Adaptive Finite Element Methods for Partial Differential Equations

G14SCD

Scientific Computation Dissertation

Summer 2011/2012

School of Mathematical Sciences

University of Nottingham

Burhan ALVEROGLU

Supervisor: Dr Edward HALL

Division: Applied Mathematics

September 2012

I have read and understood the School and University guidelines on plagiarism.

I confirm that this work is my own, apart from the acknowledged references.

To my beloved late mother who passed away in 2009, my dear family and my loving girlfriend Dilara...

Acknowledgements

Firstly, my infinite thanks should be given to the most gracious and merciful ALLAH(SWT) for helping me to complete this study.

Secondly, I am grateful to have been advised and supported from Dr. Edward Hall and I cannot thank enough for the scientific and technical help of him.

Last but not least, I am grateful to Prof. Paul Houston for advising me on the whole year.

Abstract

Most of physical phenomena can be modeled by partial differential equations. However, it is extremely hard to find an exact solution of partial differential equations in most cases. Therefore, the derivation of numerical solutions of them is very important. On the other hand, the usual methods in numerical approximation methods such as approximating the numerical solutions by refining solution mesh uniformly have expensive computational costs. Also, the accuracy of the solution is not acceptable. In this study, therefore, we tried to implement adaptive finite element methods to reduce the computational costs and get more accurate numerical solutions. To do this, adaptive mesh refinement algorithms were produced in the software toolkit APTOFEM. In addition, theoretical methods of adaptive finite element methods were included in the study.

Contents

List of Figures

List of Tables

1 Introduction

Most solutions in applied mathematics problems arise in many real physical areas such as heat distribution from a source, electrodynamics and fluid flow modelings. All of these can be found as the solution of partial differential equations with suitable boundary conditions [1]. Therefore, finding the solutions of the partial differential equations is very important. However, finding an exact solution of a partial differential equation with classical calculus methods is not possible. Instead, mathematicians have been trying to develop computational methods to find approximate solutions since the first attempts of Wilhelm Leibniz (July 1, 1646-November 14, 1716). A milestone in these studies was the invention of modern computers in the 1940s [2]. Since this invention, many methods have been developed for the numerical solution of partial differential equations. Finite element methods are one of the most important types of those methods.

Finite element methods have had a great impact on the numerical solution of partial differential equations. The main idea of the finite element methods, FEM, is to use piecewise approximations, usually polynomials, instead of continuous functions [1]. On the other hand, they are constructed on weak and variational formulations of the partial differential equations. This property provides a proper construction for the solution of the problems and the solution can be represented in the integral of a quantity over the domain. However, we can break up the integral of a measurable function over a domain into the sum of in-

tegrals over subdomains/elements whose union is the whole domain. In this way we can approximate the solution locally over a typical element [3]. Because of those properties, finite elements methods are extremely useful for the numerical solution of partial differential equations.

In this study we will try to analyse the numerical solutions of incompressible fluid flow problems. However, one important and common problem that occurs in finding numerical solutions of practical problems of computational fluid dynamics is local singularities. These singularities usually arise from corners of the domain or on the interior/boundary layers (section 3.5) and they reduce the accuracy of the numerical approximation. Those parts of the domain are usually called critical parts/areas of the domain. One obvious solution to that problem is to re-mesh the domain by putting more node points near the critical areas. However, the question is how we can identify the critical parts of the domain and create a good balance between the re-meshed parts of the domain and the other parts to achieve an optimal accuracy for the approximation [4]. The first solution is to refine the domain uniformly without identifying critical areas. However, this solution increase the cost of the numerical computation. Therefore, a new method was developed by mathematicians called the adaptive mesh refinement or the h-refinement method.

The main goal of the adaptive mesh refinement method is to make a number of local refinements to distribute the errors equally among the mesh and achieve a maximum accuracy for the desired approximation [5], [6]. The required informa-

tion about distribution of the errors is satisfied by a *posteriori* error estimators that relate to the approximated solution [6].

Suppose that u is a continuous function which satisfies:

$$\mathcal{L}(u) = f, \tag{1.1}$$

subject to the appropriate initial/boundary conditions, where $\mathcal{L}$ is a partial differential operator and f is a continuous function. Also, suppose that u_h is satisfied by:

$$\mathcal{L}_h(u_h) = f, \tag{1.2}$$

subject to the initial/boundary conditions of (1.1), where $\mathcal{L}_h$ is the discretisation of the operator $\mathcal{L}$. Then the a *priori* error bound can be written as

$$\|u - u_h\|_V \leq \mathrm{C}\|u\|_W, \tag{1.3}$$

for some constant C and appropriate vector norms [7]. In addition, the a posteriori error bound can be written as:

$$\|u - u_h\|_V \leq \mathrm{C}\|u_h\|_W, \tag{1.4}$$

for some constant C and appropriate vector norms, usually H^1 or L^2 norms known as global norms [7]. However, we do not know the exact solution u for most ap-

plications but we usually find an approximate solution u_h. Therefore, computing a posteriori error bounds is easier than computing a priori error bounds.

In practical applications two types of estimators are usually used. Typically, they are residual type error estimators, which are derived by the residual of the finite element approximation, and recovery type error estimators, which are used for time-dependent Navier-Stokes equations. In this study, we will not use the second type but see [8],[9] and [10] for details of this type of estimator [11].

The construction of a posteriori error estimators is very important to determine the critical parts of the domain [12]. In addition, it is desirable that the calculation of the a posteriori error estimator requires less computation than calculation of the numerical solution [4]. However, the usual a posteriori error analysis uses global norms but these norms involve stability constants which are usually unknown. Also, they do not provide sufficient error bounds in the case of computing derived quantities such as lift or drag instead of an overall solution. Thereby, using those norms in a posteriori error estimation is not useful in most practical applications [13]. One way to overcome this problem is by using duality arguments for the a posteriori error estimation of the finite element methods [14].

The a posteriori error estimation with duality arguments can be constructed with the dual weighted residual method. In this method, the error is computed by using a linearised dual problem and the method is commonly used to compute or optimise a quantity $J(u)$ of a functional $J(\cdot)$ from the solution u of the discretised continuous model. If we assume that u_h is the approximate solution of the

model and the domain is divided into subdomains, then the aim is to compute the error $J(u) - J(u_h)$ in terms of local residues $\rho_E(u_h)$. These local residues can be computed on each element E. Simply:

$$|J(u) - J(u_h)| \approx \sum_{E \in \mathbb{T}_h} \rho_E(u_h)\omega_E,$$

where weights ω_E are computed from the approximate solution of the dual problem [14]. As Becker & Rannacher mentioned in [14], the construction of this method started with the studies of Babuška & Miller [15],[16] and [17].

On the other hand, there is a tool to measure the quality of the estimators. This tool is called the effectivity index:

$$\text{effectivity} = \frac{\text{approximated error}}{\text{exact error}}.$$

A property related to effectivity is asymptotic exactness. Simply, if the effectivity index of an estimator converges to one when mesh size gets smaller and smaller, then the estimator is asymptotically exact [18]. As stated in the same paper [18], a complete investigation of asymptotically exact error estimators was published in the papers of Babuška and Rheinboldt [19], [20].

The development of the adaptive procedures in finite element methods began in the middle of the 1970s with I.Babuška's studies [21],[22],[23] and [24], and adaptive methods have improved since the first Workshop on Adaptive Methods

for Partial Differential equations in 1982 [25, p.150]. The method has since become a robust tool in many areas of scientific computation [26]. R.Rannacher identified the main steps of the method in [4] as follows:

1. Defining a tolerance, TOL, to measure the error.

2. Computing a numerical solution

3. Computing a computable a posteriori error bound. If the error $<$ TOL, go to step 7.

4. Identifying elements where the error is locally small or large.

5. Identifying elements for refinement or derefinement according to step 2.

6. Changing the size of local elements or degree of local polynomials. Then go to step 1.

7. Stop.

The content of this study is as follows. Firstly, I will conduct a review of adaptive refinement methods for finite element methods by providing theoretical information about the method and supporting it with some computational examples. In the second section some basic definitions and theorems for finite element methods will be stated so that the reader can easily analyse the other sections.

The third section can be thought of as an introduction to adaptive refinement methods. The interior/boundary layers are the regions of the domain in which

the solution changes too sharply to be well approximated by a coarse grid. In such areas, an adaptive refinement method needs to be applied carefully. On the other hand, the layers usually arise in convection-diffusion flow models. Therefore, the second section will include a brief overview of finite element methods for convection-diffusion flow problems. The construction of the weak formulation and the finite element discretisation for some of the one dimensional convection-diffusion mathematical models will be stated. Also, a posteriori error bounds for the model will be presented with a brief proof. In addition, the adaptive refinement method will be applied to the convection-diffusion flow problems by using the code generated in the software package APTOFEM [27]. Finally, the results for adaptive mesh refinement and uniform mesh refinement methods will be stated to show the effectivity of adaptive mesh refinements.

The fourth section will involve a review of the theory of finite element methods for the Stokes equations, which is summarised from the book 'Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics'[28]. The construction of the weak formulation and finite element discretisation will be analysed deeply and arguments for the existence and uniqueness of solutions for the Stokes equations will be considered. Then some error theories will be discussed for those equations. Finally, a numerical solution which was constructed by using APTOFEM [27] for a chosen problem will be stated.

The fifth section will be about the finite element approximation of the Navier-Stokes equations and this section is the most important part of this study. This

part was also summarised from the same book [28] as the previous section and it will involve a review of finite element modeling of the Navier-Stokes equations. Additionally, the Newton iteration, which is required to find a solution, will be discussed. Finally, the adaptive refinement method with a dual weighted residual error estimator will be applied by using APTOFEM [27] to a specific incompressible flow problem and the graphs of the solution data and refined mesh will be exhibited to show the effectiveness of the DWR method.

2 Basic concepts and related theorems

In this section some basic concepts and related theorems about finite element spaces and the adaptive mesh refinement method will be stated to give clear understanding for the following sections.

Definition 1. Let $\Omega \subset \mathbb{R}^n, n \geq 1$, be an open bounded set.

- The notation $C(\Omega)$ denotes the set of all real-valued continuous functions on Ω.
- The notation $C(\overline{\Omega})$ denotes the space of all real-valued uniformly continuous functions on $\overline{\Omega}$.

(see [29]).

Definition 2. Introduce the multi-index:

$$\alpha = (\alpha_1, ..., \alpha_n) \in N.$$

Define the length of the multi-index as:

$$|\alpha| = \alpha_1 + ...\alpha_n.$$

Also, define:

$$D^\alpha = \frac{\partial^{|\alpha|}}{\partial^{\alpha_1} x_1 ... \partial^{\alpha_n} x_n},$$

where D^α is the weak derivative operator [29]. Then, the spaces of continuous functions can be written as:

$$C^m(\Omega) = \{v \in C(\Omega) : \ D^\alpha v \in C(\Omega) \ \forall |\alpha| \leq m\},$$

$$C^m(\overline{\Omega} = \{v \in C(\Omega) : \ D^\alpha v \in C(\overline{\Omega}) \ \forall |\alpha| \leq m\}.$$

(see [29]).

Definition 3. Let $\Omega \subset \mathbb{R}^n$. Then, the notation $L_2(\Omega)$ denotes the set of all real-valued functions defined on an open subset Ω such that:

$$\|u\|_{L_2(\Omega)} := \left(\int_\Omega |u(x)|^2 dx \right)^{1/2} < \infty.$$

The space $L_2(\Omega)$ can be equipped with the inner product:

$$(u, v) := \int_\Omega u(x)v(x)dx.$$

Clearly:

$$\|u\|_{L_2(\Omega)} = (u, u)^{1/2}.$$

(see [29]).

Definition 4. : (**Sobolev Space.**)

For a non-negative integer m define the following space:

$$H^m(\Omega) = \{u \in L_2(\Omega) : D^\alpha u \in L_2(\Omega), |\alpha| \leq m\}.$$

This space is called a Sobolev space of order m. Also, it is equipped with the Sobolev norm:

$$\|u\|_{H^m(\Omega)} := \left(\sum_{|\alpha| \leq m} \|D^\alpha u\|^2_{L_2(\Omega)} \right)^{1/2}.$$

In detail, the following subspace is used commonly in the finite element method:

$$H_0^m(\Omega) = \{u \in H^m(\Omega) : u|_{\partial\Omega} = 0\},$$

for a non-negative integer m and an open interval (a,b). In addition, the Sobolev semi-norm can be defined as:

$$|u|_{H^m(\Omega)} := \left(\sum_{|\alpha| = m} \|D^\alpha u\|^2_{L_2(\Omega)} \right)^{1/2}.$$

Lemma 2.1. *(**The Cauchy-Schwarz inequality**). Let u and v belong to $L_2(\Omega)$, then:*

$$|(u, v)| \leq \|u\|_{L_2(\Omega)} \|v\|_{L_2(\Omega)}.$$

[30, p.34]

Let V denotes a Hilbert space equipped with inner product $(\cdot, \cdot)_V$ and norm $\|\cdot\|_V$. Then, the following definition can be stated:

Lemma 2.2. *(**Poincaré Friedrichs inequality**). Assume that* Ω *is a bounded subset of Euclidean space* R^n *with a sufficiently smooth boundary* $\partial\Omega$ *and let* $u \in H_0^1$. *Then there exists a constant in* Ω, *independent of* u, *such that:*

$$\|u\|_{L_2(\Omega)} \leq c_* \|\nabla u\|^2_{L_2(\Omega)}.$$

[28, p.37]

Definition 5. (**Linear and bilinear functionals**). Let V be a linear space. Then, ℓ is a linear functional on V if ℓ: V$\rightarrow R$ and ℓ is linear, i.e

$$\ell(\alpha u + \beta v) = \alpha\ell(u) + \beta\ell(v),$$

for all u,v $\in$ V and $\alpha, \beta \in R$.

Also, a$(\cdot,\cdot)$ is a bilinear functional on V X V if a:V X V $\rightarrow R$ and is linear in each argument, i.e.:

$$a(u, \alpha v + \beta w) = \alpha a(u, v) + \beta a(u, w),$$

$$a(\alpha u + \beta v, w) = \alpha a(u, v) + \beta a(u, w),$$

for all u, v, w $\in$ V and $\alpha,\ \beta \in R$ [30, p.34].

Definition 6. (**Coercivity**). A bilinear functional a$(\cdot,\cdot)$ is said to be coercive

on V x V if there exists a positive constant c_0 such that:

$$a(u,u) \geq c_0 \|u\|_V^2 \quad \forall u \in V.$$

[28, p.121]

Definition 7. (**Continuity**). A bilinear functional a$(\cdot,\cdot)$ is said to be continuous on V x V if there exists a positive constant c_1 such that:

$$a(u,v) \leq c_1 \|u\|_V \|v\|_V \quad \forall u,v \in V.$$

Moreover, a linear functional $\ell(v)$ is said to be continuous on V if there exists a positive constant c_2 such that:

$$\ell(v) \leq c_3 \|v\|_V \quad \forall u \in V.$$

[28, p.121]

Theorem 2.3. *(**Lax-Milgram theorem**).*

If $a(\cdot,\cdot)$ is continuous and coercive over $V \times V$ and $\ell(\cdot)$ is continuous over V then there exists a unique solution $u \in V$ for:

$$a(u,v) = \ell(v) \quad \forall v \in V.$$

[28, p.122].

3 Adaptive finite element methods for one dimensional convection-diffusion equations

In this section, we will focus on finding an approximate solution of the one dimensional convection-diffusion problem with a finite element method including an adaptive mesh refinement strategy. Hence, the constructions of the weak formulation, the finite element space for the numerical solution and finite element approximation method will be considered in detail. Also, some error theories related to the a posteriori error estimation will be stated to understand the accuracy of the approximate solution. We will also investigate the dual weighted residual methods for the a posteriori error estimation to increase the accuracy. After this, we will try to measure the error accuracy in two examples and the approximate solutions of them will be satisfied to show that our code satisfies the adaptive mesh refinement method.

3.1 The model problem

The general convection-diffusion equation can be stated as:

$$-\epsilon\nabla^2 u + \vec{\omega} \cdot \nabla u = f, \tag{3.1}$$

with the suitable boundary conditions where $\epsilon > 0$ and $f \in L_2(\Omega)$ [28, p.113]. First of all, we will try to solve a particular one dimensional convection-diffusion

problem which is defined as:

$$\begin{aligned} -u'' + b(x) \cdot u' &= f(x), \qquad x \in (0,1), \\ u(0) = 0 \ \text{ and } \ u(1) &= 0. \end{aligned} \tag{3.2}$$

Some assumptions are made to guarantee the existence of a solution to (3.2) and (2.2). They are:

$$b(x) \in C[0,1], \quad f \in L_2(0,1),$$

$$\text{and} \quad -\frac{1}{2}b'(x) \geq 0, \quad x \in (0,1).$$

[7].

3.2 Weak formulation of the model problem

The weak formulation of (3.2) has been constructed as:

$$a(u,v) = \ell(v) \quad \forall\ v \in H_0^1(\Omega), \tag{3.3}$$

where bilinear functional $a(\cdot,\cdot)$ and linear functional $\ell(\cdot)$ are defined respectively to be:

$$\begin{aligned} a(u,v) &= \int_0^1 (u'v' + bu'v)dx, \\ \ell(v) &= \int_0^1 fvdx. \end{aligned} \tag{3.4}$$

See lecture notes [7].

Also, it can easily be shown that the bilinear functional $a(\cdot,\cdot)$ is continuous and coercive over V×V and linear functional $\ell(\cdot)$ is continuous over V. Thereby, the Lax-Milgram theorem (2.3) guarantees that there exists a unique solution u ∈ V such that:

$$a(u,v) = \ell(v) \quad \forall v \in V.$$

See lecture notes [7].

3.3 Finite element method for the model problem

Computing the finite element solution for (3.2) is one of our main concerns. Firstly, we can construct the initial mesh by dividing the domain [0,1] into N subintervals. Figure 1 displays such a mesh. Then, the finite element solution, say u_h, will be written as a linear combination of the finite element basis functions which are defined as:

$$\phi_k = (1 - \left|\frac{x - x_k}{h_k}\right|)_+, \qquad k = 1, ..., N-1.$$

If those finite element basis functions are used to define the finite element space V_h as:

$$V_h = span\{\phi_i\}, \quad i = 1, ..., N-1.$$

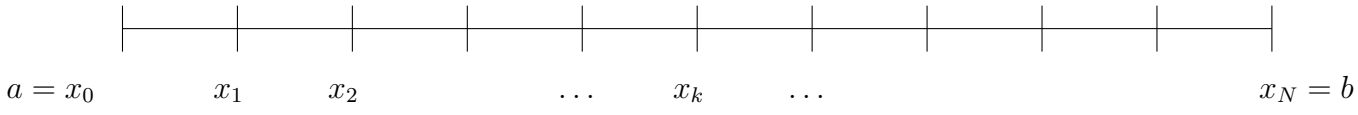

Figure 1: One dimensional uniform mesh.

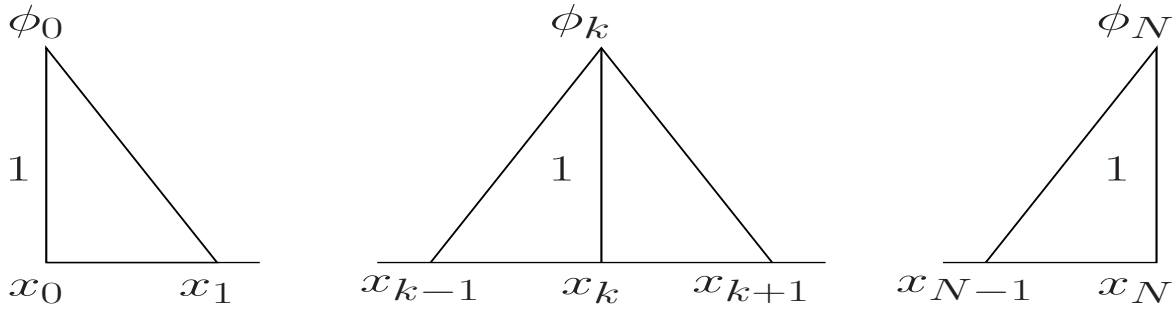

Figure 2: Shape of the one dimensional basis functions

Figure 2 displays shape of these basis functions. Then the finite element approximation is given by:

Find $u_h \in V_h$ such that

$$a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h. \tag{3.5}$$

The existence and uniqueness of the solution can be shown [29] easily by using Lax-Milgram theorem (2.3). See lecture notes [7].

3.4 A posteriori error bounds for the model problem

A fundamental question in the general numerical methods is how close the approximate solution u_h is to an exact solution of the problem. Since it is hard to find exact solutions for most problems occurring in science, the answer is not quite

as easy as it seems. However, we can measure the accuracy of the approximate solution in the absence of an exact solution to the problem by finding a posteriori error bounds. To do this, finite element residuals are defined as:

$$R(u_h)(x) = f(x) - (-u_h''(x) + b(x)u_h'(x)), \quad x \in (x_{i-1}, x_i), \tag{3.6}$$

for i=1,...,N.

Basically, the residual measures the extent to which u_h fails to satisfy (3.2) on the interval (0,1) [7].

Before giving the theorem that satisfies a posteriori error estimates for the error measured in L_2-norm, we need to state the fundamentals of the interpolation theory.

Definition 8. (**Interpolant**.) Finite element interpolant of a funtion u$\in H_0^1$ can be written as

$$\mathcal{I}u(x) = \sum_{i=1}^{n-1} u(x_i)\phi(x_i),$$

where n is the number of elements. Clearly $\mathcal{I}u \in V_h$.

Theorem 3.1. ***(Interpolation error bounds.)*** *Assume that u$\in H^2(0,1)$ and let $\mathcal{I}u \in V_h$ be the interpolant of u. Then, the following error bounds can be written:*

$$\|u - \mathcal{I}u\|_{L_2(0,1)} \leq (\frac{h}{\pi})^2 \|u''\|_{L_2(0,1)},$$

$$\|u' - (\mathcal{I}u)'\|_{L_2(0,1)} \leq (\frac{h}{\pi}) \|u''\|_{L_2(0,1)}.$$

See [29]. Then, the following theorem gives a posteriori error bounds in L_2-norm.

Theorem 3.2. *Given that $u \in H^2(0,1)$, the following a posteriori error bound holds*

$$\|u - u_h\|_{L_2(0,1)} = C\|h^2 R(u_h)\|_{L_2(0,1)}, \tag{3.7}$$

where $C = C_s C_{i,2}$, $C_s = 1 + 1/\sqrt{2}\max_{x\in[0,1]}|b(x)| + 1/2\max_{x\in[0,1]}|b'|$ and $C_{i,2} = \frac{1}{\pi^2}$.

It should be noted that since u_h can be computed by finite element approximation, the error bound in (3.7) is fully computable. The following proof of the theorem had been implemented from the lecture notes [7].

Proof: It is necessary to introduce an adjoint (or dual) problem for the proof of the theorem. Hence, consider the following auxiliary boundary value problem: Find z such that

$$-z'' - (b(x)z)' = (u - u_h)(x), \qquad x \in (0,1) \tag{3.8}$$

$$z(0) = z(1) = 0. \tag{3.9}$$

Now, if we use the definition of (3.8) and apply integration by parts, the following

expression can easily be written:

$$\begin{aligned}\|u-u_h\|^2_{L_2(0,1)} &= (u-u_h, u-u_h) = (u-u_h, -z''-(bz)') \\ &= \int_0^1 (u-u_h)(-z''-(bz)')dx \\ &= \int_0^1 ((u-u_h)'z' + b(u-u_h)' - u_h)z)dx \\ &= a(u-u_h, z).\end{aligned} \tag{3.10}$$

It follows that

$$\|u-u_h\|^2_{L_2(0,1)} = a(u-u_h, z). \tag{3.11}$$

Before continuing the proof, we state the Galerkin orthogonality property as follows.

Definition 9. (**Galerkin orthogonality**.) Let $a(\cdot,\cdot)$ be bilinear form and satisfy the Lax-Milligram theorem. Also, suppose that u ∈V and $u_h \in S_h \subset V$ are the solutions of the variational problem. Then:

$$a(u-u_h, v) = 0 \quad \forall v \in S_h,$$

[31, p.55]. Therefore, by the Galerkin orthogonality property, the following equation can be written:

$$a(u-u_h, z_h) = 0 \quad \forall z_h \in V_h.$$

Also, if we use the interpolation estimate for z_h, *i.e* $z_h = Iz \ \in V_h$, the equality

(3.11) turns into:

$$\begin{aligned} \|u - u_h\|^2_{L_2(0,1)} &= a(u - u_h, z - Iz) \\ &= a(u, z - Iz) - a(u_h, z - Iz) \\ &= \ell(z - Iz) - a(u_h, z - Iz) \\ &= (f - (z - Iz)) - a(u_h, z - Iz). \end{aligned} \tag{3.12}$$

It can be seen from the last line that the right-hand side no longer involves the exact solution, u. Now:

$$\begin{aligned} a(u_h, z - Iz) = &\sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} u_h'(z - Iz)'(x)dx \\ &+ \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} b(x)u_h'(z - Iz)(x)dx. \end{aligned} \tag{3.13}$$

Then applying integration by parts in each of the (N-1) integrals in the first sum on (3.13) and putting $(z - Iz)(x_i) = 0$ for i=0,...,N gives:

$$a(u_h, z - Iz) = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} [-u_h'' + b(x)u_h'](z - Iz)(x)dx.$$

Hence, if the Cauchy-Schwarz inequality is applied to the previous equation, the

following equation can be deduced:

$$\begin{aligned}\|u-u_h\|^2_{L_2(0,1)} &= \sum_{i=1}^{N}\int_{x_{i-1}}^{x_i} h_i^2 R(U_h)(x)h_i^{-2}(z-Iz)(x)dx \\ &\leq \sum_{i=1}^{N}\int_{x_{i-1}}^{x_i} \|h_i^2 R(U_h)\|_{L_2(x_{i-1},x_i)}\|h_i^{-2}(z-Iz)\|_{L_2(x_{i-1},x_i)} \\ &\leq \|h_i^2 R(U_h)\|_{L_2(0,1)}\|h_i^{-2}(z-Iz)\|_{L_2(0,1)}.\end{aligned} \tag{3.14}$$

Also, it can be written from the interpolation error bound:

$$\|z-Iz\|_{L_2(x_{i-1},x_i)} \leq (\frac{h_i}{\pi})^2\|z''\|_{L_2(x_{i-1},x_i)}, \quad i=1,...,N. \tag{3.15}$$

Thereby, combining the results from (3.14) and (3.15) gives:

$$\|h_i^{-2}(z-Iz)\|_{L_2(0,1)} \leq C_{i,2}\|z''\|_{L_2(0,1)},$$

where $C_{i,2} = \frac{1}{\pi^2}$. Therefore:

$$\|u-u_h\|^2_{L_2(0,1)} \leq C_{i,2}\|h^2 R(U_h)\|_{L_2(0,1)}\|z''\|_{L_2(0,1)}. \tag{3.16}$$

Now, we need to eliminate z in (3.16) to get the a posteriori error bound. From (3.8) the next equation can be written

$$z'' = u_h - u - (bz)' + (cz) = u_h - u - bz' + (c-b')z.$$

Then, taking the $L_2(0,1)$ norm of both sides of that equation and applying the triangle inequality gives:

$$\begin{aligned}\|z''\|_{L_2(0,1)} \leq \|u-u_h\|_{L_2(0,1)} + \max_{x\in[0,1]}|b(x)|\|z'\|_{L_2(0,1)} \\ + \max|c-b'|\|z\|_{L_2(0,1)}\end{aligned} \tag{3.17}$$

Next, we need to find bounds for $\|z'\|_{L_2(0,1)}$ and $\|z\|_{L_2(0,1)}$ in terms of $\|u-u_h\|_{L_2(0,1)}$. To do this, multiply the adjoint problem by z and integrate:

$$(-z''-(bz)', z) = (u-u_h, z).$$

Performing integration by part with the implementing boundary conditions of (3.8) yields:

$$(-z''-(bz)'+cz, z) = (z', z') + (bz, z') + (cz, z)$$

$$= \|z'\|^2_{L_2(0,1)} + \frac{1}{2}\int_0^1 b(x)[z^2(x)]'dx$$

$$= \|z'\|^2_{L_2(0,1)} - \frac{1}{2}\int_0^1 b'(x)[z^2(x)]dx.$$

Thereby:

$$(u-u_h, z) = \int_0^1 (-\frac{1}{2}b'(x))[z(x)^2]dx.$$

Also, we know from the assumptions of our model problem that

$$-\frac{1}{2}b^{'}(x) \geq 0, \quad x \in (0,1).$$

Hence the following bound can be written for $\|z'\|^2_{L_2(0,1)}$:

$$\|z'\|^2_{L_2(0,1)} \leq (u-u_h, z) \leq \|u-u_h\|_{L_2(0,1)}\|z\|_{L_2(0,1)}.$$

Now, The Poincaré-Friedrichs inequality states that

$$\|z\|^2_{L_2(0,1)} \leq \frac{1}{2}\|z'\|^2_{L_2(0,1)}.$$

Hence, combining the last two inequalities gives:

$$\|z\|_{L_2(0,1)} \leq \frac{1}{2}\|u-u_h\|_{L_2(0,1)}. \tag{3.18}$$

In a similar way, it can be shown that:

$$\|z'\|_{L_2(0,1)} \leq \frac{1}{\sqrt{2}}\|u-u_h\|_{L_2(0,1)}. \tag{3.19}$$

Hence, if (3.18) and (3.19) are substituted into (3.17), then the following bound for z'' can be written as:

$$\|z''\|_{L_2(0,1)} \leq \|u-u_h\|_{L_2(0,1)} + \frac{1}{\sqrt{2}}\max_{x\in[0,1]}|b(x)|\|u-u_h\|_{L_2(0,1)}$$

$$+ \max |c - b'| \|u - u_h\|_{L_2(0,1)}. \tag{3.20}$$

$$\equiv C_s \|u - u_h\|_{L_2(0,1)}$$

This result is known as the strong stability of the adjoint problem. Finally, substituting this result into (3.16) gives the desired fully computable a posteriori error bound and completes the proof:

$$\|u - u_h\|_{L_2(0,1)} \leq C \|h^2 R(U_h)\|_{L_2(0,1)}, \tag{3.21}$$

where $C = C_s C_{i,2}$.

3.5 Adaptive refinement for the one dimensional convection-diffusion problem with a priori error bounds

A uniform mesh refinement method gives an accurate solution i.e. it gives an acceptable error bound but it requires a huge computational effort. However, if the local errors in each subinterval of the mesh are identified and sorted from the biggest error to the smallest one, the mesh can be made finer just on the specified number of subintervals. In our code this number is 25% percent of all elements in a mesh, in which the local errors are biggest. This method is known as adaptive mesh or h refinement with fixed fraction method [7]. Adaptive mesh refinement requires less computational effort than making the mesh finer everywhere and

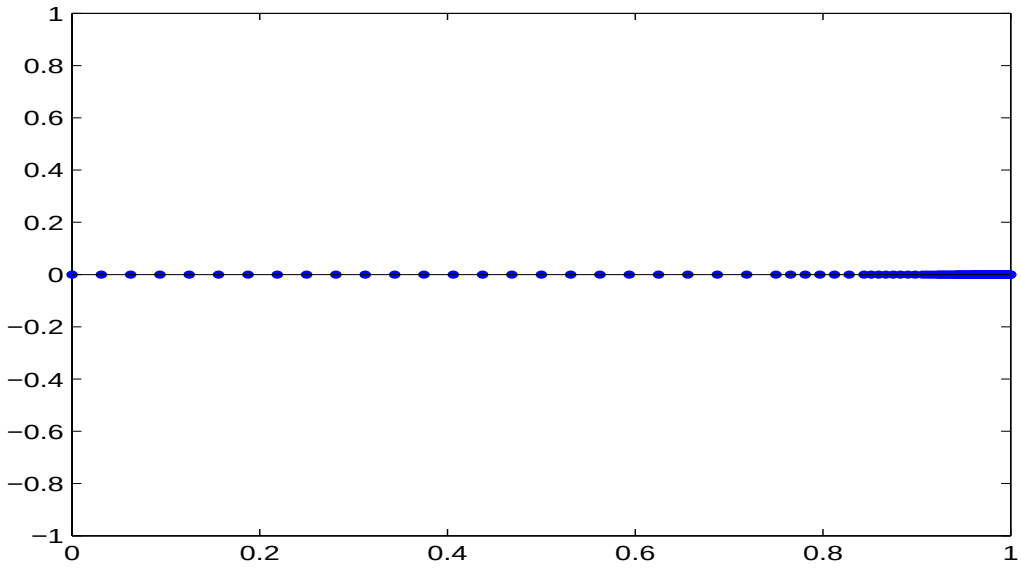

Figure 3: The refined mesh for example 1

gives a more accurate error estimate.

One important point for the convection-diffusion models is that they often lead to boundary/interior layers. Since the solution adjustment is too sharp in boundary layers, we put some numerical results for our first problem to illustrate effectivity of the adaptive mesh refinement at the end of this section. An adaptive refinement algorithm which was developed by using the APTOFEM software toolkit [27] was used to compute the numerical results. Since it is hard to find an exact solution to most partial differential equations, a priori error estimates may be not found in general, but for our first problem we used a priori error bounds in computations to compare exact error with the error found after applying the adaptive mesh refinement.

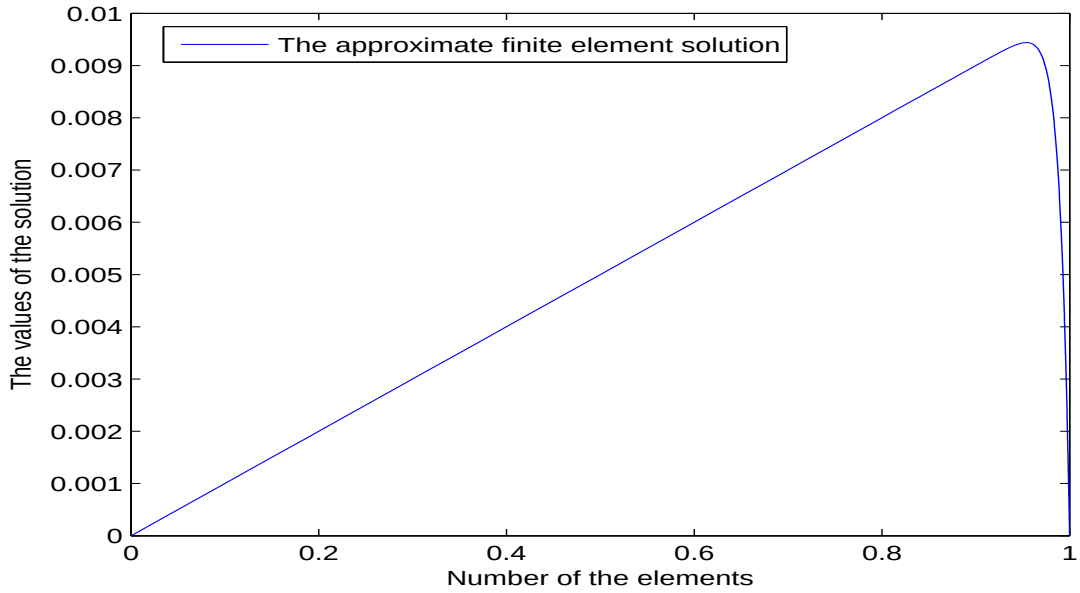

Figure 4: The approximate solution for example 1

Example 1: Consider the convection-diffusion boundary value problem by taking b(x)=100 and f(x)=1, $\forall x \in (0,1)$:

$$\begin{aligned} -u'' + 100 \cdot u' &= 1, \qquad x \in (0,1), \\ u(0) = 0 \quad &\text{and} \quad u(1) = 0. \end{aligned} \tag{3.22}$$

The refined mesh and a typical finite element solution is illustrated in Figure 3 and Figure 4. The number of elements in the refined mesh is 233. Clearly, refinement of the mesh increases near the end point, i.e. where a boundary layer starts to occur. Figure 4 indicates the sharp decrease of the solution on the boundary layer and this increases the error around this part of the domain. Therefore, one usually expects more refined elements around the boundary layer and Figure 3 represents

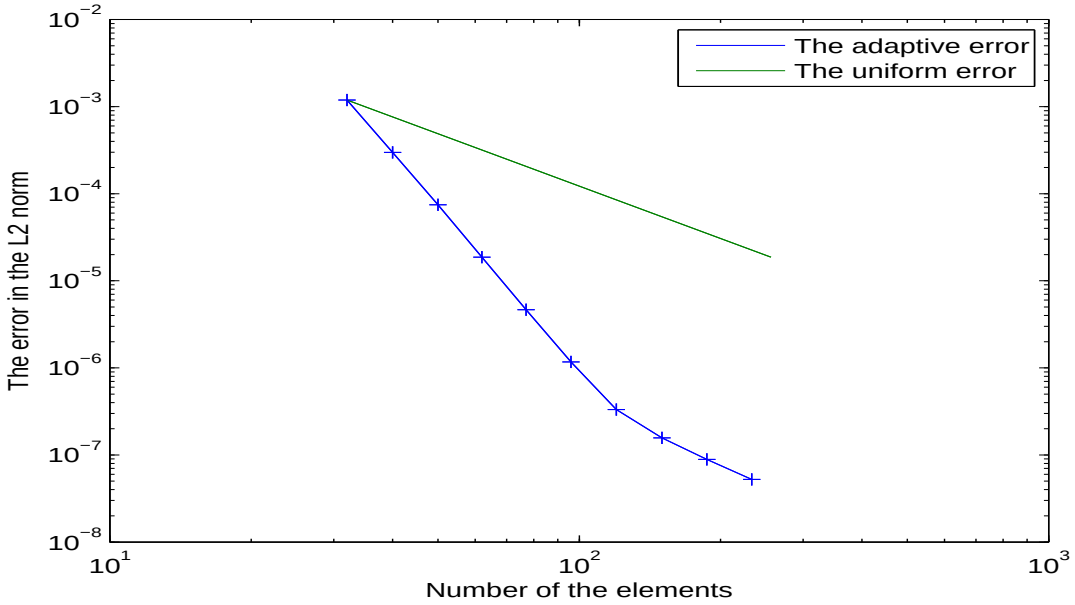

Figure 5: Performance of the adaptive refinement algorithm for example 1 with TOL=1x10^{-8}

Mesh Number	Number of elements	The approximated error	Effectivity
1	16	0.4777E-2	4.595
2	20	0.1194E-2	3.166
3	25	0.2985E-3	2.670
4	31	0.7464E-4	2.529
5	38	0.1866E-4	2.493
6	47	0.4704E-5	2.504
7	58	0.1409E-5	2.758
8	72	0.7140E-6	3.067
9	90	0.3876E-6	2.921
10	112	0.2318E-6	3.039

Table 1: Numerical results for example 1

the precise increase of the refined elements around this part as expected. Also, Figure 5 states the performance of the adaptive refinement algorithm against the uniform estimated error. The markers in Figure 5 state the number of elements after each adaptive refinement process. Clearly, a more accurate error estimate can be computed in the adaptive mesh refinement method with fewer elements than would be required for uniform mesh refinement to compute the same error estimate.

Table 1 displays some numerical results for example 1. As can clearly be seen, the effectivity index is constant for each mesh. Therefore, we can say that our method is useful. However, the effectivity index does not converge to 1 which implies that usual a posteriori error estimation is not as powerful as desired.

Consequently, although the adaptive mesh refinement method achieves a better error bound than the uniform mesh refinement method with less computational effort and with a more accurate error estimate, the effectivity index does not converge to 1. Therefore, we will introduce a new method for the a posteriori error estimation in the following section.

3.6 The dual weighted residual method. (DWR)

In many applications usual a posteriori error bounds are not useful for the purposes of real physical interests such as the lift and drag of an object in a viscous flow or the local average of a vector field [32]. Hence, a more powerful a posteriori error estimation can be obtained by using duality arguments for the

numerical solutions of the desired functional [14].

To proceed in this method, suppose we have a linear functional

$$J(\cdot) : \mathbf{H}^1(\Omega) \to \mathbb{R}.$$

Now, our aim is to find the value of $J(u)$ but we just know the value of $J(u_h)$, where u and u_h are exact and approximate solutions, respectively. However, we can try to find the value of $|J(u) - J(u_h)|$. The general weak formulation for the convection-diffusion model is:

$$a(u, v) = \ell(v) \quad \forall\ v \in H_0^1(\Omega). \tag{3.23}$$

Also, the finite element formulation is:

$$a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h, \tag{3.24}$$

where $a(\cdot, \cdot)$ and $\ell(\cdot)$ are as stated in (3.4). Since $J(\cdot)$ is linear:

$$J(u) - J(u_h) = J(u - u_h).$$

Now, we can introduce a dual problem as follows:

Find $z \in H_0^1$ such that

$$a(v, z) = J(v) \quad \forall v \in H_0^1. \tag{3.25}$$

Then, putting $u - u_h$ instead of v gives:

$$\begin{aligned} J(u - u_h) &= a(u - u_h, z) \\ &= a(u, z) - a(u_h, z) \\ &= \ell(z) - a(u_h, z). \end{aligned}$$

However, finding z is clearly as hard a problem as finding u, hence, if we approximate z with z_h,then

$$J(u - u_h) \approx \ell(z_h) - a(u_h, z_h),$$

but if $z_h \in V_h$, i.e. if it is in the same space as u_h, then Galerkin orthogonality implies:

$$\ell(z_h) - a(u_h, z_h) = 0.$$

Hence, we approximate z_h from another space, typically using quadratic finite element methods. Thereby:

$$z_h = \sum_{i=1}^{n-1} z_i \phi_i + \sum_{i=n}^{2n-1} z_i \phi_i,$$

where the basis functions in the first summation are linear and the basis functions in the second summation are quadratic. Also, 2n-1 is the degrees of freedom.

Hence, we deduce that:

$$|J(u) - J(u_h)| \approx |\ell(z_h) - a(u_h, z_h)|$$

[32]. Now, let us consider our second convection-diffusion example to show that our code satisfies this method.

Example 2:

$$\begin{aligned} -u'' &= tanh(20(x-0.5)) + 1, \qquad x \in (0,1), \\ u(0) &= 0 \quad \text{and} \quad u(1) = 2, \end{aligned} \tag{3.26}$$

and we try to find the value of the functional:

$$J(u) = \int_0^1 (tanh(20(x-0.5)) + 1) \cdot u,$$

which is a weighted average u, where u is the solution of (3.26).

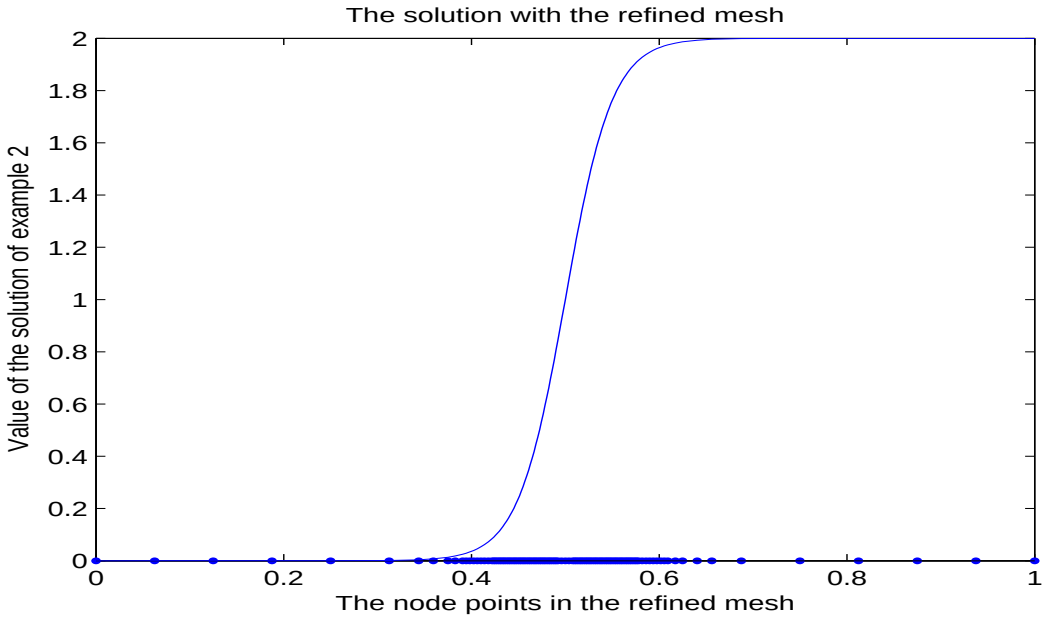

Figure 6: The solution of example 2 with the refined mesh

We computed the solution with the dual weighted residual method by using APTOFEM [27]. Then, we state the following figures to represent some properties of the solution of (3.26). Figure 6 shows the graph of the solution with the refined mesh. The number of elements in this mesh is 233. As seen in this figure, the solution has an interior layer, i.e. there exist sharp changes in the solution so that the refined elements were clustered around the interior layer as expected.

Figure 7 represents a comparison for the value of $|J(u) - J(u_h)|$ on both uniformly and adaptively refined meshes. Clearly, the adaptive refinement method achieves smaller errors in fewer elements than the uniform refinement method, i.e. it converges more rapidly than the uniform refinement method. Finally, we inserted the Figure 8 to illustrate the graph of the dual solution of the linearised

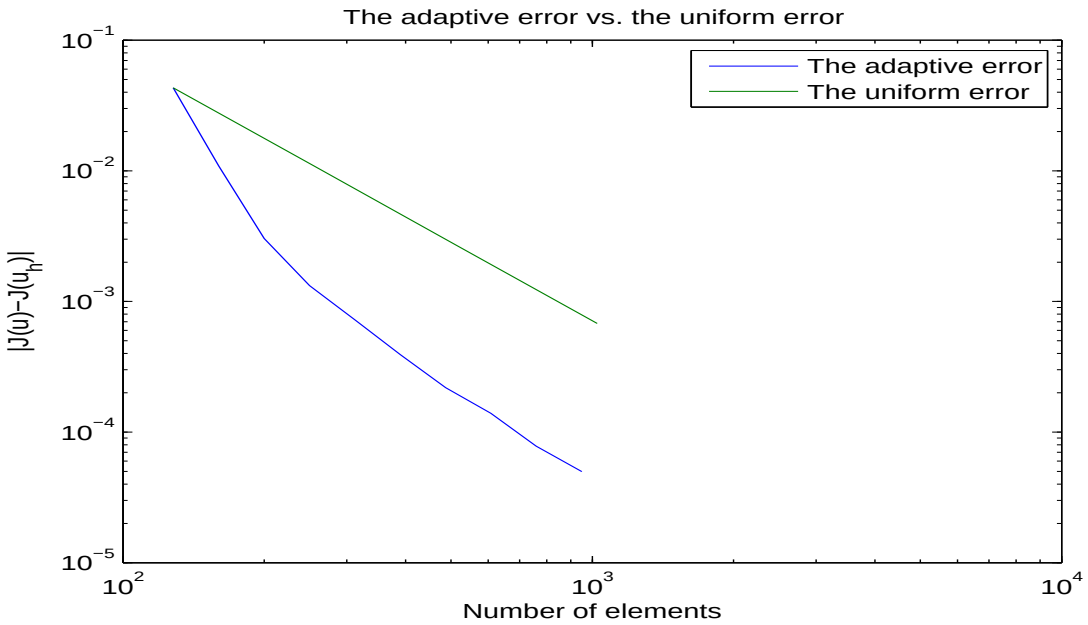

Figure 7: A comparison of the adaptive error and uniform error for example 2

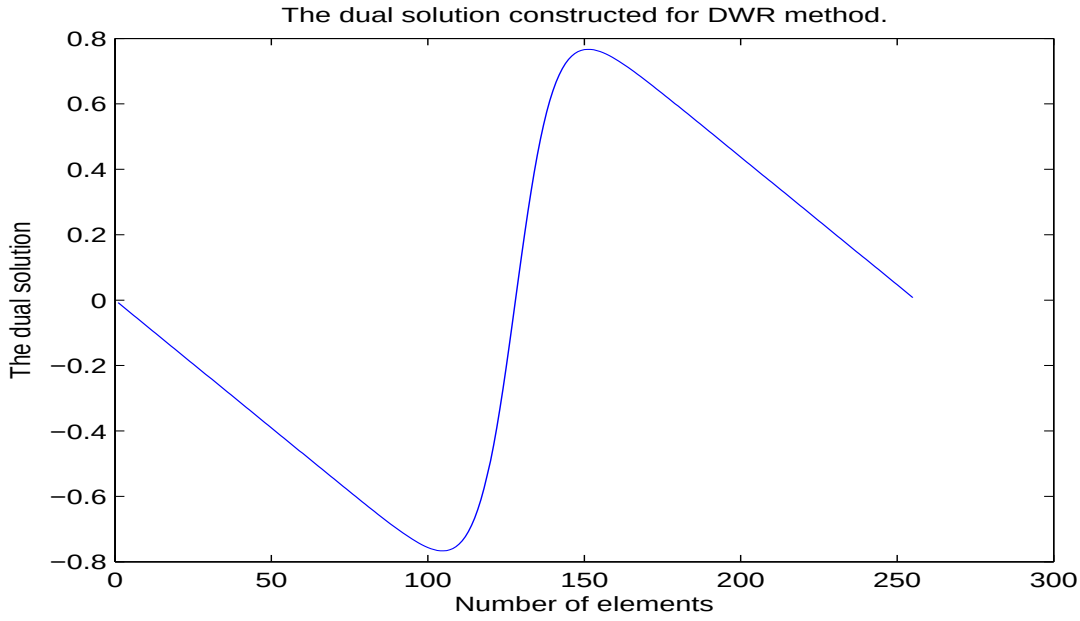

Figure 8: The solution of the dual problem for example 2

Mesh Number	Number of elements	The approximated error	Effectivity
1	32	0.6714	0.982
2	40	0.1725	0.995
3	50	0.4965E-1	0.997
4	62	0.2222E-1	0.996
5	77	0.1194E-1	0.996
6	96	0.6656E-2	0.995
7	120	0.3629E-2	0.998
8	150	0.2317E-2	0.998
9	187	0.1313E-2	0.998
10	233	0.8234E-3	0.998

Table 2: Numerical results for example 2

dual problem which were constructed to use the DWR method for the example 2.

Table 2 exhibits some numerical results for example 2. It should be noted that the refined mesh has the same number of elements in both examples. As can be seen from this table, the effectivity index for the DWR method converges to 1, whereas for the previous example it is constant but does not converge to 1. Therefore, we can conclude that a posteriori error estimation with the dual weighted residual is a more robust tool to compute a more accurate approximate solution.

4 Finite element methods for Stokes equations

In this section, we will focus on finding approximate solutions for the Stokes equations using the adaptive finite element methods. To do this, we will firstly consider the constructions of weak formulations, finite element spaces for both velocity and pressure variables and the finite element approximation methods in detail. Also, we will describe a priori error estimation theory to understand the accuracy of the approximate solutions. After than, a numerical example will be satisfied and the results computed in APTOFEM with uniform mesh refinement method will be given to show that results of our code match with the theory.

4.1 Stokes equations

The equation system:

$$\begin{aligned} -\nabla^2 \vec{u} + \nabla p &= \vec{0}, \qquad \underline{x} \in \Omega \subset \mathbb{R}^d, \quad d = 1, 2. \\ \nabla \cdot \vec{u} &= 0, \end{aligned} \tag{4.1}$$

is known as Stokes equation system, and it is a fundamental model of an incompressible fluid flow. In this equation system, the vector valued function $\vec{u} = (u_1, u_2)$ denotes the velocity of the fluid where u_1 and u_2 denote x-direction and y-direction components of the function of $\vec{u}$, respectively. Also, the scalar variable p denotes the pressure of the fluid. The first equation of (4.1) is called the momentum equation and it formulates conservation of the momentum of the

incompressible fluid. The second equation in (4.1) formulates mass conservation and it is also called *the incompressibility condition*. We will assume in this model that the velocity of the flow is so slow that we can ignore any effects of convection. In addition, we can state the boundary value problem of (4.1) with the following boundary conditions on $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ such that:

$$\vec{u} = \vec{w} \text{ on } \partial\Omega_D \text{ and } \frac{\partial\vec{u}}{\partial n} - \vec{n} \cdot p = \vec{s} \; on \; \partial\Omega_N \tag{4.2}$$

where Ω denotes a two dimensional domain and $\vec{n}$ is outward normal to the boundary.

The Dirichlet boundary part of boundary Ω is the part on which the value of the velocity function $\vec{u}$ is known and it is denoted by $\partial\Omega_D$. Also, The Neumann boundary part of boundary Ω is the part on which the value of the derivative of the velocity function $\vec{u}$ is known and it is denoted by $\partial\Omega_N$. Furthermore, we need to assume that $\int_{\partial\Omega_D} \neq 0$ to guarantee the existence of the velocity solution.
Also, we can write the Dirichlet boundary as the union of the following boundary parts according to its relation with the normal component of the velocity $\vec{u}$.

$$\partial\Omega_+ = \{x \in \partial\Omega \mid \vec{\omega} \cdot \vec{n} > 0\}, \quad \text{the outflow boundary}$$

$$\partial\Omega_0 = \{x \in \partial\Omega \mid \vec{\omega} \cdot \vec{n} = 0\}, \quad \text{the characteristic boundary}$$

$$\partial\Omega_- = \{x \in \partial\Omega \mid \vec{\omega} \cdot \vec{n} < 0\}, \quad \text{the inflow boundary}$$

If the value of velocity is known on the whole boundary, i.e. if $\partial\Omega_D = \partial\Omega$, then the pressure solution of the problem (4.1)-(4.2) is only unique up to a constant.

Furthermore, if we integrate the incompressibility condition, the second equation of (4.1), over Ω by using the divergence theorem, then we get:

$$0 = \int_{\Omega} \nabla \cdot \vec{u} \, d\Omega = \int_{\partial\Omega} \vec{u} \cdot \vec{n} \, ds = \int_{\partial\Omega} \vec{\omega} \cdot \vec{n} \, ds = \int_{\partial\Omega_-} \vec{\omega} \cdot \vec{n} \, ds + \int_{\partial\Omega_+} \vec{\omega} \cdot \vec{n} \, ds. \quad (4.3)$$

In other words, the velocity solution needs to satisfy the condition (4.3) on the boundary. Simply, the volume of fluid entering the domain must be same as the volume of the fluid going out from the domain. This condition is also known as the mass conservation. Clearly, if $\vec{\omega} \cdot \vec{n} = 0$ everywhere in the domain, the condition (4.3) is satisfied automatically. Such problems are called the enclosed flow problem. However, if the flow is not enclosed and (4.3) is not satisfied then the solution of the problem (4.1)-(4.2) will not exist. On the other hand, this problem can be solved by replacing the Dirichlet outflow condition with the Neumann outflow condition in (4.2). If $\int_{\partial\Omega_N} \neq 0$, then $\vec{u} \cdot \vec{n}$ automatically changes itself and satisfies the mass conservation. In addition, the pressure solution will be unique in this case [28, p.214-215].

4.2 Weak formulation of Stokes equations

We can derive a weak formulation of Stokes equation system by multiplying (4.1) and (4.2) with smooth test functions $\vec{v}$ and q , respectively:

$$\vec{v} \cdot (-\nabla^2 \vec{u} + \nabla p) = 0, \quad (4.4)$$

$$q\nabla \cdot \vec{u} = 0. \tag{4.5}$$

Then integrating both of (4.4) and (4.5) over Ω gives:

$$\int_\Omega \vec{v} \cdot (-\nabla^2 \vec{u} + \nabla p)\ d\Omega = 0, \tag{4.6}$$

$$\int_\Omega q\nabla \cdot \vec{u}\ d\Omega = 0. \tag{4.7}$$

By using integration by parts we can write:

$$\begin{aligned} -\int_\Omega \vec{v} \cdot (\nabla^2 \vec{u})\ d\Omega &= \int_\Omega \nabla \vec{u} \cdot \nabla \vec{v}\ d\Omega - \int_\Omega \nabla \cdot (\nabla \vec{u} \cdot \vec{v})\ d\Omega \\ &= \int_\Omega \nabla \vec{u} \cdot \nabla \vec{v}\ d\Omega - \int_\Omega (\vec{n} \cdot \nabla \vec{u}) \cdot \vec{v}\ d\Omega. \end{aligned} \tag{4.8}$$

Also:

$$\int_\Omega \vec{v} \cdot \nabla p\ d\Omega = -\int_\Omega p\nabla \cdot \vec{v}\ d\Omega + \int_\Omega \nabla \cdot (p\vec{v})\ d\Omega$$

.

$$= -\int_\Omega p\nabla \cdot \vec{v}\ d\Omega + \int_\Omega p\vec{n} \cdot \vec{v}\ d\Omega. \tag{4.9}$$

Combining (4.6), (4.8) and (4.9) gives the following equation:

$$\int_\Omega \vec{v}\cdot(-\nabla^2\vec{u}+\nabla p)\ d\Omega = \int_\Omega \nabla\vec{u}\cdot\nabla\vec{v}\ d\Omega - \int_\Omega p\nabla\cdot\vec{u}\ d\Omega - \int_{\partial\Omega} (\frac{\partial\vec{u}}{\partial n} - \vec{n})\cdot\vec{v}\ ds, \tag{4.10}$$

for all $\vec{v}$ in a suitable space of test functions.

Now, we need to be careful to determinate the test spaces for both velocity

and pressure variables. Since there is no pressure variable in the incompressibility condition, we can not choose these test spaces independently from each other. However, the boundary terms in (4.10) match with the Neumann part of the boundary conditions of the Stokes equation system. Also, the second term of the right-hand side of (4.10) is a reflection of the left-hand side of (4.7). Therefore, we can define the following spaces for the velocity solution $\vec{u}$ and the test functions by using equations (4.7) and (4.10) together with the boundary condition (4.2):

$$\mathbf{H}^1_E := \{\vec{u} \in H^1(\Omega)^d \mid \vec{u} = \vec{w} \; on \; \partial\Omega_D\}, \tag{4.11}$$

$$\mathbf{H}^1_{E_0} := \{\vec{v} \in H^1(\Omega)^d \mid \vec{v} = \vec{0} \; on \; \partial\Omega_D\}, \tag{4.12}$$

where $d = 2$ denotes the spatial dimension. Also, we can use $L_2(\Omega)$ space for the pressure variable because there are no derivatives of the pressure variable on the right-hand side of (4.10). Finally, we can state the weak formulation as:

Find $\vec{u} \in \mathbf{H}^1_E$ and $p \in L_2(\Omega)$ such that

$$\int_\Omega \nabla\vec{u} \cdot \nabla\vec{v} - \int_\Omega p\nabla \cdot \vec{v} \;=\; \int_{\partial\Omega_N} \vec{s} \cdot \vec{v} \quad \text{for all } \vec{v} \in \mathbf{H}^1_{E_0}, \tag{4.13}$$

$$\int_\Omega q\nabla \cdot \vec{u} \;=\; 0 \quad \text{for all } q \in L_2(\Omega). \tag{4.14}$$

Clearly, any solution of (4.1) also satisfies the equations in the weak formulation. However, we need to prove that weak solutions are uniquely defined. (If $\partial\Omega =$

$\partial\Omega_D$, then the pressure solution needs to be unique up to the a constant). To prove it, we need to consider the homogeneous weak formulation such that:

Find $\vec{u} \in \mathbf{H}^1_{E_0}$ and $p \in L_2$ such that

$$\int_\Omega \nabla\vec{u} \cdot \nabla\vec{v} - \int_\Omega p\nabla \cdot \vec{v} \ = 0 \quad for\ all\ \vec{v} \in \mathbf{H}^1_{E_0}, \tag{4.15}$$

$$\int_\Omega q\nabla \cdot \vec{u} \ = \ 0 \quad for\ all\ q \in L_2(\Omega). \tag{4.16}$$

and just need to prove that $\vec{u} = \vec{0}$ and p=constant. To do this, let (u_1, p_1) and (u_2, p_2) be two solutions of (4.15) and (4.16), then we will try to show that $u_1 = u_2$ and $p_1 = p_2 +$ constant. Putting $\vec{v} = \vec{u}$ in (4.15) and q=p in (4.16) gives:

$$\int_\Omega \nabla\vec{u} \cdot \nabla\vec{u} = 0.$$

However we know that:

$$a(\vec{u}, \vec{u}) = \int_\Omega \nabla\vec{u} \cdot \nabla\vec{u} \geq \int_\Omega \|\nabla u\|^2,$$

i.e. the bilinear functional $a(\cdot, \cdot)$ is coercive over $\mathbf{H}^1_{E_0}$. Therefore, $\vec{u} = \vec{0}$ as desired. Now, we put $\vec{u} = \vec{0}$ in (4.15) to proof of the pressure condition and observe that:

$$\int_\Omega p\nabla \cdot \vec{v} \ = 0 \ for\ all\ \vec{v} \in \mathbf{H}^1_{E_0}.$$

Then, we need to show that:

$$\int_{\Omega} p\nabla \cdot \vec{v} \ = 0 \quad for\ all\ \vec{v} \in \mathbf{H}^1_{E_0}, \tag{4.17}$$

which implies that $p = constant$ for the enclosed flow case. This is called the solvability condition and we could show this condition by using the following *inf-sup* condition:

$$\inf_{q \neq constant} \sup_{\vec{v} \neq 0} \frac{|(q, \nabla \cdot \vec{v})|}{\|\vec{v}\|_{1,\Omega} \|q\|_{0,\Omega}} \geq \gamma > 0, \tag{4.18}$$

where $\|\vec{v}\|_{1,\Omega} = (\int_{\Omega} \vec{v} \cdot \vec{v} + \nabla \vec{v} \cdot \nabla \vec{v})^{1/2}$ is a norm for functions in $\mathbf{H}^1_{E_0}$, and $\|q\|_{0,\Omega} = \|q - (1/|\Omega|) \int_{\Omega} q\|$ is a quotient space norm. This condition implies that for any $q \neq constant \in L_2(\Omega)$, there exists $\vec{v_q} \in \mathbf{H}^1_{E_0}$ such that

$$\frac{(q, \nabla \cdot \vec{v_q})}{\|\vec{v_q}\|_{1,\Omega}} \geq \gamma \|q\|_{0,\Omega}.$$

If we choose q as pressure solution p, then the numerator in this expression is zero for all $\vec{v} \in \mathbf{H}^1_{E_0}$. Hence, $\|p\|_{0,\Omega} = 0$ from that inequality. But this is the same as p=$\overline{p}$, where $\overline{p} = (1/|\Omega|) \int_{\Omega} p$ as required. In all, this means that the *inf-sup* condition is a sufficient condition for the uniqueness of the pressure for enclosed flows. Hence, we need to be careful about satisfying the solvability condition in (4.17) when we construct finite element approximations of the weak formulation [28, p.222-224].

4.3 Finite element approximation for Stokes equations

Now, we can define a finite element approximation for Stokes equations by using finite dimensional spaces $\mathbf{X}_0^h \subset \mathbf{H}_{E_0}^1$ and $M^h \subset L_2(\Omega)$. Hence, given a velocity solution space $\mathbf{X}_E^h$, we can write the finite element approximation as:

Find $\vec{u_h} = (u_{1_h}, u_{2_h}) \in \mathbf{X}_E^h$ *and* $p_h \in M^h$ such that

$$\int_\Omega \nabla \vec{u_h} \cdot \nabla \vec{v_h} - \int_\Omega p_h \nabla \cdot \vec{v_h} = \int_{\partial\Omega_N} \vec{s_h} \cdot \vec{v_h} \quad \forall \vec{v_h} \in \mathbf{X}_0^h, \tag{4.19}$$

$$\int_\Omega q_h \nabla \cdot \vec{u_h} = 0 \quad \forall q_h \in M^h. \tag{4.20}$$

Also, we can introduce the velocity basis set from a standard space of scalar basis functions $\{\phi_j\}_{j=1}^n$, where n is the degree of freedom. Hence, if we say n_u=2n, then we can define the velocity basis set as:

$$\{\vec{\phi_j}\}_{j=1}^{2n} = \{(\phi_j, 0)^T, (0, \phi_j)^T\}_{j=1}^n$$

Then, we can write the following discretisations with the scalar basis functions $\{\phi_j\}$ and $\{\phi_j\}$ for velocity and a set of scalar basis functions $\{\psi_k\}$ such that:

$$u_{1_h} = \sum_{j=1}^{n_u} \mathbf{u}_{1_j} \phi_j + \sum_{j=n_u+1}^{n_u+n_\partial} \mathbf{u}_{1_j} \phi_j, \tag{4.21}$$

$$u_{2_h} = \sum_{j=1}^{n_u} \mathbf{u}_{2_j} \phi_j + \sum_{j=n_u+1}^{n_u+n_\partial} \mathbf{u}_{2_j} \phi_j, \tag{4.22}$$

$$p_h = \sum_{k=1}^{n_p} \mathbf{p}_k \psi_k \ . \tag{4.23}$$

where $\mathbf{u}_{1_j}, \mathbf{u}_{2_j}$ and $\mathbf{p}_k$ are coefficients that we need to determine and $\sum_{j=1}^{n_u} \mathbf{u}_{i_j} \vec{\phi_j} \in \mathbf{X}_0^h,$ for $i = 1, 2$. In addition, we interpolate boundary data on $\partial\Omega_D$ by using the coefficients $\mathbf{u}_{i_j} : j = n_u{+}1, ..., n_u{+}n_\partial$ and $i = 1, 2$. In those expressions, n_u and n_p denote the number of interior nodes for velocity and pressure variables, respectively. Also, n_∂ denotes the number of boundary nodes for velocity. Hence, we can write finite element approximation (4.19)-(4.20) as a linear system of equations

$$\begin{pmatrix} A & 0 & B_1^T \\ 0 & A & B_2^T \\ B_1 & B_2 & 0 \end{pmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_1 \\ \mathbf{g} \end{bmatrix} . \tag{4.24}$$

The matrix A is an $n \times n$ scalar Laplacian matrix (Section 1.6 of [28]) and the matrices B_1 and B_2 are weak derivatives in the x and y directions, respectively. We can write their entries as:

$$A = [a_{mj}], \quad a_{mj} = \int_\partial \nabla\phi_m \cdot \nabla\phi_j, \tag{4.25}$$

$$B_1 = [b_{1,km}], \quad b_{1,km} = -\int_\Omega \psi_k \frac{\partial\phi_m}{\partial x}, \tag{4.26}$$

$$B_2 = [b_{2,kj}], \quad b_{2,kj} = -\int_\Omega \psi_k \frac{\partial \phi_j}{\partial y}, \tag{4.27}$$

for m and $j = 1, ..., n_u$ and $k = 1, ..., n_p$. Also, the entries of the right-hand side vector are given by:

$$\begin{aligned}
\mathbf{f}_1 = [\boldsymbol{f}_{1_m}], \quad \boldsymbol{f}_{1_m} &= \int_{\partial\Omega_N} s_1 \phi_m \; - \sum_{j=n_u+1}^{n_u+n_\partial} \mathbf{u_{1_j}} \int_\Omega \nabla\phi_m \cdot \nabla\phi_j, \\
\mathbf{f}_2 = [\boldsymbol{f}_{2_m}], \quad \boldsymbol{f}_{2_m} &= \int_{\partial\Omega_N} s_2 \phi_m \; - \sum_{j=n_u+1}^{n_u+n_\partial} \mathbf{u_{2_j}} \int_\Omega \nabla\phi_m \cdot \nabla\phi_j, \\
\mathbf{g} = [\boldsymbol{g}_k], \quad \boldsymbol{g}_k &= \sum_{j=n_u+1}^{n_u+n_\partial} \mathbf{u}_{1_j} \int_\Omega \psi_k \frac{\partial \phi_j}{\partial x} + \sum_{j=n_u+1}^{n_u+n_\partial} \mathbf{u}_{2_j} \int_\Omega \psi_k \frac{\partial \phi_j}{\partial y}.
\end{aligned} \tag{4.28}$$

The mixed finite element solution is the function pair $(\vec{u_h}, p_h)$ and the system (4.24)-(4.28) is also referred to as the discrete Stokes problem.

Now, we need to analyse the following homogeneous system to determine the uniqueness of the solution of the matrix system (4.24):

$$\begin{aligned}
\mathbf{A}\mathbf{u} + B^T\mathbf{p} &= 0, \\
B\mathbf{u} &= 0,
\end{aligned} \tag{4.29}$$

where $\mathbf{A}$ is a vector Laplacian matrix and B is a divergence matrix. Firstly, multiply the first and second equations by $\mathbf{u}^T = (\mathbf{u}_1, \mathbf{u}_2)^T$ and $\mathbf{p}^T$, respectively. Then, we see that $\mathbf{u}_1^T A\mathbf{u}_1 + \mathbf{u}_2^T A\mathbf{u}_2$=0. Thus, $\mathbf{u}$=0 because of the positive-definiteness of A. This implies the uniqueness of the velocity solution. If we put $\mathbf{u} = 0$ into (4.29), then we can observe $B^T p = 0$. This implies that the uniqueness of the

pressure solution is related to the null space of the matrix B^T. Hence, we need to satisfy that $\mathbf{null}(B^T) = \mathbf{1}$ for an enclosed flow and $\mathbf{null}(B^T) = \mathbf{0}$ in other cases by choosing the mixed approximation spaces. Hence, we can discretise the solvability condition for the pressure solution of the flow problems without a natural outflow boundary condition as:

$$\int_\Omega p_h \nabla \cdot \vec{v_h} = 0 \quad \forall\ \vec{v_h} \in \mathbf{X}_0^h \ \Rightarrow p_h = \text{constant}. \tag{4.30}$$

If we express p_h as in (4.23), then (4.30) turns into:

$$\langle B^T \mathbf{p}, \vec{\mathbf{v}} \rangle = 0 \quad \forall\ \vec{\mathbf{v}} \in \mathbf{R}^{n_u} \Rightarrow \mathbf{p} = \text{constant}, \tag{4.31}$$

and this is the same as the condition $\mathbf{null}(B^T) = \mathbf{1}$. Also, if the velocity is defined everywhere on the boundary, then $\mathbf{1} \in \mathbf{null}(B^T)$. This emphasises that the system (4.24) is singular so that there is a compatibility condition for the right-hand side vector in (4.24). Simply, we need to choose the values of velocity on the inflow and outflow boundaries to guarantee that the volumes of the fluid entering and leaving the domain are exactly the same.

The final question is to find the finite element approximation spaces for (4.1). The *inf-sup* condition (4.18) ensures that the solvability condition (4.30) is satisfied for any possible grids and it is also important to find a priori error bounds. However, natural choices of approximation spaces of velocity and pressure are not

stable. For instance, bilinear approximation, also called $\mathbf{Q}_1 - \mathbf{Q}_1$ approximation, for both velocity and pressure variables is unstable. In general, the dimension of the velocity space needs to be greater than dimension of the pressure space, otherwise the coefficient matrix in (4.24) does not have a full rank. Therefore, we will use $\mathbf{Q}_2 - \mathbf{Q}_1$ approximation in the rest of this section to have stable approximations [28, p. 224-229].

4.4 Error analysis for Stokes equations

The first step is the rearranging of the weak formulation of the Stokes problem. Hence, define the following continuous bilinear forms $a : \mathbf{H}^1 \; X \; \mathbf{H}^1 \to \mathbb{R},$ and $b : \mathbf{H}^1 \; X \; L_2(\Omega) \to \mathbb{R}$, such that:

$$a(\vec{u}, \vec{v}) := \int_\Omega \nabla \vec{u} \cdot \nabla \vec{v}, \quad b(\vec{v}, q) := -\int_\Omega q \nabla \cdot \vec{v}, \tag{4.32}$$

and the continuous functional $\ell : \mathbf{H}^1 \to \mathbb{R}$, such that:

$$\ell(\vec{v}) := \int_{\partial\Omega_N} \vec{s} \cdot \vec{v}. \tag{4.33}$$

Now, we can rewrite the weak formulation (4.13) and (4.14) as:

Find $\vec{u} \in \mathbf{H}^1_E$ and $p \in \mathrm{L}_2(\Omega)$ such that

$$\begin{aligned} a(\vec{u},\vec{v}) + b(\vec{v},p) &= \ell(\vec{v}) \quad \text{for all } \vec{v} \in \mathbf{H}^1_{E_0}, \\ b(\vec{u},q) &= 0 \quad \text{for all } q \in L_2(\Omega). \end{aligned} \tag{4.34}$$

Also, we can write the discrete problem (4.19)-(4.20) with a proper mixed approximation as:

Find $\vec{u_h} \in \mathbf{X}^h_E$ and $p_h \in M^h$ such that

$$\begin{aligned} a(\vec{u_h},\vec{v_h}) + b(\vec{v_h},p_h) &= \ell(\vec{v_h}) \quad \text{for all } \vec{v_h} \in \mathbf{H}^1_{E_0}, \\ b(\vec{u_h},q_h) &= 0 \quad \text{for all } q_h \in L_2(\Omega). \end{aligned} \tag{4.35}$$

The next task is finding the error bounds which are the L_2-norm for the velocity $\|\nabla(\vec{u} - \vec{u_h})\|$ and the quotient norm for pressure $\|p - p_h\|_{0,\Omega}$ defined earlier. We can also use the L_2 norm for the pressure variable of a flow problem which has a natural error bound [28, p.249-250].

4.4.1 A priori error bounds

Theorem 4.1. *If the problem (4.34) is solved on a grid* T_h *of rectangular elements which satisfies the aspect ratio condition, then there exists a constant* C_2 *such that:*

$$\|\nabla(\vec{u} - \vec{u_h})\| + \|p - p_h\|_{0,\Omega} \leq C_2 h^2 (\|D^3\vec{u}\| + \|D^2 p\|), \tag{4.36}$$

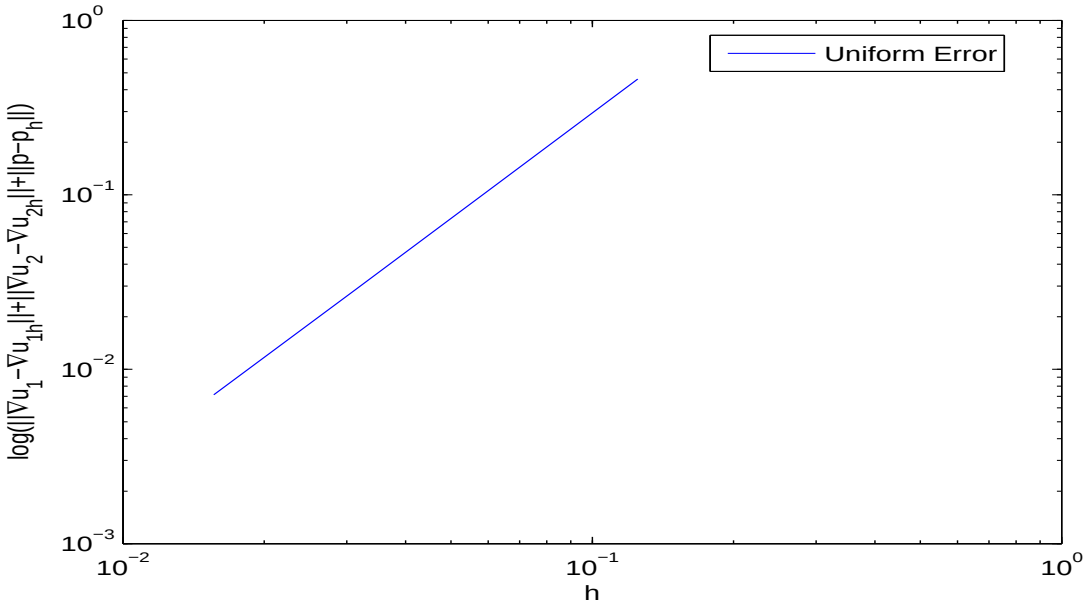

Figure 9: The error representation of example 3 w.r.t size of h on each grid

where $\|D^3\vec{u}\|$ *and* $\|D^2p\|$ *measures the* H^3 *regularity of the exact solution, and h is the length of the longest edge of* T_h*. [28, Theorem 5.6.]*

Example 3: This is a test example associated with the solution of Stokes equations:

$$u_1 = 20xy^3, \quad u_2 = 5x^4 - 5y^4, \quad p = 60x^2y - 20y^3$$

Our aim is to find a numerical approximation by solving this problem on a square grid to show that the error bound in (4.1) holds. Therefore, we computed the numerical solution on a square domain by refining the mesh uniformly on each grid with the software APTOFEM [27]. Figure 9 represents the approximate error w.r.t size of h on each grid as a straight line. If we perform a logarithmic

Mesh Number	h	The approximated error	order of convergence
1	1/8	0.4609	-
2	1/16	0.1143	2.01
3	1/32	0.2854E-1	2.00
4	1/64	0.7132E-2	2.00
5	1/128	0.1783E-2	2.00
6	1/256	0.4457E-3	2.00

Table 3: Numerical results for example 3

analysis on the numerical values exhibited in the Table 3, it gives:

$$\text{order of convergence} = \frac{log(error(i+1) - log(error(i)}{h(i+1) - h(i)} \approx 2.00, \quad \text{for } i = 1, 2, 3.$$

Since $log(error) = 2log(h) + constant$ according to (4.1), the numerical result of our code satisfies the theory.

5 Adaptive finite element methods for the Navier-Stokes equations

In this section, we will focus on finding approximate solutions for the Navier-Stokes equations with the finite element approximation method. Hence, we will consider constructions of weak formulation and the Newton iteration method for the solution in detail. Also, we will state the error theory related to the dual weighted residual method to understand accuracy of the approximate solutions. Finally, we will give an example of the Navier-Stokes equation with the numerical results computed in APTOFEM with adaptive mesh refinement method to show that our adaptive code is useful.

5.1 The Navier-Stokes equations

The equation system:

$$\begin{aligned} -\nu\nabla\cdot\vec{u}+\vec{u}\cdot\nabla u+\nabla p &= \vec{f}, \\ \nabla\cdot\vec{u} &= 0, \end{aligned} \tag{5.1}$$

where the constant $\nu > 0$ is called the *kinematic viscosity*. Also, this equation system is known as a steady state, i.e. not time dependent, Navier-Stokes equation. Here, the vector function $\vec{u}$ represents fluid velocity together with p, which represents fluid pressure. We can observe the Navier-Stokes equations from the

Stokes equations (4.1) by adding a convection term $\vec{u}\cdot\nabla u$ and a forcing function $\vec{f}$. On the other hand, the convection term in the first equation causes layers in the solution. We can state the boundary value problem of (5.1) with the boundary conditions on $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ such that:

$$\vec{u} = \vec{w} \;\; on \;\; \partial\Omega_D \;\; and \;\; \nu\frac{\partial\vec{u}}{\partial n} \; - \; \vec{n}\cdot p = \vec{0} \;\; on \;\; \partial\Omega_N \tag{5.2}$$

where Ω denotes a two or three dimensional domain and $\vec{n}$ is outward normal to the boundary.

We can normalise the system with respect to the size of the domain and the magnitude of the velocity to have a quantitative measure of the contributions of convection and viscous diffusion. To do this, let L represent a length scale of the domain and U represent the magnitude of velocity. If $\vec{x}$ denotes the points in the domain Ω then we can normalise it by dividing by length scale L. So, $\vec{\xi} = \vec{x}/L$ denotes the points in the normalised domain. Also, we can define $\vec{u}$ as $\vec{u} = U\vec{u_*}$. Then, the pressure can be normalised as $p(\vec{x}) = p(\vec{L\xi}) = U_2 p_*(\vec{\xi})$. Now, the first equation of (5.1) can be rearranged as:

$$-\frac{1}{\mathcal{R}}\nabla\cdot\vec{u_*} \; + \; \vec{u_*}\cdot\nabla\vec{u_*} \; + \; \nabla p_* \; = \; \frac{L}{U^2}\vec{f}. \tag{5.3}$$

Here, $\mathcal{R}$ is the Reynolds number and it is defined as:

$$\mathcal{R} := \frac{UL}{\nu}, \tag{5.4}$$

and the relative contributions of convection and diffusion are characterised by it. In other words, if we can choose suitable values of L and U, then the meaning of $\mathcal{R} \leq 1$ is that the equation (5.3) is diffusion-dominated and the velocity solution is uniquely defined whereas convection-dominated flows occur when $\mathcal{R} \geq 1$ [28, p.313- 314].

5.2 Weak formulation and linearisation of the Navier-Stokes equations

We can derive the weak formulation for the Navier-Stokes equations (5.1) and (5.2) as in the derivation of the weak formulation of the Stokes equations in Section 4.2:

Find $\vec{u} \in \mathbf{H}^1_E$ and $p \in L_2(\Omega)$ such that

$$\nu \int_\Omega \nabla\vec{u} \cdot \nabla\vec{v} + \int_\Omega (\vec{u} \cdot \nabla\vec{u}) \cdot \vec{v} - \int_\Omega p(\nabla \cdot \vec{v}) \;=\; \int_\Omega \vec{f} \cdot \vec{v} \quad \text{for all } \vec{v} \in \mathbf{H}^1_{E_0}, \tag{5.5}$$

$$\int_\Omega q(\nabla \cdot \vec{u}) \;=\; 0 \quad \text{for all } q \in L_2(\Omega), \tag{5.6}$$

where the test spaces $\mathbf{H}^1_E$ and $\mathbf{H}^1_{E_0}$ are defined in (4.11) and (4.12). However, the convection term $\int_\Omega (\vec{u}\cdot\nabla\vec{u})\cdot\vec{v}$ is the main difference between the formulations (5.5)-

(5.6) and (4.11)-(4.12). We can analyse this difference by defining the following trilinear form $c : H^1_{E_0}\ X\ H^1_{E_0}\ X\ H^1_{E_0} \to \mathbb{R}$:

$$c(\vec{z};\vec{u},\vec{v}) := \int_\Omega (\vec{z}\cdot\nabla\vec{u})\cdot\vec{v}. \tag{5.7}$$

The existence of the weak solution can be shown by using Lax-Milgram lemma. Let us define the subspace of velocity vectors whose divergences are zero in the domain Ω as:

$$\mathbf{V}_{E_0} := \{\vec{z}\in\mathbf{H}^1_{E_0} : \nabla\cdot\vec{z} = 0\ in\ \Omega\}, \tag{5.8}$$

and also define the bilinear functional $a_{\vec{z}}(\cdot,\cdot) : \mathbf{V}_{E_0}\ X\ \mathbf{V}_{E_0} \to \mathbb{R}$ as:

$$a_{\vec{z}}(\vec{u},\vec{v}) := \nu\int_\Omega \nabla\vec{u}\cdot\nabla\vec{v} + c(\vec{z};\vec{u},\vec{v}), \tag{5.9}$$

where the vector $\vec{z}$ belongs to $\mathbf{V}_{E_0}$. On the other hand, the convection terms are skew-symmetric, i.e. $c(\vec{z};\vec{u},\vec{v}) = -c(\vec{z};\vec{u},\vec{v})$ over $\mathbf{V}_{E_0}$. Hence, $c(\vec{z};\vec{u},\vec{v}) = 0$ so it follows:

$$a_{\vec{z}}(\vec{u},\vec{v}) := \nu\|\nabla\vec{u}\|^2 \quad \forall\vec{u}\in\mathbf{V}_{E_0}. \tag{5.10}$$

However, satisfying the continuity:

$$c(\vec{z};\vec{u},\vec{v}) \le \Gamma\|\nabla\vec{z}\|\|\nabla\vec{u}\|\|\nabla\vec{v}\| \tag{5.11}$$

is more complicated, for details see Girault & Raviart [33, p.284].

The analysis of the uniqueness of the solution of the weak formulations (5.5) and (5.6) is quite complicated because the Navier-Stokes equations are nonlinear. To identify this, let (u_1, p_1) and (u_2, p_2) be two distinct solutions and $\vec{u} := \vec{u_1} - \vec{u_2}$ and $\vec{p} := \vec{p_1} - \vec{p_2}$. Then:

$$c(\vec{u_1}; \vec{u_1}, \vec{v}) - c(\vec{u_2}; \vec{u_2}, \vec{v}) \neq c(\vec{u_1} - \vec{u_2}; \vec{u_1} - \vec{u_2}, \vec{v})$$

means that we cannot use the homogenous problem which is analogous to that used in (4.15) and (4.16) to get uniqueness. Instead, if we can define the norm for the forcing function $\vec{f}$ in the case of enclose flow, i.e. $\vec{w} = \vec{0}$ as:

$$\|\vec{f}\|_* := \sup_{\vec{v} \in \mathbf{V}_{E_0}} \frac{(\vec{f}, \vec{v})}{\|\nabla \vec{v}\|},$$

then:

$$\|\vec{f}\|_* \leq \frac{\nu^2}{\Gamma_*}, \tag{5.12}$$

is a sufficient condition for uniqueness, where Γ_* is a constant that satisfies the continuity in (5.11). This means that if ν is large enough, then there exists a unique weak solution [28, p.319-320].

5.3 A solution method for Navier-Stokes equations: The Newton iteration

We need to use a nonlinear iteration in which a linearised problem is solved in each step for solving the Navier-Stokes equations. Basically, the iteration starts with an initial guess $(\vec{u_0}, p_0) \in \mathbf{H}^1_E \times L_2(\Omega)$ and computes a sequence of iterates $(\vec{u_0}, p_0), (\vec{u_1}, p_1), (\vec{u_2}, p_2)... \in \mathbf{H}^1_E \times L_2(\Omega)$. The desired result is convergence of those iterates to the solution of the weak formulation. In this study we will use the Newton iteration which is a classical linearisation method although one can find many other methods.

The first step of the Newton iteration is computing the nonlinear residual associated with the weak formulation (5.5)-(5.6) for a given iterate $(\vec{u_k}, p_k)$. It is the following pair $R_k(\vec{v}), r_k(q)$

$$R_k = \int_\Omega \vec{f} \cdot \vec{v} - c(\vec{u_k}; \vec{u_k}, \vec{v}) - \nu \int_\Omega \nabla \vec{u_k} \cdot \nabla \vec{v} + \int_\Omega p_k (\nabla \cdot \vec{v}), \tag{5.13}$$

$$r_k = -\int_\Omega q(\nabla \cdot \vec{u_k}), \tag{5.14}$$

for any $\vec{v} \in \mathbf{H}^1_{E_0}$ and $L_2(\Omega)$. Then write the solution of (5.5)-(5.6) as $\vec{u} = \vec{u_k} + \delta\vec{u_k}$ and $p_k + \delta p_k$ where the perturbations $\delta\vec{u_k} \in \mathbf{H}^1_{E_0},\ \delta p_k \in L_2(\Omega)$. Substituting them

into the weak formulation gives:

$$D(\vec{u_k}, \delta\vec{u_k}, \vec{v}) + \nu \int_\Omega \nabla\delta\vec{u_k} \cdot \nabla\vec{v} - \int_\Omega \delta p_k(\nabla \cdot \vec{v}) = R_k(\vec{v}), \quad (5.15)$$

$$\int_\Omega q((\nabla \cdot \delta\vec{u_k}) = r_k(q), \quad (5.16)$$

for all $\vec{v} \in \mathbf{H}^1_{E_0}$ and $L_2(\Omega)$, where $D(\vec{u_k}, \delta\vec{u_k}, \vec{v}) = c(\delta\vec{u_k}; \delta\vec{u_k}, \vec{v}) + c(\delta\vec{u_k}; \vec{u_k}, \vec{v}) + c(\vec{u_k}; \delta\vec{u_k}, \vec{v})$. Finally, dropping quadratic term $c(\delta\vec{u_k}; \delta\vec{u_k}, \vec{v})$ from $D(\vec{u_k}, \delta\vec{u_k}, \vec{v})$ gives the linear problem:

for all $\vec{v} \in \mathbf{H}^1_{E_0}$ and $L_2(\Omega)$, find $\delta\vec{u_k} \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ satisfying:

$$\begin{aligned} c(\delta\vec{u_k}; \vec{u_k}, \vec{v}) + c(\vec{u_k}; \delta\vec{u_k}, \vec{v}) + \nu \int_\Omega \nabla\delta\vec{u_k} \cdot \nabla\vec{v} - \int_\Omega \delta p_k(\nabla \cdot \vec{v}) &= R_k(\vec{v}), \\ \int_\Omega q(\nabla \cdot \delta\vec{u_k}) &= r_k(q). \end{aligned} \quad (5.17)$$

The solution to this system is called the Newton correction. Now, we can define the next iterate by updating the current iterate as $\vec{u_{k+1}} = \vec{u_k} + \delta\vec{u_k}$, $p_{k+1} = p_k + \delta p_k$ [28, p.324-325].

5.4 Dual weighted residual error analysis for nonlinear equations

In this section, we will state a posteriori error bounds based on the dual weighted residual method for Navier-Stokes equations. Then, we will give a numerical example with the computed results to show that our numerical solution

satisfies this analysis. To continue, let $\underline{u} = (\vec{u}, p)$ and $\underline{v} = (\vec{v}, q)$. Then, we can rewrite our weak formulation (5.5)-(5.6) in the semilinear form

$$\mathcal{N}(\underline{u}, \underline{v}) = 0 \text{ in the domain } \Omega, \tag{5.18}$$

where $\Omega \subset \mathbb{R}^2$and:

$$\mathcal{N}(\underline{u}, \underline{v}) = \nu \int_\Omega \nabla\vec{u} \cdot \nabla\vec{v} + \int_\Omega (\vec{u} \cdot \nabla\vec{u}) \cdot \vec{v} - \int_\Omega p(\nabla \cdot \vec{v}) - \int_\Omega \vec{f} \cdot \vec{v} + \int_\Omega q(\nabla \cdot \vec{u}).$$

Also, set the spaces $V = \{H(\Omega)^1_D\}^2 \times L_2(\Omega)$, where $H(\Omega)^1_D = \{\vec{u} \in H(\Omega)^1 \; s.t \; \vec{u} = \vec{\omega} \; on \; \partial\Omega$ and $V_h = P_2^2 \times P_1$, where P_2 and P_1 are quadratic and bilinear test spaces with triangular elements, respectively. Also, the semilinear form $\mathcal{N}(\cdot, \cdot)$ is nonlinear in the first argument but linear in the second argument. Then, we can discretise (5.18) as follows:

find $\underline{u}_h \in V_h$ such that

$$\mathcal{N}(\underline{u}_h, \underline{v}_h) = 0 \quad \forall \underline{v}_h \in V_h. \tag{5.19}$$

Now, if we chose $\underline{v} = \underline{v}_h$ in (5.18) and subtract (5.19) from (5.18), then we obtain:

$$\mathcal{N}(\underline{u}, \underline{v}_h) - \mathcal{N}(\underline{u}_h, \underline{v}_h) = 0 \quad \forall \underline{v}_h \in V_h. \tag{5.20}$$

This property, (5.20), is also known as Galerkin orthogonality. Now, suppose we have a nonlinear and differentiable functional $J(\cdot)$ and define the mean-value

linearisation of it as

$$\hat{J}(\underline{u}, \underline{u}_h; \underline{u} - \underline{u}_h) = J(\underline{u}) - J(\underline{u}_h) = \int_0^1 J'[\theta\underline{u} + (1-\theta)\underline{u}_h](\underline{u} - \underline{u}_h)d\theta, \quad (5.21)$$

where $J'[w](\cdot)$ is the Gâteaux derivative of $J(\cdot)$ at the some point $w \in V$.

Similarly, we can define the mean-value linearization of $\mathcal{N}(\cdot, \underline{v})$ for $\underline{v} \in V$ as follows

$$M(\underline{u}, \underline{u}_h; \underline{u}-\underline{u}_h, \underline{v}) = \mathcal{N}(\underline{u}, \underline{v})-\mathcal{N}(\underline{u}_h, \underline{v}) = \int_0^1 \mathcal{N}'[\theta\underline{u}+(1-\theta)\underline{u}_h](\underline{u}-\underline{u}_h)d\theta, \quad (5.22)$$

where $\mathcal{N}'[w](\cdot, v)$ is the Gâteaux derivative of $\mathcal{N}$ at the some point $w \in V$.

Now, we can introduce the following dual problem:

find z in V such that

$$M(\underline{u}, \underline{u}_h; w, z) = J(\underline{u}, \underline{u}_h; w) \quad \forall w \in V. \quad (5.23)$$

If we put $\underline{u} - \underline{u}_h$ instead of w in (5.23) and apply Galerkin orthogonality (5.20) we get:

$$\begin{aligned} J(\underline{u}) - J(\underline{u}_h) &= \hat{J}(\underline{u}, \underline{u}_h; \underline{u} - \underline{u}_h) = M(\underline{u}, \underline{u}_h; \underline{u} - \underline{u}_h, z) \\ &= M(\underline{u}, \underline{u}_h; \underline{u} - \underline{u}_h, z - z_h) = -\mathcal{N}(\underline{u}_h, z - z_h) \quad \forall z_h \in V_h. \end{aligned} \quad (5.24)$$

However, the right hand side of (5.24) depends on z, which is the formal solution of (5.23), and z also depends on the solution u of (5.18). In other words, we do not know the value of z. Hence, let us say:

$$-\mathcal{N}(\underline{u}_h, z - z_h) = R(\underline{u}_h, z - z_h),$$

then we can write the error representation as:

$$J(\underline{u}) - J(\underline{u}_h) = R(\underline{u}_h, z - z_h). \tag{5.25}$$

Now, we need to introduce the linearised dual problem as:

find $\hat{z}$ in V such that

$$\mathcal{N}'[\underline{u}_h](w, \hat{z}) = J'[\underline{u}_h](w) \quad \forall w \in V. \tag{5.26}$$

then we approximate $\hat{z}$ by the discrete adjoint problem:

find $\hat{z}_h \in \hat{V}_h$ such that

$$\mathcal{N}'[\underline{u}_h](w_h, \hat{z}_h) = J'[\underline{u}_h](w_h) \quad \forall w_h \in \hat{V}_h, \tag{5.27}$$

where $\hat{V}_h$ is an adjoint finite element space. However, if $\hat{V}_h = V_h$, i.e. if we approximate $\hat{z}$ by using the same finite element method for the primal problem, then the error representation (5.25) will be zero. Therefore, we will approximate

$\hat{z}$ in the same space used for the primal problem but with a higher degree of polynomials. Then, we can obtain that:

$$\begin{aligned} J(\underline{u}) - J(\underline{u}_h) &= R(\underline{u}_h, z - z_h) \\ &= R(\underline{u}_h, z - \hat{z} + \hat{z} - z_h) \\ &= R(\underline{u}_h, z - \hat{z}) + R(\hat{z} - \hat{z}_h) + R(\underline{u}_h, \hat{z}_h - z_h) \end{aligned} \tag{5.28}$$

However, if $\underline{u} \approx \underline{u}_h$ then the first two terms of the last line are zero. These two terms are known as linear and discreatisation errors, respectively. Thereby:

$$J(\underline{u}) - J(\underline{u}_h) = R(\underline{u}_h, \hat{z}_h - z_h) \equiv \sum_{E \in T_h} \hat{\eta}_k, \tag{5.29}$$

where $|\hat{\eta}_k|$ are local error indicators and they include the primal local residuals weighted with the solution of the discrete adjoint problem. Therefore, they are called dual-weighted residual error indicators [34].

Now, we state the following Poiseuille flow example on an L-shaped domain to show that our numerical solution that was computed by APTOFEM [27] matches with the DWR theory.

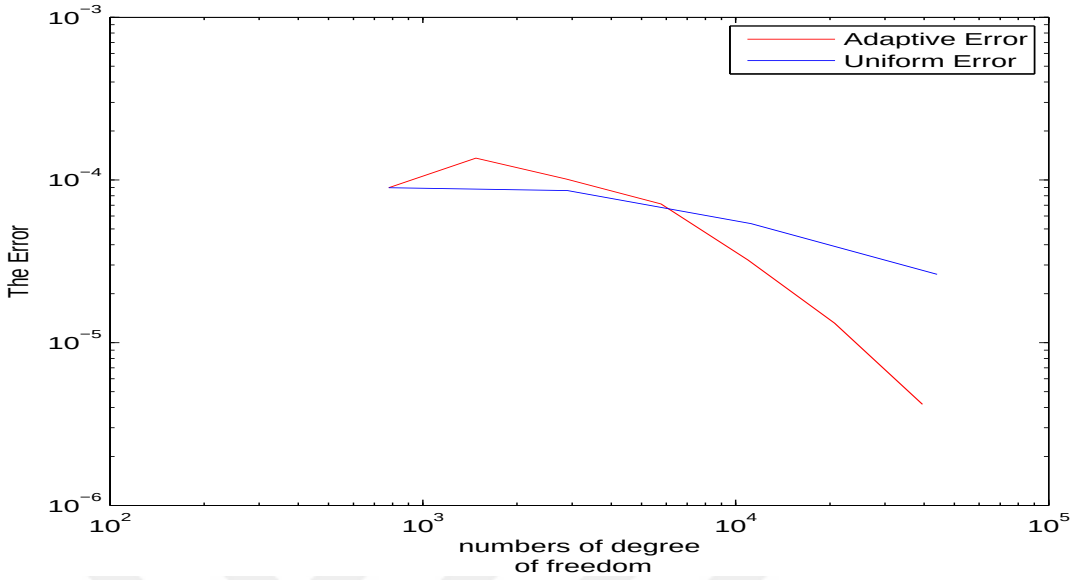

Figure 10: The adaptive error vs uniform error for example 4

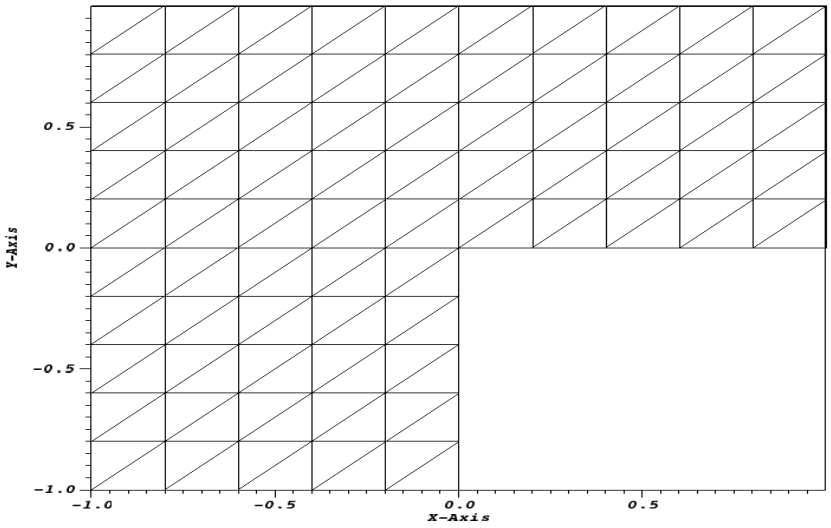

Figure 11: The initial mesh of example 4

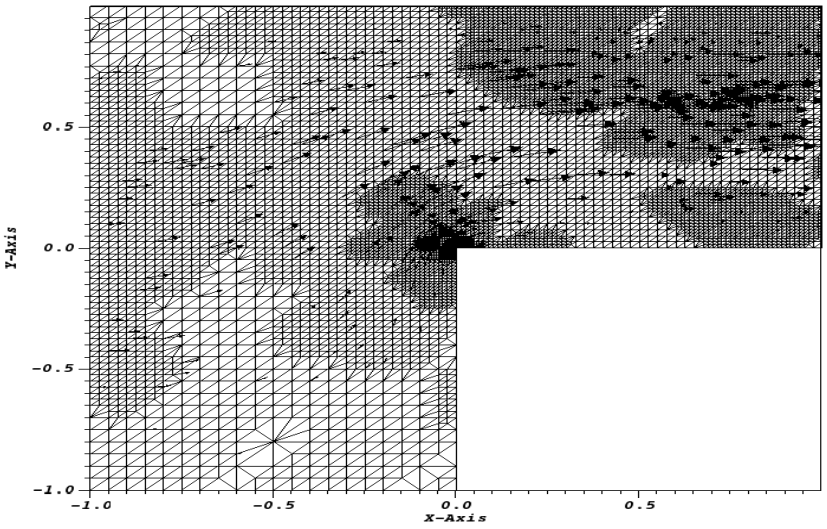

Figure 12: The vector graph of the solution of example 4

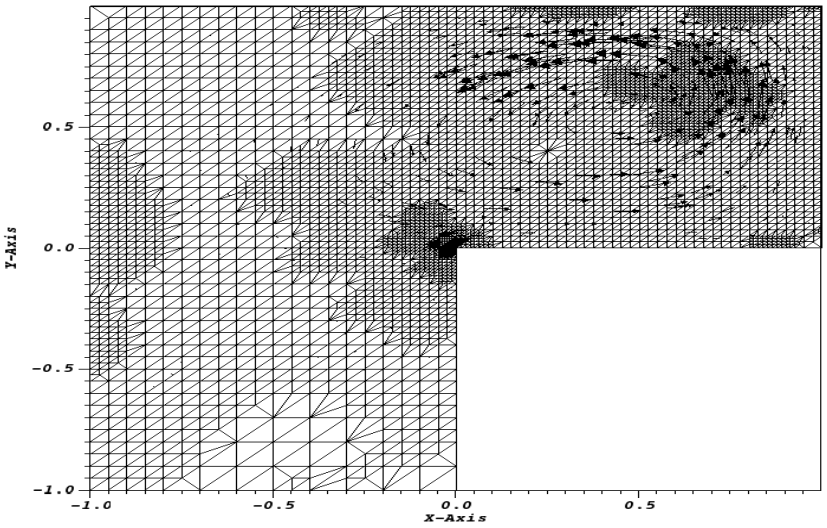

Figure 13: The vector graph of the solution of dual problem in example 4

Mesh Number	number of degrees of freedom	The approximated error	Effectivity
1	778	0.89740E-4	0.231
2	1478	0.1361E-3	0.512
3	2877	0.1011E-3	0.550
4	5769	0.7127E-4	0.624
5	10934	0.3222E-4	0.672
6	20690	0.1318E-4	0.776
7	39522	0.4181E-5	1.137

Table 4: Numerical results for example 4.

Example 4: Consider a fluid flow into a channel with a sudden contraction on the channel. We set $u_1 = -(y-1)(y+1)$ on the inflow boundary $(x = -1; -1 \leq y \leq 1)$ and set $u_1 = -8y(y-1)$ on the outflow boundary $(x = 1; -1 \leq y \leq 1)$ and zero otherwise. Also, $u_2 = 0$ and $p = 0$ in all the domain. Also, we impose no flow conditions on the upper and lower boundaries. Our aim is to compute the approximate value of the functional $J(\cdot) : \Omega \to \mathbb{R}$:

$$J(\underline{u}) = \int_{\Omega} \vec{w} \cdot \underline{u} d\Omega,$$

where $\underline{u} = (u_1, u_2, p)$ and $\vec{w} = (1, 0, 0)$ for $0.6 \leq x, y \leq 0.8$ and zero, otherwise.

Figure 11 represents the initial mesh with triangular elements. We used triangular elements for this problem because using rectangular elements produces hanging nodes which make the adaptive mesh refinement harder [7]. Figure 12 shows the vector field of the solution of the Navier-Stokes equation. We can clearly see that the refined elements were clustered around the corner, which usually causes singularity, in the domain as expected. The Figure 13 shows the

solution of the dual problem. Also, we can see from Table 3 that the adaptive error is reducing in each grid. Also, since the linear and discretisation errors are not sufficiently small on the first meshes, effectivity is too small on them. On the final mesh our approximation is close to the true solution so that effectivity is nearly 1 at the final grid. On the other hand, Figure 10 shows the comparison of adaptive and uniform errors. Although, uniform refinement is better in the first grids, adaptive refinement method produces small errors fewer computational steps than uniform refinement method at the end. So we can say that our code is useful for numerical approximations based on DWR methods.

6 Conclusion and Future Work

6.1 Conclusion

In this study, we have tried to review adaptive finite element methods for partial differential equations arising in incompressible fluid flow problems. To do this, we have satisfied the theoretical background of finite element methods for the one dimensional convection-diffusion equations and the two dimensional Stokes and Navier-Stokes equations. Also, we have tried to investigate the a posteriori error estimations for these three equations. We then used the software toolkit APTOFEM [27] to compute the finite element approximations with the adaptive mesh refinement method for these equations.

In the second section, we have just satisfied some general definitions and theorems which were used in the following sections of the study. Hence, the reader can understand the other sections clearly.

In the third section, we have computed a finite element approximation for two examples of one dimensional convection-diffusion problems by using the algorithm produced in APTOFEM and observed the effect of the layers that occurred in the solutions on the accuracy of the error. As a result, we observed that the number of refined elements had increased around the layers. On the other hand, we have observed that using the dual weighted residual method for a posteriori error estimation increases the accuracy of the solution, depending on the effectivity index . In all, the main result observed in that section is that we had more accurate

finite element approximations by refining elements adaptively than by refining them uniformly by providing the computational results. Thereby, we conclude that adaptive mesh refinement methods are cheaper in the cost of computational resources.

In the fourth section, we have studied the two dimensional steady-state Stokes equations. Firstly, we have given information about the theory of finite element methods such as the construction of weak formulation and finite element approximation for the Stokes equations. Hence, we observed that there is a solvability condition called the *inf-sup* condition which needs to be satisfied for the existence and uniqueness of the solution. At the end of the section, we have provided a priori error analysis and computed a finite element approximations by uniformly mesh refining with rectangular elements for an example of the Stokes equations. Finally, we have observed that the computational results had matched with the priori error analysis. Hence, we presented plots of the results.

In the fifth section, we have studied the two dimensional steady-state Navier-Stokes equations. Firstly, we have given information about the theory of construction of weak formulation for finite element method and the Newton iteration method to solve the equation iteratively. After than, we have stated the error analysis for the Navier-Stokes equations by using dual weighted residual method. Then, we observed from our numerical solutions produced in APTOFEM that using the dual weighted residual methods is very useful for reducing computational costs.

6.2 Future Work

Unfortunately, we could not analyse adaptive refinement techniques in detail. In this study, we were interested in a posteriori error estimation of some partial differential equations with adaptive mesh refinement methods, also known as h-refinement methods. However, we were also able to study hp-refinement methods. This method is based on increasing degrees of the polynomial approximations of the local elements of the domain in which the solution is smooth and refining the mesh with low order polynomials around the singularities [35]. This method is useful because it is convenient for adaptive control of error and it can achieve high-order accuracy for the smooth solutions. Also, the computational cost of the solution is acceptable [36].

Another approach that we could use is three dimensional adaptive methods. Most physical problems such as air flow turbulence problems and fluid flow problems, have singularities in local regions. Hence, the mesh needs to be extremely fine around the singularities to find a acceptable numerical solution. However, using uniform mesh refinement methods increase the cost of the computation significantly [37]. Therefore, adaptive methods need to be used to control the computational cost.

We were interested in steady state Navier-Stokes equations in Section 5. However, stability analysis of the time dependent Navier-Stokes equations is an important topic that we could explore. Briefly, we assume that the pair $\vec{u}(\vec{x})$ -

$p(\vec{x})$ is the solution of the weak formulation (5.5)-(5.6). This solution is known as a fixed point. The main aim of stability theory is to study disturbances (or perturbations) from such a fixed point. If all the perturbations decay to zero as time evolves, then the fixed point is unstable. Conversely, if perturbations grow as time evolves, then the fixed point is unstable.

To analyse stability, define a time-dependent solution such that:

$$\vec{v}(\vec{x},t) = \vec{u}(\vec{x}) + \delta\vec{u}(\vec{x},t), \quad q(\vec{x},t) = p(\vec{x}) + \delta p(\vec{x},t), \tag{6.1}$$

where $\delta\vec{u}(\vec{x},t)$ and $\delta p(\vec{x},t)$ are perturbations. Also, we assume that the boundary condition (5.2) holds for the velocity $\vec{v}(\vec{x},t)$ so that the perturbation $\delta\vec{u}(\cdot,t)$ is zero on the Dirichlet boundary part for all time. Now, if we substitute this solution into a time dependent version of the weak formulation then we can obtain the following initial-boundary value problem: Find $\vec{v}(\vec{x},t)$ and $\delta p(\vec{x},t)$ such that

$$\begin{aligned} \int_\Omega \frac{\partial\delta\vec{u}}{\partial t}\cdot\vec{v} + D(\vec{u},\delta\vec{u},\vec{v}) + \nu\int_\Omega \nabla\delta\vec{u}\cdot\nabla\vec{v} - \int_\Omega \delta p(\nabla\cdot\vec{v}) &= \vec{0}, \qquad (6.2)\\ \int_\Omega q\nabla\cdot\delta\vec{u} &= \vec{0}, \qquad (6.3) \end{aligned}$$

for all $\vec{v} \in \mathbf{H}^1_{E_0}$ and q $\in L_2(\Omega)$. Here, D$(\vec{u},\delta\vec{u},\vec{v})$ is the difference of the nonlinear terms and simply it can be written as:

$$D(\vec{u},\delta\vec{u},\vec{v}) = c(\delta\vec{u};\delta\vec{u},\vec{v}) + c(\delta\vec{u};\vec{u},\vec{v}) + c(\vec{u};\delta\vec{u},\vec{v}), \tag{6.4}$$

where the trilinear form $c(\cdot;\cdot,\cdot)$ is the same as in the previous section.

Now, we try to determine the stability of the solution without solving (6.2)-(6.4). We start the linear stability analysis by dropping the nonlinear term $c(\delta\vec{u},\delta\vec{u},\vec{v})$ in (6.4) and redefine $\delta\vec{u}(\vec{x},t) \leftarrow e^{\lambda t}\delta\vec{u}(\vec{x})$ and $\delta p(\vec{x},t) \leftarrow e^{\lambda t}\delta p(\vec{x})$ since they grow or decay in time. This gives the following eigenvalue problem:

Find eigenvalues λ and corresponding eigenvectors $(\delta\vec{u},\delta p)$ such that

$$\begin{aligned} c(\delta\vec{u};\vec{u},\vec{v}) + c(\vec{u};\delta\vec{u},\vec{v}) + \nu\int_\Omega \nabla\delta\vec{u}\cdot\nabla\vec{v} - \int_\Omega \delta p(\nabla\cdot\vec{v}) &= \lambda\int_\Omega \delta\vec{u}\cdot\vec{v}, \\ \int_\Omega q\nabla\cdot\delta\vec{u} &= \vec{0}. \end{aligned} \tag{6.5}$$

If at least the real part of one eigenvalue is greater than zero, then we have linear instability. In contrast, if all real parts of all eigenvalues smaller than zero then we have linear stability. However, linear stability is not enough for full nonlinear stability. In order to establish full nonlinear stability we need to use the energy analysis. In summary, this method gives a simple linear differential inequality:

$$\frac{dK}{dt} + (1-\mathcal{R}^2)K \leq 0, \tag{6.6}$$

where $K(t) := \|\delta\vec{u}(\cdot,t)\|^2$. Finally, the energy decay bound

$$K(t) \leq K(0)e^{-(1-\mathcal{R}^2)t}, \tag{6.7}$$

can be driven by introducing the positive integrating factor $e^{(1-\mathcal{R}^2)t}$ and integrating (6.6) from time t=0. Therefore, if $\mathcal{R} < 0$ then all perturbations will decay zero. Thereby, we have nonlinear stability of the solution.

However, this theory does not hold in some real fluid examples. For example, the fluid flow around a cylinder placed in the center of a channel is stable for low fluid speeds, i.e. for small Reynolds numbers. But if the velocity of fluid increases, then it loses the stability after a certain value of Reynolds number. This number is known as the critical Reynolds number [28, p320-323]. Therefore, the determination of critical Reynolds number is an important matter. Although we used standard Galerkin finite element methods, you can see the paper [38] in which the approximation methods of the critical Reynolds number with adaptive methods were discussed with discontinuous Galerkin finite element methods.

Finally, even if we used usual norms in this study, we could use energy norm in the derivation of a posteriori error estimation.

References

[1] AJ Davies. *The Finite Element Method: An Introduction with Partial Differential Equations.* OUP Oxford, 2011.

[2] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numerica*, 4:105–158, 1995.

[3] T. Oden. Some historic comments on finite elements. In *Proceedings of the ACM conference on History of scientific and numeric computation*, HSNC '87, pages 125–130, New York, NY, USA, 1987. ACM.

[4] R. Verfurth. A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics*, 50(1-3):67–83, 1994.

[5] R. Rannacher. Adaptive finite element methods for partial differential equations. *ArXiv Mathematics e-prints*, April 2003.

[6] Y. Huang, H. Wei, W. Yang, and N. Yi. A new a posteriori error estimate for adaptive finite element methods. *Domain Decomposition Methods in Science and Engineering XIX*, pages 63–74, 2011.

[7] Paul Houston. Error and uncertainty quantification. University of Nottingham–University Lecture, 2011/2012.

[8] J. Tinsley Oden, W. Wu, and M. Ainsworth. An a posteriori error estimate for finite element approximations of the navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 111(1-2):185–202, 1994.

[9] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques*, volume 1. Wiley-Teubner New York, Stuttgart, 1996.

[10] S. Prudhomme and JT Oden. A posteriori error estimation and error control for finite element approximations of the time-dependent navier-stokes equations. *Finite elements in analysis and design*, 33(4):247–262, 1999.

[11] J.Z. Zhu and Z. Zhang. The relationship of some a posteriori error estimators. *Studies in Applied Mechanics*, 47:25–42, 1998.

[12] A. Russo. A posteriori error estimators for the stokes problem. *Applied Mathematics Letters*, 8(2):1–4, 1995.

[13] M. Grajewski, J. Hron, and S. Turek. Dual weighted a posteriori error estimation for a new nonconforming linear finite element on quadrilaterals. *Applied numerical mathematics*, 54(3):504–518, 2005.

[14] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica 2001*, 10(1):1–102, 2001.

[15] I. Babuka and A. Miller. The post-processing approach in the finite element method, part 1: Calculation of displacements, stresses and other higher derivatives of the displacements. *International Journal for Numerical Methods in Engineering*, 20(6):1085–1109, 1984.

[16] I. Babuka and A. Miller. The post-processing approach in the finite element method, part 2: The calculation of stress intensity factors. *International Journal for Numerical Methods in Engineering*, 20(6):1111–1129, 1984.

[17] I. Babuka and A. Miller. The post-processing approach in the finite element method, part 3: a posteriori error estimates and adaptive mesh selection. *International Journal for Numerical Methods in Engineering*, 20(12):2311–2324, 1984.

[18] I. Babuska, R. Durán, and R. Rodríguez. Analysis of the efficiency of an a posteriori error estimator for linear triangular finite elements. *SIAM journal on numerical analysis*, pages 947–964, 1992.

[19] I. Babuska and W.C. Rheinboldt. A posteriori error analysis of finite element solutions for one-dimensional problems. *SIAM Journal on Numerical Analysis*, pages 565–589, 1981.

[20] I. Babuska and WC Rheinboldt. Analysis of optimal finite-element meshes in r/1. *Mathematics of Computation*, 33(146):435–463, 1979.

[21] I. Babuska. The selfadaptive approach in the finite element method. *The Mathematics of Finite Elements and Application II*, pages 125–142, 1976.

[22] I. Babuska and W.C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, 1978.

[23] I. Babuska and W.C. Rheinboldt. Reliable error estimation and mesh adaptation for the finite element method. Technical report, Maryland Univ., College Park (USA). Inst. for Physical Science and Technology, 1979.

[24] I. Babuska and A. Miller. A-posteriori error estimates and adaptive techniques for the finite element method. Technical report, DTIC Document, 1981.

[25] J.E. Flaherty. *Adaptive methods for partial differential equations*. Number 39. Society for Industrial & Applied, 1989.

[26] R. Verfürth. A review of a posteriori error estimation techniques for elasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 176(1):419–440, 1999.

[27] E. Hall, P. Houston and S. Giani. Aptofem v2, users manual. University Of Nottingham, 2012. Software toolkit.

[28] H.C. Elman, Silvester D.J., and A.J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, USA, 2005.

[29] Edward Hall. Variational methods. University of Nottingham-University Lecture, 2011/2012.

[30] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Books on Mathematics. Dover Publications, 2009.

[31] D. Braess. *Finite elements: Theory, fast solvers, and applications in elasticity theory*. Cambridge Univ Pr, 2007.

[32] E.H. Georgoulis, E. Hall, and P. Houston. Discontinuous galerkin methods for advection-diffusion-reaction problems on anisotropically refined meshes. *SIAM Journal on Scientific Computing*, 30(1):246–271, 2007.

[33] V. Girault and P.A. Raviart. Finite element methods for navier-stokes equations: theory and algorithms. *NASA STI/Recon Technical Report A*, 87:52227, 1986.

[34] Ralf Hartmann and Paul Houston. Error estimation and adaptive mesh refinement for aerodynamic flows. In H. Deconinck, editor, *VKI LS 2010-01: 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.

[35] P. Houston, C. Schwab, and E. Süli. Stabilized hp-finite element methods for first-order hyperbolic problems. *SIAM Journal on Numerical Analysis*, 37(5):1618–1643, 2000.

[36] J.T. Oden, I. Babuŝka, and C.E. Baumann. A discontinuous *hp* finite element method for diffusion problems. *Journal of computational physics*, 146(2):491–519, 1998.

[37] D. Wang and X.P. Wang. A three-dimensional adaptive method based on the iterative grid redistribution. *Journal of Computational Physics*, 199(2):423–436, 2004.

[38] K.A. Cliffe, E.J.C. Hall, P. Houston, E.T. Phipps, and A.G. Salinger. Adaptivity and a posteriori error control for bifurcation problems ii: Incompressible fluid flow in open systems with z 2 symmetry. *Journal of Scientific Computing*, 47(3):389–418, 2011.