INCOMPRESSIBLE FLOW SIMULATIONS USING LEAST SQUARES SPECTRAL ELEMENT METHOD ON ADAPTIVELY REFINED TRIANGULAR GRIDS

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

BY

OSMAN AKDAĞ

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

SEPTEMBER 2012

Approval of the thesis:

INCOMPRESSIBLE FLOW SIMULATIONS USING LEAST SQUARES SPECTRAL ELEMENT METHOD ON ADAPTIVELY REFINED TRIANGULAR GRIDS

submitted by **OSMAN AKDAĞ** 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

Asst. Prof. Dr. Merve Erdal _____

Mechanical Engineering Dept., METU

Dr. Tahsin Çetinkaya _____

Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Hakan Tarman _____

Engineering Sciences Dept., METU

Date: 12/09/2012

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: OSMAN AKDAĞ

Signature:

ABSTRACT

INCOMPRESSIBLE FLOW SIMULATIONS USING LEAST SQUARES SPECTRAL ELEMENT METHOD ON ADAPTIVELY REFINED TRIANGULAR GRIDS

Akdağ, Osman

M.S., Department of Mechanical Engineering

Supervisor: Asst. Prof. Dr. Cüneyt Sert

September 2012, 60 Pages

The main purpose of this study is to develop a flow solver that employs triangular grids to solve two-dimensional, viscous, laminar, steady, incompressible flows. The flow solver is based on Least Squares Spectral Element Method (LSSEM). It has p-type adaptive mesh refinement/coarsening capability and supports p-type nonconforming element interfaces. To validate the developed flow solver several benchmark problems are studied and successful results are obtained. The performances of two different triangular nodal distributions, namely Lobatto distribution and Fekete distribution, are compared in terms of accuracy and implementation complexity. Accuracies provided by triangular and quadrilateral grids of equal computational size are compared. Adaptive mesh refinement studies are conducted using three different error indicators, including a novel one based on elemental mass loss. Effect of modifying the least-squares functional by multiplying the continuity equation by a weight factor is investigated in regards to mass conservation.

Keywords: Least Squares Spectral Element Method, Triangular Lobatto Points, Triangular Fekete Points, Adaptive Mesh Refinement, Mass Conservation

Öz

UYUMLU SIKLAŞTIRILABİLEN ÜÇGEN AĞLAR ÜZERİNDE EN KÜÇÜK KARELER SPEKTRAL ELEMAN YÖNTEMİ KULLANILARAK SIKIŞTIRILAMAZ AKIŞ BENZETİMLERİ YAPILMASI

Akdağ, Osman

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Yar. Doç. Dr. Cüneyt Sert

September 2012, 60 Sayfa

Bu çalışmanın temel amacı, üçgen ağlar üzerinde iki boyutlu, viskoz, laminer, zamana bağlı olmayan, sıkıştırılamaz akışları çözen bir akış çözücüsü geliştirmektir. Bu akış çözücüsü En Küçük Kareler Spektral Eleman Yöntemine dayalıdır. Çözücü p-tipi ağ sıklaştırma/kabalaştırma yeteneğine sahiptir ve p-tipi ağ uyuşmazlığını desteklemektedir. Çözücüyü doğrulamak için çeşitli problemler çözülmüş ve başarılı sonuçlar elde edilmiştir. Lobatto dağılımı ve Fekete dağılımı olarak adlandırılan iki tip üçgen eleman düğüm noktası dağılımının doğruluk ve uygulama zorluğu bazında performans karşılaştırılması yapılmıştır. Ayrıca eşit sayıda bilinmeyene sahip üçgen ve dikdörtgen elemanların doğrulukları da karşılaştırılmıştır. Uyumlu ağ sıklaştırma çalışmaları üç farklı tip hata göstergesi kullanılarak yapılmıştır. Bu göstergelerden birisi ilk olarak bu çalışmada kullanılan, eleman üzerindeki kütle kaybına dayalı hata göstergesidir. Kütlenin korunumu denkleminin bir ağırlık katsayısı ile çarpılmasının metodun kütle korunumu performansına etkisi incelenmiştir.

Anahtar Kelimeler: En Küçük Kareler Spektral Eleman Yöntemi, Üçgende Lobatto Noktaları, Üçgende Fekete Noktaları, Uyumlu Ağ Sıklaştırma, Kütle Korunumu

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor Assist. Prof. Dr. Cüneyt Sert for his precious guidance, advice and support throughout this study. His willingness to motivate me contributed greatly to this research. During this study, I have learned a lot from him and I feel very lucky for having chance to work with him.

I am also thankful to my examining committee members Prof. Dr. M. Haluk Aksel, Assist. Prof. Dr. Merve Erdal, Dr. Tahsin Çetinkaya and Assoc. Prof. Dr. Hakan Tarman for their valuable suggestions and comments.

I want to express my appreciation to my friends and workmates Ali Karakuş, Alper Çelik and Eren Demircan for their technical and moral support all throughout this study.

I am thankful to Yusuf Yılmaz for his friendship, he is the *best man* ever.

I owe a special thank to Altuğ Özçelikkale for his technical support from thousands of miles away.

Especially, I owe my sincere gratitude to my family; Faika Akdağ, Mesut Akdağ, Pelin Çevik, Olcay Çevik and little member of our family Öykü Çevik. Thank you for all touches to my life.

I am grateful to my wife, Ayşe, whose everlasting love, support and encouragement kept me going on. It would be impossible to finish this work without her.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

x, y	Global Cartesian coordinates
ξ, η	Local standard element coordinates
u, v, p, ω	x-velocity, y-velocity, pressure, vorticity respectively
ψ	Nodal expansion bases on triangular region
φ	Modal expansion bases on triangular region
P_{kl}	Proriol polynomials
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{F}$	Force vector
$\boldsymbol{L}$	Differential operator
$\boldsymbol{V}$	Vandermonde matrix
$\boldsymbol{J}$	Jacobian of the coordinate transformation
ε_{ls}	Error estimate based on least squares functional
ε_{sp}	Error estimate based on spectral coefficients
ε_{mass}	Error estimate based on elemental mass deficit

CHAPTER 1

INTRODUCTION

1.1. Background and related work

Computational Fluid Dynamics (CFD) makes use of numerical methods to solve governing partial differential equations of fluid flow and heat transfer problems. Advances in computational resources in terms of both hardware and software such as increase in processor speeds and storage capacities, development of alternative parallelization tools and new modeling and solution techniques make CFD a faster, more accurate and more robust tool for the simulation of thermofluidic transport problems. Today it is widely used in various engineering disciplines as a primary design and analysis tool.

CFD solvers are based on a number of different methods, such as Finite Difference Method (FDM), Finite Volume Method (FVM), Finite and Spectral Element Methods (FEM/SEM), Boundary Element Method (BEM), Meshless Methods, Lagrangian description based particle methods, etc. Among these, FDM, FVM and FEM are the most commonly used ones. In FDM, unknowns of a flow problem such as velocity or pressure are calculated on discrete points of a flow field by approximating the differential operators of the governing equations using finite difference formulae [1]. The main advantage of FDM is its simplicity in terms of mathematical background and computer implementation, which is lost for complex problem domains that typically require unstructured grids for space discretization. Therefore FDM studies are typically restricted to simple geometries such as channel flows.

Among all others, FVM is the most commonly used technique for developing flow solvers. Most of today's frequently used commercial CFD software is based on FVM, in which the domain is divided into small regions called *cells* or *finite volumes*. Governing differential equations are integrated over each cell, which basically requires the calculation of fluxes that pass through cell boundaries [2].

By the use of simple cell shapes such as triangles and quadrilaterals in 2D and hexahedrons and tetrahedrons in 3D, FVM can be applied to complex geometries easily. Other advantages of it are its local conservation property and simple, physics-based formulation that can easily be understood and implemented by engineers.

FEM and SEM are similar to FVM in terms of discretizing the geometry into small cells, known as *elements* in FEM. Actually it is possible to show that under certain conditions FEM and FVM yield identical discretizations. However, their mathematical foundations and formulation details are quite different. In FEM unknowns of the problem are approximated over each element using *interpolating polynomials* (*basis functions*) and algebraic equations are obtained by using these approximations in weighted integral forms of the governing equations. FEM and SEM are quite similar in formulation and application. Their main difference is that, in FEM, relatively small sized elements with low order Lagrange type interpolating polynomials based on equi-spaced nodes are used [3], while in SEM, relatively large sized and high order elements with Legendre or Chebyshev polynomials are preferred [4].

SEM is the technique used in the current work. It can be seen as a combination of FEM and Spectral Method (SM). FEM uses local expansion bases over each element and ends up with a sparse linear algebraic system of equations for which there are many well-known efficient solution methods. On the other hand, SM uses global basis functions, which results in a fully populated equation system to be solved [5]. Also SM is limited to very simple rectangular or prism shaped geometries while FEM is very flexible in handling complex geometries. The most important advantage of SM is that the accuracy of the solution is higher than FEM for the same number of degrees of freedom. That is to say SM requires less storage to achieve the same accuracy as FEM. SEM utilizes the most powerful features of FEM and SM; the domain is divided into elements just as in FEM, and high order expansions are used as in SM to achieve fast spectral convergence.

The current study is the continuation of a previous research conducted by Ozcelikkale and Sert [6,7], in which a 2D incompressible flow solver based on Least Squares Spectral Element Method (LSSEM) was developed. LSSEM is one of the several alternative variational formulations used with SEM. Actually the Galerkin formulation is the most commonly used one, especially for structural

mechanics problems. Unfortunately, the success of Galerkin formulation for structural mechanics cannot be achieved in convection dominated flow problems. When the Galerkin formulation is used for highly convective flows, unphysical oscillations are observed, which can only be avoided by excessive mesh refinement [8]. Another disadvantage of the Galerkin method is that when it is used to solve Navier-Stokes equations, basis functions of different orders must be used for velocity and pressure approximation to satisfy the Ladyzhenskaya-Babuska-Brezzi (LBB) or the so called inf-sup condition [9,10]. Researchers tried to overcome these deficiencies by using modified versions of Galerkin formulation such as Streamline Upwind Petrov-Galerkin [11], Galerkin Least Squares [12] and Taylor-Galerkin [13], which brings in debatable concepts of upwinding and artificial dissipation. Also Galerkin formulation applied to Navier-Stokes equations results in a non-positive definite coefficient matrix which is hard to solve [8].

Not suffering from the disadvantages mentioned above, the Least Squares (LS) formulation is a better alternative to Galerkin formulation for fluid mechanics problems [8]. It allows the use of base functions of the same order for all unknowns, without violating the LBB condition. Also it results in a symmetric and positive definite coefficient matrix, for which there are efficient solvers available such as the preconditioned conjugate gradient method. Furthermore, it does not require special formulations or treatments for different flow regimes, i.e. the same formulation can be used for all Mach numbers covering both incompressible and compressible flows.

LS formulation has two main drawbacks [14]. When least squares FEM and SEM formulations are directly applied to second order differential equations, basis functions should be C^1 continuous, i.e. both the functions and their first derivatives should be continuous over the problem domain. However it is difficult to establish C^1 continuity especially for 3D problems and it creates ill-conditioned coefficient matrices [8,15]. To overcome this difficulty, Lynn and Arya offered converting high order differential equations to a set of first order ones by introducing new variables [16]. The first successful LS finite element formulation based on this idea was applied to elasto-static problems by Zienkiewicz et al. [17]. Applying LS formulation to first order differential equations enables the use of C^0 continuous interpolations, which are continuous but not continuously differentiable. This allows the use of piecewise continuous functions, just like in the case of classical Galerkin FEM. Converting the second order derivatives into

first order ones requires the definition of new variables, which increases the computational size of the problem. But on the other hand these new variables are chosen to be of physical interest, such as vorticity for flow problems, which eliminates error prone post processing calculations. Depending on the choice of this extra variable, LS formulations are named as velocity-pressure-vorticity (*U-p-ω*) formulation, velocity-pressure-stress (*U-p-σ*) formulation, etc. Among the others, *U-p-ω* formulation is the most commonly used one in incompressible flow analyses [18,19]. In the present work this formulation is employed and its details will be given in Chapter 2.

The second drawback of the LS formulation is poor mass conservation especially for incompressible, internal flow problems with sudden contraction regions [14,20,21]. The reason for poor mass conservation is the minimization of the LS functional of the continuity equation together with all other governing equations, but not using the continuity equation as a constraint as it is used in the Galerkin formulation [21]. In the literature several cures are suggested to improve poor mass conservation drawback of LS formulation. Chang and Nelson used continuity equation as a constraint and it is enforced to be satisfied at all points of the discretization [20]. The main disadvantage of this method is that the resulting coefficient matrix is not positive definite. Pontaza tried to improve the weak velocity-pressure coupling with a regularized form of the continuity constraint [22,23] Pontaza also introduced a splitting scheme to decouple velocity and pressure computation and achieved good mass conservation properties [24]. Penalty LS formulation is also used to enhance mass conservation by replacing the continuity equation by the pressure with the help of a penalty parameter [25]. These remedies are not used in the present study since they all have negative side effects such as destroying the symmetric and positive definite property of the coefficient matrix and making the formulation and programming complex. However, one last remedy known as *weighted LSFEM* [14,26], which is not difficult to apply on a *U-p-ω* LSSEM formulation, will be visited in the following chapters. This idea tries to improve the mass conservation by simply increasing the significance of the continuity equation among the others by multiplying it with a weighting factor.

Another possibility to improve mass conservation of LS formulation in an efficient manner is to use high order elements instead of special treatments discussed in the previous paragraph. It has been shown that using high order elements provides a considerable improvement in mass conservation performance

[7,15,21,27]. The current study also makes use of this approach by implementing LS formulation in a spectral element framework.

Spectral element formulations for 2D fluid mechanics problems were first applied on quadrilateral elements [28]. In SEM, using rectangular elements has some advantages such as high convergence rate, high accuracy and easy construction of multi-dimensional expansion bases using tensor product of 1D polynomials. Almost all LSSEM base flow solvers existing in the literature are also limited to rectangular elements [29-31], including the one the current study is based on. However, it is not easy to discretize complex 2D geometries by the use of quadrilateral elements only, which brings in the necessity of using triangular elements. One of the main objectives of the current study is to add triangular element support to the code developed by Özçelikkale and Sert [7].

FEM and SEM formulations on triangular elements may employ two types of expansion bases, *modal (hierarchical)* or *nodal*. When modal expansion bases are used, there are no physical grid points (nodes) on the elements. A p^{th} order element makes use of polynomials of all orders from one to p and the expansion set of an element of order p contains all polynomials of an element of order p-1. This is why the modal expansion is also called as hierarchical expansion. Using modal expansion bases may be advantageous when adaptive mesh refinement techniques are used [32]. Sherwin and Karniadakis used triangular modal expansions based on Dubiner polynomials [33] in their spectral element formulation [34,35]. On the other hand in a nodal expansion, all interpolating polynomials of an element have the same degree. An expansion polynomial of an element has the value 1 at its corresponding node and it is equal to 0 at all other nodes. This is known as the *Kronecker-delta* property of nodal expansion and it provides simplicity in implementation. Nodal triangular expansions are widely used in high order FEM and SEM studies [22,36-41], which is also the choice in the present study.

When nodal quadrilateral elements are used in SEM, interior nodal points of a 2D or 3D element are simply calculated as the tensor product of 1D nodal points, which are commonly chosen as Gauss-Lobatto-Legendre (GLL) quadrature points. The expansion basis of a quadrilateral element is also calculated by the tensor product of the polynomials passing through 1D points. However, in triangular elements, finding positions of nodal points is not that straightforward. Because of the geometry of the triangle, a simple tensor product of a 1D

expansion is not possible. Therefore, several researches have been conducted to find the optimal nodal distribution over a triangle. One way of placing the nodes on a triangle is to calculate the nodes on a quadrilateral element and then mapping them into a triangular region as done in [36-38] But this nodal distribution in triangle leads to coefficient matrices of high condition numbers and low interpolation quality.

There are several nodal distributions specifically optimized for triangular elements. One of those widely used triangular interpolation grid is composed of *electrostatic points*, introduced by Hesthaven [42]. Calculation of the position of the electrostatic points is based on minimization of electrostatic potential energy. This interpolation grid is used by Warburton et. al. to solve incompressible Navier-Stokes equations with a pseudo-spectral scheme [39]. Another commonly used triangular nodal set is *Fekete points*, which are calculated by maximizing the determinant of generalized Vandermonde matrix [43], [44], [45]. Coordinates of Fekete nodal points are not given by an explicit equation, but rather they are calculated and tabulated up to order 19 by Taylor et. al. [45]. They are widely used in FEM/SEM codes since they provide a high interpolation quality even for very high orders [23,39,41]. More recently, another triangular nodal distribution known as *Lobatto grid*, which is based on geometric issues, is proposed by Blyth and Pozrikidis [46]. The interpolation quality of Lobatto grid is competitive with those of electrostatic and Fekete points, while creating the Lobatto grid is much simpler [47,48]. The common property of these three different triangular grids is that the nodal distribution on the edges of the triangle is overlapping with 1D GLL points. Thus, the triangles and quadrilaterals can be used together without any nonconformity. Fekete and Lobatto interpolation grids are used and their performances are compared in the present study. The related details are given in Chapter 3.

One of the main features of the code developed in the current study is solution based adaptive mesh refinement (AMR). AMR strategies can be classified in three types; *h-refinement*, *p-refinement* and *r-refinement*. In h-refinement technique, elements of high gradient regions are divided into smaller ones and the total number of elements is increased. In p-refinement, the number of elements is kept constant but the degrees of the desired elements are increased; therefore, the number of nodes increases. In r-refinement, however, neither the number of elements nor the number of nodes is changed. The r-refinement is done by moving the nodes to the areas where a finer mesh is required. These three

techniques can be used separately or they can be combined in a hybrid AMR strategy.

In literature, studies that employ AMR with LS formulation are limited. Jiang and Carey introduced an error estimator based on elemental residual and used it for adaptive h-refinement in solution of Laplace equations with LSFEM formulation [49]. Taghaddosi et. al. utilized adaptive r-refinement in the solution of compresibble Euler equations with LSFEM [50]. They used an error indicator based on the second derivative of the flow variables. Heinrichs applied adaptive h-refinement on triangular grids to solve Poisson and incompressible Navier-Stokes equations with LSSEM formulation [41]. A simple error measure based on the solution gradients is used in this study. Galvao et. al. used a combined refinement concept, *hp-refinement*, with an error indicator based on least squares functional [51]. They solved hyperbolic partial differential equations using LSSEM formulation. In the present study, adaptive p-refinement is performed using different error indicators. The first error indicator is based on least squares functional, which is proposed by Jiang and Carey [49]. The second one is based on spectral coefficients, proposed by Henderson [52]. The last error measure used in the present study is based on the elemental mass deficit. Ozcelikkale and Sert mentioned about the potential use of this error indicator but did not implement it [7]. In the present work it is implemented and will be discussed in Chapter 4.

1.2. Thesis Outline

In Chapter 1 the background and related works are introduced. The details about least squares formulation are given in Chapter 2. In chapter 3, the triangular nodal distributions and the expansion bases are presented. In Chapter 4, the details of implementation of least squares spectral element method are given. Chapter 5 presents the numerical results obtained by developed flow solver. In the last chapter, Chapter 6, the results and findings of the study are summarized.

CHAPTER 2

LEAST SQUARES FORMULATION

In this chapter, the details of the least-squares (LS) formulation are given. For clarity, the formulation is first presented for a first order ordinary differential equation. Then it is extended to the governing equations of 2D, steady, incompressible flow.

2.1. LS formulation for a first order ordinary differential equation

A linear, first order model ODE is selected as follows

$$a\frac{du}{dx} + cu = f \tag{2.1}$$

where, *u* is the dependent variable (unknown), and *a*, *c* and *f* are known functions of *x*. For simplicity, a differential operator, *L*, can be defined as

$$L = a\frac{d}{dx} + c \tag{2.2}$$

which can be used to write the ODE in the following form

$$L\,u = f \tag{2.3}$$

Similar to the classical Galerkin FEM, in LSFEM the unknown *u* is approximated over the elements of the computational mesh by using expansion bases (shape functions), ψ, details of which will be given in Chapter 3. The approximation is done as follows

$$u_h^e = \sum_{j=1}^{N} c_j^e \psi_j \tag{2.4}$$

where, u_h^e is the approximated unknown over an arbitrary element *e*, *N* is the number of nodes of this element at which discrete unknown values are stored and c_j^e's are the coefficients to be calculated. Since the expansion bases have the Kronecker-delta property in nodal FEM/SEM, the coefficient, c_j^e, is the same as the approximated value of the unknown, u_j^e, at the j^{th} node of element *e*. Therefore, equation (2.4) can also be written as

$$u_h^e = \sum_{j=1}^{N} u_j^e \psi_j \tag{2.5}$$

To calculate the nodal unknowns, u_j^e, the differential equation needs to be converted into an algebraic equation over each element, steps of which are explained next.

First, the approximation of the unknown given by Eqn. (2.5) is substituted into the differential equation and its residual, *R(x)*, is obtained as

$$R(x) = a\frac{du_h^e}{dx} + cu_h^e - f \tag{2.6}$$

which can also be written as follows

$$R(x) = a\sum_{j=1}^{N} u_j^e \frac{d\psi_j}{dx} + c\sum_{j=1}^{N} u_j^e \psi_j - f \tag{2.7}$$

This residual is then used into the following *weighted residual statement* written for element *e*

$$0 = \int_{\Omega_e} w(x)R(x)d\Omega_e \tag{2.8}$$

where $w(x)$ are the weight functions, that need to be selected. For an element with *N* nodes , *N* many algebraic equations are needed to solve for *N* unknowns. Therefore, *N* many weight functions need to be selected. Different FEM/SEM formulations such as Galerkin, Least-Squares (LS), etc. differ in selecting these weight functions. In LS formulation, i^{th} weight function is selected to be the derivative of the residual with respect to the i^{th} nodal unknown of element *e*, as given below

$$w_i = \frac{\partial R}{\partial u_i^e} \tag{2.9}$$

which takes the following form for the selected model differential equation

$$w_i = \frac{\partial R}{\partial u_i^e} = a\frac{d\psi_i}{dx} + c\psi_i \tag{2.10}$$

Using the notation introduced in Eqn. (2.2) i^{th} weight function becomes

$$w_i = L\,\psi_i \tag{2.11}$$

Now, the i^{th} algebraic equation over element e can be formed by substituting the i^{th} weight function into the weighted residual statement given by Eqn. (2.8) to get

$$0 = \int_{\Omega_e} \left(a\frac{d\psi_i}{dx} + c\psi_i\right)\left(a\sum_{j=1}^{N} u_j^e\frac{d\psi_j}{dx} + c\sum_{j=1}^{N} u_j^e\psi_j - f\right)d\Omega_e \tag{2.12}$$

$$\sum_{j=1}^{N}\left[\int_{\Omega_e}\left(a\frac{d\psi_i}{dx} + c\psi_i\right)\left(a\frac{d\psi_j}{dx} + c\psi_j\right)dx\right]u_j^e = \int_{\Omega_e} f\left(a\frac{d\psi_i}{dx} + c\psi_i\right)d\Omega_e \tag{2.13}$$

Using the differential operator, L, Eqn. (2.13) can be put into the following compact form

$$\sum_{j=1}^{N}\left[\int_{\Omega_e}(L\psi_i)(L\psi_j)dx\right]u_j^e = \int_{\Omega_e} f(L\psi_i)d\Omega_e \tag{2.14}$$

which can further be simplified into the following matrix form

$$[K^e]\{u^e\} = \{F^e\} \tag{2.15}$$

where $[K^e]$ and $\{F^e\}$ are the elemental stiffness matrix and elemental force vector, respectively, given as follows

$$K_{ij}^e = \int_{\Omega_e}(L\psi_i)(L\psi_j)d\Omega_e \tag{2.16a}$$

$$F_i^e = \int_{\Omega_e} f(L\psi_i)d\Omega_e \tag{2.16b}$$

The integrand in the definition of the elemental stiffness matrix is

$$(L\psi_i)\big(L\psi_j\big) = \begin{bmatrix} (L\psi_1)(L\psi_1) & \cdots & (L\psi_1)(L\psi_N) \\ \vdots & \ddots & \vdots \\ (L\psi_N)(L\psi_1) & \cdots & (L\psi_N)(L\psi_N) \end{bmatrix} \tag{2.17}$$

which can also be written as

$$(L\psi_i)\big(L\psi_j\big) = (L\psi)^T(L\psi) \tag{2.18}$$

where, ψ is the following row vector of shape functions

$$\psi = \{\psi_1\ ,\ \psi_2\ , \cdots , \psi_N\} \tag{2.19}$$

To finish the formulation, elemental stiffness matrix and force vector can be expressed in their following final forms

$$K_{ij}^e = \int_{\Omega_e} (L\psi)^T(L\psi) d\Omega_e \tag{2.20a}$$

$$F_i^e = \int_{\Omega_e} f(L\psi) d\Omega_e \tag{2.20b}$$

These integrals need to be evaluated separately for each element of the computational mesh and the obtained elemental equation systems should be assembled into a global one, which can then be solved to obtain the discrete nodal unknowns.

2.2. LS formulation for 2D, steady, incompressible flow

The governing equations of 2D, steady, incompressible flow of a Newtonian fluid are the following conservation of mass and linear momentum

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{2.21a}$$

$$\boldsymbol{u} \cdot \boldsymbol{\nabla u} + \frac{1}{\rho}\boldsymbol{\nabla} p - \nu \boldsymbol{\nabla^2 u} = \boldsymbol{f} \tag{2.21b}$$

where, $\boldsymbol{u}$ is the velocity vector and p is the pressure, which are the primary unknowns of the problem. $\boldsymbol{f}$ is the body force vector, ρ is the known constant

density and ν is the constant kinematic viscosity. In CFD studies these two equations together are called Navier-Stokes (N-S) equations.

N-S equations are second order and nonlinear and therefore the formulation given in Section 2.1 needs some modifications. As mentioned in Chapter 1, when using LS formulation, second order differential equations should be replaced with an equivalent first order set. This modification is done so that C^0 continuous expansion bases, which are easy to construct, can be used. For this purpose the second order term $\boldsymbol{\nabla^2 u}$ of Eqn. (2.21b) is replaced with the curl of the vorticity vector, $\boldsymbol{\nabla \times \omega}$, and the definition of vorticity is added as an extra equation to Eqn. (2.21). This formulation is called as velocity-pressure-vorticity (*U-p-ω*) formulation. The final form of the differential equation set is given below

$$\boldsymbol{\nabla \cdot u} = 0 \tag{2.22a}$$

$$\boldsymbol{u \cdot \nabla u} + \frac{1}{\rho}\boldsymbol{\nabla} p + \nu \boldsymbol{\nabla \times \omega} = \boldsymbol{f} \tag{2.22b}$$

$$\boldsymbol{\omega - \nabla \times u} = 0 \tag{2.22c}$$

The first term in Eqn. (2.22b) (the convective term) is nonlinear and needs to be linearized, which can be done using the following iterative Newton's linearization technique [8]

$$\boldsymbol{u}^{k+1} \cdot \boldsymbol{\nabla u}^{k+1} \cong \boldsymbol{u}^{k} \cdot \boldsymbol{\nabla u}^{k+1} + \boldsymbol{u}^{k+1} \cdot \boldsymbol{\nabla u}^{k} - \boldsymbol{u}^{k} \cdot \boldsymbol{\nabla u}^{k} \tag{2.23}$$

where *k+1* denotes the current iteration and *k* denotes the previous one. At convergence of this iterative procedure $\boldsymbol{u}^{k+1}$ will approach to $\boldsymbol{u}^{k}$ and the right and left hand sides of Eqn. (2.23) will be equal. Using Eqn. (2.23) in (2.22b), linearized form of the momentum conservation can be obtained as follows

$$\boldsymbol{u}^{0} \cdot \boldsymbol{\nabla u} + \boldsymbol{u} \cdot \boldsymbol{\nabla u}^{0} + \frac{1}{\rho}\boldsymbol{\nabla} p + \nu \boldsymbol{\nabla \times \omega} = \boldsymbol{f} + \boldsymbol{u}^{0} \cdot \boldsymbol{\nabla u}^{0} \tag{2.24}$$

where, for simplicity, no superscript is used for the velocity of the current iteration and *zero* ($\boldsymbol{u}^0$) is used for that of the previous iteration.

Linearization iterations are stopped at a certain point where the current and previous velocities are close enough. The error measure used for this purpose is L^2 norm of the velocity vector, $\|\boldsymbol{V}_i\|_2$, where, $\boldsymbol{V}_i = \sqrt{u_i^2 + v_i^2}$, and u_i and v_i are the

nodal velocities in *x* and *y* directions, respectively. So the convergence criteria for nonlinear iterations can be defined as

$$\frac{\left\|\boldsymbol{V}_i - \boldsymbol{V}_i^0\right\|_2}{\left\|\boldsymbol{V}_i\right\|_2} < \varepsilon_{nonlinear} \quad , \quad 1 < i < \text{Number of mesh nodes} \tag{2.25}$$

where $\varepsilon_{nonlinear}$ is a user selected tolerance value.

After introducing the new variable, vorticity, and linearizing the N-S equations, now there are four scalar equations and four scalar unknowns for a 2D problem. For a flow in the *xy* plane, scalar unknowns are the *x* and *y* components of the velocity vector, pressure and *z* component of the vorticity vector (u, v, p, ω). Four scalar equations in their open form are given as follows

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.26a}$$

$$u_0\frac{\partial u}{\partial x} + v_0\frac{\partial u}{\partial y} + u\frac{\partial u_0}{\partial x} + v\frac{\partial u_0}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial \omega}{\partial y} = f_x + u_0\frac{\partial u_0}{\partial x} + v_0\frac{\partial u_0}{\partial y} \tag{2.26b}$$

$$u_0\frac{\partial v}{\partial x} + v_0\frac{\partial v}{\partial y} + u\frac{\partial v_0}{\partial x} + v\frac{\partial v_0}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\frac{\partial \omega}{\partial x} = f_y + u_0\frac{\partial v_0}{\partial x} + v_0\frac{\partial v_0}{\partial y} \tag{2.26c}$$

$$\omega + \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0 \tag{2.26d}$$

After this point, the formulation is similar to the one presented in Section 2.1. The differential operator, *L*, can now be defined as a 4x4 matrix

$$L = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 & 0 \\ u_0\frac{\partial}{\partial x} + v_0\frac{\partial}{\partial y} + \frac{\partial u_0}{\partial x} & \frac{\partial u_0}{\partial y} & \frac{1}{\rho}\frac{\partial}{\partial x} & \nu\frac{\partial}{\partial y} \\ \frac{\partial v_0}{\partial x} & u_0\frac{\partial}{\partial x} + v_0\frac{\partial}{\partial y} + \frac{\partial v_0}{\partial y} & \frac{1}{\rho}\frac{\partial}{\partial y} & \nu\frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & -\frac{\partial}{\partial x} & 0 & 1 \end{bmatrix} \tag{2.27}$$

which enables us to express the differential equation set as follows

$$L\{U\} = \{F\} \tag{2.28}$$

where the unknown vector $\{U\}$ and the right hand side vector $\{F\}$ are given as

$$U = (u, v, p, \omega)^T \tag{2.29}$$

$$F = \begin{Bmatrix} 0 \\ u_0 \dfrac{\partial u_0}{\partial x} + v_0 \dfrac{\partial u_0}{\partial y} \\ u_0 \dfrac{\partial v_0}{\partial x} + v_0 \dfrac{\partial v_0}{\partial y} \\ 0 \end{Bmatrix} \tag{2.30}$$

An element has 4*N* many unknowns, where *N* is the number of nodes of the element. The approximation of the elemental unknowns is done as

$$U_h^e = \begin{Bmatrix} u_h^e \\ v_h^e \\ p_h^e \\ \omega_h^e \end{Bmatrix} = [\psi]\{U^e\} \tag{2.31}$$

where the details of $[\psi]$ and $\{U^e\}$ are given below

$$\psi = \left[\begin{array}{cccc|cccc|c|cccc} \psi_1 & 0 & 0 & 0 & \psi_2 & 0 & 0 & 0 & \cdots & \psi_N & 0 & 0 & 0 \\ 0 & \psi_1 & 0 & 0 & 0 & \psi_2 & 0 & 0 & & 0 & \psi_N & 0 & 0 \\ 0 & 0 & \psi_1 & 0 & 0 & 0 & \psi_2 & 0 & & 0 & 0 & \psi_N & 0 \\ 0 & 0 & 0 & \psi_1 & 0 & 0 & 0 & \psi_2 & & 0 & 0 & 0 & \psi_N \end{array}\right] \tag{2.32}$$

$$U^e = (\{u_1^e, v_1^e, p_1^e, \omega_1^e\}, \{u_2^e, v_2^e, p_2^e, \omega_2^e\}, \cdots, \{u_N^e, v_N^e, p_N^e, \omega_N^e\})^T \tag{2.33}$$

Finally the elemental stiffness matrix and force vector can be obtained using Eqn. (2.20), but this time the differential operator defined in Eqn. (2.27) and the elemental shape function matrix defined in Eqn. (2.32) are used. One important missing detail is the selection of shape functions, which will be explained in the next chapter.

As mentioned before, weighted LSSEM formulation is also used in the present study. In weighted LSSEM, the continuity equation in the system defined by Eqn. (2.26) is modified as;

$$W\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0 \tag{2.34}$$

where W is the weight factor. The differential operator, *L*, is also modified accordingly. The effect of this modification is discussed in Chapter 5.

CHAPTER 3

NODAL POINTS AND EXPANSIONS IN A TRIANGULAR ELEMENT

In the formulation presented in Chapter 2, the details about the shape functions, which are the interpolating polynomials used in the approximation of unknowns, are not given. As mentioned in Chapter 1, there are several alternatives of triangular interpolation grids such as electrostatic points, Fekete points and Lobatto points. Computing the electrostatic points and Fekete points requires a considerable amount of work [42,45]. On the other hand the Lobatto grid is easy to construct while it is providing a comparable interpolation quality. In the present work Fekete points calculated by Taylor et. al. [45] and Lobatto points proposed by Blyth and Pozrikidis [46] are used. In this chapter, calculation details of the nodal point coordinates and the shape functions are explained.

3.1. Master element concept

In implementing FEM/SEM, for the sake of simplicity and generality, elemental calculations are performed on a standard element of fixed shape and size, called the *master element*. Figure 3.1 shows the 2D triangular master element in $\xi\eta$ coordinates defined over the region $-1 \le \xi \le 1$ and $-1 \le \eta \le -\xi$.

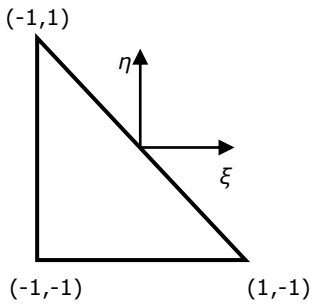


Figure 3.1 Triangular master element

3.2. Nodal points of Lobatto triangular grid

For 1D interpolation, nodal points of an element can be selected as zeros of *Lobatto polynomial*, *Lo*. Lobatto polynomial is defined as the first derivative of *Legendre polynomial*, *L*, which is a type of *Jacobi polynomials*, $P_n^{\alpha,\beta}$. The Lobatto and Legendre polynomials of order *n* are defined as

$$Lo_n(x) = \frac{d}{dx} L_n(x) \tag{3.1}$$

$$L_n(x) = P_n^{0,0}(x) \tag{3.2}$$

In the grid shown in Figure 3.2a, the nodal points on the edges of the triangle are chosen as zeros of Lobatto polynomial, t_i, and the interior nodes are placed so that $(\xi_i, \eta_j) = (t_i, t_j)$. The *three-fold symmetry* (symmetry with respect to the edges of the triangle) of a nodal distribution is important due to its better interpolation property. To test the three-fold symmetry, the nodal points on the master element can be transformed to an arbitrary equilateral triangle. The nodal points of the Lobatto grid shown in Figure 3.2a are transformed to the equilateral triangle shown in Figure 3.2b. For this particular master element and equilateral triangle, the following mapping functions are used for the transformation

$$\xi_e = \frac{\xi + 1}{2} + \frac{\eta + 1}{4} \tag{3.3a}$$

$$\eta_e = \frac{\sqrt{3}}{4}(\eta + 1) \tag{3.3b}$$

As can be seen in the figure, the node distribution does not satisfy the three-fold symmetry.

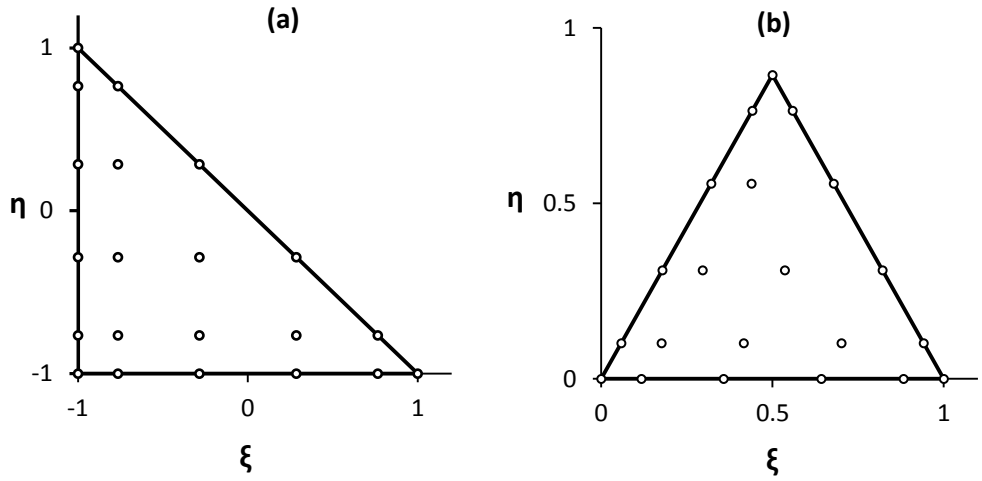


Figure 3.2 (a) Triangular Lobatto grid on master element for p=5. (b) Triangular Lobatto grid on equilateral triangle for p=5.

Blyth and Pozrikidis [46] worked on the Lobatto nodal distribution to improve its three-fold symmetry. In the Lobatto grid they proposed, the nodal points on the edges are again zeros of the Lobatto polynomials, but the positions of interior nodes for an element of order p are modified as follows

$$\xi_i = \frac{1}{3}\left(2 + 2t_i - t_j - t_k\right) - 1 \tag{3.4a}$$

$$\eta_j = \frac{1}{3}\left(2 + 2t_j - t_i - t_k\right) - 1 \tag{3.4b}$$

where, i=1,2,...,p+1, j=1,2,....,p+2-i and k=p+3-i-j. The three-fold symmetry is achieved by this modification. An example of the nodal set and its three-fold symmetry for an element of order p=5 is presented in Figure 3.3.

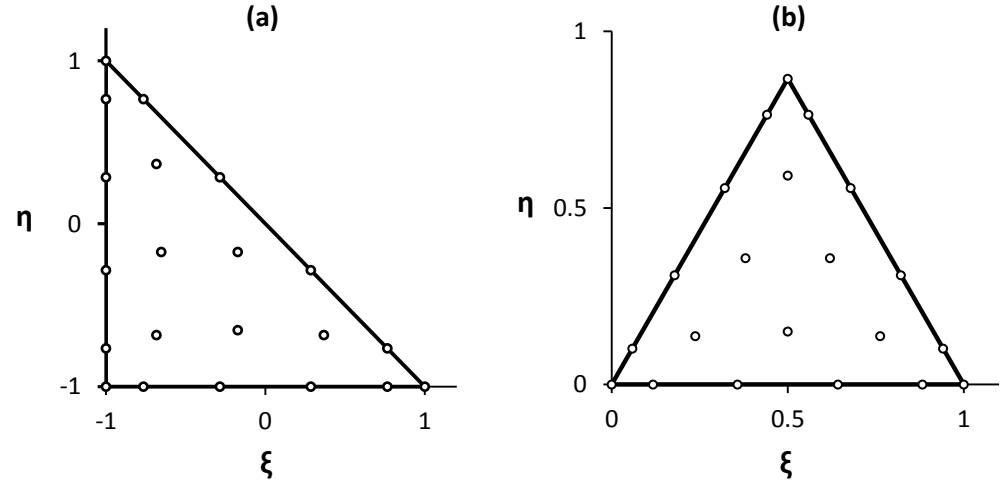


Figure 3.3 (a) Modified triangular Lobatto grid on master element for p=5. (b) The same grid transferred on an equilateral triangle.

3.3. Nodal points of Fekete triangular grid

The interpolation quality of a nodal set can be measured by calculating the *Lebesgue constant*. Using a nodal distribution which minimizes the Lebesgue constant in the triangular region is reasonable since it has high interpolation quality. Computing the Fekete points is based on minimizing the Lebesgue constant by maximizing the determinant of *generalized Vandermonde matrix*, the definition of which can be found in Section 3.4. There is not a closed form to calculate the coordinates of the Fekete points; special techniques must be used to compute them.

Fekete points on triangle were first evaluated by Bos [43] who computed the coordinates of the nodal set up to order *p=3*. Then Chen and Babuška[44] extended his work and computed points up to order *p=13*. More recently Taylor et. al. [45] proposed a new algorithm and computed the Fekete points up to order *p=19*. The Fekete points computed in their study has the minimum Lebesgue constant.

In the present work the nodal set computed by Taylor et. al. [45] is used.

3.4. Triangular expansion base

In this section, the procedure to calculate the shape functions is explained. Note that the same procedure is applied for both of the nodal sets.

In a triangle of order *p* there are *N* many nodal points and *N* many expansion polynomials, where

$$N = \frac{1}{2}(p+1)(p+2) \tag{3.5}$$

Since the coordinates of the nodal points are now known, shape (basis) functions forming the expansion base of the element can be calculated. Unfortunately the shape functions do not have closed form representations, but rather they are calculated using the orthogonal polynomials, $\varphi_j(\xi,\eta)$. The i^{th} shape function can be calculated as

$$\psi_i(\xi,\eta) = \sum_{j=1}^{N} c_j^i \, \varphi_j(\xi,\eta) \tag{3.6}$$

where, c_j^i are the unknown coefficients which are to be calculated using the Kronecker-delta property of the expansion base, which simply states that a shape function of an element takes the value 1 at its corresponding node and becomes 0 at all other *N*-1 nodes. To calculate the coefficients of Eqn. (3.6), the *generalized Vandermonde matrix,* $\boldsymbol{V}$, should be introduced. The Vandermonde matrix holds the values of polynomials, φ_j, evaluated at nodal points. It can be defined as

$$V_{ij} = \varphi_j((\xi,\eta)_i) \tag{3.7}$$

where $(\xi,\eta)_i$ are the coordinates of the i^{th} node. In our case, the functions which are to be evaluated at the nodes are the orthogonal polynomials, $\varphi_j(\xi,\eta)$. When the Vandermonde matrix is multiplied with the coefficients c_j^i, the result will be a *unit vector*, e_i, whose elements are all zero except the i^{th} one.

$$\boldsymbol{V} \cdot \boldsymbol{c}^i = \boldsymbol{e}_i \tag{3.8}$$

Indeed, the i^{th} element of e_i is calculated by applying the summation in Eqn. (3.6) with the orthogonal polynomials. Since the value of the i^{th} shape function, $\psi_i(\xi,\eta)$, at the i^{th} node is 1, i^{th} element of e_i is unity.

To be able to solve the system in Eqn. (3.8), the Vandermonde matrix must be well-conditioned. The condition number of the Vandermonde matrix depends on polynomials, $\varphi_j(\xi,\eta)$, which should be chosen as orthogonal or near-orthogonal. Alternative choices are possible, such as Proriol and Appell polynomials. In the present study, Proriol polynomials are employed since they create the desired well-conditioned Vandermonde matrix [46].

To define the Proriol polynomials on a triangular region, the triangular master element ($-1 \le \xi \le 1$ and $-1 \le \eta \le -\xi$) is mapped to the quadrilateral master element ($-1 \le \xi' \le 1$ and $-1 \le \eta' \le 1$) as shown in Figure 3.4. This transformation is done with the following mapping functions

$$\xi = \frac{1+\xi'}{1-\eta'} - 1 \tag{3.9a}$$

$$\eta = \eta' \tag{3.9b}$$

And the inverse mapping functions are given by

$$\xi' = \frac{(1 + \xi)(1 - \eta)}{2} - 1 \tag{3.10a}$$

$$\eta' = \eta \tag{3.10b}$$

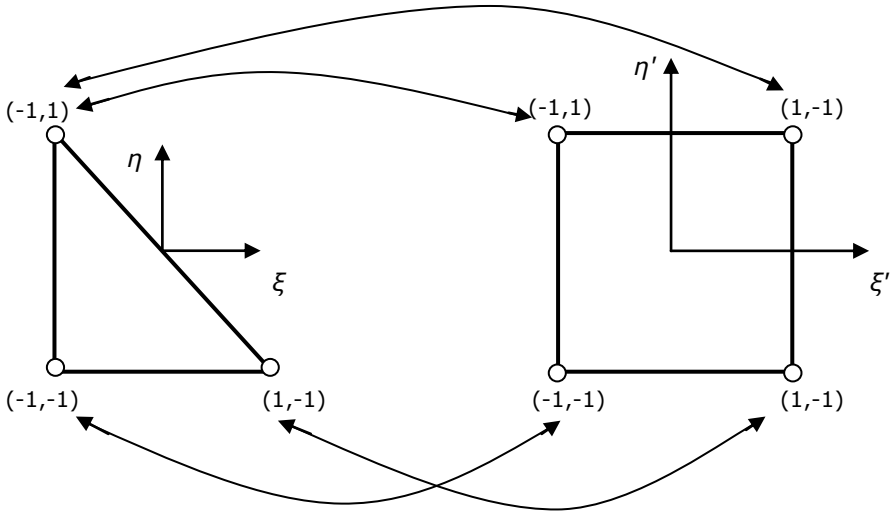


Figure 3.4 Mapping between triangular and quadrilateral master elements

With this coordinate transformation, Proriol polynomials can be defined as

$$P_{kl} = L_k(\xi')\left(\frac{1 - \eta'}{2}\right)^k J_l^{(2k+1,0)}(\eta') \tag{3.11}$$

where, the pair (*k,l*) is defined so that $0 \leq k+l \leq p$. L_k is the k^{th} order Legendre polynomial and $J_l^{(2k+1,0)}$ is the Jacobi polynomial of order *l*.

So, the orthogonal polynomial, φ_j, introduced in Eqn. (3.6) is defined as

$$\varphi_j = P_{kl} \tag{3.12}$$

where, *j* is associated with any unique pair (*k,l*).

3.5. Sample shape function calculation

To make the calculation of the shape functions clear, the whole process is explained for a low order (p=2) triangular element with 6 nodes. Lobatto nodal set is used for this purpose but note that after computing the coordinates of the nodal points, the procedure is the same for both of the nodal distributions. The second order element is shown in Figure 3.5 with two different node numberings. In (a), the node numbering is shown as (*i,j*) pairs and in (b), each node has a single unique number, which is chosen arbitrarily.

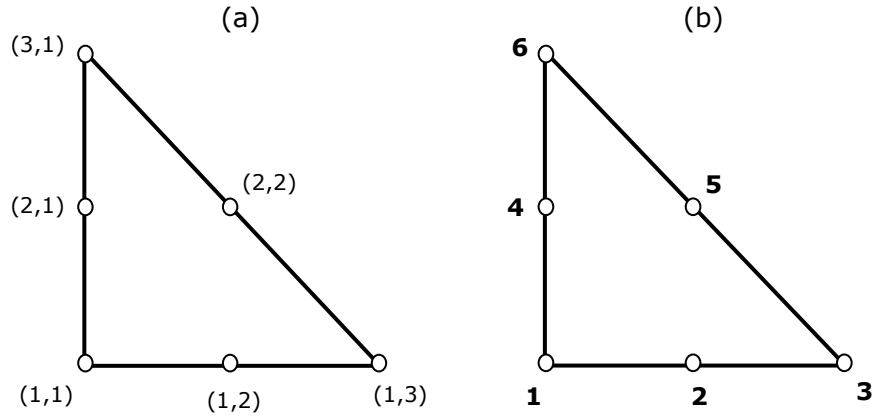


Figure 3.5 Node labeling in two different ways

To obtain the shape functions, first, the nodal points must be calculated. Based on the triangular master element of Figure 3.1, the three Lobatto points on the edges of the 2nd order element are located at $t_i = \{-1, 0, 1\}$. Substituting t_i into Eqn. (3.4), the following nodal coordinates for all 6 nodes can be calculated.

Table 3.1 Nodal coordinates for a 2nd order element

Node Number *i*	$(\xi, \eta)_i$
1	(-1,-1)
2	(0,-1)
3	(1,-1)
4	(-1,0)
5	(0,0)
6	(-1,1)

Now, define the Proriol polynomials, P_{kl}, for which the (k,l) pair is defined as $0 \leq k+l \leq 2$

$$
\begin{aligned}
P_{00} &= 1 \\
P_{01} &= \frac{1}{2}(2\xi + \eta + 1) \\
P_{02} &= \frac{1}{4}(6\xi^2 + 6\xi\eta + \eta^2 + 6\xi + 4\eta + 1) \\
P_{10} &= \frac{1}{2}(3\eta + 1) \\
P_{11} &= \frac{1}{4}(2\xi + \eta + 1)(5\eta + 3) \\
P_{20} &= \frac{1}{2}(5\eta^2 + 2\eta - 1)
\end{aligned}
\tag{3.13}
$$

These Proriol polynomials will be associated with the orthogonal polynomials, φ_j, in an arbitrary order. In this case the Proriol polynomials can be chosen in the order they are given in Eqn. (3.13)

$$\varphi_j = \{P_{00}, P_{10}, P_{20}, P_{01}, P_{11}, P_{02}\} \tag{3.14}$$

Now it is time to construct the Vandermonde matrix, V_{ij}, which holds the value of the j^{th} orthogonal polynomial, φ_j, evaluated at the i^{th} node.

$$
V_{ij} = \begin{bmatrix}
1 & -1 & 1 & -1 & 1 & 1 \\
1 & 0 & -0.5 & -1 & 0 & 1 \\
1 & 1 & 1 & -1 & -1 & 1 \\
1 & -0.5 & 0.25 & 0.5 & -0.75 & -0.5 \\
1 & 0.5 & 0.25 & 0.5 & 0.75 & -0.5 \\
1 & 0 & 0 & 2 & 0 & 3
\end{bmatrix}
\tag{3.15}
$$

To find the first shape function, ψ_1, the system in Eqn. (3.8) must be solved.

$$\boldsymbol{V} \cdot \boldsymbol{c}^1 = \boldsymbol{e}_1 \tag{3.16}$$

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 0 & -0.5 & -1 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -0.5 & 0.25 & 0.5 & -0.75 & -0.5 \\ 1 & 0.5 & 0.25 & 0.5 & 0.75 & -0.5 \\ 1 & 0 & 0 & 2 & 0 & 3 \end{bmatrix} \begin{Bmatrix} c_1^1 \\ c_2^1 \\ c_3^1 \\ c_4^1 \\ c_5^1 \\ c_6^1 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.17}$$

$$\boldsymbol{c}^1 = \begin{Bmatrix} 0 \\ -0.3 \\ 0.333 \\ -0.1 \\ 0.2 \\ 0.067 \end{Bmatrix} \tag{3.18}$$

Substituting c_j^1 and φ_j into Eqn. (3.6), the first shape function can be found as

$$\psi_1 = \frac{1}{2}((\xi + \eta)^2 + \xi + \eta) \tag{3.19}$$

Calculation of the other 5 shape functions follows the same procedure, except a different unit vector is used in Eqn. (3.16) for each shape function. As a final note, the calculated first shape function satisfies the Kronecker-delta property, i.e. it has a value of 1 at node 1 of Table 3.1 and it is equal to zero at the other 5 nodes of the 2nd order element.

CHAPTER 4

IMPLEMENTATION DETAILS

As mentioned before this study enhances the LSSEM flow solver previously developed by Özçelikkale [6]. Within the scope of this enhancement, the solver is now compatible with triangular meshes and uses an additional novel error indicator, which is based on elemental mass deficit, for adaptive mesh refinement. In this chapter, the details of the present LSSEM implementation are presented.

The solver is written in C++ programming language. It solves 2D, laminar, steady, incompressible Navier-Stokes equations on both quadrilateral and triangular meshes. It can perform *p-type* adaptive mesh refinement and supports a number of different error measures. It enables the use of *p-nonconformities* on neighboring element faces.

To solve a problem with the LSSEM flow solver, the domain must be discretized first. In this study, mesh generation is done by using the freely available, open source mesh generator Mesh2d [53].After the mesh is created the solution is performed through the following steps;

i. Elemental systems (elemental stiffness matrices and force vectors) are calculated.
ii. Elemental systems are assembled into a global system and the global system is solved for nodal unknowns.
iii. Errors are estimated based on the current solution.
iv. If the errors are in the allowable limits, step (v) is skipped.
v. Modifications on the grid are done according to the errors (adaptive mesh refinement is performed) and the previous steps are repeated with the new mesh.
vi. The solution is finalized by generating a post processing file.

The details of these steps are given in the following sections.

4.1. Calculations on triangular elements and solution of global system

As stated before, elemental operations are done on a master element, which is introduced in Section 3.1. For this purpose, actual elements of a mesh need to be transformed into the master element, as illustrated in Figure 4.1.

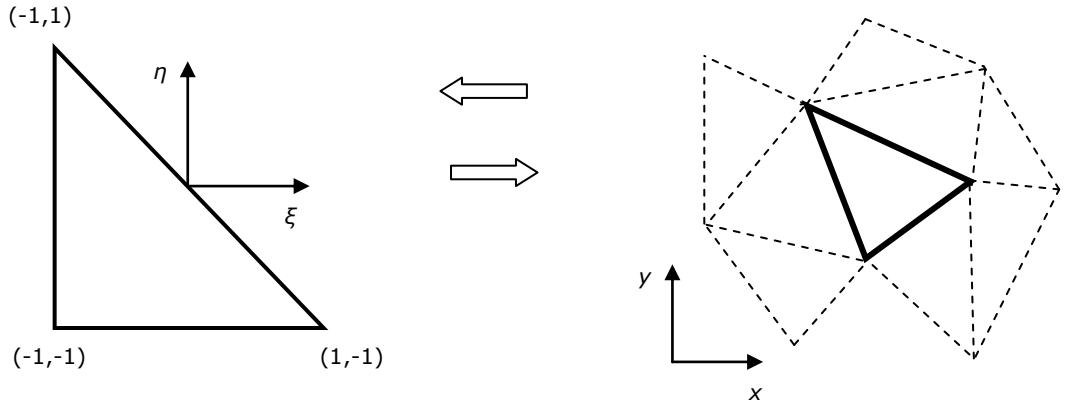


Figure 4.1 Triangular master element transformation

The transformation is done by using the following mapping

$$x = x_0 \frac{-\xi - \eta}{2} + x_1 \frac{\xi + 1}{2} + x_2 \frac{\eta + 1}{2} \tag{4.1a}$$

$$y = y_0 \frac{-\xi - \eta}{2} + y_1 \frac{\xi + 1}{2} + y_2 \frac{\eta + 1}{2} \tag{4.1b}$$

where x_0, x_1, x_2 and y_0, y_1, y_2 are the x and y coordinates of the three corners of the actual triangular element. To calculate the elemental stiffness matrix and the force vector, integrals given in Eqn. (2.16) must be evaluated. Integration over an arbitrary element is transformed to the master element as follows

$$\int_{\Omega^e} f(x,y)\, dxdy = \int_{\Omega_m^e} f(\xi,\eta)\, |\boldsymbol{J}|\, d\xi d\eta \tag{4.2}$$

where, Ω^e and Ω_m^e are the triangular regions of the arbitrary element and the master element, respectively, $|\boldsymbol{J}|$ is the determinant of Jacobian matrix, defined as follows

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{4.3}$$

Integrals over the master element are evaluated using *Gauss quadrature* integration. In Gauss quadrature method, the definite integral is expressed as a finite weighted sum of integrand values at special quadrature points. In one dimension, the definite integral of $f(\xi)$ can be defined as

$$\int_{-1}^{1} f(\xi)dx = \sum_{i=1}^{n} w_i f(\xi_i) \tag{4.4}$$

where, w_i are the quadrature weights, ξ_i are the quadrature points in the interval [-1,1], and n is the number of quadrature points. In practice, to evaluate an integral with limits [*a,b*], a coordinate transformation is required to change the limits of the integral as [-1,1].

There are several alternatives for the quadrature point set, ξ_i. One possible choice is the zeros of Legendre polynomial in interval [-1,1]. When these points are used, the quadrature technique is called as *Gauss-Legendre* quadrature, which can be classified in three types depending on the selection of the quadrature points. The first type is *Gauss-Legendre* quadrature, which uses interior points but excludes the end points (-1,1). The second type is *Gauss-Radau-Legendre* quadrature in which the interior points and only one of the end points, usually $\xi = -1$, are used. In *Gauss-Lobatto-Legendre* type quadrature, which is the last type, the interior points and both of the end points are included to the set of quadrature points. In FEM/SEM applications, Gauss-Lobatto-Legendre (GLL) quadrature is the most commonly used one, since imposing the boundary conditions is easier when the end points are included.

In nodal, 2D SEM applications, when quadrilateral elements are used, the quadrature points can be the same as the nodal points, which are GLL points. This is an advantage of using quadrilateral elements since it decreases the complexity of computation. Also the quadrature points and weights can be calculated using the tensor product of 1D points and weights. However, when triangular elements are used, the quadrature points and weights cannot be expressed in terms of 1D points and weights. Therefore, to perform integration with Gauss quadrature over a triangular region, the triangle must be

transformed into a quadrilateral. This transformation can be done as explained in Section 3.4.

As this transformation is done, the top vertex of the triangle is transformed into the $\eta' = 1$ edge of the quadrilateral as seen in Figure 3.4. This top vertex of the triangle and the top edge of the quadrilateral are called as degenerate vertex and degenerate line, respectively. If the integrals are evaluated by using GLL quadrature rule, the integrands of both stiffness matrix and force vector integrals must be evaluated at this degenerate line of the quadrilateral. The values of the shape functions at this line are known since the shape functions have the Kronecker-delta property. However, evaluating the derivatives of the shape functions at this line is problematic because when the derivative of the shape function is transformed to the quadrilateral there will be terms which include $1/(1-\eta')$ arising from the transformation given by Eqn. (3.10a). This term is undefined on the $\eta' = 1$ line (degenerate line). Karniadakis and Sherwin [54] offers not evaluating the shape function derivatives at this degenerate line by using GLL quadrature in ξ' direction and Gauss-Radau-Legendre quadrature in η' direction, which excludes the end point $\eta' = 1$.

Indeed, all the shape functions and their derivatives have the term $(1-\eta')^k$ (see Eqn. (3.12)). When the shape function derivatives are calculated explicitly, the term $(1-\eta')^k$ is cancelled out with the term, $1/(1-\eta')$, which arises from the transformation. Therefore, GLL quadrature rule can be applied in both directions if analytical forms of shape function derivatives are constructed using the orthogonal polynomials, which is the technique followed in the current study. The process for constructing the derivatives of shape functions is the same as explained in Section 3.4. The only difference is that, after the unknown coefficients, c_j^i, are calculated, the following formula is used to form the derivatives

$$\frac{\partial \psi_i(\xi,\eta)}{\partial \xi} = \sum_{j=1}^{N} c_j^i \, \frac{\partial \varphi_j(\xi,\eta)}{\partial \xi} \tag{4.5}$$

η derivatives can also be calculated in the same manner.

In the present work the order of the element and the GLL quadrature order are the same. The integration is not evaluated exactly but the integration error is not significant since it has the same order of magnitude with the interpolation error [34].

After the elemental operations are accomplished, the elemental systems are assembled into a global linear system. Boundary conditions must applied before solving for the unknowns. In least squares formulation, the governing differential equations involve only first order derivatives of the unknowns (see Eqn 2.22). Therefore, only essential boundary conditions are used. The developed LSSEM flow solver uses the boundary conditions shown in Table 4.1, which are in compliance with the ones presented in [6].

Table 4.1 Boundary conditions (adapted from [6])

Boundary type	***Variables at the boundary***
Wall (no slip boundary condition)	$u=0$, $v=0$
Velocity (can be used in inflow boundary)	$u=UD$*, $v=UD$
Outflow in x direction	$v=0$, $p=UD$
Outflow in y direction	$u=0$, $p=UD$
Symmetry with respect to x axis	$v=0$, $\omega=0$
Symmetry with respect to y axis	$u=0$, $\omega=0$

* *UD* stands for user defined.

After applying boundary conditions, the global linear system is solved for the unknowns. In the present work, *static condensation* is used to decompose the large elemental system into two smaller size systems. After static condensation is applied, the global unknowns are solved by the iterative element by element (EBE) *Jacobi preconditioned conjugate gradient method*. The details of static condensation and EBE Jacobi preconditioned conjugate gradient method are given by Özçelikkale [6].

4.2. Adaptive mesh refinement, error indicators and nonconformities

The developed LSSEM flow solver employs adaptive p-type mesh refinement and coarsening based on the posteriori error indicators. To do this elemental errors are kept in between the user defined lower and upper error bounds.

There are three types of error indicators that the solver uses. The first one is based on least squares functional. This error measure is proposed by Jiang and Carey [49] and a modified version of it is used by Özçelikkale and Sert [7]. The error indicator used in [7] is defined as

$$\varepsilon_{ls} = \frac{\sqrt{I^e}}{\|\boldsymbol{U}^e\|_{H^1}} \tag{4.6}$$

where, $\|\boldsymbol{U}^e\|_{H^1}$ is the H^1 norm of the unknown vector and I^e is the elemental least squares functional which is defined as

$$I^e = \boldsymbol{U}^{eT}\boldsymbol{K}^e\boldsymbol{U}^e - 2\boldsymbol{U}^{eT}\boldsymbol{F}^e + \int_{\Omega_e} \boldsymbol{f}^T\boldsymbol{f}\, d\Omega_e \tag{4.7}$$

Once the elemental system and the approximate solution is calculated, the elemental least squares functional can be evaluated. The error based on the least squares functional is a measure of how close the approximate solution is to the exact solution of the governing equations.

The other error measure used in the present work is based on the spectral coefficients, which is introduced by Henderson [52]. The error calculation is based on the idea that the calculated nodal approximate solution can also be represented as a modal expansion

$$u_h^e = \boldsymbol{u}^{eT}\boldsymbol{\psi} = \boldsymbol{a}^T\boldsymbol{\varphi} \tag{4.8}$$

where u_h^e the approximate velocity in *x* direction, $\boldsymbol{\psi}$ is the nodal expansion base vector, $\boldsymbol{\varphi}$ is modal expansion base vector, $\boldsymbol{u}^e$ is the nodal *x*-velocity vector and $\boldsymbol{a}$ is the coefficient vector of the modal expansion. An error can be calculated using the coefficients of approximate solution's modal expansion. When quadrilateral elements are employed, Legendre polynomials are used as the modal expansion basis, but on triangular elements Proriol polynomials are used to represent the approximate solution. Using the generalized Vandermonde matrix, $\boldsymbol{V}$, which holds the values of Proriol polynomials evaluated at the nodal points, the unknowns at the nodal points can be calculated as

$$\boldsymbol{u}^e = \boldsymbol{V}\boldsymbol{a} \tag{4.9}$$

So, the coefficient vector $\boldsymbol{a}$ can be found as

$$\boldsymbol{a} = \boldsymbol{V}^{-1}\boldsymbol{u}^e \tag{4.10}$$

Then, the error estimate is defined as

$$\varepsilon_{sp}^{u} = \frac{\hat{a}_p}{\|u_h^e\|_{H^1}} \tag{4.11}$$

where $\hat{a}_p$ is a constant found by lumping the coefficients a_i. The lumping is done by summation of the coefficients of the highest order modal polynomials, which includes the necessary information for error estimation. As explained in Section 3.4, each φ_j is associated with a Proriol polynomial P_{kl}. For lumping, the coefficients, a_j, of the modal expansion bases, $\varphi_j = P_{kl}$, for which $k+l=p$, are summed.

ε_{sp}^{u} in Eqn. (4.11) is the spectral error of x-velocity, u. The spectral error for the other unknowns, v, p, ω, can also be calculated with the same procedure. The spectral error measures how well one of the unknowns is interpolated.

The last error indicator that the LSSEM solver uses is based on the elemental mass deficit. According to the observations that the mass conservation of least squares formulation is poor and it can be improved by increasing the order of elements, it is reasonable to use the elemental mass deficit as an error measure and increase the order of elements that suffer from high mass deficit. For this purpose the mass flux across the faces of the elements are calculated as

$$\dot{m} = \int_{\Gamma^e} \rho \boldsymbol{V} d\boldsymbol{\Gamma}^{\boldsymbol{e}} \tag{4.12}$$

where ρ is the constant density of the fluid, $\boldsymbol{V}$ is velocity vector and $\boldsymbol{\Gamma}^{\boldsymbol{e}}$ represents the boundary of the element including the unit outward normal information. This integration can be performed with 1D GLL quadrature rule since the nodal points on the edges of the triangle coincide with 1D GLL points and the unknowns are readily available at these nodal points. Once the mass flux over each face of the element is known, the mass deficit of the element can be calculated.

Flow chart of the used adaptive refinement strategy is given in Figure 4.2. When the order of an element is increased or decreased, the nodes of the faces of this element and its neighbor elements do not necessarily coincide and the interelement C^0 continuity may be lost. A nonconforming face between 2^{nd} and 3^{rd} order elements is shown in Figure 4.3. This nonconformity, known as p-type nonconformity, can be handled in different ways. In this study, Constrained Approximation Method is used, details of which can be found in [6,7].

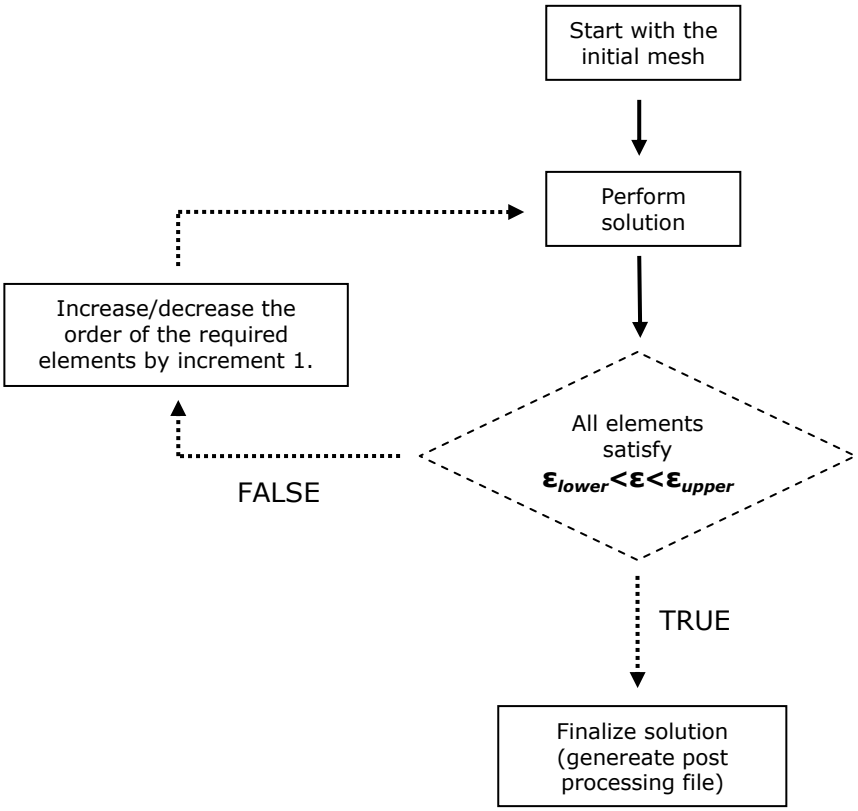


Figure 4.2 Flow chart of the adaptive refinement strategy (ε_{lower} and ε_{upper} are lower and upper error bounds respectively and ε is the elemental error estimate)

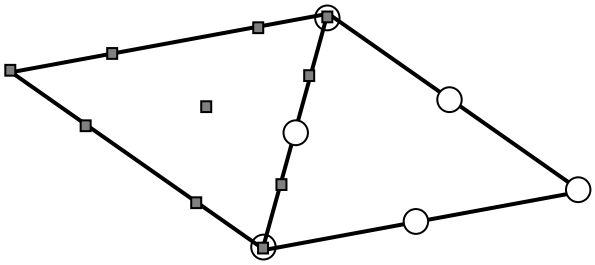


Figure 4.3 A nonconforming face between 2^{nd} and 3^{rd} order elements

CHAPTER 5

NUMERICAL RESULTS

In this chapter, the results for several benchmark problems are presented in order to validate the developed solver. The performances of two different triangular grids, namely Fekete and Lobatto grids, are compared. Results obtained by using triangular and quadrilateral meshes of the same computational size are also compared. Adaptive mesh refinement studies are conducted using three different error indicators, one of which is a novel one based on the elemental mass deficit. Also the effect of multiplying the continuity equation with a weight factor (*weighted LSSEM*) on the mass conservation is investigated.

Note that the governing equations solved in this study are in dimensional form and the results presented in this chapter are dimensional. *x* and *y* coordinates have the unit *m* and the velocities are in *m/s* for all figures presented in this chapter.

5.1. Kovasznay Flow

For code verification purposes, a problem with a known analytical solution can be solved. In the presented study this is done by studying the steady, 2D, laminar flow presented by Kovasznay [55].

The exact solution of Kovasznay flow problem is given by

$$u = 1 - e^{\lambda x}\cos(2\pi y) \tag{5.1a}$$

$$v = \frac{\lambda}{2\pi}e^{\lambda x}\sin(2\pi y) \tag{5.1b}$$

$$p = p_0 + \frac{1}{2}\left(1 - e^{2\lambda x}\right) \tag{5.1c}$$

$$\omega = \left(\frac{\lambda^2}{2\pi} - 2\pi\right) e^{\lambda x} \sin(2\pi y) \tag{5.1d}$$

where, $\lambda = {}^{1}/_{2\nu} - \left[{}^{1}/_{4\nu^2} + 4\pi^2\right]^{1/2}$, ν is the kinematic viscosity and p_0 is a constant reference pressure. The solution is done on the domain of [-0.5,1] x [-0.5,1.5] for the Reynolds number, $Re = 40$. The streamlines of the exact solution for this case is given in Figure 5.1, which resembles the flow behind the wake of two obstacles. The obstacles can be thought to be placed somewhere at the left side of the domain shown in Figure 5.1. These two obstacles are placed at y_1=0 and y_2=1 (the spacing between the obstacles is, M=1) for the case shown in Figure 5.1.

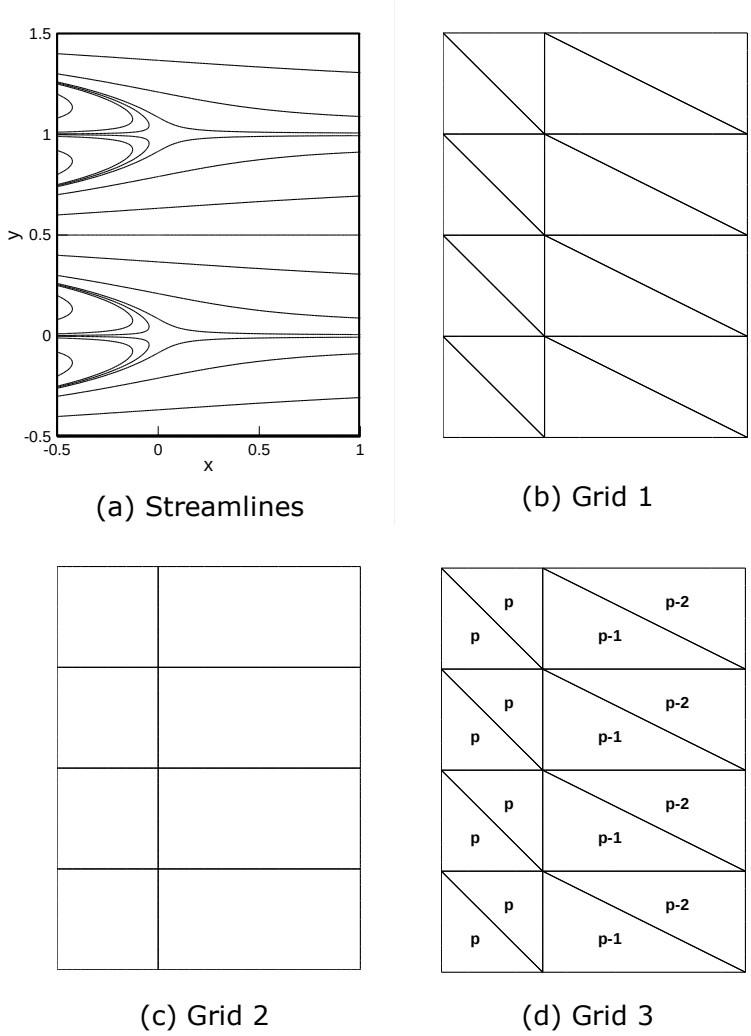
The Reynolds number for this flow is defined as;

$$Re = \frac{u_0 M}{\nu} \tag{5.2}$$

where u_0 is the average x-velocity over the domain and M is the array spacing. For the solution given by Eqn. 3.1, $u_0 = 1$ and $M = 1$.

The boundary conditions of the problem are specified using the exact solutions given by Eqn. 3.1. The x-velocity and y-velocity are specified at the boundaries of the domain and the pressure is set as p_0, which is the constant reference pressure seen in Eqn. 5.1c, at one of the nodes placed at x=0.

The problem is solved on different grids and convergence rates are compared. For each grid, the orders of elements are increased systematically to observe the convergence characteristic of the solver. Grids used in the solution are presented in Figure 5.1. Solutions are performed by using both Fekete and Lobatto triangles on conforming Grid 1. Conforming quadrilateral elements are employed in Grid 2. The last grid, Grid 3, is a nonconforming one and uses Lobatto triangles.



(a) Streamlines (b) Grid 1

(c) Grid 2 (d) Grid 3

Figure 5.1 Streamlines of Kovasznay flow and the grids used in the solution

The first solution is performed using Lobatto triangles on Grid 1 for different expansion orders and the maximum norm errors, $\|U_{exact} - U_h\|_\infty$, of all approximate unknowns, $U_h = \{u_h, v_h, p_h, \omega_h\}$, are presented in Figure 5.2a. The exponential decay is observed for all errors. Also maximum norm errors of the x-velocity, $\|u_{exact} - u_h\|_\infty$, for Fekete triangle, Lobatto triangle and quadrilateral elements are compared in Figure 5.2b. The errors are all in the same order of magnitude and the rates of convergence for these element types are similar. In Figure 5.2c, the convergence rates of a conforming grid (Grid 1) and a nonconforming grid (Grid 3) are compared. The comparison is done using the maximum norm error of the x-velocity. The error is plotted against $\sqrt{NN}$, where NN is the total number of nodes in the mesh. The exponential decay of the maximum error is also observed for nonconforming triangular mesh. Lastly, the elemental relative true error of x-velocity in H_1 norm, e_{rel}^u,

$$e_{rel}^{u} = \frac{\|u_{exact} - u_h\|_{H_1}}{\|u_{exact}\|_{H_1}} \tag{5.3}$$

is compared with the elemental mass deficit, e_{mass} . This comparison is done to see whether an error indicator based on the elemental mass deficit can be used for adaptive mesh refinement or not. As seen in Figure 5.2d, both the elemental relative true error and the elemental mass deficit have similar trends as the order of the elements are increased. Therefore, it can be concluded that the elemental mass deficit is a promising error measure for adaptive refinement.

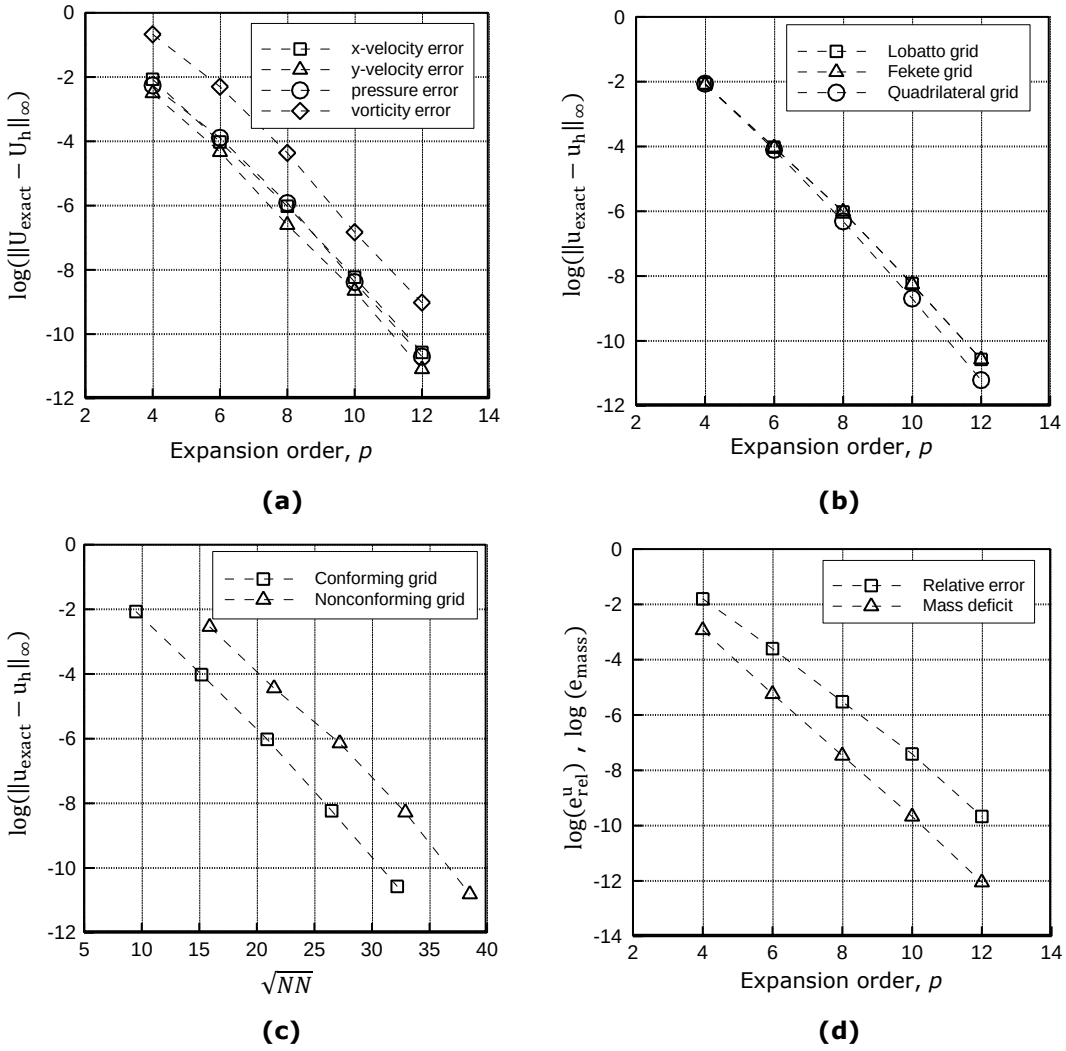


Figure 5.2 Decay of the errors in solution of Kovasznay Flow with different types of grids

5.2. Flow around a square cylinder (FASC)

As mentioned before, least squares finite and spectral element formulations suffer from poor mass conservation, especially for internal flow problems with sudden contraction regions. The problem of flow around an obstacle placed in a two dimensional channel is widely solved to test the mass conservation of these formulations. In the present study, steady, laminar flow around a square cylinder which is fixed in a two dimensional channel is solved. The purpose of solving this problem is to compare Lobatto and Fekete triangular grids, to investigate the effects of multiplying the mass conservation equation with a weight factor on the solution and to perform adaptive p-refinement study by using three different error indicators.

The problem domain is shown in Figure 5.3. The fluid enters the channel with a parabolic velocity profile. At the outlet boundary y-velocity and pressure are set as zero. No slip boundary condition is employed at the walls of the channel and walls of the square obstacle.

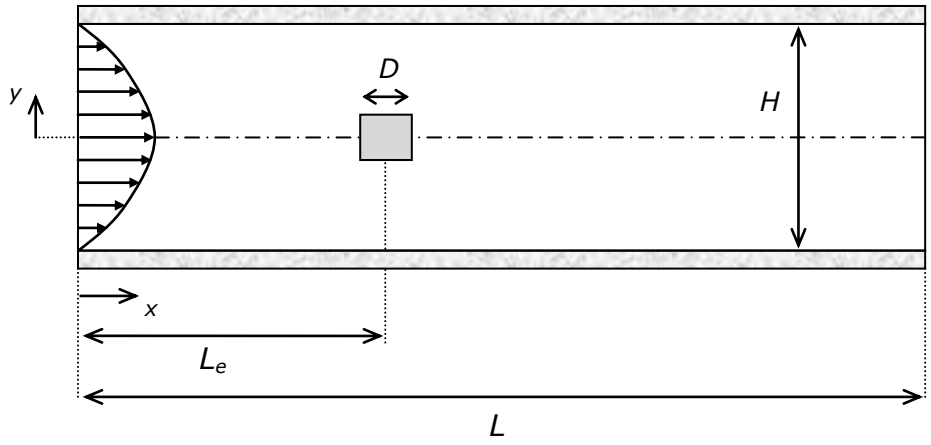


Figure 5.3 Definition of the domain for flow around a square cylinder

For the channel shown in Figure 5.3, blockage ratio, *B*, and Reynolds number, *Re*, can be defined as

$$B = \frac{D}{H} \tag{5.4a}$$

$$Re = \frac{u_{max} D}{\nu} \tag{5.4b}$$

where u_{max} is the maximum velocity at the inlet and ν is the kinematic viscosity.

For all solutions performed in this study, the blockage ratio is fixed as $B = 1/8$ and the entrance length is fixed as $L_e = L/4$.

First, a set of solutions with different Reynolds numbers are performed to compare the results with the ones available in literature. Breuer et. al. [56] used different numerical methods, namely lattice-Boltzmann and finite volume, to solve the FASC problem for different Reynolds numbers. They also worked with the blockage ratio and the entrance length stated above and obtained the following relation

$$\frac{L_r}{D} = -0.065 + 0.0554Re \quad for\ 5 < Re < 40 \tag{5.5}$$

where L_r is the length of the recirculation region that forms behind the square cylinder.

In the present work, the solutions are performed for *Re*=10 and *Re*=20 with the mesh shown in Figure 5.4. Conforming Lobatto triangles of orders *p*=12 and *p*=15 are used for *Re*=10 and *Re*=20 solutions, respectively.

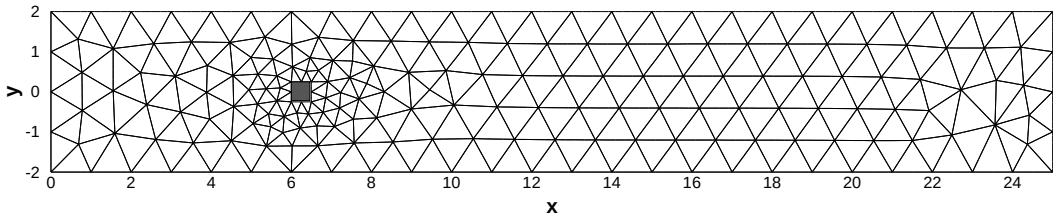


Figure 5.4 334 element mesh used for the first solutions with *Re* = 10 and *Re* = 20

The streamlines around the square can be seen in Figure 5.5. The recirculation length ratios, L_r/D, of 0.48 and 0.95 agree with the results in reference [56]. Also the mass flow rate at the exit section is calculated for both cases. The flow rate ratios, $\dot{m}_{exit}/\dot{m}_{in}$, are found as 0.94 and 0.93 for *Re*=10 and *Re*=20, respectively.

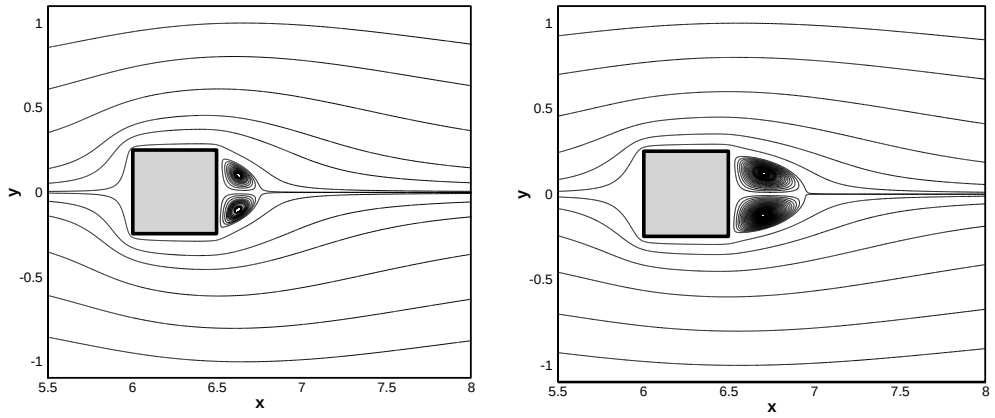


Figure 5.5 Streamlines around the square for **(a)** *Re*=10 and **(b)** *Re*=20 obtained with 334 element mesh and *p*=12 and *p*=15, respectively.

Another set of solutions is performed on the mesh shown in Figure 5.6 to compare the performances of Lobatto and Fekete triangles and to investigate the effect of multiplying the continuity equation with a weighting factor. This coarser mesh is chosen to illustrate the huge difference in mass conservation when the weight factors are used.

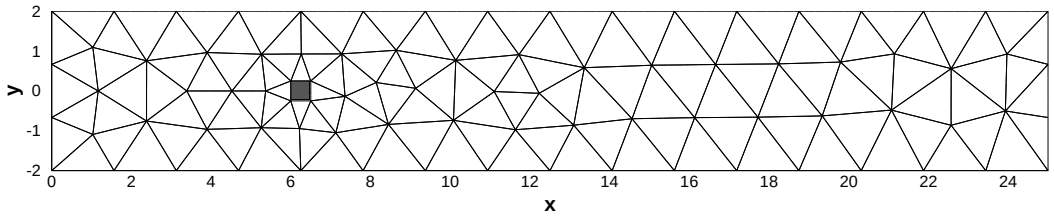


Figure 5.6 128 element mesh used for element type and mass deficit comparisons

The interpolation qualities of Lobatto and Fekete nodal distributions are compared in [47,48] by calculating Lebesgue constant and using some test functions. In both studies it is shown that the interpolation qualities of these two triangular grids are similar up to order *p*=12 but for higher orders Fekete nodal set performs much better. In the present study, their performances in LSSEM application are compared. For this purpose, the FASC problem with *Re*=10 is solved on the mesh shown in Figure 5.6. For each element types, the solutions are performed with expansion orders *p* = 6, 9, 12, 15, 18. Mass flow rate at the exit section and the recirculation length are recorded for each run and presented in Table 5.1. As seen in the table, for all expansion orders the mass flux ratios and the recirculation lengths of two different element types are very close.

Table 5.1 Mass flow rates and recirculation lengths for solutions with Lobatto and Fekete triangles

	$\dot{m}_{exit}/\dot{m}_{in}$		L_r/D	
p	Lobatto	Fekete	Lobatto	Fekete
6	0.815	0.814	0.360	0.360
9	0.838	0.837	0.454	0.454
12	0.854	0.854	0.458	0.458
15	0.868	0.868	0.460	0.460
18	0.881	0.882	0.460	0.460

Although the purpose of solving the specified case is comparing the Fekete and Lobatto distributions, it should be noted that the error of result obtained by using 18^{th} order elements is about 12% in terms of mass conservation and 6% in terms of the recirculation length. This is an evidence of poor mass conservation of least squares formulation.

The posteriori error estimates based on the least squares functional, ε_{ls}, and spectral coefficient, ε_{sp}, can be used to compare the accuracies of the results obtained by two different element types. For this purpose, maximum values of the mentioned error estimates are compared and it is observed that their first 4 digits are the same for all expansion orders. These observations show that the results obtained with both triangular element types are almost identical.

To compare the performances of the two triangular element types, another parameter to look for is the convergence rate of the iterative method used for solving the linear system of equations. For the problem solved in this part, the convergence speeds of the iterative solutions are the same for both element types up to order p=12; however, for higher orders Fekete distributions converges slightly faster. When Lobatto triangles are used, roughly 1.2 times more iterations are required to reach the same iteration error with the one obtained in Fekete triangle solutions. It is worth to note that, this comparison is also done by solving Kovasznay flow, which has a smoother solution, and it is observed that there are no differences in the convergence rates even for high orders. Therefore, it is safe to conclude that Fekete triangles can be preferred if high order (p>12) elements are required. If the lower order elements (p<12) are to be employed, both element types can be used.

Another study done with this problem is to investigate the effects of multiplying the continuity equation by a weighting factor. For this study, the solution with Lobatto triangles of order p=9 is repeated by using weight factors of 2, 5, 10, 20, 50. The mass flow rates and the recirculation lengths for each case are recorded and presented in Table 5.2. As seen in the table, the mass conservation of the method is improved considerably as the weight factor increases. Even if very high order elements are used, the mass conservation performance of the solution done by using weight factors can not be reached without weighting (see Table 5.1). However the recirculation length, which must be 0.489 as it is given in [56], is overestimated for all weight factors. Therefore, as it is also stated in [21], the overall quality of the solution is not improved when the weighting factor is used. Furthermore, the large weight factors increase the condition number of the coefficient matrix, which delays the convergence of the iterative method. Considering these negative side effects, weight factors will not be used in the solutions done in the present work.

Table 5.2 Mass flow rates and recirculation lengths for different weight factors

Weight factor	$\dot{m}_{exit}/\dot{m}_{in}$	L_r/D
1	0.838	0.454
2	0,945	0,600
5	0,988	0,680
10	0,995	0,680
20	0,997	0,700
50	0,998	0,700

The last set of solutions of the FASC problem is done to perform the adaptive p-refinement study and compare the performances of three different error measures. For this purpose, a 45° tilted square cylinder is used. This time the parameter D shown in Figure 5.3 is the diagonal of the square. The blockage ratio and the Reynolds number are calculated based on the diagonal of the square. The blockage ratio is, $B = 1/8$, and Reynolds number is, Re=10 for the problems solved in this part. The problem is first solved on a very fine mesh of 842 elements of order 19 without adaptive mesh refinement. The results of this solution are used for validation of the results obtained by using adaptive refinement. This solution is referred as "*benchmark solution*" in the following parts.

To perform the adaptive refinement study, the mesh shown in Figure 5.7 is used. Lobatto triangles are used for the solutions presented here. For all cases, solution starts with a conforming mesh of 6^{th} order elements. Then according to the error estimator used and the minimum and maximum allowable error values selected by the user, the orders of the elements are increased or decreased by 1 at each refinement level.

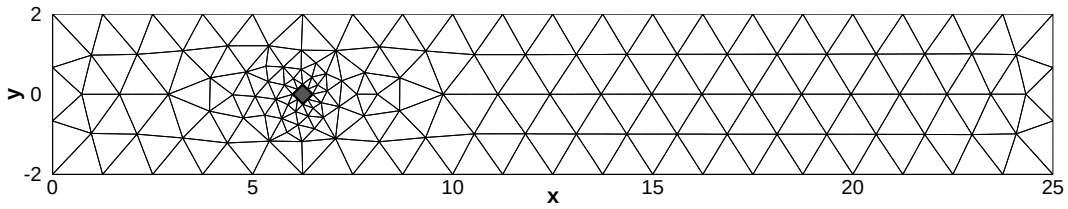


Figure 5.7 244 element mesh used in the adaptive refinement studies

The first error measure used for the refinement study is the error based on the least squares functional, ε_{ls}. The upper and lower error limits are set as 10^{-4} and 10^{-7}. The minimum and maximum allowable expansion orders are 5 and 19 respectively. With these specified parameters, the orders of all elements between $0<x<10$ are increased to 19 in 13 steps of refinement. At the 13^{th} refinement step, the elements after $x=10$ have the orders of 5 and 6. The mass flow rate ratios at $x=3$, $x=6$ (this is where the square is placed) and $x=25$ are recorded at various refinement levels and they are presented in Table 5.3. Also the x-velocity profiles at $x=6$ (velocity profile between the upper vertex of the tilted square and the upper wall of the channel) at various refinement levels are presented in Figure 5.8. It is worth to note that the velocity profile and mass flux ratios do not change considerably after the 11^{th} refinement level. Therefore results remain the same if the maximum allowable order of elements is set as 17. Furthermore, when upper and lower limits are selected as 10^{-3} and 10^{-6}, respectively, all elements between $0<x<8$ are refined up to order of 19 and the elements in the downstream part of the channel have the orders of 5 and 6, with the results being almost identical with the one obtained by setting the limits as 10^{-4} and 10^{-7}. That means the same result can be obtained by using less computer resources. The last two observations about the limits of errors and expansion orders show the difficulty in choosing convenient error and expansion order limits when adaptive p-refinement is employed.

Table 5.3 Mass flow rate ratios at various refinement steps when ε_{ls} is used

Refinement Level	$\dot{m}/\dot{m}_{in}$		
	x = 3	x = 6	x = 25
Level 0	0.923	0.840	0.843
Level 7	0.954	0.906	0.908
Level 13	0.969	0.935	0.935

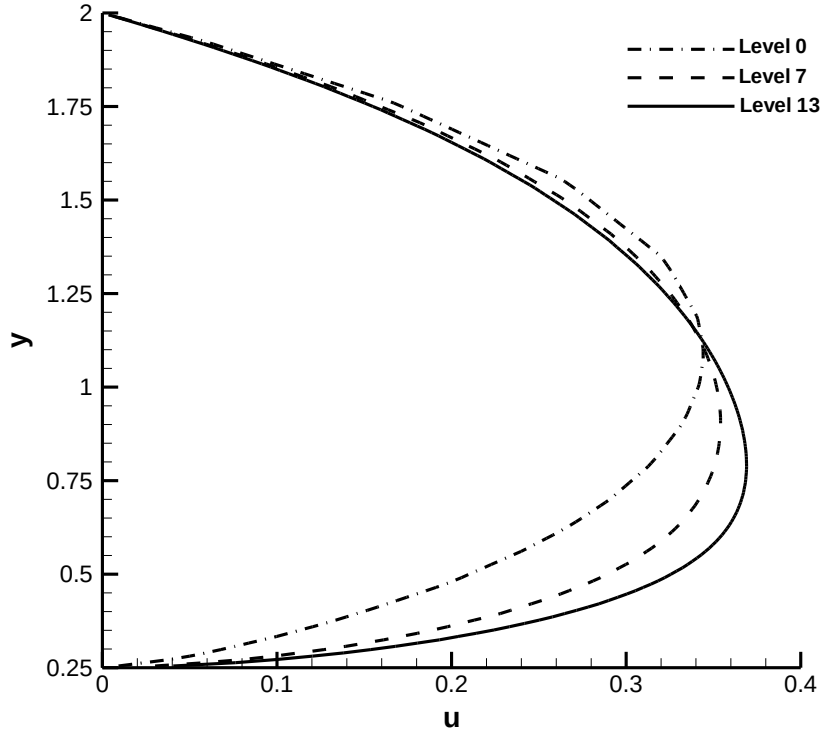


Figure 5.8 x-velocity profiles at x=6 at different levels of adaptive refinement done by using the error estimate ε_{ls}

For the second adaptive p-refinement study, spectral error estimate based on x-velocity, ε_{sp}^{u}, is used. The error limits and allowable expansion order limits are the same as the previous refinement study. The solution is performed in 14 successive refinements. This time the resulting mesh near the square is not uniform; therefore, in Figure 5.9 the orders of the elements near the square cylinder are shown for the last refinement level. The elements in upstream have the orders of 6 and 7 and the elements in the downstream have the order of 6. Also it should be noted that, at the inlet boundary (at x=0), there are two elements which are refined up to order 11. The flow rate ratios for this case are presented in Table 5.4, and the velocity profiles at x=6 are presented in Figure

5.10. As can be seen in Table 5.4, the mass conservation performance of this solution is almost same with the one obtained by using ε_{ls} error measure. However, the resulting mesh obtained by using ε_{sp}^{u} has a total number of nodes, *NN*=8565, while the mesh obtained by using ε_{ls} has *NN*=28588. This shows the importance of making refinements in the proper regions of the problem domain. The refinement studies are also done with the spectral coefficient error estimates based on other variables, v, p, ω and the results obtained turned out to be almost identical. The only difference is that, when these three error estimates (ε_{sp}^{v}, ε_{sp}^{p}, ε_{sp}^{w}) are used, the elements near the inlet and outlet boundaries are also refined up to the orders 13 and 15. However these refinements do not have a considerable effect on the results.

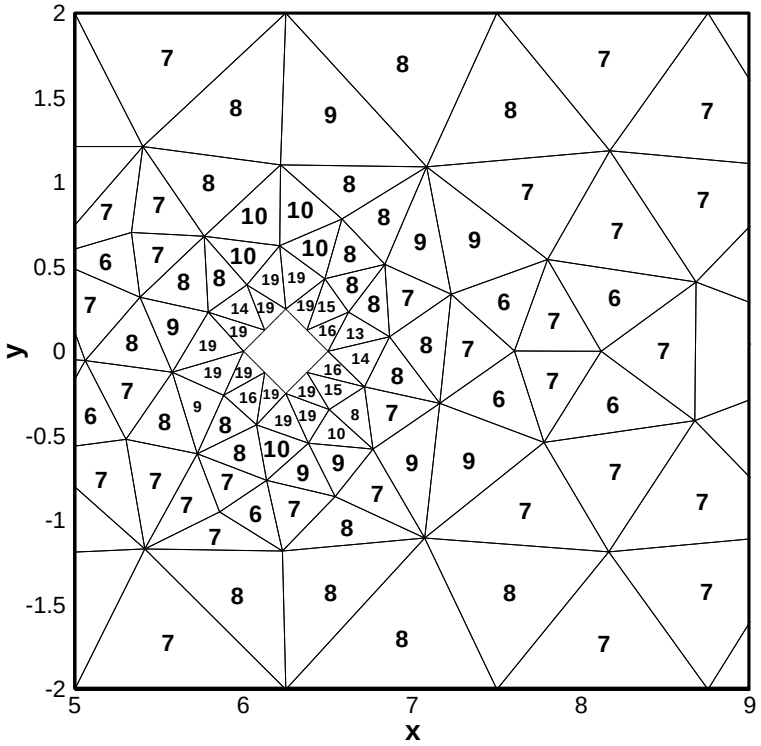


Figure 5.9 Orders of elements near the square obstacle at the last level of adaptive refinement done by using the error estimate $\boldsymbol{\varepsilon_{sp}^{u}}$

Table 5.4 Mass flow rate ratios at various refinement steps when $\boldsymbol{\varepsilon_{sp}^{u}}$ is used

Refinement Level	$\dot{m}/\dot{m}_{in}$		
	x = 3	x = 6	x = 25
Level 0	0.923	0.840	0.843
Level 8	0.956	0.910	0.912
Level 14	0.968	0.936	0.936

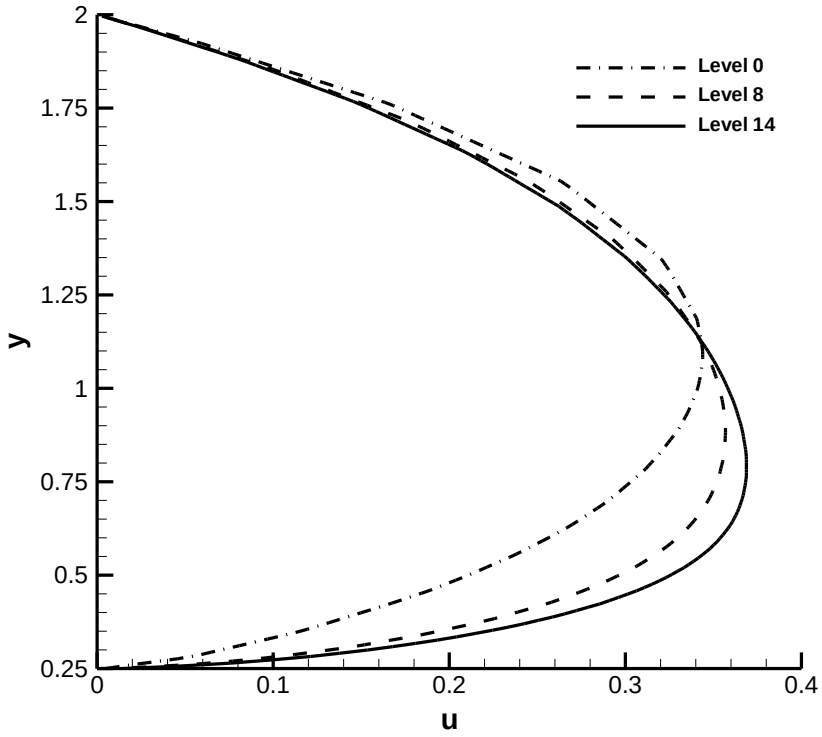


Figure 5.10 *x*-velocity profiles at *x*=6 at different levels of adaptive refinement done by using the error estimate $\boldsymbol{\varepsilon_{sp}^{u}}$

The last adaptive p-refinement study is done by using the elemental mass deficit, ε_{mass}, as the error measure. The upper and lower error measures are set as 10^{-6} and 10^{-8} respectively. The allowable minimum and maximum expansion orders are 5 and 19 respectively. When the adaptive refinement is done with these parameters, the resulting mesh is very similar to the one obtained by using ε_{ls}. The orders of $0<x<11$ are increased to 19 and the orders of the elements in the downstream are 5 and 6. There are a total of 15 refinement steps. The orders of elements between $0<x<11$ are refined uniformly between step 1 and 13, and the orders of some elements at downstream are increased at the last two refinement steps, which does not affect the final solution considerably. The flow rate ratios are presented in Table 5.5 and the velocity profiles are shown in Figure 5.11, which are very similar to the ones obtained by using other error measures. The change in the results is negligible when the error limits are increased by an order of 10 but increasing the limits by order of 10^2 results in an unsatisfactory solution for this problem.

Table 5.5 Mass flow rate ratios at various refinement steps when ε_{mass} is used

Refinement Level	$\dot{m}/\dot{m}_{in}$		
	x = 3	x = 6	x = 25
Level 0	0.923	0.840	0.843
Level 8	0.957	0.912	0.914
Level 15	0.969	0.935	0.935

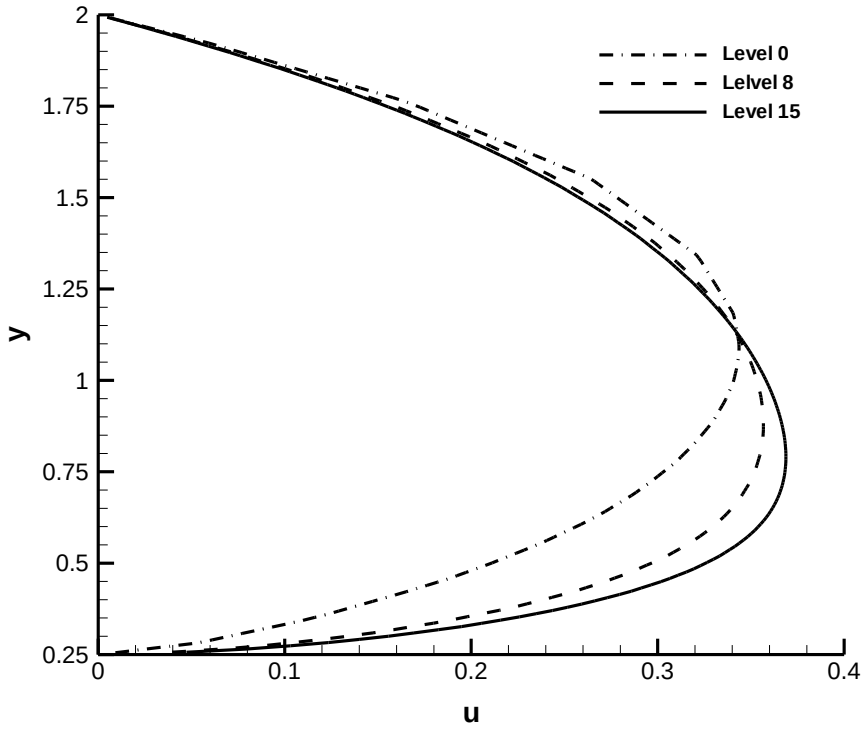


Figure 5.11 x-velocity profiles at x=6 at different levels of adaptive refinement done by using the error estimate ε_{mass}

The mass flow rate ratios are very similar for all adaptive refinement solutions done by different error measures; however, for this specific problem, when the error measure based on the spectral coefficients is used, the same result is obtained by using less number of nodes. Therefore the spectral coefficients error measure has the best performance for this problem when storage requirements and the run time are considered. But it is worth to note that the performance of the error measure is problem dependent; that is to say, ε_{sp}^{u} may not come up with a better mesh when used with a different problem. Backward facing step problem, which is discussed in Section 5.3, is an example of this.

For all types of error measures mentioned in the present work, selection of the error limits depends on the problem and it is hard to automate. One way of selecting them properly is to perform a pre-solution with the initial mesh and set the error limits according to the orders of magnitude of the maximum and minimum errors of this first solution. The other way is to watch the change of variables at several points throughout refinement steps and stop the solution when the changes in the variables at these points are small enough. Furthermore, for the mass deficit error measure the flow rate at the inlet boundary can be used to specify the error. For instance, upper bound of the error can be selected just by multiplying the inlet mass flux with 10^{-6}, which is done to specify the error limit in the present work. The total number of elements in the mesh can also be used together with inlet mass flow rate to estimate the error limits.

As can be seen in the mass flow rate ratio tables, all of the solutions results in about 6% of mass deficit in the channel. Figure 5.12 shows the velocity profile at x=6 for these adaptive refinement solutions together with the one of the benchmark solution mentioned before. Note that the velocity profiles at x=6 match almost perfectly for all three adaptive solutions; therefore, they are shown with a single line in Figure 5.12. The mass deficit in the channel is about 2% for the benchmark solution. When Figure 5.12 and the mass flow rate tables are investigated, it is obvious that the further p-refinement is required to obtain more accurate results. However, when the orders of elements are so high, the run time increases dramatically. Therefore, it is a better idea to use not only p-refinement but also h-refinement in adaptive solutions. To solve the problems with high gradients in an efficient way, a solution should be done which starts with a coarse and low order mesh, then employs an *hp* adaptive refinement strategy which keeps the orders of elements in a reasonable range.

Note that the same adaptive refinement study is also done with a coarser initial mesh of 138 elements. For this solution, the mesh around the obstacle is as coarse as possible, which means there is only one element on each face of the tilted square obstacle. Very similar refinement patterns are observed for each error estimate but of course the mass conservation through channel is worse for this case. The results of this solution are not presented here since they do not change the conclusions done about the performances of error measures.

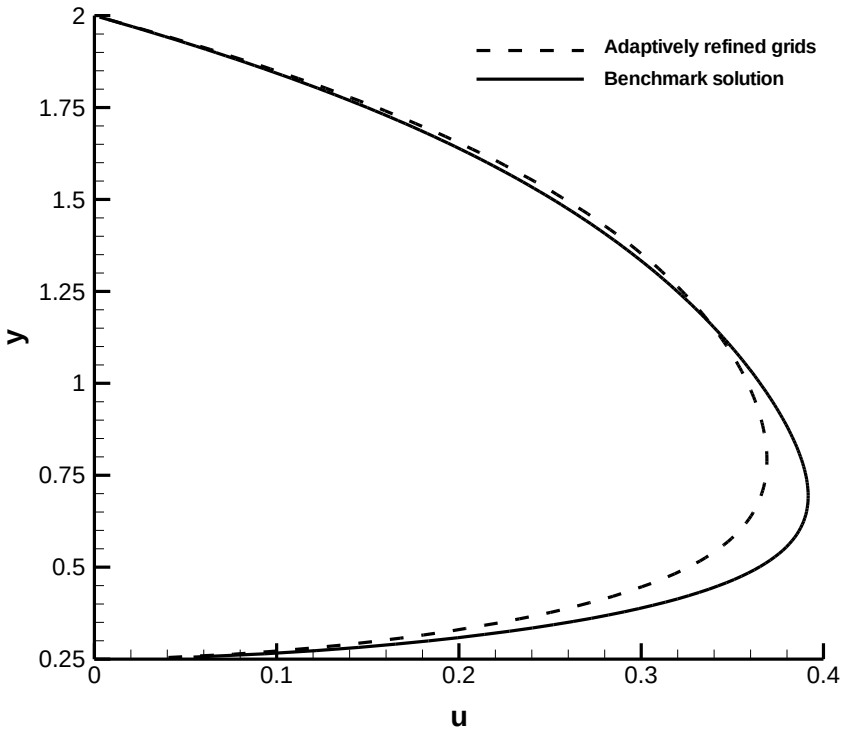


Figure 5.12 Comparison of the *x*-velocity profiles at *x*=6 for adaptively refined grids and the benchmark solution

5.3. Flow over backward facing step (BFS)

Steady viscous flow over two-dimensional backward facing step (BFS) is a standard test problem visited by many researchers before. In the present work, this problem is used for testing adaptive refinement performance of the solver. In this part, mainly two set of solutions are performed. First, a comparison of the refinement pattern and the results of two adaptive solutions obtained by using quadrilateral and triangular elements is done. With the second set of solutions, the performances of different error measures used with triangular elements are compared.

In a previous work a BFS solution with adaptive p-refinement on quadrilateral elements was performed by Özçelikkale and Sert [7]. The same problem is solved here by using a triangular mesh which is created by dividing the quadrilateral elements into two triangular elements. The problem domain and

the quadrilateral and triangular meshes are shown in Figure 5.13. Fekete nodal points are used with the triangular elements.

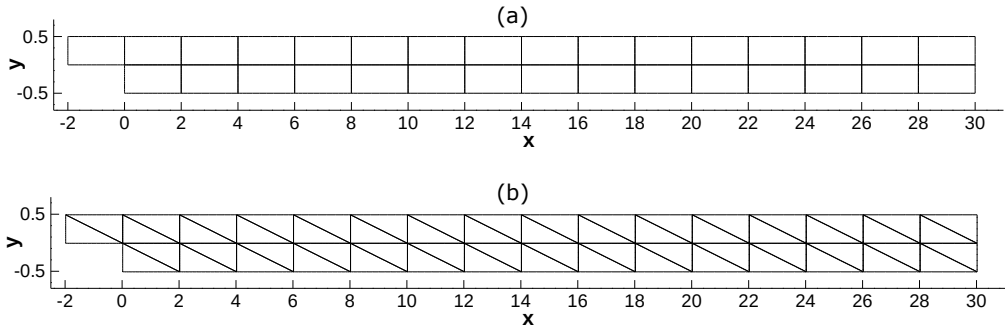


Figure 5.13 31 element quadrilateral and 62 element triangular meshes used for the BFS problem (y axis is stretched so that y/x=2)

Fluid enters the region with a developed, parabolic velocity profile, pressure and the y-velocity are set to zero at the outlet boundary and no slip boundary condition applies at the walls of the channel. Reynolds number based on the average inlet velocity and channel height is set as 800.

The case solved here is the same with the one presented in [7]. All the elements are 4^{th} order at the start of the adaptive solution. The spectral coefficient error measure based on the *x*-velocity, ε_{sp}^{u}, is used with the limits of 10^{-4} and 10^{-6}. The minimum and maximum allowable expansion orders are set as 4 and 19 respectively.

During the adaptive solutions the refinement of the triangular mesh follows a similar pattern with the quadrilateral one and the two solutions end up with perfectly matching results. The final mesh between -2<*x*<10 obtained with triangular elements are shown in Figure 5.14. Note that the orders at the downstream decreases from 10 to 6 gradually. At the beginning, both meshes have total number of nodes, *NN*=569 and when the refinement ends they come up with close *NN* values, 3451 and 3112 for triangular and quadrilateral meshes, respectively. This shows that the adaptive refinement works in a similar way for both triangular and quadrilateral elements.

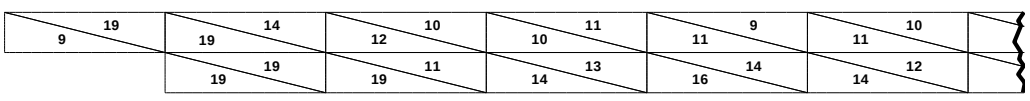


Figure 5.14 Resulting mesh for triangular elements between -2<*x*<10

Figure 5.15 shows the streamlines together with the *x*-velocity contour up to *x*=15, after where the flow is fully developed. There are two recirculation regions attached to the upper and lower walls of the channel. The lengths of the recirculation regions and the vorticities at the vortex centers are presented in Table 5.6 together with the ones obtained by Gartling [57]. Figure 5.16 shows the velocity profiles at *x*=7 for various refinement levels together with the results presented in [57].

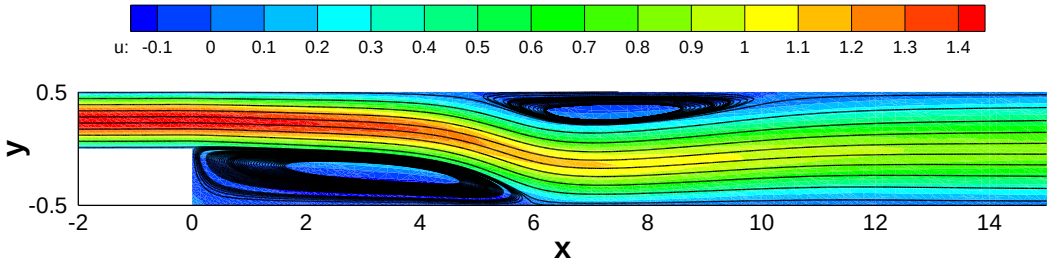


Figure 5.15 Stream lines and x-velocity profile between -2<*x*<15 (y axis is stretched so that y/x=2)

Table 5.6 The upper and lower recirculation lengths and vorticities at the vortex centers

	Lower wall		***Upper wall***	
	Recirculation length	Vorticity at the vortex center	Recirculation length	Vorticity at the vortex center
Present work	6.03	-2.252	5.50	1.150
Gartling[57]	6.10	-2.283	5.63	1.322

BFS problem is also used to compare the performances of the three error estimates. An adaptive refinement study is done with all three types of the error measures. The refinement patterns are similar for all error measures; the elements close to the step are refined the most and the order of elements decreases gradually through the downstream. They also performed similar in terms of efficiency. The solution starts with total number of nodes, *NN*=569 and at the last level of the refinements *NN*=3451 for ε_{sp}^{u}, *NN*=3915 for ε_{ls}, *NN*=3099 for ε_{mass}. The results obtained by these adaptive refinement studies are almost same. Therefore, each error estimate can be used for this problem.

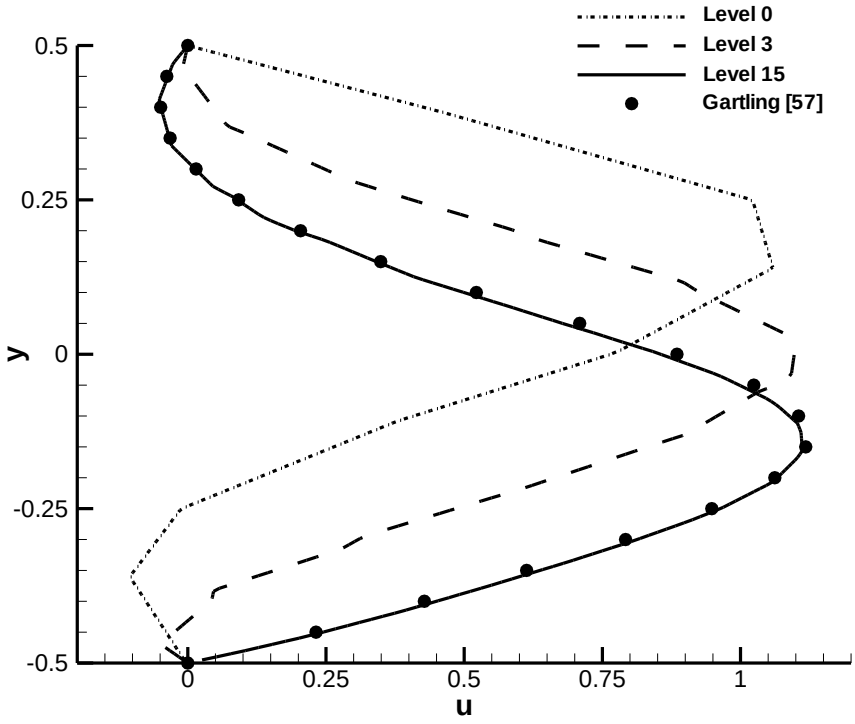


Figure 5.16 *x*-velocity profile at *x*=7 at various steps of refinement using the triangular mesh

The results and the refined meshes are not presented here for brevity; but, the distribution of the errors along the *y*=0 line is investigated, which gives an idea about the regions that are to be refined most. In BFS problem, the errors are expected to be high near the corner of the step since the gradients are high at this region. In Figure 5.17, the error distributions along *y*=0 is presented. Since the difference between the maximum and minimum errors is very high at the 0^{th} level of the refinements (all the elements are 4^{th} order at this level), the error distributions at the 2^{nd} level are presented in Figure 5.17. As expected, all error estimates have their maximum near the step and they are very low at downstream where the flow is fully developed. Note that, for brevity only spectral coefficient error estimate based on *x*-velocity is presented in Figure 5.17, but the trends of spectral coefficient error based on the other variables are also similar.

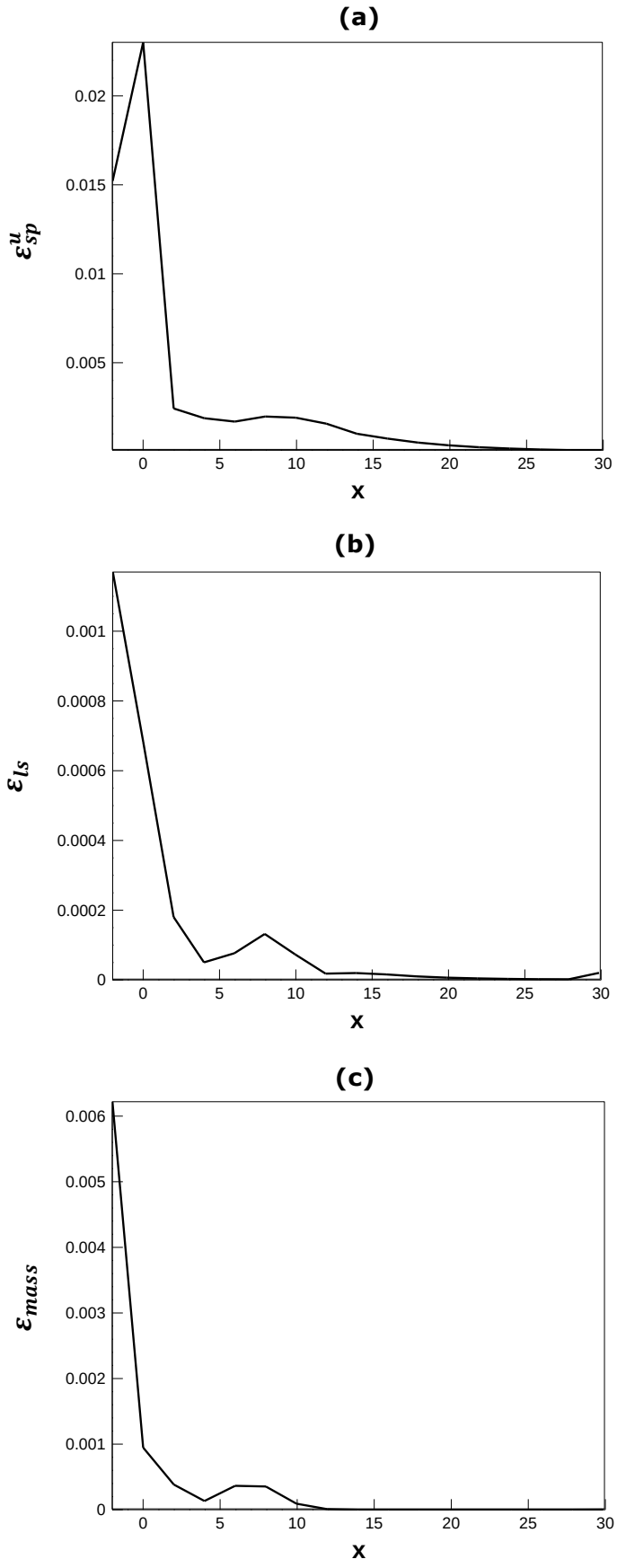


Figure 5.17 Error estimate distributions along the line y=0

CHAPTER 6

CONCLUSION

This study is a continuation of the work done by Özçelikkale and Sert [6,7], in which a flow solver based on Least-Squares Spectral Element Method (LSSEM) was developed. With the enhancements done in the present work, the solver is now compatible with the triangular elements and it uses a new error measure based on the elemental mass deficit. The developed code is capable of solving 2D, steady, incompressible, laminar flows on p-type nonconforming grids and has the capability of adaptive p-refinement. Using the solver, several benchmark problems are solved for validation and comparison purposes.

Firstly, the Kovasznay flow, which has a smooth solution, is solved and it is shown that the meshes of triangular elements with and without nonconforming interfaces provide exponential convergence. Also it is shown that the maximum true errors for this problem are very close for meshes of triangular Fekete, triangular Lobatto and quadrilateral elements. The convergence rate of the elemental mass deficit is compared with the relative true error. The exponential decay in the elemental mass deficit is observed, which makes it a promising candidate to be used as an error measure in adaptive mesh refinement.

Secondly, the confined flow over an obstacle is solved, which is a challenging problem in terms of mass conservation since it includes a sudden contraction region. Solving this problem, the performances of the Lobatto and Fekete triangular nodal distributions are compared and the same accuracy is obtained for both element types. However, for the expansion orders higher than 12, the iterative solution converged faster when Fekete nodal distribution is used. With this problem, the effect of multiplying the continuity equation with a weighting factor is also investigated and it is shown that the weighting does not improve the overall quality of the solution since it deteriorates the proper prediction of the overall flow field while improving the mass conservation. This problem is also used for comparing the performances of three error measures based on spectral coefficients, least squares functional and elemental mass deficit. The results

obtained by using all three error measures are very similar. However, the error measure based on the spectral coefficients performed best for this problem, achieving the same solution with the others by making less refinements.

The last problem solved is the flow over a backward facing step. This problem is solved for two purposes; comparing the refinements done on quadrilateral and triangular elements and comparing the performances of three different error measures. The case solved by Özçelikkale and Sert [7] is revisited to compare the refinements done on quadrilateral and triangular elements. A triangular mesh is created by dividing each quadrilateral element into two and the adaptive mesh refinement study is performed with the same error measures for both meshes. It is observed that the refinement patterns are very similar and the results obtained are identical. Adaptive refinement studies are done using the different three error measures and it is observed that the performances of all three error indicators are very similar for this problem.

There is no single error measure that works best for all problems. Therefore the selection of the error indicator depends on the problem. In the present work, this is shown by solving the last two problems. In the first problem the error based on the spectral coefficients performed the best but in the second problem there is no remarkable difference in their performances.

The present work has two main interests. The first one is the selection of triangular element type. As stated in [7], use of very high order elements decreases the solution efficiency dramatically. Therefore from a practical point of view comparison of triangular elements should be done for relatively low orders ($p<12$). In the present work it is shown that, for orders $p<12$, the performances of Fekete and Lobatto triangles are the same in terms of accuracy and convergence speed. The two element types should also be compared in terms of the complexity of the grid construction. Knowing that constructing the grid for Fekete nodal points is cumbersome and constructing the Lobatto grid is much simpler; using Lobatto triangles can be preferable.

The second interest of the research is on the adaptive refinement performance of the error measure based on the elemental mass deficit. The performance of this error estimator depends on the problem, which is also the case for other error measures. Furthermore, the difficulty of selecting the error limits is an issue for this error measure too. Still it can be used in problems in which poor mass conservation is expected.

As a future work, the *hp* adaptive refinement strategy should be implemented to the solver to obtain accurate results in an efficient way. In h-refinement, using triangular elements is advantageous since it provides an opportunity to divide elements without creating h-nonconforming faces. Also, the weighted LSSEM formulation can be used in an adaptive refinement strategy in which different weight factors are used for each element according to the elemental error estimates. By using different weight factors for each element, the mass conservation may be improved without affecting the prediction of the flow field significantly. Another future work is to add the ability to work on curved edge triangular elements; the solver does not support the triangular elements with curved edges at the moment. Furthermore, using modal expansion bases has some advantageous in adaptive refinement implementations, which can be employed and compared against the nodal expansions already used. Lastly, the solver works on hybrid triangular/quadrilateral meshes; therefore, fully automated solutions with Cartesian grids supporting cut cells can be performed.

REFERENCES

[1] J. H. Ferziger ve M. Peric, Computational Methods for Fluid Dynamics, Springer-Verlag, 2002.

[2] H. K. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Prentice Hall, 1995.

[3] J. N. Reddy, An Introduction to The Finite Element Method, McGraw-Hill, 1993.

[4] C. Pozrikidis, Introduction to Finite and Spectral Element Methods Using MATLAB, Chapman and Hall/CRC, 2005.

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

[6] A. Ozcelikkale, *Development of an incompressible, laminar flow solver based on least squares spectral element method with p-type adaptive refinement capabilities,* M.S. Thesis, METU, June 2010.

[7] A. Ozcelikkale and C. Sert, "Least-squares spectral element solution of incompressible Navier-Stokes with adaptive refinement," *Journal of Computational Physics,* vol. 231, pp. 2755-3769, 2012.

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

[9] I. Babuska, "Error bounds for finite element method," *Numerische Mathematik,* vol. 16, pp. 322-333, 1971.

[10] F. Brezzi, "On the existence, uniqueness and approximation of the saddle point problems arising from lagrangian multipliers," *R.A.I.R.O.,* vol. 8, pp. 129-151 , 1974.

[11] A. N. Brooks and T. J. R. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the

incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 32, pp. 199-259, 1982.

[12] T. J. R. Hughes, L. P. Franca and G. M. Hulbert, "A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 73, pp. 173-189, 1989.

[13] J. Donea, "A Taylor–Galerkin method for convective transport problems," *International Journal for Numerical Methods in Engineering,* vol. 20, pp. 101-119, 1984.

[14] 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, pp. 878-906, 1998.

[15] J. P. Pontaza and J. N. Reddy, "Least-squares finite element formulations for viscous incompressible and compressible fluid flows," *Computer Methods in Applied Mechanics and Engineering,* vol. 195, pp. 2454-2494, 2006.

[16] 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, pp. 75-88, 1973.

[17] 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, pp. 341-358, 1974.

[18] P. B. Bochev and M. D. Gunzburger, "Finite element methods of least-squares type," *SIAM Rev.,* vol. 60, pp. 789-837, 1998.

[19] 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, pp. 183-197, 2003.

[20] 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, pp. 480-489, 1997.

[21] M. M. J. Proot and M. I. 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,* vol. 27, pp. 189-401, 2006.

[22] 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.

[23] 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,* vol. 221, pp. 649-665, 2007.

[24] 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,* vol. 225, pp. 1590-1602, 2007.

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

[26] P. Bolton and T. R. W., "A least-squares finite element method for the Navier–Stokes equations," *Journal of Computational Physics,* vol. 213, pp. 174-183, 2005.

[27] T. Kattelans and W. Heinrichs, "Conservation of mass and momentum of the least squares spectral collocation scheme for the Stokes problem," *Journal of Computational Physics,* vol. 228, pp. 4649-4664, 2009.

[28] A. T. Patera, "A spectral element method for fluid dynamics: Laminar flow in a channel expansion," *Journal of Computational Physics,* vol. 54, pp. 468-488, 1984.

[29] 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.

[30] 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.

[31] W. Heinrichs, "Least-squares spectral collocation for the Navier–Stokes equations," *Journal of Scientific Computing,* vol. 21, pp. 81-90, 2004.

[32] O. C. Zienkiewicz, J. P. D. S. R. Gago and D. W. Kelly, "The hierarchical concept in finite element analysis," *Computers & Structures,* vol. 16, pp. 53-65, 1983.

[33] M. Dubiner, "Spectral methods on triangles and other domains," *Journal of Scientific Computing,* vol. 6, pp. 345-390, 1991.

[34] 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, pp. 3775-3802, 1995.

[35] 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, pp. 189-229, 1995.

[36] C. Mavriplis and A. Van Rosendale, "Triangular spectral elements for incompressible fluid flow," ICASE Report, 1993.

[37] W. Heinrichs, "Spectral collocation on triangular elements," *Journal of Computational Physics,* vol. 145, pp. 743-757, 1998.

[38] W. Heinrichs and B. Loch, "Spectral schemes on triangular elements," *Journal of Computational Physics,* vol. 173, pp. 279-301, 2001.

[39] T. Warburton, L. Pavarino and J. S. Hesthaven, "A pseudo-spectral scheme for the incompressible Navier–Stokes equations using unstructured nodal elements," *Journal of Computational Physics,* vol. 164, pp. 1-21, 2000.

[40] R. Pasquetti and F. Rapetti, "Spectral element methods on triangles and quadrilaterals: comparisons and applications," *Journal of Computational*

Physics, vol. 198, pp. 349-362, 2004.

[41] W. Heinrichs, "An adaptive least-squares spectral collocation method with triangular elements for the incompressible Navier-Stokes equations," *Journal of Engineering Mathematics,* vol. 56, pp. 337-350, 2006.

[42] J. S. Hesthaven, "From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex," *SIAM Journal on Numerical Analysis,* vol. 35, pp. 655-676, 1998.

[43] L. Bos, "Bounding the Lebesgue function for Lagrange interpolation in a simplex," *Journal of Approximation Theory,* vol. 38, pp. 43-59, 1983.

[44] Q. Chen and I. Babuška, "Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle," *Computer Methods in Applied Mechanics and Engineering,* vol. 128, pp. 405-417, 1995.

[45] M. A. Taylor, B. A. Wingate and R. E. Vincent, "An algorithm for computing Fekete points in the triangle," *SIAM Journal on Numerical Analysis,* vol. 38, pp. 1707-1720, 2000.

[46] M. G. Blyth and C. Pozrikidis, "A Lobatto interpolation grid over triangle," *IMA Journal of Applied Mathematics,* vol. 71, pp. 153-169, 2006.

[47] M. G. Blyth, H. Luo and C. Pozrikidis, "A comparison of interpolation grids over the triangle or the tetrahedron," *Journal of Engineering Mathematics,* vol. 56, pp. 263-272, 2006.

[48] R. Pasquetti and R. F., "Spectral element methods on unstructured meshes: which interpolation points?," *Numerical Algorithms,* vol. 55, pp. 349-366, 2010.

[49] 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, pp. 569-580, 1987.

[50] F. Taghaddosi, W. Habashi, G. Guevremont ve D. Ait-Ali-Yahia, "An adaptive least-squares method for the compressible Euler equations," *International Journal for Numerical Methods in Fluids,* vol. 31, pp. 395-411, 1999.

[51] A. Galvao, M. Gerritsma and B. D. Maerschalck, "hp-Adaptive least squares spectral element method for hyperbolic partial differential equations," *Journal of Computational and Applied Mathematics,* vol. 215, pp. 409-418, 2008.

[52] R. Henderson, "Dynamic refinement algorithms for spectral element methods," *Computer Methods in Applied Mechanics and Engineering,* vol. 175, pp. 395-411, 1999.

[53] D. Engwirda, "MESH2D - Automatic Mesh Generation - File Exchange - MATLAB Central," [Online]. Available: http://www.mathworks.com. [Accessed 27 08 2012].

[54] G. E. Karniadakis and S. J. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, Oxford University Press, 2005.

[55] L. I. G. Kovasznay, "Laminar flow behind a two-dimensional grid," *Mathematical Proceedings of the Cambridge Philosophical Society,* vol. 44, pp. 58-62, 1948.

[56] M. Breuer, J. Bernsdorf, T. Zeiser and F. Durst, "Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite volume," *International Journal of Heat and Fluid Flow,* vol. 21, pp. 186-196, 2000.

[57] D. K. Gartling, "A test problem for outflow boundary conditions-flow over a backward-facing step," *International Journal for Numerical Methods in Fluids,* vol. 11, pp. 953-967, 1990.