	(0.0636, 0.9087)	2.0480	(0.5151, 0.5352)	-1.9626
Present Study, Grid 4	(0.0633, 0.9092)	2.0570	(0.5151, 0.5352)	-1.9640
Ghia et al. [2]	(0.0625, 0.9102)	-2.0884	(0.5117, 0.5352)	1.8602
Erturk et al. [3]	(0.0633, 0.9100)	2.0628	(0.5150, 0.5350)	-1.9266
Barragy and Carey [78]	(0.0635, 0.9092)	-	(0.5151, 0.5359)	-
Sahin and Owens [79]	(0.0621, 0.9108)	-	(0.5134, 0.5376)	-1.9392
Bruneau and Saad [77]	-	-	(0.5147, 0.5352)	1.9322

(a) Top left and central vortices

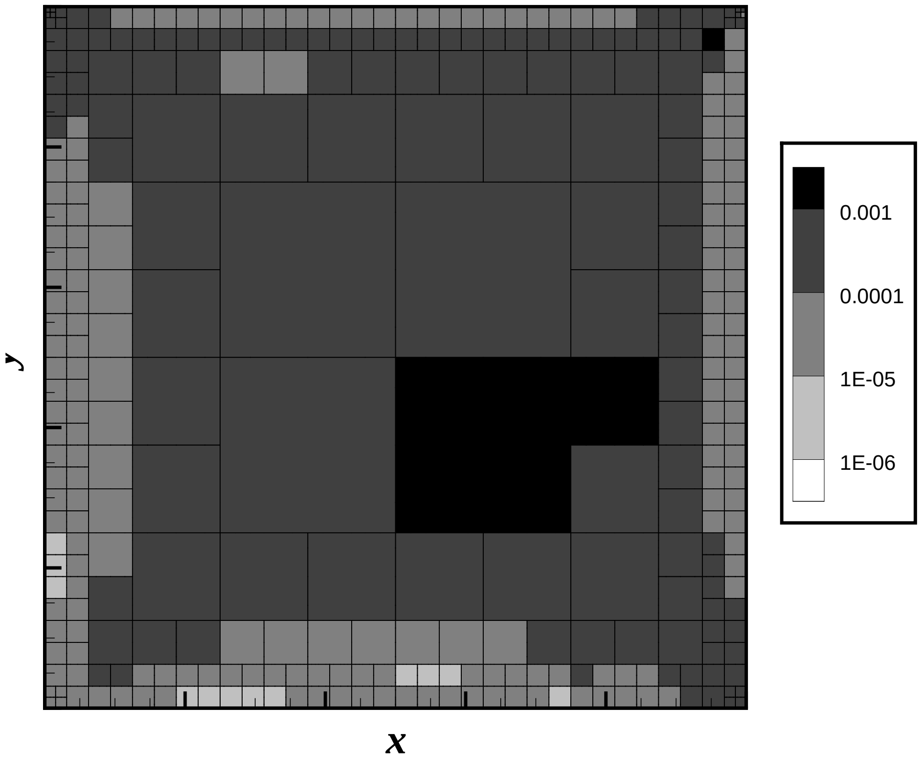
	Bottom Left-1		Bottom Left-2	
	$(x,y)_c$	ω_c	$(x,y)_c$	ω_c
Present Study, Grid 1	(0.0701, 0.1433)	1.5191	(0.0064,0.0084)	-0.0070
Present Study, Grid 2	(0.0724, 0.1368)	1.4255	(0.0066, 0.0085)	-0.0088
Present Study, Grid 3	(0.0734, 0.1358)	1.4840	(0.0079, 0.0079)	-0.0102
Present Study, Grid 4	(0.0730, 0.1367)	1.5008	(0.0078, 0.0080)	-0.0102
Ghia et al. [2]	(0.0703, 0.1367)	-1.5306	(0.0117, 0.0078)	0.0188
Erturk et al. [3]	(0.0733, 0.1367)	1.5026	(0.0083, 0.0083)	-0.0123
Barragy and Carey [78]	(0.0725, 0.1370)	-	(0.0079, 0.0079)	-
Sahin and Owens [79]	(0.0720, 0.1382)	-		-
Bruneau and Saad [77]	-	-	-	-

(b) Bottom Left Vortices

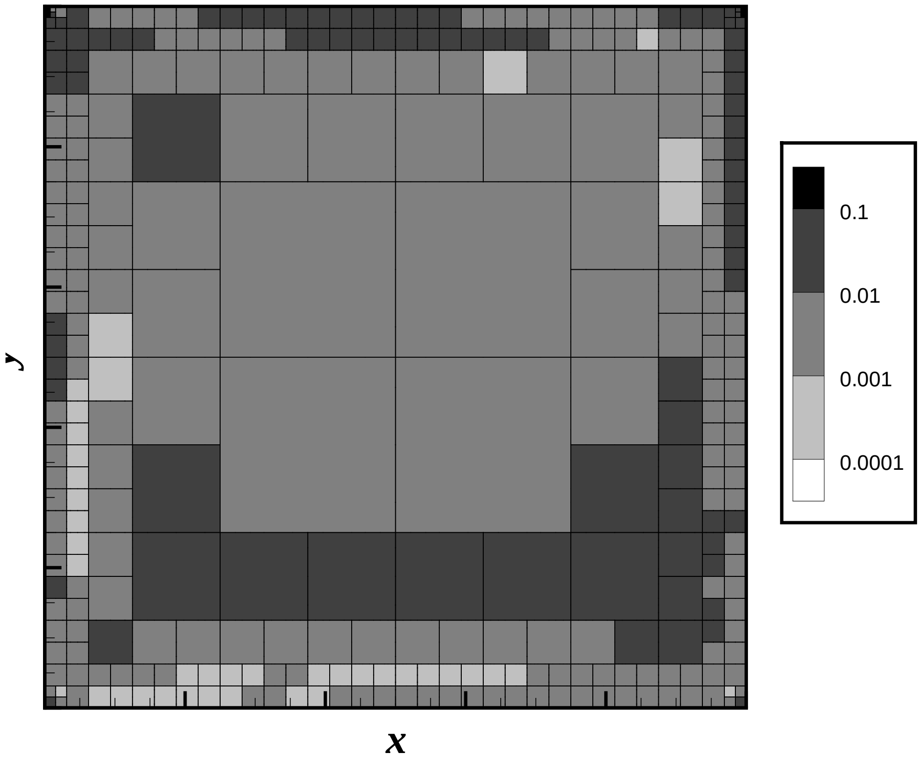
	Bottom Right-1		Bottom Right-2	
	$(x,y)_c$	ω_c	$(x,y)_c$	ω_c
Present Study, Grid 1	(0.8030, 0.0717)	2.9788	(0.9752, 0.0146)	-0.0216
Present Study, Grid 2	(0.8019, 0.0725)	2.7645	(0.9788, 0.0178)	-0.0299
Present Study, Grid 3	(0.8057, 0.0734)	2.7156	(0.9787, 0.0185)	-0.0338
Present Study, Grid 4	(0.8045, 0.0728)	2.7359	(0.9788, 0.0186)	-0.0336
Ghia et al. [2]	(0.8086, 0.0742)	-2.6635	(0.9805, 0.0195)	0.0319
Erturk et al. [3]	(0.8050, 0.0733)	2.7245	(0.9783, 0.0183)	-0.0341
Barragy and Carey [78]	(0.8041, 0.0725)	-	(0.9786, 0.0188)	-
Sahin and Owens [79]	(0.8081, 0.0741)	-		-
Bruneau and Saad [77]	(0.8057, 0.0732)	2.7245	-	-

(c) Bottom Right Vortices

10^{-4} are concentrated near the center and the corners except the lower left corner. On the contrary, the regions immediately near the walls and the lid are associated with $\tilde{e}_{ls}$ that are less than 10^{-4}. Therefore the regions that are expected to be refined in the solution with grid (2) are the center and the corners. This observation is verified in the refinement patterns presented in figure 4.15a. When figure 4.14b is examined it is seen that $\tilde{e}^{u}_{spectral}$ is greater than $\varepsilon_{max} = 10^{-3}$ almost everywhere in the domain. Therefore it may be thought that the grid should be refined uniformly by the adaptive refinement procedure based on $\tilde{e}^{u}_{spectral}$. However, since multiple refinement levels are possible, the actual level of refinement may vary from one region to another. Figure 4.15b shows that different regions are indeed refined by different levels. In particular, the regions near the wall boundary at the top right corner and the region between the center and the bottom right corner have the highest expansion orders i.e. $p = 10$. In contrast to grid refinement based on $\tilde{e}_{ls}$, the central region is refined by only one or two levels. The boundary layer regions near the walls that are not refined in grid (2) are also refined. Since both grids (2) and (3) are associated with very good agreement with the literature, declaring one's superiority over the other is not possible. It can be noted that the error estimates result in comparable results while refining different regions. Therefore, the it can be suggested that the adaptive refinement procedure can be improved by employing both error estimates so that only the intersection of the regions refined by the individual estimates is refined.

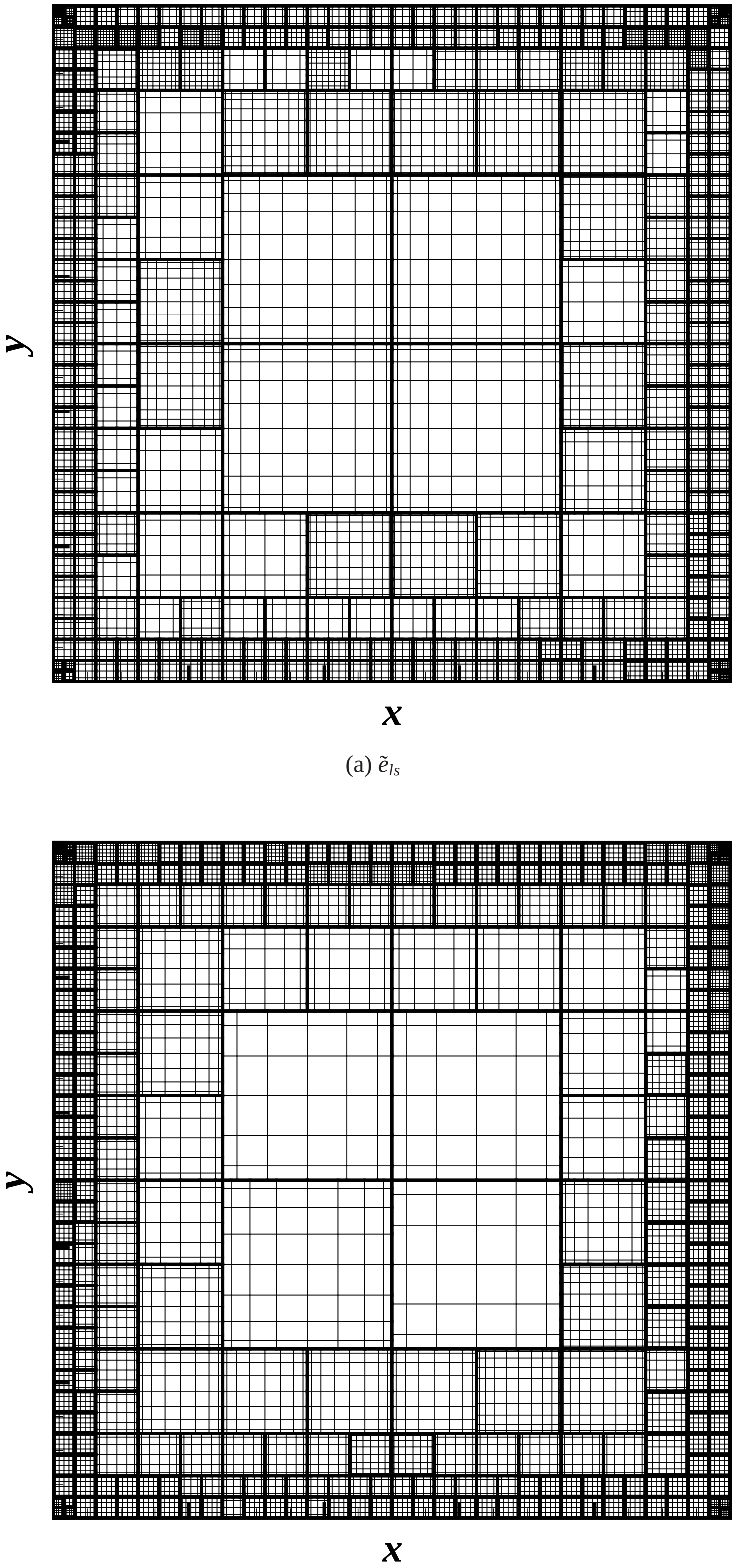


(a) $\tilde{e}_{ls}$



(b) $\tilde{e}^{u}_{spectral}$

Figure 4.14: Lid driven cavity flow. Distribution of error estimates.



(a) $\tilde{e}_{ls}$

(b) $\tilde{e}^{u}_{spectral}$

Figure 4.15: Lid driven cavity flow. Adaptive grid refinements.

4.3 Flow Past a Cylinder in a Channel

In this section the problem of interest is the laminar flow in a channel where flow is partially blocked by a large circular cylindrical obstacle. This problem and its variants are encountered frequently in the mass conservation studies of least squares formulation [47, 30, 39, 38, 49, 42, 46]. In this study, the effectiveness of p-type adaptive refinement procedures on the mass conservation of LSSEM are investigated.

It is known that LSFEM and LSSEM are associated with poor mass conservation especially in the case of problems with inflow and outflow [47]. Several remedies proposed to improve the mass conservation properties of least-squares methods were reviewed in Chapter 1. Those remedies generally involve the modification of the original formulation and they are associated with some drawbacks such as the loss of favorable properties of the coefficient matrix i.e. symmetry and the positive definiteness [47], increased condition number leading to poor performance with iterative solvers [30], introduction of additional adjustable parameters [39, 42], or additional equations and complicated boundary conditions [49] in the formulation. Recently it has been shown that good mass conservation can be achieved by the original least-squares formulation using few and large elements with very high expansion orders ($p \propto 20$) [46]. Implementing such an approach on a conforming grid results in excessive and unnecessary refinement. Therefore it is computationally impractical. in this study, it is shown that the same accuracy in mass conservation can be achieved with relative ease using LSSEM flow solver's p-type adaptive refinement capabilities.

The problem setup is illustrated in figure 4.16 is solved on a rectangular channel section: $[-11, 14] \times [-1, 1]$ ($L = 25,\ H = 2$) where a cylinder with diameter $D = 1$ is positioned at the origin. No slip boundary conditions are applied at the upper and lower channel walls and the surface of the cylinder. A uniform inflow $u = 1, v = 0$ is prescribed at $x = -11$ and a fully developed outflow boundary condition where tangential velocity v and pressure p are set to zero is enforced at the downstream boundary. The viscosity is assigned such that the resulting Reynolds number based on cylinder diameter is 40 for which a steady, laminar solution exists.

The problem is solved on a geometrically conforming grid composed of $NE = 32$ elements as illustrated in figure 4.17. The solution is started from an stagnant flow field. The steady

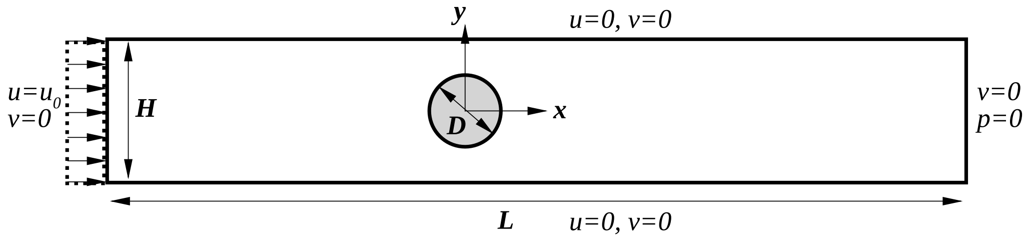


Figure 4.16: Flow past a cylinder in a channel: Problem setup

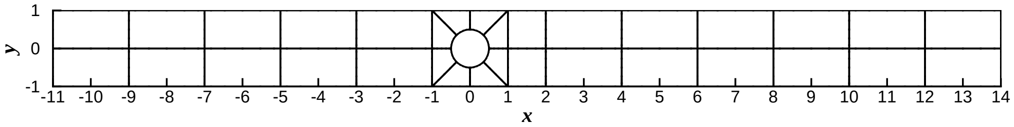


Figure 4.17: Flow past a cylinder in a channel: Computational grid. $NE = 32$

state solution is obtained by marching in time through a time interval of 40 seconds using a time step of $\Delta t = 1.0$ seconds. Such a time interval is found sufficient to have the steadiness criterion given by eqn.(4.3) satisfied. The elements near the cylinder have curved edges to represent the cylinder geometry accurately. The cylinder geometry is provided as line-segmented data from which the coordinates of the nodal points on the related edges are interpolated using an 8th degree Lagrange interpolation. Coordinates of the rest of the nodes and the Jacobians of the elements with curved edges are calculated using Gordon-Hall trans-finite interpolation [70] as presented in Chapter 3.

A total of 10 solutions are obtained by employing p-type adaptive refinement procedure with different error estimates and refinement criteria. Each solution is associated with a grid whose specifications are presented in table 4.4. The grids (1) and (2) are obtained by employing the expansion orders $p = 6$ and $p = 18$ respectively in all elements and all space dimensions. No adaptive refinement is applied for those cases. The grids referred by label “a” are the results of adaptive refinement procedures using the error estimate based on the least-squares functional. Grids (a1) through (a4) are obtained by imposing different error upper bounds on this error estimate. Similarly the grids referred by label “b” are due to use of error estimate based on the spectral coefficients for x-velocity. Again, 4 grids, (b1) through (b4), are obtained by imposing different error upper bounds on this error estimate. It is important to note that the adaptive grids described here are the final grids observed at the time of steady-state solution.

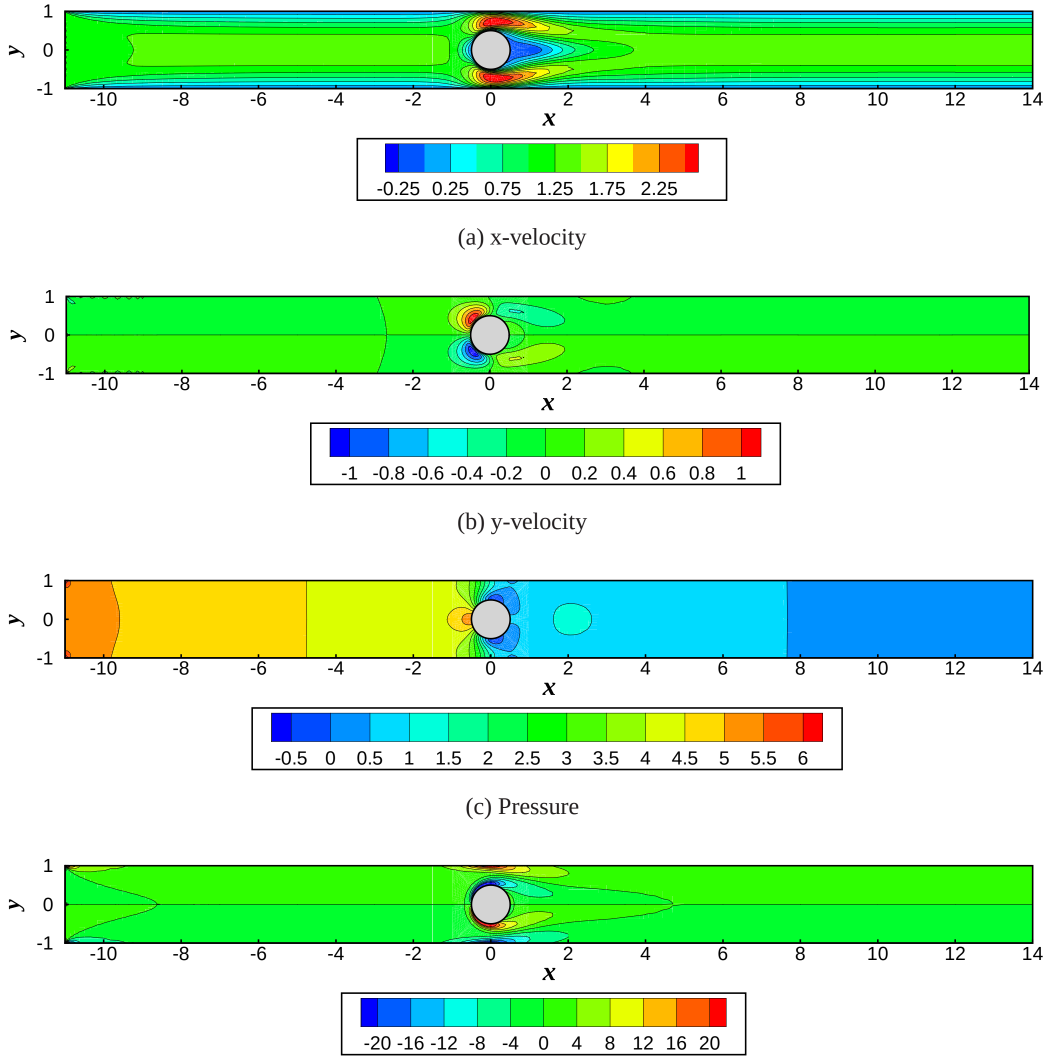
Table 4.4: Flow past a cylinder in a channel: Specifications of the grids used in the p-type adaptive refinement study

Grid no	NE	p	Error Estimate	Error Upper Bound (ε_{max})	(p_{min}, p_{max})
1	32	6	Nonadaptive	-	-
2	32	18	Nonadaptive	-	-
$a1$	32	6	$\tilde{e}_{ls}$	10^{-2}	$(6, 18)$
$a2$	32	6	$\tilde{e}_{ls}$	10^{-3}	$(6, 18)$
$a3$	32	6	$\tilde{e}_{ls}$	10^{-4}	$(6, 18)$
$a4$	32	6	$\tilde{e}_{ls}$	10^{-5}	$(6, 18)$
$b1$	32	6	$\tilde{e}^{u}_{spectral}$	10^{-2}	$(6, 18)$
$b2$	32	6	$\tilde{e}^{u}_{spectral}$	10^{-3}	$(6, 18)$
$b3$	32	6	$\tilde{e}^{u}_{spectral}$	10^{-4}	$(6, 18)$
$b4$	32	6	$\tilde{e}^{u}_{spectral}$	10^{-5}	$(6, 18)$

All adaptive refinement procedures start from the initial grid, grid (1). Grids evolve to their final states by refinements made once in every 2 time step.

Contour plots of x-velocity, y-velocity, pressure and vorticity are presented in figure 4.18. The main flow features are the developing boundary layers in the upstream of the cylinder and the maximum velocity region at $x = 0$ where high velocity and vorticity gradients are observed. The effect of mass conservation on the aspects of the recirculation zone behind the cylinder is of particular interest. Figure 4.19 illustrates the streamlines near the cylinder for the solutions obtained with the non-adaptive grid, grid (1), and one of the adaptive grids, grid a4, for which very good mass conservation is achieved. The length of the vortex at the cylinder wake should extend beyond $x = 1$ for a mass conserving solution [45]. It is observed that this length is underestimated by the non-adaptive grid solution considerably. On the other hand, the recirculation zone expands well beyond $x = 1$ up to $x = 1.35$ for the mass conserving grid (4a) solution. It is seen that mass conservation has significant effects on the flow features of this problem. Furthermore, the non-adaptive grid shows poor mass conservation performance.

The effects of p-type regular and adaptive grid refinement on mass conservation can be observed clearly in table 4.5 where the mass flow rates at different vertical sections across the channel are reported. First of all, as seen before, the coarse non-adaptive grid, grid (1) results in the poor mass conservation in the upstream since almost half of the mass has been lost as the flow reaches the cylinder. On the other hand, the fine non-adaptive grid, grid (2) is associated with very good mass conservation. This result is in accordance with the results of Kattelans



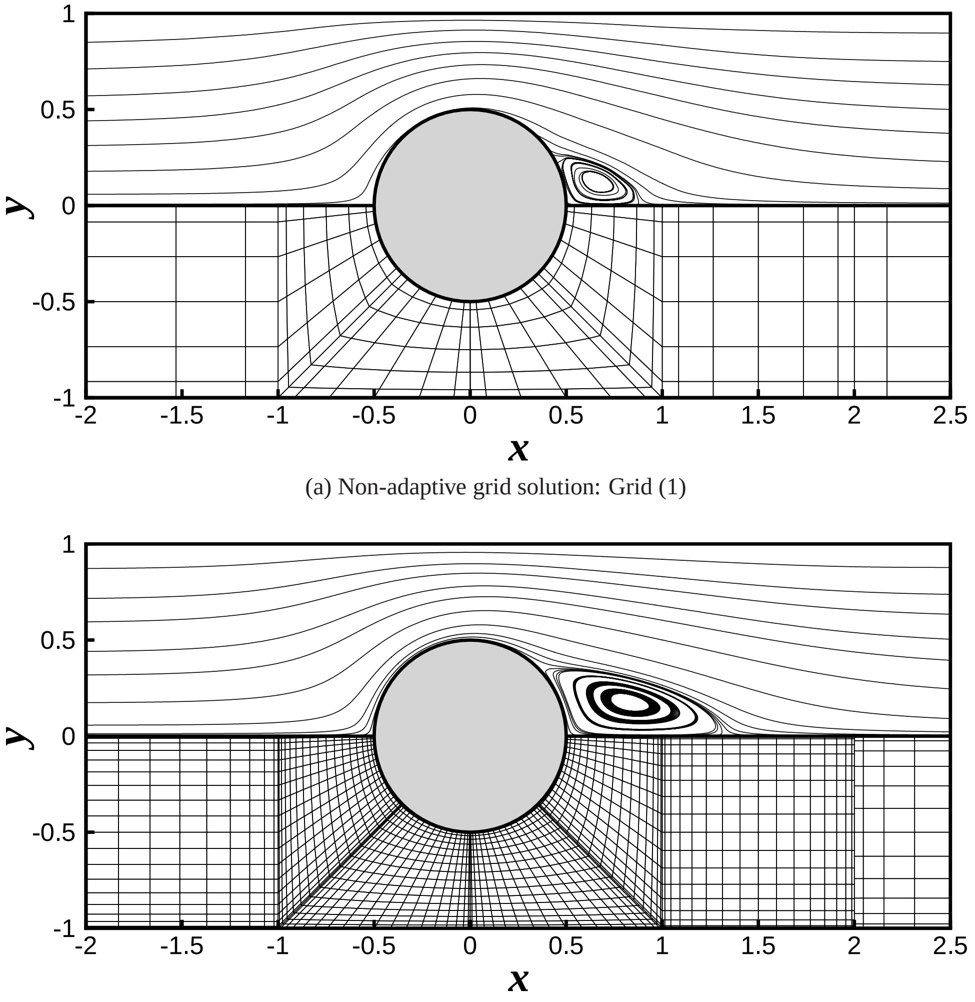
(a) x-velocity

(b) y-velocity

(c) Pressure

(d) Vorticity

Figure 4.18: Flow Past a Cylinder In a Channel: Contour plots of dependent variables. The contours are based on the solution with grid (a4) for which very good mass conservation is achieved.



(a) Non-adaptive grid solution: Grid (1)

(b) An adaptive grid solution: Grid (a4)

Figure 4.19: Flow past a cylinder in a channel: Comparison of the streamlines near the cylinder for the non-adaptive and adaptive solutions

Table 4.5: Flow past a cylinder in a channel: Mass flow rates ($\dot{m}$) across several cross sections

	$\dot{m}$				
Grid	$x = -11$	$x = -5$	$x = 0$	$x = 7$	$x = 14$
1	2.000	1.549	1.203	1.238	1.238
a1	2.000	1.600	1.208	1.243	1.243
a2	2.000	1.917	1.832	1.841	1.843
a3	2.000	1.990	1.976	1.983	1.984
a4	2.000	1.993	1.986	1.987	1.993
b1	2.000	1.619	1.307	1.345	1.346
b2	2.000	1.968	1.954	1.961	1.963
b3	2.000	1.983	1.982	1.982	1.974
b4	2.000	1.988	1.986	1.983	1.984
2	2.000	1.994	1.987	1.994	1.994

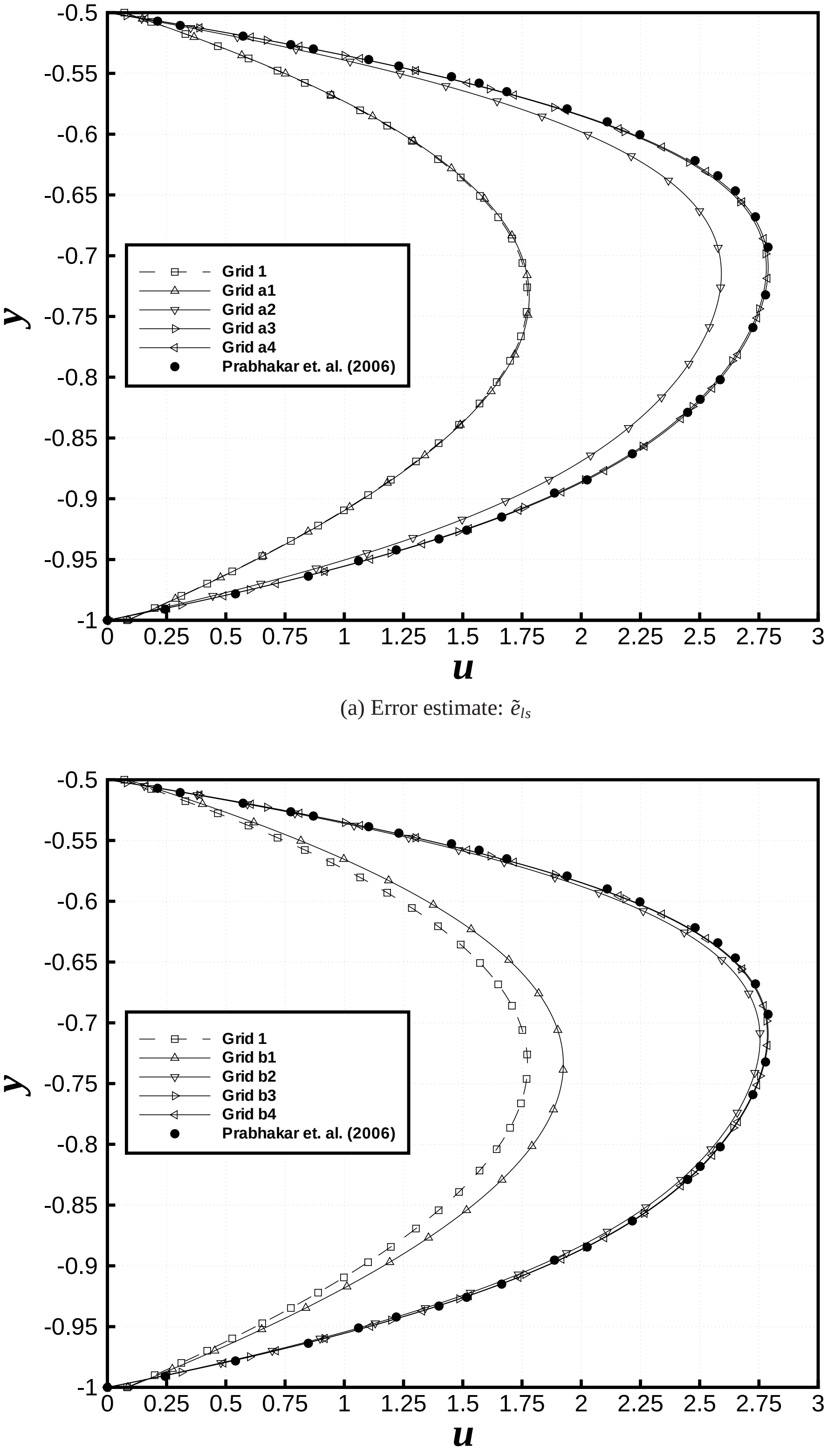
and Heinrichs [46] who reports that use of few, large, high order elements result improves the mass conservation properties of the least-squares methods considerably. However, it can be argued that some regions of the grid (2) are over-resolved as far as mass conservation is concerned. This idea follows form the observation that, the violation of mass conservation at the downstream of the cylinder is negligible when compared to the violation in the upstream even for the coarse non-adaptive grid. So the regions at the downstream may not require refinement as much as the regions at the upstream to obtain good overall mass conservation. In other words, an adaptive refinement procedure aimed to improve mass conservation should be able to distinguish between the upstream and downstream regions. This is, in fact, the case for both sets of adaptive grid solutions introduced in this study. The refinement patterns obtained by the adaptive refinement procedures are discussed later in this section.

Another important observation from table 4.5 is that adaptive grid refinement definitely improves the mass conservation for both error estimates and for all levels of error upper bounds. The extend of improvement increases as the error upper bound is made smaller. In particular, the mass conservation accuracy of adaptive grid solutions come very close (up to 2. decimal digit) to the accuracy of the fine non-adaptive grid for error upper bounds $\varepsilon_{max} = 10^{-4}$ and $\varepsilon_{max} = 10^{-5}$.

Figures 4.20a and 4.20b present the x-velocity profiles at section $x = 0$ between the cylinder and lower channel wall for the adaptive grid solutions based on $\tilde{e}_{ls}$ and $\tilde{e}^{u}_{spectral}$ respectively. The spectral/hp least-squares penalty solution of Prabhakar and Reddy [42] which is associated with very good mass conservation properties is also presented as a reference. It is seen that the shape of the profile and the maximum value of the x-velocity is directly related to the mass conservation performance. The profile corresponding to the coarse non-adaptive grid solution underestimates the maximum value of the x-velocity significantly. In contrary, the adaptive solutions with $\varepsilon_{max} = 10^{-4}$ and $\varepsilon_{max} = 10^{-5}$ are in excellent agreement with Prabhakar and Reddy [42]'s results. Even grid (b2) with $\varepsilon_{max} = 10^{-3}$ agrees well with the reference. Since the majority of the mass loss occurs in the upstream of the cylinder, it is informative to examine the variation of centerline x-velocity with distance in x. Figures 4.21a and 4.21b illustrate the centerline x-velocity plotted against distance x in the upstream of the cylinder. By physical intuition, it is expected that the centerline x-velocity should increase with x as the boundary layers at the upper and lower channel walls develop and restrict the area of core flow. However the opposite is observed for the solutions with coarse non-adaptive grid and adaptive grids with large error upper bounds indicating severe mass loss. The expected increase in centerline x-velocity is observed with the rest of the adaptive solutions.

Up to this point, it has been observed that adaptive-refinement procedures based on $\tilde{e}_{ls}$ and $\tilde{e}^{u}_{spectral}$ result in comparable mass conservation enhancement over the coarse non-adaptive grid solution. In particular, solutions with $\varepsilon_{max} = 10^{-4}$ and $\varepsilon_{max} = 10^{-5}$ conserve mass as good as the fine non-adaptive grid solution. However, the aim of an adaptive refinement strategy is to refine the regions of the grid that require higher resolution selectively. In this problem, the regions that require higher resolution are identified from table 4.5 as the upstream of the cylinder, especially the regions close to the inflow boundary and the vicinity of the cylinder. Downstream of the cylinder away from the recirculation region does not experience any significant mass flow rate change.The success of adaptive refinement procedures employed in this study can be discussed based on this observation. The adaptive grids that result from different error upper bounds for $\tilde{e}_{ls}$ and $\tilde{e}^{u}_{spectral}$ are illustrated in figures 4.22 and 4.23 respectively.

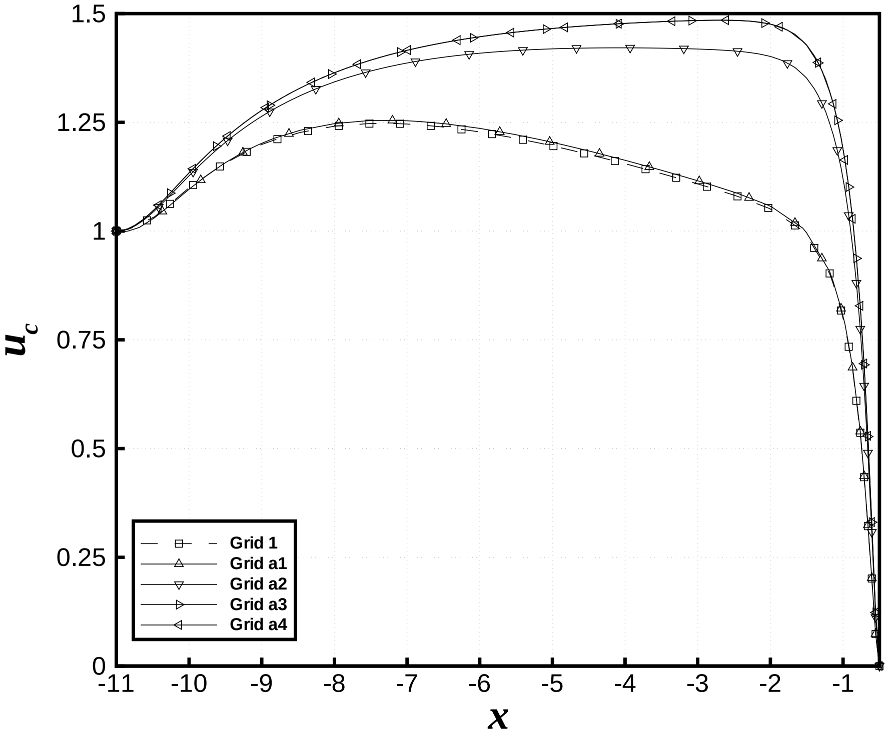
The adaptive refinement procedure based on $\tilde{e}_{ls}$ refines only the elements that are adjacent to the inflow boundary for high error upper bound ($\varepsilon_{max} = 10^{-2}$). The reason of this refinement is most likely due to corner singularities induced by uniform inflow boundary conditions rather than the mass conservation violation. As the error upper bound is decreased to 10^{-3}, the rest of the upstream and the vicinity of the cylinder are refined as well while leaving the far downstream unrefined as expected. At this level of error criterion, the emphasis is on the



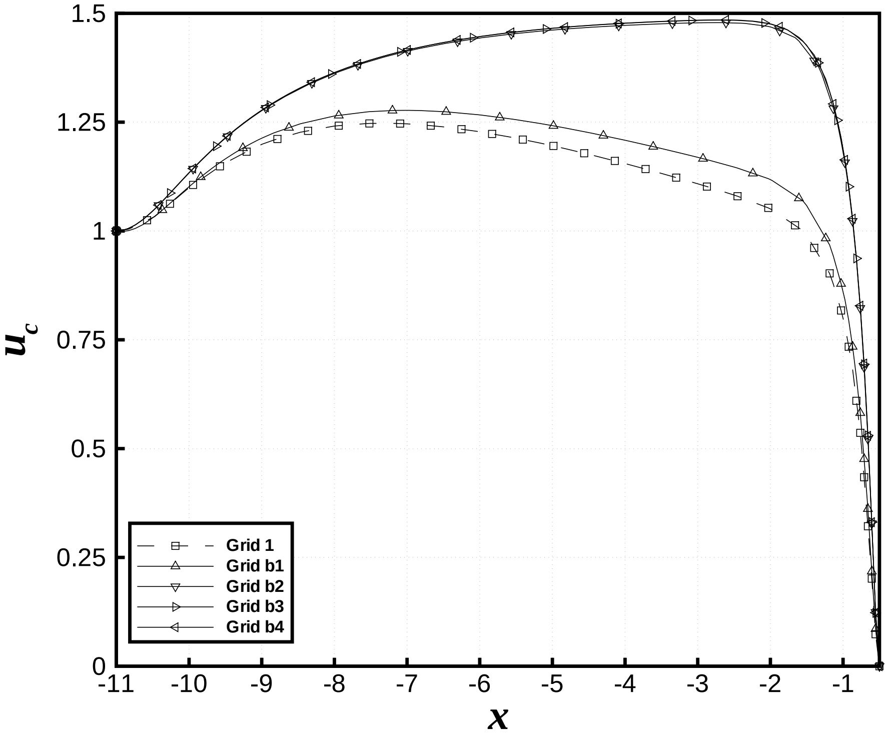
(a) Error estimate: $\tilde{e}_{ls}$

(b) Error estimate: $\tilde{e}^{u}_{spectral}$

Figure 4.20: Flow past a cylinder in a channel: x-velocity profiles at $x = 0$ between the cylinder and lower channel wall for adaptive grid solutions



(a) Error estimate: $\tilde{e}_{ls}$

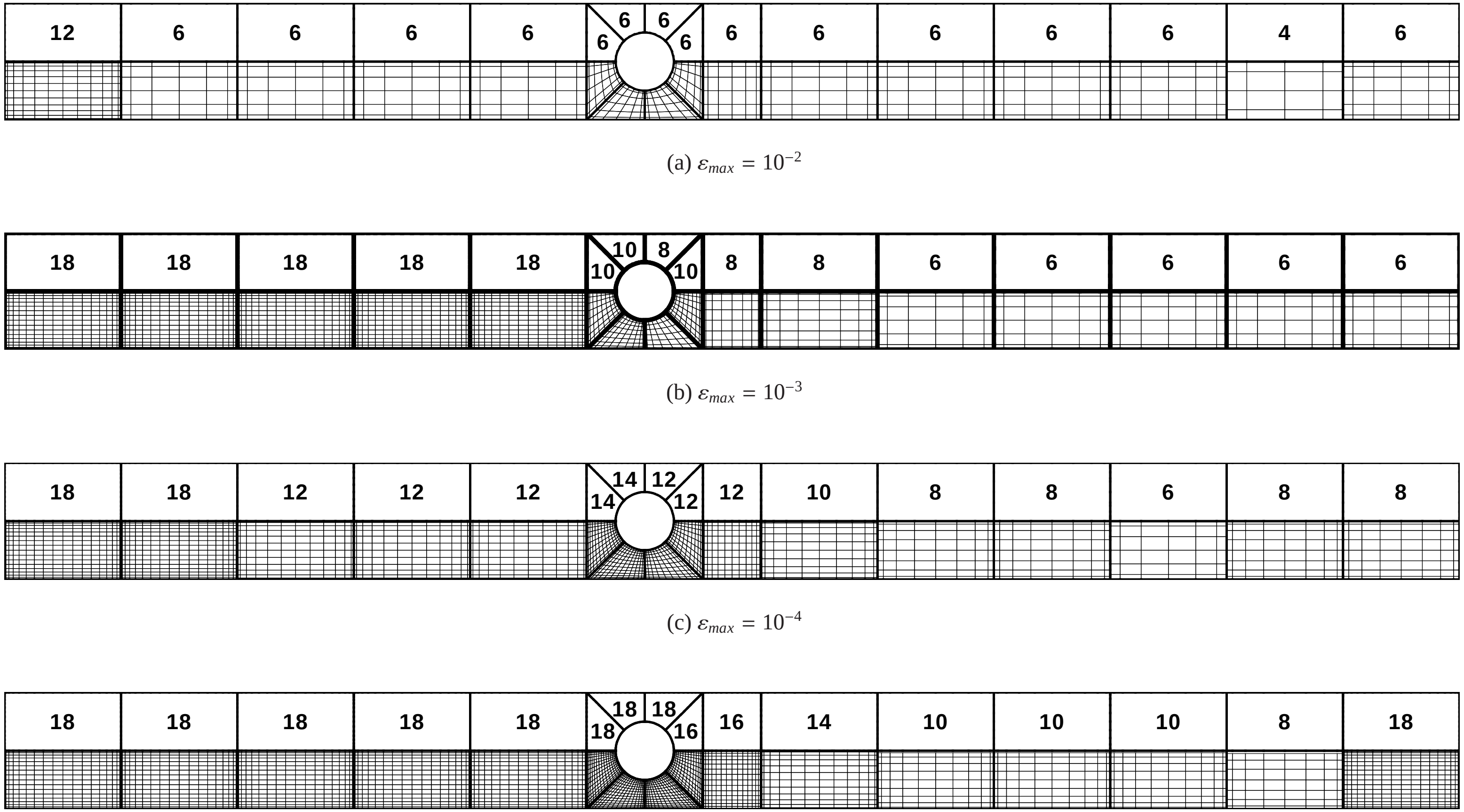


(b) Error estimate: $\tilde{e}^{u}_{spectral}$

Figure 4.21: Flow past a cylinder in a channel: Centerline x-velocity (u_c) profiles along the upstream of the cylinder

whole upstream of the domain and the expansion orders there reach the maximum allowable expansion order: $p_{max} = 18$. The vicinity of the cylinder is refined less when compared to the upstream. When the error upper bound is decreased to 10^{-4} some interesting changes in the refinement pattern are observed. The upstream of the cylinder away from the inflow boundary are less emphasized than the previous level of error criterion. In fact, even though the error criterion is made stricter, the expansion orders of the elements at this region are less than those in the case of $\varepsilon_{max} = 10^{-3}$. Instead, the vicinity of the cylinder and recirculation zone undergo several levels of refinement. It can be argued that the coarsening in the upstream for $\varepsilon_{max} = 10^{-4}$ may result in inferior mass conservation performance when compared to $\varepsilon_{max} = 10^{-3}$. However, this is not the case as seen from table 4.5. Grid (a3) actually conserves mass better than grid (a2) at all of the cross-sections reported in the table. Therefore it can be concluded that the mass conservation is improved by the interplay of refinements in the upstream and in the vicinity of the cylinder rather than refinements made in individual regions. The solution with $\varepsilon_{max} = 10^{-4}$ captures this coupling more accurately than the solution with $\varepsilon_{max} = 10^{-3}$. As the error upper bound is decreased further to $\varepsilon_{max} = 10^{-5}$, the upstream and the vicinity of the cylinder are refined up to $p_{max} = 18$ while the refinement levels in the downstream are lower. The refinements based on $\varepsilon_{max} = 10^{-5}$ results in little improvement in mass conservation over the solution with $\varepsilon_{max} = 10^{-4}$.

The adaptive refinement procedures based on $\tilde{e}^{u}_{spectral}$ result in refinement patterns similar to the case of $\tilde{e}_{ls}$. The major difference is that $\tilde{e}^{u}_{spectral}$ emphasizes both the upstream near the inflow boundary and the vicinity of the cylinder for refinement even in the case of high error upper bounds. In other words, the interplay of vicinity of the cylinder and the upstream is captured well for all levels of error criterion. This observation is justified by the mass flow rates reported in table 4.5 where grids (b1) and (b2) result in significantly higher mass flow rates than grids (a1) and (a2). As in the case of $\tilde{e}_{ls}$, the far downstream is refined less than other regions. As a side note, it can be argued that the deficiency of the mass conservation in this problem is caused by the unphysical corner singularities induced by the uniform inflow boundary condition rather than poor mass conservation properties of LSSEM. Nevertheless, the problem was also solved with a fully developed parabolic inflow boundary condition and similar amounts of mass loss were observed in that version of the problem. In this study, the results for uniform inflow boundary condition are presented to stay consistent with the literature [42].



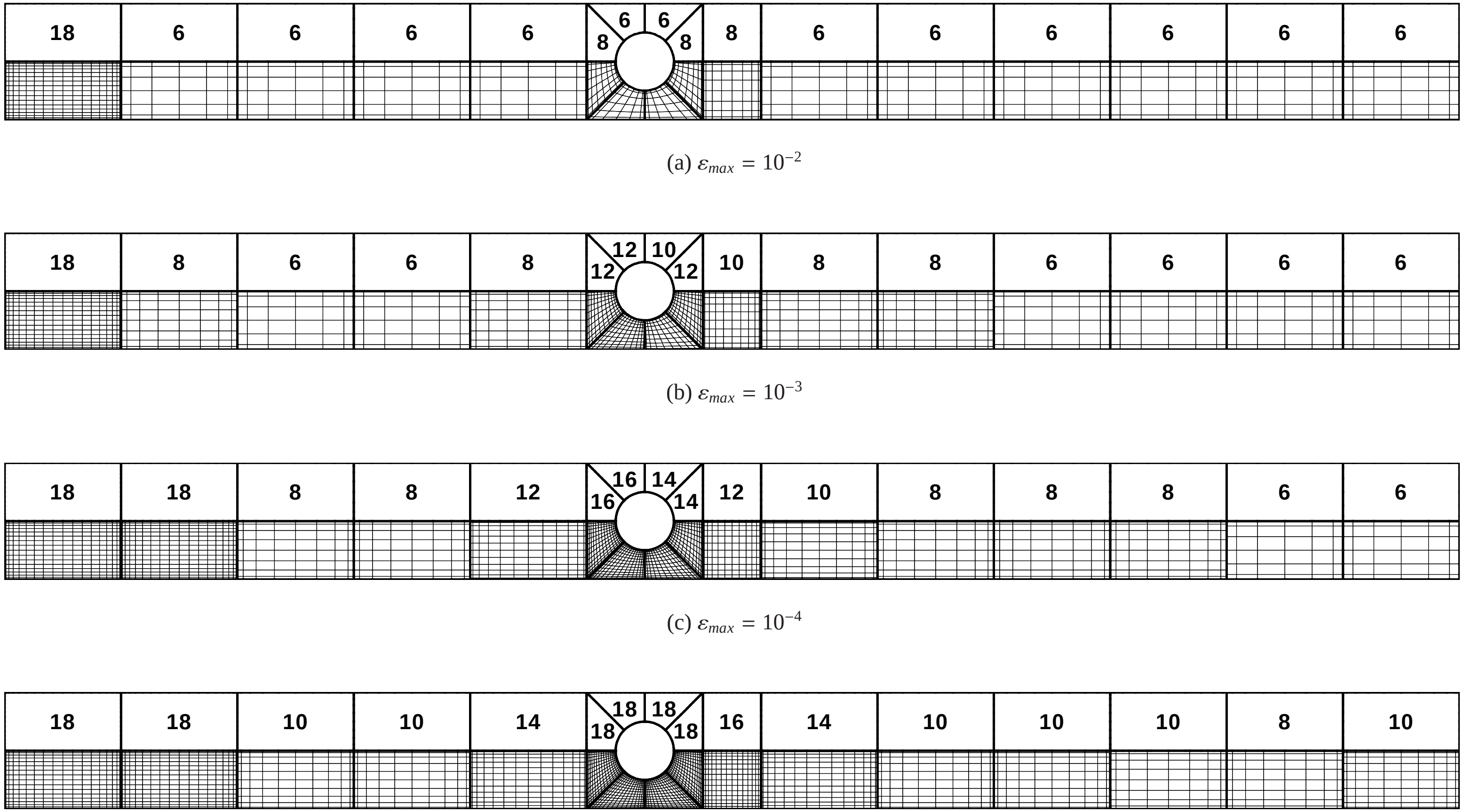
(a) $\varepsilon_{max} = 10^{-2}$

(b) $\varepsilon_{max} = 10^{-3}$

(c) $\varepsilon_{max} = 10^{-4}$

(d) $\varepsilon_{max} = 10^{-5}$

Figure 4.22: Flow past a cylinder in a channel: Adaptive grids that result from different error upper bounds for $\tilde{e}_{ls}$. The numbers in the upper half of the domain indicate the expansion orders employed in individual elements. Note that adaptive refinement is generally symmetric with respect to horizontal centerline.



(a) $\varepsilon_{max} = 10^{-2}$

(b) $\varepsilon_{max} = 10^{-3}$

(c) $\varepsilon_{max} = 10^{-4}$

(d) $\varepsilon_{max} = 10^{-5}$

Figure 4.23: Flow past a cylinder in a channel: Adaptive grids that result from different error upper bounds for $\tilde{e}^{u}_{spectral}$. The numbers in the upper half of the domain indicate the expansion orders employed in individual elements. Note that adaptive refinement is generally symmetric with respect to horizontal centerline.

Table 4.6: Flow past a cylinder in a channel: The number of degree of freedoms per dependent variable (N_{DOF}) and wall clock times of the solutions. The mass flow rate at $x = 0$ is also provided for comparison.

Grid	N_{DOF}	Wall Clock Time (s)	$\dot{m}_{(x=0)}$
1	1272	28	1.203
a1	1448	51	1.208
a2	4784	722	1.832
a3	4810	579	1.976
a4	8220	1351	1.986
b1	2026	165	1.307
b2	3016	231	1.954
b3	4872	569	1.982
b4	6730	1133	1.986
2	10728	3306	1.987

The practical advantages of p-type adaptive refinement over uniform p-type refinement can be seen by examining the number of degree of freedoms per dependent variable (N_{DOF}) used in the solutions and the wall clock solution times presented in table 4.6. Even the adaptive grid solutions with the lowest error upper bounds, (a4) and (b4), are completed in less than half the time of the solution with fine non-adaptive grid, grid (2). Solutions with higher error upper bounds present a better compromise between mass conservation accuracy and wall clock time. Especially the the adaptive refinement procedures that resulted in grids (a3), (b2), and (b3) come forward by being more than five times faster than fine non-adaptive grid solution while offering comparable mass conservation accuracy.

4.4 Unsteady Flow Past Cylinder

The problem of interest in this section is the unsteady flow past a circular cylinder. It solved to demonstrate the time dependent solution and adaptive refinement capabilities of LSSEM flow solver.

Flow past a cylinder exhibits different flow features for different Reynolds numbers. For slow flows ($Re < 50$) steady recirculation zones are observed behind the cylinder as illustrated in section 4.3. When some critical Reynolds number is exceeded ($Re^c \simeq 50$ [80]), the wake region begins to exhibit unsteady flow features that finally evolves into a periodic steady state where vortices of alternating directions are shed from top and bottom of the cylinder towards the downstream. This vortex structure is known as the *Von Karman Vortex Street*.

Figure 4.24 illustrates the problem setup. In order to decrease the effect of boundaries, the solution is performed on a large computational domain bounded by the rectangular region: $[-6, 20] \times [-6, 6]$ that corresponds to $l = H = 6$ and $L = 20$. A cylinder of unit diameter ($D = 1$) is positioned at the origin. It is assumed that the cylinder starts motion in a large bulk of fluid towards $-x$-direction at time $t = 0s$. The computational domain represents a control volume moving together with the cylinder. Therefore uniform inflow boundary condition with inflow velocity of 1 is specified at the upstream boundary. The upper and lower boundaries have the velocity components specified as $u = 1$ and $v = 0$ which is consistent with the fact that the flow remains stagnant sufficiently away from the cylinder. At the downstream boundary, a constant pressure boundary condition with $p = 0$ is specified. While other outflow boundary conditions such as specification of pressure only at midpoint of the boundary [8] or unsteady convective type of boundary conditions where a combination of pressure and velocity gradients is specified [62] are also possible. However, it is observed that constant pressure boundary condition and specification of pressure at midpoint result in similar solutions. The constant pressure boundary condition can be observed in flows discharging to atmosphere at the outflow boundary. The initial conditions involve uniform flow in x-direction with $u_0 = 1, v_0 = 0$ except at region where $x > 0$ and $y > 0$. In this region, a uniform flow with a lower velocity is specified: $u_0 = 0.3, v_0 = 0$. This artificial disturbance is used to have the onset of unsteady periodic vortex shedding earlier in solution.

The solutions are performed on a h-type conforming grid composed of $NE = 174$ elements

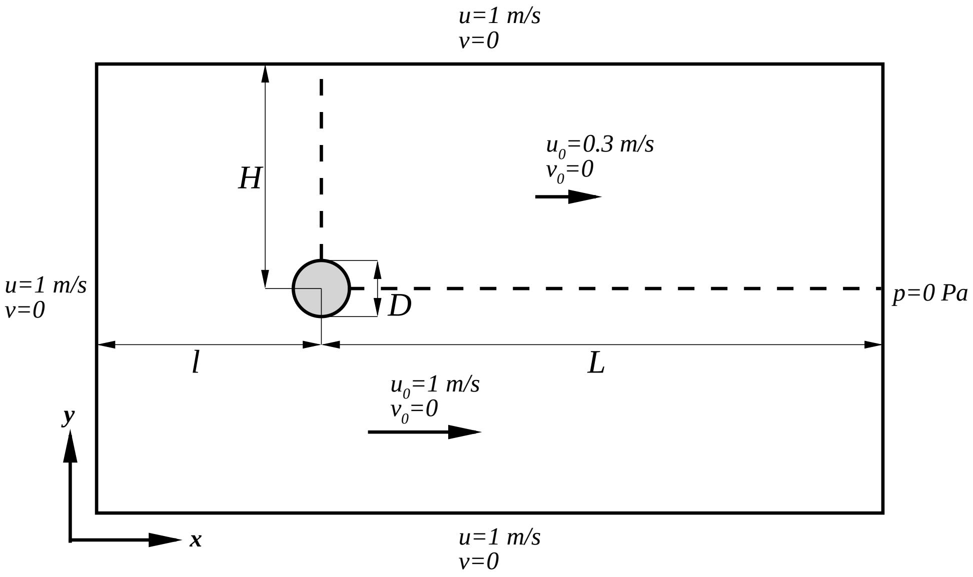
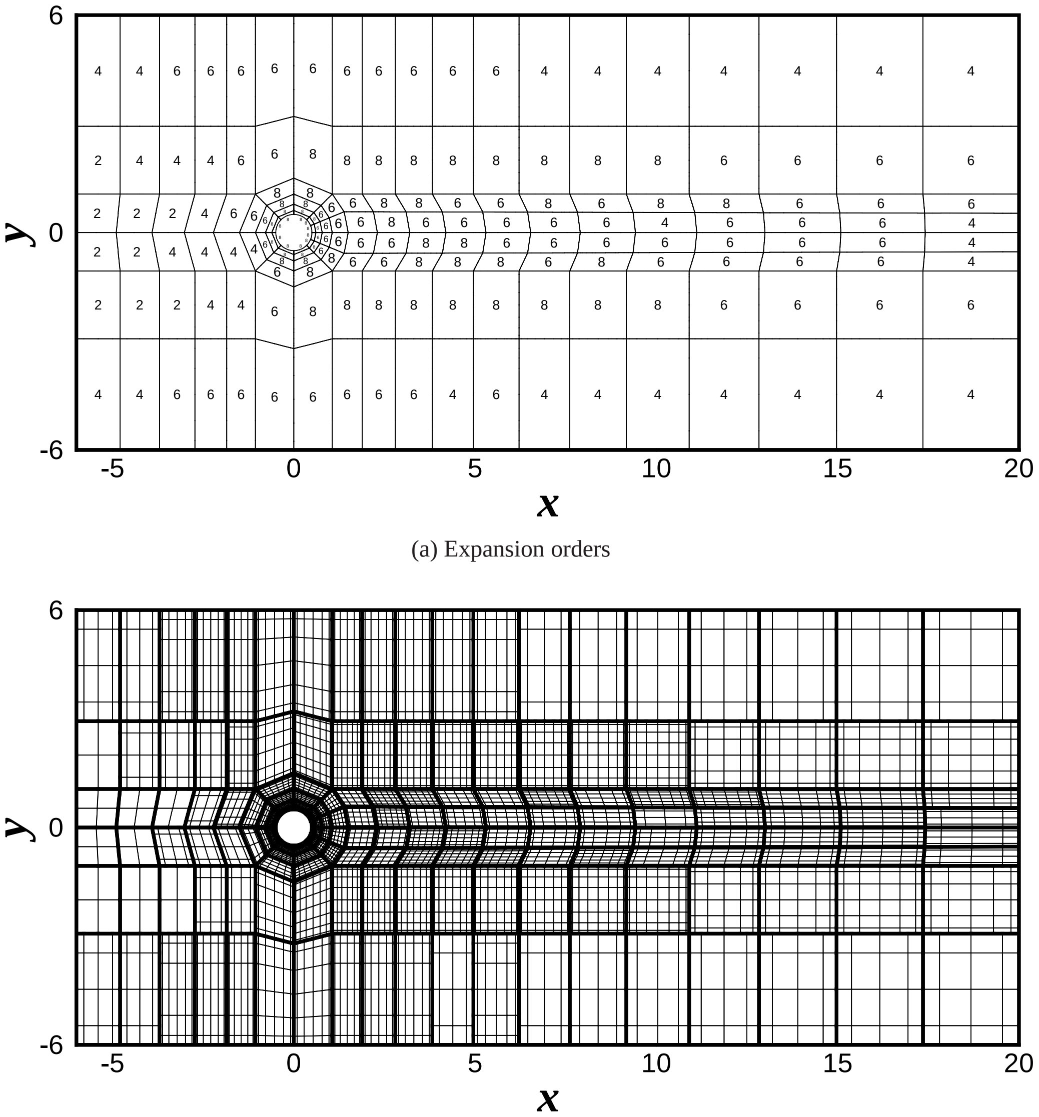


Figure 4.24: Unsteady flow past cylinder. Problem setup.

Table 4.7: Unsteady flow past a cylinder. Specifications of the computational grids

Grid no	NE	p	Error Estimate	$(\varepsilon_{min}, \varepsilon_{max})$	(p_{min}, p_{max})
1	174	4	Non-adaptive	-	-
2	174	4	$\tilde{e}^u_{spectral}$	$(10^{-4}, 10^{-3})$	$(2, 8)$

as illustrated in Figure 4.25a. Grid is kept finer near the cylinder and along the downstream where the vortex street is expected to be observed. The solutions are first performed on a non-adaptive grid, referred as *grid (1)*, where elements have expansion orders $p = 4$ in both space dimensions. Solutions are also obtained on an adaptive grid, referred as *grid (2)*, where $\tilde{e}^u_{spectral}$ is used as the error estimate with $(\varepsilon_{min}, \varepsilon_{max}) = (10^{-4}, 10^{-3})$ and $(p_{min}, p_{max}) = (2, 8)$ with 2 increments. The solution begins from $t = 0$ and integrated up until $t = 200 - 250$ seconds to make sure that periodic unsteady solution is well developed. For solutions with adaptive grid, the solutions of non-adaptive grid at $t = 100$ and $t = 150$ seconds are provided as the initial guesses to reduce the turn around time of the simulations. The specifications of the grids are summarized in table 4.7. For time dependent problems, an α-family time stepping scheme was introduced in Chapter 2. It is desirable to employ either one of Crank-Nicholson ($\alpha = 0.5$) or Galerkin schemes ($\alpha = 2/3$) that are both second order accurate in time. However, numerical experiments conducted in this study reveal that use of those schemes in solution of this problem leads to temporal instabilities. The velocity field becomes unbounded during long time integration. It is found that first order accurate implicit Euler scheme ($\alpha = 1.0$), on the other hand, results in stable periodic solution. Therefore, the



(a) Expansion orders

(b) Detailed grid

Figure 4.25: Unsteady flow past a cylinder. Instantaneous adaptive computational grid at *t* = 195 s. NE=174.

solutions presented in this study are obtained by implicit Euler scheme.

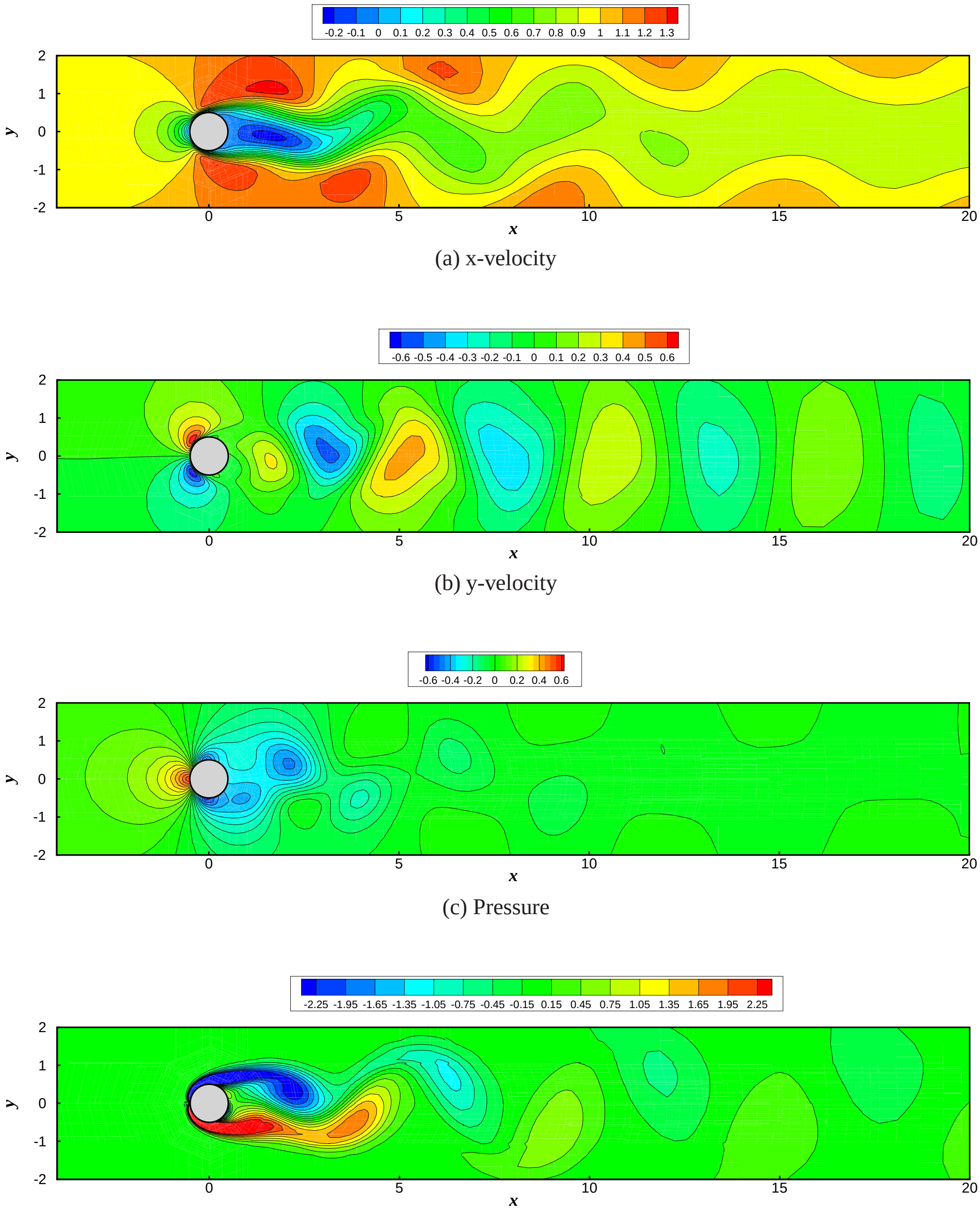
The instantaneous contours of flow variables at $t = 190$ s are presented in figure 4.26 It can be observed from the vorticity contours that the vortices shed from the top and bottom of the cylinder have alternating directions and a decaying strength as flow moves towards the downstream. The time history of the y-velocity or vorticity at a point along the horizontal centerline of the domain can be used to judge the periodicity of the solution. Figure 4.27a presents such a time history plot at $(x = 2.3, y = 0)$. It is seen that the velocity field has reached to a well-developed steady periodic solution by $t = 180$ s.

When the time history of the pressure at $(x = 2.3, y = 0)$ is examined, it is observed that the periodic solution of pressure is contaminated by intermittent jumps. This is obviously is not desirable. It can be argued that this unrealistic behavior of the pressure field may be caused by the constant pressure boundary condition at the downstream boundary, that is, the pressure field is overdetermined by the boundary condition. However, same pressure jumps are observed even when the problem is solved with a pressure specified only at the midpoint of the downstream boundary. Secondly, the ill-behaved temporal evolution of the pressure field is also reported in the literature. In particular, Pontaza [39] also observed a similar problem and argued that the temporal pressure oscillations are caused by the poor velocity-pressure coupling in the least-squares formulation. A remedy which improves the velocity-pressure coupling numerically by regularizing the continuity equation with pressure perturbations was also suggested [39]. While this approach is implemented in LSSEM flow solver, the preliminary studies are inconclusive and this issue is a subject of further study. It is seen in figure 4.27c that the jumps in the pressure field also affects the vorticity field but up to small extend and the y-velocity is continuous regardless of the pressure field.

Despite the intermittent jumps in the pressure field, it can shown that the results obtained in this study compare well with the literature based on some flow metrics. One measure for this purpose is the dimensionless Strouhal number based on cylinder diameter that is calculated from the shedding frequency (or period) as follows:

$$St_D = \frac{\text{Characteristic Flow Time}}{\text{Period of the Oscillation}} = \frac{fD}{u_\infty} = \frac{D}{\tau u_\infty} \tag{4.4}$$

where f is the vortex shedding frequency, τ is the vortex shedding period, D is the cylinder diameter, and u_∞ is the free stream velocity. The amplitude of the y-velocity oscillations at a specific point can also be used as a measure of the quality of the solution.



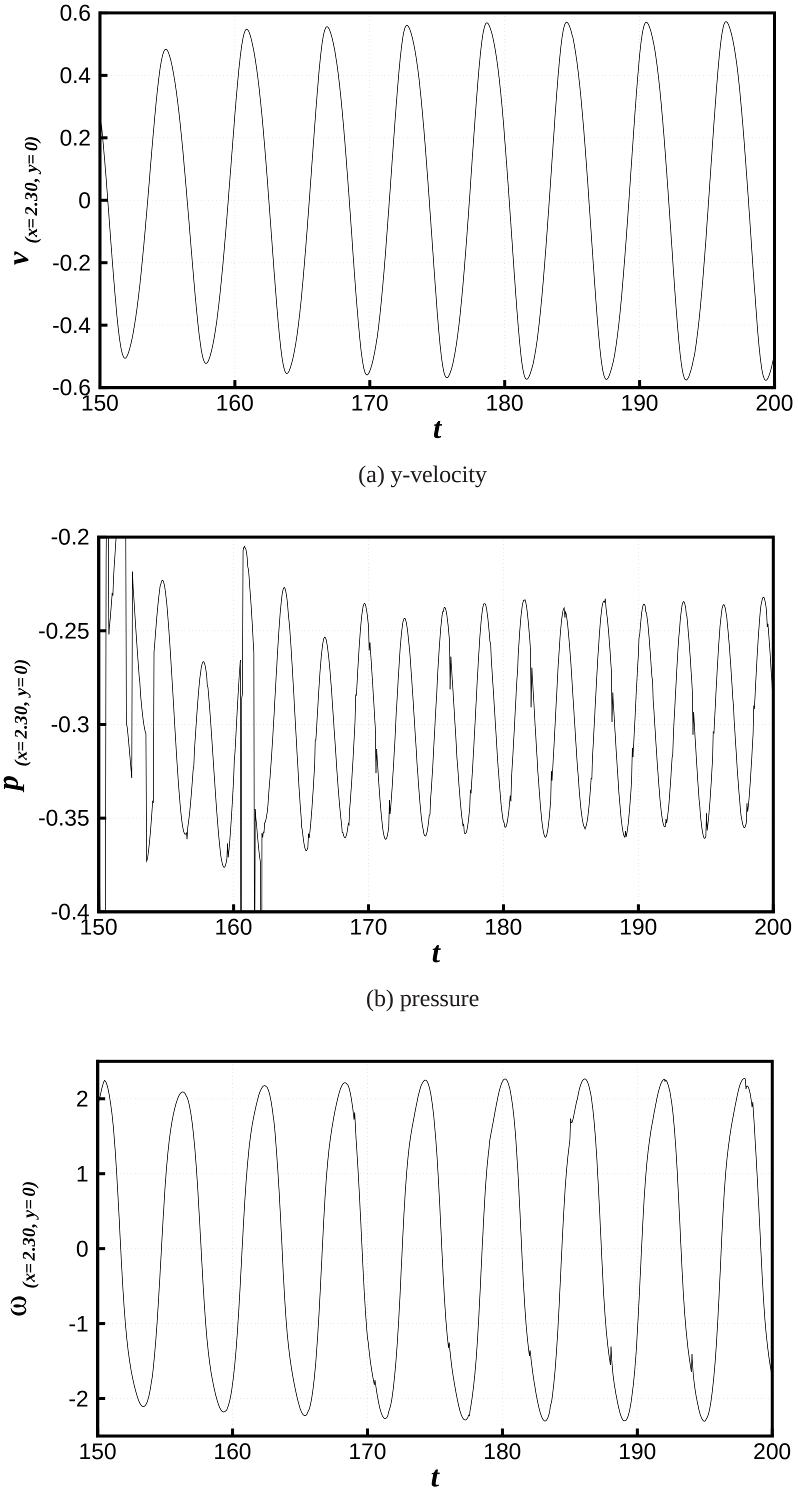
(a) x-velocity

(b) y-velocity

(c) Pressure

(d) Vorticity

Figure 4.26: Unsteady flow past a cylinder. Instantaneous contours of dependent variables at $t = 190$ s



(a) y-velocity

(b) pressure

(c) vorticity

Figure 4.27: Unsteady flow past a cylinder. Time history plots of several flow variables at $x(=2.3m, y=0)$.

Table 4.8 presents the Strouhal numbers and maximum value of the y-velocity at $(x = 2.0, y = 0)$ for solutions obtained by grids (1) and (2) with various time steps in comparison to available data in the literature. First of all, it is observed that the results improve by decrease in time step and increase in grid resolution through adaptive refinement independently. To illustrate this point, the non-adaptive grid, grid 1, with $\Delta t = 0.1$ s, overestimates the vortex shedding period and underestimates the amplitude of the y-velocity oscillations considerably. the adaptive grid with the same time step results in flow metrics closer to the common results in the literature. When the time step is decreased regardless of the spatial grid resolution, the estimated vortex shedding period decreases and drops below the values reported in the literature. On the other hand, the amplitude of the y-velocity approaches to value reported by Pontaza and Reddy [62]. It is seen that the convergence upon decrease of time step is not achieved for time steps reported in table 4.8. This lack of convergence can be explained by the fact that, the implicit Euler scheme is only first order accurate in time. Therefore it may be necessary to solve the problem for even lower time steps to achieve convergence with respect to time steps. It is worth to mention that the results reported by the references in table 4.8 were obtained by higher order accurate time stepping schemes. In particular, a third order backward differencing scheme [39], a third order Adams-Bashforth scheme [81] a second order Crank-Nicholson scheme [82] were used with time steps similar to these used in the present study. The results of Pontaza and Reddy [62] were obtained by a space time coupled LSSEM where time was also discretized by third order spectral elements. The second possible explanation for the lack of convergence in the present study is the existence of the pressure jumps that become more pronounced as the time step is decreased. Regardless, it is seen that the vortex shedding periods presented in this study are within 5 % of the values reported in the literature which is sufficient for most engineering applications.

It is worth to mention that the time dependent formulation, eqn. (2.8), presented in Chapter 2 results in deteriorating numerical results as the time step is decreased below certain values. In particular, it has not been possible to obtain stable and bounded flow fields with time steps less than $\Delta t = 0.1$ s with the original formulation. The results presented in table 4.8 for $\Delta t < 0.1$ s could be obtained by scaling the momentum equation, eqn. (2.8), by Δt based on the work of Pontaza [39] as follows:

$$\mathbf{u} + \alpha \Delta t \left(\mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p + \nu \nabla \times \omega \right) = \Delta t \, \mathbf{f}^* \tag{4.5}$$

Then the momentum equation is included in the least-squares functional in the form of eqn.

Table 4.8: Unsteady flow past a cylinder. Strouhal numbers and amplitude of y-velocity at $x = 2\,m$, $y = 0$

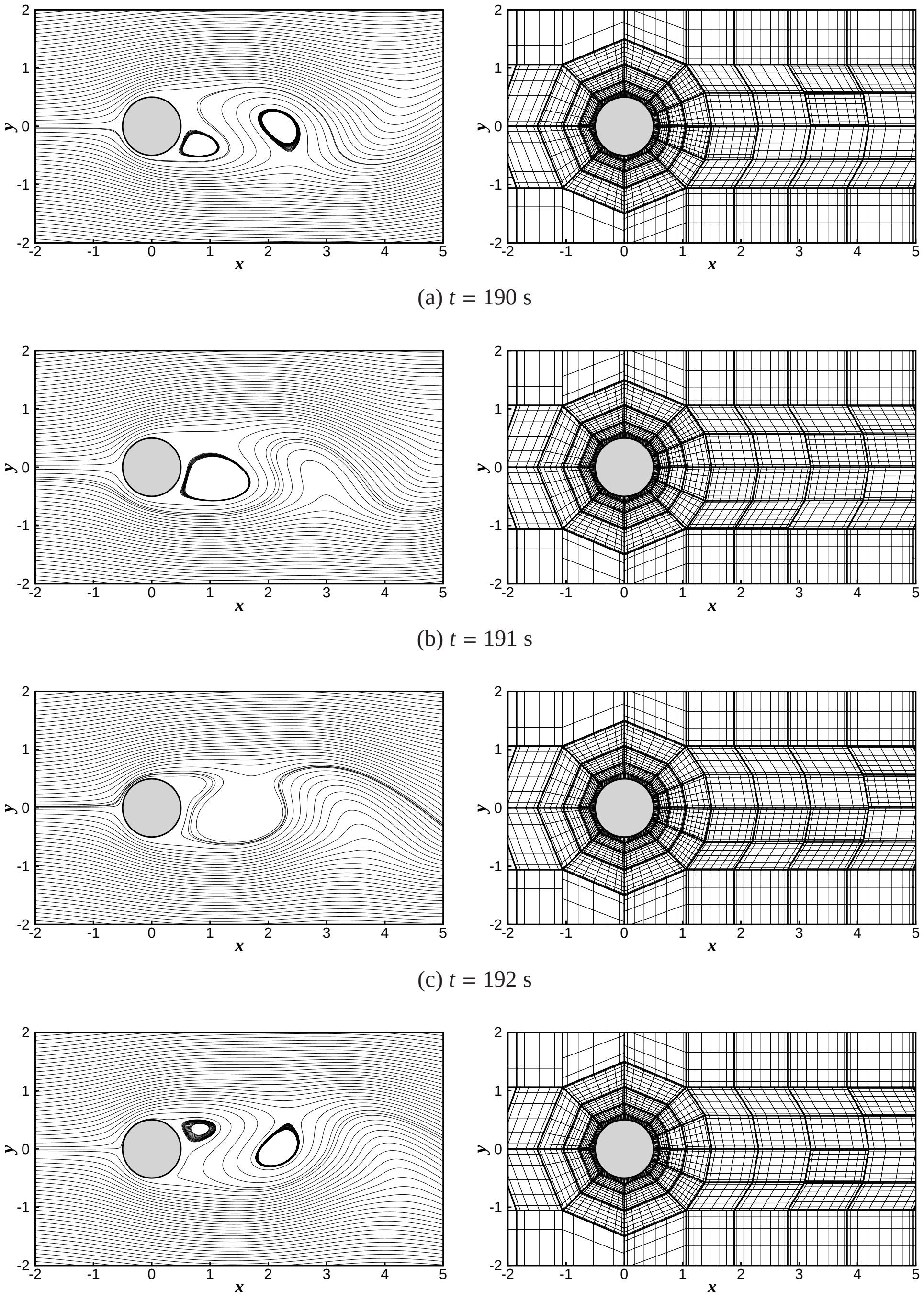
Grid	Δt	τ	St_D	$v_{max}(x = 2, y = 0)$
Grid 1	0.1	6.40	0.1563	0.34
	0.05	6.05	0.1563	0.44
	0.02	5.90	0.1695	0.47
	0.05	5.84	0.1712	0.47
Grid 2	0.1	6.10	0.1639	0.42
	0.05	5.95	0.1681	0.51
	0.02	5.80	0.1724	0.51
Williamson [83]		6.09	0.1643	
Sherwin and Karniadakis [81]		6.00	0.1667	
Pontaza and Reddy [62]		6.05	0.1653	0.55
Pontaza [39]		6.00	0.1667	
Rajani et al. [82]		6.38	0.1569	

(4.5). The scaling of the momentum equation eliminates the term $\frac{\mathbf{u}}{\Delta t}$ which grows fast as time step is decreased with respect to other terms in the least-squares functional. The conditioning of the linear system is also improved, leading to faster linear system solution with the preconditioned conjugate gradient solver.

Another aim of this study is to investigate the application of p-type adaptive refinement procedures to time dependent problems. The past three sections dealed with steady state solutions where the adaptivity of the grid across time steps was not of particular concern. Here, the periodicity of the vortex shedding structure enables one to study the time response of adaptive grid. In this study, only the error estimate based on spectral expansion coefficients with a single set of error upper and lower bounds is considered for simplicity. Figure 4.25 illustrates the instantaneous computational grid at $t = 195$ s as a representation of distribution of grid resolution across the domain. The expansion orders of individual elements are shown in figure 4.25a and the detailed grid is shown in figure 4.25b. It is observed that the adaptive refinement procedure refines the elements that are immediately near the cylinder and in the near wake region. The upstream away from the cylinder is actually unrefined to expansion order $p = 2$. The elements near the upper and lower boundaries are either not refined or refined up to $p = 6$. Therefore the overall distribution of grid resolution is concentrated near the cylinder and it's wake region where the vortices are shed.

Figure 4.28 illustrates the time evolution of the streamlines and the grid resolution across one vortex shedding period. At $t = 190$ s, a vortex starts to emerge from the bottom of the cylinder.

As it moves towards the downstream and dissipate, another vortex emerges from the top of the cylinder. At t = 196 s, another vortex emerges from the bottom of the cylinder as before, completing the periodic behavior. It is important to note that at each time the computational grid evolves such that the vortex edges where high velocity gradients are present, are resolved with elements with highest expansion order available, i.e. $p = 8$. The other regions are kept at a lower expansion order. The refinement-unrefinement action of the computational grid follows the same period as the vortex shedding. This observation suggests that the adaptive grid refinement captures the periodic unsteady nature of the flow.



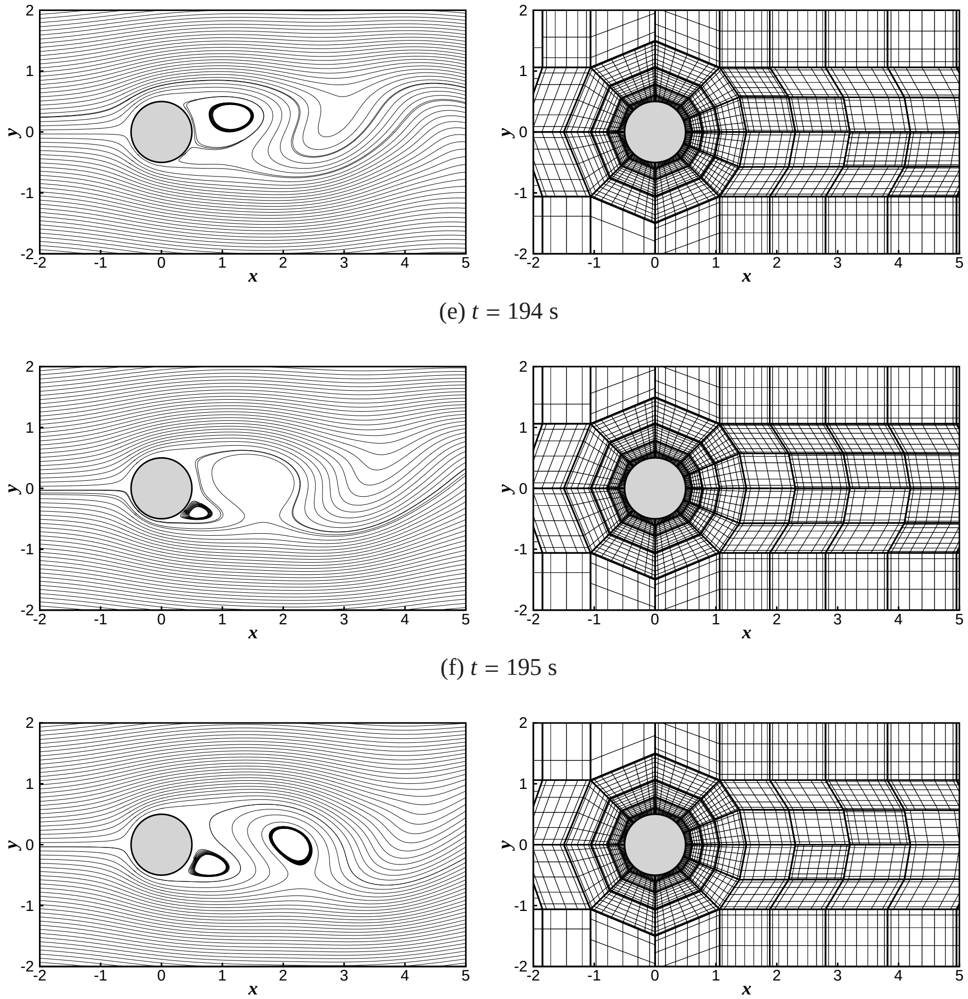
(a) t = 190 s

(b) t = 191 s

(c) t = 192 s

(d) t = 193 s

Figure 4.28: Unsteady flow past a cylinder. Time evolution of the streamlines and the grid resolution



(e) *t* = 194 s

(f) *t* = 195 s

(g) *t* = 196 s

Figure 4.28: (Cont.) Unsteady flow past a cylinder. Time evolution of the streamlines and the grid resolution

CHAPTER 5

CONCLUSION

In this study an two-dimensional incompressible flow solver based on least-squares spectral element method (LSSEM) was developed. The LSSEM flow solver can work on hp-type non-conforming computational grids and it can perform p-type adaptive refinement.

Several test problems were solved to validate the LSSEM flow solver and successful results were obtained. In particular it was shown that LSSEM flow solver exhibits the favorable exponential convergence characteristics of spectral element methods upon p-type refinement even in the presence of h-type and p-type nonconformities. It was also observed that both the least-squares functional based error estimate $\tilde{e}_{ls}$ and the spectral expansion coefficients based error estimate $\tilde{e}_{spectral}$ follow the trends of the change in relative true error with expansion order and therefore can be used as qualitative error estimates for p-type adaptive refinement.

It was also shown that problems with singular solutions like the lid driven cavity flow can be solved accurately by LSSEM flow solver on an hp-type non-conforming grid. Obtaining an accurate solution becomes challenging as Reynolds number increases. In that case, the accuracy of the solution were improved by p-type adaptive refinement capabilities of LSSEM flow solver.

The studies regarding the steady channel flow past a cylinder revealed that the mass conservation performance of LSSEM can be enhanced by using p-type adaptive refinement strategies up to the same accuracy as the uniform refinement while keeping computational costs practical. This is the first use of p-type adaptive refinement in the literature as a way to improve the mass conservation properties of LSSEM to the best of author's knowledge. While both error estimates result in grid adaptations that capture the regions requiring refinement correctly,

$\tilde{e}_{spectral}$ may be preferable to $\tilde{e}_{ls}$ since it shows better compromise between mass conservation accuracy and computational cost. Regardless, this study shows that the poor mass conservation properties of LSSEM may not be an issue when the solution is obtained by p-type adaptive refinement strategies.

The last test problem featured an unsteady flow past a circular cylinder where the periodic vortex shedding action was captured correctly and the related flow metrics agreed with the literature within an accuracy of 5%. The low order accuracy of the time stepping scheme and the lack of velocity-pressure coupling in the least-squares method were identified as the obstacles to obtaining more accurate solutions. It was also noted that the adaptive refinement procedure based on $\tilde{e}_{spectral}$ followed the same period as the vortex shedding, resolving regions with high velocity gradients at each time step. As a result the adaptive grid clearly captured the unsteady periodic nature of the flow.

The following lines of research can be suggested as a future work to the present study:

- The LSSEM flow solver currently works on computational grids that is composed of quadrilateral elements only. Allowing triangular spectral elements will enable use of cut-cell method with non-conforming Cartesian grids. Such a future will provide great flexibility in solving problems with complex immersed boundaries.

- The LSSEM flow solver currently has the ability to work on hp- type non-conforming grids and perform p-type adaptive refinement. If LSSEM flow solver is coupled with a non-conforming Cartesian grid generator, hp-type adaptive refinement strategies can be realized.

- It is observed that the current time stepping scheme, implicit Euler method, may be improved to increase the accuracy in time. Problems inherent to least-squares formulation like the poor velocity-pressure coupling must also be solved to avoid ill-behaved pressure fields in time dependent solutions. For this purpose, high order multi-step time integration schemes based on backward differencing formulas can be employed. The regularized continuity equation approach of Pontaza [39] may be effective in mitigating the ill behavior of the pressure field.

- LSSEM flow solver currently solves the linear system of equations using an element-by-element approach that holds all element systems in memory. Memory requirements can be reduced by implementing a matrix-free element-by-element solution procedure [21]. CPU time performance can be improved by using CPU parallelization and recent advances in GPU parallel computing hardware and software.

REFERENCES

[1] A. J. Chorin, “Numerical solution of Navier-Stokes equations,” *Mathematics of Computation*, vol. 22, no. 104, pp. 745–&, 1968.

[2] U. Ghia, K. Ghia, and C. Shin, “High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method,” *Journal of Computational Physics*, vol. 48, no. 3, pp. 387–411, 1982.

[3] E. Erturk, T. C. Corke, and C. Gökçöl, “Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers,” *International Journal for Numerical Methods in Fluids,* vol. 48, no. 7, pp. 747–774, 2005.

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

[5] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*. Dover, 2000.

[6] J. N. Reddy, *Energy Principles and Variational Methods in Applied Mechanics*. John Wiley & Sons, Inc., 2002.

[7] O. C. Zienkiewicz and J. Wu, “Incompressibility without tears - how to avoid restrictions of mixed formulation,” *International Journal for Numerical Methods in Engineering*, vol. 32, no. 6, pp. 1189–1203, 1991.

[8] B. Jiang, *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics*. Springer-Verlag, 1998.

[9] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy, “Toward a universal h-p adaptive finite element strategy, part 1. constrained approximation and data structure,” *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 79 – 112, 1989.

[10] A. Björck, *Numerical Methods For Least Squares Problems*. SIAM, 1996.

[11] J. H. Bramble and A. H. Schatz, "Least squares methods for 2mth order elliptic boundary-value problems," *Mathematics of Computation*, vol. 25, no. 113, pp. 1–32, 1971.

[12] J. P. Pontaza and J. N. Reddy, "Spectral/*hp* least-squares finite element formulation for the Navier– Stokes equations," *Journal of Computational Physics*, vol. 190, pp. 523–549, 2003.

[13] P. P. Lynn and S. K. Arya, "Use of the least squares criterion in the finite element formulation," *International Journal for Numerical Methods in Engineering*, vol. 6, no. 1, pp. 75–88, 1973.

[14] O. C. Zienkiewicz, D. R. J. Owen, and K. N. Lee, "Least square-finite element for elasto-static problems. use of reduced integration," *International Journal for Numerical Methods in Engineering*, vol. 8, no. 2, pp. 341–358, 1974.

[15] G. E. Karniadakis and S. J. Sherwin, *Spectral*/hp *Element Methods For Computational Fluid Dynamics*. OXFORD University Press, 2005.

[16] O. Zienkiewicz, R. L. Taylor, and P. Nithiarasu, *The Finite Element Method For Fluid Dynamics*, 6th ed. Elsevier Butterworth-Heinemann, 2005.

[17] I. Babuska, "Error bounds for finite element method," *Numer. Math.*, vol. 16, pp. 322–333, 1971.

[18] P. B. Bochev and M. D. Gunzburger, "Finite element methods of least-squares type," *SIAM Rev.*, vol. 40, no. 4, pp. 789–837, 1998.

[19] J. N. Reddy, *An Introduction To The Finite Element Method*, 3rd ed. McGraw-Hill, 2006.

[20] B.-N. Jiang, "A least-squares finite element method for incompressible Navier-Stokes problems," *International Journal for Numerical Methods in Fluids*, vol. 14, no. 7, pp. 843–859, 1992.

[21] B. N. Jiang and V. Sonnad, "Least-squares solution of incompressible navier-stokes equations with the p-version of finite elements," *Computational Mechanics*, vol. 15, no. 2, pp. 129–136, Nov. 1994.

[22] B.-N. Jiang, “Least-squares finite element methods with element-by-element solution including adaptive refinement,” Ph.D. dissertation, The University of Texas at Austin, 1986.

[23] M. M. J. Proot, “The least-squares spectral element method: Theory, implementation and application to incompressible flows,” Ph.D. Thesis, Technische Universiteit Delft, February 2003.

[24] J. P. Pontaza, “Least-squares variational principles and the finite element method: Theory, formulations and models for solid and fluid mechanics,” Ph.D. dissertation, Texas A&M University, December 2003.

[25] W. Heinrichs, “Least-squares spectral collocation for the Navier–Stokes equations,” *Journal of Scientific Computing*, vol. 21, no. 1, pp. 81–90, Aug. 2004.

[26] X. Ding and T. T. H. Tsang, “On first-order formulations of the least-squares finite element method for incompressible flows,” *International Journal of Computational Fluid Dynamics*, vol. 17, no. 3, pp. 183–197, 2003.

[27] B.-N. Jiang and L. A. Povinelli, “Least-squares finite element method for fluid dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 81, no. 1, pp. 13 – 37, 1990.

[28] B.-N. Jiang, “On least-squares method,” *Comput. Methods Appl. Mech. Engrg.*, vol. 152, no. 1998, pp. 239–257, 1998.

[29] P. B. Bochev and M. D. Gunzburger, “Analysis of least squares finite element methods for the Stokes equations,” *Mathematics of Computation*, vol. 63, no. 208, pp. 479–506, Oct. 1994.

[30] J. M. Deang and M. D. Gunzburger, “Issues related to least-squares finite element methods for the Stokes equations,” *SIAM Journal of Scientific Computing*, vol. 20, no. 3, pp. 878–906, 1998.

[31] P. B. Bochev and M. D. Gunzburger, *Least-Squares Finite Element Methods*, S. S. Antman, J. E. Marsden, and L. Sirovich, Eds. Springer, 2009.

[32] B. C. Bell and K. S. Surana, “A space-time coupled p-version least-squares finite element formulation for unsteady fluid dynamics problems,” *International Journal for Numerical Methods in Engineering*, vol. 37, no. 20, pp. 3545–3569, 1994.

[33] B. C. Bell and K. S. Surana, “A space-time coupled p-version least squares finite element formulation for unsteady two-dimensional Navier–Stokes equations,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2593–2618, 1996.

[34] J. P. Pontaza, “Least-squares variational principles and the finite element method: theory, formulations, and models for solid and fluid mechanics,” *Finite Elements in Analysis and Design*, vol. 41, pp. 703–728, 2005.

[35] J. P. Pontaza and J. N. Reddy, “Least-squares finite element formulations for viscous incompressible and compressible fluid flows,” *Comput. Methods Appl. Mech. Engrg.*, vol. 195, pp. 2454–2494, 2006.

[36] M. M. J. Proot and M. I. Gerritsma, “Least-squares spectral elements applied to the Stokes problem,” *Journal of Computational Physics*, vol. 181, pp. 454–477, 2002.

[37] M. M. J. Proot and M. I. Gerritsma, “Application of the least-squares spectral element method using Chebyshev polynomials to solve the incompressible Navier-Stokes equations,” *Applied Mathematics and Computation*, vol. 38, pp. 155–172, 2005.

[38] M. Proot and M. Gerritsma, “Mass and momentum conservation of the least-squares spectral element method for the Stokes problem,” *Journal of Scientific Computing*, vol. 27, no. 1, pp. 389–401, Jun. 2006.

[39] J. P. Pontaza, “A least-squares finite element formulation for unsteady incompressible flows with improved velocity–pressure coupling,” *Journal of Computational Physics*, vol. 217, pp. 563–588, 2006.

[40] J. P. Pontaza, “A new consistent splitting scheme for incompressible Navier-Stokes flows: A least-squares spectral element implementation,” *Journal of Computational Physics*, vol. 225, pp. 1590–1602, 2007.

[41] J. P. Pontaza, “A spectral element least-squares formulation for incompressible Navier-Stokes flows using triangular nodal elements,” *Journal of Computational Physics*, vol. 221, no. 2, pp. 649 – 665, 2007.

[42] Prabhakar and J. N. Reddy, "Spectral/*hp* penalty least-squares finite element formulation for the steady incompressible Navier–Stokes equations," *Journal of Computational Physics*, vol. 215, pp. 274–297, 2006.

[43] W. Heinrichs, "Least-squares spectral collocation for discontinuous and singular perturbation problems," *Journal of Computational and Applied Mathematics*, vol. 157, no. 2, pp. 329 – 345, 2003.

[44] W. Heinrichs and T. Kattelans, "A direct solver for the least-squares spectral collocation system on rectangular elements for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 227, pp. 4776–4796, 2008.

[45] Prabhakar, J. P. Pontaza, and J. N. Reddy, "A collocation penalty least-squares finite element formulation for incompressible flows," *Comput. Methods Appl. Mech. Engrg.*, vol. 197, pp. 449–463, 2008.

[46] T. Kattelans and W. Heinrichs, "Conservation of mass and momentum of the least-squares spectral collocation scheme for the Stokes problem," *J. Comput. Phys.*, vol. 228, no. 13, pp. 4649–4664, 2009.

[47] C. L. Chang and J. J. Nelson, "Least-squares finite element method for the Stokes problem with zero residual of mass conservation," *SIAM Journal on Numerical Analysis*, vol. 34, no. 2, pp. 480–489, 1997.

[48] P. Bolton and R. W. Thatcher, "A least-squares finite element method for the Navier-Stokes equations," *Journal of Computational Physics*, vol. 213, pp. 174–183, 2006.

[49] J. J. Heys, E. Lee, T. A. Manteuffel, and S. F. McCormick, "On mass-conserving least-squares methods," *SIAM Journal on Scientific Computing*, vol. 28, no. 5, pp. 1675–1693, 2006.

[50] J. J. Heys, E. Lee, T. A. Manteuffel, and S. F. McCormick, "An alternative least-squares formulation of the Navier-Stokes equations with improved mass conservation," *Journal of Computational Physics*, vol. 226, pp. 994–1006, 2007.

[51] W. Coirier and K. Powell, "Solution-adaptive Cartesian cell approach for viscous and inviscid flows," *AIAA Journal*, vol. 34, no. 5, pp. 938–945, May 1996.

[52] M. Aftosmis, M. Berger, and J. Melton, “Robust and efficient Cartesian mesh generation for component-based geometry,” *AIAA Journal*, vol. 36, no. 6, pp. 952–960, Jun 1998.

[53] Z. J. Wang, “A quadtree-based adaptive cartesian/quad grid flow solver for Navier-Stokes equations,” *Computers & Fluids*, vol. 27, no. 4, pp. 529 – 549, 1998.

[54] H. S. Udaykumar, S. Krishnan, and S. V. Marella, “Adaptively refined, parallelised sharp interface cartesian grid method for three-dimensional moving boundary problems,” *International Journal of Computational Fluid Dynamics*, vol. 23, no. 1, pp. 1–24, 2009.

[55] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, “An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries,” *Journal of Computational Physics*, vol. 156, no. 2, pp. 209 – 240, 1999.

[56] R. Mittal and G. Iaccarino, “Immersed boundary methods,” *Annual Review of Fluid Mechanics*, vol. 37, no. 1, pp. 239–261, 2005.

[57] J. T. Oden, L. Demkowicz, W. Rachowicz, and T. A. Westermann, “Toward a universal h-p adaptive finite element strategy, part 2. a posteriori error estimation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 113 – 180, 1989.

[58] W. Rachowicz, J. T. Oden, and L. Demkowicz, “Toward a universal h-p adaptive finite element strategy part 3. design of h-p meshes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 181 – 212, 1989.

[59] R. Henderson, “Dynamic refinement algorithms for spectral element methods,” *Computer Methods in Applied Mechanics and Engineering*, vol. 175, no. 3-4, pp. 395 – 411, 1999.

[60] B.-N. Jiang and G. F. Carey, “Adaptive refinement for least-squares finite elements with element-by-element conjugate gradient solution,” *International Journal for Numerical Methods in Engineering*, vol. 24, no. 3, pp. 569–580, 1987.

[61] X.-X. Cai, B. Jiang, and G. Liao, “Adaptive grid generation based on the least-squares finite-element method,” *Computers and Mathematics with Applications*, vol. 48, pp. 1077–1085, 2004.

[62] J. P. Pontaza and J. N. Reddy, “Space-time coupled spectral/*hp* least-squares finite element formulation for the incompressible Navier-Stokes equations,” *Journal of Computational Physics*, vol. 197, pp. 418–459, 2004.

[63] C. T. Kelley, *Solving Nonlinear Equations with Newton’s Method*, ser. Fundamentals of Algorithms. SIAM, 2003.

[64] D. Reddy, B, *Introductory Functional Analysis with Applications to Boundary Value Problems and Finite Elements*, ser. Texts in Applied Mathematics. Springer-Verlag Newyork Inc., 1998.

[65] A. T. Patera, “A spectral element method for fluid dynamics: Laminar flow in a channel expansion,” *Journal of Computational Physics*, vol. 54, no. 3, pp. 468 – 488, 1984.

[66] M. G. Blyth and C. Pozrikidis, “A lobatto interpolation grid over the triangle,” *IMA J Appl. Math.*, vol. 71, no. 1, pp. 153–169, Feb. 2006.

[67] C. Pozrikidis, “A spectral collocation method with triangular boundary elements,” *Engineering Analysis with Boundary Elements*, vol. 30, no. 4, pp. 315 – 324, 2006.

[68] S. J. Sherwin and G. E. Karniadakis, “A new triangular and tetrahedral basis for high-order hp- finite element methods,” *International Journal for Numerical Methods in Engineering*, vol. 38, no. 22, pp. 3775–3802, 1995.

[69] B. Stroustrup, *The C++ Programming Language. Special Edition*. Addison-Wesley, 2000, no. ISBN 0-201-70073-5.

[70] W. J. Gordon and C. A. Hall, “Transfinite element methods: Blending-function interpolation over arbitrary curved element domains,” *Numerische Mathematik*, vol. 21, no. 2, pp. 109–129, Apr. 1973.

[71] R.-Y. Chang and C.-H. Hsu, “A variable-order spectral element method for incompressible viscous flow simulation,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. 17, pp. 2865–2887, 1996.

[72] C. Sert, “Nonconforming formulations with spectral element methods,” Ph.D. dissertation, Texas A&M University, August 2003.

[73] J.-P. Berrut and L. N. Trefethen, “Barycentric Lagrange interpolation,” *SIAM Rev*, vol. 46, pp. 501–517, 2004.

[74] C. Sert and A. Beskok, "Spectral element formulations on non-conforming grids: A comparative study of pointwise matching and integral projection methods," *Journal of Computational Physics*, vol. 211, no. 1, pp. 300 – 325, 2006.

[75] L. I. G. Kovasznay, "Laminar flow behind a two-dimensional grid," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 44, no. 01, pp. 58–62, 1948.

[76] O. Botella and R. Peyret, "Benchmark spectral results on the lid-driven cavity flow," *Computers & Fluids*, vol. 27, no. 4, pp. 421 – 433, 1998.

[77] C.-H. Bruneau and M. Saad, "The 2D lid-driven cavity problem revisited," *Computers & Fluids*, vol. 35, no. 3, pp. 326 – 348, 2006.

[78] E. Barragy and G. Carey, "Stream function-vorticity driven cavity solution using p finite elements," *Computers & Fluids*, vol. 26, no. 5, pp. 453–468, JUN 1997.

[79] M. Sahin and R. Owens, "A novel fully implicit finite volume method applied to the lid-driven cavity problem - part i: High Reynolds number flow calculations," *International Journal for Numerical Methods in Fluids*, vol. 42, no. 1, pp. 57–77, MAY 10 2003.

[80] R. D. Henderson, "Details of the drag curve near the onset of vortex shedding," *Physics of Fluids*, vol. 7, no. 9, pp. 2102–2104, 1995.

[81] S. J. Sherwin and G. E. Karniadakis, "A triangular spectral element method; applications to the incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 123, no. 1-4, pp. 189 – 229, 1995.

[82] B. Rajani, A. Kandasamy, and S. Majumdar, "Numerical simulation of laminar flow past a circular cylinder," *Applied Mathematical Modelling*, vol. 33, no. 3, pp. 1228 – 1247, 2009.

[83] C. H. K. Williamson, "Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers," *Journal of Fluid Mechanics*, vol. 206, no. -1, pp. 579–627, 1989.