ADAPTIVE DISCONTINUOUS GALERKIN METHODS FOR CONVECTION DOMINATED OPTIMAL CONTROL PROBLEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAMDULLAH YÜCEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
SCIENTIFIC COMPUTING

JULY 2012

Approval of the thesis:

ADAPTIVE DISCONTINUOUS GALERKIN METHODS FOR CONVECTION DOMINATED OPTIMAL CONTROL PROBLEMS

submitted by **HAMDULLAH YÜCEL** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Bülent Karasözen _____
Head of Department, **Scientific Computing**

Prof. Dr. Bülent Karasözen _____
Supervisor, **Institute of Applied Mathematics, METU**

Examining Committee Members:

Prof. Dr. Münevver Tezer _____
Institute of Applied Mathematics, METU

Prof. Dr. Bülent Karasözen _____
Institute of Applied Mathematics, METU

Prof. Dr. Ömer Akın _____
Department of Mathematics,
TOBB Economics and Technology University

Prof. Dr. Gerhard Wilhelm Weber _____
Institute of Applied Mathematics, METU

Assoc. Prof. Dr. Songül Kaya Merdan _____
Department of Mathematics, METU

Date: _____

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

Name, Last Name: HAMDULLAH YÜCEL

Signature :

ABSTRACT

ADAPTIVE DISCONTINUOUS GALERKIN METHODS FOR CONVECTION DOMINATED OPTIMAL CONTROL PROBLEMS

Yücel, Hamdullah
Ph. D., Department of Scientific Computing
Supervisor : Prof. Dr. Bülent Karasözen

July 2012, 156 pages

Many real-life applications such as the shape optimization of technological devices, the identification of parameters in environmental processes and flow control problems lead to optimization problems governed by systems of convection diffusion partial differential equations (PDEs). When convection dominates diffusion, the solutions of these PDEs typically exhibit layers on small regions where the solution has large gradients. Hence, it requires special numerical techniques, which take into account the structure of the convection. The integration of discretization and optimization is important for the overall efficiency of the solution process. Discontinuous Galerkin (DG) methods became recently as an alternative to the finite difference, finite volume and continuous finite element methods for solving wave dominated problems like convection diffusion equations since they possess higher accuracy.

This thesis will focus on analysis and application of DG methods for linear-quadratic convection dominated optimal control problems. Because of the inconsistencies of the standard stabilized methods such as streamline upwind Petrov Galerkin (SUPG) on convection diffusion optimal control problems, the *discretize-then-optimize* and the *optimize-then-discretize* do not

commute. However, the upwind symmetric interior penalty Galerkin (SIPG) method leads to the same discrete optimality systems. The other DG methods such as nonsymmetric interior penalty Galerkin (NIPG) and incomplete interior penalty Galerkin (IIPG) method also yield the same discrete optimality systems when penalization constant is taken large enough. We will study a posteriori error estimates of the upwind SIPG method for the distributed unconstrained and control constrained optimal control problems. In convection dominated optimal control problems with boundary and/or interior layers, the oscillations are propagated downwind and upwind direction in the interior domain, due the opposite sign of convection terms in state and adjoint equations. Hence, we will use residual based a posteriori error estimators to reduce these oscillations around the boundary and/or interior layers. Finally, theoretical analysis will be confirmed by several numerical examples with and without control constraints.

Keywords: Optimal control problems, discontinuous Galerkin Method, convection dominated problems, adaptive mesh refinement, a posteriori estimates

ÖZ

KONVEKSİYON AĞIRLIKLI ENİYİLEMELİ KONTROL PROBLEMLERİNİN UYARLAMALI KESİNTİLİ GALERKİN YÖNTEMLERİYLE ÇÖZÜMÜ

Yücel, Hamdullah
Doktora, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. Bülent Karasözen

Temmuz 2012, 156 sayfa

Gerçek yaşamda karşılaşılan, teknolojik sistemlerin eniyileme yöntemleriyle kontrolü, çevresel süreç içindeki parametrelerin belirlenmesi, akışkan kontrol problemleri gibi çok sayıda problem, konveksiyon difüzyon terimleri içeren kısmi türevli denklem sistemlerinden oluşan eniyileme modelleri şeklindedir. Konveksiyon terimlerinin difüzyon terimlerinden çok daha büyük olduğu durumlarda, bu tür denklemlerin çözümleri, çözümün yüksek eğime sahip olduğu bölgelerde katmanlar oluşturmaktadır. Bu tür kısmi türevli denklemlerin sayısal çözümleri genelde istenmeyen salınımlar ürettiğinden, konveksiyon teriminin yapısı göz önüne alınarak, uygun yöntemlerin uygulanması gerekmektedir. Problemdeki uzay değişkenlerinin ayrıklaştırılması ve eniyileme yöntemlerinin entegrasyonu, problemin çözümünün elde edilmesi sürecinin verimliliği açısından da önemlidir. Son yıllarda, sınır ya da iç bölgelerde salınımlar gösteren konveksiyon ağırlıklı denklemlerin sayısal çözümlerinde, süreksiz Galerkin sonlu elemanlar yöntemi, yüksek mertebeden kesinliklikte iyi sonuçlar verdiğinden sonlu farklar, sonlu hacimler ve sürekli sonlu elemanlar yöntemlerine bir seçenek olarak ortaya çıkmıştır.

Bu tez, konveksiyon ağırlıklı ikinci dereceden doğrusal eniyileme kontrol problemleri için

süreksiz Galerkin yöntemlerinin çözülmesini ve analizini içermektedir. Konveksiyon ağırlıklı eniyileme problemlerinin kararlaştırılmasında kullanılan standart sonlu elemanlar yöntemleri eniyileme problemleri için tutarsızlıklar oluşturduğundan, eniyilemeli kontrol probleminin doğrudan ayrıklaştırılmasıyla elde edilen sonuçlarla, eniyileme koşullarından elde edilen sisteminin ayrıklaştırılması sonucu elde edilen sonuçlar birbirinden farklılıklar göstermektedir. Buna karşın, simetrik süreksiz Galerkin yöntemleri aynı ayrık eniyileme koşullarını vermektedir. Simetrik olmayan, süreksiz Galerkin yöntemleri ise ancak cezalandırma sabiti yeterince büyük alındığında benzer ayrık eniyileme koşullarını oluşturmaktadır. Sayısal sonuçları içeren sonradan hata tahminleri kısıtsız ve kontrol kısıtlı eniyileme kontrol problemleri üzerinde simetrik kesintili Galerkin yöntemi kullanılarak elde edilmiştir. Sınır ya da iç katmanlara sahip konveksiyon ağırlıklı eniyileme problemleri, durum ve eşlenik kısmi türevli denklemlerinin zıt yönlü konveksiyon terimi içermesinden dolayı hem konveksiyon teriminin yönünde hem de onun ters yönünde salınımlar yapar. Sınır ya da iç katmanlar üzerindeki bu salınımları azaltmak için uyarlamalı ağ daraltma yöntemi kullanıldı. Son olarak, kısıtsız ve kontrol kısıtlı eniyileme örneklerinden elde edilen sayısal sonuçlar teorik analizden elde edilen sonuçları doğrulamaktadır. Bu da kesintili Galerkin yöntemlerinin eniyileme kontrol problemleri üzerindeki etkinliğini göstermektedir.

Anahtar Kelimeler: Eniyilemeli kontrol problemleri, kesintili Galerkin yöntemleri, konveksiyon ağırlıklı problemler, uyarlamalı ağ yöntemleri, sonradan hata tahminleri

To my family

ACKNOWLEDGMENTS

It is a great pleasure to have opportunity to express my thankfulness to all those people who have helped in the presentation of this thesis.

First of all, I would like to thank my supervisor Prof. Dr. Bülent Karasözen for invaluable guidance and helpful suggestions throughout not only this thesis but also all my academic studies. His careful proofreading and useful comments improved this thesis. Without his great support I would have not been able to continue and complete my PhD studies.

I would like to thank Prof. Dr. Matthias Heinkenschloss for his hosting in Houston. My special thanks to him also for the effort and time he spent for my improvement in research.

I am also grateful to the members of Prof. Dr. Matthias Heinkenschloss research group, Sean Hardesty, Drew Kouri, Harbir Antil and Jane Ghiglieri, for their unlimited help, precious discussion and collaboration. I would also like to take this opportunity to thank for the hospitality of Computational & Applied Mathematics, Rice University.

I would like to sincerely thank Prof. Dr. Ronald W. Hoppe (University of Houston-University of Augsburg) for helpful discussions on my studies.

I am especially grateful to Prof. Dr. Gerhard Wilhelm Weber for his kindness and proofreading during my studies.

I am very thankful to Harbir Antil for implementation of finite element methods and Sevtap Özışık for discussions on discontinuous Galerkin methods.

I am also thankful to my thesis defence committee members for their useful comments and discussions.

I also gratefully acknowledge the financial support of Turkish Scientific and Technical Research Council (TÜBİTAK).

I would like to thank all members of the Institute of Applied Mathematics and Department of

Mathematics at Middle East Technical University for the pleasant atmosphere, and everybody who helped me directly or indirectly to accomplish this thesis.

Lastly, I would like to special thank my family for the support which they provided me through my entire life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Many real-life applications such as the optimization of technological devices [80], the optimal control of systems [47], the identification of parameters in environmental process and flow control problems [37, 44, 83] lead to optimization problems governed by systems of convection diffusion partial differential equations (PDEs). In these systems, the convection term dominates the diffusion term, in general; boundary and/or interior layers occur on a small region with large derivatives of the solution. Boundary layers stem out since the interior solution having strong convection suddenly has to match the Dirichlet boundary conditions on the outflow boundary, whereas interior layers arise a discontinuity at the inflow boundary data. Hence, the special numerical techniques considering the structure of the convection are needed to solve these PDEs numerically. It is well known that the standard Galerkin finite element method is not suitable for the solution of convection diffusion equations. To enhance the stability and the accuracy of solution of the convection dominated problems, several stabilization techniques have been proposed and analyzed such as the streamline upwind Petrov Galerkin (SUPG) [61], edge stabilization Galerkin method [27]. However, to find a suitable stabilization parameter depending on convection term is not always easy while using the stabilization techniques. Hence, discontinuous Galerkin (DG) methods having free parameters are introduced in [5, 102]. DG methods have recently become popular to solve wave dominated problems like convection diffusion equations. They have higher accuracy and work better in complex geometries with respect to standard continuous Galerkin methods. Additionally, DG methods have greater flexibilities to locally adapt the mesh or the polynomial degree of the basis functions and can capture possible discontinuous in the solution due to the discontinuous approximation spaces. The other interesting property of DG methods is conservation of mass on each mesh element. This local mass conservation property of DG methods makes

them to be a good candidate to solve in flow and transport problems. Further, DG methods have a compact formulation since the solution within each element is not reconstructed by looking to neighboring elements. Its compact formulation can be applied near boundaries without special treatment, which greatly increases the robustness and accuracy of any boundary condition implementation. The compact form of the DG method also makes them well suited for parallel computer platforms. Because of that, DG methods have become popular for convection dominated equations [7, 59].

In this thesis, we consider the linear-quadratic optimal control problems

$$\text{minimize} \quad J(y,u) := \frac{1}{2}\int_\Omega (y(x)-y_d(x))^2 dx + \frac{\omega}{2}\int_\Omega u(x)^2 dx \tag{1.1}$$

governed by convection diffusion equations

$$-\epsilon\Delta y(x) + \beta(x)\cdot\nabla y(x) + r(x)y(x) = f(x)+u(x), \qquad x\in\Omega, \tag{1.2a}$$

$$y(x) = g_D(x), \qquad x\in\Gamma, \tag{1.2b}$$

where Ω be a bounded open, convex domain in $\mathbb{R}^2$ with boundary $\Gamma = \nabla\Omega$. Let $f, y_d \in L^2(\Omega), g_D \in H^{3/2}(\Gamma), \beta \in (W^{1,\infty}(\Omega))^d$ and $r \in L^\infty(\Omega)$ be given functions with $r - \frac{1}{2}\nabla\cdot\beta \geq 0$ a.e. in Ω, and let $\epsilon, \omega > 0$ be given scalars.

The main aim of an optimal control problem is that a state denoted by y should be approximated a given desired state y_d as well as possible by using a control variable u. The objective function $J(y,u)$, which should be minimized, consists of the measure of approximate state to the desired state and a term penalizing the high cost of control. In this thesis, the constraint of the optimization problem is the partial differential equations, such as convection diffusion equations. Additionally, there can be constraints on the control and/or the state, like a lower and/or an upper bounds, i.e., $u_a \leq u \leq u_b$ with constant u_a and u_b. Such an additional constraints change the structure of optimal control problems. The optimality conditions of unconstrained optimal control problems consists of a system of partial differential equations, whereas variational inequalities additionally appear in the system for control and/or state constrained optimal control problems.

The numerical solution of the optimization problems governed by convection diffusion PDEs (1.1)-(1.2) require additional challenges, unlike the single convection diffusion equations (1.2). One reason why the solution of the optimization problems governed by convection diffusion PDEs (also referred to as the state PDEs) provides additional challenges is that the

optimality conditions of such problems also involve another convection diffusion PDE, the so-called adjoint PDE (1.3) with the negative convection term.

$$-\epsilon\Delta p(x) - \beta(x)\cdot\nabla y(x) + (r(x) - \nabla\cdot\beta(x))y(x) = -(y(x) - y_d(x)), \quad x \in \Omega, \tag{1.3a}$$

$$p(x) = 0, \qquad x \in \Gamma. \tag{1.3b}$$

In optimal control context, the following question arises whether it is better to formulate the control problem on the continuous level and then discretize, "optimize-discretize" or to discretize the control problem and derive optimality system, "discretize-optimize". It is known that both approaches lead to same discretization schemes when pure Galerkin finite element discretization is used. The standard stabilized methods such as SUPG [61] are not well suited for the optimal control problems governed by convection diffusion equations. It has been shown in [35] that the optimality system of the SUPG discretized optimal control problem is not equivalent to the SUPG discretization of the optimality system, i.e., optimization and discretization do not commute. To overcome this shortcoming, several symmetric stabilization methods have been recently developed leading the same discrete optimality system, such as local projection stabilization [18] and edge stabilization [54, 104] to solve unconstrained and control constrained problems governed by convection diffusion equations. For convection dominated optimal control problems, Leykekhman and Heinkenschloss [48] have shown that the local error of the SUPG discretized optimal control problem is not optimal even if the error is computed locally in region away from the boundary layer. Then, this problem has been overcome by using the symmetric interior penalty Galerkin (SIPG) method in [73]. The reason why the SIPG method gives an optimal convergence is the weak treatment of DG methods for boundary conditions, whereas SUPG methods have strongly imposition of boundary conditions. While solving control constrained optimal control problems governed by elliptic equations, different approaches have been proposed and applied: fully discretization [4, 28, 29, 90] and variational discretization [52], i.e., the control is not discretized, post-processing [79]. For optimal control problems governed by convection diffusion equations, it has been shown in [18, 104] that the priori convergence rate of control is $O(h^{3/2})$ by using the piecewise linear discretization of the control, whereas it is improved to $O(h^2)$ by using variational discretization in [54]. Furthermore, mixed finite elements [55] and Raviart-Thomas mixed FEM/DG [105] are applied to optimal control problems governed by convection diffusion equations.

The optimal control problems governed by convection dominated PDEs have boundary and/or

interior layers generated in the state PDE as well as in the adjoint PDE. Errors caused by spurious oscillations are propagated up- and downwind from the layers due to the coupling of convection diffusion PDEs having the opposite convection. This motives the use of mesh adaption for the solution of convection dominated optimal control problems. With the help of adaptive finite element methods (AFEMs), the regions can be found where the solution is hard to approximate with a less degrees of freedom and as less computational time. AFEMs and the a posteriori error analysis of finite elements methods became a standard approach in the finite element literature in solving single equations and optimal control problems. There exist mainly two approaches in mesh adaptivity, the one with residual based [3, 8, 97] and goal-oriented [17, 23, 69] estimators. Explicit a posteriori estimators of DG methods applied to pure diffusion problems have been studied in [2, 14, 20, 57, 64, 66, 88]. The convection diffusion case has been studied in [43, 59, 91]. In [91] Schötzau and Zhu have extended the results in [99] to SIPG discretization for the convection diffusion problems by developing a robust estimator with respect to the ratio of diffusion and convection coefficients.

The a posteriori error analysis of AFEM for elliptic optimal control problems are studied in [74] for residual-type a posteriori error estimators and for goal-oriented dual weighted estimators in [10, 49]. There are few works using AFEM for convection dominated optimal control problems. In [38], Dedè and Quarteroni have used a posteriori error estimates with a stabilization method applied to the Lagrangian functional for optimal control problems governed by convection diffusion equations. Nederkoorn in [81] has combined adaptive finite element methods with SUPG stabilization introduced in [35], to linear-quadratic convection dominated elliptic optimal control problems. For control constrained optimal control problems governed by convection diffusion equations, a posteriori error estimates have been used with the edge stabilization [54, 104] and with RT mixed FEM/DG [105]. For DG discretization case, there are a few work for the solution of optimal control problems governed by convection diffusion equations. A priori error estimator has been derived in [106] using the local discontinuous Galerkin (LDG) method, whereas both a priori and a posteriori error estimates have been given in [103] using nonsymmetric interior penalty Galerkin (NIPG) method. However, to our knowledge there is any numerical results to illustrate the performance of theoretical results in literature.

The goal of this thesis is the development, analysis and application of discontinuous Galerkin (DG) methods for linear-quadratic elliptic convection dominated optimal control problems.

We have investigated basic questions related to the issue whether a DG discretization of the optimal control problem leads to the same result as the same DG discretization applied to the optimality system. In addition, we have applied AFEM to SIPG discretized convection dominated optimal control problems (1.1)-(1.2) for both unconstrained and control constrained case. We have analyzed the reliability and the efficiency of the error estimator using data approximation errors for the discretized optimal control problem. We use the upper and lower bounds given in terms of energy norm and a semi-norm associated with the convective terms for the state, adjoint variables as in [91] and [99]. For the control, we use the error estimators in L_2 norm, which were introduced [74, 103].

This thesis first describes the DG methods for a single partial differential equations and optimal control problems governed by partial differential equations, and then applies the adaptive finite element method to solve unconstrained and control constrained optimal control problems governed by convection diffusion equations. The rest of the thesis is organized as follows:

In Chapter 2, we firstly describes the DG methods for elliptic problems. The derivation of some DG methods using numerical fluxes and comparison of DG methods in terms of consistency, adjoint consistency, rate of convergence are surveyed. Then, convection diffusion problems with DG methods are introduced by giving error analysis in energy norm. In Chapter 3, the unconstrained optimal control problems are described and analyzed by the finite element discretization and DG discretization. The diffusion term is discretized by symmetric interior penalty Galerkin (SIPG), the nonsymmetric interior penalty Galerkin (NIPG) or incomplete interior penalty Galerkin (IIPG) method, whereas the convection term is discretized by upwind discretization. Then, two numerical approaches, the *discretize-then-optimize* and the *optimize-then-discretize*, to solve the optimal control problems governed by convection diffusion equations are presented and compared by using DG methods.

In Chapter 4, the unconstrained convection dominated optimal control problems are solved using AFEM with upwind SIPG discretization. Each step of the adaptive loop is described. We give a robust posteriori error estimator by showing the reliability and efficiency of it. Numerical examples are also presented to illustrate the performance of the DG estimator. In Chapter 5, we solve control constrained optimal control problems governed by convection diffusion equations using upwind SIPG discretization. The a posteriori error estimator given

in Chapter 4 is extended to control constrained case. Similar to the works in Chapter 4, the reliability and the efficiency of the error estimator is analyzed using data approximations errors for the discretized optimal control problem. Furthermore, we present the numerical results to illustrate the performance of the adaptive method on convection dominated control constrained optimal control problems.

In Appendix A, the MATLAB routines used for solving the optimal control problems using DG and AFEM are explained.

CHAPTER 2

DISCONTINUOUS GALERKIN METHODS

The discontinuous Galerkin method was firstly introduced for first-order linear hyperbolic problems by Reed and Hill [86] in 1973. Independently of this work, Dougles, Dupont and Wheeler [40, 102] and Arnold [5] introduced the discontinuous Galerkin methods for elliptic and parabolic equations in seventies. In time, advanced versions of discontinuous Galerkin methods have been presented and studied for elliptic problems in [9, 12, 26, 84, 89] and for convection diffusion problems in [7, 13, 34, 46, 59].

One reason of why DG methods have been so popular is their flexibility with respect to mesh and the local polynomial degree of the basis functions. Then, DG methods can handle complicated geometries by the use of unstructured grids or hanging nodes. In addition, different orders of approximations can be used with DG discretization on each element. Hence, *hp*-methods [95] combining adaptively elements with variable size h and polynomial degree p are especially suitable with DG methods. The other interesting property of DG methods is conservation of mass on each mesh element. This local mass conservation property of DG methods makes them to be a good candidate to solve in flow and transport problems. Further, DG methods have a compact formulation since the solution within each element is not reconstructed by looking to neighboring elements. Its compact formulation can be applied near boundaries without special treatment, which greatly increases the robustness and accuracy of any boundary condition implementation. The compact form of the DG method also makes them well suited for parallel computer platforms. However, drawbacks of the DG methods are the large number of degrees of freedom compared to standard finite element methods and ill-conditioning of the matrices with increasing degree of the basis polynomials.

In this chapter, we firstly give information about the discontinuous Galerkin methods for el-

liptic problems. We survey the derivation of some discontinuous Galerkin methods using numerical fluxes and compare DG methods in terms of consistency, adjoint consistency, rate of convergence. In the second part of this chapter, convection diffusion problems are introduced by giving error analysis in energy norm, based on DG discretization.

2.1 Preliminaries

2.1.1 Sobolev Spaces

Let Ω be a bounded polygonal domain in $\mathbb{R}^d$. For $1 \leq p \leq \infty$ the vector spaces $L^p(\Omega)$ are defined

$$L^p(\Omega) = \{\upsilon \text{ Lebesgue measurable} : \ \|\upsilon\|_{L^p(\Omega)} < \infty\}$$

with the norms

$$\|\upsilon\|_{L^p(\Omega)} = \left(\int_\Omega |\upsilon(x)|^p\right)^{1/p}$$

and

$$\|\upsilon\|_{L^\infty(\Omega)} = ess\sup\{|\upsilon(x)| : \ x \in \Omega\}.$$

We define

$$\begin{aligned} L^p_{loc}(\Omega) &= \{\upsilon \text{ Lebesgue meausurable} : \ \upsilon \in L^p(K) \text{ for all } K \subset \Omega \text{ compact}\}, \\ C^\infty_c(\Omega) &= \{\upsilon \in C^\infty(\bar{\Omega}) : \ supp(\upsilon) \subset \Omega \text{ compact}\}. \end{aligned}$$

Definition 2.1.1 *Let $u \in L^1_{loc}(\Omega)$. If there exits a function $w \in L^1_{loc}(\Omega)$ such that*

$$\int_\Omega w\varphi dx = (-1)^{|\alpha|}\int_\Omega \upsilon D^\alpha\varphi dx, \quad \forall\varphi \in C^\infty_c(\Omega),$$

then $D^\alpha u := w$ is called the α-th weak partial derivative of u.

Now, we introduce the Sobolev space $W^{k,p}(\Omega)$ by

$$W^{s,p}(\Omega) = \{\upsilon \in L^p(\Omega) : \ \upsilon \text{ has weak derivatives } D^\alpha\upsilon \in L^p(\Omega) \text{ for all } |\alpha| \leq s\}$$

with the norm

$$\begin{aligned} \|\upsilon\|_{W^{s,p}(\Omega)} &= \left(\sum_{|\alpha|\leq s}\|D^\alpha\upsilon\|^p_{L^p}\right)^{1/p}, \quad p \in [1,\infty), \\ \|\upsilon\|_{W^{s,\infty}(\Omega)} &= \sum_{|\alpha|\leq s}\|D^\alpha\upsilon\|_{L^\infty}. \end{aligned}$$

Remark 2.1.2 *In the case $p = 2$, $H^s(\Omega) \overset{def}{=} W^{s,2}(\Omega)$. Then, we can write*

$$H^s(\Omega) = \{\upsilon \in L^2(\Omega) : \ \forall\ 0 \le |\alpha| \le s,\ D^\alpha \upsilon \in L^2(\Omega)\},$$

associated with the Sobolev norm and the Sobolev seminorm, respectively, defined by

$$\begin{aligned} \|\upsilon\|_{H^s(\Omega)} &= \left(\sum_{|\alpha|\le s} \|D^\alpha u\|^2_{L^p}\right)^{1/2}, \\ \|\upsilon\|_{H^s(\Omega)} &= \|\nabla^s \upsilon\|_{L^2(\Omega)} = \left(\sum_{|\alpha|= s} \|D^\alpha \upsilon\|^2_{L^2(\Omega)}\right)^{1/2}. \end{aligned}$$

By using the subdivision domain ξ_h obtained by dividing the polygonal domain Ω into triangle elements E, we define the broken Sobolev space by

$$H^s(\xi_h) = \{\upsilon \in L^2(\Omega) : \ \forall E \in \xi_h,\ \upsilon|_E \in H^s(E)\},$$

associated with broken Sobolev norm and broken gradient seminorm, respectively, defined by

$$\begin{aligned} \|\upsilon\|_{H^s(\xi_h)} &= \left(\sum_{E\in\xi_h} \|\upsilon\|^2_{H^s(E)}\right)^{1/2}, \\ \|\nabla\upsilon\|_{H^0(\xi_h)} &= \left(\sum_{E\in\xi_h} \|\nabla\upsilon\|^2_{L^2(E)}\right)^{1/2}. \end{aligned}$$

2.1.2 Trace Theorems

Theorem 2.1.3 *[87, Theorem 2.5] Trace operators $\gamma_0 : H^s(\Omega) \to H^{s-1/2}(\partial\Omega)$ for $s > 1/2$ and $\gamma_1 : H^s(\Omega) \to H^{s-3/2}(\partial\Omega)$ for $s > 3/2$ which are extensions of the boundary values and boundary normal derivatives, respectively, can be defined on the bounded domain Ω with polygonal boundary $\partial\Omega$. In addition, under the condition $\upsilon \in C^1(\bar{\Omega})$, these operators satisfy the following conditions:*

$$\gamma_0\upsilon = \upsilon|_{\partial\Omega}, \qquad \gamma_1\upsilon = \nabla\upsilon \cdot \mathbf{n}|_{\partial\Omega}.$$

We need some trace inequalities that are used in analysis of the DG methods. These trace inequalities are given by

$$\|\upsilon\|_{L^2(e)} \le C_t h_E^{-1/2} \|\upsilon\|_{L^2(E)}, \tag{2.1a}$$

$$\|\nabla\upsilon \cdot \mathbf{n}\|_{L^2(e)} \le C_t h_E^{-1/2} \|\nabla\upsilon\|_{L^2(E)}, \tag{2.1b}$$

where the positive constant C_t is independent of diameter of E, h_E.

2.1.3 Cauchy-Schwarz's and Young's Inequalities

Both inequalities are the most used inequalities in analysis part of numerical methods. We will use them at several places in this thesis also.

- Cauchy-Schwarz's inequality:

$$|(f, g)_\Omega| \leq \|f\|_{L^2(\Omega)} \|g\|_{L^2(\Omega)}, \qquad \forall f, g \in L^2(\Omega). \tag{2.2}$$

- Young's inequality:

$$ab \leq \frac{\gamma}{2} a^2 + \frac{1}{2\gamma} b^2, \qquad \forall \gamma > 0,\ \forall a, b \in \mathbb{R}. \tag{2.3}$$

2.2 Discontinuous Galerkin Methods for Elliptic Problems

This section shows the derivation of discontinuous Galerkin methods for elliptic problems using numerical fluxes. Additionally, some properties of DG methods, i.e., consistency, adjoint consistency, stability and convergence rates are presented and compared. Furthermore, we explain the differences between interior penalty (symmetric interior penalty Galerkin (SIPG)) method, local discontinuous Galerkin (LDG) method [34] and its modified form, i.e., compact discontinuous Galerkin (CDG) method [84].

2.2.1 The Primal Formulation of DG Methods and Their Properties

We consider the purely elliptic problem as a model problem:

$$\begin{aligned} -\nabla \cdot (\epsilon \nabla y) &= f, && \text{in} \quad \Omega, \\ y &= g_D, && \text{on} \quad \Gamma_D, \\ \epsilon \frac{\partial y}{\partial n} &= g_N, && \text{on} \quad \Gamma_N, \end{aligned} \tag{2.4}$$

where Ω is a bounded domain in $\mathbb{R}^d$ with a boundary $\Gamma = \Gamma_D \cup \Gamma_N$ and d=1, 2 or 3 as the dimension. In addition, $f \in L^2(\Omega)$ and $\epsilon \in \mathbb{R}^d$.

The problem (2.4) can be rewritten as a first order system of equations

$$
\begin{aligned}
-\nabla \cdot \boldsymbol{\sigma} &= f, \qquad \text{in} \quad \Omega, && \text{(2.5a)}\\
\boldsymbol{\sigma} &= \epsilon \nabla y, \qquad \text{in} \quad \Omega, && \text{(2.5b)}\\
y &= g_D, \qquad \text{on} \quad \Gamma_D, && \text{(2.5c)}\\
\boldsymbol{\sigma} \cdot \mathbf{n} &= g_N, \qquad \text{on} \quad \Gamma_N, && \text{(2.5d)}
\end{aligned}
$$

where $\mathbf{n}$ is the outward unit normal to the boundary of Ω.

To obtain the weak formulation, we multiply (2.5a) and (2.5b) by test functions τ and υ, respectively, and apply integration by parts on a subset of E of Ω.

$$
\begin{aligned}
\int_E \boldsymbol{\sigma} \cdot \tau \, dx &= -\int_E y \nabla \cdot \tau \, dx + \int_{\partial E} y \mathbf{n_E} \cdot \tau \, ds,\\
\int_E \boldsymbol{\sigma} \cdot \nabla \upsilon \, dx &= \int_E f \upsilon \, dx + \int_{\partial E} \boldsymbol{\sigma} \cdot \mathbf{n_E} \upsilon \, ds,
\end{aligned}
$$

where $\mathbf{n_E}$ is the outward unit normal to ∂E.

The broken Sobolev spaces given in Section 2.1.1 are natural spaces to work with the DG methods since they depend strongly on the partition of domain. Then, the broken spaces $V(\xi_h)$ and $\Sigma(\xi_h)$ associated with the triangulation $\xi_h = \{E\}$ of Ω are introduced such as

$$
\begin{aligned}
V &= \{\upsilon \in L^2(\Omega) \mid \upsilon|_E \in H^1(E), \ \ \forall E \in \xi_h\}, && \text{(2.6)}\\
\Sigma &= \{\tau \in [L^2(\Omega)]^d \mid \tau|_E \in [H^1(E)]^d, \ \ \forall E \in \xi_h\}. && \text{(2.7)}
\end{aligned}
$$

Then, the finite element subspaces $V_h \subset V$ and $\Sigma_h \subset \Sigma$ are

$$
\begin{aligned}
V_h &= \{\upsilon \in L^2(\Omega) \mid \upsilon|_E \in P_k(E), \ \ \forall E \in \xi_h\}, && \text{(2.8)}\\
\Sigma_h &= \{\boldsymbol{\tau} \in [L^2(\Omega)]^d \mid \boldsymbol{\tau}|_E \in [P_k(E)]^d, \ \ \forall E \in \xi_h\}, && \text{(2.9)}
\end{aligned}
$$

where $P_k(E)$ is the space of polynomial functions of degree at most $k \geq 1$ on E.

Then, the DG formulation is such that: find $y_h \in V_h$ and $\boldsymbol{\sigma}_h \in \Sigma_h$ such that for all $E \in \xi_h$ we have

$$
\begin{aligned}
\int_E \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau} dx &= -\int_E y_h \nabla \cdot (\epsilon \tau) dx + \int_{\partial E} \hat{y} \epsilon \boldsymbol{\tau} \cdot \mathbf{n_E} ds, \qquad \forall \boldsymbol{\tau} \in [P_k(E)]^d, && \text{(2.10)}\\
\int_E \boldsymbol{\sigma}_h \cdot \nabla \upsilon dx &= -\int_E f \upsilon dx + \int_{\partial E} \hat{\boldsymbol{\sigma}} \cdot \mathbf{n_E} \upsilon ds, \qquad \forall \upsilon \in P_k(E), && \text{(2.11)}
\end{aligned}
$$

where $\hat{\boldsymbol{\sigma}}$ and $\hat{y}$ are called the numerical fluxes which are approximations to $\boldsymbol{\sigma} = \epsilon \nabla y$ and to y, respectively. These fluxes can be thought as the local quantities depending on the traces

of the edge e of functions $y_h|_{E^1,E^2}, \boldsymbol{\sigma}_h|_{E^1,E^2}$ and/or $\epsilon\nabla y|_{E^1,E^2}$, where $e \in E^1 \cap E^2$. To satisfy conservation property in numerical methods, these functions require some basic properties such as *consistency*, that is, $\hat{y}_h = y|_e$ and $\hat{\boldsymbol{\sigma}}_h = \nabla y|_e$, for a smooth function y. In addition, numerical fluxes, $\hat{\boldsymbol{\sigma}}$ and $\hat{y}$ are called conservative if they are single-valued on e, that is $\hat{y}_h|_{E^1} = \hat{y}_h|_{E^2}$ and $\hat{\boldsymbol{\sigma}}_h|_{E^1} = \hat{\boldsymbol{\sigma}}_h|_{E^2}$, where $e \in E^1 \cap E^2$. Thus, the conservative property of the numerical fluxes implies adjoint consistency which is the consistency property of the adjoint problem of (2.4).

To write the expressions (2.10) and (2.11) on the whole domain, we define an additional notation as used in [6]. We split the set of all edges Γ_h into the set and Γ_h^0 of interior edges of ξ_h and the set Γ_h^∂ of boundary edges so that $\Gamma_h = \Gamma_h^\partial \cup \Gamma_h^0$. With each edge e, we associate a unit normal vector $\mathbf{n_e}$. If e is on the boundary Γ_h^∂, then $\mathbf{n_e}$ is taken to be unit outward vector normal to Γ_h^∂. If two elements E_1^e and E_2^e are neighbors and share one common side e, there are two traces of υ and $\boldsymbol{\tau}$ along e. Assume that the normal vector $\mathbf{n_e}$ is oriented from E_1^e to E_2^e, then jump and average operators can be defined as

$$\begin{aligned} [\boldsymbol{\tau}] &= (\boldsymbol{\tau}|_{E_1^e} - \boldsymbol{\tau}|_{E_2^e}) \cdot \mathbf{n_e}, & [\upsilon] &= (\upsilon|_{E_1^e} - \upsilon|_{E_2^e}) \cdot \mathbf{n_e}, \\ \{\boldsymbol{\tau}\} &= (\boldsymbol{\tau}|_{E_1^e} + \boldsymbol{\tau}|_{E_2^e})/2, & \{\upsilon\} &= (\upsilon|_{E_1^e} + \upsilon|_{E_2^e})/2. \end{aligned} \tag{2.12}$$

Now, by summing (2.10) and (2.11) over all elements, the following expression is obtained: find $y_h \in V_h$ and $\boldsymbol{\sigma}_h \in \Sigma_h$ such that

$$\int_\Omega \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau} dx = -\int_\Omega y_h \nabla_h \cdot (\epsilon\tau) dx + \int_{\Gamma_h^0} \hat{y}[\epsilon\boldsymbol{\tau}] ds + \int_{\Gamma_h^\partial} \hat{y}\epsilon\boldsymbol{\tau} \cdot \mathbf{n_e} ds, \ \forall \boldsymbol{\tau} \in \Sigma_h, \tag{2.13}$$

$$\int_\Omega \boldsymbol{\sigma}_h \cdot \nabla_h \upsilon dx = \int_\Omega f\upsilon dx + \int_{\Gamma_h^0} \hat{\boldsymbol{\sigma}} \cdot [\upsilon] ds + \int_{\Gamma_h^\partial} \upsilon\hat{\boldsymbol{\sigma}} \cdot \mathbf{n_e} ds, \qquad \forall \upsilon \in V_h. \tag{2.14}$$

Note that, ∇_h denotes the broken gradient operator, i.e., $\nabla_h\upsilon$ and $\nabla_h \cdot \boldsymbol{\tau}$ are functions whose restriction to E is equal to $\nabla\upsilon$ and $\nabla \cdot \boldsymbol{\tau}$, respectively.

By using the following expression obtained integration by parts,

$$-\int_\Omega \upsilon \nabla_h \cdot \boldsymbol{\tau} dx = \int_\Omega \boldsymbol{\tau} \cdot \nabla_h \upsilon dx - \int_{\Gamma_h^0} ([\upsilon] \cdot \{\boldsymbol{\tau}\} + \{\upsilon\}[\boldsymbol{\tau}]) ds - \int_{\Gamma_h^\partial} \upsilon\boldsymbol{\tau} \cdot \mathbf{n_e} ds.$$

Then, (2.13) can be rewritten as

$$\begin{aligned} \int_\Omega \boldsymbol{\sigma}_h \cdot \boldsymbol{\tau} dx &= \int_\Omega \boldsymbol{\tau} \cdot (\epsilon\nabla_h y_h) dx - \int_{\Gamma_h^0} ([y_h] \cdot \{\epsilon\boldsymbol{\tau}\} - \{\hat{y} - y_h\}[\epsilon\boldsymbol{\tau}]) ds \\ &+ \int_{\Gamma_h^\partial} (\hat{y} - y_h)\epsilon\boldsymbol{\tau} \cdot \mathbf{n_e} ds, \qquad \forall \boldsymbol{\tau} \in \Sigma_h. \end{aligned} \tag{2.15}$$

Finally, the DG formulation can be completed by defining the numerical fluxes $\hat{\boldsymbol{\sigma}}$ and $\hat{y}$ in terms of $\boldsymbol{\sigma}_h$ and u_h. Different numerical fluxes yield different types of discontinuous Galerkin methods. Some of them are summarized in Table 2.1 [6].

Table 2.1: Some DG methods with their numerical fluxes

Method	$\hat{y}$	$\hat{\boldsymbol{\sigma}}$
Bassi-Rebay [12]	$\{y_h\}$	$\{\sigma_h\}$
Brezzi et al. [26]	$\{y_h\}$	$\{\sigma_h\} - C_3\{r^e([y_h])\}$
LDG [34]	$\{y_h\} - C_2 \cdot [y_h]$	$\{\sigma_h\} + C_2[\sigma_h] - C_1[y_h]$
CDG [84]	$\{y_h\} - C_2 \cdot [y_h]$	$\{\sigma_h^e\} + C_2[\sigma_h^e] - C_1[y_h]$
IP [40]	$\{y_h\}$	$\{\nabla_h y_h\} - C_1[y_h]$
Bassi et al. [12]	$\{y_h\}$	$\{\nabla_h y_h\} - C_3\{r^e([y_h])\}$
Baumann-Oden [13]	$\{y_h\} + \mathbf{n_E} \cdot [y_h]$	$\{\nabla_h y_h\}$
NIPG [89]	$\{y_h\} + \mathbf{n_E} \cdot [y_h]$	$\{\nabla_h y_h\} - C_1[y_h]$
Babuška-Zlámal [9]	$(y_h\|_E)\|_{\partial E}$	$-C_1[y_h]$
Brezzi et al. [26]	$(y_h\|_E)\|_{\partial E}$	$-C_3\{r^e([y_h])\}$

Although the vector numerical flux $\hat{\boldsymbol{\sigma}}$ is conservative for all methods, the scalar flux $\hat{y}$ is conservative for the first six methods listed in Table (2.1), so they are adjoint consistent because of the conservative property of fluxes. However, the other DG methods in Table (2.1) are not adjoint consistent. In fact, the Baumann-Oden method [13] and its stabilized version, the non-symmetric interior penalty Galerkin method (NIPG) [89], have not even a symmetric primal form. On the other hand, in spite of the symmetric primal forms of Babuška-Zlámal [9] and Brezzi et al. [26], they are not adjoint consistent since they are not consistent.

The DG methods that are completely consistent and stable, i.e., LDG, CDG, IP, converge with optimal order with respect to L^2 norm and H^1 norm. However, the inconsistent pure penalty methods, i.e., Babuška-Zlámal [9], Brezzi et al. [26], do not achieve optimal order convergence in L^2. The methods having a lack of adjoint consistency, i.e., the method of Baumann-Oden ($k \geq 2$) [13] and its stabilized form, NIPG [89], has a suboptimal rate of convergence in the L^2 norm. The suboptimal rate is recovered when the superpenalized version of NIPG method is used [87]. Although this superpenalty approach overcomes the suboptimality of NIPG method, it increases the condition number of the stiffness matrix (see Castillo [30]). The results including consistency, adjoint consistency, stability and rate of convergence in H^1 and L^2 for various DG methods are shown in Table 2.2.

Table 2.2: Properties of the DG methods

Method	Cons.	A.C.	Stab.	H^1	L^2
Brezzi et al. [26]	✓	✓	✓	h^k	h^{k+1}
LDG [34]	✓	✓	✓	h^k	h^{k+1}
CDG [84]	✓	✓	✓	h^k	h^{k+1}
IP [40]	✓	✓	✓	h^k	h^{k+1}
Bassi et al. [12]	✓	✓	✓	h^k	h^{k+1}
NIPG [89]	✓	×	✓	h^k	h^k
Babuška-Zlámal [9]	×	×	✓	h^k	h^{k+1}
Brezzi et al. [26]	×	×	✓	h^k	h^{k+1}
Baumann-Oden ($k = 1$) [13]	✓	×	×	×	×
Baumann-Oden ($k \geq 2$) [13]	✓	×	×	h^k	h^k
Bassi-Rebay [12]	✓	✓	×	$[h^k]$	$[h^{k+1}]$

2.2.2 Examples of DG Methods

Now, we introduce some DG methods, i.e., interior penalty (symmetric interior penalty Galerkin (SIPG)) method [5, 102], local discontinuous Galerkin (LDG) method [34] and compact discontinuous Galerkin (CDG) method [84] by numerical fluxes $\hat{\boldsymbol{\sigma}}, \hat{y}$.

2.2.2.1 Interior penalty method

The interior penalty method (also called symmetric interior penalty Galerkin (SIPG) method) was introduced in the late 1970s by Arnold and Wheeler [5, 102]. The numerical fluxes $(\hat{\boldsymbol{\sigma}}, \hat{y})$ in (2.14) and (2.15) are chosen as

$$\hat{\boldsymbol{\sigma}} = \{\nabla_h y_h\} - C_1[y_h], \qquad \hat{y} = \{y_h\},$$

for the interior faces, and

$$\begin{aligned} \hat{\boldsymbol{\sigma}} &= \nabla_h y_h - C_1(y_h - g_D), \quad \hat{y} = g_D \quad \text{on} \quad \Gamma_D, \\ \hat{\boldsymbol{\sigma}} &= g_N \mathbf{n}, \qquad\qquad\qquad\quad \hat{y} = y_h \quad \text{on} \quad \Gamma_N, \end{aligned}$$

for the boundary faces. Here, C_1 is a penalty weighting function given by $\sigma_e \epsilon h_e^{-1}$ on each $e \in \Gamma_h$ with σ_e being a positive number.

To obtain the primal form of the SIPG method, we need the following lifting operators [6]

$r : [L^2(\Gamma_h^0)]^d \to \Sigma_h$, $l : L^2(\Gamma_h^0) \to \Sigma_h$, and $r_D : L^2(\Gamma_D) \to \Sigma_h$:

$$\begin{aligned}
\int_\Omega r(\phi) \cdot \boldsymbol{\tau} dx &= -\int_{\Gamma_h^0} \phi \cdot \{\boldsymbol{\tau}\} ds, \qquad \forall \boldsymbol{\tau} \in \Sigma_h, \\
\int_\Omega l(q) \cdot \boldsymbol{\tau} dx &= -\int_{\Gamma_h^0} q[\boldsymbol{\tau}] ds, \qquad \forall \boldsymbol{\tau} \in \Sigma_h, \\
\int_\Omega r_D(q) \cdot \boldsymbol{\tau} dx &= -\int_{\Gamma_h^D} q\boldsymbol{\tau} \cdot \mathbf{n} ds, \qquad \forall \boldsymbol{\tau} \in \Sigma_h.
\end{aligned} \tag{2.16}$$

By using the numerical fluxes in the equations (2.14) and (2.15) with $\tau = \nabla_h \upsilon$ and lifting operators introduced in (2.16), the following system is obtained:

$$a_h^{SIPG}(y_h, \upsilon) = l_h^{SIPG}(\upsilon), \qquad \forall \upsilon \in V_h, \tag{2.17}$$

where the bilinear form $a_h^{SIPG} : V_h \times V_h \to \mathbb{R}$ is given by

$$\begin{aligned}
a_h^{SIPG}(y, \upsilon) &= \sum_{E \in \xi_h} (\epsilon \nabla y, \nabla \upsilon)_E - \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} (\{\epsilon \nabla \upsilon \cdot n_e\}, [y])_e \\
&- \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} (\{\epsilon \nabla y \cdot n_e\}, [\upsilon])_e + \sum_{e \in \Gamma_h^0 \cup \Gamma_h^D} \frac{\sigma_e \epsilon}{h_e} ([y], [\upsilon])_e
\end{aligned} \tag{2.18}$$

and the linear form $l_h^{SIPG} : V_h \to \mathbb{R}$ is given by

$$l_h^{SIPG}(\upsilon) = \sum_{E \in \xi_h} (f, \upsilon)_E + \sum_{e \in \Gamma_h^D} (g_D, -\epsilon \nabla \upsilon \cdot \mathbf{n_e} + \frac{\sigma_e \epsilon}{h_e} \upsilon)_e + \int_{\Gamma_h^N} \upsilon g_N ds. \tag{2.19}$$

Numerical fluxes $(\hat{\boldsymbol{\sigma}}, \hat{y})$ of SIPG method are consistent and conservative. Then, the method is symmetric and achieves optimal rates of convergence for both L^2 and H^1 norms. In addition, the penalty parameter σ_e must be chosen large to make the bilinear form coercive and to satisfy the convergence of the method.

2.2.2.2 The local discontinuous Galerkin method

The LDG method was introduced in [34]. For the LDG method, the numerical fluxes $(\hat{\boldsymbol{\sigma}}, \hat{y})$ in (2.14) and (2.15) are given by

$$\begin{aligned}
\hat{\boldsymbol{\sigma}} &= \{\boldsymbol{\sigma}_h\} - C_1[y_h] + C_2[\boldsymbol{\sigma}_h], \\
\hat{y} &= \{y_h\} - C_2 \cdot [y_h],
\end{aligned}$$

for the interior faces, and

$$\begin{aligned}
\hat{\boldsymbol{\sigma}} &= \boldsymbol{\sigma}_h - C_1(y_h - g_D)\mathbf{n}, \quad \hat{y} = g_D \quad \text{on} \quad \Gamma_D, \\
\hat{\boldsymbol{\sigma}} &= g_N \mathbf{n}, \qquad\qquad\qquad\quad \hat{y} = y_h \quad \text{on} \quad \Gamma_N,
\end{aligned}$$

for the boundary faces. Here, C_1 is a positive constant and C_2 is a vector which is defined for each interior face according to

$$C_2 = \frac{1}{2}(S^{E_e^2}_{E_e^1} - S^{E_e^1}_{E_e^2})\mathbf{n_e},$$

where $S^{E_e^2}_{E_e^1} \in \{0, 1\}$ is a switch which is defined for each element face. The switches always satisfy that

$$S^{E_e^2}_{E_e^1} + S^{E_e^1}_{E_e^2} = 1.$$

By using the numerical fluxes in the equations (2.14) and (2.15) with $\tau = \nabla_h \upsilon$ and lifting operators introduced in (2.16), the following system is obtained:

$$a_h^{LDG}(y_h, \upsilon) = l_h^{LDG}(\upsilon), \qquad \forall \upsilon \in V_h, \tag{2.20}$$

where the bilinear form $a_h^{LDG} : V_h \times V_h \to \mathbb{R}$ is given by

$$\begin{aligned} a_h^{LDG}(y, \upsilon) &= \int_\Omega \nabla_h \upsilon \cdot (\epsilon \nabla_h y) dx - \int_{\Gamma_h^0} ([y] \cdot \{\epsilon \nabla_h \upsilon\} + \{\epsilon \nabla_h y\} \cdot [\upsilon]) ds \\ &- \int_{\Gamma_h^0} (C_2 \cdot [y][\epsilon \nabla_h \upsilon] + [\epsilon \nabla_h y] C_2 \cdot [\upsilon]) ds + \int_{\Gamma_h^0} C_1 [y] \cdot [\upsilon] ds \\ &+ \int_\Omega \epsilon (r([y]) + l(C_2 \cdot [y]) + r_D(y)) \cdot (r([\upsilon]) + l(C_2 \cdot [\upsilon]) + r_D(\upsilon)) dx \\ &- \int_{\Gamma_h^D} (\epsilon \nabla_h y \cdot \mathbf{n} \upsilon + y \epsilon \nabla_h \upsilon \cdot \mathbf{n}) ds + \int_{\Gamma_h^D} C_1 y \upsilon ds \end{aligned} \tag{2.21}$$

and the linear form $l_h^{LDG} : V_h \to \mathbb{R}$ is given by

$$\begin{aligned} l_h^{LDG}(\upsilon) &= \int_\Omega f \upsilon dx - \int_{\Gamma_h^D} g_D (\epsilon \nabla_h \upsilon + r([\upsilon] + l(C_2 \cdot [\upsilon])) \cdot \mathbf{n} ds \\ &- \int_{\Gamma_h^D} \epsilon \upsilon r_D(g_D) \cdot \mathbf{n} ds + \int_{\Gamma_h^D} C_1 g_D \upsilon ds + \int_{\Gamma_h^N} \upsilon g_N ds, \qquad \forall \upsilon \in V_h. \end{aligned} \tag{2.22}$$

The LDG scheme is conservative, adjoint consistent and optimal convergence for L^2 and H^1 norms. However, the discretization is not compact in multiple dimensions in the sense of the connection of the nonneighboring elements. These connections are caused by the product of lifting operators in (2.21).

2.2.2.3 The compact discontinuous Galerkin method

The compact discontinuous Galerkin method was introduced in [84] by Peraire and Persson. This method is designed to overcome the compactness problem of the LDG method. Here, the

lifting operators in (2.16) are introduced slightly different. For all $e \in \Gamma_h^0$, $r^e : [L^2(e)]^d \to \Sigma_h$, $l^e : L^2(e) \to \Sigma_h$, and for each $e \in \Gamma_D$, $r_D^e : L^2(e) \to \Sigma_h$,

$$\begin{aligned}
\int_\Omega r^e(\phi) \cdot \boldsymbol{\tau} dx &= -\int_e \phi \cdot \{\boldsymbol{\tau}\} ds, \qquad \forall \boldsymbol{\tau} \in \Sigma_h, \\
\int_\Omega l^e(q) \cdot \boldsymbol{\tau} dx &= -\int_e q[\boldsymbol{\tau}] ds, \qquad \forall \boldsymbol{\tau} \in \Sigma_h, \\
\int_\Omega r_D^e(q) \cdot \boldsymbol{\tau} dx &= -\int_e q\boldsymbol{\tau} \cdot \mathbf{n} ds, \qquad \forall \boldsymbol{\tau} \in \Sigma_h.
\end{aligned} \tag{2.23}$$

Then, we will have

$$r(\phi) = \sum_{e \in \Gamma_h^0} r^e(\phi), \quad l(q) = \sum_{e \in \Gamma_h^0} l^e(q), \quad r_D(q) = \sum_{e \in \Gamma_h^\partial} r_D^e(q).$$

The numerical fluxes $(\hat{\boldsymbol{\sigma}}, \hat{y})$ are chosen such that

$$\begin{aligned}
\hat{\boldsymbol{\sigma}} &= \{\boldsymbol{\sigma}_h^e\} - C_1[y_h] + C_2[\boldsymbol{\sigma}_h^e], \\
\hat{y} &= \{y_h\} - C_2 \cdot [y_h],
\end{aligned}$$

for the interior faces, and

$$\begin{aligned}
\hat{\boldsymbol{\sigma}} &= \boldsymbol{\sigma}_h^e - C_1(y_h - g_D)\mathbf{n}, \quad \hat{y} = g_D \quad \text{on} \quad \Gamma_D, \\
\hat{\boldsymbol{\sigma}} &= g_N \mathbf{n}, \qquad\qquad\qquad\quad \hat{y} = y_h \quad \text{on} \quad \Gamma_N,
\end{aligned}$$

for the boundary faces. By using the numerical fluxes in the equations (2.14) and (2.15) with $\tau = \nabla_h \upsilon$ and lifting operators introduced in (2.23), the following system is obtained:

$$a_h^{CDG}(y_h, \upsilon) = l_h^{CDG}(\upsilon), \qquad \forall \upsilon \in V_h, \tag{2.24}$$

where the bilinear form $a_h^{LDG} : V_h \times V_h \to \mathbb{R}$ is given by

$$\begin{aligned}
a_h^{CDG}(y, \upsilon) &= \int_\Omega \nabla_h \upsilon \cdot (\epsilon \nabla_h y) dx - \int_{\Gamma_h^0} ([y] \cdot \{\epsilon \nabla_h \upsilon\} + \{\epsilon \nabla_h y\} \cdot [\upsilon]) ds \\
&- \int_{\Gamma_h^0} (C_2 \cdot [y][\epsilon \nabla_h \upsilon] + [\epsilon \nabla_h y] C_2 \cdot [\upsilon]) ds + \int_{\Gamma_h^0} C_1 [y] \cdot [\upsilon] ds \\
&+ \sum_{e \in \Gamma_h^0} \int_\Omega \epsilon (r^e([y]) + l^e(C_2 \cdot [y]) + r_D^e(y)) \cdot (r^e([\upsilon]) + l^e(C_2 \cdot [\upsilon]) + r_D^e(\upsilon)) dx \\
&- \int_{\Gamma_h^D} (\epsilon \nabla_h y \cdot \mathbf{n} \upsilon + y \epsilon \nabla_h \upsilon \cdot \mathbf{n}) ds + \int_{\Gamma_h^D} C_1 y \upsilon ds
\end{aligned} \tag{2.25}$$

and the linear form $l^{CDG} : V_h \to \mathbb{R}$ is the same as l^{LDG} given in (2.22).

The only difference between the LDG and CDG schemes is the stabilization term involving the product of the lifting functions. In spite of this difference, the CDG method inherits all the

attractive features of the LDG method. In addition, the numerical experiments in [84] indicate that the CDG scheme is slightly more stable than the LDG method and is less sensitive to the element or interface orientation.

2.3 Discontinuous Galerkin Methods for Convection Diffusion Problems

In this section, we present the discontinuous Galerkin methods for convection diffusion equations. DG scheme is given by using SIPG [5, 102], NIPG [89] and IIPG [36] methods for the diffusion term and upwind discretization [70, 86] for the convection term. Additionally, the discontinuous finite element spaces, transformations between physical elements and reference elements and basis functions are explained. Furthermore, we give error estimation of SIPG method using energy norm.

We consider the following convection diffusion problem:

$$\begin{aligned} -\epsilon\Delta y(x) + \beta(x)\cdot\nabla y(x) + r(x)y(x) &= f(x), \qquad x \in \Omega, \\ y(x) &= g_D(x), \qquad x \in \Gamma_D, \end{aligned} \tag{2.26}$$

where Ω is a bounded open, convex domain in $\mathbb{R}^2$ with boundary $\Gamma = \partial\Omega = \Gamma_D$.

Let us assume that

$$f \in L^2(\Omega),\ g_D \in H^{3/2}(\Gamma_D),\ 0 < \epsilon,\ \beta(x) \in W^{1,\infty}(\Omega)^2,\ \text{and } r \in L^\infty(\Omega). \tag{2.27a}$$

Further, we have following assumptions for $r_0 \geq 0$ and $c_* \geq 0$:

$$r(x) - \frac{1}{2}\nabla\cdot\beta(x) \geq r_0 \geq 0, \qquad x \in \Omega, \tag{2.27b}$$

$$\|-\nabla\cdot\beta(x) + r(x)\|_{L^\infty(\Omega)} \leq c_* r_0. \tag{2.27c}$$

For the finite element discretization we consider a family ξ_h, $h > 0$, of partitions of Ω into triangulations. We have two conditions for triangulations. The first one is a topological property called as conforming property or compatibility of a triangulation ξ_h. If the intersection of any two triangle E and $E^{'}$ in ξ_h is either consists of a common vertex, edge or empty. The second requirement is the geometric structure. A triangulation ξ_h is called shape regular if there exists a constant c_0 such that

$$\max_{E\in\xi_h} \frac{h_E^2}{|E|} \leq c_0,$$

where h_E is the diameter of E and $|E|$ is the area of E.

The boundary edges are decomposed into edges Γ_h^- and Γ_h^+ corresponding to the inflow and outflow boundaries, respectively:

$$\Gamma_h^- = \{x \in \partial\Omega : \ \beta(x) \cdot \mathbf{n} < 0\}, \quad \Gamma_h^+ = \{x \in \partial\Omega : \ \beta(x) \cdot \mathbf{n} \geq 0\},$$

where $\mathbf{n}$ is the outward normal to boundary of Ω at $x \in \partial\Omega$.

Similarly, the inflow and outflow boundaries of an element E are defined by

$$\partial E_h^- = \{x \in \partial E : \ \beta(x) \cdot \mathbf{n_E} < 0\}, \quad \partial E_h^+ = \{x \in \partial E : \ \beta(x) \cdot \mathbf{n_E} \geq 0\},$$

respectively, where $\mathbf{n_E}$ denotes the unit outward vector to ∂E at $x \in \partial E$.

If two elements E_1^e and E_2^e are neighbors and share one common side e, there are two traces of υ along e. Then assuming that the normal vector $\mathbf{n_e}$ is oriented from E_1^e to E_2^e, the jump and average operators for the diffusion term are given by

$$[\upsilon] = (\upsilon|_{E_1^e} - \upsilon|_{E_2^e}), \quad \{\upsilon\} = (\upsilon|_{E_1^e} + \upsilon|_{E_2^e})/2 \qquad \forall e \in \partial E_1^e \cap \partial E_2^e. \tag{2.28}$$

The upwind discretization [70, 86] is used to discretize the convection term. Hence, we define $y^{\pm}(x) = \lim_{\delta \to 0^+} y(x \pm \delta\beta)$. Here, y^+ and y^- are called the interior trace and exterior trace of y on ∂E, respectively.

2.3.1 Discontinuous Galerkin scheme

In literature, different types of discontinuous Galerkin schemes have been introduced. Some of them are shown in Table 2.1. In this thesis, we will only consider SIPG [5, 102], NIPG [89] and IIPG [36] to discretize the diffusion term and the original upwind discretization [70, 86] for the convection term.

The DG scheme for convection diffusion equations is constructed by using the fact that the solution y of (2.26) belongs to $H^s(\Omega)$ for $s > 3/2$. Then, the solution y satisfies

$$a_\kappa(y, \upsilon) + b(y, \upsilon) = l_\kappa(\upsilon), \qquad \forall \upsilon \in V_h, \tag{2.29}$$

where

$$\begin{aligned} a_\kappa(y,v) &= \sum_{E\in\xi_h}(\epsilon\nabla y,\nabla v)_E + \sum_{E\in\xi_h}(ry,v)_E \\ &+ \sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\kappa(\{\epsilon\nabla v\cdot n_e\},[y])_e - \sum_{e\in\Gamma_h^0\cup\Gamma_h^D}(\{\epsilon\nabla y\cdot n_e\},[v])_e \\ &+ \sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\frac{\sigma\epsilon}{h_e}([y],[v])_e, \end{aligned} \tag{2.30a}$$

$$b(y,v) = \sum_{E\in\xi_h}(\beta\cdot\nabla y,v)_E + \sum_{e\in\Gamma_h^0}(y^+-y^-,|n\cdot\beta|v^+)_e + \sum_{e\in\Gamma_h^-}(y^+,v^+|n\cdot\beta|)_e, \tag{2.30b}$$

$$\begin{aligned} l_\kappa(v) &= \sum_{E\in\xi_h}(f,v)_E + \sum_{e\in\Gamma_h^D}\kappa(g_D,\epsilon\nabla v\cdot\mathbf{n_e}+\frac{\sigma\epsilon}{|e|^{\beta_0}}v)_e \\ &+ \sum_{e\in\Gamma_h^-}(g_D,|\beta\cdot\mathbf{n}|v^+)_e, \end{aligned} \tag{2.30c}$$

with σ called *penalty parameter* being a nonnegative real number and β_0 being a positive number which depends on the dimension d. Depending on the choice of parameter κ in a_κ, one obtains:

- If $\kappa = -1$, the resulting DG method is called the symmetric interior penalty Galerkin (SIPG) method, introduced in the late 1970s by Wheeler and Arnold [5, 102]. When the penalty parameter σ is large enough, the method converges.
- If $\kappa = 1$ and σ is nonnegative, the resulting method is called nonsymmetric interior penalty Galerkin (NIPG) method, introduced by Rivière, Wheeler and Girault [89]. The method converges for any nonnegative values of the penalty parameter σ. The case where $\sigma = 0$ is called as Baumann-Oden method [13]. Baumann-Oden method is convergent when a higher degree ($k \geq 2$) basis are used.
- If $\kappa = 0$, the resulting method is called the incomplete interior penalty Galerkin (IIPG) method, introduced by Dawson, Sun and Wheeler [36]. This method converges under same condition as SIPG method.

Remark 2.3.1 *In standard penalization $\beta_0 = (d-1)^{-1}$, NIPG and IIPG methods have suboptimal convergence rates if the polynomial degree is even, but the convergence rate is optimal for odd polynomial degrees. By using superpenalization $\beta_0 \geq 3(d-1)^{-1}$, this shortcoming can be solved [87].*

2.3.2 Finite Element Spaces

By considering the finite dimensional subspaces of broken Sobolev space $H^s(\xi_h)$ for $s > 3/2$, we introduce the discontinuous finite element subspaces as in (2.8):

$$V_h = \{\upsilon \in L^2(\Omega) \mid \ \upsilon|_E \in P_k(E), \quad \forall E \in \xi_h\}. \tag{2.31}$$

Here, any $\upsilon \in V_h$ is called as *test function* and it is discontinuous along the faces of the mesh.

In finite element methods, the computation on the physical elements may be more difficult and costly since the elements can be very small triangles. The standard technique in FEM is to use a *reference element* instead of the physical elements.

We choose the reference triangle $\hat{E}$ with vertices $\hat{A}_1(0,0), \hat{A}_2(1,0), \hat{A}_3(0,1)$ and the physical element E is given with vertices $A_i(x_i, y_i)$ for $i = 1, 2, 3$. The invertible affine mapping $F_E : \hat{E} \to E$ is of the form

$$F_E \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad x = \sum_{i=1}^{3} x_i \hat{\phi}_i(\hat{x}, \hat{y}), \quad y = \sum_{i=1}^{3} x_i \hat{\phi}_i(\hat{x}, \hat{y}),$$

where

$$\hat{\phi}_1(\hat{x}, \hat{y}) = 1 - \hat{x} - \hat{y}, \quad \hat{\phi}_2(\hat{x}, \hat{y}) = \hat{x}, \quad \hat{\phi}_2(\hat{x}, \hat{y}) = \hat{y}.$$

Rewriting the mapping

$$\begin{pmatrix} x \\ y \end{pmatrix} = F_E \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = B_E \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + b_E,$$

where B_E is an invertible matrix and b_E is a translation vector

$$B_E = \begin{pmatrix} a_{11}^E & a_{12}^E \\ a_{21}^E & a_{22}^E \end{pmatrix} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix}, \quad b_E = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}.$$

Now, we can write the explicit representation of $F_E^{-1} : E \to \hat{E}$ by

$$F_E^{-1}(x) = B_E^{-1}(x - b_E) = \hat{x},$$

where

$$B_E^{-1} = \frac{1}{\det B_K} \begin{pmatrix} a_{22}^E & -a_{12}^E \\ -a_{21}^E & a_{11}^E \end{pmatrix} = \frac{1}{2|E|} \begin{pmatrix} \hat{a}_{11}^E & \hat{a}_{12}^E \\ \hat{a}_{21}^E & \hat{a}_{22}^E \end{pmatrix} = \frac{1}{2|E|} \hat{B}_E.$$

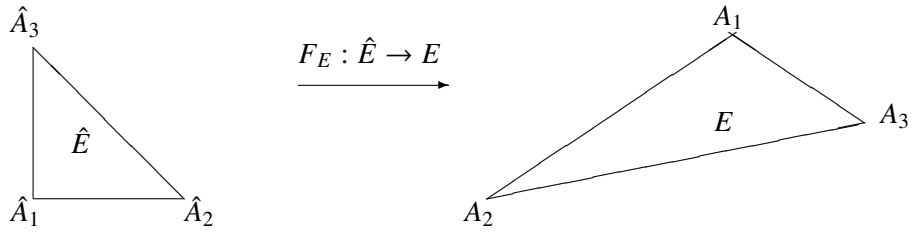


Figure 2.1: Affine transformation from Reference triangular element $\hat{E}$ to physical element E.

Then, we have

$$\begin{aligned} \hat{v}(\hat{x}, \hat{y}) &= v(x, y), \\ \hat{\nabla}\hat{v}(\hat{x}, \hat{y}) &= B_E^T \nabla v(x, y). \end{aligned} \tag{2.32}$$

2.3.3 Basis Functions

The choice of the approximation basis functions is important issue in the implementation of DG methods. The basis functions in discontinuous finite element subspace V_h have a support in a single element E since there is a lack of continuity between mesh elements. Then the subspace V_h can be written in terms of the basis functions ϕ_i^E for each element:

$$V_h = span\{\phi_i^E : 1 \leq i \leq N_{loc}, E \in \xi_h\} \tag{2.33}$$

with

$$\phi_i^E(x) = \begin{cases} \hat{\phi}_i \circ F_E(x), & \text{if} \quad x \in E, \\ 0, & \text{if} \quad x \notin E. \end{cases}$$

where the local basis functions $(\hat{\phi}_i)_{1 \leq i \leq N_{loc}}$ and the local dimension $N_{loc} = \frac{(k+1)(k+2)}{2}$ with k is total polynomial degree are defined on the reference element $\hat{E}$.

The flexibility of DG methods allows to easily change basis functions. Therefore, we use two different basis functions satisfying a desired orthogonality, i.e., monomial polynomial basis, and Dubiner polynomial basis [39]. The latter one is more accurate for higher order basis functions whereas the implementation of former ones is easy.

Monomial Polynomial Basis

We use monomial polynomial basis due to the easy implementation. In two dimension, we have

$$\hat{\phi}_i(\hat{x}, \hat{y}) = \hat{x}^I \hat{y}^J, \qquad I + J = i, \qquad 0 \leq i \leq k.$$

For instance, we have the following:

- Piecewise linear:

$$\hat{\phi}_0(\hat{x}, \hat{y}) = 1, \qquad \hat{\phi}_1(\hat{x}, \hat{y}) = \hat{x}, \qquad \hat{\phi}_2(\hat{x}, \hat{y}) = \hat{y}.$$

- Piecewise quadratics:

$$\hat{\phi}_0(\hat{x}, \hat{y}) = 1, \qquad \hat{\phi}_1(\hat{x}, \hat{y}) = \hat{x}, \qquad \hat{\phi}_2(\hat{x}, \hat{y}) = \hat{y}.$$

$$\hat{\phi}_0(\hat{x}, \hat{y}) = \hat{x}^2, \qquad \hat{\phi}_1(\hat{x}, \hat{y}) = \hat{x}\hat{y}, \qquad \hat{\phi}_2(\hat{x}, \hat{y}) = \hat{y}^2.$$

Dubiner polynomial basis

We can also use Dubiner basis [39] to compute the integrals on the reference elements since they yield more accurate for higher order basis function. By transforming the Jacobi polynomials defined on intervals to form polynomials on triangles, the Dubiner basis on triangles are obtained.

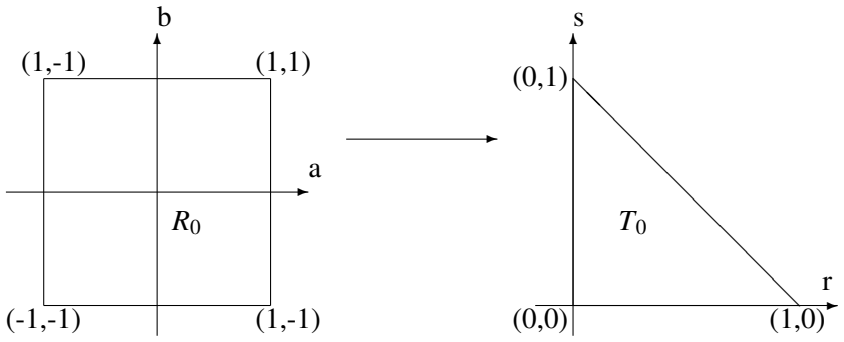


Figure 2.2: The mapping between the square R_0 and the triangle T_0.

To construct an orthogonal basis on the triangle T_0 whose vertices are (0,0), (1,0) and (0,1), we follow the idea in [94] and consider the transformation in Figure 2.2 between the reference

square R_0 whose vertices (-1,-1), (1,-1), (1,1), (-1,1) and the reference triangle T_0 defined by

$$r = \frac{(1+a)(1-b)}{4}, \quad s = \frac{1+b}{2} \quad \text{or} \quad a = \frac{2r}{1-s} - 1, \quad b = 2s - 1.$$

Then, by using a generalized tensor product of the Jacobi polynomials on the interval $[-1, 1]$ to form a basis on the square R_0, which is then transformed by the above collapsing mapping to a basis on the triangle T_0, the Dubiner basis are constructed. The Dubiner basis on the triangle T_0 can be defined as

$$\begin{aligned} g_{mn}(r,s) &= P_m^{0,0}(a)(1-b)^m P_n^{2m+1,0}(b) \\ &= 2^m P_m^{0,0}(\frac{2r}{1-s} - 1)(1-s)^m P_n^{2m+1,0}(2s-1) \quad 0 \le m, n, m+n \le N_{loc}. \end{aligned}$$

The first six un-normalized Dubiner basis functions on the triangle T_0 are

$$\begin{aligned} g_{00}(r,s) &= 1, \\ g_{10}(r,s) &= 4r + 2s - 2, \\ g_{01}(r,s) &= 3s - 1, \\ g_{20}(r,s) &= 24r^2 + 24rs + 4s^2 - 24r - 8s + 4, \\ g_{11}(r,s) &= 20rs + 10s^2 - 4r - 12s + 2, \\ g_{02}(r,s) &= 10s^2 - 8s + 1. \end{aligned}$$

2.3.4 Derivative Transformations

We also need to express the partial derivatives on $\hat{E}$ instead of E since the quadrature is performed over the reference element $\hat{E}$. Now, we will express the first and second order partial derivatives using the basis defined on the reference element $\hat{E}$.

First-Order Partial Derivatives

$$\frac{\partial \phi_j}{\partial x} = \frac{\partial \hat{\phi}_j}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial \hat{\phi}_j}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x} = \frac{1}{2|E|}(\frac{\partial \hat{\phi}_j}{\partial \hat{x}}\hat{a}_{11}^E + \frac{\partial \hat{\phi}_j}{\partial \hat{y}}\hat{a}_{21}^E)$$

and

$$\frac{\partial \phi_j}{\partial y} = \frac{\partial \hat{\phi}_j}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial \hat{\phi}_j}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y} = \frac{1}{2|E|}(\frac{\partial \hat{\phi}_j}{\partial \hat{x}}\hat{a}_{12}^E + \frac{\partial \hat{\phi}_j}{\partial \hat{y}}\hat{a}_{22}^E).$$

Then, we obtain

$$\partial_n \phi_j = \frac{1}{2|E|}[(\frac{\partial \hat{\phi}_j}{\partial \hat{x}}\hat{a}_{11}^E + \frac{\partial \hat{\phi}_j}{\partial \hat{y}}\hat{a}_{21}^E)n_x + (\frac{\partial \hat{\phi}_j}{\partial \hat{x}}\hat{a}_{12}^E + \frac{\partial \hat{\phi}_j}{\partial \hat{y}}\hat{a}_{22}^E)n_y].$$

Second-Order Partial Derivatives

$$\begin{aligned}\frac{\partial^2\phi_j}{\partial x^2} &= \frac{\partial}{\partial x}(\frac{\partial\hat{\phi}_j}{\partial x}) = \frac{\partial}{\partial\hat{x}}(\frac{\partial\hat{\phi}_j}{\partial x})\frac{\partial\hat{x}}{\partial x} + \frac{\partial}{\partial\hat{y}}(\frac{\partial\hat{\phi}_j}{\partial x})\frac{\partial\hat{y}}{\partial x} \\ &= \left(\frac{1}{2|E|}\right)^2 [\hat{a}_{11}^2\frac{\partial^2\hat{\phi}_j}{\partial\hat{x}^2} + 2\hat{a}_{11}\hat{a}_{12}\frac{\partial^2\hat{\phi}_j}{\partial\hat{x}\partial\hat{y}} + \hat{a}_{21}^2\frac{\partial^2\hat{\phi}_j}{\partial\hat{y}^2}]\end{aligned}$$

and

$$\begin{aligned}\frac{\partial^2\phi_j}{\partial y^2} &= \frac{\partial}{\partial y}(\frac{\partial\hat{\phi}_j}{\partial y}) = \frac{\partial}{\partial\hat{x}}(\frac{\partial\hat{\phi}_j}{\partial y})\frac{\partial\hat{x}}{\partial y} + \frac{\partial}{\partial\hat{y}}(\frac{\partial\hat{\phi}_j}{\partial y})\frac{\partial\hat{y}}{\partial y} \\ &= \left(\frac{1}{2|E|}\right)^2 [\hat{a}_{12}^2\frac{\partial^2\hat{\phi}_j}{\partial\hat{x}^2} + 2\hat{a}_{12}\hat{a}_{22}\frac{\partial^2\hat{\phi}_j}{\partial\hat{x}\partial\hat{y}} + \hat{a}_{22}^2\frac{\partial^2\hat{\phi}_j}{\partial\hat{y}^2}].\end{aligned}$$

Hence, the Laplacian is defined by

$$\begin{aligned}\Delta\phi_j &= \frac{\partial^2\phi_j}{\partial x^2} + \frac{\partial^2\phi_j}{\partial y^2} \\ &= \left(\frac{1}{2|E|}\right)^2 [(\hat{a}_{11}^2 + \hat{a}_{12}^2)\frac{\partial^2\hat{\phi}_j}{\partial\hat{x}^2} + 2(\hat{a}_{11}\hat{a}_{12} + \hat{a}_{12}\hat{a}_{22})\frac{\partial^2\hat{\phi}_j}{\partial\hat{x}\partial\hat{y}} + (\hat{a}_{21}^2 + \hat{a}_{22}^2)\frac{\partial^2\hat{\phi}_j}{\partial\hat{y}^2}].\end{aligned}$$

2.3.5 Numerical Quadrature

It is important to have high order quadrature rules since DG methods easily allow for high order approximations. Hence, we use the numerical quadrature [42] to approximate the integrals:

$$\int_{\hat{E}}\hat{\upsilon} \approx \sum_{j=1}^{Q_D} w_j\hat{\upsilon}(s_{x,j}, s_{y,j}),$$

where the set of weights w_j and nodes $(s_{x,j}, s_{y,j}) \in \hat{E}$. By using the affine transformation F_E, we obtain

$$\begin{aligned}\int_E \upsilon &= \int_{\hat{E}} \upsilon\circ F_E \det(B_E) = 2|E|\int_{\hat{E}}\hat{\upsilon} \approx 2|E|\sum_{j=1}^{Q_D} w_j\hat{\upsilon}(s_{x,j}, s_{y,j}), \\ \int_E \nabla\upsilon\cdot\mathbf{w} &\approx 2|E|\sum_{j=1}^{Q_D} w_j(B_E^T)^{-1}\hat{\nabla}\hat{\upsilon}(s_{x,j}, s_{y,j})\cdot\hat{\mathbf{w}}(s_{x,j}, s_{y,j}), \\ \int_E \nabla\upsilon\cdot\nabla w &\approx 2|E|\sum_{j=1}^{Q_D} w_j(B_E^T)^{-1}\hat{\nabla}\hat{\upsilon}(s_{x,j}, s_{y,j})\cdot(B_E^T)^{-1}\hat{\nabla}\hat{w}(s_{x,j}, s_{y,j}).\end{aligned}$$

2.3.6 Error Analysis

In this section, we give a convergence analysis of the SIPG method for convection diffusion equations (2.26). See [7, 63, 87, 89] for other DG methods. We first show the coercivity of

the bilinear form $a_\kappa(y,v) + b(y,v)$. Then, error estimation is given in Theorem 3.1.1 for the constant convection coefficient β as in [46].

Coercivity is shown separately for $a_{-1}(v,v)$ and $b(v,v)$. We define the diffusion norm such that

$$\|v\|^2_{diff} = \sum_{E\in\xi_h}(\epsilon\|\nabla y\|^2_E + r_0\|y\|^2_E) + \sum_{e\in\Gamma_h}\frac{\epsilon}{h_e}\sigma\|[y]\|^2_e$$

and $a_{-1}(v,v)$ is given such that

$$a_{-1}(v,v) = \sum_{E\in\xi_h}\int_E \epsilon(\nabla v)^2 + rv^2 - 2\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\int_e\{\epsilon\nabla v\cdot n_e\}[v] + \frac{\sigma\epsilon}{h_e}[v]^2.$$

By using Cauchy-Schwarz inequality, we obtain an upper bound for $\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\int_e\{\epsilon\nabla v\cdot n_e\}[v]$:

$$\begin{aligned}\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\int_e\{\epsilon\nabla v\cdot n_e\}[v] &\leq \sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\|\{\epsilon\nabla v\cdot n_e\}\|_{L^2(e)}\|[v]\|_{L^2(e)}\\ &\leq \sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\|\{\epsilon\nabla v\cdot n_e\}\|_{L^2(e)}\left(\frac{1}{|e|^{\beta_0}}\right)^{1/2-1/2}\|[v]\|_{L^2(e)}.\end{aligned}$$

The average of fluxes for $e\in\Gamma_h^0$ and trace inequality (2.1b) give us

$$\begin{aligned}\|\{\epsilon\nabla v\cdot n_e\}\|_{L^2(e)} &\leq \frac{1}{2}\|(\epsilon\nabla v\cdot n_e)|_{E_1^e}\|_{L^2(e)} + \frac{1}{2}\|(\epsilon\nabla v\cdot n_e)|_{E_2^e}\|_{L^2(e)}\\ &\leq \frac{C_t\epsilon}{2}h_{E_1^e}^{-1/2}\|\nabla v\|_{L^2(E_1^e)} + \frac{C_t\epsilon}{2}h_{E_2^e}^{-1/2}\|\nabla v\|_{L^2(E_2^e)}.\end{aligned}$$

Assuming h be maximum element diameter, $h = \max(h_E)$, we have $|e| \leq h_E^{d-1} \leq h^{d-1}$. Then, we obtain

$$\begin{aligned}\int_e\{\epsilon\nabla v\cdot n_e\}[v] &\leq \frac{C_t\epsilon}{2}|e|^{\beta_0/2}\left(h_{E_1^e}^{-1/2}\|\nabla v\|_{L^2(E_1^e)} + h_{E_2^e}^{-1/2}\|\nabla v\|_{L^2(E_2^e)}\right)\left(\frac{1}{|e|^{\beta_0}}\right)^{1/2}\|[v]\|_{L^2(e)}\\ &\leq \frac{C_t\epsilon}{2}\left(h_{E_1^e}^{\frac{\beta_0}{2}(d-1)-\frac{1}{2}} + h_{E_2^e}^{\frac{\beta_0}{2}(d-1)-\frac{1}{2}}\right)\left(\|\nabla v\|^2_{L^2(E_1^e)} + \|\nabla v\|^2_{L^2(E_2^e)}\right)^{1/2}\left(\frac{1}{|e|^{\beta_0}}\right)^{1/2}\|[v]\|_{L^2(e)}\\ &\leq C_t\epsilon\left(\|\nabla v\|^2_{L^2(E_1^e)} + \|\nabla v\|^2_{L^2(E_2^e)}\right)^{1/2}\left(\frac{1}{|e|^{\beta_0}}\right)^{1/2}\|[v]\|_{L^2(e)}\end{aligned}$$

if β_0 satisfies the condition $\beta_0(d-1) \geq 1$. A similar bound can be obtained for boundary edges. With the maximum number of neighbors of a triangle element in a conforming mesh

which equals to 3, we obtain

$$\begin{aligned}\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\int_e\{\epsilon\nabla v\cdot n_e\}[v] &\leq C_t\epsilon\left(\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\frac{1}{|e|^{\beta_0}}\|[v]\|^2_{L^2(e)}\right)^{1/2}\\ &\times\left(\sum_{e\in\Gamma_h^0}\|\nabla v\|^2_{L^2(E_1^e)}+\|\nabla v\|^2_{L^2(E_2^e)}+\sum_{e\in\Gamma_h^D}\|\nabla v\|^2_{L^2(E_1^e)}\right)^{1/2}\\ &\leq C_t\epsilon\sqrt{3}\left(\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\frac{1}{|e|^{\beta_0}}\|[v]\|^2_{L^2(e)}\right)^{1/2}\left(\sum_{E\in\xi_h}\|\nabla v\|^2_{L^2(E)}\right)^{1/2}.\end{aligned}$$

Using Young's inequality, we obtain for $\gamma > 0$

$$\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\int_e\{\epsilon\nabla v\cdot n_e\}[v]\leq\frac{\gamma}{2}\sum_{E\in\xi_h}\|\sqrt{\epsilon}\nabla v\|^2_{L^2(E)}+\frac{C_t^2\epsilon\sqrt{3}}{2\gamma}\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\frac{1}{|e|^{\beta_0}}\|[v]\|^2_{L^2(e)}.$$

Hence, by (2.27b) we obtain,

$$a_{-1}(v,v)\geq(1-\gamma)\sum_{E\in\xi_h}\|\sqrt{\epsilon}\nabla v\|^2_{L^2(E)}+\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\frac{\sigma-\frac{C_t^2\epsilon\sqrt{3}}{\gamma}}{|e|^{\beta_0}}\|[v]\|^2_{L^2(e)}.$$

Choosing $\gamma = 1/2$ and σ large enough, we have

$$a_{-1}(v,v)\geq\frac{1}{2}\|v\|^2_{diff}. \tag{2.34}$$

The following equality holds [63]:

$$b(v,v)=\frac{1}{2}\sum_{e\in\Gamma_h^-}|\mathbf{n_e}\cdot\beta|\|v^+\|^2_e+\frac{1}{2}\sum_{e\in\Gamma_h^+}|\mathbf{n_e}\cdot\beta|\|v^-\|^2_e+\frac{1}{2}\sum_{e\in\Gamma_h^0}|\mathbf{n_e}\cdot\beta|\|v^+-v^-\|^2_e. \tag{2.35}$$

By using (2.34) and (2.35), coercivity is shown

$$a_{-1}(v,v)+b(v,v)\geq\|v\|^2_{DG},\qquad\forall v\in V_h \tag{2.36}$$

with the DG norm given by

$$\begin{aligned}\|v\|^2_{DG} &= \sum_{E\in\xi_h}(\epsilon\|\nabla y\|^2_E+r_0\|y\|^2_E)+\sum_{e\in\Gamma_h}\frac{\epsilon}{h_e}\sigma\|[y]\|^2_e\\ &+ \frac{1}{2}\sum_{e\in\Gamma_h^-}|\mathbf{n_e}\cdot\beta|\|y^+\|^2_e+\frac{1}{2}\sum_{e\in\Gamma_h^+}|\mathbf{n_e}\cdot\beta|\|y^-\|^2_e+\frac{1}{2}\sum_{e\in\Gamma_h^0}|\mathbf{n_e}\cdot\beta|\|y^+-y^-\|^2_e.\end{aligned} \tag{2.37}$$

In [59] the error estimate using nonsymmetric interior penalty Galerkin (NIPG) method has been studied for convection diffusion equations with a similar norm defined in (2.37). However, this estimate is not strong enough for convection dominated problems. Therefore, to

control the gradient of error in the direction of β as ϵ tends to zero, a stronger norm is defined in [46] for the convection dominated problems such as

$$|||v|||^2 \quad = \quad \|v\|_{DG}^2 + \sum_{E\in\xi_h} h_E \|\beta\cdot\nabla v\|_E^2, \qquad v \in V_h. \tag{2.38}$$

Now, we give the error estimate of the SIPG method using the norm in (2.38) and Lemma 2.3.2.

Lemma 2.3.2 *[46, Lemma 6.1] There exist constants C and C_1 such that for all $v \in V_h$*

$$C|||v|||^2 \le a_{-1}(v, C_1 v + h\beta\cdot\nabla v) + b(v, C_1 v + h\beta\cdot\nabla v).$$

Theorem 2.3.3 *[46, Theorem 5.1] Let us assume that y and y_h are solutions of (2.26) and (2.29), respectively. Then, the following error estimate holds for $y \in H^s(\Omega)$ with $3/2 < s$:*

$$|||y - y_h||| \le C \max(\sqrt{\epsilon}h^{s-1}, h^{s-1/2})|y|_{H^s(\Omega)},$$

where a positive constant C is independent of ϵ and h.

Proof. The continuous interpolant $\tilde{y}$ of y [93] satisfies the following approximation property

$$\|y - \tilde{y}\|_{L^2(\Omega)} + h|y - \tilde{y}|_{H^1(\Omega)} \le Ch^s|y|_{H^s(\Omega)}. \tag{2.39}$$

Using the trace inequality, we obtain

$$|||y - \tilde{y}||| \le C(\sqrt{\epsilon}h^{s-1} + h^{s-1/2})|y|_{H^s(\Omega)}. \tag{2.40}$$

Let $E_h = y_h - \tilde{y}$, then

$$|||y - y_h||| \le |||y - \tilde{y}||| + |||E_h|||.$$

We need a bound for E_h similar to that in (2.40). By the orthogonal Galerkin property, we have

$$a_{-1}(y - y_h, w_h) + b(y - y_h, w_h) = 0, \quad \forall w_h \in V_h.$$

Denoting $E_h^\beta = C_1 E_h + h\beta\cdot\nabla E_h$ and then, by Lemma 2.3.2 and Galerkin orthogonality, we have

$$\begin{aligned} C|||E_h|||^2 &\le a_{-1}(E_h, E_h^\beta) + b(E_h, E_h^\beta) \\ &= a_{-1}(y - \tilde{y}, E_h^\beta) + b(y - \tilde{y}, E_h^\beta). \end{aligned} \tag{2.41}$$

To estimate the first term on the right hand side of (2.30), we use Cauchy-Schwarz inequalty and the continuity property of $y - \tilde{y}$:

$$\begin{aligned} a_{-1}(y-\tilde{y}, E_h^\beta) &\leq \epsilon |y-\tilde{y}|_{H^1(\Omega)} \left(\sum_{E\in\xi_h} \|\nabla E_h^\beta\|^2_{L^2(E)} \right)^{1/2} \\ &+ \left(\sum_{e\in\Gamma_h^0\cup\Gamma_h^D} h \|\{\epsilon\nabla(y-\tilde{y})\cdot n_e\}\|^2_{L^2(e)} + \sum_{e\in\Gamma_h^\partial} h^{-1}\|y-\tilde{y}\|^2_{L^2(e)} \right)^{1/2} \left(\sum_{e\in\Gamma_h^0\cup\Gamma_h^D} h^{-1}\|[E_h^\beta]\|^2_{L^2(e)} \right)^{1/2}. \end{aligned}$$

A Bramble-Hilbert argument [25] shows that

$$h \sum_{e\in\Gamma_h^0\cup\Gamma_h^D} \|\{\epsilon\nabla(y-\tilde{y})\cdot n_e\}\|^2_{L^2(e)} \leq \epsilon h^{2s-2}|y|^2_{H^s(\Omega)}.$$

Local inverse inequalities imply

$$\begin{aligned} \sum_{E\in\xi_h} \|\nabla E_h^\beta\|^2_{L^2(E)} &\leq C \sum_{E\in\xi_h} (\|\nabla E_h\|^2_{L^2(E)} + \|\beta\cdot\nabla E_h\|^2_{L^2(E)}), \\ \sum_{e\in\Gamma_h^0\cup\Gamma_h^D} h^{-1}\|[E_h^\beta]\|^2_{L^2(e)} &\leq C \sum_{e\in\Gamma_h^0\cup\Gamma_h^D} h^{-1}\|[E_h]\|^2_{L^2(e)} + C\sum_{E\in\xi_h} \|\beta\cdot\nabla E_h\|^2_{L^2(e)}. \end{aligned}$$

These estimates together with a trace theorem and (2.39) yield

$$a_{-1}(y-\tilde{y}, E_h^\beta) \leq C\epsilon h^{s-1}|y|_{H^s(\Omega)} a_{-1}(E_h, E_h)^{1/2}. \tag{2.42}$$

Now, we estimate the second term on the right hand side of (2.41). Integration by parts gives us

$$b(u,\upsilon) = (u, -\beta\cdot\nabla\upsilon) + (u^-, \upsilon^- - \upsilon^+)_{\Gamma_h^0} + (u^-, \upsilon^+)_{\Gamma_h^+}.$$

Then,

$$b(u, C_1\upsilon + h\beta\cdot\nabla\upsilon) \leq C \left(h^{-1/2}\|u\|_{L^2(\Omega)} + (u^-, u^-)^{1/2}_{\Gamma_h^0} + (\sum_{E\in\xi_h} h\|\beta\cdot\nabla u\|^2)^{1/2} + b(u,u)^{1/2} \right) |||\upsilon|||.$$

Using $u = y - \tilde{y}$, $\upsilon = E_h$, trace inequality (2.1b) and (2.39), we obtain that

$$b(y-\tilde{y}, E_h^\beta) \leq Ch^{s-1/2}|y|_{H^s(\Omega)}|||E_h|||. \tag{2.43}$$

Combining (2.42) and (2.43) and using it in (2.41), we have

$$|||E_h||| \leq C(\sqrt{\epsilon}h^{s-1} + h^{s-1/2})|y|_{H^s(\Omega)}.$$

This together with (2.40) proves the theorem. ■

CHAPTER 3

DISTRIBUTED OPTIMAL CONTROL PROBLEMS

Many real-life applications such as shape optimization of technological devices [80], optimal control of systems [47], identification of parameters in environmental processes, and flow control problems [37, 44, 83] lead to optimization governed by systems of convection diffusion partial differential equations (PDEs). An appropriate mathematical treatment of PDE constrained optimization problems requires the integrated use of advanced methodologies from the theory of optimization and optimal control in a functional setting, the theory of PDEs as well as the development and implementation of powerful algorithmic tools from numerical mathematics and scientific computing.

To solve the optimal control problems numerically, there exist two different approach, *discretize-then-optimize* and *optimize-then-discretize*. In the former one, the optimal control problem is first discretized using a suitable numerical method and then the resulting finite dimensional optimization problem is solved. In the latter one, one first computes the infinite dimensional optimality system and then the optimality system is discretized. Both approaches were studied in [1, 18, 21, 22, 35, 38, 54, 104] for optimal control problems.

It is known that when pure Galerkin finite element discretization is used, both approaches lead to same discretization schemes. When the conventional residual based nonsymmetric stabilized finite element method SUPG (streamline upwind Petrov-Galerkin method) is used on the discretization of optimal control problems governed by convection diffusion equations as in [35], the *discretize-then-optimize* and the *optimize-then-discretize* lead to different discretization schemes. In [35], it was shown that when SUPG discretization is used, *optimize-then-discretize* has better asymptotic convergence properties. For both approaches, the difference in solution is small when piecewise linear polynomials are used for discretization of state,

adjoint and controls but they can be significant for higher order finite elements. The largest difference is observed in the adjoint variable. In the *discretize-then-optimize* approach, although the discretized state equation is strongly consistent, the discrete adjoint and gradient equations are inconsistent. The reason of this lack of consistency is that the discrete adjoint equation is not a method of weighted residuals for continuous adjoint problem. However, the discretized state, adjoint and gradient equations are strongly consistent in the *optimize-then-discretize* approach. In addition, *optimize-then-discretize* approach leads to a nonsymmetric system.

The approach proposed by Dedè and Quarteroni in [38] is based on a stabilization method applied to the Lagrangian functional, rather than stabilizing the state and adjoint equations, separately. If the Lagrangian of the optimal control problem is given in (3.8), then the stabilized Lagrangian is defined as

$$L_h(y, u, p) = L(y, u, p) + S_h(y, u, p),$$

where

$$S_h(y, u, p) = \sum_{E \in \xi_h} \delta_E \int_E R^s(y, u) R^a(p, y).$$

The terms $R^s(y, u)$ and $R^a(p, y)$ represent the residuals of the state and adjoint equations, respectively, and δ_E is a stabilization parameter depending on the Pèclet number. This stabilization method yields a coherence between state and adjoint stabilized equations. However, the same coherence is not obtained by stabilizing directly the state equation by means of a strongly consistent method like GLS (Galerkin least squares). In [1], Galerkin/Least-Squares (GLS) stabilization has been used for the optimal boundary control problems governed by the Oseen equation but significant differences are observed between both approaches.

Recently, new stabilization techniques like local projection (LPS) and edge-stabilization have been developed which are symmetric and not residual-based. The method presented in [18] uses the standard finite element discretization with stabilization based on local projections (LPS method). The stabilization term in this method has the following form:

$$S_h^{\delta}(y_h, \upsilon_h) = \delta(\beta \cdot \nabla y_h - \pi_h(\beta \cdot \nabla y_h), \beta \cdot \nabla \upsilon_h - \pi_h(\beta \cdot \nabla \upsilon_h),$$

where δ is a positive stabilization parameter and π_h is an L^2-orthogonal projection operator. Then, the discrete semilinear form is

$$a_h(y_h, \upsilon_h) = a(y_h, \upsilon_h) + S_h^{\delta}(y_h, \upsilon_h).$$

Hence,

$$a_h(y_h, \upsilon_h) = (f + u_h, \upsilon_h).$$

In [104], Yan and Zhou have used the edge stabilization Galerkin approximation for the numerical solution of constraint optimal control problem governed by convection dominated problems. Here, least-squares stabilization of the gradient jumps across element edges is used. The stabilization form S is given as

$$S(y_h, \upsilon_h) = \sum_{e \in \Gamma_h^0} \int_e \gamma h_e^2 [\mathbf{n}_e \cdot \nabla y_h][\mathbf{n}_e \cdot \nabla \upsilon_h],$$

where γ is constant independent of h_e and $[q]$ denotes the jump of q for all interior edges, $e \in \Gamma_h^0$.

The LPS method [18] and edge stabilization [104] are symmetric stabilization methods. Then, both approaches, the *discretize-then-optimize* and the *optimize-then-discretize*, lead to same discrete equations. In addition, Braack in [21] has applied the symmetric stabilization methods, LPS and edge-stabilization for the optimal control problem governed by Oseen equation. See also [77] for the numerical analysis of quadratic optimal control problems with distributed and Robin boundary control governed by an elliptic problem using the local projection approach (LPS method). A comparison of symmetric stabilization methods (LPS and edge-stabilization) and residual-based techniques can be found in [22].

The studies have shown that the solution of optimization problems governed by convection diffusion PDEs (also referred to as the state PDEs) provides an additional challenges since the optimality conditions for such optimization problems do not only involve the convection diffusion state equation (3.2), but also another convection diffusion PDE (3.10), the so-called *adjoint PDE*. The diffusion part of the adjoint PDE is equal to that of the state PDE, but the convection in the adjoint PDE is equal to the negative of the convection in the state PDE. Therefore, it is difficult to find a suitable method to solve the optimal control problems governed by convection diffusion equations. DG methods have higher accuracy and work better in complex geometries due to their local nature in constraint to standard continuous Galerkin methods. In addition, they have a weak treatment for the boundary conditions in contrast to SUPG. With these properties of DG methods, we examine whether a DG discretization of the optimal control problem leads to the same result as the same DG discretization applied to the optimality system of unconstrained optimal control problem governed by convection

diffusion equations as in [72]. Additionally, we survey the effect of superpenalization form of nonsymmeteric DG methods on optimal control problems.

This chapter extends DG methods introduced in previous chapter for single convection diffusion equations to the unconstrained optimal control problems. We firstly introduce the optimal control problems governed by convection diffusion equations. Then, two numerical approaches, the *discretize-then-optimize* and the *optimize-then-discretize*, to solve the optimal control problem are presented and compared by using DG methods.

3.1 Introduction

Let Ω be a bounded open, convex domain in $\mathbb{R}^2$ and $\Gamma = \partial\Omega$. We consider the following linear-quadratic optimal control problem:

$$\text{minimize} \quad J(y,u) := \frac{1}{2}\int_\Omega (y(x)-y_d(x))^2 dx + \frac{\omega}{2}\int_\Omega u(x)^2 dx \tag{3.1}$$

subject to

$$-\epsilon\Delta y(x) + \beta(x)\cdot\nabla y(x) + r(x)y(x) = f(x)+u(x), \qquad x\in\Omega, \tag{3.2a}$$

$$y(x) = g_D(x), \qquad x\in\Gamma, \tag{3.2b}$$

where f, β, r, y_d, g_D are given functions, diffusion and regularization parameters $\epsilon, \omega > 0$ are given scalars.

We define the state and control space

$$Y = \{y\in H^1(\Omega): \quad y = g_D \text{ on } \Gamma\}, \qquad U = L^2(\Omega), \tag{3.3}$$

and space of the test functions

$$V = \{\upsilon\in H^1(\Omega): \quad \upsilon = 0 \text{ on } \Gamma\}. \tag{3.4}$$

Then, the weak form of the state equation is

$$a(y,\upsilon) + b(y,\upsilon) = (f,\upsilon), \qquad \forall\upsilon\in V, \tag{3.5}$$

where

$$\begin{aligned} a(y,\upsilon) &= \int_\Omega (\epsilon\nabla y\cdot\nabla\upsilon + \beta\cdot\nabla y\upsilon + ry\upsilon)\,dx, \\ b(y,\upsilon) &= -\int_\Omega u\upsilon\,dx, \qquad (f,\upsilon) = \int_\Omega f\upsilon\,dx. \end{aligned}$$

Then, our problem (3.1)-(3.2) can be written as

$$\text{minimize} \quad J(y,u) := \frac{1}{2}\|y - y_d\|_\Omega^2 + \frac{\omega}{2}\|u\|_\Omega^2 \tag{3.6a}$$

$$\begin{aligned} \text{subject to} \quad a(y,\upsilon) \quad &+ \quad b(u,\upsilon) = (f,\upsilon), \qquad \forall \upsilon \in V, \\ (y,u) \quad &\in \quad Y \times U. \end{aligned} \tag{3.6b}$$

In order to show that the optimal control problem (3.1)-(3.2) has an unique solution, we need the following assumptions:

$$f, y_D \in L^2(\Omega),\ g_D \in H^{3/2}(\Gamma),\ 0 < \epsilon,\ \beta(x) \in W^{1,\infty}(\Omega)^2,\ 0 < \omega \text{ and } r \in L^\infty(\Omega). \tag{3.7a}$$

Further, we have following assumptions for $r_0 \geq 0$ and $c_* \geq 0$:

$$r(x) - \frac{1}{2}\nabla \cdot \beta(x) \quad \geq \quad r_0 \geq 0, \qquad x \in \Omega, \tag{3.7b}$$

$$\| - \nabla \cdot \beta(x) + r(x)\|_{L^\infty(\Omega)} \quad \leq \quad c_* r_0. \tag{3.7c}$$

The conditions (3.7a,3.7b) ensure the well-posedness of the optimal control problem [48, 73]. The condition (3.7c) [91] is needed in next chapters to show efficiency of error estimator in a posteriori error analysis.

By [76, Sec. II.1], under the assumptions defined in (3.7) the existence of a unique solution $(y,u) \in Y \times U$ of (3.1)-(3.2) is guaranteed and necessary and sufficient optimality conditions are provided. The necessary and sufficient optimality conditions are obtained using the Lagrangian of the optimal control problem:

$$L(y,u,p) = \frac{1}{2}\|y - y_d\|_{L^2(\Omega)}^2 + \frac{\omega}{2}\|u_h\|_{L^2(\Omega)}^2 + a(y,p) + b(u,p) - (f,p). \tag{3.8}$$

By setting the partial derivatives of Lagrangian with respect to state y, control u and adjoint p equal to zero, we obtain the following optimality system consisting of the adjoint equation, the gradient equation and the state equation, respectively:

$$a(\psi,p) \quad = \quad -(y - y_d, \psi), \qquad \forall \psi \in V, \tag{3.9a}$$

$$b(w,p) + \omega(u,w) \quad = \quad 0, \qquad \forall w \in U, \tag{3.9b}$$

$$a(y,\upsilon) + b(u,\upsilon) \quad = \quad (f,\upsilon), \qquad \forall \upsilon \in V. \tag{3.9c}$$

Equation (3.9a) can be interpreted as the weak form of a convection diffusion equation, but convection is now given by $-\beta$:

$$\begin{aligned} -\epsilon\nabla p(x) - \beta(x)\cdot\nabla p(x) + (r(x) - \nabla\cdot\beta(x))p(x) \quad &= \quad -(y(x) - y_d(x)), \qquad x \in \Omega, \\ p(x) \quad &= \quad 0, \qquad\qquad\qquad\qquad x \in \Gamma, \end{aligned} \tag{3.10}$$

and equation (3.9b) corresponds to gradient equation

$$p(x) = \omega u(x). \tag{3.11}$$

Then, the following Theorem 3.1.1 can be stated by the theory in [76, Sec. II.1].

Theorem 3.1.1 *If the assumptions (3.7) are satisfied, then the optimal control problem (3.6) has a unique solution* $(y, u) \in Y \times U$. *The functions* $(y, u) \in Y \times U$ *solve (3.6) if and only if* $(y, u, p) \in Y \times U \times Y$ *is unique solution for the following optimality system:*

$$\begin{aligned} a(\psi, p) + (y, \psi) &= (y_d, \psi), \qquad \forall \psi \in Y, && (3.12a)\\ b(w, p) + \omega(u, w) &= 0, \qquad \forall w \in U, && (3.12b)\\ a(y, \upsilon) + b(u, \upsilon) &= (f, \upsilon), \qquad \forall \upsilon \in Y. && (3.12c) \end{aligned}$$

For the numerical solution of the optimal control problem (3.1)-(3.2), there exist two different approaches. In the *discretize-then-optimize*, first the state equation is discretized and then the optimality conditions for the finite dimensional system are derived. In the *optimize-then-discretize*, the optimality conditions are formulated on the continuous level for the state, adjoint and gradient equations, then these equations are discretized. Now, we will show whether the optimality system of DG discretized optimal control problem is equivalent to the DG discretization of the optimality system, or not.

3.2 Discretize-then-Optimize Approach

In this approach, the optimal control problem is first discretized, using the DG method for the discretization of the state convection diffusion equation, and then the resulting finite dimensional optimization problem is solved by using a suitable optimization algorithm. Here, the discretization of the optimal control problem typically follows discretization techniques used for governing state equations.

We select the following spaces for the discretization of the state and the control, respectively:

$$\begin{aligned} V_h = Y_h &= \{y_h \in L^2(\Omega) \mid \ y|_E \in P_n(E), \ \ \forall E \in \xi_h\}, && (3.13)\\ U_h &= \{u_h \in L^2(\Omega) \mid \ u|_E \in P_m(E), \ \ \forall E \in \xi_h\}. && (3.14) \end{aligned}$$

The orders $n, m \in \mathbb{N}$ of the finite element approximation can be different for the state and controls. We can take that the space of test functions V_h are identical to the space of state Y_h due to the weak treatment of the boundary conditions in DG discretizatin.

Then, the discretized optimal control problem is

$$\text{minimize } J(y_h, u_h) := \frac{1}{2} \sum_{E \in \xi_h} \|y_h - y_d\|_E^2 + \frac{\omega}{2} \sum_{E \in \xi_h} \|u_h\|_E^2 \tag{3.15a}$$

$$\begin{aligned} \text{subject to} \quad a_h^s(y_h, \upsilon_h) \quad &+ \quad b_h(u_h, \upsilon_h) = l_h^s(\upsilon_h), \qquad \forall \upsilon_h \in V_h, \\ (y_h, u_h) \quad &\in \quad Y_h \times U_h. \end{aligned} \tag{3.15b}$$

The Lagrangian of the discretized problem (3.15) is defined by

$$L_h(y_h, u_h, p_h) = \frac{1}{2} \sum_{E \in \xi_h} \|y_h - y_d\|_E^2 + \frac{\omega}{2} \sum_{E \in \xi_h} \|u_h\|_E^2 + a_h^s(y_h, p_h) + b_h(u_h, p_h) - l_h^s(p_h), \tag{3.16}$$

where $y_h \in Y_h$, $u_h \in U_h$ and $p_h \in \Lambda_h = V_h$. Then, necessary and sufficient optimality conditions for the discretized problem are obtained by setting the partial derivatives of Lagrange function (3.16) to zero

$$\nabla_y L_h(y_h, u_h, p_h) = 0, \tag{3.17a}$$

$$\nabla_u L_h(y_h, u_h, p_h) = 0, \tag{3.17b}$$

$$\nabla_p L_h(y_h, u_h, p_h) = 0. \tag{3.17c}$$

Then, we obtain the following optimality system consisting of:

the discrete adjoint equation

$$a_h^s(\psi_h, p_h) = -(y_h - y_d, \psi_h), \qquad \forall \psi_h \in V_h, \tag{3.18a}$$

the discrete gradient equation

$$b_h(w_h, p_h) + \omega(u_h, w_h) = 0, \qquad \forall w_h \in U_h, \tag{3.18b}$$

and the discretized state equation

$$a_h^s(y_h, \upsilon_h) + b_h(u_h, \upsilon_h) = l_h^s(\upsilon_h), \qquad \forall \upsilon_h \in V_h. \tag{3.18c}$$

where

$$
\begin{aligned}
a_h^s(y_h, \upsilon_h) &= \sum_{E\in\xi_h} (\epsilon\nabla y_h, \nabla\upsilon_h)_E \\
&+ \kappa \sum_{e\in\Gamma_h} (\{\epsilon\nabla\upsilon_h \cdot n_e\}, [y_h])_e - \sum_{e\in\Gamma_h} (\{\epsilon\nabla y_h \cdot n_e\}, [\upsilon_h])_e \\
&+ \sum_{e\in\Gamma_h} \frac{\sigma\epsilon}{h_e^{\beta_0}} ([y_h], [\upsilon_h])_e + \sum_{E\in\xi_h} (\beta\cdot\nabla y_h + r y_h, \upsilon_h)_E \\
&+ \sum_{e\in\Gamma_h^0} (y_h^+ - y_h^-, |n\cdot\beta|\upsilon_h^+)_e + \sum_{e\in\Gamma_h^-} (y_h^+, \upsilon_h^+ |n\cdot\beta|)_e, \qquad (3.19a) \\
b_h(u_h, \upsilon_h) &= -\sum_{E\in\xi_h} (u_h, \upsilon_h)_E, \qquad (3.19b) \\
l_h^s(\upsilon_h) &= \sum_{E\in\xi_h} (f, \upsilon_h)_E + \sum_{e\in\Gamma_h^\partial} \left(\frac{\sigma\epsilon}{h_e^{\beta_0}} (g_D, [\upsilon_h])_e - (\epsilon g_D, \nabla\upsilon_h)_e \right) \\
&+ \sum_{e\in\Gamma_h^-} (g_D, \upsilon_h^+ |n\cdot\beta|)_e. \qquad (3.19c)
\end{aligned}
$$

The superscript s is used to indicate that the DG methods are applied to the state equation. Here, discrete adjoint equation and discrete gradient equation are referred to the adjoint and gradient equations for the discretized problem (3.15). However, the discretized state equations are meant as direct discretization of the state equation (3.2).

Remark 3.2.1 *Upon integration by parts of the convection term, we obtain*

$$
\begin{aligned}
a_h^s(y_h, \upsilon_h) &= \sum_{E\in\xi_h} (\epsilon\nabla y_h, \nabla\upsilon_h)_E \\
&+ \kappa \sum_{e\in\Gamma_h} (\{\epsilon\nabla\upsilon_h \cdot n_e\}, [y_h])_e - \sum_{e\in\Gamma_h} (\{\epsilon\nabla y_h \cdot n_e\}, [\upsilon_h])_e \\
&+ \sum_{e\in\Gamma_h} \frac{\sigma\epsilon}{h_e^{\beta_0}} ([y_h], [\upsilon_h])_e + \sum_{E\in\xi_h} (\beta\cdot\nabla y_h + (r - \nabla\cdot\beta) y_h, \upsilon_h)_E \\
&+ \sum_{e\in\Gamma_h^0} (\upsilon_h^+ - \upsilon_h^-, |n\cdot\beta| y_h^+)_e + \sum_{e\in\Gamma_h^+} (y_h^+, \upsilon_h^+ |n\cdot\beta|)_e.
\end{aligned}
$$

The state and control spaces can also be rewritten such that

$$
\begin{aligned}
Y_h &= span\{\varphi_i^E : 1 \le i \le N_{loc},\ E \in \xi_h\}, \\
U_h &= span\{\psi_i^E : 1 \le i \le N_{loc},\ E \in \xi_h\},
\end{aligned}
$$

where φ_i and ψ_i are bases functions for the state and control spaces, respectively. Then, the

state y and the control u are functions of the form

$$y(x) = \sum_{m=1}^{N} \sum_{j=1}^{N_{loc}} y_j^m \varphi_m^j(x), \tag{3.20a}$$

$$u(x) = \sum_{m=1}^{N} \sum_{j=1}^{N_{loc}} u_j^m \psi_m^j(x). \tag{3.20b}$$

Set

$$\vec{y} = (y_1^1, y_2^1, \ldots, y_{N_{loc}}^1, \ldots, y_1^N, y_2^N, \ldots, y_{Nloc}^N)^T,$$
$$\vec{u} = (u_1^1, u_2^1, \ldots, u_{N_{loc}}^1, \ldots, u_1^N, u_2^N, \ldots, u_{Nloc}^N)^T,$$

where N is the number of triangles and N_{loc} is local dimension.

If we insert (3.20a) and (3.20b) into (3.15), we obtain the following discretized optimization problem

$$\text{minimize } J(\vec{y}, \vec{u}) := \frac{1}{2}\vec{y}^T \mathbb{M}\vec{y} - \vec{b}^T\vec{y} + \frac{\omega}{2}\vec{u}^T \mathbb{Q}\vec{u} + \int_\Omega \frac{1}{2} y_d^2 \, dx \tag{3.21a}$$

$$\text{subject to } \quad \mathbb{A}_s\vec{y} + \mathbb{B}\vec{u} = \vec{f}, \tag{3.21b}$$

where $\mathbb{A}_s, \mathbb{M}, \mathbb{Q}, \mathbb{B} \in \mathbb{R}^{(N_{loc}\times N)\times(N_{loc}\times N)}$ and $\vec{b}, \vec{f} \in \mathbb{R}^{N_{loc}\times N}$.

$\mathbb{A}_s$, $\vec{f}$ correspond to the bilinear form $a_h^s(y_h, \upsilon_h)$ (3.19a) and the linear form $l_h^s(\upsilon_h)$ (3.19c), respectively. Additionally, the matrices $\mathbb{M}$, $\mathbb{B}$ and $\mathbb{Q}$ are given by

$$(\mathbb{M})_{ij} = \int_E \varphi_j\varphi_i \, dx, \quad (\mathbb{Q})_{ij} = \int_E \psi_j\psi_i \, dx, \quad (\mathbb{B})_{ij} = -\int_E \varphi_j\psi_i \, dx,$$

and the entries of the vector $\vec{b}$ are

$$(\vec{b})_i = \int_E y_d\varphi_i \, dx.$$

The Lagrangian for the discretized problem (3.21) is given by

$$L(\vec{y}, \vec{u}, \vec{p}) = \frac{1}{2}\vec{y}^T \mathbb{M}\vec{y} - \vec{b}^T\vec{y} + \frac{\omega}{2}\vec{u}^T \mathbb{Q}\vec{u} + \int_\Omega \frac{1}{2} y_d^2 \, dx + \vec{p}^T(\mathbb{A}_s\vec{y} + \mathbb{B}\vec{u} - \vec{f}).$$

By setting the partial derivatives of the Lagrangian equal to zero, the optimality system of the discretized problem (3.21) written as

$$\begin{aligned} \nabla_y L(\vec{y}, \vec{u}, \vec{p}) &= 0, & \mathbb{M}\vec{y} + \mathbb{A}_s^T\vec{p} &= \vec{b}, \\ \nabla_u L(\vec{y}, \vec{u}, \vec{p}) &= 0, \quad \Rightarrow & \omega\mathbb{Q}\vec{u} + \mathbb{B}^T\vec{p} &= 0, \\ \nabla_p L(\vec{y}, \vec{u}, \vec{p}) &= 0, & \mathbb{A}_s\vec{y} + \mathbb{B}\vec{u} &= \vec{f}. \end{aligned}$$

then the optimality system can be rewritten such as:

$$\begin{pmatrix} \mathbb{M} & 0 & \mathbb{A}_s^T \\ 0 & \omega\mathbb{Q} & \mathbb{B}^T \\ \mathbb{A}_s & \mathbb{B} & 0 \end{pmatrix} \begin{pmatrix} \vec{y} \\ \vec{u} \\ \vec{p} \end{pmatrix} = \begin{pmatrix} \vec{b} \\ 0 \\ \vec{f} \end{pmatrix}.$$

3.3 Optimize-then-Discretize Approach

The alternative approach to the *discretize-then-optimize* approach is the *optimize-then-discretize*. In this approach, one first computes the infinite dimensional optimality system, involving the state convection diffusion equation as well as the adjoint convection diffusion equation and then discretizes the optimality system using the DG methods.

We use (3.13) and (3.14) for the state space and the control space, respectively, and the following space:

$$\Lambda_h = \{p_h \in L^2(\Omega) \mid \ p|_E \in P_l(E), \ \ \forall E \in \xi_h\} \tag{3.22}$$

for the discretization of the adjoint. It is possible to choose $l \neq k$. By discretizing the adjoint equation (3.10), the gradient equation (3.11) and the state equation (3.2) using the DG methods, we obtain the discretized state, adjoint and gradient equations, respectively:

$$\begin{aligned} a_h^s(y_h, \upsilon_h) + b_h(u_h, \upsilon_h) &= l_h^s(\upsilon_h), \qquad \forall v_h \in Y_h, && \text{(3.23a)} \\ a_h^a(p_h, \psi_h) + (y_h, \psi_h) &= (y_d, \psi_h), \qquad \forall \psi_h \in \Lambda_h, && \text{(3.23b)} \\ b_h(w_h, p_h) + \omega(u_h, w_h) &= 0, \qquad \forall w_h \in U_h, && \text{(3.23c)} \end{aligned}$$

where

$$\begin{aligned} a_h^a(p_h, \psi_h) &= \sum_{E\in\xi_h} (\epsilon\nabla p_h, \nabla\psi_h)_E \\ &+ \kappa \sum_{e\in\Gamma_h} (\{\epsilon\nabla\psi_h \cdot n_e\}, [p_h])_e - \sum_{e\in\Gamma_h} (\{\epsilon\nabla p_h \cdot n_e\}, [\psi_h])_e \\ &+ \sum_{e\in\Gamma_h} \frac{\sigma\epsilon}{h_e^{\beta_0}} ([p_h], [\psi_h])_e + \sum_{E\in\xi_h} (-\beta\cdot\nabla p_h + (r - \nabla\cdot\beta)p_h, \psi_h)_E \\ &+ \sum_{e\in\Gamma_h^0} (p_h^+ - p_h^-, |n\cdot\beta|\psi_h^+)_e + \sum_{e\in\Gamma_h^+} (p_h^+, \psi_h^+|n\cdot\beta|)_e. \end{aligned} \tag{3.24}$$

The superscript a indicates that the DG methods are applied to the adjoint equation. In addition, (3.23a) and (3.23c) are identical to (3.19c) and (3.19b), respectively.

Remark 3.3.1 *Γ_h^+ is inflow boundary for the adjoint equation (3.10)*

$$\begin{aligned}\Gamma_h^+ &= \{x \in \partial\Omega : \quad \beta(x) \cdot n(x) > 0\} \\ &= \{x \in \partial\Omega : \quad -\beta(x) \cdot n(x) < 0\}.\end{aligned}$$

We similarly rewrite the adjoint space Λ_h such that

$$\Lambda_h = span\{\varphi_i^E : \ 1 \le i \le N_{loc}, \ E \in \xi_h\},$$

where φ_i are basis functions for the adjoint space. Then, the adjoint p is function of the form

$$p(x) = \sum_{m=1}^{N} \sum_{j=1}^{N_{loc}} p_j^m \varphi_m^j(x). \tag{3.25}$$

Also, y and u are defined as in (3.20a) and (3.20b), respectively.

In addition, we set

$$\vec{p} = (p_1^1, p_2^1, \ldots, p_{N_{loc}}^1, \ldots, p_1^N, p_2^N, \ldots, p_{N_{loc}}^N)^T.$$

If we insert (3.20a), (3.20b) and (3.25) into (3.2), (3.10) and (3.11), respectively, we obtain the following system

$$\begin{aligned}\mathbb{M}\vec{y} + \mathbb{A}_a\vec{p} &= \vec{b}, \\ \omega\mathbb{Q}\vec{u} + \mathbb{B}\vec{p} &= 0, \\ \mathbb{A}_s\vec{y} + \mathbb{B}\vec{u} &= \vec{f},\end{aligned}$$

where $\mathbb{A}_s, \mathbb{A}_a, \mathbb{M}, \mathbb{Q}, \mathbb{B} \in \mathbb{R}^{(N_{loc}\times N)\times(N_{loc}\times N)}$ and $\vec{b}, \vec{f} \in \mathbb{R}^{N_{loc}\times N}$. Furthermore, $\mathbb{A}_a$ corresponds to the bilinear form $a_h^a(p_h, \upsilon)$ in (3.24).

Then, the optimality system is written such as:

$$\begin{pmatrix} \mathbb{M} & 0 & \mathbb{A}_a \\ 0 & \omega\mathbb{Q} & \mathbb{B} \\ \mathbb{A}_s & \mathbb{B} & 0 \end{pmatrix} \begin{pmatrix} \vec{y} \\ \vec{u} \\ \vec{p} \end{pmatrix} = \begin{pmatrix} \vec{b} \\ 0 \\ \vec{f} \end{pmatrix}.$$

Theorem 3.3.2 *The optimality system (3.18) of the DG discretized optimal control problem (3.15) is equivalent to the DG discretization (3.23) of the optimality system of the optimal control problem (3.1)-(3.2) for symmetric DG methods, i.e., SIPG. On the other hand, it does not hold for nonsymmetric DG methods, i.e., NIPG and IIPG.*

Proof. Let us define

$$\varrho(y_h, \upsilon_h) = \sum_{E\in\xi} \int_E \beta \cdot \nabla y_h \upsilon_h \, dx + \sum_{e\in\Gamma_h^0} \int_e (y_h^+ - y_h^-)|n\cdot\beta|\upsilon_h^+ \, ds + \sum_{e\in\Gamma_h^-} \int_e y_h^+ \upsilon_h^+ |n\cdot\beta| \, ds. \quad (3.26)$$

When we apply integration by parts on the first integrand in (3.26), we obtain

$$\begin{aligned} \varrho(y_h, \upsilon_h) &= \sum_{E\in\xi} \int_E -\beta\cdot\nabla y_h \upsilon_h \, dx + \sum_{E\in\xi} -\nabla\cdot\beta y_h \upsilon_h \, dx \\ &+ \sum_{e\in\Gamma_h^0} \int_e (\upsilon_h^+ - \upsilon_h^-)|n\cdot\beta| y_h^+ \, ds + \sum_{e\in\Gamma_h^+} \int_e y_h^+ \upsilon_h^+ |n\cdot\beta| \, ds. \end{aligned} \quad (3.27)$$

Writing $a_h^s(\psi_h, p_h)$ clearly using (3.19a), we get

$$\begin{aligned} a_h^s(\psi_h, p_h) &= \sum_{E\in\xi_h} (\epsilon\nabla\psi_h, \nabla p_h)_E + \sum_{E\in\xi_h} (r\psi_h, p_h)_E \\ &+ \kappa \sum_{e\in\Gamma_h} (\{\epsilon\nabla p_h\cdot n_e\}, [\psi_h])_e - \sum_{e\in\Gamma_h} (\{\epsilon\nabla\psi_h\cdot n_e\}, [p_h])_e \\ &+ \sum_{e\in\Gamma_h} \frac{\sigma\epsilon}{h_e^{\beta_0}} ([\psi_h], [p_h])_e + \sum_{E\in\xi_h} (\beta\cdot\nabla\psi_h + r\psi_h, p_h)_E \\ &+ \sum_{e\in\Gamma_h^0} (\psi_h^+ - \psi_h^-, |n\cdot\beta| p_h^+)_e + \sum_{e\in\Gamma_h^-} (\psi_h^+, p_h^+ |n\cdot\beta|)_e. \end{aligned}$$

Upon integration by parts of convection term as (3.26), we have

$$\begin{aligned} a_h^s(\psi_h, p_h) &= \sum_{E\in\xi_h} (\epsilon\nabla\psi_h, \nabla p_h)_E \\ &+ \kappa \sum_{e\in\Gamma_h} (\{\epsilon\nabla p_h\cdot n_e\}, [\psi_h])_e - \sum_{e\in\Gamma_h} (\{\epsilon\nabla\psi_h\cdot n_e\}, [p_h])_e \\ &+ \sum_{e\in\Gamma_h} \frac{\sigma\epsilon}{h_e^{\beta_0}} ([\psi_h], [p_h])_e + \sum_{E\in\xi_h} (-\beta\cdot\nabla p_h + (r - \nabla\cdot\beta) p_h, \psi_h)_E \\ &+ \sum_{e\in\Gamma_h^0} (p_h^+ - p_h^-, |n\cdot\beta|\psi_h^+)_e + \sum_{e\in\Gamma_h^+} (p_h^+, \psi_h^+ |n\cdot\beta|)_e \\ &\stackrel{?}{=} a_h^a(p_h, \psi_h). \end{aligned}$$

Now, when we examine the cases with respect to the parameter κ:

- for $\kappa = -1$, i.e., SIPG, $\quad a_h^s(\psi_h, p_h) = a_h^a(p_h, \psi_h), \quad \forall\psi \in V_h,$
- for $\kappa = 1$, i.e., NIPG and $\kappa = 0$, i.e., IIPG, $\quad a_h^s(\psi_h, p_h) \neq a_h^a(p_h, \psi_h), \quad \forall\psi \in V_h.$

■

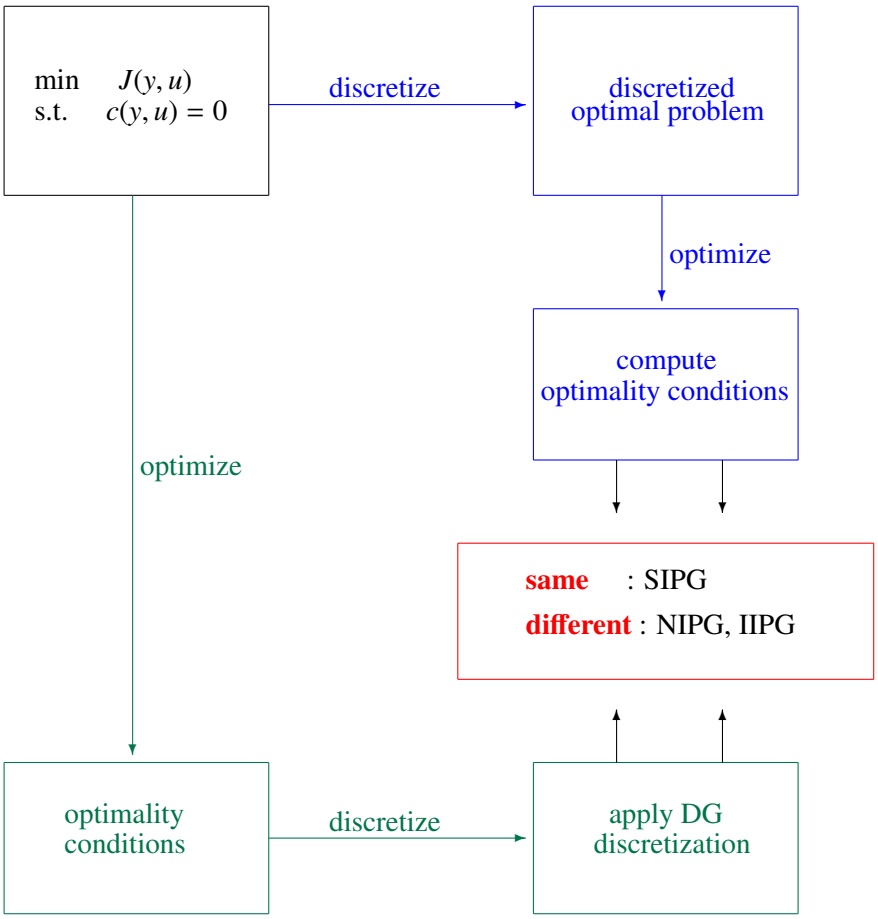


Figure 3.1: *Discretize-then-optimize* versus *optimize-then-discretize* with DG discretization.

3.4 Numerical Results

In this section, we give several numerical results for optimal control problems governed by convection diffusion equations using the DG methods. We use piecewise linear ($k = 1$) and piecewise quadratic ($k = 2$) polynomials. The penalty parameter is $\sigma = 1$ for all edges for NIPG. For SIPG and IIPG we set $\sigma = 3k(k+1)$ for interior edges and $\sigma = 6k(k+1)$ on boundary edges.

If standard penalization $\beta_0 = 1$ is used in NIPG and IIPG, we refer to these methods as NIPG1 and IIPG1, respectively. If superpenalization $\beta_0 = 3$ is used, we refer to these methods as NIPG3 and IIPG3, respectively.

Remark 3.4.1 *The convergence rates are obtained numerically for h sufficiently small by*

applying following formula:

$$rate_{L_2} = \frac{1}{2}\ln\left(\frac{\|e_h\|_{L_2}}{\|e_{h/2}\|_{L_2}}\right).$$

Example 3.4.2 *This following example has been studied by Collis, Heinkenschloss and Leykekhman [35, 48].*

The problem data are given by

$$\Omega = [0,1]^2, \quad \theta = 45^o, \quad \beta = (\cos\theta, \sin\theta), \quad r = 0 \ \text{ and } \ \omega = 1.$$

The source function f, the desired state y_d and Dirichlet boundary conditions g_D are chosen such that the analytical solutions of the state and the adjoint are given by

$$y(x_1, x_2) = \eta(x_1)\eta(x_2), \quad p(x_1, x_2) = \mu(x_1)\mu(x_2),$$

where

$$\eta(z) = z - \frac{\exp((z-1)/\epsilon) - \exp(-1/\epsilon)}{1 - \exp(-1/\epsilon)}, \quad \mu(z) = 1 - z - \frac{\exp(-z/\epsilon) - \exp(-1/\epsilon)}{1 - \exp(-1/\epsilon}.$$

We have tested this example with different values of diffusion parameter; $\epsilon = 1$ and $\epsilon = 10^{-2}$. In terms of comparison of the *discretize-then-optimize* and the *optimize-then-discretize*, we have obtained the same results for different values of diffusion parameter. However, to reach optimal convergence rates we need much mesh refinement for the convection dominated case.

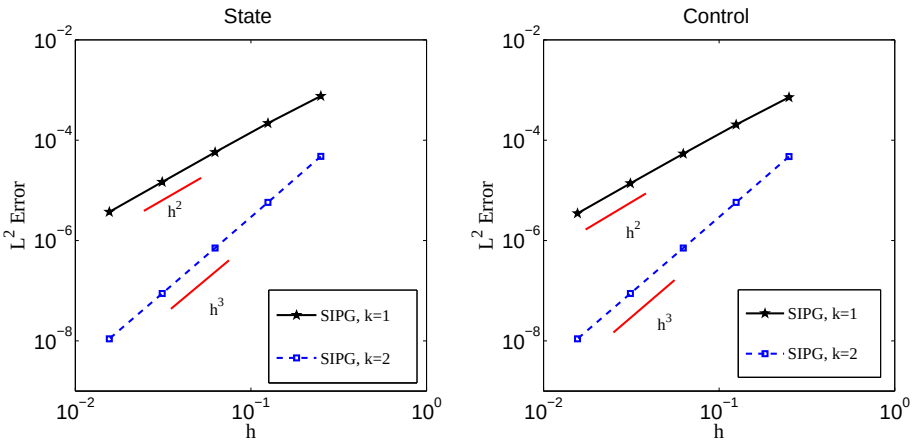


Figure 3.2: L_2 error for SIPG with $\epsilon = 1$.

The *discretize-then-optimize* and the *optimize-then-discretize* approaches are equivalent for SIPG method independent of degree of basis functions. The results in Figure 3.2 show that

the orders of convergence for state and control variables are optimal for both approaches in SIPG case. Figure 3.2 reveals that it is $O(h^2)$ and $O(h^3)$ for linear and quadratic elements, respectively.

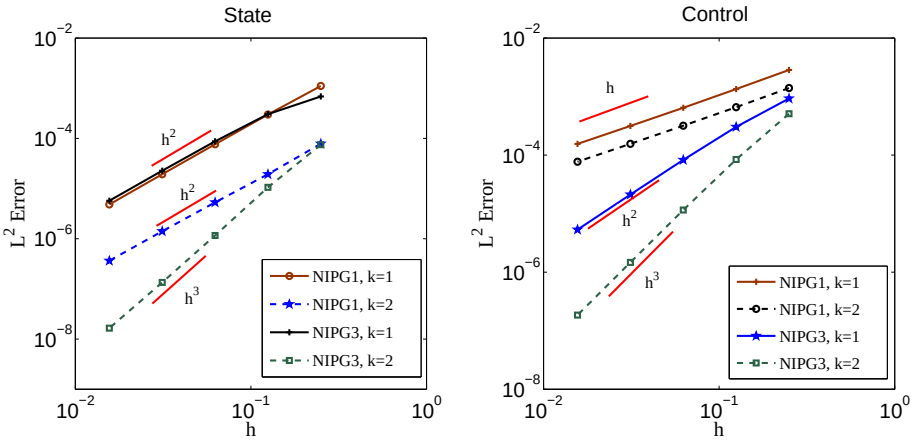


Figure 3.3: L_2 error for NIPG1 and NIPG3 with $\epsilon = 1$: *discretize-then-optimize*.

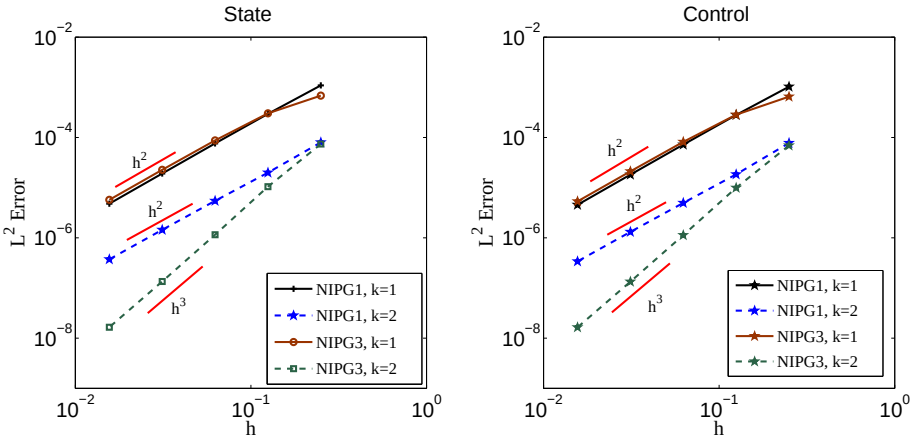


Figure 3.4: L_2 error for NIPG1 and NIPG3 with $\epsilon = 1$: *optimize-then-discretize*.

In the nonsymmetric case, i.e., NIPG1 and IIPG1, two approaches to solve the optimal control problem numerically are not equivalent as shown in Figures 3.3 and 3.4 for the NIPG1 method. The numerical results confirm the Theorem 3.3.2. The convergence rate of the control obtained from the *discretize-then optimize* is linear whereas it is quadratic for the *optimize-then-discretize* approach. For the convergence order of state in NIPG1, the results in Figures 3.3 and 3.4 reveal that it is almost the same for linear and quadratic elements.

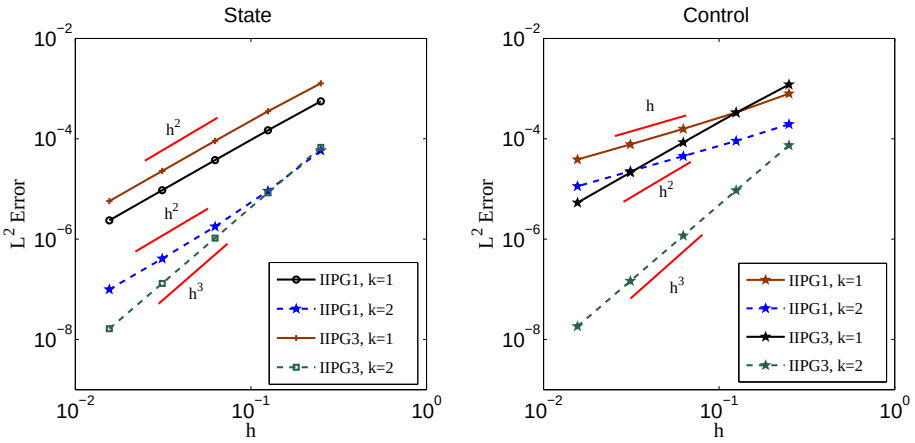


Figure 3.5: L_2 error for IIPG1 and IIPG3 with $\epsilon = 1$: *discretize-then-optimize*.

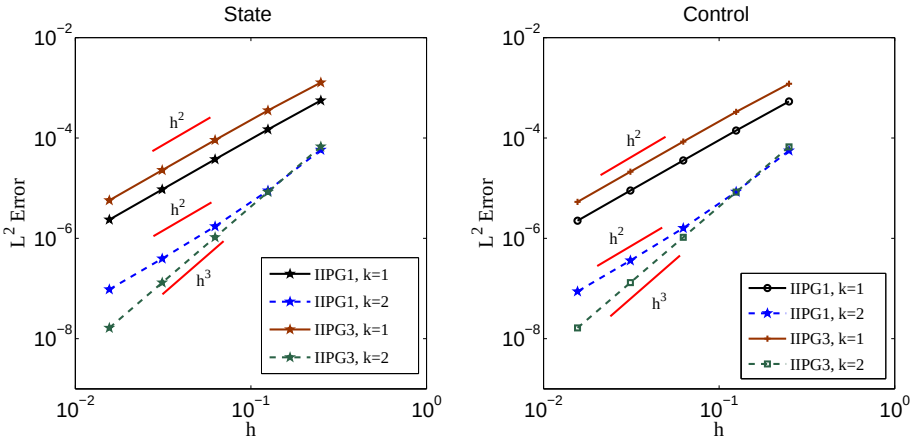


Figure 3.6: L_2 error for IIPG1 and IIPG3 with $\epsilon = 1$: *optimize-then-discretize*.

To overcome this shortcoming property of the NIPG1 method, superpenalization has been proposed in [87]. With superpenalization, the lack of adjoint consistency is reduced and hence, we obtain $O(h^2)$ and $O(h^3)$ rates for linear and quadratic elements, respectively, for the state variable in NIPG3 case. In addition, we observe that in Figures 3.3 and 3.4 the *discretize-then-optimize* approach and the *optimize-then-discretize* approach are almost the same for NIPG3. However, the drawback of superpenalization is to increase the condition number of the system. See [30] for details. Similar numerical results for IIPG1 and IIPG3 as NIPG method are shown in Figures 3.5 and 3.6.

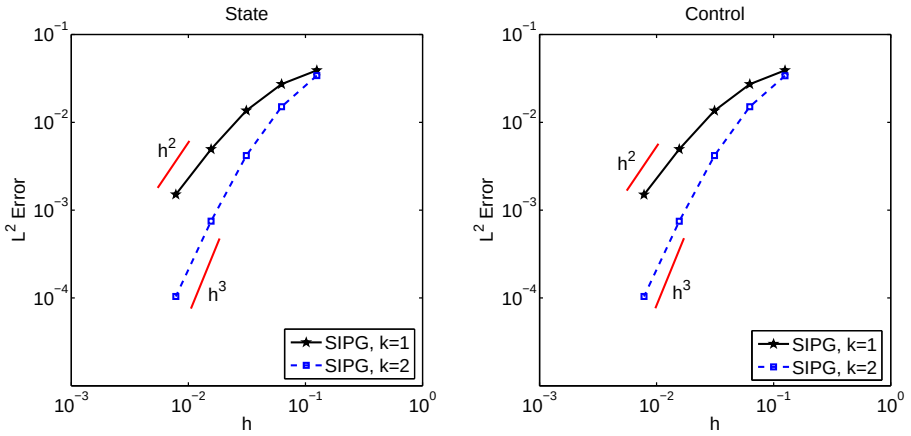


Figure 3.7: L_2 error for SIPG with $\epsilon = 10^{-2}$.

For the convection dominated case ($\epsilon = 10^{-2}$), we have similar results in terms of comparison of both approaches to solve the optimal control problem. However, the numerical results in Figure 3.7, 3.8 and 3.9 reveal that the convergence rates are not achieved at the same number of mesh refinement as $\epsilon = 1$. The reason of this situation is that this example has a boundary layer where $x_1 = 1$ or $x_2 = 1$ for the state equation and $x_1 = 0$ or $x_2 = 0$ for the adjoint equation, as well as the control. To solve the convection dominated problems having boundary and/or interior layers with a minimum degree of freedom, we need an adaptive strategy.

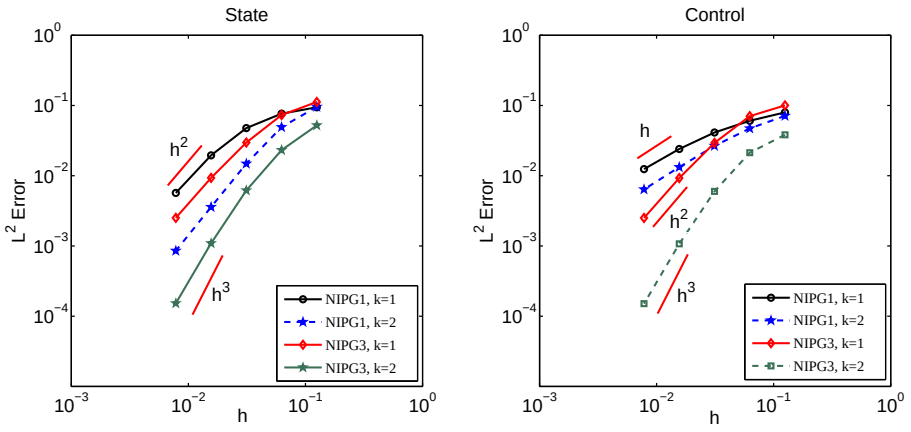


Figure 3.8: L_2 error for NIPG1 and NIPG3 with $\epsilon = 10^{-2}$: *discretize-then-optimize*.

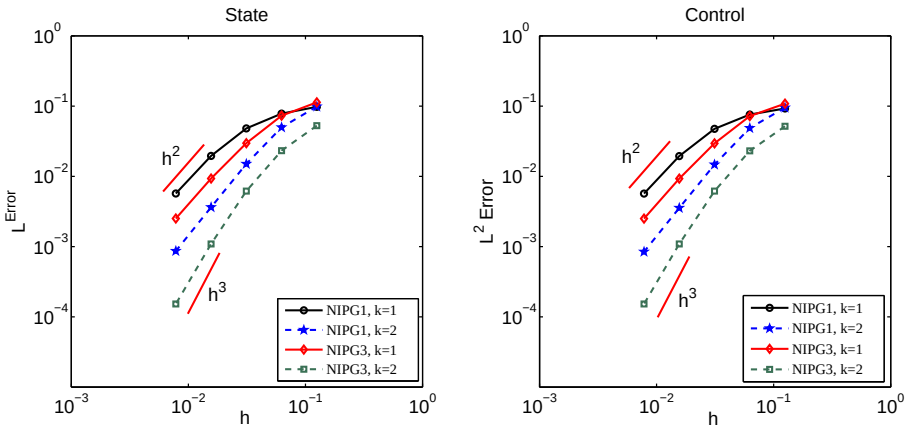


Figure 3.9: L_2 error for NIPG1 and NIPG3 with $\epsilon = 10^{-2}$: *optimize-then-discretize*.

CHAPTER 4

DISTRIBUTED OPTIMAL CONTROL PROBLEMS WITH ADAPTIVITY

When convection dominates diffusion, the solutions of these PDEs typically exhibit layers on small regions where the solution has large gradients; we speak of the boundary and/or interior layers. The standard finite element methods (FEMs) applied to convection dominated diffusion problems lead to strong oscillations when layers are not properly solved. To overcome this difficulty, we need special numerical techniques, which take into account the structure of the convection. The adaptive finite element methods are an effective numerical resolution to overcome the difficulty caused by boundary and/or layers.

Adaptive finite element methods can be summarized such as computation of a numerical solution on a triangulation, estimation of the local error on each single element, marking of the elements, and refinement of the selected elements. This iteration is repeated until a desired accuracy or a maximum number of degree of freedom is reached.

The key part of adaptive finite element methods is a posteriori error estimators that provide information to select the elements to refine. In the literature, there exist a huge amount of study related to error estimators. The simple estimator is the Zienkiewicz-Zhu estimator which only uses the numerical solution and does not need the problem data, has introduced in [108] by Zienkiewicz and Zhu. This estimation is an averaging technique to obtain a high-order recovery for the gradient of solution. For an extensive study on averaging techniques, see [11]. In literature, there are two main approaches in the context of mesh adaptivity based on a posteriori error estimates: error estimation with respect to the natural *energy norm* induced by the given variational form and goal-oriented error estimation with respect to a preassigned *quantity of interest*. The former one has been studied in [3, 8, 97, 98, 99, 100]. Verfürth has

studied the local Dirichlet (or Neumann) estimator that is based on the solution of auxiliary local discrete problems with Dirichlet (or Neumann) boundary conditions and the norm-residual based estimator that is incorporate the norm of residuals in [97, 98, 99, 100]. Then, Verfürth has proposed a fully robust error estimator with a dual norm of the convective derivative in [99]. It has been robust in sense of uniformly boundedness of the upper and lower bounds of estimators with respect to the size of the convection.

The second main approach called *goal-oriented error estimator* in finite element approximations has been developed by Becker and Rannacher [16, 17] and is referred as the dual-weighted residual (DWR) method. Goal-oriented error estimation consists of the solution of a problem dual to the original problem and the computation of several global error estimates. Oden and Prudhomme have used this approach for elliptic problem in [85] and have extended it in computational mechanics [82]. For convection diffusion problems, it has been studied in [23, 24, 33, 69].

For the numerical solution of convection diffusion problems DG methods have became popular because of their stability properties in the around of boundary and/or interior layers. Different kinds of error estimations with DG methods can also be found in the literature. Explicit a posteriori estimators for the error in DG approximation measured in mesh-dependent energy type norms for DG methods applied to pure diffusion problems has been studied in [14, 20, 57, 65, 66]. Karakashian and Pascal [66] have firstly studied a convergence analysis of a residual type a posteriori error estimator based on DG discretization. Then, Hoppe et al. [57] have proven the convergence analysis of the a posterior estimator in [66] by using any multiple interior nodes for refined elements of triangulation as in [66]. Recently, Bonito and Nochetto [20] have extended and improved [57, 66] in several respects: allowing discontinuity of diffusion parameter and nonconforming meshes. Ainsworth [2] has derived a posteriori error estimator that is free of any unknown constant. In [64], Kanschat and Rannacher have proposed a functional error estimation with symmetric interior Galerkin discretization (SIPG). On the other hand, Rivièra et al. [88] has used the nonsymmetric interior penalty Galerkin (NIPG) method to introduce a posteriori estimator with L_2 norm. For convection diffusion problems, Ern et al. has proposed a posteriori error estimator with inhomogeneous and anisotropic diffusion approximated by weighted interior penalty discontinuous Galerkin methods. Furthermore, Houston et al. [59] has derived a-posteriori estimator using energy norm for hp-adaptivity and [60] has given a goal-oriented a posteriori error estimation for

both conforming and DG finite element methods.

In this thesis, our studies with adaptive approach are based on the error estimator proposed by Schötzau and Zhu in [91]. This estimation is an extensive of [99] to symmetric interior penalty Galerkin (SIPG) discretization. The upper and lower bounds of this estimator are measured in terms of the natural energy norm and a semi-norm associated with convective terms. In addition, it is a robust estimator in sense that the reliability and efficiency bounds are uniformly bounded with respect to the ratio of convection and diffusion coefficients.

Solving an optimization problems governed by convection diffusion equation is substantially different from solving a single convection diffusion PDE. One reason why the solution of optimization problems governed by the convection diffusion PDEs (also referred to as the state PDEs) provides additional challenges is that the optimality conditions for such optimization problems do not only involve the original convection diffusion state equation, but also another convection diffusion PDE, the so-called adjoint PDE. The diffusion part of the adjoint PDE is equal to that of the state PDE, but the convection in the adjoint PDE is equal to the negative of the convection in the state PDE (in case of nonlinear state PDEs, the convection in the adjoint PDE is equal to the linearized state PDE). This has important implications for the behavior of the solution, as well as for numerical methods for their solution. In the optimal control context, boundary and/or interior layers are generated in the state PDE as well as in the adjoint PDE. These layers are determined by the convection as well as by its negative. This leads to different error propagation properties in the optimization context compared to what one may expect from studying the solution of a single convection diffusion PDE.

As far as the a posteriori error analysis of adaptive finite element schemes for optimal control problem is concerned, there is few work for unconstrained case of optimal control problems. In [10, 15], error and mesh adaptivity has been described for the discretization of optimal control problems governed by elliptic PDEs. Becker et al. [15] has proposed a residual-based a-posteriori error estimates called the goal-oriented dual weighted approach derived by duality arguments employing the cost functional of the optimization problem for controlling the discretization. In [38], Dedè and Quarteroni have used a posteriori error estimator with a stabilization method applied to the Lagrangian functional for optimal control problems governed by convection diffusion equations. Their a posteriori estimates stems from splitting the error on the cost functional into the sum of an iteration error plus a discretization error. Adaptivity

strategy has been applied on discretization error after reducing the former error below a given threshold. Nederkoorn in [81] has combined adaptive finite element methods with SUPG stabilization introduced in [35], to solve linear-quadratic convection dominated elliptic optimal control problems. For convection dominated optimal control problems, Leykekhman and Heinkenschloss [48] have shown that the local error for the SUPG discretized optimal control problem is not optimal even if the error is computed locally in a region away from the boundary layer. Then, Leykekhman and Heinkenschloss [73] have overcome this problem by using the symmetric interior penalty Galerkin (SIPG) method. The reason why the SIPG method gives an optimal convergence is the weak treatment of boundary conditions which is natural for DG methods, whereas SUPG methods have the strong imposition of boundary conditions. In [73], the authors have only solved the convection dominated problems on the region away from the boundary and/or interior layers. To solve such problems on hull region, the adaptive strategy can be an effective way. To our knowledge, there is any work related to adaptive discontinuous Galerkin methods for the convection dominated unconstrained optimal control problems.

In this chapter, we solve the convection dominated optimal control problems, based on DG discretization. We firstly propose a posteriori error estimator which are apparently not available in the literature. Then, we analyze the reliability and the efficiency of the error estimator using data approximation error. Finally, the numerical results are presented that illustrate the performance of the proposed error estimator.

4.1 The Adaptive Loop

Adaptive procedure can be summarized such as computation of a numerical solution on a triangulation, estimation the local error on each single element, marking of the elements, and refinement of the selected elements. This iteration is repeated until a desired accuracy or a maximum number of degree of freedom is reached.

$$\textbf{SOLVE} \rightarrow \textbf{ESTIMATE} \rightarrow \textbf{MARK} \rightarrow \textbf{REFINE}.$$

Here, **SOLVE** stands for the numerical solution of optimal control problem (3.1)-(3.2) with respect to the given triangulation ξ_h using the discontinuous Galerkin methods given in Chapter 2.

The key part of the adaptive loop is **ESTIMATE** part since it provides information to mark the elements to refine. We use a residual-type error estimator to mark the elements which have a large error caused from state, adjoint and control variables. The error indicators of the state and the adjoint used here are based on the error estimator given in [91] for a single convection dominated PDE.

Set

$$\rho_E = \min\{h_E\epsilon^{-\frac{1}{2}}, r_0^{-\frac{1}{2}}\}, \qquad \rho_e = \min\{h_e\epsilon^{-\frac{1}{2}}, r_0^{-\frac{1}{2}}\}.$$

When $r_0 = 0$, $\rho_E = h_E\epsilon^{-\frac{1}{2}}$ and $\rho_e = h_e\epsilon^{-\frac{1}{2}}$ are taken.

For each element $E \in \xi_h$, the error indicators of state η_E^y, adjoint η_E^p and control η_E^u are

$$\begin{aligned}
(\eta_E^y)^2 &= \left[(\eta_{E_R}^y)^2 + (\eta_{e_D}^y)^2 + (\eta_{e_J}^y)^2\right], \\
(\eta_E^p)^2 &= \left[(\eta_{E_R}^p)^2 + (\eta_{e_D}^p)^2 + (\eta_{e_J}^y)^2\right], \\
(\eta_E^u)^2 &= \left[(\eta_{E_R}^u)^2\right].
\end{aligned}$$

Here, η_E stands for the element residual

$$\begin{aligned}
\eta_{E_R}^y &= \rho_E\|f_h + u_h + \epsilon\Delta y_h - \beta_h\cdot\nabla y_h - r_h y_h\|_{L^2(E)}, && E \in \xi_h, \\
\eta_{E_R}^p &= \rho_E\|-(y_h - (y_d)_h + \epsilon\Delta p_h + \beta_h\cdot\nabla p_h - r_h p_h\|_{L^2(E)}, && E \in \xi_h, \\
\eta_{E_R}^u &= \|\omega u_h - p_h\|_{L^2(E)}, && E \in \xi_h.
\end{aligned}$$

where y_h, p_h, u_h be the discontinuous Galerkin approximations. Moreover, f_h, $(y_d)_h$ and β_h, r_h denote approximations in V_h to the right-hand sides and the coefficient functions, respectively. The edge residual is denoted by η_{e_D} and η_{e_J} comes from the jump of numerical solutions

$$\begin{aligned}
(\eta_{e_D}^y)^2 &= \frac{1}{2}\sum_{\Gamma_h^0}\epsilon^{-\frac{1}{2}}\rho_e\|[\epsilon\nabla y_h]\|_e^2, \\
(\eta_{e_J}^y)^2 &= \frac{1}{2}\sum_{\Gamma_h^0}(\frac{\sigma\epsilon}{h_e} + r_0 h_e + \frac{h_e}{\epsilon})\|[y_h]\|_e^2 + \sum_{\Gamma_h^\partial}(\frac{\sigma\epsilon}{h_e} + r_0 h_e + \frac{h_e}{\epsilon})\|[g_D - y_h]\|_e^2, \\
(\eta_{e_D}^p)^2 &= \frac{1}{2}\sum_{\Gamma_h^0}\epsilon^{-\frac{1}{2}}\rho_e\|[\epsilon\nabla p_h]\|_e^2, \\
(\eta_{e_J}^p)^2 &= \frac{1}{2}\sum_{\Gamma_h^0}(\frac{\sigma\epsilon}{h_e} + r_0 h_e + \frac{h_e}{\epsilon})\|[p_h]\|_e^2 + \sum_{\Gamma_h^\partial}(\frac{\sigma\epsilon}{h_e} + r_0 h_e + \frac{h_e}{\epsilon})\|[p_h]\|_e^2.
\end{aligned}$$

Then, a posteriori error estimators for optimal control problems are

$$\eta^y = \left(\sum_{E\in\xi_h}(\eta_E^y)^2\right)^{\frac{1}{2}}, \quad \eta^p = \left(\sum_{E\in\xi_h}(\eta_E^p)^2\right)^{\frac{1}{2}}, \quad \eta^u = \left(\sum_{E\in\xi_h}(\eta_E^u)^2\right)^{\frac{1}{2}}. \tag{4.1}$$

Additionally, we give data approximation terms by

$$
\begin{aligned}
(\theta_E^y)^2 &= \rho_E^2(\|f - f_h\|^2_{L^2(E)} + \|(\beta - \beta_h)\cdot\nabla y_h\|^2_{L^2(E)} + \|(r - r_h)y_h\|^2_{L^2(E)}),\\
(\theta_E^p)^2 &= \rho_E^2(\|(y_d)_h - y_d\|^2_{L^2(E)} + \|(\beta - \beta_h)\cdot\nabla p_h\|^2_{L^2(E)} + \|(r - \nabla\cdot\beta) - (r_h - \nabla\cdot\beta_h)p_h\|^2_{L^2(E)}).
\end{aligned}
$$

Then, the data approximation errors are

$$
\theta^y = \left(\sum_{E\in\xi_h}(\theta_E^y)^2\right)^{\frac{1}{2}}, \qquad \theta^p = \left(\sum_{E\in\xi_h}(\theta_E^p)^2\right)^{\frac{1}{2}}. \tag{4.2}
$$

Note that y, p and u stand for the state, the adjoint and the control, respectively.

Our a posteriori error indicators (4.1) can defined for $r_0 \geq 0$. However, for the proof of our reliability and efficiency estimate we need $r_0 > 0$. We will comment in Remark 4.2.8 below on how this assumption enters our proof. Our numerical examples in Section 4.3 indicate that the a posteriori error indicators (4.1) can also be used if $r_0 = 0$. However, no proof is available for this case yet. We note that the assumption $r_0 > 0$ is also made for the analysis of discretization schemes for convection dominated elliptic optimal control problems in the papers [18, 48, 54, 73, 104].

In the **MARK** step of the adaptive loop, we specify the edges and elements of the triangulation using the a posteriori error indicator defined in (4.1) that have to be selected for refinement in order to achieve a reduction of error. Most strategies that are in use are of heuristic type and realize some sort of equidistribution of the error based on either the maximum or the average of the local components of the error estimator. We use the bulk criterion firstly proposed by Döfler [41] since the approximation error is decreased by a fixed factor for each loop and thus the local refinement will convergence. It can be described such that for a given universal constant θ, we choose subsets $M_E \subset \xi_h$ such that the following bulk criterion is satisfied:

$$
\sum_{E\in\xi_h}(\eta_E)^2 \leq \theta \sum_{E\in M_E}(\eta_E)^2. \tag{4.3}
$$

Bigger θ produce more refinement of triangles in one loop and smaller θ yield more optimal grid with more refinement loops.

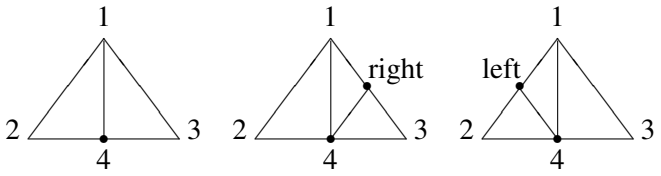


Figure 4.1: Divide a triangle according to the marked edges.

In the **REFINEMENT** step, the marked elements are divided using the newest vertex bisection (see, e.g., [31, 32, 45]). The procedure can be summarized as (see Figure 4.1): given a shape regular triangulation ξ_h of Ω, for each triangle $E \in \xi_h$, we label one vertex of E as a peak or newest vertex. The opposite edge of the peak is called as base or refinement edge. This process is called a labeling. Then, we divide a triangle according to the marked edges by satisfying;

1. a triangle is bisected to two new children triangles by connecting the peak to the midpoint of the base,

2. the new vertex created at a midpoint of a base is assigned to be the peak of the children.

After the labeling is done for initial triangulation, there is no more need to done it since the decent triangulation inherit the label by the rule (2). This refinement process conserves the conformity property and the shape regularity of the triangulation.

4.2 A Posteriori Error Estimation

We use an energy norm and a semi-norm associated with convective terms introduced in [91] to show the efficiency and the reliability of the error indicators defined in (4.1):

$$|||y|||^2 = \sum_{E\in\xi_h} (\|\epsilon\nabla y\|^2_{L^2(E)} + r_0\|y\|^2_{L^2(E)}) + \sum_{e\in\Gamma_h} \frac{\sigma\epsilon}{h_e}\|[y]\|^2_{L^2(e)}. \tag{4.4}$$

This norm can be viewed as the energy norm associated with the discontinuous Galerkin discretization of the convection diffusion.

The semi-norm $|.|_A$ with convective term is described such that

$$|y|_A^2 = |\beta y|_*^2 + \sum_{e\in\Gamma}(r_0 h_e + \frac{h_e}{\epsilon})|||[y]|||_{L^2(e)}^2, \tag{4.5}$$

where for $q \in L^2(\Omega)^2$,

$$|q|_* = \sup_{\upsilon\in H_0^1(\Omega)\setminus\{0\}} \frac{\int_\Omega q\cdot\nabla\upsilon dx}{|||\upsilon|||}. \tag{4.6}$$

The terms $|\beta y|_*^2$ and $h_e\epsilon^{-1}|||[y]|||_{L^2(e)}^2$ of the semi-norm $|.|_A$ will be used to bound the convective derivative, similarly as in [99]. The other term $r_0 h_e|||[y]|||_{L^2(e)}^2$ is related to the reaction term in the state or the adjoint equation.

We have to show two main properties of a posteriori estimators in (4.1) called as the reliability and the efficiency of estimator. We firstly give some necessary steps to show the proof of bounds. The proof is given as in [91] for the problems with homogeneous Dirichlet boundary conditions (3.2). In addition, the proof is proceeded by using the symmetric interior penalty Galerkin (SIPG) method ($\kappa = -1$ is taken in 3.19).

4.2.1 Auxiliary Forms and Their Properties

To make the discontinuous Galerkin form $a_h(y,\upsilon)$ in (3.19) well-define for functions $y,\upsilon \in H_0^1$, we use the following auxiliary forms as in [91]:

$$\begin{aligned}
D_h^y(y,\upsilon) &= \sum_{E\in\xi_h}\int_E(\epsilon\nabla y\cdot\nabla\upsilon + (r-\nabla\cdot\beta)y\upsilon)\,dx,\\
D_h^p(y,\upsilon) &= \sum_{E\in\xi_h}\int_E(\epsilon\nabla y\cdot\nabla\upsilon + ry\upsilon)\,dx,\\
O_h^y(y,\upsilon) &= -\sum_{E\in\xi_h}\int_E\beta y\cdot\nabla\upsilon\,dx + \sum_{e\in\Gamma_h^0}\int_e|\beta\cdot n|\,y^+(\upsilon^+-\upsilon^-)\,ds + \sum_{e\in\Gamma_h^+}\int_e|\beta\cdot n|\,y^+\upsilon^+\,ds,\\
O_h^p(y,\upsilon) &= \sum_{E\in\xi_h}\int_E\beta\upsilon\cdot\nabla y\,dx + \sum_{e\in\Gamma_h^0}\int_e|\beta\cdot n|\,(y^+-y^-)\upsilon_h^+\,ds + \sum_{e\in\Gamma_h^-}\int_e|\beta\cdot n|\,y^+\upsilon^+\,ds,\\
K_h(y,\upsilon) &= -\sum_{e\in\Gamma_h}\int_e\{\epsilon\nabla y\}[\upsilon]\,ds - \sum_{e\in\Gamma_h}\int_e\{\epsilon\nabla\upsilon\}[y]\,ds,\\
J_h(y,\upsilon) &= \sum_{e\in\Gamma_h}\frac{\epsilon\sigma}{h_e}\int_e[y][\upsilon]\,ds.
\end{aligned}$$

The bilinear form $\tilde{a}_h(y,\upsilon)$ is well-defined for all $y_h,\upsilon_h \in Y_h + H_0^1(\Omega)$:

$$\begin{aligned}
\tilde{a}_h(y,\upsilon) &= D_h^y(y,\upsilon) + O_h^y(y,\upsilon) + J_h(y,\upsilon)\\
&= D_h^p(y,\upsilon) + O_h^p(y,\upsilon) + J_h(y,\upsilon).
\end{aligned}$$

Then, the discontinuous Galerkin bilinear form $a_h(y,\upsilon)$ becomes

$$\tilde{a}_h(y_h,\upsilon_h) = a_h(y_h,\upsilon_h), \qquad \forall y,\upsilon_h \in H_0^1, \tag{4.7}$$

$$a_h(y_h,\upsilon_h) = \tilde{a}_h(y_h,\upsilon_h) + K_h(y_h,\upsilon_h), \qquad \forall y_h,\upsilon_h \in Y_h. \tag{4.8}$$

Lemma 4.2.1 *[91, Lemma 4.1]*

$$\tilde{a}_h(\upsilon,\upsilon) \geq |||\upsilon|||^2, \qquad \forall \upsilon \in H_0^1.$$

In this thesis, the symbols $\lesssim$ and $\gtrsim$ are used to denote bounds that are valid up to positive constants independent of the local mesh sizes, the diffusion coefficient ϵ and the penalty parameter σ, provided that $\sigma \geq 1$.

Lemma 4.2.2 *[91, Lemma 4.2] The auxiliary forms have continuity property such that*

$$\begin{aligned}
|D_h^y(y,\upsilon)| &\lesssim |||y|||\,|||\upsilon|||, && y,\upsilon \in Y_h + H_0^1,\\
|D_h^p(y,\upsilon)| &\lesssim |||y|||\,|||\upsilon|||, && y,\upsilon \in Y_h + H_0^1,\\
|J_h(y,\upsilon)| &\lesssim |||y|||\,|||\upsilon|||, && y,\upsilon \in Y_h + H_0^1,\\
|O_h^y(y,\upsilon)| &\lesssim |\beta y|_*\,|||\upsilon|||, && y \in Y_h + H_0^1, \upsilon \in H_0^1,\\
|O_h^p(y,\upsilon)| &\lesssim |\beta \upsilon|_*\,|||y|||, && \upsilon \in Y_h + H_0^1, y \in H_0^1.
\end{aligned}$$

Proof. The first one results from the Cauchy-Schwarz inequality and the bound in (3.7c). The second and third are straightforward consequence of the Cauchy-Schwarz inequality. The last two inequalities come from the definition of $|\cdot|_*$ given in (4.6). ■

Lemma 4.2.3 *[91, Lemma 4.3] For $y \in Y_h$ and $\upsilon \in H_0^1 \cap Y_h$, we have*

$$K(y,\upsilon) \lesssim \sigma^{-\frac{1}{2}} \left(\sum_{e\in\Gamma_h} \frac{\sigma\epsilon}{h_e} \|[y]\|_{L^2(e)}^2 \right)^{\frac{1}{2}} |||\upsilon|||.$$

Lemma 4.2.4 *[91, Lemma 4.4]* inf-sup *condition: There is constant $C > 0$ such that*

$$\inf_{y\in H_0^1(\Omega)\setminus\{0\}} \sup_{\upsilon\in H_0^1\setminus\{0\}} \frac{\tilde{a}(y,\upsilon)}{(|||y||| + |\beta y|_*)\,|||\upsilon|||} \geq C > 0.$$

Proof. The proof of this lemma is analogously to [99]. Let $y \in H_0^1$ and $\theta \in (0,1)$. Then there exits $w_\theta \in H_0^1$ such that

$$|||w_\theta||| = 1, \qquad O^y(y, w_\theta) = -\int_\Omega \beta y \cdot \nabla w_\theta \, dx \geq \theta |\beta y|_*.$$

From the continuity properties in Lemma 4.2.2, we obtain

$$\begin{aligned} \tilde{a}(y, w_\theta) &= D^y(y, w_\theta) + O^y(y, w_\theta) + J(y, w_\theta) \\ &\geq \theta |\beta y|_* - C|||y||| \, |||w_\theta||| \\ &= \theta |\beta y|_* - C|||y|||, \end{aligned}$$

where for a constant $C > 0$. Let us then define $\upsilon_\theta = y + \frac{|||y|||}{1+C} w_\theta$.

By Lemma 4.2.1, $a(y, y) \geq |||y|||^2$, so that

$$\begin{aligned} \sup_{\upsilon \in H_0^1(\Omega)\setminus\{0\}} \frac{\tilde{a}(y, \upsilon)}{|||\upsilon|||} &\geq \frac{\tilde{a}(y, \upsilon_\theta)}{|||\upsilon_\theta|||} \\ &\geq \frac{|||y|||^2 + (1+C)^{-1}|||y|||(\theta|\beta y|_* - C|||y|||)}{(1 + 1/(1+C))|||y|||} \\ &= \frac{1}{2+C}(|||y||| + \theta|\beta y|_*). \end{aligned}$$

Since $\theta \in (0,1)$ and $y \in H_0^1(\Omega)$ are arbitrary, we obtain the inf-sup condition.

Note that the proof can also be proceeded by using D_h^p and O_h^p. ■

4.2.2 Approximation Operators

Lemma 4.2.5 *For any $\upsilon \in V_h$, the following inequalities hold*

$$\begin{aligned} \sum_{E \in \xi_h} \|\upsilon - A_h \upsilon\|_{L^2(E)}^2 &\lesssim \sum_{e \in \Gamma_h} \int_e h_e |[\upsilon]|^2 \, ds, \\ \sum_{E \in \xi_h} \|\nabla(\upsilon - A_h \upsilon)\|_{L^2(E)}^2 &\lesssim \sum_{e \in \Gamma_h} \int_e h_e^{-1} |[\upsilon]|^2 \, ds, \end{aligned}$$

where the approximation operator $A_h : V_h \to V_h^c$ is defined in [58, 65], and Y_h^c be the conforming subspace of Y_h given by $V_h^c = Y_h \cap H_0^1(\Omega)$.

Lemma 4.2.6 *The interpolation operator constructed in [99]*

$$I_h : H_0^1(\Omega) \to \{\varphi \in C(\bar{\Omega}) : \ \varphi|_E \in P_1(E), \ \forall E \in \xi_h, \ \varphi = 0 \text{ on } \Gamma\}$$

that satisfies $|||I_h v||| \lesssim |||v|||$ *and*

$$\begin{aligned}\left(\sum_{E\in\xi_h} \rho_E^{-2}\|v - I_h v\|^2_{L^2(E)}\right)^{\frac{1}{2}} &\lesssim |||v|||,\\ \left(\sum_{e\in\Gamma} \epsilon^{\frac{1}{2}}\rho_e^{-1}\|v - I_h v\|^2_{L^2(e)}\right)^{\frac{1}{2}} &\lesssim |||v|||,\end{aligned}$$

where for any $v \in H_0^1(\Omega)$.

4.2.3 Reliability and Efficiency of a Posteriori Error Estimator

For the proof of the convergence a posteriori error estimation, we follow [74], firstly by establishing the connection between the control and the adjoint.

Lemma 4.2.7 *Let* (y, u, p) *and* (y_h, u_h, p_h) *be the solution of (3.12) and (3.18), respectively. Then*

$$\|u - u_h\|^2_{L^2(\Omega)} \lesssim \|p_h - p[u_h]\|^2_{L^2(\Omega)} + (\eta^u)^2, \tag{4.9}$$

where $p[u_h]$ *satisfies the following equation:*

$$a(y[u_h], w) - (u_h, w) = l(w), \qquad \forall w \in V, \tag{4.10a}$$

$$a(w, p[u_h]) + (y[u_h], w) = k(w), \quad \forall w \in V, \tag{4.10b}$$

where $l(w) = (f, w)$ *and* $k(w) = (y_d, w)$.

Proof. Let u be the optimal control, then $(y, u, p) = (y[v], u, p[v])$ is the solution of the system (4.10). The gradient of the objective function $\hat{J}(v)$ satisfies

$$(\nabla\hat{J}(v), w) \quad = \quad -(w, p[v]) + \omega(v, w), \qquad \forall w \in U.$$

Consider the following for any $w \in U$

$$\begin{aligned}(\nabla\hat{J}(u) - \nabla\hat{J}(u_h), w) &= (\nabla\hat{J}(u), w) - (\nabla\hat{J}(u_h), w)\\ &= -(w, p) + \omega(u, w) + (w, p[u_h]) - \omega(u_h, w)\\ &= (w, p[u_h] - p) + \omega(u - u_h, w).\end{aligned}$$

Setting $w = u - u_h$ leads to

$$\begin{aligned}(\nabla \hat{J}(u) - \nabla \hat{J}(u_h), u - u_h) &= (u - u_h, p[u_h] - p) + \omega(u - u_h, u - u_h) \\ &= (u - u_h, p[u_h] - p) + \omega \|u - u_h\|^2_{L^2(\Omega)}. \end{aligned} \tag{4.11}$$

The equations (4.10a) and (4.10b) yield

$$\begin{aligned}(u - u_h, p[u_h] - p) &= (u, p[u_h] - p) - (u_h, p[u_h] - p) \\ &= a(y, p[u_h] - p) - l(p[u_h] - p) - a(y[u_h], p[u_h] - p) + l(p[u_h] - p) \\ &= a(y - y[u_h], p[u_h] - p) \\ &= a(y - y[u_h], p[u_h]) - a(y - y[u_h], p) \\ &= k(y - y[u_h]) - (y[u_h], y - y[u_h]) - k(y - y[u_h]) + (y, y - y[u_h]) \\ &= (y - y[u_h], y - y[u_h]) = \|y - y[u_h]\|^2 \geq 0.\end{aligned}$$

The optimality condition of the unconstrained problem, $\nabla \hat{J}(u) = 0$, yields

$$\begin{aligned}\omega \|u - u_h\|^2_{L^2(\Omega)} &\leq (\nabla \hat{J}(u) - \nabla \hat{J}(u_h), u - u_h) \\ &= -(\nabla \hat{J}(u_h), u - u_h) \\ &= (u - u_h, p[u_h]) - \omega(u_h, u - u_h) \\ &= (p[u_h], u - u_h) - \omega(u_h, u - u_h) + (p_h, u - u_h) - (p_h, u - u_h) \\ &= (p[u_h] - p_h, u - u_h) + (p_h - \omega u_h, u - u_h) \\ &\lesssim \|p_h - p[u_h]\|^2_{L^2(\Omega)} + \|u - u_h\|^2_{L^2(\Omega)} + \|p_h - \omega u_h\|^2_{L^2(\Omega)} + \|u - u_h\|^2_{L^2(\Omega)}.\end{aligned}$$

Then, we obtain

$$\|u - u_h\|^2_{L^2(\Omega)} \lesssim \|p_h - p[u_h]\|^2_{L^2(\Omega)} + (\eta^u)^2.$$

■

Now, we need to find a bound for $\|p_h - p[u_h]\|^2_{L^2(\Omega)}$ in order to estimate $\|u - u_h\|^2_{L^2(\Omega)}$.

Remark 4.2.8 *If $r_0 = 0$, then, since $v \in V = H^1_0(\Omega)$, $(g, v) \leq \|g\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} \lesssim \epsilon^{-1}\|g\|_{L^2(\Omega)}|||v|||$, and therefore the constants in (4.12 and (4.15) would depend on ϵ^{-1}. The assumption $r_0 > 0$ makes the constant in the estimate $(g, v) \leq \|g\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} \lesssim \|g\|_{L^2(\Omega)}|||v|||$ independent of ϵ, as desired. In the following we will use the bound $\|v\|_{L^2(\Omega)} \lesssim |||v|||$ a few times, which is possible because of $r_0 > 0$.*

Lemma 4.2.9 *It holds*

$$|||p[u_h] - p_h||| + |p[u_h] - p_h|_A \lesssim \eta^p + \theta^p + \|y_h - y[u_h]\|_{L^2(\Omega)}. \tag{4.12}$$

Proof. Firstly, we rewrite the discontinuous Galerkin solution p_h in terms of a conforming part and a remainder as in [91]:

$$p_h = p_h^c + p_h^r,$$

where $p_h^c = A_h p_h \in Y_h^c$ with A_h as the operator from Lemma 4.2.5. The triangle inequality yields

$$|||p[u_h] - p_h||| + |p[u_h] - p_h|_A \leq |||p[u_h] - p_h^c||| + |p[u_h] - p_h^c|_A + |||p_h^r||| + |p_h^r|_A.$$

Similarly as in [91, Lemma 4.7], we can find a bound for the rest term p_h^r such that

$$|||p_h^r||| + |p_h^r|_A \lesssim \eta^p. \tag{4.13}$$

Now, we will find a bound for the continuous error $p[u_h] - p_h^c$ in terms of the adjoint error estimator, η^p in (4.1).

Since $p[u_h] - p_h^c$ is continuous, by definition of $|\cdot|_A$ in (4.5), we have $|p[u_h] - p_h^c|_A = |\beta(p[u_h] - p_h^c)|_*$.

The inf-sup condition in Lemma 4.2.4 gives

$$|||p[u_h] - p_h^c||| + |\beta(p[u_h] - p_h^c)|_* \lesssim \sup_{q \in H_0^1 \setminus \{0\}} \frac{\tilde{a}_h(q, p[u_h] - p_h^c)}{|||q|||}.$$

For $q \in H_0^1$,

$$\begin{aligned}
\tilde{a}_h(q, p[u_h] - p_h^c) &= \tilde{a}_h(q, p[u_h]) - \tilde{a}_h(q, p_h^c) \\
&= k(q) - (y[u_h], q) - \tilde{a}_h(q, p_h^c) \\
&= k(q) - (y[u_h], q) - D_h^p(q, p_h^c) - J_h(q, p_h^c) - O_h^p(q, p_h^c) \\
&= k(q) - (y[u_h], q) - \tilde{a}_h(q, p_h) + D_h^p(q, p_h^r) + J_h(q, p_h^r) + O_h^p(q, p_h^r) \\
&= \int_\Omega y_d q \, dx - (y[u_h], q) - \tilde{a}_h(q, p_h) + D_h^p(q, p_h^r) + J_h(q, p_h^r) + O_h^p(q, p_h^r).
\end{aligned}$$

By using (4.10b) and the operator I_h introduced in Lemma 4.2.6, we have

$$\begin{aligned}
\int_\Omega y_d I_h q \, dx &= a_h(I_h q, p_h) + (y_h, I_h q) \\
&= \tilde{a_h}(I_h q, p_h) + K_h(I_h q, p_h) + (y_h, I_h q).
\end{aligned}$$

$$
\begin{aligned}
\tilde{a}_h(q, p[u_h] - p_h^c) &= \int_\Omega y_d(q - I_h q)\, dx - \tilde{a}_h(q - I_h q, p_h) + D_h^p(q, p_h^r) + J_h(q, p_h^r) + O_h^p(q, p_h^r) \\
&+ K_h(I_h q, p_h) - (y[u_h], q) + (y_h, I_h q) + (y_h, q) - (y_h, q) \\
&= \int_\Omega (y_d - y_h)(q - I_h q)\, dx - \tilde{a}_h(q - I_h q, p_h) + D_h^p(q, p_h^r) + J_h(q, p_h^r) + O_h^p(q, p_h^r) \\
&+ K_h(I_h q, p_h) + (y_h - y[u_h], q) \\
&= T_1 + T_2 + T_3 + T_4.
\end{aligned}
$$

For $(q - I_h q) \in H_0^1$ and using integration by part, we obtain

$$
\begin{aligned}
T_1 &= \int_\Omega (y_d - y_h)(q - I_h q)\, dx - \tilde{a}_h(q - I_h q, p_h) \\
&= \sum_{E \in \xi_h} \int_E (y_d - y_h)(q - I_h q)\, dx - \left(\sum_{E \in \xi_h} \int_E (\epsilon \nabla (q - I_h q) \nabla p_h + r(q - I_h q) p_h)\, dx \right) \\
&- \sum_{E \in \xi_h} \int_E \beta \cdot \nabla (q - I_h q) p_h\, dx - \sum_{e \in \Gamma_h^0} \int_e |\beta \cdot n_e| ((q - I_h q)^+ - q - I_h q)^-) p_h^+\, ds \\
&= \underbrace{\sum_{E \in \xi_h} \int_E \left[(y_d - y_h) + \epsilon \Delta p_h + \beta \cdot \nabla p_h - (r - \nabla \cdot \beta) p_h \right] (q - I_h q)\, dx}_{M_1} \\
&- \underbrace{\sum_{e \in \Gamma_h} \int_e \epsilon \nabla p_h \cdot n_e (q - I_h q)\, ds}_{M_2} - \underbrace{\sum_{e \in \Gamma_h^0} \int_e |\beta \cdot n_e| (q - I_h q)^+ (p_h^+ - p_h^-)\, ds}_{M_3}.
\end{aligned}
$$

In the term M_1, we add and subtract the data approximation. This gives us

$$
\begin{aligned}
M_1 &= \sum_{E \in \xi_h} \int_E \left[(y_d - y_h) + \epsilon \Delta p_h + \beta \cdot \nabla p_h - (r - \nabla \cdot \beta) p_h \right] (q - I_h q)\, dx \\
&= \sum_{E \in \xi_h} \int_E \left[(y_d)_h - y_h + \epsilon \Delta p_h + \beta_h \cdot \nabla p_h - (r_h - \nabla \cdot \beta_h) p_h \right] (q - I_h q)\, dx \\
&+ \sum_{E \in \xi_h} \int_E \left[(y_d - (y_d)_h) - (\beta_h - \beta) \nabla p_h - ((r - \nabla \cdot \beta) - (r_h - \nabla \cdot \beta_h)) p_h \right] (q - I_h q)\, dx.
\end{aligned}
$$

Using the Cauchy-Schwarz inequality and Lemma 4.2.6, we obtain

$$
\begin{aligned}
M_1 &\lesssim \left(\sum_{E \in \xi_h} (\eta_{R_E}^p)^2 \right)^{\frac{1}{2}} \left(\sum_{E \in \xi_h} \rho_E^{-2} \|q - I_h q\|_{L^2(E)}^2 \right)^{\frac{1}{2}} + \left(\sum_{E \in \xi_h} (\theta_E^p)^2 \right)^{\frac{1}{2}} \left(\sum_{E \in \xi_h} \rho_E^{-2} \|q - I_h q\|_{L^2(E)}^2 \right)^{\frac{1}{2}} \\
&\lesssim \left(\sum_{E \in \xi_h} (\eta_R^p)^2 + (\theta_E^p)^2 \right)^{\frac{1}{2}} |||q|||.
\end{aligned}
$$

The term M_2 can be written in terms of the jump of $\epsilon\nabla p_h$; then we obtain

$$\begin{aligned} M_2 &= -\sum_{e\in\Gamma_h}\int_e \epsilon\nabla p_h\cdot n_e(q-I_hq)\,ds = -\sum_{e\in\Gamma_h^0}\int_e [\epsilon\nabla p_h](q-I_hq)\,ds \\ &\lesssim \left(\sum_{e\in\Gamma_h^0}\epsilon^{-\frac{1}{2}}\rho_e|||[\epsilon\nabla p_h]|||^2_{L^2(e)}\right)^{\frac{1}{2}}\left(\sum_{e\in\Gamma_h^0}\epsilon^{\frac{1}{2}}\rho_e^{-1}\|q-I_hq\|^2_{L^2(e)}\right)^{\frac{1}{2}} \\ &\lesssim \left(\sum_{E\in\xi_h}(\eta^p_{e_D})^2\right)^{\frac{1}{2}}|||q|||. \end{aligned}$$

By using the Cauchy-Schwarz inequality and Lemma 4.2.6 with the fact that $\rho_e \le h_E\epsilon^{-\frac{1}{2}}$, we obtain a bound for M_3 such that

$$\begin{aligned} M_3 &= -\sum_{e\in\Gamma_h^0}\int_e |\beta\cdot n_e|(q-I_hq)^+(p_h^+-p_h^-)\,ds \lesssim \left(\sum_{e\in\Gamma_h^0}\epsilon^{-\frac{1}{2}}\rho_e|||[p_h]|||^2_{L^2(e)}\right)^{\frac{1}{2}}\left(\sum_{e\in\Gamma_h^0}\epsilon^{\frac{1}{2}}\rho_e^{-1}\|q-I_hq\|^2_{L^2(e)}\right)^{\frac{1}{2}} \\ &\lesssim \left(\sum_{E\in\xi_h}(\eta^p_{e_J})^2\right)^{\frac{1}{2}}|||q|||. \end{aligned}$$

Combination of M_1, M_2, M_3 gives

$$T_1 \lesssim (\eta^p+\theta^p)|||q|||.$$

By the continuity result in Lemma 4.2.2 and the approximation property in (4.13), we obtain

$$\begin{aligned} T_2 &= D_h^p(q,p_h^r)+J_h(q,p_h^r)+O_h^p(q,p_h^r) \\ &\lesssim |||q||||||p_h^r||| + |||q||||||p_h^r||| + |\beta p_h^r|_*|||q||| \\ &= (|||p_h^r||| + |\beta p_h^r|_*)|||q||| \le \eta^p|||q|||. \end{aligned}$$

To bound T_3, use Lemma 4.2.3 with the $|||\cdot|||$-stability of the operator I_h defined in Lemma 4.2.6

$$T_3 = K(I_hq,p_h) \lesssim \sigma^{-\frac{1}{2}}\left(\sum_{e\in\Gamma_h}\frac{\epsilon\sigma}{h_e}|||[p_h]|||^2_{L^2(e)}\right)^{\frac{1}{2}}|||I_hq||| \lesssim \sigma^{-\frac{1}{2}}\left(\sum_{E\in\xi_h}(\eta^p_{e_J})^2\right)^{\frac{1}{2}}|||q|||.$$

Finally, we obtain a bound for T_4 by the Cauchy-Schwarz inequality and the definition of the norm $|||\cdot|||$ defined in (4.4):

$$T_4 = (y_h - y[u_h], q) \lesssim \sum_{E\in\xi_h}\|y_h-y[u_h]\|_{L^2(\Omega)}|||q|||.$$

Taking all bounds together for T_1, T_2, T_3, T_4, we obtain

$$|||p[u_h]-p_h^c||| + |p[u_h]-p_h^c|_A \lesssim \eta^p+\theta^p+\|y_h-y[u_h]\|_{L^2(\Omega)}. \tag{4.14}$$

Hence, combining (4.13) and (4.14), the proof is completed.

$$|||p[u_h] - p_h||| + |p[u_h] - p_h|_A \lesssim \eta^p + \theta^p + \|y_h - y[u_h]\|_{L^2(\Omega)}.$$

■

Now, we apply same procedure as adjoint case to find a bound for $\|y_h - y[u_h]\|_{L^2(\Omega)}$.

Lemma 4.2.10 *The following inequality*

$$|||y[u_h] - y_h||| + |y[u_h] - y_h|_A \lesssim \eta^y + \theta^y \tag{4.15}$$

holds.

Proof. Similarly, we decompose y_h into the conforming part and the remainder,

$$y_h = y_h^c + y_h^r.$$

Using the triangle inequality we obtain

$$|||y[u_h] - y_h||| + |y[u_h] - y_h|_A \leq |||y[u_h] - y_h^c||| + |y[u_h] - y_h^c|_A + |||y_h^r||| + |y_h^r|_A.$$

The rest term y_h^r is bounded by state error estimator as shown in [91, Lemma 4.7]:

$$|||y_h^r||| + |y_h^r|_A \lesssim \eta^y. \tag{4.16}$$

Now, we will prove that the continuous error $y[u_h] - y_h^c$ is bounded by the state error estimator, η^y given in (4.1).

Since $y[u_h] - y_h^c$ is continuous, by definition of $|\cdot|_A$ in (4.5), we have $|y[u_h] - y_h^c|_A = |\beta(y[u_h] - y_h^c)|_*$.

Then, we obtain using the inf-sup condition (Lemma 4.2.4),

$$|||y[u_h] - y_h^c||| + |\beta(y[u_h] - y_h^c)|_* \lesssim \sup_{v \in H_0^1 \setminus \{0\}} \frac{\tilde{a}_h(y[u_h] - y_h^c, v)}{|||v|||}.$$

For $v \in H_0^1$,

$$\begin{aligned}
\tilde{a}_h(y[u_h] - y_h^c, v) &= \tilde{a}_h(y[u_h], v) - \tilde{a}_h(y_h^c, v) \\
&= l(v) + (u_h, v) - \tilde{a}_h(y_h^c, v) \\
&= l(v) + (u_h, v) - D_h^y(y_h^c, v) - J_h(y_h^c, v) - O_h^y(y_h^c, v) \\
&= l(v) + (u_h, v) - \tilde{a}_h(y_h, v) + D_h^y(y_h^r, v) + J_h(y_h^r, v) + O_h^y(y_h^r, v) \\
&= \int_\Omega f v \, dx + (u_h, v) - \tilde{a}_h(y_h, v) + D_h^y(y_h^r, v) + J_h(y_h^r, v) + O_h^y(y_h^r v).
\end{aligned}$$

By using (4.10a) and the operator I_h introduced in Lemma 4.2.6, we have

$$\begin{aligned}\int_\Omega f I_h\upsilon\, dx &= a_h(y_h, I_h\upsilon) - (u, I_h\upsilon)\\ &= \tilde{a}_h(y_h, I_h\upsilon) + K_h(y_h, I_h\upsilon) - (u_h, I_h\upsilon).\end{aligned}$$

$$\begin{aligned}\tilde{a}_h(y[u_h] - y_h^c, \upsilon) &= \int_\Omega (f + u_h)(\upsilon - I_h\upsilon)\, dx - \tilde{a}_h(y_h, \upsilon - I_h\upsilon)\\ &+ D_h^y(y_h^r, \upsilon) + J_h(y_h^r, \upsilon) + O_h^y(y_h^r, \upsilon) + K_h(y_h, I_h\upsilon)\\ &= T_1 + T_2 + T_3.\end{aligned}$$

For $\upsilon - I_h\upsilon \in H_0^1$ and using integration by parts, we obtain

$$\begin{aligned}T_1 &= \int_\Omega (f + u_h)(\upsilon - I_h\upsilon)\, dx - \tilde{a}_h(y_h, \upsilon - I_h\upsilon)\\ &= \sum_{E\in\xi_h}\int_E (f + u_h)(\upsilon - I_h\upsilon)\, dx - \left[\sum_{E\in\xi_h}\int_E (\epsilon\nabla y_h\nabla(\upsilon - I_h\upsilon) + (r - \nabla\cdot\beta)y_h(\upsilon - I_h\upsilon))\, dx\right]\\ &+ \sum_{E\in\xi_h}\int_E \beta y_h\nabla(\upsilon - I_h\upsilon)\, dx - \sum_{e\in\Gamma_h^0}\int_e |\beta\cdot n_e| y_h^+((\upsilon - I_h\upsilon)^+ - (\upsilon - I_h\upsilon)^-)\, ds\\ &= \underbrace{\sum_{E\in\xi_h}\int_E (f + u_h + \epsilon\Delta y_h - \beta\cdot\nabla y_h - r y_h)(\upsilon - I_h\upsilon)\, dx}_{M_1}\\ &- \underbrace{\sum_{e\in\Gamma_h}\int_e \epsilon\nabla y_h\cdot n_e(\upsilon - I_h\upsilon)\, ds}_{M_2} + \underbrace{\sum_{e\in\Gamma_h^0} |\beta\cdot n_e|(y_h^+ - y_h^-)(\upsilon - I_h\upsilon)^+\, ds}_{M_3}.\end{aligned}$$

In the term M_1, we add and subtract the data approximation terms. This gives us

$$\begin{aligned}M_1 &= \sum_{E\in\xi_h}\int_E (f + u_h + \epsilon\Delta y_h - \beta\cdot\nabla y_h - r y_h)(\upsilon - I_h\upsilon)\, dx\\ &= \sum_{E\in\xi_h}\int_E (f_h + u_h + \epsilon\Delta y_h - \beta_h\nabla y_h - r_h y_h)(\upsilon - I_h\upsilon)\, dx\\ &+ \sum_{E\in\xi_h}\int_E \left[(f - f_h) - (\beta - \beta_h)\nabla y_h - (r - r_h)y_h\right](\upsilon - I_h\upsilon)\, dx.\end{aligned}$$

Using the Cauchy-Schwarz inequality and Lemma 4.2.6, we obtain

$$\begin{aligned}M_1 &\lesssim \left(\sum_{E\in\xi_h}(\eta_{R_E}^y)^2\right)^{\frac{1}{2}}\left(\sum_{E\in\xi_h}\rho_e^{-2}\|\upsilon - I_h\upsilon\|_{L^2(E)}^2\right)^{\frac{1}{2}} + \left(\sum_{E\in\xi_h}(\theta_E^y)^2\right)^{\frac{1}{2}}\left(\sum_{E\in\xi_h}\rho_e^{-2}\|\upsilon - I_h\upsilon\|_{L^2(E)}^2\right)^{\frac{1}{2}}\\ &\lesssim \left(\sum_{E\in\xi_h}(\eta_R^y)^2 + (\theta_E^y)^2\right)^{\frac{1}{2}}|||\upsilon|||.\end{aligned}$$

Now, we write $\epsilon\nabla y_h$ in terms of the jump to obtain

$$\begin{aligned} M_2 &= -\sum_{e\in\Gamma_h}\int_e \epsilon\nabla y_h\cdot n_e(\upsilon - I_h\upsilon)\,ds = -\sum_{e\in\Gamma_h^0}\int_e [\epsilon\nabla y_h](\upsilon - I_h\upsilon)\,ds \\ &\lesssim \left(\sum_{e\in\Gamma_h^0}\epsilon^{-\frac{1}{2}}\rho_e\|[\epsilon\nabla y_h]\|^2_{L^2(e)}\right)^{\frac{1}{2}}\left(\sum_{e\in\Gamma_h^0}\epsilon^{\frac{1}{2}}\rho_e^{-1}\|\upsilon - I_h\upsilon\|^2_{L^2(e)}\right)^{\frac{1}{2}} \\ &\lesssim \left(\sum_{E\in\xi_h}(\eta^y_{e_D})^2\right)^{\frac{1}{2}}|||\upsilon|||. \end{aligned}$$

By using the Cauchy-Schwarz inequality and Lemma 4.2.6 with the fact that $\rho_e \le h_E\epsilon^{-\frac{1}{2}}$, we obtain a bound for M_3 such that

$$\begin{aligned} M_3 &= \sum_{e\in\Gamma_h^0}\int_e |\beta\cdot n_e|(y_h^+ - y_h^-)(\upsilon - I_h\upsilon)^+\,ds \lesssim \left(\sum_{e\in\Gamma_h^0}\epsilon^{-\frac{1}{2}}\rho_e\|[y_h]\|^2_{L^2(e)}\right)^{\frac{1}{2}}\left(\sum_{e\in\Gamma_h^0}\epsilon^{\frac{1}{2}}\rho_e^{-1}\|\upsilon - I_h\upsilon\|^2_{L^2(e)}\right)^{\frac{1}{2}} \\ &\lesssim \left(\sum_{E\in\xi_h}(\eta^y_{e_J})^2\right)^{\frac{1}{2}}|||\upsilon|||. \end{aligned}$$

Now, combining M_1, M_2, M_3 to obtain

$$T_1 \lesssim (\eta^y + \theta^y)|||\upsilon|||.$$

The continuity result defined in Lemma 4.2.2 and the approximation property in (4.16) yield

$$\begin{aligned} T_2 &= D_h^y(y_h^r,\upsilon) + J_h(y_h^r,\upsilon) + O_h^y(y_h^r,\upsilon) \\ &\lesssim |||\upsilon|||||y_h^r||| + |||\upsilon|||||y_h^r||| + |\beta y_h^r|_*|||\upsilon||| \\ &\lesssim (|||y_h^r||| + |\beta y_h^r|_*)|||\upsilon||| \le \eta^y|||\upsilon|||. \end{aligned}$$

Lemma 4.2.3 and the $|||\cdot|||$-stability of the operator I_h defined in Lemma 4.2.6 give us

$$T_3 = K(y_h, I_h\upsilon) \lesssim \sigma^{-\frac{1}{2}}\left(\sum_{e\in\Gamma_h}\frac{\epsilon\sigma}{h_e}\|[y_h]\|^2_{L^2(e)}\right)^{\frac{1}{2}}\|I_h\upsilon\| \lesssim \sigma^{-\frac{1}{2}}\left(\sum_{E\in\xi_h}(\eta^y_{e_J})^2\right)^{\frac{1}{2}}|||\upsilon|||.$$

Combining the three bounds T_1, T_2, T_3, we obtain

$$|||y[u_h] - y_h^c||| + |y[u_h] - y_h^c|_A \lesssim \eta^y + \theta^y. \tag{4.17}$$

Hence, from (4.16) and (4.17) we obtain

$$|||y[u_h] - y_h||| + |y[u_h] - y_h|_A \lesssim \eta^y + \theta^y.$$

■

Theorem 4.2.11 *Let (y, u, p) and (y_h, u_h, p_h) be the solutions of (3.12) and (3.18), respectively. Let the error estimators η^y, η^p, η^u be defined by (4.1) and the data approximation errors θ^y, θ^p by (4.2). Then we have the a posteriori error bound*

$$\|u - u_h\|_{L^2(\Omega)} + |||y - y_h||| + |y - y_h|_A + |||p - p_h||| + |p - p_h|_A \lesssim \eta^u + \eta^y + \theta^y + \eta^p + \theta^p.$$

Proof. From (4.10a)-(4.10b) and (3.9), we have

$$a(y - y[u_h], \upsilon) = (u - u_h, \upsilon), \qquad \forall \upsilon \in V_h, \tag{4.18}$$

$$a(w, p - p[u_h]) = -(y - y[u_h], w), \qquad \forall w \in V_h. \tag{4.19}$$

Note that $|y - y[u_h]|_A = |\beta(y - y[u_h])|_*$, by the continuity of y and $y[u_h]$. By using inf-sup condition defined in Lemma 4.2.4 and taking $\upsilon = y - y[u_h]$, we obtain

$$\begin{aligned}
(|||y - y[u_h]||| + |y - y[u_h]|_A)\,|||y - y[u_h]||| &= (|||y - y[u_h]||| + |\beta(y - y[u_h])|_*)\,|||y - y[u_h]||| \\
&\lesssim \tilde{a}_h(y - y[u_h], y - y[u_h]) \\
&\lesssim a_h(y - y[u_h], y - y[u_h]) \\
&= (u - u_h, y - y[u_h]) \\
&\lesssim \|u - u_h\|_{L^2(\Omega)} |||y - y[u_h]|||.
\end{aligned}$$

Then,

$$|||y - y[u_h]||| + |y - y[u_h]|_A \lesssim \|u - u_h\|_{L^2(\Omega)}. \tag{4.20}$$

Using the same procedure for adjoint case, we obtain

$$|||p - p[u_h]||| + |p - p[u_h]|_A \lesssim \|y - y[u_h]\|_{L^2(\Omega)} \lesssim \|u - u_h\|_{L^2(\Omega)}. \tag{4.21}$$

Using the triangular inequality and the inequalities (4.20)-(4.21), we obtain

$$\begin{aligned}
|||y_h - y||| + |y_h - y|_A &\leq |||y_h - y[u_h]||| + |y_h - y[u_h])|_A + |||y[u_h] - y||| + |y[u_h] - y|_A \\
&\lesssim |||y_h - y[u_h]||| + |y_h - y[u_h]|_A + \|u - u_h\|_{L^2(\Omega)}.
\end{aligned} \tag{4.22}$$

$$\begin{aligned}
|||p_h - p||| + |p_h - p|_A &\leq |||p_h - p[u_h]||| + |p_h - p[u_h]|_A + |||p[u_h] - p||| + |p[u_h] - p|_A \\
&\lesssim |||p_h - p(u_h)||| + |p_h - p(u_h)|_A + \|u - u_h\|_{L^2(\Omega)}.
\end{aligned} \tag{4.23}$$

From Lemma 4.2.7 and the definition energy norm associated with DG defined in (4.4), we have

$$\|u - u_h\|_{L^2(\Omega)} \lesssim \eta^u + |||p_h - p[u_h]|||. \tag{4.24}$$

Combining the inequalities (4.12)-(4.15) and (4.20-4.24), we have

$$\|u-u_h\|_{L^2(\Omega)} + |||y-y_h||| + |y-y_h|_A + |||p-p_h||| + |p-p_h|_A \lesssim \eta^u + \eta^y + \theta^y + \eta^p + \theta^p. \quad (4.25)$$

■

For any interior edge $e \in \Gamma_h^0$, denote by ω_e the union of two elements that share it. We use element and edge bubble functions defined in [99] to derive the lower error bounds.

$$\|\psi_E\|_{L^\infty(E)} = 1, \quad \forall \psi_E \in H_0^1(E) \quad \textit{and} \quad \|\psi_e\|_{L^\infty(e)} = 1, \quad \forall \psi_e \in H_0^1(w_e). \quad (4.26)$$

The local energy norm $|||\cdot|||_D$ for a set of elements D is

$$|||y|||_D^2 = \sum_{E\in D} (\epsilon \|\nabla u\|_{L^2(E)}^2 + r_0 \|y\|_{L^2(E)}^2).$$

Lemma 4.2.12 *[99, Lemma 3.6] The following estimates hold for any element E, edge e, and polynomials υ and σ defined on elements and edges, respectively,*

$$\begin{aligned}
\|\upsilon\|_{L^2(E)}^2 &\lesssim (\upsilon, \psi_E \upsilon)_E, && (4.27)\\
|||\psi_E \upsilon|||_E &\lesssim \rho_E^{-1} \|\upsilon\|_{L^2(E)}, && (4.28)\\
\|\sigma\|_{L^2(e)}^2 &\lesssim (\sigma, \psi_e \sigma)_e, && (4.29)\\
\|\psi_e \sigma\|_{L^2(w_e)} &\lesssim \epsilon^{1/4} \rho_e^{1/2} \|\sigma\|_{L^2(e)}, && (4.30)\\
\|\psi_e \sigma\|_{w_e} &\lesssim \epsilon^{1/4} \rho_e^{-1/2} \|\sigma\|_{L^2(e)}. && (4.31)
\end{aligned}$$

In the last two inequalities, the polynomials σ defined on e is extended to $\mathbb{R}^2$ in a canonical fashion.

Lemma 4.2.13 *The following inequality*

$$\eta^y \lesssim |||y-y_h||| + |y-y_h|_A + \theta^y + \|u-u_h\|_{L^2(\Omega)}$$

holds.

Proof. By continuity of y, $[y] = 0$ then we obtain

$$\sum_{E\in\xi_h} (\eta_{e_J}^y)^2 \lesssim |||y-y_h||| + |y-y_h|_A. \quad (4.32)$$

Hence, we only need to show the efficiency of the indicators η_R^y and $\eta_{e_D}^y$, respectively.

Define $R_E = (f_h + u_h + \epsilon\Delta y_h - \beta_h \cdot \nabla y_h - r_h y_h)|_E$, and set $W|_E = \rho_E^2 R\psi_E$.

By inequality (4.27), we obtain

$$\begin{aligned}\sum_{E\in\xi_h}(\eta_E^y)^2 &= \sum_{E\in\xi_h}\rho_E^2\|R\|^2_{L^2(E)} \lesssim \sum_{E\in\xi_h}(R,\rho_E^2\psi_E R)_E \\ &= \sum_{E\in\xi_h}(R,W)_E = \sum_{E\in\xi_h}(f_h + u_h + \epsilon\Delta y_h - \beta_h\cdot\nabla y_h - r_h y_h, W)_E.\end{aligned}$$

The exact solution satisfies $(f + u + \epsilon\Delta y - \beta\cdot\nabla y - ry)|_E = 0$. Then, using integration by parts and addition and substraction of the exact data,

$$\begin{aligned}\sum_{E\in\xi_h}(\eta_E^y)^2 &\lesssim \sum_{E\in\xi_h}(f_h + u_h + \epsilon\Delta y_h - \beta_h\cdot\nabla y_h - r_h y_h, W)_E + \sum_{E\in\xi_h}(\beta\cdot\nabla y_h + r y_h, W) \\ &- \sum_{E\in\xi_h}(\beta\cdot\nabla y_h + r y_h, W) - \sum_{E\in\xi_h}(f + u + \epsilon\Delta y - \beta\cdot\nabla y - ry, W)_E \\ &\lesssim \sum_{E\in\xi_h}(\epsilon(\nabla(y-y_h),\nabla W)|_E - (\beta(y-y_h),\nabla W)_E) + \sum_{E\in\xi_h}((r - \nabla\cdot\beta)(y-y_h), W)_E \\ &+ \sum_{E\in\xi_h}((f_h - f) + (u_h - u) + (\beta - \beta_h)\cdot\nabla y_h + (r - r_h)y_h, W)_E.\end{aligned}$$

Here, $W|_{\partial E} = 0$ since $\psi_E \in H_0^1(E)$. Then, by the Cauchy-Schwarz inequality, the bound in (3.7c), the definition of $|\cdot|_A$ in (4.5) and the data approximation error θ_E^y given in (4.2), we obtain

$$\sum_{E\in\xi_h}(\eta_{R_E}^y)^2 \lesssim (|||y - y_h||| + |y - y_h|_A + \theta_A + \|u - u_h\|_{L^2(\Omega)})\left(\sum_{E\in\xi_h}|||W|||_E^2 + \rho_E^{-2}\|W\|^2_{L^2(E)}\right)^{\frac{1}{2}}.$$

By using the inequality (4.28) and (4.26), we obtain

$$|||W|||_E^2 \lesssim \rho_E^2\|R\|^2_{L^2(E)} \quad \text{and} \quad \rho_E^{-2}|||W|||^2_{L^2(E)} \lesssim \rho_E^2\|R\|^2_{L^2(E)}.$$

Then

$$\sum_{E\in\xi_h}(\eta_R^y)^2 \lesssim (|||y - y_h||| + |y - y_h|_A + \theta^y + \|u - u_h\|_{L^2(\Omega)})\left(\sum_{E\in\xi_h}(\eta_R^y)^2\right)^{\frac{1}{2}}.$$

Hence, we obtain

$$\left(\sum_{E\in\xi_h}(\eta_E^y)^2\right)^{\frac{1}{2}} \lesssim (|||y - y_h||| + |y - y_h|_A + \theta^y + \|u - u_h\|_{L^2(\Omega)}). \tag{4.33}$$

Now, we will give a bound related to $\eta_{e_D}^y$. Set

$$\kappa = \sum_{e\in\Gamma_h^0}\epsilon^{-\frac{1}{2}}\rho_e\|\epsilon\nabla y_h\|\psi_e.$$

Using the inequality defined in (4.29) and the property of y, saying that $[\epsilon\nabla y] = 0$ on interior edges, Γ_h^0, we obtain

$$\sum_{E\in\xi_h}(\eta_{e_D}^y)^2 \lesssim \sum_{e\in\Gamma_h^0}([\epsilon\nabla y_h],\kappa)_e = \sum_{e\in\Gamma_h^0}([\epsilon\nabla(y_h-y)],\kappa)_e.$$

To apply integration by parts over each of the two elements of w_e yields

$$\sum_{e\in\Gamma_h^0}([\epsilon\nabla(y_h-y)],\kappa)_e = \sum_{e\in\Gamma_h^0}\int_{w_e}(\epsilon(\Delta y_h-\Delta y)\kappa+\epsilon(\nabla y_h-\nabla y)\cdot\nabla\kappa)\,dx.$$

By the differential equation $-\epsilon\Delta y = f + u - \beta\cdot\nabla y - ry$ and addition and substraction of approximate data, we obtain

$$\begin{aligned}\sum_{E\in\xi_h}(\eta_{e_D}^y)^2 \;\lesssim\; & \sum_{e\in\Gamma_h^0}\int_{w_e}(f_h+u_h+\epsilon\Delta y_h-\beta_h\cdot\nabla y_h-r_hy_h)\kappa\,dx\\ &+\sum_{e\in\Gamma_h^0}\int_{w_e}((\beta\cdot\nabla(y_h-y)+r(y_h-y))\kappa+\epsilon(\nabla y_h-\nabla y)\cdot\nabla\kappa)\,dx\\ &+\sum_{e\in\Gamma_h^0}\int_{w_e}((f-f_h)+(u-u_h)+(\beta_h-\beta)\cdot\nabla y_h+(r_h-r)y_h)\kappa\,dx.\end{aligned}$$

Integration by parts over w_e of the convection term $\beta\cdot\nabla(y_h-y)$ yields

$$\sum_{E\in\xi_h}(\eta_{e_D}^y)^2 \lesssim T_1+T_2+T_3+T_4+T_5,$$

where

$$\begin{aligned}T_1 &= \sum_{e\in\Gamma_h^0}\int_{w_e}(f_h+u_h+\epsilon\Delta y_h-\beta_h\cdot\nabla y_h-r_hy_h)\kappa\,dx,\\ T_2 &= \sum_{e\in\Gamma_h^0}\int_{w_e}(r(y_h-y)\kappa+\epsilon(\nabla y_h-\nabla y)\cdot\nabla\kappa)\,dx,\\ T_3 &= -\sum_{e\in\Gamma_h^0}\int_{w_e}\beta(y_h-y)\cdot\nabla\kappa\,dx,\\ T_4 &= \sum_{e\in\Gamma_h^0}\int_{e}\beta\cdot[y_h]\kappa\,ds,\\ T_5 &= \sum_{e\in\Gamma_h^0}\int_{w_e}((f-f_h)+(u-u_h)+(\beta_h-\beta)\cdot\nabla y_h+(r_h-r)y_h)\kappa\,dx.\end{aligned}$$

For T_1, T_2, T_3, T_4 terms, we obtain following upper bounds as shown in [91, Lemma 4.12]

$$\begin{aligned} T_1 &\lesssim (|||y - y_h||| + |y - y_h|_A + \theta^y)\left(\sum_{E\in\xi_h}(\eta^y_{e_D})^2\right)^{\frac{1}{2}}, \\ T_2 &\lesssim |||y - y_h|||\left(\sum_{E\in\xi_h}(\eta^y_{e_D})^2\right)^{\frac{1}{2}}, \\ T_3 &\lesssim |y - y_h|_A\left(\sum_{E\in\xi_h}(\eta^y_{e_D})^2\right)^{\frac{1}{2}}, \\ T_4 &\lesssim |y - y_h|_A\left(\sum_{E\in\xi_h}(\eta^y_{e_D})^2\right)^{\frac{1}{2}}. \end{aligned}$$

Finally, the last term T_5 can be bounded using the Cauchy-Schwarz inequality:

$$T_5 \lesssim (\theta^y + \|u - u_h\|_{L^2(\Omega)})\left(\sum_{E\in\xi_h}\rho_e^{-2}\|\kappa\|^2_{L^2(E)}\right) \lesssim (\theta^y + \|u - u_h\|_{L^2(\Omega)})\left(\sum_{E\in\xi_h}(\eta^y_{e_D})^2\right)^{\frac{1}{2}}.$$

By combining the bounds of $T_1 - T_5$, we obtain

$$\left(\sum_{E\in\xi_h}(\eta^y_{e_D})^2\right)^{\frac{1}{2}} \lesssim |||y - y_h||| + |y - y_h|_A + \theta^y + \|u - u_h\|_{L^2(\Omega)}. \tag{4.34}$$

Combining (4.32-4.34), the proof is completed.

■

Lemma 4.2.14 *The following inequality holds:*

$$\eta^p \lesssim |||p - p_h||| + |p - p_h|_A + \theta^p + \|y - y_h\|_{L^2(\Omega)}.$$

Proof. We follow the same procedure in Lemma 4.2.13 to show the inequality.

By continuity of p, we have $[p] = 0$. Then, we obtain

$$\sum_{E\in\xi_h}(\eta^p_{e_J})^2 \lesssim |||p - p_h||| + |p - p_h|_A. \tag{4.35}$$

Define $R_E = (-(y_h - (y_d)_h) + \epsilon\Delta p_h + \beta_h \cdot \nabla p_h - (r_h - \nabla\cdot\beta_h)p_h)|_E$, and set $W|_E = \rho_E^2 R\psi_E$.

By inequality (4.27), we obtain

$$\begin{aligned} \sum_{E\in\xi_h}(\eta^p_E)^2 &= \sum_{E\in\xi_h}\rho_E^2\|R\|^2_{L^2(E)} \lesssim \sum_{E\in\xi_h}(R, \rho_E^2\psi_E R)_E \\ &= \sum_{E\in\xi_h}(R, W)_E = \sum_{E\in\xi_h}(-(y_h - (y_d)_h) + \epsilon\Delta p_h + \beta_h\cdot\nabla p_h - (r_h - \nabla\cdot\beta_h)p_h, W_E. \end{aligned}$$

The exact solution satisfies $(-(y-y_d)+\epsilon\Delta p+\beta\cdot\nabla p-(r-\nabla\cdot\beta)p)|_E=0$. Then, using the integration by parts and addition and substraction of the exact data,

$$\begin{aligned}\sum_{E\in\xi_h}(\eta_E^p)^2 &\lesssim \sum_{E\in\xi_h}(-(y_h-(y_d)_h)+\epsilon\Delta p_h+\beta_h\cdot\nabla p_h-(r_h-\nabla\cdot\beta_h)p_h,W)_E\\ &+\sum_{E\in\xi_h}(\beta\cdot\nabla p_h+rp_h,W)-\sum_{E\in\xi_h}(\beta\cdot\nabla p_h+(r-\nabla\cdot\beta)p_h,W)\\ &-\sum_{E\in\xi_h}(-(y-y_D)+\epsilon\Delta p+\beta\cdot\nabla p-(r-\nabla\cdot\beta)p,W)_E\\ &\lesssim \sum_{E\in\xi}(\epsilon(\nabla(p-p_h),\nabla W)_E+(\beta(p-p_h),\nabla W)_E+\sum_{E\in\xi}((r-\nabla\cdot\beta)(p-p_h),W)_E\\ &+\sum_{E\in\xi}(((y_d)_h-y_d)+(y-y_h)+(\beta_h-\beta)\cdot\nabla p_h+((r-\nabla\beta)-(r_h-\nabla\cdot\beta_h))p_h,W)_E.\end{aligned}$$

Here, $W|_{\partial E}=0$, since $\psi_E\in H_0^1(E)$. Then, by the Cauchy-Schwarz inequality, the bound in (3.7c), the definition of $|\cdot|_A$ in (4.5) and the data approximation error θ_E^y in (4.2), we obtain

$$\sum_{E\in\xi_h}(\eta_E^p)^2 \lesssim (|||p-p_h|||+|p-p_h|_A+\theta^p+\|y-y_h\|_{L^2(\Omega)})\left(\sum_{E\in\xi_h}|||W|||_E^2+\rho_E^{-2}|||W|||_{L^2(E)}^2\right)^{\frac{1}{2}}.$$

By using the inequality (4.28) and (4.26), we obtain

$$|||W|||_E^2\lesssim\rho_E^2\|R\|_{L^2(E)}^2 \quad\text{and}\quad \rho_E^{-2}|||W|||_{L^2(E)}^2\lesssim\rho_E^2\|R\|_{L^2(E)}^2.$$

This gives

$$\sum_{E\in\xi_h}(\eta_E^p)^2\lesssim(|||p-p_h|||+|p-p_h|_A+\theta^p+\|y-y_h\|_{L^2(\Omega)})\left(\sum_{E\in\xi_h}(\eta_E^p)^2\right)^{\frac{1}{2}}.$$

Hence, we obtain

$$\left(\sum_{E\in\xi_h}(\eta_E^p)^2\right)^{\frac{1}{2}}\lesssim(|||p-p_h|||+|p-p_h|_A+\theta^p+\|y-y_h\|_{L^2(\Omega)}). \tag{4.36}$$

Finally, using the same procedure as previous Lemma 4.2.13 for $\eta_{e_D}^p$ as $\eta_{e_D}^y$, we obtain

$$\left(\sum_{E\in\xi_h}(\eta_{e_D}^p)^2\right)^{\frac{1}{2}}\lesssim|||p-p_h|||+|p-p_h|_A+\theta^p+\|y-y_h\|_{L^2(\Omega)}. \tag{4.37}$$

Combining (4.35)-(4.37), we obtain

$$\eta^p\lesssim|||p-p_h|||+|p-p_h|_A+\theta^p+\|y-y_h\|_{L^2(\Omega)}.$$

■

Theorem 4.2.15 *Let* (y, u, p) *and* (y_h, u_h, p_h) *be the solution of (3.12) and (3.18). Let the error estimators* η^y, η^p, η^u *be defined by (4.1) and the data approximation errors* θ^y, θ^p *be defined by (4.2). Then we have such a lower bound:*

$$\eta^y + \eta^p + \eta^u \lesssim \|u - u_h\|_{L^2(\Omega)} + |||y - y_h||| + |y - y_h|_A + |||p - p_h||| + |p - p_h|_A + \theta^y + \theta^p.$$

Proof. In the discretized optimality condition, we have $\omega u_h = p_h$. Hence, the element residual of control indicator is $\eta_E^u = 0$. This implies that $\eta^u = 0$. Then, by using Lemma 4.2.13, Lemma 4.2.14 and the definition of $|||\cdot|||$ defined in (4.4), the proof is completed. ■

4.3 Numerical Results

In this section, we give several numerical results for convection dominated problems with boundary and/or interior layers: boundary layer in Example 4.3.1, a circular and a straight interior layers in Example 4.3.2, a single straight interior layer in Example 4.3.3, boundary layer and interior layer in Example 4.3.4 and, finally, boundary layer in Example 4.3.5. When the analytical solutions of the state and the adjoint are given, the Dirichlet boundary condition g_D, the source function f and the desired state y_d are computed from (3.2) and (3.10) using the exact state, adjoint and control. In all the examples, we use the SIPG method [5, 102] to discretize the diffusion term and the original upwind discretization [70, 86] for the convection term. At some examples, we have also tested other DG methods, i.e., NIPG1 (standard penalization), NIPG3 (superpenalized) and IIPG. Additionally, the marking parameter θ varies between 0.3-0.6.

Example 4.3.1 ***Boundary layer***

This following example has been studied by Collis, Heinkenschloss and Leykekhman [35, 48]. The problem data are given by

$$\Omega = [0, 1]^2, \quad \theta = 45^o, \quad \beta = (\cos\theta, \sin\theta)^T, \quad r = 0 \ \text{ and } \ \omega = 1.$$

The analytical solutions of the state and the adjoint given by

$$\begin{aligned} y(x_1, x_2) &= \eta(x_1)\eta(x_2), \\ p(x_1, x_2) &= \mu(x_1)\mu(x_2), \end{aligned}$$

where

$$\eta(z) = z - \frac{\exp((z-1)/\epsilon) - \exp(-1/\epsilon)}{1 - \exp(-1/\epsilon)},$$
$$\mu(z) = 1 - z - \frac{\exp(-z/\epsilon) - \exp(-1/\epsilon)}{1 - \exp(-1/\epsilon}.$$

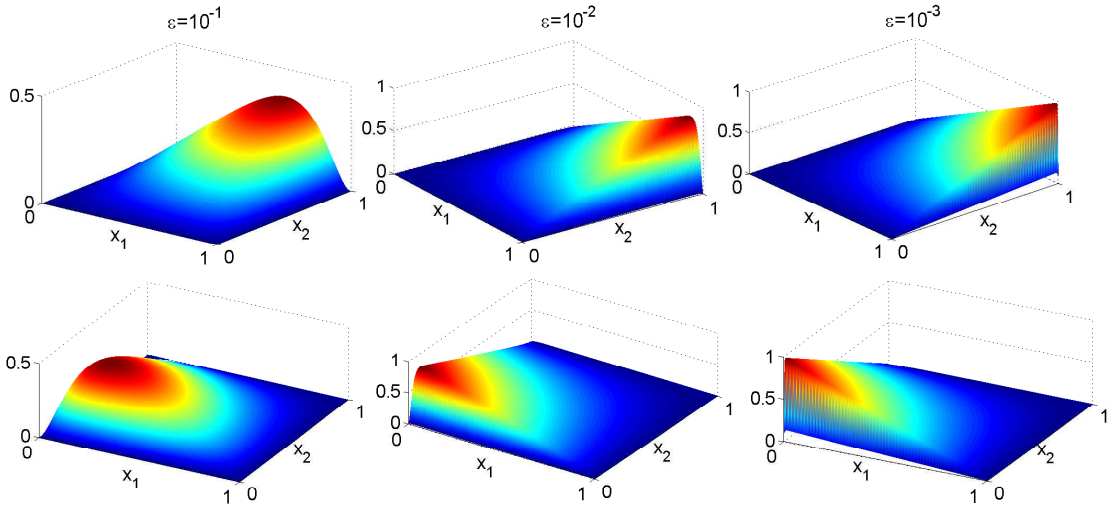


Figure 4.2: Surfaces of the exact state (top row) and the exact control (bottom row) for $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$ (fixed ϵ for each column) in Example 4.3.1.

For various diffusion parameters, the exact solutions of the state and the control are given in Figure 4.2. The solution of the state has a boundary layer on $x_1 = 1$ or $x_2 = 1$, whereas the control variable exhibits a boundary layer on on $x_1 = 0$ or $x_2 = 0$ as ϵ becomes smaller.

The initial mesh is generated by starting on uniform square mesh 16×16 and then dividing each square into two triangles. Let $\epsilon = 10^{-3}$, one can see from Figure 4.3, for uniformly refined mesh (16641 nodes), the spurious oscillations are present on the boundary layers. However, due to the weak treatment of the boundary conditions in DG methods, these oscillations do not propagate into the interior region in contrast to the SUPG method. See [48, 73] for details.

The reason why the spurious oscillations are present on uniformly refined mesh is that the boundary layers are not picked out well and, hence, are not solved properly. By applying an adaptive refinement procedure introduced in Section 4.1, we can select these boundary layers and make a refinement around these layers. However, the boundary layer in the state and the adjoint/control need to be resolved together, unlike in case of the single convection dominated PDEs. For example, when the boundary layer in the state is not solved properly,

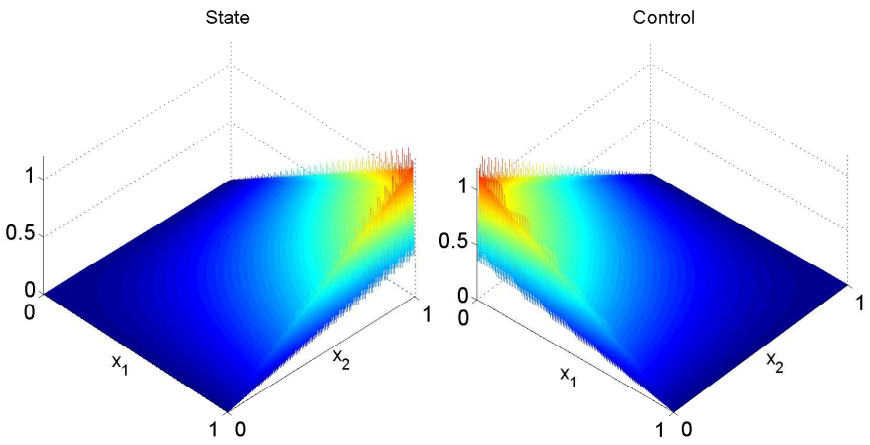


Figure 4.3: Computed solutions of the state (left) and the control (right) on uniformly refined mesh (16641 nodes) using linear elements for $\epsilon = 10^{-3}$ in Example 4.3.1.

the error on the state propagates through in the domain of adjoint. In adaptive procedure, the boundary layers in the state and the adjoint/control are resolved properly and then the spurious oscillations disappear. Figure 4.4 shows the computed state and control variables on adaptively refined mesh (15032 nodes) using linear elements.

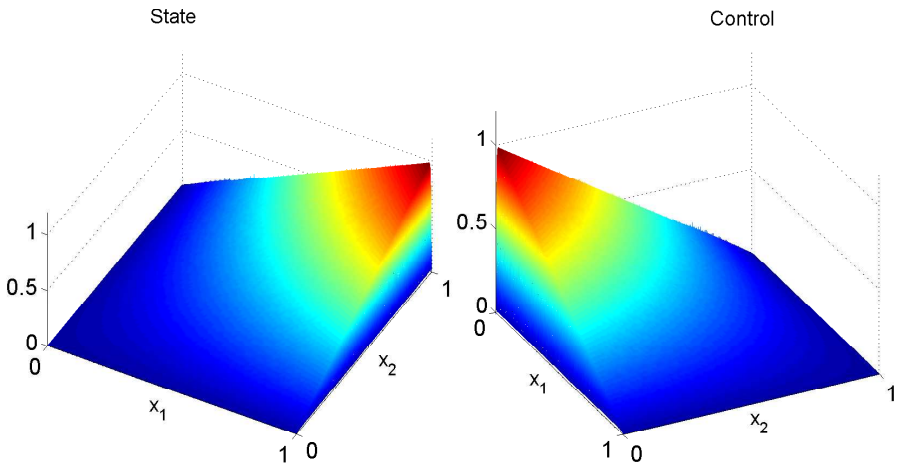


Figure 4.4: Computed solutions of the state (left) and the control (right) on adaptively refined mesh (15032 nodes) using linear elements for $\epsilon = 10^{-3}$ in Example 4.3.1.

Figure 4.5 shows the locally refined mesh by using the error indicator in (4.1). All refinements is done around the boundary layers of the state and the adjoint/control. Hence, this shows that the error indicator pick up the right elements to refine.

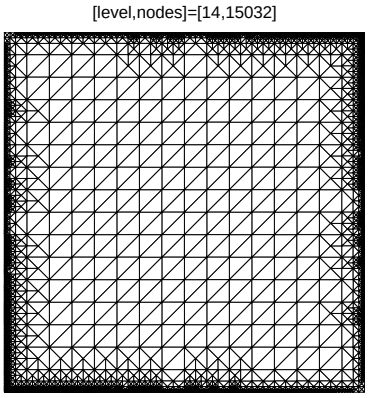


Figure 4.5: Adaptive mesh for $\epsilon = 10^{-3}$ in Example 4.3.1.

Figure 4.6 shows the global L_2 error for the state and the control with linear and quadratic elements for adaptively and uniformly refined meshes. Since the error indicator in (4.1) mostly picks out the boundary layers for refinement, the error decreases for the adaptive refinement more rapidly than for the uniform refinement. The error reduction for quadratic elements on adaptively refined mesh is visible with increasing number of nodes. The same example is examined with various error indicators, i.e., Zienkiewicz-Zhu estimation, the norm-residual based and the local Neumann problem estimator, in [81] using the SUPG method introduced in [35] for linear elements. The errors in Figure 4.6 decrease monotonically, whereas for SUPG in [81], the errors initially oscillate, after layers are sufficiently resolved they decrease monotonically.

All numerical results above are obtained with symmetric interior penalty Galerkin (SIPG) method. Although the error indicator in (4.1) is given for the SIPG method we want to observe the numerical results obtained from other DG methods, i.e., NIPG1 (standard NIPG method), IIPG and NIPG3 (superpenalized NIPG method) with the error indicator in (4.1). Figure 4.7 and Figure 4.8 exhibit L_2 error of various DG method with linear elements for the state and the control, respectively. Similar to SIPG case, the global errors on adaptively refined mesh do not exhibit any oscillation and, hence, decrease monotonically for all DG methods. The IIPG method gives more similar results with respect to SIPG method. The other interesting result is that the rate of convergence of the NIPG3 method is better than the one of the NIPG1 method on both uniformly and adaptively refined meshes. This supports the numerical results obtained in Chapter 3.

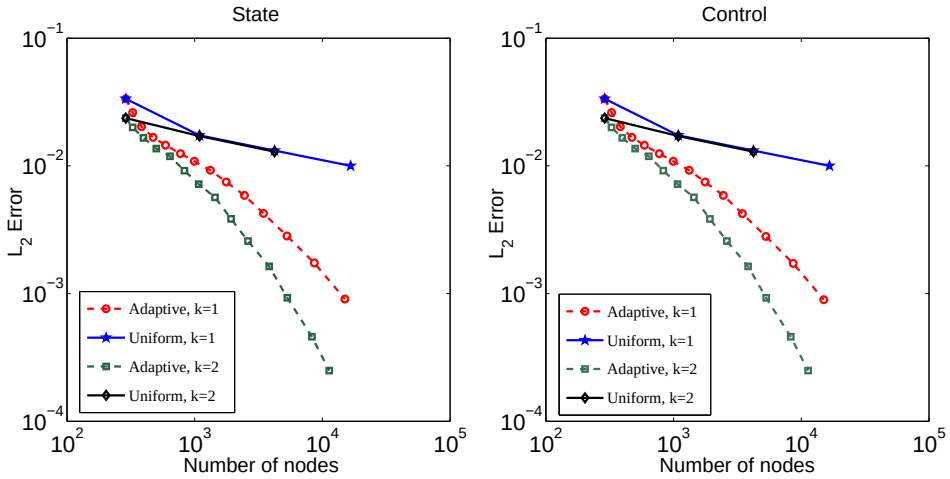


Figure 4.6: Errors in L^2 norm of the state (left) and the control (right) with linear and quadratic elements for $\epsilon = 10^{-3}$ in Example 4.3.1.

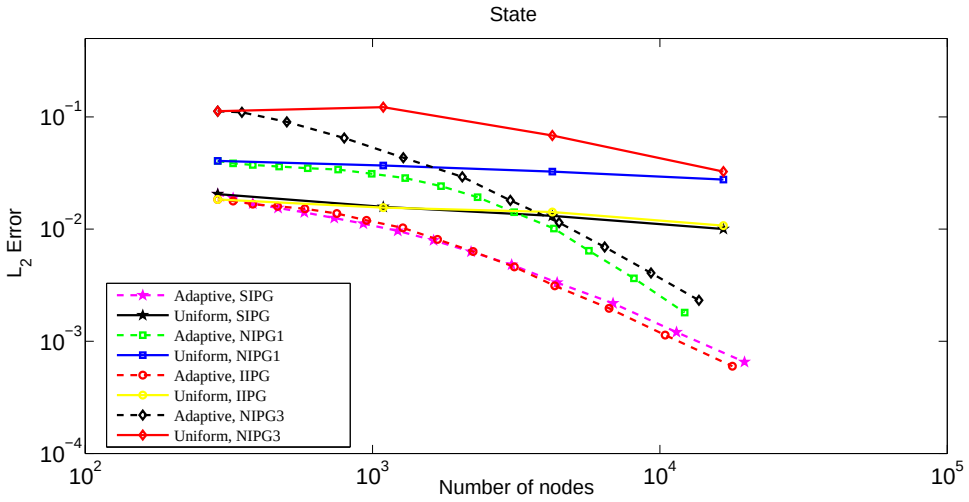


Figure 4.7: Errors in L^2 norm of the state with linear elements using various DG methods for $\epsilon = 10^{-3}$ in Example 4.3.1.

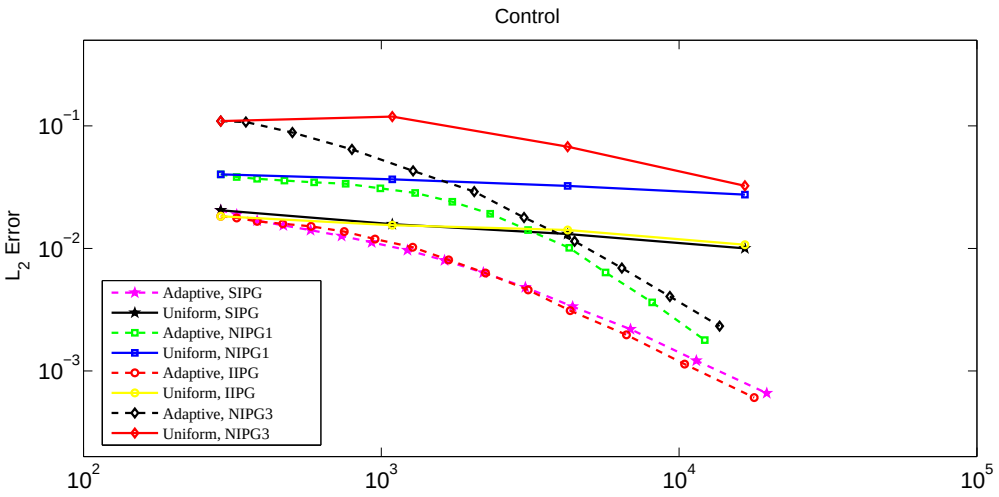


Figure 4.8: Errors in L^2 norm of the control with linear elements using various DG methods for $\epsilon = 10^{-3}$ in Example 4.3.1.

Example 4.3.2 ***Circular and straight interior layer***

This example has been studied by Hinze, Yan and Zhou [54] using the norm-residual based estimator with an edge stabilization technique for control constrained optimal control problems. Let

$$\Omega = [0,1]^2, \quad \beta = (2,3)^T, \quad r = 1 \ \text{ and } \ \omega = 0.1.$$

The analytical solution of state is given by

$$y(x_1,x_2) \;=\; \frac{2}{\pi}\arctan\left(\frac{1}{\sqrt{\epsilon}}\left[-\frac{1}{2}x_1 + x_2 - \frac{1}{4}\right]\right),$$

which is a function with a straight interior layer. The corresponding adjoint is

$$\begin{aligned} p(x_1,x_2) \;&=\; 16x_1(1-x_1)x_2(1-x_2) \\ &\times\; \left(\frac{1}{2} + \frac{1}{\pi}\arctan\left[\frac{2}{\sqrt{\epsilon}}\left(\frac{1}{16} - \left(x_1 - \frac{1}{2}\right)^2 - \left(x_2 - \frac{1}{2}\right)^2\right)\right]\right), \end{aligned}$$

which is a function with a circular interior layer.

Figure 4.9 shows the optimal state and control for $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$. As ϵ becomes smaller, the state exhibits a straight interior layer and the control exhibits a circular interior layer. Figure 4.10 shows the numerical solutions of state and control on the uniform mesh (289 nodes) with linear elements for various values of ϵ. The propagations on the interior layer in the direction $\pm\beta$ are visible for $\epsilon = 10^{-6}$.

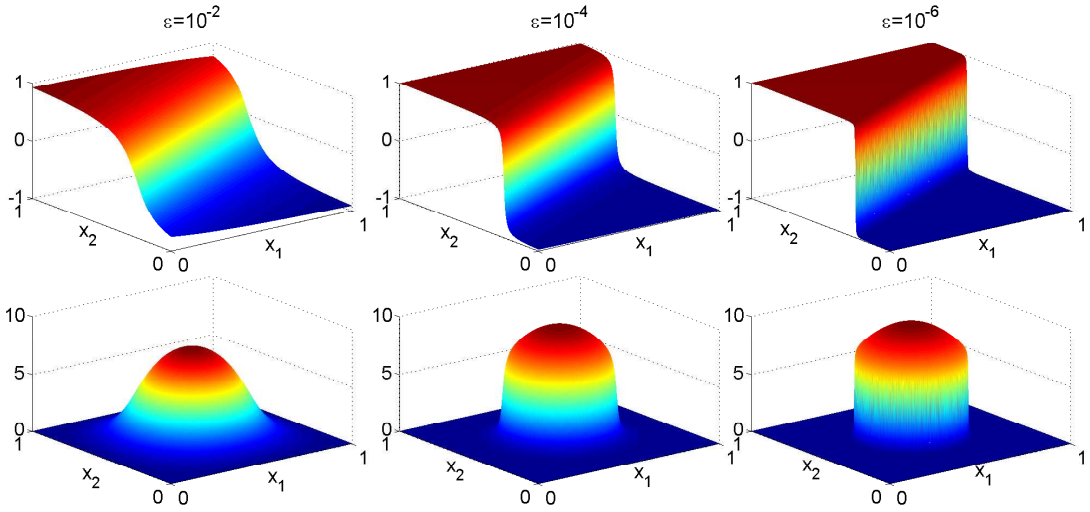


Figure 4.9: Surfaces of the exact state (top row) and the exact control (bottom row) for $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$ (fixed ϵ for each column) in Example 4.3.2.

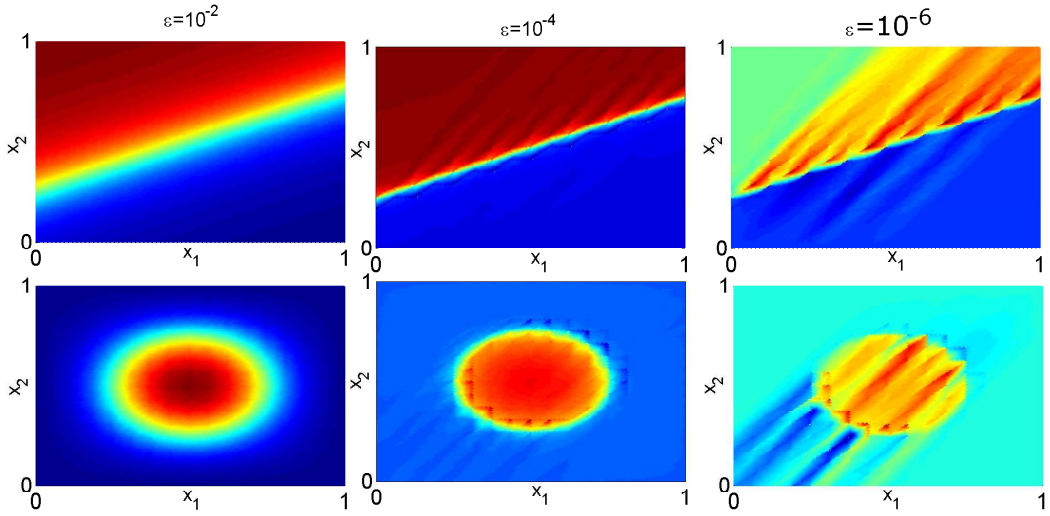


Figure 4.10: Computed state (top row) and control (bottom row) on uniform mesh (129 nodes) for $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$ with linear elements (fixed ϵ per column) in Example 4.3.2.

The initial mesh is generated by starting on a uniform square mesh 8×8 and then dividing each square into two triangles. When the numerical solutions of the state and the control are computed on uniformly refined mesh (16641 nodes) with linear elements for various values of ϵ, Figure 4.11 shows that oscillations are more visible for $\epsilon = 10^{-6}$ case. By applying adaptive procedure, we reduce these oscillations; see Figure 4.12.

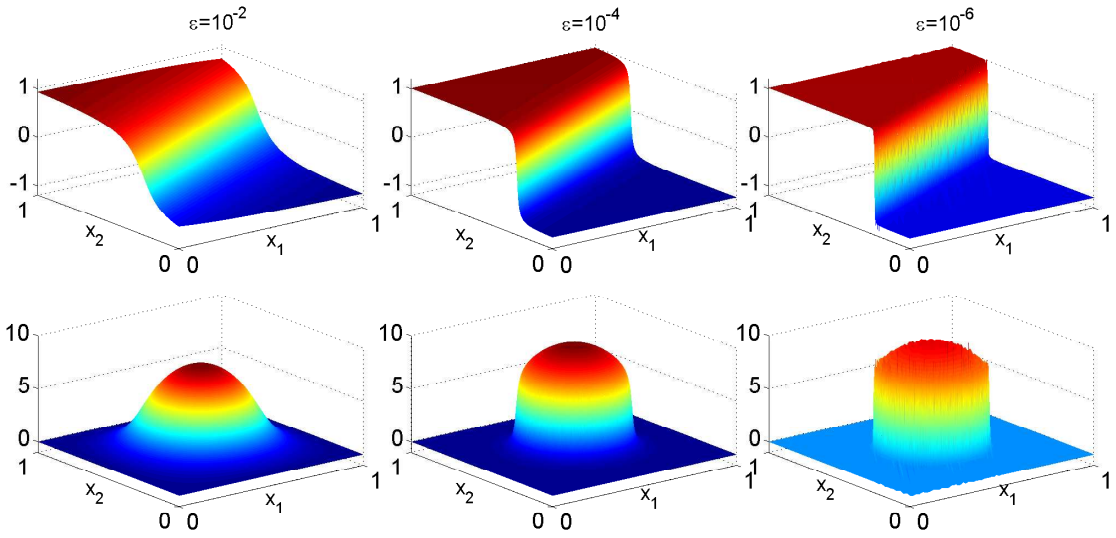


Figure 4.11: Computed state (top row) and control (bottom row) on uniform mesh (16641 nodes) with linear elements for $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$ in Example 4.3.2.

Figure 4.13 shows that the locally refined meshes generated by the error indicator in (4.1) for various values of ϵ. Our indicator picks out the layers reasonably well especially for small values of $\epsilon = 10^{-6}$. Same example has been studied in [81] using SUPG method with five generated error estimators. None of error estimators in [81] pick out the layers for $\epsilon = 10^{-6}$ as the error estimator in (4.1).

In Figure 4.14(a) and Figure 4.14(b), the errors in L_2 norm are given on adaptively and uniformly refined meshes using linear and quadratic elements and $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$ for the state and the control, respectively. For high values of ϵ, we do not see any effect of the error estimator (4.1) in Figure 4.14, but the numerical results on adaptively refined mesh are more accurate for small values of ϵ. Additionally, Figure 4.14 reveals that the difference of solutions between linear and quadratic elements decreases as ϵ becomes smaller on a uniformly refined mesh, since quadratic elements are not so effective when the right region of domain is not resolved. However, the more accurate results are obtained when quadratic elements are used in adaptive loop. Similar to the previous example, the errors decrease monotonically at

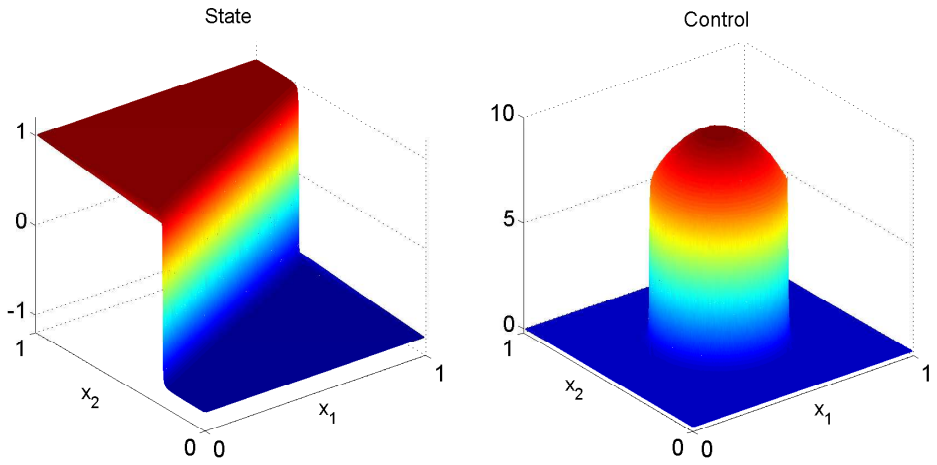


Figure 4.12: Computed state (left) and control (right) on an adaptive refinement mesh with linear elements for $\epsilon = 10^{-6}$ in Example 4.3.2.

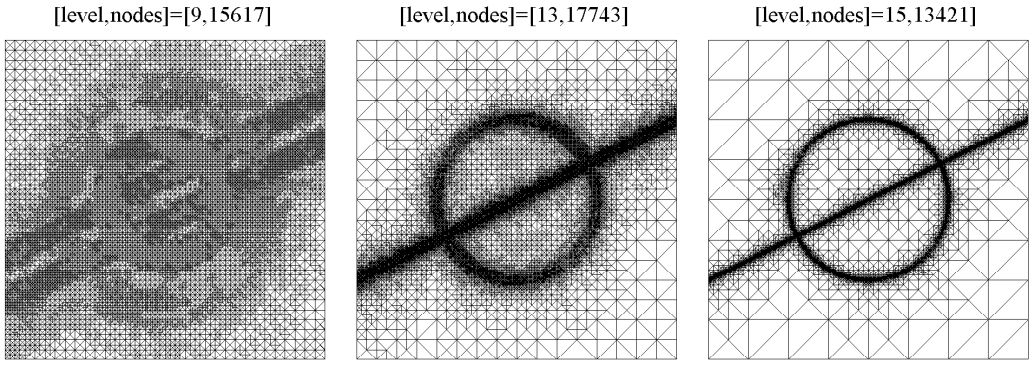
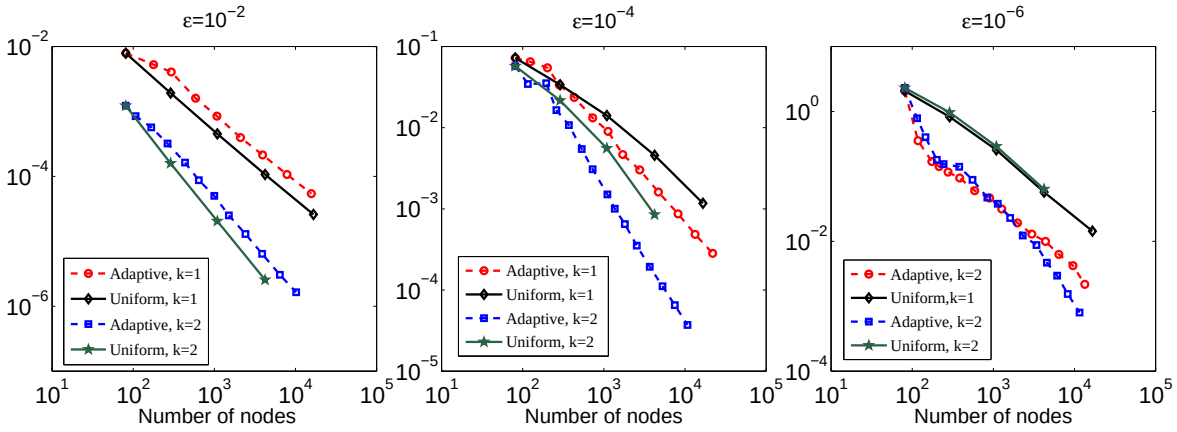
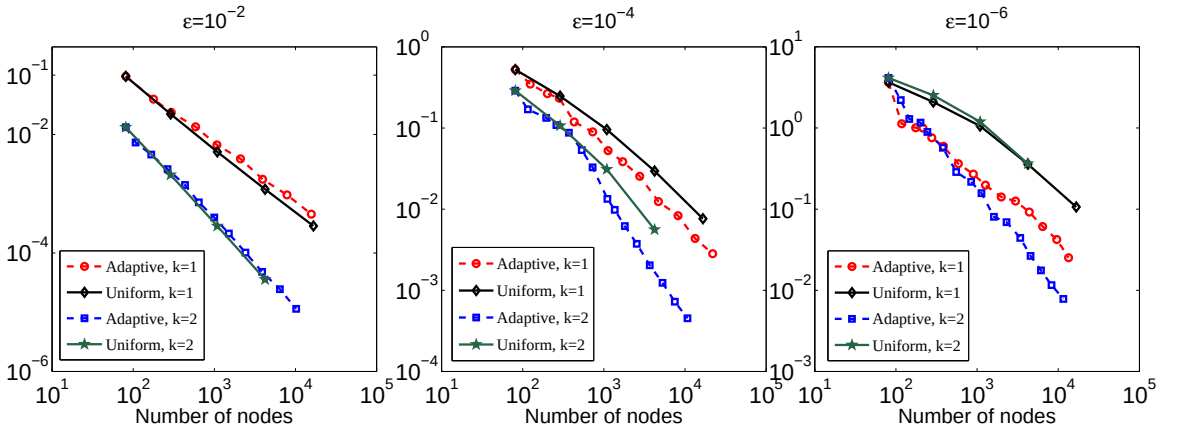


Figure 4.13: Adaptively refined meshes with linear elements for $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$ in Example 4.3.2.



(a) State



(b) Control

Figure 4.14: Errors in L_2 norm of state (top row) and control (bottom row) with linear and quadratic elements for $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$ in Example 4.3.2.

all cases in Figure 4.14, whereas the global errors in [81] have oscillations for the convection dominated case ($\epsilon = 10^{-6}$).

We have also studied the extremely convection dominated case, $\epsilon = 10^{-8}$, to see how the adaptivity resolves the layers. Figure 4.16 reveals the locally refined meshes at the different levels using linear elements. We observe that firstly the straight interior layer is refined then the circular interior layer is refined. Hence, we can expect more accurate results for the state with respect to the control. Figure 4.15 reveals that large oscillations occur when initial mesh is refined uniformly. However, picking out the layers by using the error indicator given in (4.1), the oscillations are reduced by adaptive procedure.

In Figure 4.17, the global errors with L_2 norm of the state and the control using linear elements are given. As being the case in previous example, we do not obtain a monotonically decrease for the error of adaptive refinement since ϵ is so small.

Example 4.3.3 ***Single straight interior layer***

The following example has firstly been studied by Heinkenschloss and Leykekhman in [48]. The problem data are given by

$$\Omega = [0,1]^2, \quad \epsilon = 10^{-7}, \quad \beta = (1,2)^T, \quad r = 0 \ \textit{and} \ \omega = 10^{-2}.$$

The true state and adjoint are defined by

$$\begin{aligned} y(x_1, y_1) &= (1-x_1)^3 \arctan\left(\frac{x_2 - 0.5}{\epsilon}\right), \\ p(x_1, x_2) &= x_1(1-x_1)x_2(1-x_2). \end{aligned}$$

Figure 4.18 shows the exact state and control for $\epsilon = 10^{-7}$. The state exhibits a sharp interior layer along the line $x_2 = 0.5$, whereas the control is very smooth. Because of the coupling of state and adjoint (or control), the control is not solved well on $x_2 = 0.5$, despite the fact that the exact control is smooth. This can be observed in Figure 4.19. The errors on a uniformly refined mesh (16641 nodes) are larger than the ones on adaptively refined mesh (9252 nodes).

The initial mesh is constructed by beginning with a uniform square mesh 16×16 and, then, dividing each square into two triangles. Figure 5.12 shows the meshes generated by the error estimator in (4.1). The left one in Figure 5.12 is obtained by using linear elements, whereas

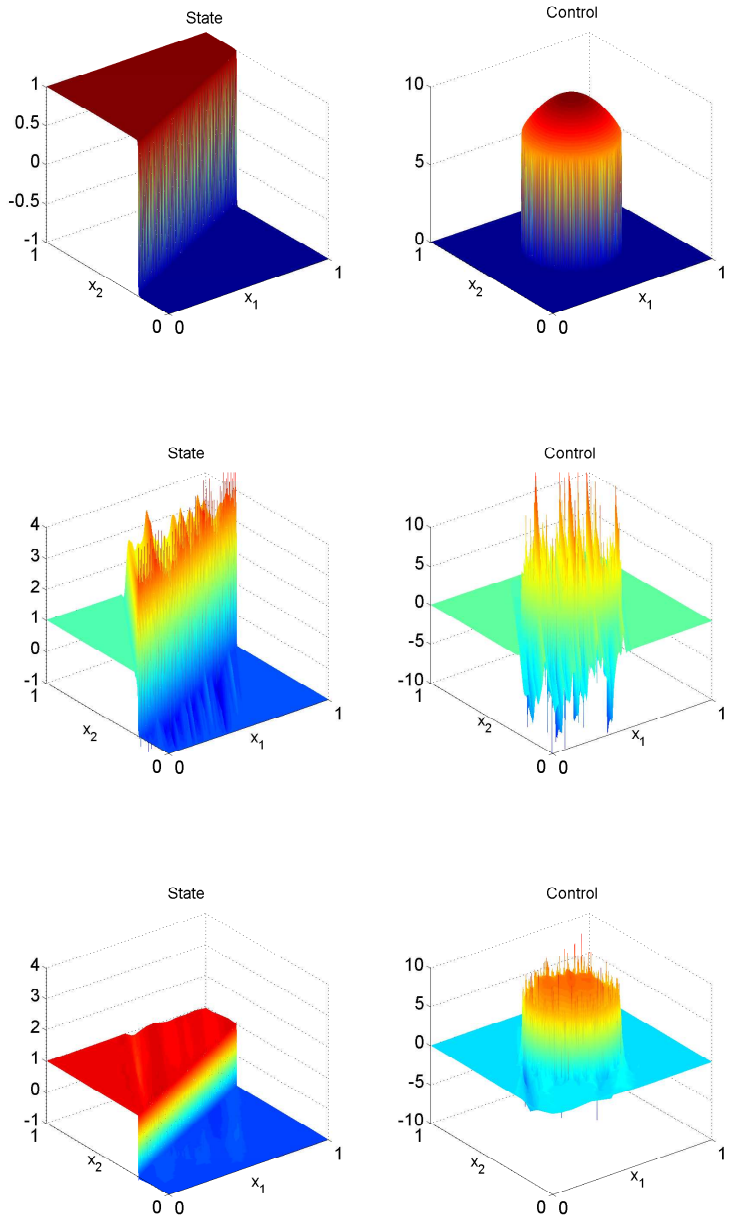


Figure 4.15: The top plot shows exact state and control. The middle plot exhibits solutions of the state (left) and the control (right) on uniformly refined mesh (16641 nodes) and the bottom plot shows on adaptively refined mesh (12257 nodes) with linear elements for $\epsilon = 10^{-8}$ in Example 4.3.2.

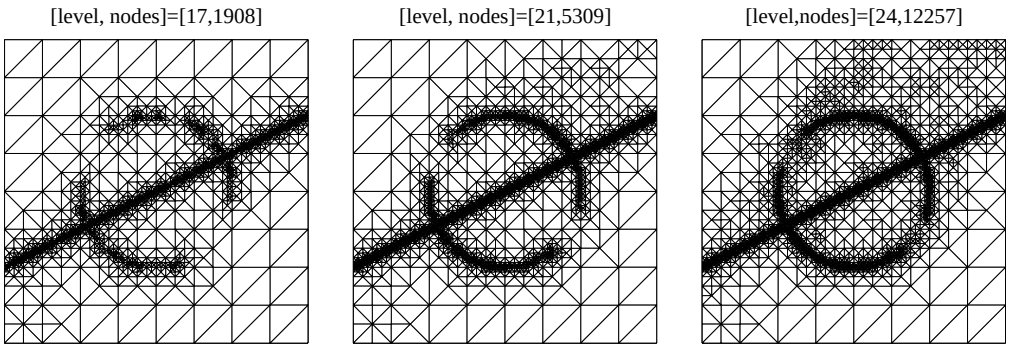


Figure 4.16: Generated locally refined meshes with linear elements at various refinement levels for $\epsilon = 10^{-8}$ in Example 4.3.2.

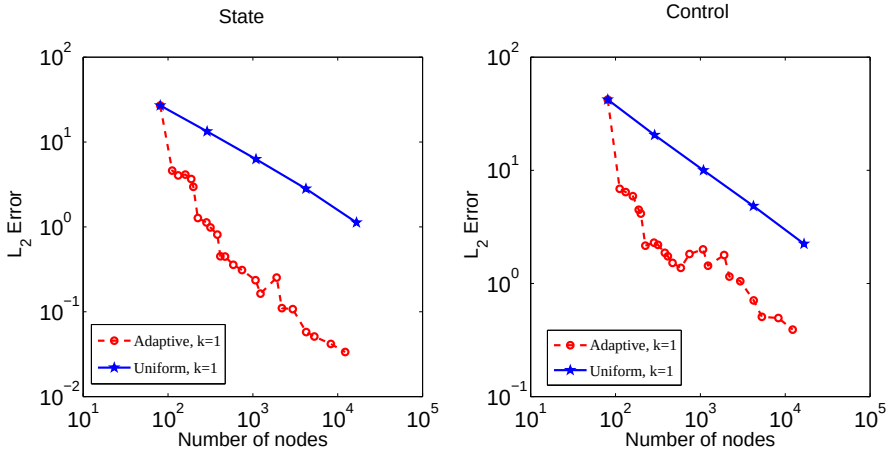


Figure 4.17: Errors in L_2 norm of state (left) and control (right) with linear elements for $\epsilon = 10^{-8}$ in Example 4.3.2.

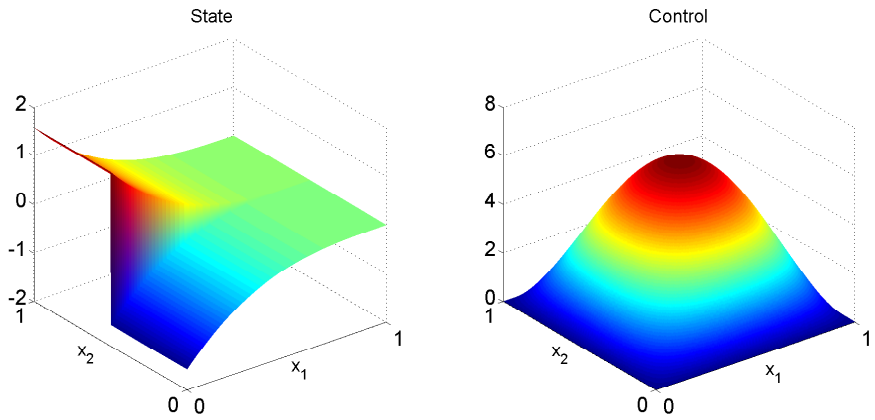


Figure 4.18: Surfaces of the exact state (left) and the exact control (right) for $\epsilon = 10^{-7}$ in 4.3.3.

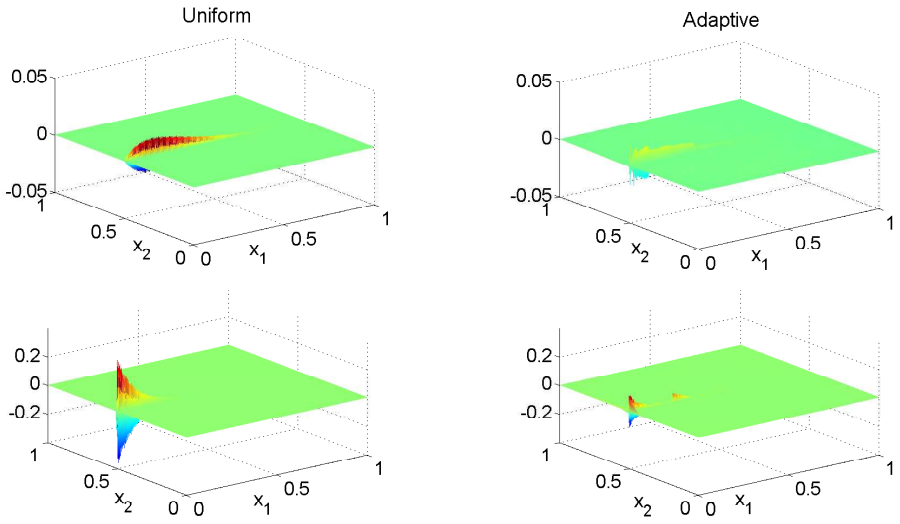


Figure 4.19: Error between the exact solution and the numerical solution on uniformly refined mesh (16641 nodes) and adaptively refined mesh (9252 nodes) using linear elements for $\epsilon = 10^{-7}$ in Example 4.3.3: state (top row), control (bottom row).

the right ones is obtained from quadratic elements. The error indicator mark the elements on $x_2 = 0.5$ where the state has a sharp interior layer.

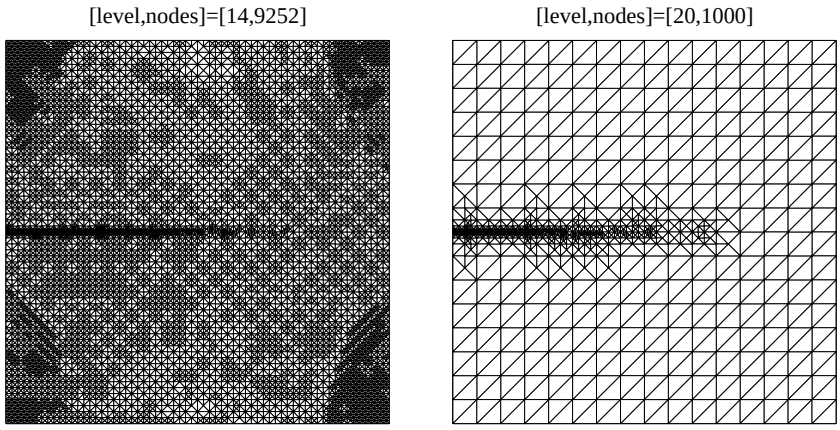


Figure 4.20: Adaptively refined meshes with linear elements (left,9252 nodes) and quadratic elements (right, 1000 nodes) for $\epsilon = 10^{-7}$ in Example 4.3.3.

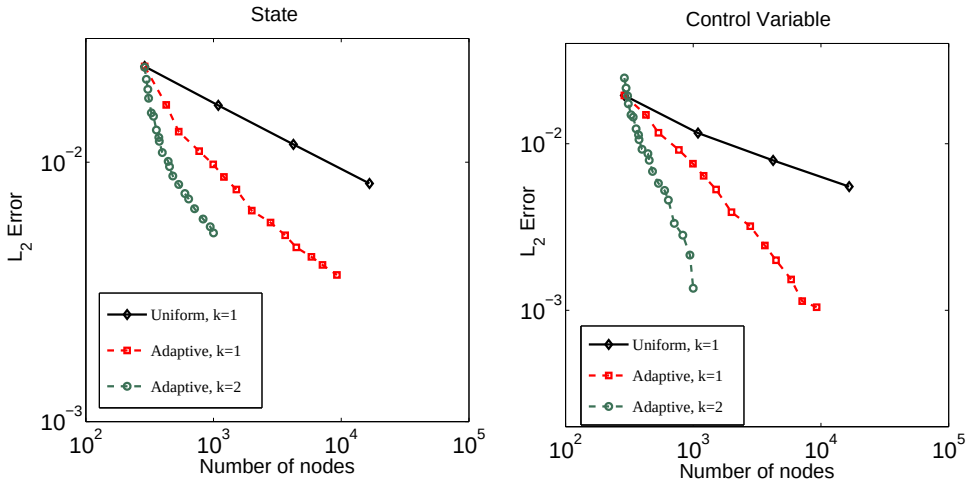


Figure 4.21: Errors in L^2 norm of state (left) and control (right) on uniformly and adaptively refined mesh for $\epsilon = 10^{-7}$ in Example 4.3.3.

The L_2 global error of the computed state and control are given in Figure 4.21. When the adaptive procedure is combined with quadratic elements, the more accurate results are obtained. Comparing the results in Figure 4.21 with in [81] using SUPG, it turns out that the estimators in [81] do not work for the control. In addition, in [81] the error of state is almost 10^{-2} using linear elements for approximately 12000 nodes, whereas it is much less than 10^{-2}

using linear elements with 9252 nodes at our case.

Example 4.3.4 ***Interior and boundary layer with unknown solutions***

This example is taken from [48]. Let

$$\Omega = [0,1]^2, \quad \epsilon = 10^{-4}, \quad \beta = (\cos\theta, \sin\theta)^T, \quad \theta = 47.3^o, \quad r = 0 \ \ and \ \ f = 0.$$

The Dirichlet boundary conditions are defined by

$$g_D(x_1, x_2) = \begin{cases} 1, & if\ x_1 = 0\ and\ x_2 = 0.25, \\ 1, & if\ x_2 = 0, \\ 0, & else. \end{cases}$$

The desired state is given by

$$y_d(x_1, x_2) = 1.$$

The control variable u is controlled by the the regularization parameter ω. Since the control u can be seen as the sole forcing term in the system, it enables to study the effect of the regularization parameter ω on adaptivity.

The exact solutions of the state, the adjoint and the control for this problem are not known. The computed solutions on the uniform refined mesh (289 nodes) using linear elements for $\omega = 1, 10^{-2}, 10^{-4}$ are shown in Figure 4.22 and 4.23. The state exhibits a sharp boundary layer and a straight interior layer. The interior layer in the state disappears as ω becomes smaller.

The initial mesh is constructed by beginning with a uniform square mesh 4×4 and then dividing each square into two triangles. Figures 4.24, 4.26 and 4.28 show the refined meshes at various refined levels for $\omega = 1, 10^{-2}$ and 10^{-4}, respectively. For large ω, the error indicator in (4.1) firstly marks the elements on the boundary layer, then the straight inner layer is resolved.

Figures 4.25, 4.27 and 4.29 reveal the computed solutions of the state and the control using linear and quadratic elements for $\omega = 1, 10^{-2}$ and 10^{-4}, respectively. The oscillations are reduced by using quadratic elements.

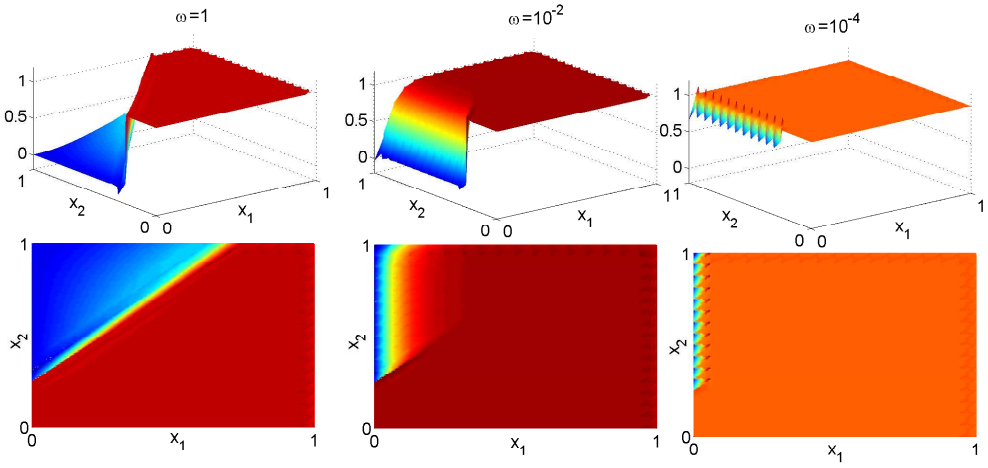


Figure 4.22: Computed state on uniform mesh (289 nodes) with linear elements for $\omega = 1, 10^{-2}, 10^{-4}$ in Example 4.3.4.

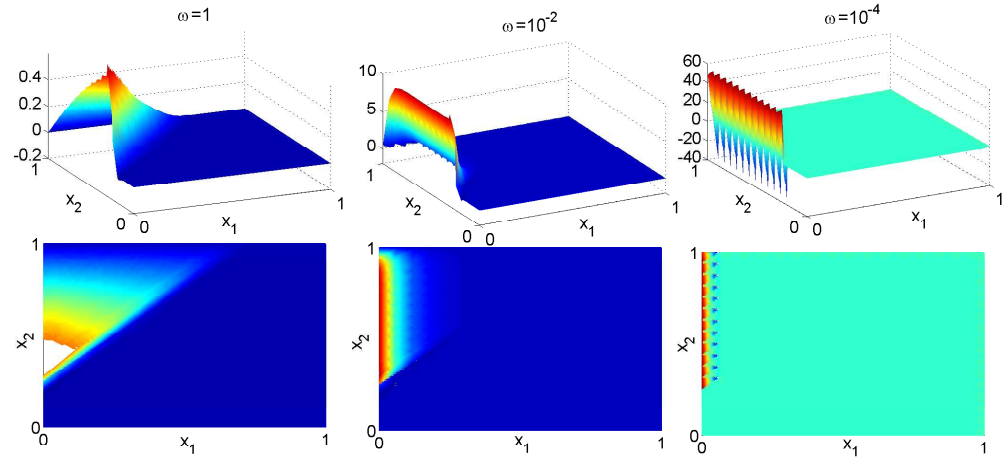


Figure 4.23: Computed control on uniform mesh (289 nodes) with linear elements for $\omega = 1, 10^{-2}, 10^{-4}$ in Example 4.3.4.

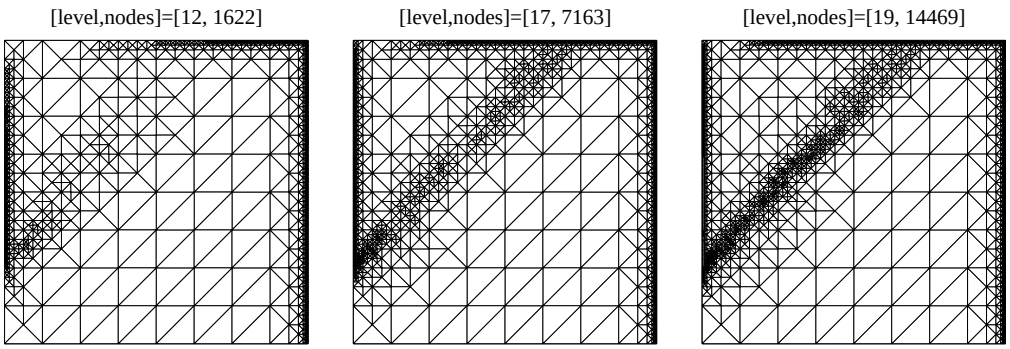


Figure 4.24: Adaptively refined meshes with linear elements at various refinement levels for $\omega = 1$ in Example 4.3.4.

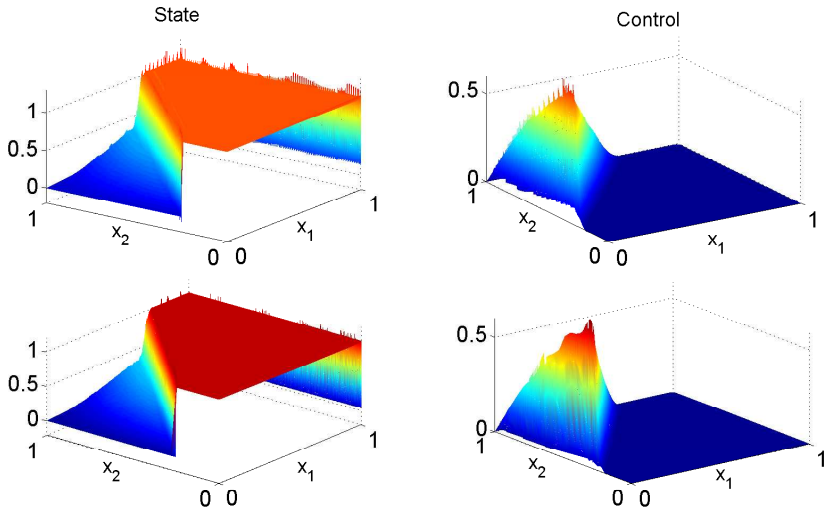


Figure 4.25: Computed solutions of state (left) and control (right) on adaptively refined mesh with linear elements (top row, 14469 nodes) and with quadratic elements (bottom row, 13892 nodes) for $\omega = 1$ in Example 4.3.4.

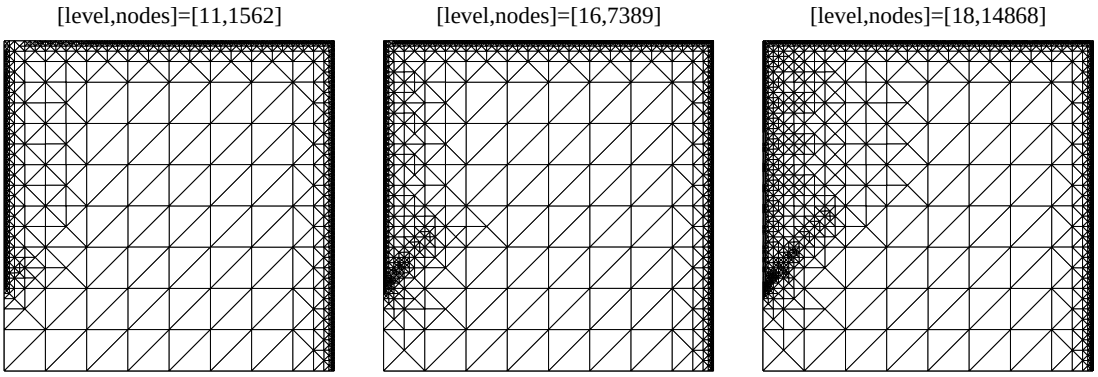


Figure 4.26: Adaptively refined meshes with linear elements at various refinement levels for $\omega = 10^{-2}$ in Example 4.3.4.

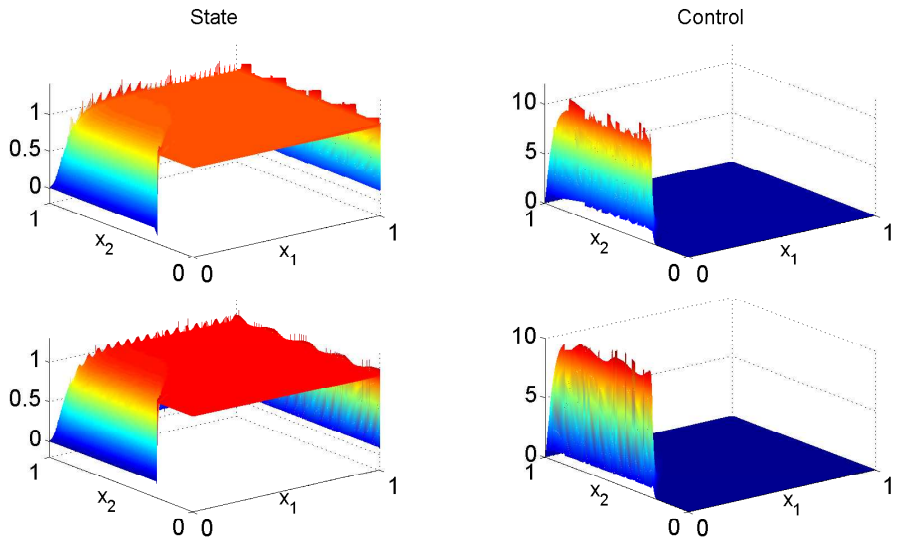


Figure 4.27: Computed solutions of state (left) and control (right) on adaptively refined mesh with linear elements (top row, 14868 nodes) and with quadratic elements (bottom row, 10702 nodes) for $\omega = 10^{-2}$ in Example 4.3.4.

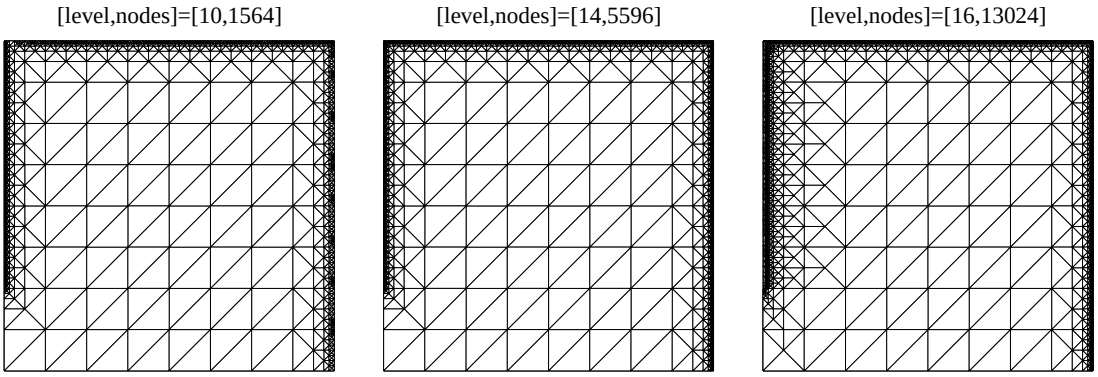


Figure 4.28: Adaptively refined meshes with linear elements at various refinement levels for $\omega = 10^{-4}$ in Example 4.3.4.

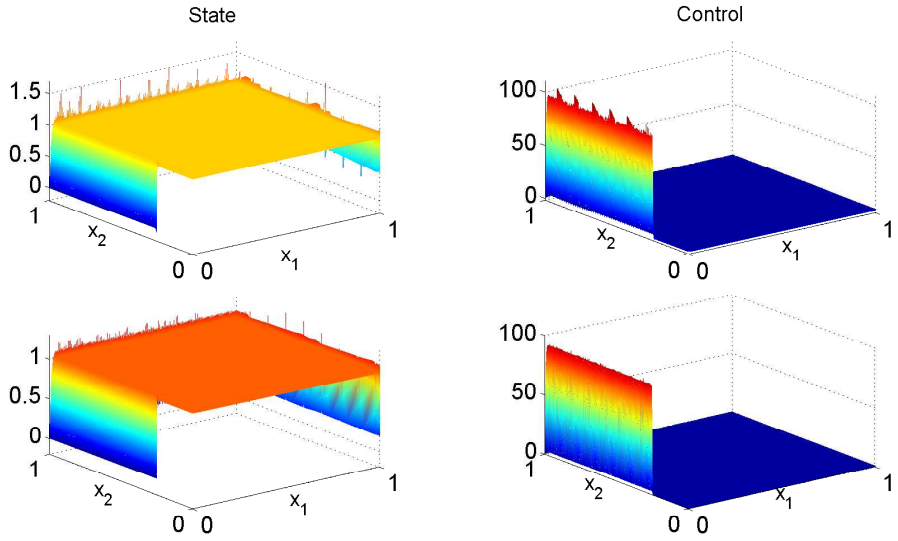


Figure 4.29: Computed solutions of state (left) and control (right) on adaptively refined mesh with linear elements (top row, 13024 nodes) and with quadratic elements (bottom row, 10791 nodes) for $\omega = 10^{-4}$ in Example 4.3.4.

Example 4.3.5 ***Boundary layers with unknown solutions***

The following example with unknown solutions has been studied in [77] and in [18] with control constraints. The problem data are given by

$$\Omega = (0,1)^2, \quad \beta = (-1,-2)^T, \quad r = 1 \ \ and \ \ \omega = 0.1.$$

The Dirichlet boundary condition g_D, the source function f and the desired state y_d are defined by

$$g_D = 0 \ \ on \ \partial\Omega, \quad f = 1 \ \ and \ \ y_d = 1.$$

Since the exact solution of the optimal control problem is not known, we examine the value of the cost function

$$J(y,u) := \frac{1}{2}\int_\Omega (y(x) - y_d(x))^2 dx + \frac{\omega}{2}\int_\Omega u(x)^2 dx.$$

for the various values of the diffusion parameter $\epsilon = 10^{-3}, 10^{-5}, 10^{-7}$. The approximate order of convergence is computed using the following formula:

$$order = \log_2\left(\frac{J(y_h,u_h) - J(y_{2h},u_{2h})}{J(y_{2h},u_{2h}) - J(y_{4h},u_{4h})}\right).$$

h	Nodes	$J(y_h,u_h)$	$J(y_h,u_h) - J(y_{2h},u_{2h})$	order
2.50e-001	25	0.259842439	-	-
1.25e-001	81	0.260011380	1.689408e-004	-
6.25e-002	289	0.259954555	-5.682475e-005	4.80
3.13e-002	1089	0.259892666	-6.188863e-005	0.12
1.56e-002	4225	0.259855658	-3.700797e-005	0.74
7.81e-003	16641	0.259840366	-1.529209e-005	1.28

Table 4.1: Evolution of values of the cost functional $J(y_h,u_h)$ for a sequence of uniformly refined meshes with $\epsilon = 10^{-3}$ in Example 4.3.5.

The evolution of the values of the cost functional $J(y_h,u_h)$ on a sequence of uniformly refined meshes for $\epsilon = 10^{-3}, 10^{-5}, 10^{-7}$ are given in the Figures 4.1, 4.2 and 4.3, respectively.

Boris and Vexler [18] have also solved this example for $\epsilon = 10^{-3}$ with control constraints by local projection stabilization (LPS). We will survey the control constrained case at next Chapter. However, the unconstrained case for $\epsilon = 10^{-5}$ has been studied in [77] by using LPS method. Comparing the results in Table 4.2 and the results obtained in [77], the order of convergence in Table 4.2 is much higher.

h	Nodes	$J(y_h, u_h)$	$J(y_h, u_h) - J(y_{2h}, u_{2h})$	order
2.50e-001	25	0.258736669	-	-
1.25e-001	81	0.258736669	2.970944e-004	-
6.25e-002	289	0.259088388	5.462482e-005	2.44
3.13e-002	1089	0.259098154	9.765670e-006	2.48
1.56e-002	4225	0.259099788	1.634528e-006	2.58
7.81e-003	16641	0.259099982	1.935335e-007	3.08

Table 4.2: Evolution of values of the cost functional $J(y_h, u_h)$ for a sequence of uniformly refined meshes with $\epsilon = 10^{-5}$ in Example 4.3.5.

h	Nodes	$J(y_h, u_h)$	$J(y_h, u_h) - J(y_{2h}, u_{2h})$	order
2.50e-001	25	0.258725441	-	-
1.25e-001	81	0.259023696	2.982547e-004	-
6.25e-002	289	0.259079347	5.565064e-005	2.42
3.13e-002	1089	0.259089772	1.042580e-005	2.42
1.56e-002	4225	0.259091782	2.009823e-006	2.38
7.81e-003	16641	0.259092181	3.985889e-007	3.33

Table 4.3: Evolution of values of the cost functional $J(y_h, u_h)$ for a sequence of uniformly refined meshes with $\epsilon = 10^{-7}$ in Example 4.3.5.

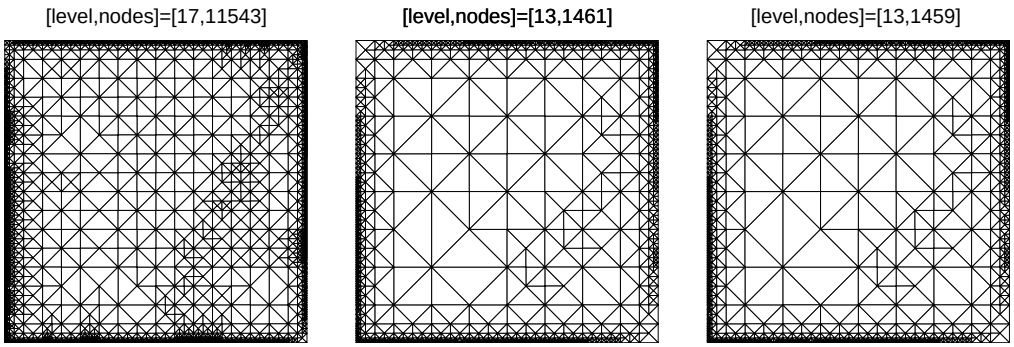


Figure 4.30: Adaptively refined meshes using linear elements for $\epsilon = 10^{-3}, \epsilon = 10^{-5}, \epsilon = 10^{-7}$, respectively, in Example 4.3.5.

The computed solutions on the uniform refined meshes (16641 nodes) are shown in Figure 4.31. We need an adaptive approach to reduce the oscillations in Figure 4.31. Figure 4.30 displays the adaptively refined mesh on a course mesh constructed by beginning a uniform square mesh 4×4 and then dividing each triangle into two triangles. Figure 4.30 shows that the problem exhibits boundary layers. By resolving the boundary layers, the oscillations in Figure 4.32 are reduced. Thus, it is evident that the adaptive meshes save substantial computing time.

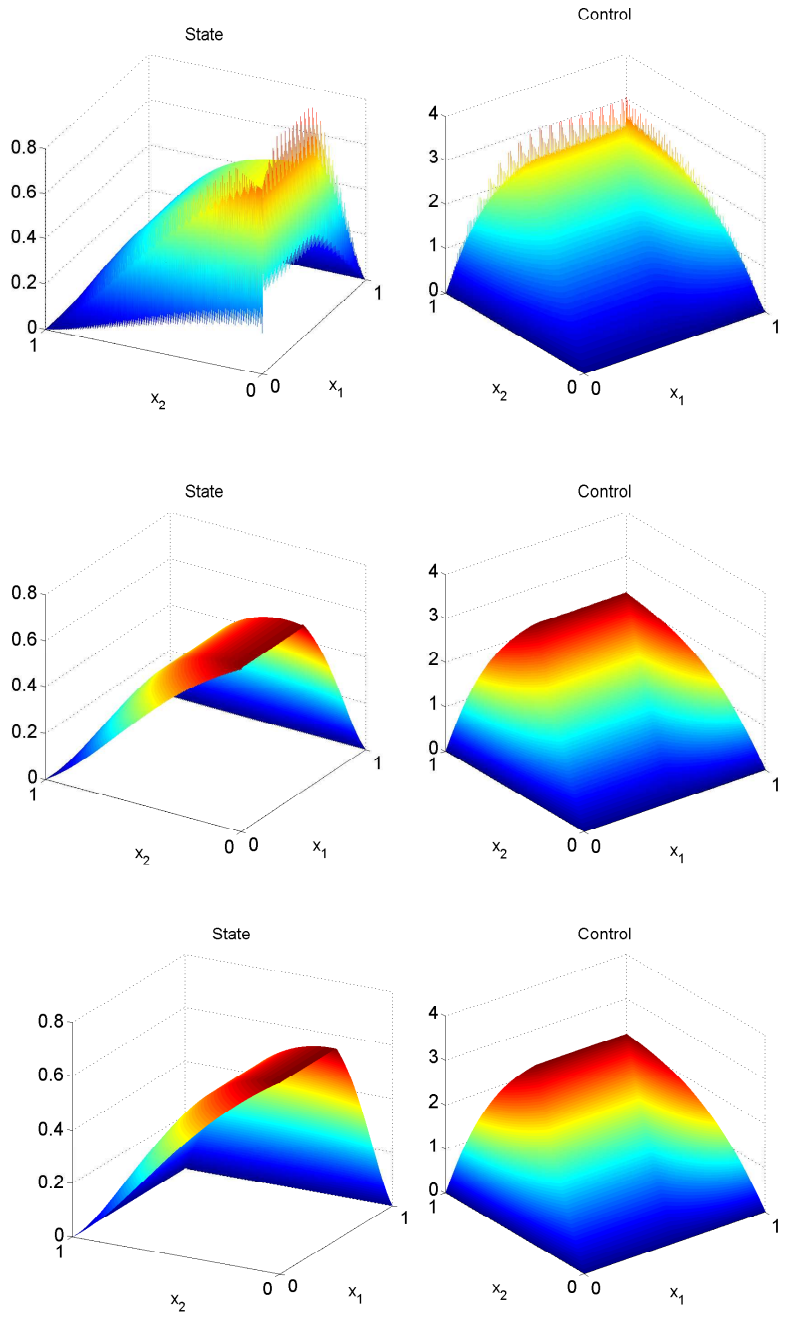


Figure 4.31: The computed solutions of the state (left) and the control (right) on uniformly refined meshes (16641 nodes) for $\epsilon = 10^{-3}$ (top plot), $\epsilon = 10^{-5}$ (middle plot) and $\epsilon = 10^{-7}$ (bottom plot) using linear elements in Example 4.3.5.

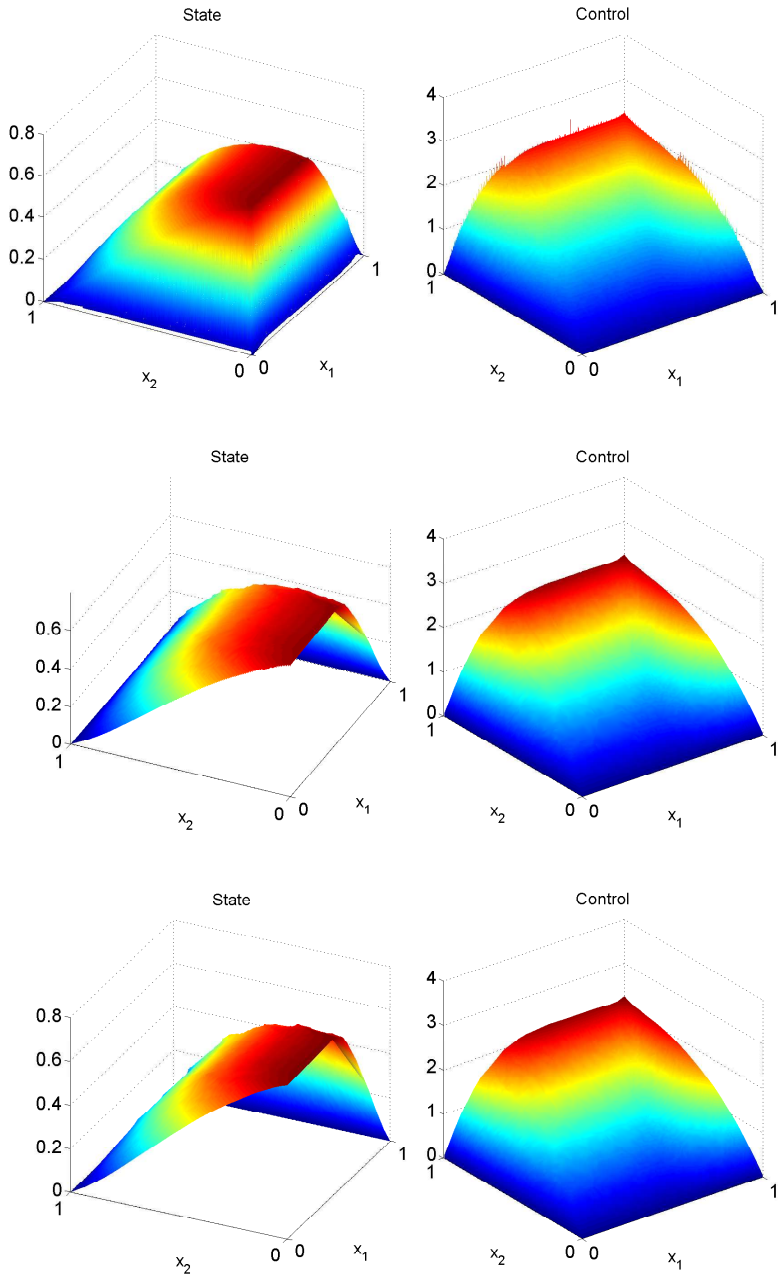


Figure 4.32: The computed solutions of the state (left) and the control (right) on adaptively refined meshes for $\epsilon = 10^{-3}$ (11543 nodes, top plot), $\epsilon = 10^{-5}$ (1461 nodes, middle plot) and $\epsilon = 10^{-7}$ (1459 nodes, bottom plot) using linear elements in Example 4.3.5.

CHAPTER 5

DISTRIBUTED OPTIMAL CONTROL PROBLEMS WITH CONTROL CONSTRAINTS

The control constrained optimal control problems governed by elliptic PDEs have been studied in [4, 28, 29, 52, 56, 79, 90, 96]. In literature, the common concept for solving the control constrained optimal control problem numerically is based on discretization of the set of control u. The authors [4] have proved that the convergence rate of the form $\|u-u_h\|_{L^2} = O(h)$ for piecewise constant control approximations. It has extended to piecewise linear in [90] and the form of convergence rate is $\|u - u_h\|_{L^2} = O(h^{3/2})$ due to the regularity of u. The control u is not regular, where it switches between activity and inactivity. In [52], Hinze has proposed the variational discretization concept for control constrained optimal control problems where the control u is not discretized. Since there is only finite element approximation for control u, the order of the finite element approximation $O(h^2)$ is obtained in [52, 53] for the convergence of control. The other interesting study related to convergence of optimal control problem has been a post-processing in [79] by Meyer and Rösch. In [79], the control u is calculated by projection of the adjoint p_h in a post-processing step after the optimal control problem is solved by a fully discretization. Post-processing step improves the convergence order from $O(h)$ to $O(h^2)$ when piecewise constant functions are used for approximation of control.

For control constrained optimal control problems governed by convection diffusion equations, stabilized finite element methods have been applied in [18, 104]. Stabilization in [18] is based on symmetry penalty terms, where local projections (the so called LPS method), whereas the edge stabilization is used in [104]. Due to the symmetry property of stabilization in [18, 104], the formulating of control problem on the continuous level and then discretizing the optimality conditions are equivalent to discrete control problem. It has been shown in [18, 104] that

the optimal control u only has $O(h^{3/2})$ convergent rate for the piecewise linear discretization of the control space since u is in general not in $H^2(\Omega)$. Additionally, variational dicretization [52] and the edge stabilization Galerkin method have been combined in [54] to solve optimal control problems governed by convection diffusion equations. Recently mixed finite elements [55] and RT mixed FEM/DG [105] have been applied to optimal control problems governed by convection diffusion equations. In [55], the authors have proved a second order convergence results for state y, adjoint p and control u for piecewise linear polynomial approximations provided some assumptions on the regularity and mesh. RT mixed FEM/DG [105] is a combination of Raviart-Thomas mixed finite element methods reducing the second-order equation to a system of the first order equations and discontinuous Galerkin method having good properties for first order hyperbolic equations.

A residual-type a posteriori error estimators in the control constrained case have been derived and analyzed in [50, 67, 74, 75]. In [74] the authors have proposed a posteriori error estimators for an optimal control problem governed by a linear elliptic boundary value problem with convex differentiable objective and box constraints on the control. In contrast to the approach in [74, 75], the error analysis in [50] is related to the error in the state, the adjoint state, the control, and the adjoint control and incorporates in terms of the data of the problem. The goal-oriented dual weighted approach has been applied in [49, 101] for control constrained optimal control problems governed by elliptic PDEs. For control constrained optimal control problems governed by convection diffusion equations, a posteriori error estimate using edge stabilization is given in [54, 104]. In [104] a fully discretization of the state, adjoint and control is used, whereas variational discretization is used in [54]. The error estimator [54] only contains contributions from local residuals in the state and the adjoint equations due to not discretization of control. For discontinuous Galerkin methods, there are a few work to solve optimal control problems [103, 106]. In [103], a posteriori and a priori error estimates have been derived for optimal control problem governed by convection diffusion equations using nonsymmetric interior Galerkin penalty (NIPG) method [88]. Zhou et al. [106] has analyzed the local discontinuous Galerkin (LDG) method for the constrained optimal control problem governed by convection diffusion equations. For the discretization of the control, the authors [106] have discussed two different approaches: variational discretization and full discretization. However, to our knowledge any numerical results has been presented for the solution of control constrained optimal control problems using DG discretization.

In this chapter, we solve control constrained optimal control problems governed by convection diffusion equations using upwind SIPG discretization. We extend the a posteriori error estimator of unconstrained case in previous chapter to control constrained case. The reliability and the efficiency of the error estimator will be analyzed using data approximations errors for the discretized optimal control problem. Lastly, the numerical results are presented to illustrate performance of the adaptive method.

5.1 Prime-Dual Active Set (PDAS) Strategy

In this section, we consider the following constrained optimal control problem governed by convection diffusion equation

$$\min_{u\in U_{ad}\subset U} J(y,u) := \frac{1}{2}\int_\Omega (y(x)-y_d(x))^2dx + \frac{\omega}{2}\int_\Omega (u(x)-u_d(x))^2dx \tag{5.1}$$

subject to

$$\begin{aligned} -\epsilon\Delta y(x) + \beta(x)\cdot\nabla y(x) + r(x)y(x) &= f(x)+u(x), \qquad x\in\Omega, &(5.2a)\\ y(x) &= g_D(x), \qquad x\in\Gamma, &(5.2b)\end{aligned}$$

where $\omega > 0$ is a constant, Ω is a bounded open, convex domain in $\mathbb{R}^2$ and $\Gamma = \partial\Omega$, $U_{ad} \subset U = L^2(\Omega)$ denotes a closed convex set. We use

$$U_{ad} = \{u\in U:\ u_a \le u \le u_b \text{ a.e in } \Omega\}, \tag{5.3}$$

where $u_a < u_b$ denote constants. In addition, the desired control is denoted by $u_d(x) \in L^2(\Omega)$.

Then, the variational formulation corresponding to (5.1)-(5.2) can be rewritten as

$$\min_{u\in U_{ad}} J(y,u) := \frac{1}{2}\|y-y_d\|_\Omega^2 + \frac{\omega}{2}\|u-u_d\|_\Omega^2 \tag{5.4a}$$

$$\begin{aligned} \text{subject to}\quad a(y,\upsilon) + b(u,\upsilon) &= (f,\upsilon), \qquad \forall\upsilon\in V, &(5.4b)\\ (y,u) &\in Y\times U_{ad}.\end{aligned}$$

Due to the convex optimal control problem (5.4) (see [76]), the optimal control problem (5.4) has a unique solution $(y,u)\in Y\times U_{ad}$ provided that the assumptions (3.7) are satisfied. The functions $(y,u)\in Y\times U_{ad}$ solve (5.4) if and only if $(y,u,p)\in Y\times U_{ad}\times Y$ is unique solution

for the following optimality system:

$$\begin{aligned} a(y,\upsilon) + b(u,\upsilon) &= (f,\upsilon), \qquad \forall \upsilon \in V, && (5.5a)\\ a(\psi,p) + (y,\psi) &= (y_d,\psi), \qquad \forall \psi \in V, && (5.5b)\\ (\omega(u-u_d) - p, w-u) &\geq 0, \qquad \forall w \in U_{ad}. && (5.5c) \end{aligned}$$

By introducing the Lagrange multipliers, $\lambda_a, \lambda_b \in L^2(\Omega)$, we obtain the following optimality system of the problem (5.1)-(5.2):

$$\begin{aligned} -\epsilon\Delta y + \beta\cdot\nabla y + ry &= f+u \quad \text{in } \Omega, \qquad y = g_D \ \text{ on } \Gamma,\\ -\epsilon\Delta p - \beta\cdot\nabla p + (r - \nabla\cdot\beta)p &= -(y-y_d) \quad \text{in } \Omega, \qquad p = 0 \quad \text{on } \Gamma,\\ \omega(u-u_d) - p - \lambda_a + \lambda_b &= 0 \qquad \text{a.e in } \Omega, \qquad (5.6)\\ \lambda_a \geq 0, \qquad u_a - u &\leq 0, \quad \lambda_a(u-u_a) = 0 \quad \text{a.e. in } \Omega,\\ \lambda_b \geq 0, \qquad u - u_b &\leq 0, \quad \lambda_b(u_b-u) = 0 \quad \text{a.e. in } \Omega. \end{aligned}$$

We follow Bergounioux, Ito and Kunisch [19], who developed PDAS strategy for control constraints in elliptic control problem. The PDAS strategy has been also studied in [51, 62, 68]. Then, the PDAS method has been interpreted as a semismooth Newton method in [51]. In this section, we will derive the system of optimality condition using the PDAS method as a semismooth Newton method.

With the help of the solution operators, $S : L^2(\Omega) \to L^2(\Omega)$ and $S^* : L^2(\Omega) \to L^2(\Omega)$ for the state and adjoint equation, one obtains $y = Su$ and $p = S^*(y - y_d)$ so that the optimality system can be written in compact form:

$$\begin{aligned} -S^*(Su - y_d) + \omega(u-u_d) - \lambda_a + \lambda_b &= 0 \quad \text{a.e in } \Omega,\\ \lambda_a \geq 0, \qquad u_a - u &\leq 0, \quad \lambda_a(u-u_a) = 0 \quad \text{a.e. in } \Omega, \qquad (5.7)\\ \lambda_b \geq 0, \qquad u - u_b &\leq 0, \quad \lambda_b(u_b-u) = 0 \quad \text{a.e. in } \Omega. \end{aligned}$$

The system (5.7) containing inequalities can be solved by combining PDAS strategy with Newton method as in [51, 78].

Definition 5.1.1 *A function $\Psi : \mathbb{R}^2 \to \mathbb{R}$ with the property*

$$\Psi(a,b) = 0 \iff a \leq 0, \quad b \leq 0, \quad ab = 0$$

is called complementary function.

By using the complimentary function $\Psi(a,b) = \max\{a, cb\}$ with $\lambda = \lambda_b - \lambda_a$, we obtain

$$\begin{aligned} -S^*(Su - y_d) + \omega(u - u_d) + \lambda &= 0 \quad \text{a.e. in } \Omega, \\ \lambda - \min\{0, \lambda - c(u_a - u)\} - \max\{0, \lambda + c(u - u_b)\} &= 0 \quad \text{a.e. in } \Omega. \end{aligned} \tag{5.8}$$

We use Newton-differentiability [78] to solve this nonlinear system of equations for $(u, \lambda) \in L^2(\Omega) \times L^2(\Omega)$.

Definition 5.1.2 *[78, Definition 16.2] (**Newton-differentiability**)*

$F : U \to V$ is called Newton-differentiable at the point $u \in U$ if an open neighborhood U_0 of u and $G : U_0 \to L(U, V)$ exists, such that:

$$\frac{\|F(u+h) - F(u) - G(u+h)h\|_V}{\|h\|_U} \to 0 \quad \text{as } \|h\| \to 0.$$

G is called a generalized derivation or Newton derivation of F.

Newton's derivation $G(u)$ is not pointwise derivative concept as the Frèchet-differentiability. G is a whole family of linear operators in the vicinity of u. With the help of the Newton-derivative, a generalized Newton method for solving F (u) = 0 can be formulated:

$$u_{n+1} = u_n - G^{-1}(u_n)F(u_n) \quad \Longrightarrow \quad G(u_n)u_{n+1} = G(u_n)u_n - F(u_n). \tag{5.9}$$

Let us write the system (5.8) together, then we obtain

$$\begin{aligned} F(u) := -S^*(Su - y_d) + \omega(u - u_d) + \min\{0, S^*(Su - y_d) - \omega(u - u_d) - c(u_a - u)\} \\ + \max\{0, S^*(Su - y_d) - \omega(u - u_d) - c(u - u_b)\} = 0. \end{aligned} \tag{5.10}$$

Let us choose $c = \omega$,

$$\begin{aligned} F(u) := -S^*(Su - y_d) + \omega(u - u_d) \quad &+ \quad \min\{0, S^*(Su - y_d) + \omega(u_d - u_a)\} \\ &+ \quad \max\{0, S^*(Su - y_d) + \omega(u_d - u_b)\} = 0. \end{aligned} \tag{5.11}$$

Then, the Newton derivative of $F(u)$ is

$$\begin{aligned} G(u)h &= -S^*Sh + \omega h + (\chi_{A^-(u)} + \chi_{A^+(u)})S^*Sh \\ &= -\chi_{I(u)}S^*Sh + \omega h, \end{aligned} \tag{5.12}$$

with active sets

$$A^-(u) = \{x \in \Omega : S^*(Su - y_d) + \omega(u_d - u_a) < 0\}, \tag{5.13}$$

$$A^+(u) = \{x \in \Omega : S^*(Su - y_d) + \omega(u_d - u_b) > 0\} \tag{5.14}$$

and the inactive set $I(u) = \Omega\backslash(A^+(u) \cup A^-(u))$. By Newton method in (5.9),

$$G(u_n)u_{n+1} = G(u_n)u_n - F(u_n)$$

$$\Longleftrightarrow$$

$$\begin{aligned} -\chi_{I(u)}S^*Su_{n+1} + \omega u_{n+1} &= -\chi_{I(u)}S^*Su_n + \omega u_n - \omega u_n + \omega u_d + S^*Su_n - S^*y_d \\ &- \min\{0, S^*(Su - y_d) + \omega(u_d - u_a)\} - \max\{0, S^*(Su - y_d) + \omega(u_d - u_b)\}. \end{aligned} \tag{5.15}$$

By the definitions in (5.13) and (5.14),

$$\begin{aligned} \min\{0, S^*(Su_n - y_d) + \omega(u_d - u_a)\} &= \chi_{A_n^-}(S^*(Su_n - y_d) + \omega(u_d - u_a)), \\ \max\{0, S^*(Su_n - y_d) + \omega(u_d - u_b)\} &= \chi_{A_n^+}(S^*(Su_n - y_d) + \omega(u_d - u_b)). \end{aligned}$$

For the right side of (5.15) one obtains

$$\begin{aligned} (\chi_{A_n^-} + \chi_{A_n^+})S^*(Su_n - y_d) &- \chi_{I_n}S^*y_d + \omega u_d \\ &- \chi_{A_n^-}(S^*(Su_n - y_d) + \omega(u_d - u_a)) - \chi_{A_n^+}(S^*(Su_n - y_d) + \omega(u_d - u_b)) \\ &= \chi_{A_n^-}\omega u_a + \chi_{A_n^+}\omega u_b - \chi_{I_n}S^*y_d + \chi_{I_n}u_d. \end{aligned}$$

Then, (5.15) equals to

$$\omega u_{n+1} - \chi_{I_n}S^*(Su_{n+1} - y_d) = \chi_{A_n^-}\omega u_a + \chi_{A_n^+}\omega u_b + \chi_{I_n}u_d. \tag{5.16}$$

Hence, the optimality system (5.6) is equivalent to

$$\begin{aligned} -\epsilon\Delta y_{n+1} + \beta\cdot\nabla y_{n+1} + ry_{n+1} &= f + u_{n+1} \quad \text{in } \Omega, \quad y = g_D \text{ on } \Gamma, \\ -\epsilon\Delta p_{n+1} - \beta\cdot\nabla p_{n+1} + (r - \nabla\cdot\beta)p_{n+1} &= -(y_{n+1} - y_d) \text{ in } \Omega, \quad p = 0 \text{ on } \Gamma, \\ \omega u_{n+1} - \chi_{I_n}S^*(Su_{n+1} - y_d) &= \chi_{A_n^-}\omega u_a + \chi_{A_n^+}\omega u_b + \chi_{I_n}u_d. \end{aligned} \tag{5.17}$$

Then, we can write the DG approximation of the optimal control problem (5.4) as follows:

$$\min_{u_h \in U_h^{ad}} J(y_h, u_h) := \frac{1}{2}\|y_h - y_d\|_\Omega^2 + \frac{\omega}{2}\|u_h - u_d\|_\Omega^2 \tag{5.18a}$$

$$\begin{aligned} \text{subject to} \quad a(y_h, v_h) \;+\; b(u_h, v_h) &= (f, v_h), \qquad \forall v_h \in V_h, \\ (y_h, u_h) &\in Y_h \times U_h^{ad}, \end{aligned} \tag{5.18b}$$

with the spaces

$$\begin{aligned} V_h = Y_h &= \{y_h \in L^2(\Omega) \mid \; y|_E \in P_n(E), \;\; \forall E \in \xi_h\}, \\ U_h^{ad} &= \{u_h \in L^2(\Omega) \mid \; u|_E \in P_m(E), \;\; \forall E \in \xi_h\}. \end{aligned}$$

The DG dicretized optimal control problem (5.18) has a unique solution $(y_h, u_h) \in Y_h \times U_h^{ad}$. The functions $(y_h, u_h) \in Y_h \times U_h^{ad}$ solve (5.18) if and only if $(y_h, u_h, p_h) \in Y_h \times U_h^{ad} \times Y_h$ is unique solution for the following optimality system:

$$a(y_h, v_h) + b(u_h, v_h) = (f, v_h), \qquad \forall v_h \in V_h, \tag{5.19a}$$

$$a(\psi_h, p_h) + (y_h, \psi_h) = (y_d, \psi_h), \qquad \forall \psi_h \in V_h, \tag{5.19b}$$

$$(\omega(u_h - (u_d)_h) - p_h, w_h - u_h) \geq 0, \qquad \forall w_h \in U_h^{ad}. \tag{5.19c}$$

Applying upwind SIPG discretization to the optimality system (5.17), we obtain the following system

$$\begin{aligned} \mathbb{M}\vec{y} + \mathbb{A}_a\vec{p} &= \vec{b}, \\ \mathbb{A}_s\vec{y} + \mathbb{B}\vec{u} &= \vec{f}, \\ \omega\mathbb{Q}\vec{u} + \text{diag}(\chi_I)\mathbb{B}\vec{p} &= \omega\mathbb{Q}(\chi_{A^-}u_a + \chi_{A^+u_b} + \chi_I u_d), \end{aligned}$$

where $\mathbb{A}_s, \mathbb{A}_a, \mathbb{M}, \mathbb{Q}, \mathbb{B} \in \mathbb{R}^{(N_{loc}\times N)\times(N_{loc}\times N)}$ and $\vec{b}, \vec{f} \in \mathbb{R}^{N_{loc}\times N}$, see Section 3.2 and 3.3 for details.

Then, the optimality system is written such as:

$$\begin{pmatrix} \mathbb{M} & 0 & \mathbb{A}_a \\ 0 & \omega\mathbb{Q} & \text{diag}(\chi_I)\mathbb{B} \\ \mathbb{A}_s & \mathbb{B} & 0 \end{pmatrix} \begin{pmatrix} \vec{y} \\ \vec{u} \\ \vec{p} \end{pmatrix} = \begin{pmatrix} \vec{b} \\ \omega\mathbb{Q}(\chi_{A^-}u_a + \chi_{A^+}u_b + \chi_I u_d) \\ \vec{f} \end{pmatrix}.$$

5.2 A Posteriori Error Analysis

Different posteriori error estimates were used for convection dominated control constrained optimal control problems. In [104], the residual-type a posteriori error estimates have been

presented using the edge stabilization Galerkin method to solve unilateral control constrained optimal control problems governed by convection diffusion equations. The estimators of the state and adjoint variables are similar to in [74] for elliptic problems, except that they contains an extra term coming from edge stabilization term in weak formulation of state equation (or adjoint). However, the residual of variational inequality in optimality system is computed on non-contact set without approximation of integral averaging operator as in [74]. In [54], residual type a posteriori error estimators contain only contributions from the residuals of state and adjoint equations, because the control is not discretized. The contributions coming from the state and adjoint equation is the same as in [104] due to the same discretization on the state and adjoint equations. In [103], a posteriori error estimates are obtained convection-diffusion equations for the nonsymmetric interior penalty Galerkin (NIPG) method using the technique in [74].

Here, we use the residual-type a posteriori error estimators for the state and the adjoint, in [99] for standard finite element and in [91] for the SIPG method. They consist of the element residuals such as in [74] and of the edge residuals similar to those in [54, 104]. For the SIPG method, they contain extra terms measuring the jumps of approximate solutions. We use the a posteriori error estimator introduced in [74] for the variational inequality arising from the control constraints.

The error estimator consists of three parts; state, adjoint and variational inequality in (5.5). Since the error estimators of state and adjoint are as given in (4.1), our purpose is to introduce an estimator contributed from the approximation error of the variational inequality using the DG discretization. We consider first the unilateral control constraints

$$U_{ad} = \{u \in U : \ u \geq u_a \ \text{ a.e. in } \Omega\},$$

and, then, we will extend it to bilateral control constraints.

5.2.1 Unilateral Control Constraint

We divide the domain as in [74]

$$\Omega^- = \{x \in \Omega \ : u(x) = u_a\}, \qquad \Omega^+ = \{x \in \Omega \ : u(x) > u_a\},$$

$$\Omega_h^- = \{\cup \bar{E} \ : E \subset \Omega^-\}, \qquad \Omega_h^+ = \{\cup \bar{E} \ : E \subset \Omega^+\},$$

$$\Omega_h^\alpha = \Omega \backslash (\Omega_h^- \cup \Omega_h^+), \quad \Omega_h^{+\alpha} = \Omega_h^\alpha \cup \Omega_h^+, \quad \Omega_h^{-\alpha} = \Omega_h^\alpha \cup \Omega_h^-.$$

It can be seen that the inequality in (5.5c) is equivalent to the following:

$$\omega(u - u_d) - p \geq 0, \quad u \geq u_a, \quad (\omega(u - u_d) - p)(u - u_a) = 0, \quad \text{a.e. in } \Omega. \tag{5.20}$$

Lemma 5.2.1 *[74, Lemma 3.4] Let $\pi_h : L^1(K) \to K_h$ be the integral averaging defined in such that $\pi_h u = \frac{1}{|E|}\int_E u dx$, then for m=0, 1 and $1 \leq q \leq \infty$,*

$$\|\upsilon - \pi_h \upsilon\|_{0,q,E} \leq C h_E^m |\upsilon|_{m,q,E}, \qquad \forall \upsilon \in W^{m,q}(K). \tag{5.21}$$

We follow here [74] to establish a connection between the control and the adjoint variables in order to find the a posteriori error bounds. The a posteriori estimators in [74] for the control constrained optimal control problem are derived from the residual of the state and adjoint as a single partial differential equation [3]. For the variational inequality, the residual is computed on the region except for the non-coincidence set (non-contact set) to enable sharp error estimates.

Remark 5.2.2 *As unconstrained case in Chapter 4, the proof of reliability and efficiency of our error indicator for control constrained optimal control problem is valid provided that $r_0 \geq 0$.*

Lemma 5.2.3 *Let (y, u, p) and (y_h, u_h, p_h) be the solutions of (5.5) and (5.19), respectively. Then*

$$\|u - u_h\|^2_{L^2(\Omega)} \lesssim (\eta_1^u)^2 + (\theta^u)^2 + \|p_h - p[u_h]\|^2_{L^2(\Omega)}, \tag{5.22}$$

where $p[u_h]$ satisfies the following equation:

$$a(y[u_h], w) - (u_h, w) = l(w), \qquad \forall w \in V, \tag{5.23a}$$

$$a(w, p[u_h]) + (y[u_h], w) = (y_d, w), \quad \forall w \in V, \tag{5.23b}$$

and

$$\eta_1^u = \sum_{E \in \xi_h} h_E \|\nabla(\omega(u_h - (u_d)_h) - p_h)\chi_{\Omega_h^{+\alpha}}\|_{L^2(\Omega)}. \tag{5.24}$$

$$\theta^u = \omega\|u_d - (u_d)_h\|_{L^2(\Omega)}. \tag{5.25}$$

Proof. It follows from the inequalities (5.5c) and (5.19c) that, for any $\upsilon_h \in U_h^{ad}$,

$$\begin{aligned}
\omega\|u-u_h\|_{L^2}^2 &= (\omega(u-u_d),u-u_h)-(\omega(u_h-(u_d)_h),u-u_h)+(\omega(u_d-(u_d)_h),u-u_h),\\
&\le (p,u-u_h)-(\omega(u_h-(u_d)_h),u-u_h)+(\omega(u_h-(u_d)_h)-p_h,\upsilon_h-u_h)\\
&+(\omega(u_d-(u_d)_h),u-u_h),\\
&=(p-p_h,u-u_h)+(\omega(u_h-(u_d)_h)-p_h,\upsilon_h-u)+(\omega(u_d-(u_d)_h),u-u_h).
\end{aligned} \tag{5.26}$$

The equations (5.23a) and (5.23b) yield

$$\begin{aligned}
(u-u_h,p[u_h]-p) &= (u,p[u_h]-p)-(u_h,p[u_h]-p),\\
&= a(y,p[u_h]-p)-l(p[u_h]-p)-a(y[u_h],p[u_h]-p)+l(p[u_h]-p),\\
&= a(y-y[u_h],p[u_h]-p),\\
&= a(y-y[u_h],p[u_h])-a(y-y[u_h],p),\\
&= (y_d,y-y[u_h])-(y[u_h],y-y[u_h])-(y_d,y-y[u_h])+(y,y-y[u_h]),\\
&= (y-y[u_h],y-y[u_h])=\|y-y[u_h]\|^2\ge 0.
\end{aligned} \tag{5.27}$$

Then, (5.27) gives us

$$\begin{aligned}
(p-p_h,u-u_h) &\le (p-p[u_h],u-u_h)+(p[u_h]-p_h,u-u_h),\\
&\le (p[u_h]-p_h,u-u_h),\\
&\lesssim \|p[u_h]-p_h\|_{L^2(\Omega)}^2+\|u-u_h\|_{L^2(\Omega)}^2.
\end{aligned} \tag{5.28}$$

By using the expressions in (5.20) we obtain

$$(\omega(u_h-(u_d)_h)-p_h,\upsilon_h-u)\le(\omega(u_h-(u_d)_h-p_h,\upsilon_h-u)_{\Omega_h^{+\alpha}}$$

Let us take $\upsilon_h=\pi_h u$ defined in Lemma 5.2.1. Then, we have

$$\begin{aligned}
(\omega(u_h-(u_d)_h)-p_h,\upsilon_h-u)_{\Omega_h^{+\alpha}} &= ((I-\pi_h)(\omega(u_h-(u_d)_h)-p_h),(\pi_h-I)(u-u_h))_{\Omega_h^{+\alpha}},\\
&\le Ch_K\|\nabla(\omega(u_h-(u_d)_h)-p_h)\|_{L^2(\Omega_h^{+\alpha})}\|u-u_h\|_{L^2(\Omega_h^{+\alpha})},\\
&\le Ch_K^2\|\nabla(\omega(u_h-(u_d)_h)-p_h)\|_{L^2(\Omega_h^{+\alpha})}^2+\frac{C}{4}\|u-u_h\|_{L^2(\Omega_h^{+\alpha})}^2,\\
&\le Ch_K^2\|\nabla(\omega(u_h-(u_d)_h)-p_h)\|_{L^2(\Omega_h^{+\alpha})}^2+\frac{C}{4}\|u-u_h\|_{L^2(\Omega)}^2.
\end{aligned} \tag{5.29}$$

Finally, we give a bound for the third term of (5.26)

$$(\omega(u_d - (u_d)_h), u - u_h) \lesssim \omega\|u_d - (u_d)_h\|^2 + \|u - u_h\|^2. \tag{5.30}$$

Combining the inequalities (5.28-5.30) above, we obtain the desired inequality. ■

From Lemma 4.2.9 and Lemma 4.2.10, we have

$$\begin{aligned} |||p[u_h] - p_h||| + |p[u_h] - p_h|_A &\lesssim \eta^p + \theta^p + \|y_h - y[u_h]\|_{L^2(\Omega)}, \\ |||y[u_h] - y_h||| + |y[u_h] - y_h|_A &\lesssim \eta^y + \theta^y. \end{aligned}$$

Theorem 5.2.4 *Let (y, u, p) and (y_h, u_h, p_h) be the solutions of (5.5) and (5.19), respectively. Let the error estimators η^y, η^p and η_1^u be defined by (4.1) and (5.25), respectively, and the data approximation errors θ^y, θ^p by (4.2) and θ^u by (5.25). Then, we have the a posteriori error bound*

$$\|u - u_h\|_{L^2(\Omega)} + |||y - y_h||| + |y - y_h|_A + |||p - p_h||| + |p - p_h|_A \lesssim \eta_1^u + \theta^u + \eta^y + \theta^y + \eta^p + \theta^p. \tag{5.31}$$

Proof. The proof is the same as that in Theorem 4.2.11. ■

Now, we wish to demonstrate the efficiency of the error estimator by establishing lower error bound for the DG approximation.

Theorem 5.2.5 *Let (y, u, p) and (y_h, u_h, p_h) be the solutions of (5.5) and (5.19), respectively. Let the error estimators η^y, η^p and η_1^u be defined by (4.1) and (5.25), respectively, and the data approximation errors θ^y, θ^p by (4.2) and θ^u by (5.25). Then, we have*

$$\begin{aligned} \eta_1^u + \eta^y + \eta^p &\lesssim \|u - u_h\|_{L^2(\Omega)} + |||y - y_h||| + |y - y_h|_A + |||p - p_h||| + |p - p_h|_A \\ &+ \theta^y + \theta^p + \theta^u + \sum_{E \in \xi_h} h_E^2 \|\nabla(\omega(u_h - (u_d)_h) - p_h)\chi_{\Omega_h^\alpha}\|_{L^2}^2. \end{aligned} \tag{5.32}$$

Proof. We deduce that $(\omega(u-u_d)-p)|_{\Omega^+}=0$ from the optimality conditions in (5.5). It follows from the inverse property that

$$\begin{aligned}
(\eta^u)^2 &= \sum_{K\in\mathcal{T}_h} h_K\|\nabla(\omega(u_h-(u_d)_h)-p_h)\chi_{\Omega_h^{+\alpha}}\|_{L^2(\Omega)},\\
&= \sum_{K\in\mathcal{T}_h} h_K\|\nabla(\omega(u_h-(u_d)_h)-p_h)\chi_{\Omega_h^{+}}\|_{L^2(\Omega)} + \sum_{K\in\mathcal{T}_h} h_K\|\nabla(\omega(u_h-(u_d)_h)-p_h)\chi_{\Omega_h^{\alpha}}\|_{L^2(\Omega)},\\
&\lesssim \|\omega(u_h-(u_d)_h)-p_h-\omega(u-u_d)+p\|^2_{L^2(\Omega_h^+)} + \sum_{K\in\mathcal{T}_h} h_K\|\nabla(\omega(u_h-(u_d)_h)-p_h)\chi_{\Omega_h^{\alpha}}\|_{L^2(\Omega)},\\
&\lesssim \omega\|u-u_h\|_{L^2(\Omega)} + \omega\|u_d-(u_d)_h\|_{L^2(\Omega)} + \|p-p_h\|_{L^2(\Omega)} + \sum_{K\in\mathcal{T}_h} h_K\|\nabla(\omega(u_h-(u_d)_h)-p_h)\chi_{\Omega_h^{\alpha}}\|_{L^2(\Omega)}.
\end{aligned}$$

We use Lemma 4.2.13 and Lemma 4.2.14, respectively, to bound η^y and η^p :

$$\begin{aligned}
\eta^y &\lesssim |||y-y_h||| + |y-y_h|_A + \theta^y + \|u-u_h\|_{L^2(\Omega)},\\
\eta^p &\lesssim |||p-p_h||| + |p-p_h|_A + \theta^p + \|y-y_h\|_{L^2(\Omega)}.
\end{aligned}$$

Thus, we have proved the desirable result. ■

Because the position of the free boundary is not known, the characteristic function $\chi_{\Omega_h^{+\alpha}}$ is not a posteriori. It was approximated in [74] by the finite element solution with the a posteriori quantity $\chi^h_{\Omega_h^{+\alpha}}$ for $\alpha>0$

$$\chi^h_{\Omega_h^{+\alpha}} = \frac{u_h-u_a}{h^\alpha+u_h-u_a}.$$

5.2.2 Bilateral Control Constraints

We now consider the control problem (5.1)-(5.2) with bilateral constrained case: $u_a<u_b$. Let

$$\begin{aligned}
U_{ad} &= \{v\in U:\ u_a\le v\le u_b\},\\
U_h^{ad} &= \{v_h\in U_h:\ u_a^h\le v_h\le u_b^h\},
\end{aligned}$$

where $u_a^h, u_b^h\in U_h$ are approximations of u_a, u_b, respectively. To generalize the ideas used in Lemma 5.2.3 to this case, we define for $i=a,b$, as in [74]

$$\Omega^-_{u_i}=\{x\in\Omega\ : u(x)=u_i\},\qquad \Omega^-_u=\Omega^-_{u_a}\cup\Omega^-_{u_b},\qquad \Omega^+_u=\Omega\backslash\Omega^-_u,$$

$$\Omega^-_{u_i,h}=\{\cup\bar{E}:\ E\subset\Omega^-_{u_i},\ E\in\xi_h\},\qquad \Omega^-_{u,h}=\Omega^-_{u_a,h}\cup\Omega^-_{u_b,h},\qquad \Omega^{+\alpha}_{u,h}=\Omega\backslash\Omega^-_{u,h},$$

$$\Omega^{-\alpha}_{u_i,h}=\{\cup\bar{E}:\ \bar{E}\cap\Omega^-_{u_i,h}\neq\emptyset,\ E\in\xi_h\}.$$

Theorem 5.2.6 *Let (y, u, p) and (y_h, u_h, p_h) be the solutions of (5.5) and (5.19), respectively. Let the error estimators η^y and η^p be defined by (4.1) and the data approximation errors θ^y, θ^p by (4.2) and θ^u by (5.25). Then, we have the a-posteriori error bound*

$$\|u-u_h\|_{L^2(\Omega)} + |||y-y_h||| + |y-y_h|_A + |||p-p_h||| + |p-p_h|_A \lesssim \eta^u + \theta^u + \eta^y + \theta^y + \eta^p + \theta^p, \quad (5.33)$$

where

$$\eta^u = \sum_{E\in\xi_h} h_E \|\nabla(\omega(u_h - (u_d)_h) - p_h)\chi_{\Omega^{+\alpha}_{u,h}}\|_{L^2(\Omega)}. \quad (5.34)$$

Proof. In this case, we have

$$(\omega(u-u_d)-p)\chi_{\Omega^-_{u_a}} \geq 0, \qquad (\omega(u-u_d)-p)\chi_{\Omega^-_{u_b}} \leq 0, \qquad (\omega(u-u_d)-p)\chi_{\Omega^+_u} = 0. \quad (5.35)$$

It follows from the inequalities (5.5c) and (5.19c) that, for any $\upsilon_h \in U_h^{ad}$,

$$\begin{aligned}
\omega\|u-u_h\|^2_{L^2} &= (\omega(u-u_d), u-u_h) - (\omega(u_h-(u_d)_h), u-u_h) + (\omega(u_d-(u_d)_h), u-u_h),\\
&\leq (p, u-u_h) - (\omega(u_h-(u_d)_h), u-u_h) + (\omega(u_h-(u_d)_h) - p_h, \upsilon_h - u_h)\\
&+ (\omega(u_d-(u_d)_h), u-u_h),\\
&= (p-p_h, u-u_h) + (\omega(u_h-(u_d)_h) - p_h, \upsilon_h - u) + (\omega(u_d-(u_d)_h), u-u_h).
\end{aligned} \quad (5.36)$$

From Lemma 5.2.3, we have

$$(u-u_h, p[u_h]-p) = (y-y[u_h], y-y[u_h]) = \|y-y[u_h]\|^2 \geq 0.$$

$$\begin{aligned}
(p-p_h, u-u_h) &\leq (p-p[u_h], u-u_h) + (p[u_h]-p_h, u-u_h),\\
&\leq (p[u_h]-p_h, u-u_h),\\
&\lesssim \|p[u_h]-p_h\|^2_{L^2(\Omega)} + \|u-u_h\|^2_{L^2(\Omega)}.
\end{aligned} \quad (5.37)$$

By using the expressions (5.35), we obtain

$$(\omega(u_h-(u_d)_h) - p_h, \upsilon_h - u) \leq (\omega(u_h-(u_d)_h - p_h, \upsilon_h - u)_{\Omega^{+\alpha}_{u,h}}$$

Let us take $\upsilon_h = \pi_h u$ defined in Lemma 5.2.1. Then, we have

$$\begin{aligned}
(\omega(u_h-(u_d)_h) - p_h, \upsilon_h - u)_{\Omega^{+\alpha}_{u,h}} &= ((I-\pi_h)(\omega(u_h-(u_d)_h) - p_h), (\pi_h - I)(u-u_h))_{\Omega^{+\alpha}_{u,h}},\\
&\leq Ch_K \|\nabla(\omega(u_h-(u_d)_h) - p_h)\|_{L^2(\Omega^{+\alpha}_{u,h})} \|u-u_h\|_{L^2(\Omega^{+\alpha}_{u,h})},\\
&\leq Ch_K^2 \|\nabla(\omega(u_h-(u_d)_h) - p_h)\|^2_{L^2(\Omega^{+\alpha}_{u,h})} + \frac{C}{4}\|u-u_h\|^2_{L^2(\Omega^{+\alpha}_{u,h})},\\
&\leq Ch_K^2 \|\nabla(\omega(u_h-(u_d)_h) - p_h)\|^2_{L^2(\Omega^{+\alpha}_{u,h})} + \frac{C}{4}\|u-u_h\|^2_{L^2(\Omega)}.
\end{aligned} \quad (5.38)$$

The rest of the proof is the same as that of Theorem 5.2.4. ■

Theorem 5.2.5 can also be generalized to the bilateral constrained case in the same way. Thus, we obtain the following result.

Theorem 5.2.7 *Let (y, u, p) and (y_h, u_h, p_h) be the solutions of (5.5) and (5.19), respectively. Let the error estimators η^y, η^p and η^u be defined by (4.1) and (5.34), respectively, and the data approximation errors θ^y, θ^p by (4.2)and θ^u by (5.25). Then we have*

$$\begin{aligned}\eta^u + \eta^y + \eta^p &\lesssim \|u - u_h\|_{L^2(\Omega)} + |||y - y_h||| + |y - y_h|_A + |||p - p_h||| + |p - p_h|_A \\ &+ \theta^y + \theta^p + \theta^u + \sum_{E\in\xi_h} h_E^2 \|\nabla(\omega(u_h - (u_d)_h) - p_h)\chi_{\Omega^\alpha_{u,h}}\|^2_{L^2(\Omega)}. \end{aligned} \tag{5.39}$$

The characteristic function $\chi_{\Omega^{+\alpha}_{u,h}}$ for the bilateral control constraints is approximated in [74] similar to the unilateral case. For $\alpha > 0$, we have

$$\chi^h_{\Omega^{+\alpha}_{u,h}} = \frac{(u_h - u_a^h)(u_b^h - u_h)}{h^\alpha + (u_h - u_a^h)(u_b^h - u_h)}.$$

5.3 Numerical Results

In this section, we give several numerical results for optimal control problems governed by convection diffusion equation. When the analytical solutions of the state and the adjoint are given, the Dirichlet boundary condition g_D, the source function f, the desired state y_d and the desired control u_d are computed from (5.6) using the exact state, adjoint and control. The linear discontinuous finite elements are used with the basis functions $(x, y, 1 - x - y)$.

Example 5.3.1 *This example has been studied in optimal control setting by Hinze, Yan and Zhou [54]. The problem data are given by*

$$\Omega = [0, 1]^2, \quad \epsilon = 10^{-3}, \quad \beta = (2, 3)^T, \quad r = 2, \quad \omega = 0.1 \ \text{ and } \ u_d = 0.$$

The admissible set $U_{ad} = \{\upsilon \in U : \ \upsilon \geq 0\}$.

The true state, adjoint and control defined by

$$\begin{aligned} y(x_1, x_2) &= 100(1 - x_1)^2 x_1^2 x_2(1 - 2x_2)(1 - x_2), \\ p(x_1, x_2) &= 50(1 - x_1)^2 x_1^2 x_2(1 - 2x_2)(1 - x_2), \\ u(x_1, x_2) &= \max\{0, -\frac{1}{\omega} p(x_1, x_2)\}. \end{aligned}$$

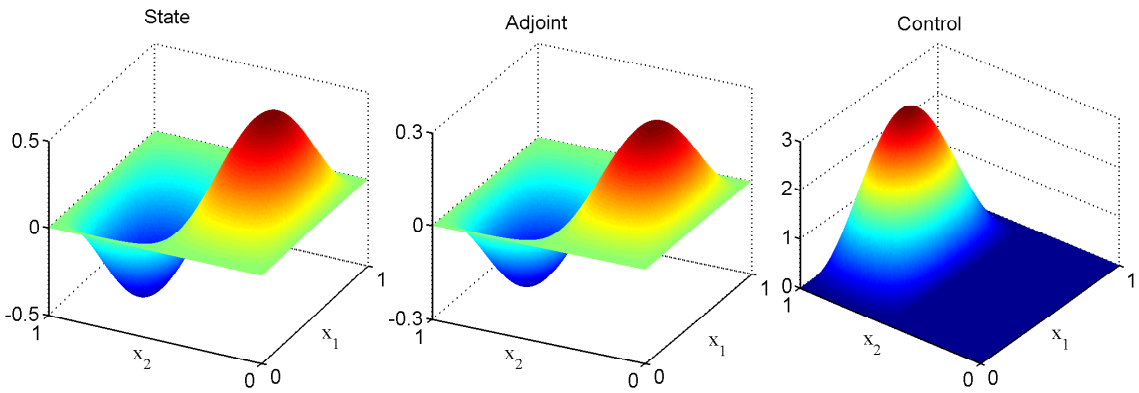


Figure 5.1: Surfaces of the exact state, adjoint, control, respectively, in Example 5.3.1.

Figure 5.1 shows the exact solutions of the state, adjoint, control respectively. In this example, the initial mesh is constructed by beginning with a uniform square mesh 4×4 and then dividing each square into two triangles. The numerical solutions are computed on a series of triangular meshes which are created from uniform refinement of an initial mesh. At each refinement, every triangle is divided into four congruent triangles. Table 5.1 shows the numerical errors and convergence orders for various uniform refinements.

Nodes	$\|y - y_h\|_{L^2}$	order	$\|p - p_h\|_{L^2}$	order	$\|u - u_h\|_{L^2}$	order
25	4.68e-2	-	2.82e-2	-	1.70e-1	-
81	1.24e-2	1.92	6.10e-3	1.90	4.84e-2	1.82
289	3.10e-3	2.00	1.54e-3	1.99	1.20e-2	2.02
1089	7.62e-4	2.02	3.80e-4	2.02	2.86e-3	2.06
4225	1.87e-4	2.02	9.38e-5	2.02	6.92e-4	2.05

Table 5.1: Convergence results on uniform mesh in Example 5.3.1.

Let us recall that it has been proved that $\|u - u_h\|_{L^2} = O(h^{3/2})$ for piecewise linear approximations of the control by the fully discrete approaches in [18, 104]. Hinze [52] improved the convergence rate to $O(h^2)$ by variational discretization. The reason of the convergence rate $O(h^{3/2})$ instead of the optimal order $O(h^2)$ in fully discrete approach is that u may not be smooth near the boundary even if y and p are smooth there. When we use SIPG method, we do not have any problem related the smoothness of u near the boundary due to the weak

treatment of DG methods for the boundary conditions. Hence, we have obtained $O(h^2)$ for the convergence rate of $\|u - u_h\|_{L^2}$, as given in Table 5.1.

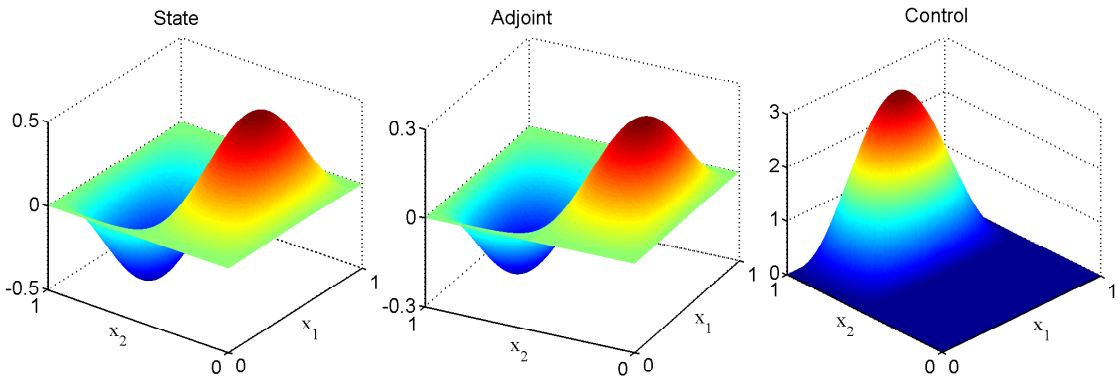


Figure 5.2: Computed state, adjoint, control, respectively on uniform mesh (4225 nodes) using linear elements in Example 5.3.1.

The same example has been solved in [54] by using variational discretization for constrained optimal control problem governed by convection diffusion equations, where the state equation is approximated by the edge stabilization Galerkin method. Comparing the results in Table 5.1 with the results obtained in [54], it turns out that the errors L_2 errors for the control in Table 5.1 are smaller than in [54] for approximately the same number of nodes.

Figure 5.2 displays the computed state, adjoint, control, respectively on uniform mesh (4225 nodes).

Example 5.3.2 *This example is taken from [54] to illustrate the efficiency of the adaptivity. Let*

$$\Omega = [0,1]^2, \quad \beta = (2,3)^T, \quad r = 1, \quad \omega = 0.1 \ \text{ and } \ u_d = 0.$$

The analytical solution of state is given by

$$y(x_1,x_2) \;=\; \frac{2}{\pi}\arctan\left(\frac{1}{\sqrt{\epsilon}}\left[-\frac{1}{2}x_1 + x_2 - \frac{1}{4}\right]\right),$$

which is a function with a straight interior layer. The corresponding adjoint is

$$\begin{aligned} p(x_1,x_2) \;&=\; 16x_1(1-x_1)x_2(1-x_2) \\ &\times\; \left(\frac{1}{2} + \frac{1}{\pi}\arctan\left[\frac{2}{\sqrt{\epsilon}}\left(\frac{1}{16} - \left(x_1 - \frac{1}{2}\right)^2 - \left(x_2 - \frac{1}{2}\right)^2\right)\right]\right), \end{aligned}$$

which is a function with a circular interior layer. The optimal control is given by

$$u(x_1, x_2) = \max\{-5, \min\{-1, -\frac{1}{\omega}p(x_1, x_2)\}\}.$$

Figure 5.3 shows the exact solutions of the state, adjoint, control, respectively, using linear elements for $\epsilon = 10^{-4}$. The state exhibits a straight interior layer, whereas the adjoint exhibits a circular interior layer. Hence, these inner layers need to be resolved to obtain more accurate solutions.

The initial mesh is constructed by beginning on uniform square 16×16, and then dividing each square into two triangles. By applying the adaptive procedure on the initial mesh, the locally generated meshes are displayed for various level in Figure 5.4. The Figure 5.4 shows that our error indicator mostly picks out the circular interior layer. Hence, the difference between uniformly refined mesh and adaptively refined mesh is not so much for the L_2 error of state. The global L_2 errors of the state, the adjoint and the control are also exhibited in Figure 5.5.

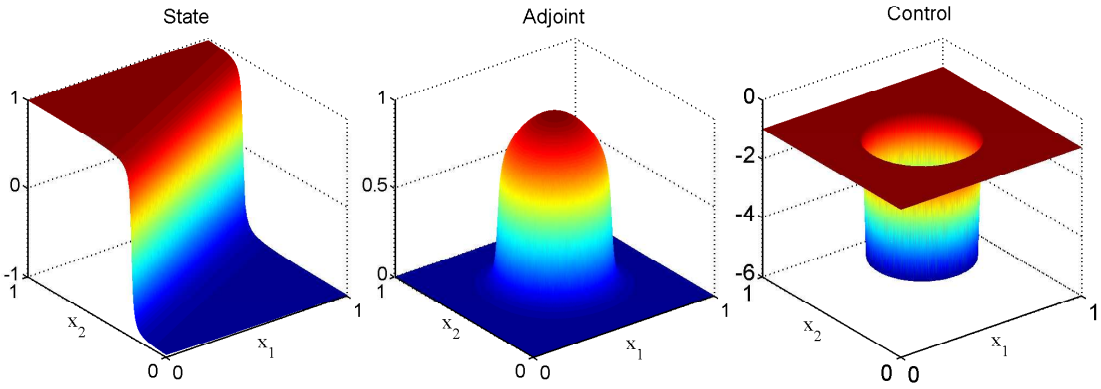


Figure 5.3: Surfaces of the exact state, adjoint, control, respectively, using linear elements for $\epsilon = 10^{-4}$ in Example 5.3.2.

This example has also been studied by Hinze, Yan and Zhou [54] using norm-residual based estimator with an edge stabilization technique. The estimator in [54] especially picks out the straight layer well with respect to ours. Hence, authors [54] obtain visually better meshes for $\epsilon = 10^{-4}$. However, comparing the results in Table 5.2 with in [54], it turns out that our error indicator gives more accurate results, especially, for the adjoint and control.

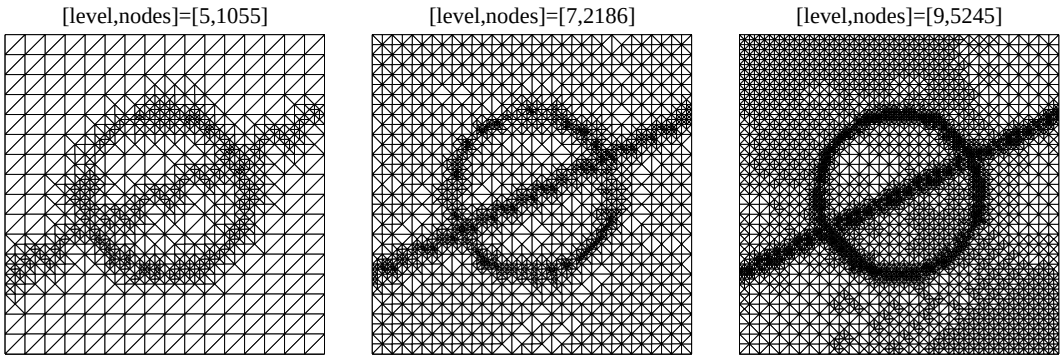


Figure 5.4: Adaptively refined meshes with linear elements for $\epsilon = 10^{-4}$ in Example 5.3.2.

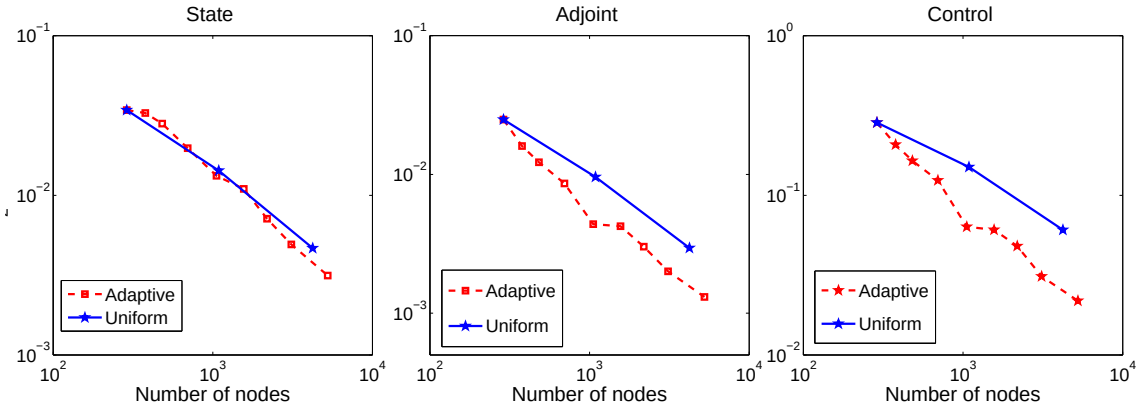


Figure 5.5: Errors in L_2 norm of state, adjoint and control using linear elements for $\epsilon = 10^{-4}$ in Example 5.3.2.

	uniform mesh, nodes=1089	adaptive mesh, nodes 1055
$\|y - y_h\|_{L^2}$	1.429e-002	1.326e-002
$\|p - p_h\|_{L^2}$	9.581e-003	4.379e-003
$\|u - u_h\|_{L^2}$	1.504e-001	6.351e-002

Table 5.2: Comparison of the error on L_2 norm of y, p and u on uniform and adaptive meshes for $\epsilon = 10^{-4}$ in Example 5.3.2.

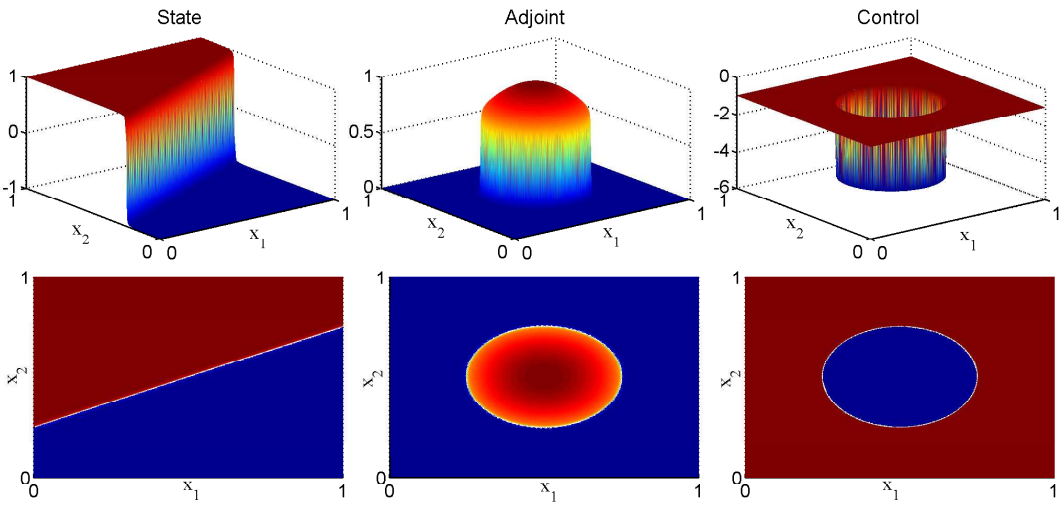


Figure 5.6: Exact solutions of the state, the adjoint and the control for $\epsilon = 10^{-6}$ in Example 5.3.2: Top row are surface plots, bottom row are top-down views.

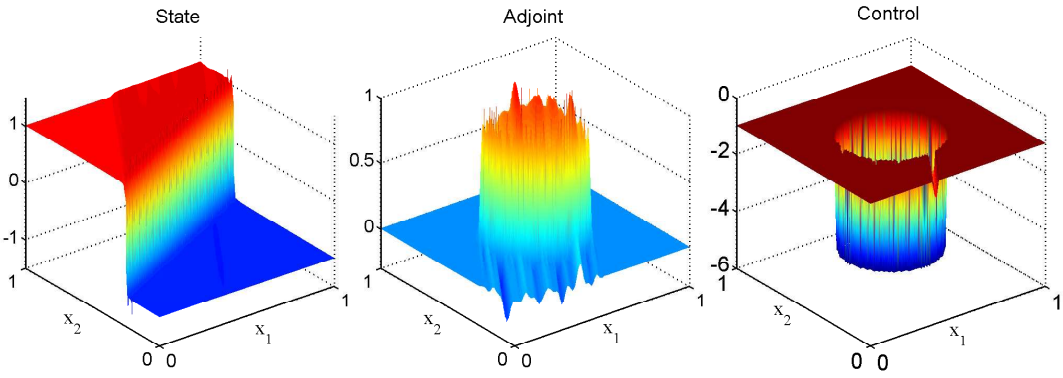


Figure 5.7: Computed state, adjoint, control, respectively on a uniform mesh (4225 nodes) using linear elements for $\epsilon = 10^{-6}$ in Example 5.3.2.

We have also tested this example for more convection-dominated case. Let $\epsilon = 10^{-6}$, Figure 5.6 displays the exact solutions of the state, the adjoint and the control.

Figure 5.7 shows that large oscillations occur on the straight interior layer for the state and on the circular interior layer for the adjoint when the initial mesh is refined uniformly. However, picking out the layers by using the error estimators given in (4.1), the oscillations are reduced in Figure 5.8. This proves the performance of the AFEM over the uniform refinement.

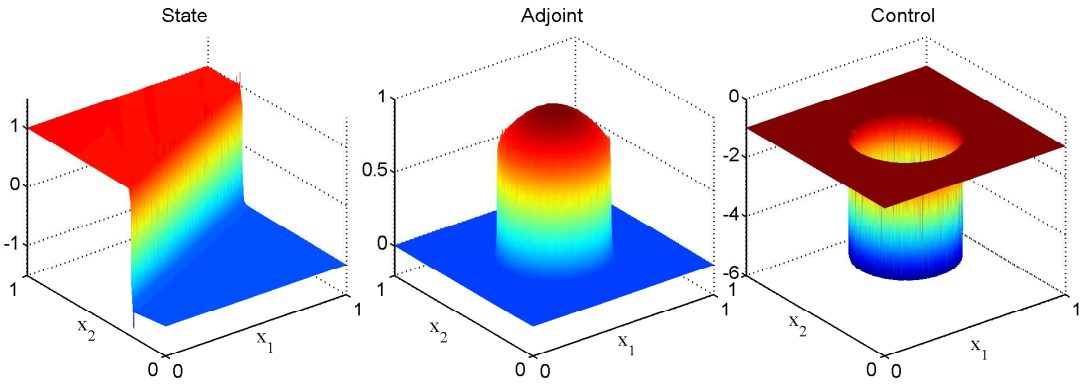


Figure 5.8: Computed state, adjoint, control, respectively on adaptively refined mesh (4135 nodes) using linear elements for $\epsilon = 10^{-6}$ in Example 5.3.2.

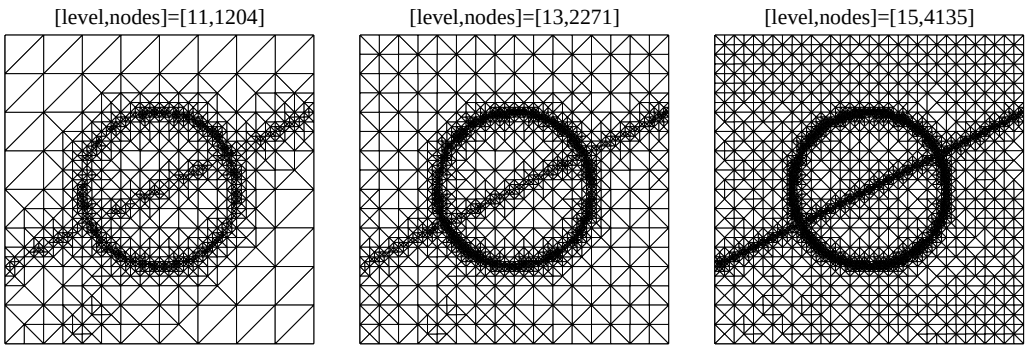


Figure 5.9: Adaptively refined meshes with linear elements at various refinement levels for $\epsilon = 10^{-6}$ in Example 5.3.2.

In Figure 5.9, the process of the adaptive procedure at the different refinement steps is shown with the estimators in (4.1) and (5.34). For the last ones, we observe that our indicator mark the elements on straight interior layer much better than $\epsilon = 10^{-4}$. Hence, we can say that the error estimators yield much accurate results for the convection-dominated cases. The L_2 errors of the state, adjoint and control are decreasing faster for AFEM than for the uniformly refined meshes in Figure 5.10.

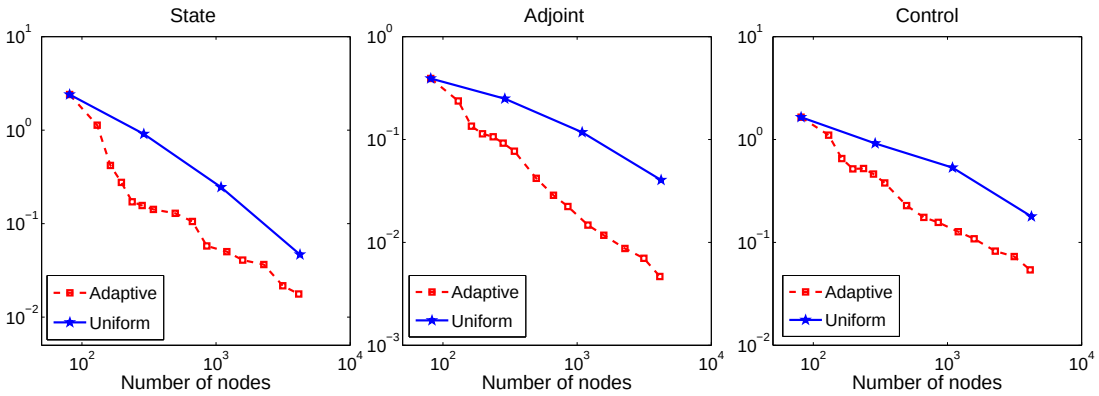


Figure 5.10: Errors in L_2 norm of the state, the adjoint and the control with linear elements and $\epsilon = 10^{-6}$ in Example 5.3.2.

Example 5.3.3 *The following example with unknown solutions is taken from Becker and Vexler [18]. The problem data are given by*

$$\Omega = (0,1)^2, \quad \epsilon = 10^{-3}, \quad \beta = (-1,-2)^T, \quad r = 1, \quad u_a = 0.5, \quad u_b = 10, \quad \omega = 0.1 \quad \textit{and} \quad u_d = 0.$$

The Dirichlet boundary condition g_D, the source function f and the desired state y_d are defined by

$$g_D = 0 \ \textit{ on } \partial\Omega, \quad f = 1 \ \textit{ and } \ y_d = 1.$$

Since the exact solution of the optimal control problem is not known, we examine the value of the cost function

$$J(y,u) := \frac{1}{2}\int_\Omega (y(x) - y_d(x))^2 dx + \frac{\omega}{2}\int_\Omega u(x)^2 dx.$$

Table 5.3 reveals evolution of values of the cost functional $J(y_h, u_h)$ for a sequence of uniformly refined meshes.

h	Nodes	$J(y_h, u_h)$	$J(y_h, u_h) - J(y_{2h}, u_{2h})$	order
2.50e-001	25	0.261645013	-	-
1.25e-001	81	0.261576704	-6.830944e-005	-
6.25e-002	289	0.261439227	-1.374771e-004	1.01
3.13e-002	1089	0.261379228	-5.999907e-005	1.20
1.56e-002	4225	0.261346837	-3.239089e-005	0.89
7.81e-003	16641	0.261334714	-1.212264e-005	1.42

Table 5.3: Evolution of values of the cost functional $J(y_h, u_h)$ for a sequence of uniformly refined meshes in Example 5.3.3.

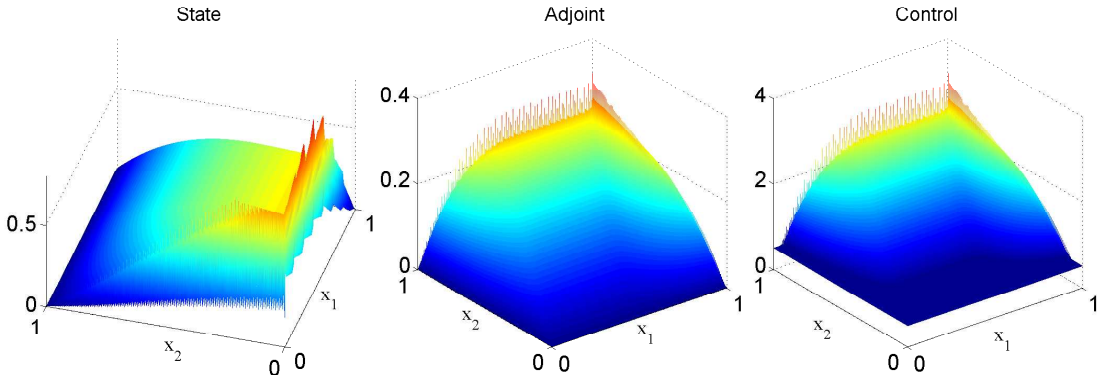


Figure 5.11: Computed state, adjoint, control, respectively on a uniform mesh (16641 nodes) using linear elements in Example 5.3.3.

Becker and Vexler [18] has also solved this example by local projection stabilization (LPS). Comparing the results in Table 5.3 and in [18], it turns out that they have obtained $J(y_h, u_h) = 0.261346$ if $h = 2^{-9}\sqrt{2}$ (approximately $2.76e-3$), but we have obtained $J(y_h, u_h) = 0.261335$ for $h = 7.81e-3$.

The computed solutions on the uniform refined mesh (16641 nodes) are shown in Figure 5.11. Since the numerical solutions exhibit oscillations, we need to resolve the some part of the domain.

Figure 5.12 displays the adaptively refined mesh on a coarse mesh constructed by beginning a uniform square mesh 4×4 and then dividing each triangle into two triangles. Figure 5.12 shows that the problem has boundary layers. By resolving the boundary layers, the oscillations in Figure 5.11 are reduced, as given in 5.13.

Evolution of values of the cost functional $J(y_h, u_h)$ for a sequence of adaptively refined meshes are given in Table 5.4.

Nodes	$J(y_h, u_h)$
25	0.261645013
55	0.261420389
101	0.261420389
190	0.261379396
359	0.261361976
619	0.261342357
863	0.261308953
1264	0.261298238
1928	0.261295682

Table 5.4: Evolution of values of the cost functional $J(y_h, u_h)$ for a sequence of adaptively refined meshes in Example 5.3.3.

The oscillations in the computed solutions using uniform meshes with 16641 nodes in Figure 5.11 are reduced in Figure 5.13 using adaptive mesh refinement with 1928 nodes. Thus, it is evident that the adaptive meshes save substantial computing time.

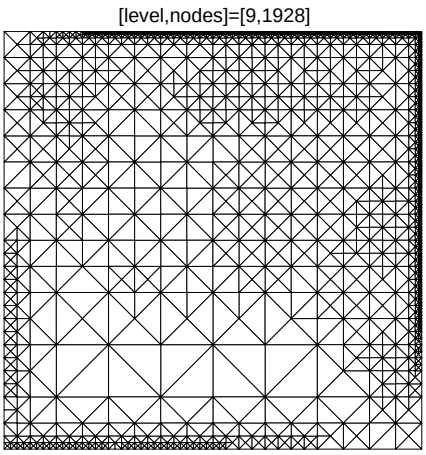


Figure 5.12: Adaptively refined mesh with linear elements in Example 5.3.3.

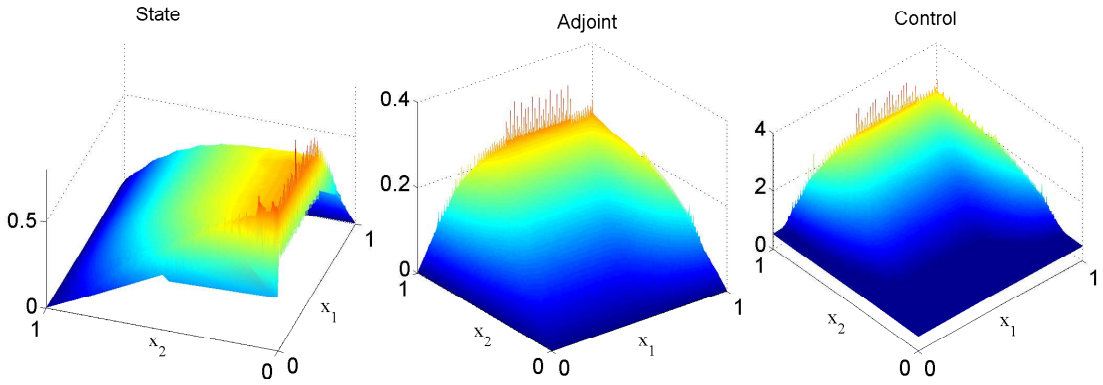


Figure 5.13: Computed state, adjoint, control, respectively on a adaptively refined mesh (1928 nodes) using linear elements in Example 5.3.3.

Example 5.3.4 *This example has been taken from [104]. The problem data are given by*

$$\Omega = [0,1]^2, \quad \epsilon = 10^{-4}, \quad \beta = (1,0), \quad r = 1 \ \text{ and } \ \omega = 1.$$

The analytical solutions of the state, adjoint and control are given by

$$y(x_1,x_2) = 4e^{(-((x_1-1/2)^2+3(x_2-0.5)^2)/\sqrt{\epsilon})}\sin(\pi x_1)\sin(\pi x_2),$$

$$p(x_1,x_2) = e^{(-((x_1-1/2)^2+3(x_2-0.5)^2)/\sqrt{\epsilon})}\sin(\pi x_1)\sin(\pi x_2),$$

$$u(x_1,x_2) = \max\{0, 2\cos(\pi x_1)\cos(\pi x_2) - 1\}.$$

At the previous examples, the layers caused from the state and adjoint equations are dominant and hence the effect of the control estimator η^u (5.34) is not seen well. By defining u_d different from zero in this example, our estimator η^u for the control picks out the layers on the boundary of the control. See Figure 5.14.

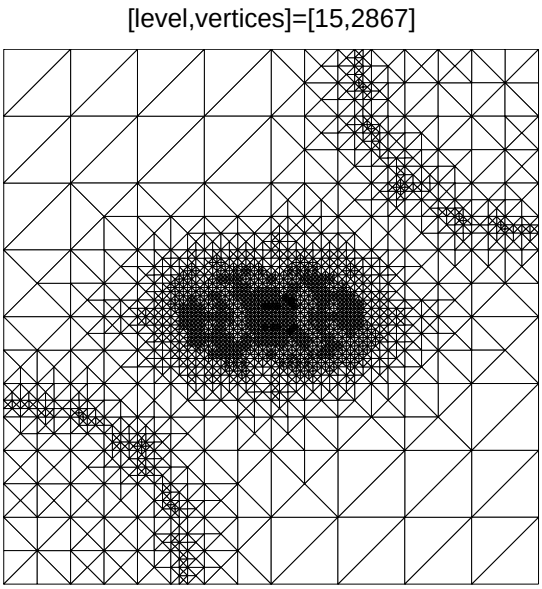


Figure 5.14: Adaptively refined mesh for Example 5.3.4

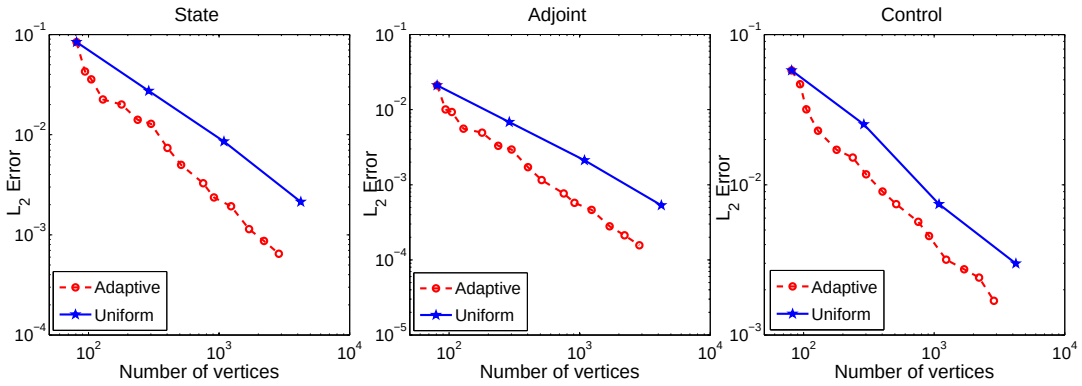


Figure 5.15: Example 5.3.4: L_2 errors in the state, the adjoint and the control.

When the uniform refinement is used, especially, the computed control exhibit spurious oscillations on the boundary of the control. See the plots in the middle row in Figure 5.16. If the adaptive refinement is used, the spurious oscillations are reduced in computed control. See the plots in the bottom row in Figure 5.16.

Figure 5.15 reveals the L_2 error for the state, the adjoint and the control for the computations using linear elements on adaptively and uniformly refined meshes. The errors decrease monotonically as the number of vertices is increased.

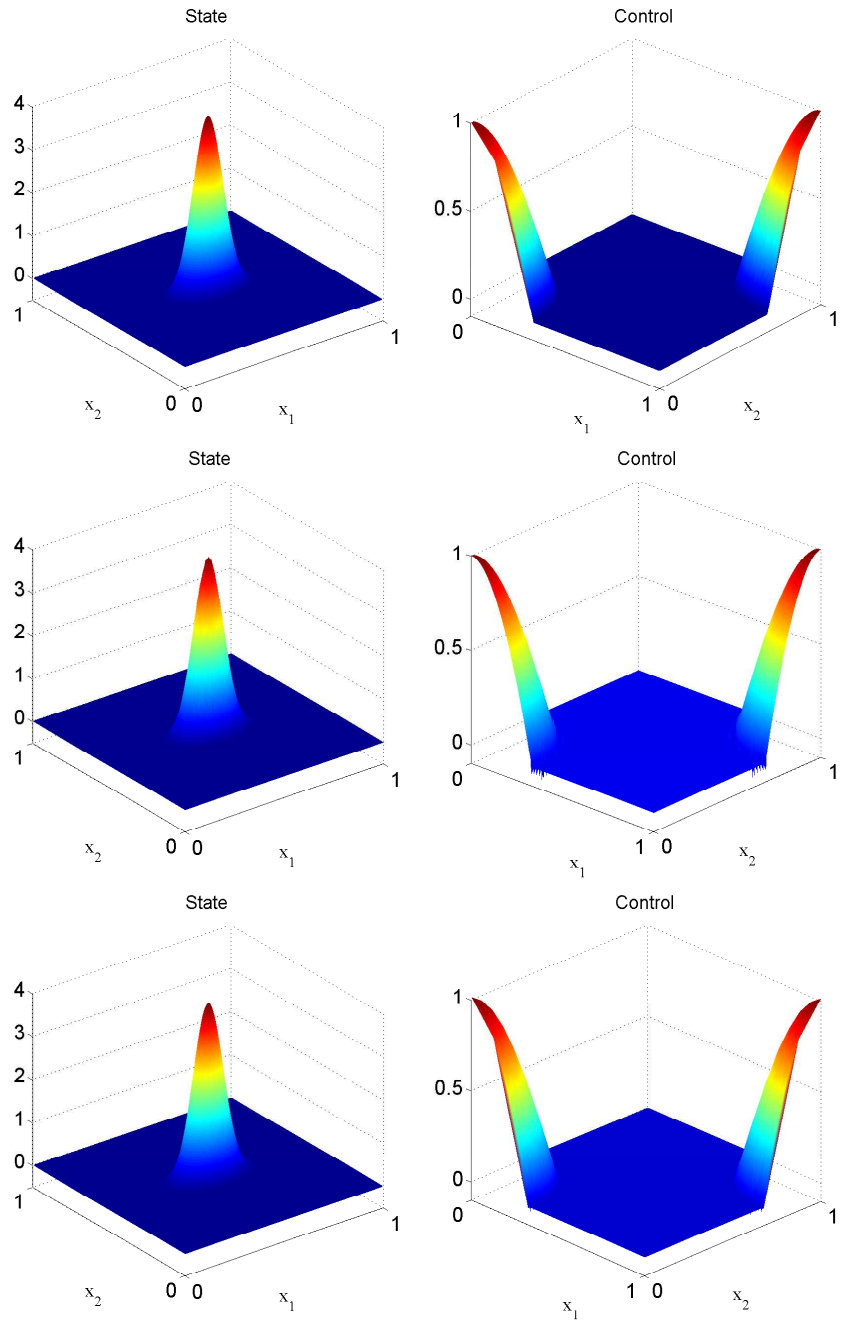


Figure 5.16: Example 5.3.2. The plots in the top row show the exact the state, adjoint and control. The plots in the middle show the computed state, adjoint and control using piecewise linear polynomials on a uniformly refined mesh with 4225 vertices; The plots in the bottom row show the computed state, adjoint and control using piecewise linear polynomials on an adaptively refinement mesh with 2867 vertices.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, we have studied the effect of the discontinuous Galerkin methods on the discretization of optimal control problems governed by linear convection diffusion equations. Moreover, we have discussed adaptive finite element method (AFEM) procedure, driven by a posteriori error estimates, for the numerical solution of unconstrained and control constrained optimal control problems governed by convection dominated equations, based on the upwind SIPG discretization.

We have shown theoretically and numerically in Chapter 3 that DG discretization of the optimal control problem leads to the same result as the DG discretization applied to optimality system for SIPG method, whereas not for nonsymmetric DG methods, i.e., NIPG, IIPG. The difference between both approaches can be reduced by superpenalization for NIPG and IIPG methods thanks to reduction on the lack of adjoint consistency. We have also studied the convergence properties of the convection dominated unconstrained optimal control problems in Chapter 4, based on the upwind SIPG discretization. The boundary and/or interior layers are resolved by applying AFEM, driven by a posteriori error estimator. Then, these results have been extended to control constraint optimal control problems in Chapter 5. Our numerical results have shown that the estimator picks out the boundary and/or interior layers effectively for both unconstrained and control constrained cases. Additionally, while solving control constrained optimal control problem, $O(h^2)$ convergence rate has been obtained for the control variable by using SIPG method at the numerical examples due to the weak treatment of SIPG method on the boundary of control.

In this thesis, we have used the conforming meshes which are implemented easily. However, there are other approaches such as nonconforming meshes which are very suitable for

DG discretization. Although there are many works for the single PDEs in this area, but not for optimal control problems. Hence, the advantages and disadvantages of nonconforming meshes should be explored in optimal control context. The elliptic DG approach in [20] allowing discontinuity of diffusion parameter ϵ and nonconforming meshes can be extended convection dominated optimal control problems. We have applied here h-adaptivity, which gives algebraic convergence rate. However, another possibility hp-adaptivity [92] applied to convection dominated single PDEs by using DG discretization in [107] can be extended to optimal control problems to obtain exponential rates of convergence.

In this thesis, we use same mesh to refine the marked elements with respect to error estimator of the state, adjoint and control although they display layers in different parts of the domain. Therefore, different meshes for each solution component, state, adjoint and control, can be alternatively used. Furthermore, the studies in here could be extended to distributed state constrained and boundary control problems.

REFERENCES

[1] F. Abraham, M. Behr, and M. Heinkenschloss, *The effect of stabilization in finite element methods for the optimal boundary control of the Oseen equations*, Finite Elem. Anal. Des. 41:229-251, 2004.

[2] M. Ainsworth, *A posteriori error estimation for discontinuous Galerkin finite element approximation*, SIAM J. Numer. Anal., 45:1777-1798, 2007.

[3] M. Ainsworth, and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Pure and Applied Mathematics, New York, 2000.

[4] N. Arada, E. Casas, and F. Tröltzsch, *Error estimates for the numerical approximation of a semilinear elliptic control problem*, Computational Optimization and Application, 23:201-229, 2002.

[5] D. N. Arnold, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal., 19:724-760, 1982.

[6] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal. 39:1749-1779, 2002.

[7] B. Ayuso, and L. D. Marini, *Discontinuous Galerkin methods for advection-diffusion-reaction problems*, SIAM J. Numer. Anal., 47:1391-1420, 2009.

[8] I. Babuška, and T. Strouboulis, *The Finite Element Method and its Reliability*, Numerical Mathematics and Scientific Computation, The Clarendon Press Oxford University Press, New York, 2001.

[9] I. Babuška, and M. Zlámal, *Nonconforming elements in the finite element method with penalty*, SIAM J. Numer. Anal., 10:45-59, 1973.

[10] W. Bangerth, and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Lectures in Mathematics, ETH-Zürich, Birkhäuser, Basel, 2003.

[11] S. Bartels, and C. Carstensen, *Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I. Low order conforming, nonconforming, and mixed FEM*, Math. Comp., 71:945-969, 2002.

[12] F. Bassi, and S. Rebay, *A high order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, J. Comput. Phys., 131:267-279, 1997.

[13] C. E. Baumann, and J. T. Oden, *A discontinuous hp finite element method for convection-diffusion problems*, Comput. Methods Appl. Mech. Engrg., 175:311-341, 1999.

[14] R. Becker, and P. Hansbo, *Energy norm a-posteriori error estimation for discontinuous Galerkin methods*, Comput. Methods Appl. Mech. Eng., 192:723-733, 2003.

[15] R. Becker, H. Kapp, and R. Rannacher, Adaptive finite element methods for optimal control of partial differential equations: Basic concepts, SIAM J. Control Optim. 39:113-132, 2000.

[16] R. Becker, and R. Rannacher, *A feed-back approach to error control in finite element methods: basic Analysis and examples*, East-West J. Numer. Math. 4:237-264, 1996.

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

[18] R. Becker, and B. Vexler, *Optimal control of the convection-diffusion equation using stabilized finite element methods*, Numer. Math. 106:349-367, 2007.

[19] M. Bergounioux, K. Ito, and K. Kunish, *Primal-dual strategy for constrained optimal control problems*, SIAM Journal on Control and Optimization, 37:1176-1194, 1999.

[20] A. Bonito, and R. Nochetto, *Quasi-optimal convergence rate of an adaptive discontinuous Galerkin method*, SIAM J. Numer. Anal., 48:734-771, 2010.

[21] M. Braack, *Optimal control in fluid mechanics by finite elements with symmetric stabilization*, SIAM J. Control Optim. 48:672-687, 2009.

[22] M. Braack, E. Burman, V. John, and G. Lube, *Stabilized finite element methods for the generalized Oseen problem*, Comput. Methods Appl. Mech. Engrg., 196:853-866, 2009.

[23] M. Braack, and A. Ern, *A posteriori control of modeling errors and discretization errors*, SIAM Mult. Mod. Sim., 1:221-238, 2003.

[24] M. Braack, and A. Ern, *Coupling multimodelling with local mesh refinement for the numerical computation of laminar flames*, Combust. Theory Model., 8:771:788, 2004.

[25] J. H. Bramble, and S. Hilbert, *Bounds for the Class of Linear Functionals with Application to Hermite Interpolation*, Numer. Math., 16:362-369, 1971.

[26] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo, *Discontinuous Galerkin approximations for elliptic problems*, Numer. Methods Partial Differential Equations, 16:365-378, 2000.

[27] E. Burman, and P. Hansbo, *Edge stabilization for Galerkin approximations of convection-diffusion-reaction problems*, Comput. Methods Appl. Mech. Eng., 193:1437-1453, 2004.

[28] E. Casas, *Optimization with PDEs*, Lecture notes of the Spring school within the programme on Optimization with interfaces and free boundaries, Regensburg, March 2009.

[29] E. Casas, and F. Tröltzsch, *Error estimates for linear-quadratic elliptic control problems, in Analysis and Optimization of Differential Systems*, Kluwer Academic, Boston, 89-100, 2003.

[30] P. Castillo, *Performance of discontinuous Galerkin methods for elliptic PDEs*, SIAM J. Scientific Computing, 24:524-547, 2002.

[31] L. Chen, and C. Zhang, *AFEM@matlab: a MATLAB package of Adaptive Finite Element Methods*, Technical Report, University of Maryland, 2006.

[32] L. Chen, *iFEM: an innovative finite element methods package in MATLAB*, Technical Report, University of California at Irvine, 2009.

[33] J. M. Cnossen, H. Bijl, M. I. Gerritsma, and B. Koren, *Aspects of goal-oriented model-error estimation in convection-diffusion problems*, Proceedings of the ECCOMAS Computational Fluid Dynamics Conference, 2006.

[34] B. Cockburn, and C. W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35:2440-2463, 1999.

[35] S. S. Collis, and M. Heinkenschloss, *Analysis of the streamline upwind/petrov Galerkin method applied to the solution of optimal control problems*, Tech. Rep. TR02-01, Department of Computational and Applied Mathematics, Rice University, 2002.

[36] C. Dawson, S. Sun, and M. Wheeler, *Compatible algorithms for coupled flow and transport*, Computer Methods in Applied Mechanics and engineering, 193:2565-2580, 2004.

[37] L. Dedè, S. Micheletti, and S. Perotto, *Anisotropic error control for environmental applications*, Appl. Numer. Math., 58:1320-1339, 2008.

[38] L. Dedé, and A. Quarteroni, *Optimal control and numerical adaptivity for advection-diffusion equations*, ESAIM: Mathematical Modelling and Numerical Analysis, 39:1019-1040, 2005.

[39] S. Deng, and W. Cai, *Analysis and Application of an Orthogonal Nodal Basis on Triangles for Discontinuous Spectral Element Methods*, Applied Numerical Analysis & Computational Mathematics, 2:326-345, 2005.

[40] J. Dougles, and T. Dupont, *Interior Penalty Procedures for Elliptic and Parabolic Galerkin Methods*, Lecture Notes in Phys. 58, Springer-Verlag, Berlin, 1976.

[41] W. Dörfler, *A convergent adaptive algorithm for Poisson's equations*, SIAM Journal on Numerical Analysis, 33:1106-1124, 1996.

[42] D. A. Dunavant, *High degree efficient symmetrical Gaussian quadrature rules for the triangle, Internat. J. Numer. Methods Engrg*, 21:1129-1148, 1985.

[43] A. Ern, A. F. Stephansen, and M. Vohralík, *Guaranted and robust discontinuous Galerkin a posteriori error estimates for convection-diffusion-reaction problems*, J. Comput. Appl. Math., 234:114-130, 2010.

[44] H. J. S. Fernando, S. M. Lee, and J. Anderson, *Urban fluid mechanics: Air circulation and contaminant dispersion in cities*, Environmental Fluid Mechanics, 1:107-164, 2001.

[45] M. S. Gockenbach, *Understanding and Implementing the Finite Element Method*, SIAM, 2006.

[46] J. Gopalakrishnan, and G. Kanschat, *A multilevel discontinuous Galerkin method*, Numer. Math., 95:527-550, 2003.

[47] M. D. Gunzburger, *Perspectives in Flow Control and Optimization*, SIAM, Philadelphia, 2003.

[48] M. Heinkenschloss, and D. Leykekhman, *Local error estimates for SUPG solutions of advection-dominated elliptic linear-quadratic optimal control problems*, SIAM J. Numer. Anal. 47:4607-4638, 2010.

[49] M. Hintermuller, R. H. W. Hoppe, *Goal-oriented adaptivity in control constrained optimal control of partial differential equations*, SIAM J. Control Optim. 47:1721-1743, 2008.

[50] M. Hintermuller, R. H. W. Hoppe, Y. Iliash, and M. Kieweg, *An a posteriori error analysis of adaptive finite element methods for distributed elliptic control problems with control constraints*, ESAIM, Control Optim. Calc. Var., 14:540-560, 2008.

[51] M. Hintermuller, K. Ito, and K. Kunish, *The primal-dual active set strategy as a semismooth Newton method*, SIAM Journal on Optimization 13:865-888, 2002.

[52] M. Hinze, *A variational discretization concept in control constrained optimization: the linear-quadratic case*, J. Computational Optimization and Applications, 30:45-63, 2005.

[53] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Mathematical Modelling: Theory and Applications, 23, Springer, 2009.

[54] M. Hinze, N. Yan, and Z. Zhou, *Variational discretization for optimal control governed by convection dominated diffusion equations*, J. Comp. Math., 27:237-253, 2009.

[55] M. Hinze, N. Yan, and Z. Zhou, A mixed finite element approximation for optimal control problems with convection-diffusion equations, Hamburger Beiträge zur Angewandten Mathematik, Report 04, 2011.

[56] M. Hinze, and F. Tröltzsch, *Discrete concept versus error analysis in pde constrained optimization*, GAMM-Mitt. 33:148-162, 2010.

[57] R. H. W. Hoppe, G. Kanschat, and T. Warburton, *Convergence analysis of an adaptive interior penalty discontinuous Galerkin method*, SIAM J. Numer. Anal., 47:534-550, 2008.

[58] P. Houston, D. Schötzau, and T. P. Wihler, *Energy norm a-posteriori error estimation of hp-adaptive discontinuous Galerkin methods for elliptic problems*, Math. Models Methods Appl. Sci. 17:33-62, 2007.

[59] P. Houston, C. Schwab, and E. Süli, *Discontinuous hp-finite element methods for advection- diffusion-reaction problems*, SIAM J. Numer. Anal., 39:2133-2163, 2002.

[60] P. Houston, and E. Süli, *Adaptive finite element approximation of hyperbolic problems, in Error Estimation and Adaptive Discretization Methods*, Computational Fluid Dynamics, Lect. Notes Comput. Sci. Engrg., 25:269-344, 2002.

[61] T. J. R. Hughes, and A. Rooks, *Streamline upwind/Petrow Galerkin formulations for convection dominated flows with particular emphasis on the incomressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Eng., 54:199-259, 1982.

[62] K. Ito, and K. Kunish, *Optimal control of elliptic variational inequalities*, Appl. Math. Optim., 41:343-364, 2000.

[63] C. Johnson, and J. Pitkäranta, *An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation*, Math. Comp., 46:1-26, 1986.

[64] G. Kanschat, and R. Rannacher, *Local error analysis of the interior penalty discontinuous Galerkin method for second order elliptic problems*, J. Numer. Math. 10:249-274, 2002.

[65] O. Karakashian, and F. Pascal, *A-posteriori error estimates for a discontinuous Galerkin approximation of second-order elliptic problems*, SIAM J. Numer. Anal. 41:2374-2399, 2003.

[66] O. Karakashian, and F. Pascal, *Convergence of adaptive discontinuous Galerkin approximations of second-order elliptic problems*, SIAM J. Numerical Analysis, 25:641-665, 2007.

[67] K. Kohls, A. Rösch, and K. G. Siebert, *A posteriori error estimators for control constrained optimal control problems*, CONSTRAINED OPTIMIZATION AND OPTIMAL CONTROL FOR PARTIAL DIFFERENTIAL EQUATIONS, International Series of Numerical Mathematics, 160:431-443, 2012.

[68] K. Kunish, and A. Rösch, *Primal-dual active set strategy for a general class of constrained optimal control problems*, SIAM Journal on Optimization 13(2):321-334, 2002.

[69] D. Kuzmin, and S. Korotov, *Goal-oriented a posteriori error estimates for transport problems*, Mathematics and Computers in Simulation, 80:1674-1683, 2010.

[70] P. Lesaint, and P. A. Raviert, *On a finite element for solving the neutron transport equation*, Mathematical aspects of finite elements in partial differential equations, Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York, 89-123, 1974.

[71] P. d. Leva, *MULTIPROD TOOLBOX, Multiple matrix multiplications, with array expansion enabled*, University of Rome Foro Italico, Rome.

[72] D. Leykekhman, *Investigation of Commutative Properties of Discontinuous Galerkin Methods in PDE Constrained Optimal Control Problems*, Department of Mathematics, University of Connecticut, 2011.

[73] D. Leykekhman, and M. Heinkenschloss, *Local error analysis of discontinuous Galerkin methods for advection-dominated elliptic linear-quadratic optimal control problems*, Tech. Rep. TR10-11, Department of Computational and Applied Mathematics, Rice University, 2010.

[74] R. Li, W. Liu, H. Ma, and T. Tang, *Adaptive finite element approximations for distributed elliptic optimal control problems*, SIAM J. Control Optim., 41:1321-1349, 2002.

[75] W. Liu, and N. N. Yan, *A posteriori error estimates for distributed optimal control problems*, Adv. Comp. Math. 15:285-309, 2001.

[76] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer Verlag, Berlin, Heidelberg, New York, 1971.

[77] G. Lube, and B. Tews, *Distributed and boundary control of singularly perturbed advection-diffusion-reaction problems*, Boundary and Interior Layers, Lecture Notes in Computational Science and Engineering, 69:205-215, 2009.

[78] C. Meyer, *Skript zur Vorlesung Optimale Steuerung partieller Differentialgleichungen*, Lecture Notes, Reseach Center Computational Engineering, Technische Universität Darmstadt, 2010.

[79] C. Meyer, and A. Rösch, *Superconvergence properties of optimal control problems*, SIAM J. Control Optim., 43:970-985, 2004.

[80] B. Mohammadi, and O. Pironneau, em Applied Shape optimization for fluids, Oxford University Press, Oxford, 2001.

[81] E. Nederkoorn, *Adaptive finite element methods for linear-quadratic convection dominated elliptic optimal control problems*, Ms. Thesis, Computational and Applied Mathematics, Rice University, 2010.

[82] J. T. Oden, and S. Prudhomme, *Estimation of Modeling Error in Computational Mechanics*, J. Comp. Phys., 182:496-515, 2002.

[83] D. Parra-Guevara, and Y. N. Skiba, *On optimal solution of an inverse air pollution problem: theory and numerical approach*, Math. comput. Modelling, 43:766-778,2006.

[84] J. Peraire, and P. O. Persson, *The Compact Discontinuous Galerkin (CDG) Method for Elliptic Problems*, SIAM J. Sci. Comput., 30:1806-1824, 2008.

[85] S. Prudhomme, and J. T. Oden, *On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors*, Computer Methods in Applied Mechanics and Engineering, 176:313-331, 1999.

[86] W. H. Reed, and T. R. Hill, *Triangular Mesh Methods for the Neutron Transport Equation*, Tech. Report LA-UR-73-479, Los Alomos Scientific Laboratory, Los Alomos, NM, 1973.

[87] B. Rivière, *Discontinuous Galerkin methods for solving elliptic and parabolic equations, Theory and implementation*, SIAM Volume 35, Frontiers in Applied Mathematics, 2008.

[88] B. Rivière, and M. Wheeler, *A-posteriori error estimates for a discontinuous Galerkin method applied to elliptic problems*, Comput. Math. Appl. 46:141-163, 2003.

[89] B. Rivière, M. F. Wheeler, and V. Girault, *Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems I*, Comput. Geosci., 3:337-360, 1999.

[90] A. Rösch, *Error estimates for linear-quadratic control problems with control constraints*, Optimization Methods and Software, 21:121-134, 2005.

[91] D. Schötzau and L. Zhu, *A robust a-posteriori error estimator for discontinuous Galerkin methods for convection-diffusion equations*, Appl. Numer. Math., Vol. 59:2236-2255, 2009.

[92] C. Schwab, *p- and hp-Finite Element Methods-Theory and Application to Solid and Fluid Mechanics*, Oxford University Press, 1998.

[93] L. R. Scott, and S. Zhang, *Finite element interpolation of nonsmooth functions satisfying boundary conditions*, Math. Comp., 54:483-493, 1990.

[94] S. J. Sherwin, and G. E. Karniadakis, *A new triangular and tetrahedral basis for high-order (hp) finite element methods*, Int. J. Numer. Meth. Engng., 38:3775-3802, 1995.

[95] P. Solin, K. Segeth, and I. Dolezel, *Higher-Order Finite Element Methods* , Chapman & Hall/CRC Press, 2003.

[96] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Graduate Studies in Mathematics, Vol. 112, American Mathematical Society, 2010.

[97] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh Refinement Techniques*, Wiley-Teubner Series: Advances in Numerical Mathematics, Wiley Teubner, Chicester, New York, Stuttgart, 1996.

[98] R. Verfürth, *A posteriori error estimates for convection-diffusion equations*, Numer. Math., 80:641-663, 1998.

[99] R. Verfürth, *Robust a-posteriori error estimates for stationary convection-diffusion equations*, SIAM J. Numer. Anal. 43:1766-1782, 2005.

[100] R. Verfürth, *Robust a posteriori error estimates for nonstationary convection diffusion equations*, SIAM J. Numer. Anal., 43:1783-1802, 2005.

[101] B. Vexler, and W. Wollner, *Adaptive finite elements for elliptic optimization problems with control constraints*, SIAM J. Control Optim., 47:1150-1177, 2008.

[102] M. F. Wheeler, *An elliptic collocation-finite element method with interior penalties*, SIAM J. Numer. Anal., 15:152-161, 1978.

[103] C. Xiong, and Y. Li, *Error analysis for optimal control problem governed by convection diffusion equations DG method*, Journal of Computational and Applied Mathematics, 235:3163-3177, 2011.

[104] N. Yan, and Z. Zhou, *A priori and a posteriori error analysis of edge stabilization Galerkin method for the optimal control problems governed by convection-dominated diffusion equation*, Journal of Computational and Applied Mathematics 223:198-217, 2009.

[105] N. Yan, and Z. Zhou, *A RT Mixed FEM/DG Scheme for Optimal Control Governed by Convection Diffusion Equations*, J. Sci. Comput., 41: 273-299,2009.

[106] Z. Zhou, and N. Yan, *The local discontinuous Galerkin method for optimal control problem governed by convection-diffusion equations*, International Journal of Numerical Analysis & Modeling 7:681-699, 2010.

[107] L. Zhu, and D. Schötzau, *A robust a-posteriori error estimate for hp-adaptive DG methods for convection-diffusion equations*, IMA J. Numer. Anal., 31:971-1005, 2011.

[108] O. C. Zienkiewicz, and J. Z. Zhu, *A simple error estimator and adaptive procedure for practical engineering analysis*, Internat. J. Numer. Meth. Engrg., 24:337-357, 1987.

Appendix A

MATLAB Routine

In this Appendix, the MATLAB routines used for solving the optimal control problems using DG and AFEM will be explained. In all programmes, the sparse matrix structure is used with coding style "vectorization". In the setting of MATLAB programming, vectorization can be understood as a way to replace for loops by matrix operations or other fast built in functions. In addition, we use Multiple matrix multiplications [71] "MULTIPROD" to decrease the number of "for" loops which affect the performance of MATLAB programme.

A.1 Sparse Matrix in MATLAB

Sparse matrix algorithms require less computational time by avoiding operations on zero entries and sparse matrix data structures require less computer memory by not storing many zero entries. A natural storage scheme has a basic idea: use a single array to store all nonzero entries and two additional integer arrays to store the indices of nonzero entries. Assuming A be a $m \times n$ matrix containing only p nonzero elements let us look at the following simple example:

$$A = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 4 & 6 \\ 0 & 0 & 0 \end{bmatrix}, \quad i = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \quad j = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad s = \begin{bmatrix} 8 \\ 4 \\ 6 \end{bmatrix}.$$

where i vector stores row indices of non-zeros, j column indices, and s the value of nonzeros. All three vectors have the same length p. There are several alternative forms to call sparse matrix using i, j, s as inputs. The most commonly used one is

```
A = sparse(i,j,s,m,n).
```

This call generates an $m \times n$ sparse matrix, using $[i; j; s]$ as the coordinate formate. The first three arguments all have the same length. If a pair of indices occurs more than once in i and j, sparse adds the corresponding values of s together. This nice summation property is very useful for finite element computation.

A.2 Multiprod Toolbox

"MULTIPROD" [71] is a powerful, quick and memory efficient generalization for N-D arrays of the MATLAB matrix multiplication operator (*). While the latter works only with 2-D arrays, "MULTIPROD" works also with multidimensional arrays. Besides the element-wise multiplication operator (.*), MATLAB includes only two functions which can perform products between multidimensional arrays: "DOT" and "CROSS". However, these functions can only perform two kinds of products: the dot product and the cross product, respectively. Conversely, "MULTIPROD" can perform any kind of multiple scalar-by-matrix or matrix multiplication:

- Arrays of scalars by arrays of scalars, vectors (*) or matrices.
- Arrays of vectors (*) by arrays of scalars, vectors (*) or matrices.
- Arrays of matrices by arrays of scalars, vectors (*) or matrices.
  (*) internally converted by MULTIPROD into row or column matrices.

A.3 The Mesh Data Structure

In this section, we will define the data structure of a triangular mesh on a polygonal domain in $\mathbb{R}^2$. The data structure presented here is based on simple arrays [31, 32] which are stored in a MATLAB "struct". A "struct" is a data structure that collects two or more data fields in one object that can then be passed to routines. The mesh "struct" has the following fields:

- Nodes, Elements, Edges, intEdges, bdEdges, EdgeEls, ElementsE,
- Dirichlet, Neumann, vertices1, vertices2, vertices3.

To initialize the mesh, we define firstly the initial nodes, the elements and the Dirichlet and Neumann conditions.

```
% Generate the mesh
Nodes = [0,0; 0.5,0; 1,0; 0,0.5; 0.5,0.5; 1,0.5; 0,1; 0.5,1;1,1];  % Nodes
Elements = [4,1,5; 1,2,5; 5,2,6; 2,3,6; 7,4,8; 4,5,8;8,5,9;5,6,9]; % Elements
Dirichlet = [1,2; 2,3; 1,4; 3,6; 4,7; 6,9; 7,8; 8,9];              % Dirichlet
Neumann   = [];                                                    % Neumann
mesh = getmesh(Nodes,Elements,Dirichlet,Neumann);
```

In the node array **Nodes**, the first and second rows contain x- and y- coordinates of the nodes. In the element array **Elements**, the three rows contain indices to the vertices of elements in the anti-clockwise order. The first and second rows of the matrix **Edges**(1:NE,1:2) contain indices of the starting and ending points. The column is sorted in the way that for the k-th edge, **Edges**$(k,1)$ < **Edges**$(k,2)$. The following code will generate an **Edges** matrix. The edge array is obtained as the following:

```
% Define Edges, Interior Edges and Boundary Edges
totalEdge = sort([Elements(:,[2,3]); Elements(:,[3,1]); Elements(:,[1,2])],2);
[i,j,s] = find(sparse(totalEdge(:,2),totalEdge(:,1),1));
Edges = [j,i];
bdEdges = find(s==1); intEdges=find(s==2);
ElementsE = reshape(j,NT,3);
```

The first line collects all edges from the set of triangles and sort the column such that totalEdge$(k,1)$ < totalEdge$(k,2)$. The interior edges are repeated twice in totalEdge. We use the summation property of sparse command to merge the duplicated indices. The nonzero vector s takes values 1 (for boundary edges) or 2 (for interior edges). We use then s to find the edge number of boundary edges, **bdEdges**, and interior edges, **intEdges**. In the last line, we obtain **ElementsE** matrix where each element is represented by edges. Furthermore, an $Ne \times 2$ array called **EdgeEls** is used to show connection between edges and elements.

```
NT=size(Elements,1);
[~, i2, j] = unique(totalEdge,'rows');
i1(j(3*NT:-1:1)) = 3*NT:-1:1; i1=i1';
k1 = ceil(i1/NT); t1 = i1 - NT*(k1-1);
k2 = ceil(i2/NT); t2 = i2 - NT*(k2-1);
EdgeEls = [t1,t2];
```

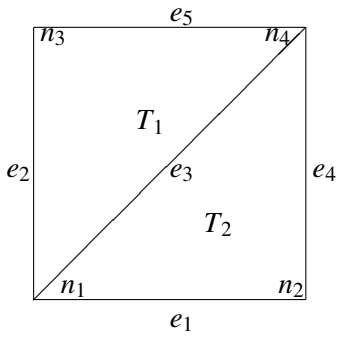


Figure A.1: A mesh with two triangles, $\Omega = [0,1] \times [0,1]$.

If the i'th edge is an interior edge, then $\mathbf{EdgeEls}(i,1) = t_{i,1}$, $\mathbf{EdgeEls}(i,2) = t_{i,2}$.

If the i'th edge is an boundary edge, then $\mathbf{EdgeEls}(i,1) = t_{i,1}$, $\mathbf{EdgeEls}(i,2) = t_{i,1}$.

If an instance of the mesh "struct" is given the variable name T, then one can refer to any of the fields using the syntax T. FieldName (for example, T. Nodes). For example we consider the mesh shown in Figure A.1 with two triangles, T_1 and T_2, five edges, e_1, e_2, e_3, e_4, e_5, and four nodes, n_1, n_2, n_3, n_4.

$$T.Edges = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 2 & 4 \\ 3 & 4 \end{bmatrix}, \quad T.intEdges = \begin{bmatrix} 3 \end{bmatrix}, \quad T.bdEdges = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 5 \end{bmatrix}$$

and

$$T.Elements = \begin{bmatrix} 3 & 1 & 4 \\ 1 & 2 & 4 \end{bmatrix}, \quad T.EdgeEls = \begin{bmatrix} 2 & 2 \\ 1 & 1 \\ 1 & 2 \\ 2 & 2 \\ 1 & 1 \end{bmatrix}.$$

There are also two MATLAB routine related to mesh structure: **label(Nodes,Elements)** to label the longest edge of each triangle as the base and **uniformrefine(mesh)** to refine the current triangulation by dividing each triangle into four triangles.

A.4 Optimal Control Problems

In this section, we describe the MATLAB routines to solve the optimal control problems. When we use the same basis spaces (ϕ_i) for the state, adjoint and control, the matrices $\mathbb{M}, \mathbb{B}$ and $\mathbb{Q}$ defined in Section 3.2 and 3.3 are the same, i.e., $\mathbb{M} = \mathbb{B} = \mathbb{Q}$.

$$(M_E)_{i,j} = \int_E \phi_{j,E}\phi_{i,E}\, dx \qquad \forall 1 \leq i, j, \leq N_{loc}.$$

We compute the integrals on the reference element $\hat{E}$ since the computation of integrals on physical element E is costly. Then, applying a change of variable with the mapping F_E given in 2.3.2, the integrals can be computed on the reference elements such that:

$$(M_E)_{i,j} = 2|E| \int_{\hat{E}} \hat{\phi}_j\hat{\phi}_i\, dx \qquad \forall 1 \leq i, j, \leq N_{loc}.$$

The following script is used to solve unconstrained optimal control problems:

```
% Compute global matrices and right-hand side for state equation
[diff_s,conv_s,reac_s,mass,f_s]=global_system(mesh,@fdiff,@fadv_state,...
       @freact_state,@fsource,@DBC_state,@NBC_state,penalty,eps,degree);
% Compute global matrices and right-hand side for adjoint equation
[diff_a,conv_a,reac_a,~,f_a]=global_system(mesh,@fdiff,@fadv_adjoint,...
       @freact_adjoint,@y_desired,@DBC_adjoint,@NBC_adjoint,penalty,eps,degree);
M=mass;                          % mass matrix
A= diff_s+conv_s+reac_s;         % stiffness matrix for state equation
Adj_A=  diff_a+conv_a+reac_a;    % stiffness matrix for adjoint equation
rs= f_s;  % right-hand side for state equation
g= f_a;   % right-hand side for adjoint equation
H=sparse(Nloc*Nel,Nloc*Nel);     % zero matrix
HR=sparse(Nloc*Nel,1);           % zero vector
switch Equation.controlapp
  case 1
  % Discretize then optimize
      K=[M,H,A'; H, Equation.omega*M, -M' ; A, -M, H ];  R=[g;HR;rs];
  case 2
  % Optimize the Discretize
      K=[M,H,Adj_A; H, Equation.omega*M, -M ; A, -M, H]; R=[g;HR;rs];
end
% Solve the linear system
[L,U,P,Q]=lu(K);
s = Q*(U\(L\(P*R)));
```

For the control contstrained case, we use the above Matlab routine in the PDAS strategy given in Chapter 5.

```
% Define coefficients of lower (u_a) and upper (u_b) bound of control
ua=zeros(Nloc*Nel,1);         ua(1:Nloc*Nel)=Equation.uA;
ub=zeros(Nloc*Nel,1);         ub(1:Nloc*Nel)=Equation.uB;
H=sparse(Nloc*Nel,Nloc*Nel);  IP=speye(Nloc*Nel);
%  Initialize Active sets
Aaold=sparse(Nloc*Nel,1);     Abold=sparse(Nloc*Nel,1);
%  Initialize inactive set
Iold=sparse(ones(Nloc*Nel,1));
% Compute global matrices and right-hand side for the state and adjoint
% Active set method
n=0; dnold=Inf; done=0;
while done==0 && n<20
    switch Equation.controlapp
        case 1
        % Discretize then optimize
        case 2
        % Optimize the Discretize
            K=[M,H,Adj_A; H,  omega*IP, -diag(Iold) ; A, -M, H];
            R=[g;(omega*Aaold.*ua)+(omega*Abold.*ub);r];
        end
% Solve the linear system
[L,U,P,Q]=lu(K);
s = Q*(U\(L\(P*R)));
% Extract coefficients of state, adjoint, control
ycoef=s(1:Nloc*Nel);
ucoef=s(Nloc*Nel+1:2*Nloc*Nel);
pcoef=s(2*Nloc*Nel+1:3*Nloc*Nel);
% Update active and inactive sets
Aaold=sparse((pcoef-(omega*ua))<0);
Abold = sparse((pcoef-(omega*ub))>0);
Iold=sparse((ones(Nloc*Nel,1)-Aaold)-Abold);
% Compute tolerans to exit active set loop
dnnew = full(( omega * omega * sum(((ucoef-ua).*Aaold).^2) / (Nloc*Nel) ) + ...
      ( omega * omega * sum(((ucoef-ub).*Abold).^2) / (Nloc*Nel) ) + ...
      ( sum(((-pcoef+(omega*ucoef)).*Iold).^2) / (Nloc*Nel)));
if ((dnnew < sqrt(eps)) && (dnold == dnnew))
    done = 1;
end
```

```
dnold=dnnew;  n=n+1;
end
```

We use global matrices and vector (for right-hand side) obtained from the state and adjoint equation to solve the optimality system, see Figure A.2.

A.5 Global Matrices and Right-Hand Side Vector

The routine for assembling of local contributions to global matrix and right-hand sides is called **global_system**.

```
[Diff_global,Conv_global,Reac_global,Fglobal]=global_system(mesh,...
        fdiff,fadv,freact,fsource,DBCexact,NBCexact,penalty,eps,degree)
```

This routine takes the structure of mesh, diffusion, convection, reaction, source function, Dirichlet and Neumann boundary conditions, penalty parameter (penal), degree of polynomials (k), a parameter to decide which DG method will be used (eps) as inputs. We can solve the system formed by the global matrices and right-hand side vector using sparse LU-factorization.

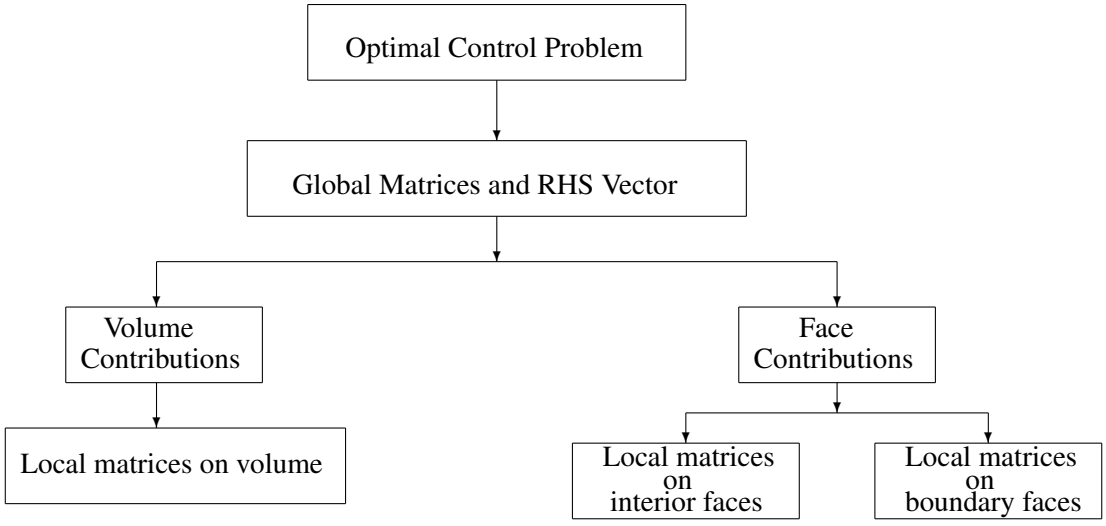


Figure A.2: The structure of MATLAB routine to solve optimal control problem.

```
% Compute global matrices
[Diff_global,Conv_global,Reac_global,Fglobal]=global_system(mesh,...
```

```
        fdiff,fadv,freact,fsource,DBCexact,NBCexact,penalty,eps,degree)
% Solve the linear system
G=Diff_global+Conv_global+Reac_global;
[L,U,P,Q]=lu(G);
y = Q*(U\(L\(P*Fglobal)))
```

The global matrices D_{global}, C_{global} and R_{global} of diffusion, convection and reaction, respectively are assembled in two steps: volume and face contributions.

A.5.1 Volume Contributions

The local matrices D_E, C_E and R_E of diffusion, convection and reaction terms, respectively, are added to the block diagonal entries of D_{global}, C_{global} and R_{global}, respectively. Assuming the mesh elements from 1 to N_{el}, the local contributions b_E can be added to b_{global} in the same algorithm.

Algorithm 1: Volume Contributions [87]
initialize k=0
loop over the elements: for k=1 to N_{el} do
  compute local matrices D_{E_k}, C_{E_k}, R_{E_k} and b_{E_k}
  for i=1 to N_{el} do
    ie=i+k
    for j=1 to N_{el} do
      je=j+k
      $D_{global}(ie, je) = D_{global}(ie, je) + D_E(i, j)$
      $C_{global}(ie, je) = C_{global}(ie, je) + C_E(i, j)$
      $R_{global}(ie, je) = R_{global}(ie, je) + R_E(i, j)$
    end
    $b_{global}(ie) = b_{global}(ie) + b_{E_k}(i)$
    k=k+Nloc
  end
end
In **Algorithm 1**, we compute the local matrices on the element E using the following:

```
% Compute local volume matrix
[Diff_loc,Floc,Conv_loc,Reac_loc]=localmat_vol(mesh,fdiff,fadv,freact,fsource,degree);
```

This routine takes the structure of mesh, diffusion, convection, reaction and source functions and degree of polynomials as inputs.

A.5.1.1 Local matrices on volume

We compute the matrices D_E, C_E, R_E resulting from the volume integral over a fixed element E :

$$(D_E)_{i,j} = \int_E \epsilon \nabla \phi_{j,E} \cdot \nabla \phi_{i,E}\, dx, \qquad (C_E)_{i,j} = \int_E \beta \cdot \nabla \phi_{j,E} \phi_{i,E}\, dx, \qquad (R_E)_{i,j} = \int_E r \phi_{j,E} \phi_{i,E}\, dx.$$

where $\forall\ 1 \leq i, j, \leq N_{loc}$. Applying a change of variable with the mapping F_E, the integrals can be computed on the reference elements:

$$\begin{aligned}
(D_E)_{i,j} &= 2|E| \int_{\hat{E}} \epsilon (B_E^T)^{-1} \hat{\nabla} \hat{\phi}_i \cdot (B_E^T)^{-1} \hat{\nabla} \hat{\phi}_j\, dx, \\
(C_E)_{i,j} &= 2|E| \int_{\hat{E}} \beta \cdot (B_E^T)^{-1} \hat{\nabla} \hat{\phi}_j \hat{\phi}_i\, dx, \\
(R_E)_{i,j} &= 2|E| \int_{\hat{E}} (r \circ F_E) \hat{\phi}_j \hat{\phi}_i\, dx.
\end{aligned}$$

The volume contributions to the local right-hand side b_E are $(b_E)_i = \int_E f \phi_{i,E}\, dx$.

Algorithm 2: Computing local contributions from element E [87]

initialize $D_E = 0$, $C_E = 0$, $R_E = 0$
initialize the quadrature weights w and points s
loop over quadrature points : for k=1 to N_G do
  compute determinant of B_E
  for i=1 to N_{loc} do
    compute values of basis functions $\phi_{i,E}(s(k))$
    compute derivatives of basis functions $\nabla_{i,E}(s(k))$
  end
  compute global coordinates x of quadrature points s(k)
  compute source function f(x)
  for i=1 to N_{loc} do
    for j=1 to N_{loc} do
      $D_E(i,j) = D_E(i,j) + w(k) det(B_E) \epsilon \nabla \phi_{j,E}(s(k)) \cdot \nabla \phi_{i,E}(s(k))$
      $C_E(i,j) = C_E(i,j) + w(k) det(B_E) \beta \cdot \nabla \phi_{j,E}(s(k)) \phi_{i,E}(s(k))$
      $R_E(i,j) = R_E(i,j) + w(k) det(B_E) \phi_{i,E}(s(k)) \phi_{j,E}(s(k))$
    end
    $b_E(i) = b_E(i) + w(k) det(B_E) f(x) \phi_{i,E}(s(k))$
  end
end

A partition of **Algorithm 2** is given below MATLAB routine;

```
%Get quadrature points and weights on reference triangle
```

```
[nodes_ref,wgt]=quadrature(10);
%Compute values and derivatives of basis functions and determinant
%and compute  global coordinates of quadarature point
[val_basis,der_basisx,der_basisy,determ,xx]=elem_basis(mesh,degree,nodes_ref);
% weights * determ and compute the transpose
vol=permute(wgt.*determ,[2,1,3]);
for i=1:Nloc
 for j=1:Nloc
   % Diffusion part
   Dloc(i,j,:) =multiprod(vol,(der_basisx(:,j,:).*der_basisx(:,i,:)...
                          +der_basisy(:,j,:).*der_basisy(:,i,:)).*diff);
   % Convection part
   Cloc(i,j,:) = multiprod(vol,(adv1.*der_basisx(:,j,:).*val_basis(:,i,:)...
                          +adv2.*der_basisy(:,j,:).*val_basis(:,i,:)));
   % Reaction part
   Rloc(i,j,:) =multiprod(vol,(val_basis(:,j,:).*val_basis(:,i,:)).*reac);
 end
 %Right-side
 Floc(i,1,:) =multiprod(vol,(val_basis(:,i,:).*source));
end
```

The above routine calls other routines such as: **quadrature** which initializes the arrays **nodes_ref** and **wgt** containing the coordinates of the quadrature points and the weights of the quadrature points, respectively and **elem_basis** which computes the values and global derivatives of the basis functions and the determinant of transformation matrix between reference element and physical element, as well as global coordinates of points on the reference element.

A.5.2 Face Contributions

Assuming the edges from 1 to N_{face} and face $k \in E_k^1 \cap E_k^2$, we assemble the local matrices $D_e^{i,j}$ and $C_e^{i,j}$ for $1 \leq i, j, \leq 2$.

Algorithm 3: Face Contributions [87]
loop over the edges: for k=1 to N_{face} do
  get face neighbors E_k^1 and E_k^2
  if face is an interior face do
  compute local matrices D_k^{11}, D_k^{22}, D_k^{12}, D_k^{21}

compute local matrices C_k^{11}, C_k^{22}, C_k^{12}, C_k^{21}
assemble D_k^{11} and C_k^{11} contributions:
for i=1 to N_{loc} do
$ie = i + (E_k^1 - 1)N_{loc}$
for j=1 to N_{loc} do
$je = j + (E_k^1 - 1)N_{loc}$
$D_{global}(ie, je) = D_{global}(ie, je) + D_k^{11}(i, j)$
$C_{global}(ie, je) = C_{global}(ie, je) + C_k^{11}(i, j)$
end
end
assemble D_k^{22} and C_k^{22} contributions:
assemble D_k^{12} and C_k^{12} contributions:
assemble D_k^{21} and C_k^{21} contributions:
for i=1 to N_{loc} do
$ie = i + (E_k^2 - 1)N_{loc}$
for j=1 to N_{loc} do
$je = j + (E_k^1 - 1)N_{loc}$
$D_{global}(ie, je) = D_{global}(ie, je) + D_k^{21}(i, j)$
$C_{global}(ie, je) = C_{global}(ie, je) + C_k^{21}(i, j)$
end
end
else if face is a boundary face do
compute local matrices D_k^{11} and C_k^{11}
compute local right-hand side b_k
assemble D_k^{11} contributions:
for i=1 to N_{loc} do
$ie = i + (E_k^1 - 1)N_{loc}$
for j=1 to N_{loc} do
$je = j + (E_k^1 - 1)N_{loc}$
$D_{global}(ie, je) = D_{global}(ie, je) + D_k^{11}(i, j)$
$C_{global}(ie, je) = C_{global}(ie, je) + C_k^{11}(i, j)$
end
$b_{global}(ie) = b_{global}(ie) + b_k(i)$
end
end

In **Algorithm 3**, we compute the local matrices on the interior ($e \in \Gamma_h^0$) and boundary edges ($e \in \Gamma_h^\partial$) calling the following routines:

```
% Compute local  matrices caused by interior edges
[B11,B22,B12,B21,C11,C22,C12,C21]=localmat_face(mesh,fdiff,fadv,penalty,eps,degree);
% Compute local  matrices and right-hand side vector  caused by boundary edges
```

```
[B11B,FlocB,C11B]=localmat_bdyface(mesh,fdiff,fadv,fexact,penalty,eps,degree);
```

A.5.2.1 Local matrices on faces

For the interior edges i.e., $e \in \Gamma_h$, the terms involving integrals on e coming from diffusion term are defined by

$$T_D = -\int_e \{\epsilon \nabla u_h \cdot n_e\}[v] + \kappa \int_e \{\epsilon \nabla v \cdot n_e\}[u_h] + \frac{\sigma}{|e|^{\beta_0}} \int_e [u_h][v].$$

Expanding the averages and jumps, we obtain

$$T = D_e^{11} + D_e^{22} + D_e^{12} + D_e^{21},$$

where the term d_e^{11} (resp.,E_e^2) corresponds to the interactions of the local basis of the neighboring element E_e^1 (resp.,E_e^2) with itself and the term d_e^{12} corresponds to the interaction of the local basis of the neighboring elements E_e^1 (resp.,E_e^2) with the elements E_e^2 (resp.,E_e^1).

$$\begin{aligned}
(D_e^{11})_{i,j} &= -\frac{1}{2}\int_e \epsilon \nabla \phi_{j,E_e^1} \cdot n_e \phi_{i,E_e^1}\, ds + \frac{\kappa}{2}\int_e \epsilon \nabla \phi_{i,E_e^1} \cdot n_e \phi_{j,E_e^1}\, ds + \frac{\sigma}{|e|^{\beta_0}}\int_e \phi_{j,E_e^1}\phi_{i,E_e^1}\, ds,\\
(D_e^{22})_{i,j} &= \frac{1}{2}\int_e \epsilon \nabla \phi_{j,E_e^2} \cdot n_e \phi_{i,E_e^2}\, ds - \frac{\kappa}{2}\int_e \epsilon \nabla \phi_{i,E_e^2} \cdot n_e \phi_{j,E_e^2}\, ds + \frac{\sigma}{|e|^{\beta_0}}\int_e \phi_{j,E_e^2}\phi_{i,E_e^2}\, ds,\\
(D_e^{12})_{i,j} &= -\frac{1}{2}\int_e \epsilon \nabla \phi_{j,E_e^2} \cdot n_e \phi_{i,E_e^1}\, ds - \frac{\kappa}{2}\int_e \epsilon \nabla \phi_{i,E_e^1} \cdot n_e \phi_{j,E_e^2}\, ds - \frac{\sigma}{|e|^{\beta_0}}\int_e \phi_{j,E_e^2}\phi_{i,E_e^1}\, ds,\\
(D_e^{21})_{i,j} &= \frac{1}{2}\int_e \epsilon \nabla \phi_{j,E_e^1} \cdot n_e \phi_{i,E_e^2}\, ds + \frac{\kappa}{2}\int_e \epsilon \nabla \phi_{i,E_e^2} \cdot n_e \phi_{j,E_e^1}\, ds - \frac{\sigma}{|e|^{\beta_0}}\int_e \phi_{j,E_e^1}\phi_{i,E_e^2}\, ds.
\end{aligned}$$

For $e \in \Gamma_h$, the terms involving integrals on e coming from convection term are defined by

$$T_C = \int_e |\beta \cdot n)|\,(y^+ - y^-)v^+ ds = c_e^{11} + c_e^{22} + c_e^{12} + c_e^{21}.$$

We define y^+ and y^- using the upwind discretization [70, 86] such that:

$$y^+ = \begin{cases} y|_{E^1}, & \text{if } \beta \cdot n_e < 0, \\ y|_{E^2}, & \text{if } \beta \cdot n_e \geq 0, \end{cases} \qquad y^- = \begin{cases} y|_{E^2}, & \text{if } \beta \cdot n_e < 0, \\ y|_{E^1}, & \text{if } \beta \cdot n_e \geq 0. \end{cases}$$

Then, the terms coming from the convection term are defined such that:

$\forall e \in \Gamma_h$ satisfying $\beta \cdot n_e < 0$,

$$(C_e^{11})_{i,j} = \int_e |\beta \cdot n|\, \phi_{j,E_e^1}\phi_{i,E_e^1}, \qquad (C_e^{12})_{i,j} = -\int_e |\beta \cdot n|\, \phi_{j,E_e^2}\phi_{i,E_e^1}$$

and $\forall e \in \Gamma_h$ satisfying $\beta \cdot n_e \geq 0$,

$$(C_e^{22})_{i,j} = \int_e |\beta \cdot n|\, \phi_{j,E_e^2}\phi_{i,E_e^2}, \qquad (C_e^{21})_{i,j} = -\int_e |\beta \cdot n|\, \phi_{j,E_e^1}\phi_{i,E_e^2}.$$

Algorithm 4: Computing local contributions from interior edges [87]

initialize $D_e^{11} = D_e^{22} = D_e^{12} = D_e^{21} = 0$

initialize $C_e^{11} = C_e^{22} = C_e^{12} = C_e^{21} = 0$

initialize parameters ϵ and σ_e^0

initialize the quadrature weights w and the points s on $[-1, 1]$

compute edge length $|e|$, normal vector n_e

get face neighbors E_e^1 and E_e^2

loop over quadrature points: for k=1 to N_G do

compute local coordinates $ss1$ on E_e^1 and $ss2$ on E_e^2 of quadrature point $s(k)$

for i=1 to N_{loc} do

compute values of basis functions $\phi_{i,E_e^1}(s(k))$ and $\phi_{i,E_e^2}(ss1)$

compute derivatives of basis functions $\nabla\phi_{i,E_e^1}(s(k))$ and $\nabla\phi_{i,E_e^2}(ss2)$

end

for i=1 to N_{loc} do

for j=1 to N_{loc} do

$$
\begin{aligned}
D_e^{11}(i,j) &= D_e^{11}(i,j) - 0.5w(k)|e|\phi_{i,E_e^1}(s(k))(\nabla\phi_{j,E_e^1}(s(k))\cdot n_e)\\
D_e^{11}(i,j) &= D_e^{11}(i,j) + 0.5\epsilon w(k)|e|\phi_{j,E_e^1}(s(k))(\nabla\phi_{i,E_e^1}(s(k))\cdot n_e)\\
D_e^{11}(i,j) &= D_e^{11}(i,j) - \frac{\sigma_e^0}{|e|^{\beta_0}}w(k)|e|\phi_{i,E_e^1}(s(k))\phi_{j,E_e^1}(s(k))
\end{aligned}
$$

$$\ldots$$

$$
\begin{aligned}
D_e^{21}(i,j) &= D_e^{21}(i,j) + 0.5w(k)|e|\phi_{i,E_e^2}(s(k))(\nabla\phi_{j,E_e^1}(s(k))\cdot n_e)\\
D_e^{21}(i,j) &= D_e^{21}(i,j) + 0.5\epsilon w(k)|e|\phi_{j,E_e^1}(s(k))(\nabla\phi_{i,E_e^2}(s(k))\cdot n_e)\\
D_e^{21}(i,j) &= D_e^{21}(i,j) - \frac{\sigma_e^0}{|e|^{\beta_0}}w(k)|e|\phi_{i,E_e^2}(s(k))\phi_{j,E_e^1}(s(k))
\end{aligned}
$$

end

end

if face is influx face do

for i=1 to N_{loc} do

for j=1 to N_{loc} do

$$
\begin{aligned}
C_e^{11}(i,j) &= C_e^{11}(i,j) + w(k)|e|\,|\beta\cdot n|\,\phi_{i,E_e^1}(s(k))\phi_{j,E_e^1}(s(k))\\
C_e^{12}(i,j) &= C_e^{12}(i,j) + w(k)|e|\,|\beta\cdot n|\,\phi_{i,E_e^2}(s(k))\phi_{j,E_e^1}(s(k))
\end{aligned}
$$

end

end

else if face is outflux face do

for i=1 to N_{loc} do

for j=1 to N_{loc} do

$$C_e^{22}(i,j) = C_e^{22}(i,j) + w(k)|e|\,|\beta \cdot n|\,\phi_{i,E_e^2}(s(k))\phi_{j,E_e^2}(s(k))$$
$$C_e^{21}(i,j) = C_e^{21}(i,j) + w(k)|e|\,|\beta \cdot n|\,\phi_{i,E_e^1}(s(k))\phi_{j,E_e^2}(s(k))$$

end

end

end

end

This algorithm which computes the local matrices obtained by the integration over interior edges is implemented at the routine called **localmat_face**. A partition of this Matlab routine is given below:

```
%Initialize the quadrature weights and points for edges
Nqu=12;  [nodes_ref,wge]=get_quadrature_segment(Nqu);
%Compute normal vector to edges E1
[normal_vec,area]=getNormal(Equation.mesh,E1,iedge);
normal_vec=permute(repmat(normal_vec,[1,1,Nqu]),[3,2,1]);
normal1=normal_vec(:,1,:);  normal2=normal_vec(:,2,:);
%Compute values and derivatives of basis functions and determinant
%and compute  global coordinates of quadarature point on E1 on E2
[val_basis1,der_basis1x,der_basis1y,~,xx]=elem_basisf(mesh,degree,E1,s1);
[val_basis2,der_basis2x,der_basis2y,~,~]=elem_basisf(mesh,degree,E2,s2);
%Define inflow and outflow edges
c=(adv1.*normal1+adv2.*normal2); a=find(c(1,:,:)<0); b=find(c(1,:,:)>=0);
 for i=1:Nloc
  for j=1:Nloc
  %Compute the entries of local matrix D11
  T11=(normal1.*der_basis1x(:,j,:)+normal2.*der_basis1y(:,j,:))
      .*val_basis1(:,i,:).*(-0.5*diff)...
      +(normal1.*der_basis1x(:,i,:)+normal2.*der_basis1y(:,i,:))
      .*val_basis1(:,j,:).*(eps*(0.5)*diff)...
      +(penalty.*val_basis1(:,i,:).*val_basis1(:,j,:));
  D11(i,j,:) = multiprod(area,T11);
```

```
  %Compute the entries of local matrix C11
  T1=(abs(c(:,:,a)).*val_basis1(:,j,a).*val_basis1(:,i,a));
  C11(i,j,a)=multiprod(area(:,:,a),T1);
   end
end
```

The above routine calls other routines such as: **get_quadrature_segment** which initializes the weights and nodes of the Gauss quadrature nodes on the interval [-1,1], **getNormal** which returns the fixed normal vector to the edge and the length of the edge and **elem_basisf** which computes the values and global derivatives of the basis functions and the determinant of transformation matrix between reference element and physical element, as well as global coordinates of points on edge of the reference element. This routine is different from the routine **elem_basis** since the quadrature points on the reference edges are different for each edge.

For the Dirichlet boundary edges, i.e., $e \in \Gamma_D$, the following local matrices D_e^{11} and C_e^{11}, of the diffusion and convection terms, respectively, are created:

$$\begin{aligned}(D_e^{11})_{i,j} &= -\int_e \epsilon \nabla \phi_{j,E_e^1} \cdot n_e \phi_{i,E_e^1}\, ds + \kappa \int_e \epsilon \nabla \phi_{i,E_e^1} \cdot n_e \phi_{j,E_e^1}\, ds + \frac{\sigma}{|e|\beta_0} \int_e \phi_{j,E_e^1} \phi_{i,E_e^1}\, ds, \\ (C_e^{11})_{i,j} &= \int_e y^+ \upsilon^+ |n \cdot \beta|\, ds\end{aligned}$$

and the local right-hand side b_e is

$$(b_e)_i = \int_e (\kappa \epsilon \nabla \phi_{i,E_e^1} \cdot n_e + \frac{\sigma}{|e|_{\beta^0}} \phi_{i,E_e^1}) g_D\, ds + \int_{\Gamma^-} |\beta \cdot n|\, g_D \upsilon^+ ds.$$

Note that if the edge $e \in \Gamma_N$, no local matrix is created, but the following right-hand side is defined:

$$(b_e)_i = \int_e \phi_{i,E_e^1} g_N\, ds.$$

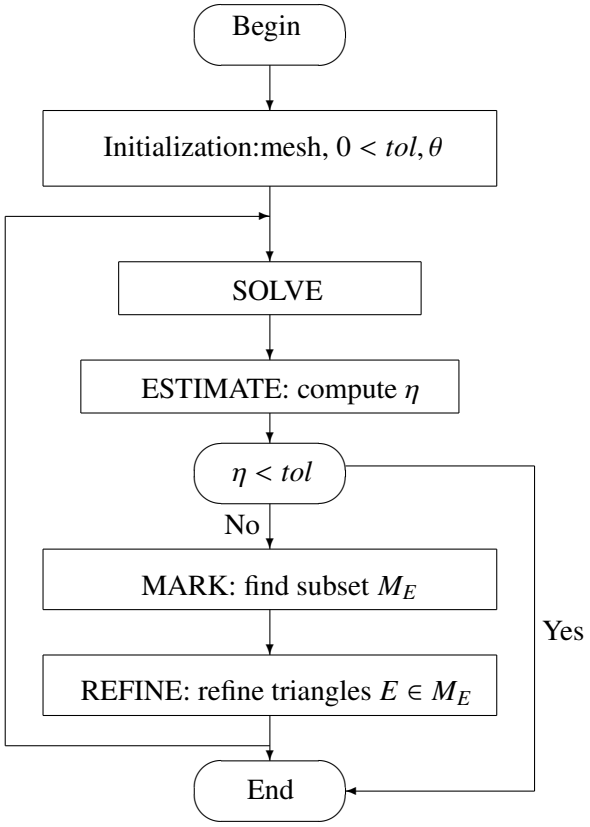


Figure A.3: Adaptive procedure.

A.6 Adaptivity Procedure

In this section, we will desribe the MATLAB routines of adaptivity procedure given in Figure A.3.

A.6.1 Estimation

We have two steps to compute error estimator given in Section 4.1: element and edge residuals. The element residual part is given by the following script:

```
% Compute values of basis functions, their  derivatives and determinant
% and compute  global coordinates of quadrature point
[val_basis,der_basisx,der_basisy,determ,xx]=elem_basis(mesh,degree,nodes_ref);
% Compute second derivative of basis functions
[der_basisxx,der_basisyy]=SecondDer_basis(mesh,degree,nodes_ref);
% Evaluate yex  at the quadrature points
```

```
source=feval(fsource,fdiff,fadv_state,freact_state,xx(:,1,:),xx(:,2,:));
% weights * determ and compute the transpose
vol = permute(wgt.*determ,[2 1 3]);
% Compute the diameter of each triangle
diam=repmat(reshape(getDiameter(mesh),1,1,Nel),[size(nodes_ref,1) 1 1]);
yval=multiprod(val_basis,yy);        % value of numerical solution  of state
ygradx=multiprod(der_basisx,yy);     % value of first derivative  wrt x
ygrady=multiprod(der_basisy,yy);     % value of first derivative  wrt y
ygradxx=multiprod(der_basisxx,yy);   % value of second derivative wrt x
ygradyy=multiprod(der_basisyy,yy);   % value of second derivative wrt y
uval=multiprod(val_basis,uu);        % value of numerical solution  of control
% Define the parameter for the triangles
if (tau==0) pa=diff.^(-0.5).*diam;
else  pa=min(diff.^(-0.5).*diam, tau^(-0.5));
end
% Compute estimator caused by triangles
T= (pa.*(source+uval+diff.*(ygradxx+ygradyy)-adv1.*ygradx-adv2.*ygrady-reac.*yval)).^2;
errInd_tri=squeeze(multiprod(vol,T));
```

Secondly, we implement the edge residual at the following script:

```
% ***************Interior Edges.............
iedge=mesh.intEdges;      % Get interior edges
Ned=length(iedge);        % number of interior edges
% Get neighbors of interior edges
edge=mesh.EdgeEls(iedge,:); E1=edge(:,1);   E2=edge(:,2);
% Compute normal vector to edges E1 and length of edges
[normal_vec,area]=getNormal(mesh,E1,iedge);
normal_vec=permute(repmat(normal_vec,[1,1,Nqu]),[3,2,1]);
normal1=normal_vec(:,1,:);   normal2=normal_vec(:,2,:);
% Compute local coordinates of quadrature points on E1 and E2
s1=loc_coor_quad(mesh,iedge,E1,nodes_ref);
s2=loc_coor_quad(mesh,iedge,E2,nodes_ref);
% Compute values of  basis functions, their derivatives and determinant
% and compute  global coordinates of quadarature point on E1 on E2
[val_basis1,der_basis1x,der_basis1y,~,xx1]=elem_basisf(mesh,degree,E1,s1);
[val_basis2,der_basis2x,der_basis2y,~,~]=elem_basisf(mesh,degree,E2,s2);
% Penalty parameter
penalty_int=permute(repmat(penalty*ones(Ned,1),[1,1,Nqu]),[3,2,1]);
% weights * area and compute the transpose
area=permute(0.5*permute(repmat(area,[1,1,Nqu]),[3,2,1]).*wge,[2,1,3]);
```

```
% Compute the values of numerical solution and its derivative on edges of E1
y1=multiprod(val_basis1,yy(:,:,E1));
ygrad1x = multiprod(der_basis1x,yy(:,:,E1));
ygrad1y = multiprod(der_basis1y,yy(:,:,E1));
% Compute the values of numerical solution and its derivative on edges of E2
y2=multiprod(val_basis2,yy(:,:,E2));
ygrad2x = multiprod(der_basis2x,yy(:,:,E2));
ygrad2y = multiprod(der_basis2y,yy(:,:,E2));
% Define the parameter for edges
if (tau==0) pe=diff.^(-0.5).*length_edge;
else pe=min(diff.^(-0.5).*length_edge,tau^(-0.5));
end
% Compute error indicator caused by interior edge
T=0.5*(diff.^(-0.5)).*pe.*(diff.*(ygrad1x-ygrad2x).*normal1 + ...
       diff.*(ygrad1y-ygrad2y).*normal2 ).^2 ...
      +0.5*((penalty_int.*diff./length_edge)+tau*length_edge+length_edge./diff).*(y1-y2).^2;
InteriorEdge_indicator=squeeze(multiprod(area,T));
% ******Boundary edges*******************************
Dbedge=mesh.DbdEdges;       % get boundary edges
DNedb=length(Dbedge);       % number of boundary edges
             .....................
% Compute  error indicator caused by boundary edges
BdyEdge_indicator=multiprod(area,((penalty_bdy.*diff./length_edge)+...
                       tau*length_edge+length_edge./diff).*(yex-y1).^2);
BdyEdge_indicator=squeeze(BdyEdge_indicator);
```

The error estimator of control is equal to zero for unconstraint case, whereas it is different from zero for control constrained case. Then, it is computed such as:

```
T=omega^2*diam.^2.*((uval-uaval).*(ubval-uval)./(h+(uval-uaval).*...
  (ubval-uval))).^2.*((ugradx+pgradx).^2+(ugrady+pgrady).^2);
errInd_tri=multiprod(vol,T);
errInd_tri=squeeze(errInd_tri);
```

A.6.2 Marking

The Matlab implementation of marking step in adaptive procedure is such that [32]:

```
function markedElem= mark(mesh,etaT,theta)
NT = size(mesh.Elements,1); isMark = false(NT,1);
```

```
[sortedEta,idx] = sort(etaT,'descend');
x = cumsum(sortedEta);
isMark(idx(x < theta* x(NT))) = 1;
isMark(idx(1)) = 1;
markedElem = uint32(find(isMark==true));
```

A.6.3 Refinement

In the REFINEMENT step, the marked elements are refined by longest edge bisection, whereas the elements of the marked edges are refined by bisection [31, 32]. In a loop for **Elements** matrix, we first check if any triangle's base is marked. If so we divide it and then check the other two edges. If one of them is marked, we divide children elements with suitable order.

```
% Refine marked edges for each triangle
numnew = 2*sum(marker~=0); % number of new Elementsents need to be added
mesh.Elements = [mesh.Elements; zeros(numnew,3)];
inew = NT + 1; % index for current new added right child
for t = 1:NT
    base = d2p(mesh.Elements(t,2),mesh.Elements(t,3));
    if (marker(base)>0)
        p = [mesh.Elements(t,:), marker(base)];
        % Case 1: divide the current marked triangle
        mesh.Elements(t,:)  = [p(4),p(1),p(2)]; % t is always a left child
        mesh.Elements(inew,:) = [p(4),p(3),p(1)]; % new is a right child
        inew = inew + 1;
        % Case 2: divide the right child, different, careful!!!
        right = d2p(p(3),p(1));
        if (marker(right)>0)
            mesh.Elements(inew-1,:) = [marker(right),p(4),p(3)];
            mesh.Elements(inew,:)   = [marker(right),p(1),p(4)];
            inew = inew + 1;
        end
        % Case 3: divide the left child, similar to the case 1.
        left = d2p(p(1),p(2));
        if (marker(left)>0)
            mesh.Elements(t,:) = [marker(left),p(4),p(1)];
            mesh.Elements(inew,:) = [marker(left),p(2),p(4)];
            inew = inew + 1;
        end
    end % end of refinement of one Elementsent
```

```
end % end of for loop on all Elementsents
% delete possible empty entries
mesh.Elements = mesh.Elements(1:inew-1,:);
```

VITA

PERSONAL INFORMATION

Surname, Name	:	Yücel, Hamdullah
Nationality	:	Turkish (TC)
Date and Place of Birth	:	October 12, 1984, Çorum
Phone	:	+90 312 210 53 58
e@mail	:	hayucel@metu.edu.tr

ACADEMIC DEGREES

Ph.D. Scientific Computing Program, Institute of Applied Mathematics,

Middle East Technical University, Ankara, TURKEY, 2012 March

Supervisor : Prof. Dr. Bülent Karasözen

Thesis Title : Adaptive Discontinuous Galerkin Methods For

Convection Dominated Optimal Control Problems

B.S. Department of Mathematics,

Middle East Technical University, Ankara, TURKEY, 2007 June

RESEARCH VISITS

September 2010-June 2011 Department of Computational and Applied Mathematics

	Rice University, Houston, USA,
	supervised by Prof. Dr. Matthias Heinkenschloss
October 2008-December 2008	Department of Mathematics
	Technische Universität Darmstadt, Germany,
	supervised by Prof. Dr. Stefan Ulbrich

EMPLOYMENT

September 2007 -	Research Assistant, Department of Mathematics,
	Middle East Technical University, Ankara, TURKEY
April 2007-June 2007	Student Assistant, Institute of Applied Mathematics,
	Middle East Technical University, Ankara, TURKEY

SCHOLARSHIPS

October 2007 -	Doctorate Scholarship
	Turkish Scientific and Technical Research Council (TÜBİTAK)
September 2010-June 2011	International Doctoral Research Fellowship Programme
	Turkish Scientific and Technical Research Council (TÜBİTAK)
October-December 2008	German Academic Exchange Service (DAAD) Scholarship

FOREIGN LANGUAGES

Turkish (native), English (fluently), German (B1)

PROGRAMMING LANGUAGES

Advanced : C, C++, MATLAB

Basic : Java, HTML, SQL, FORTRAN

PUBLICATIONS

A. Papers submitted to International Journals:

A1. H. Yücel, M. Heikenschloss, and B. Karasözen, *An Adaptive discontinuous Galerkin method for convection dominated distributed optimal control problems*, Applied Numerical Mathematics, 2012.

A2. Z. K. Seymen, H. Yücel, and B. Karasözen, *Distributed Optimal Control of Time-dependent Diffusion-Convection-Reaction Equations Using Space-Time Discretization*, Journal of Computational and Applied Mathematics, 2011.

B. Papers in Preprints:

B1. H. Yücel, M. Heikenschloss, and B. Karasözen, *A posteriori error estimates of constrained optimal control problem governed by convection diffusion equations using symmetric interior penalty Galerkin method*, Institute of Applied Mathematics, Middle East Technical University, 2012.

C. Papers submitted to International Conference Proceedings:

C1. H. Yücel, M. Heikenschloss, and B. Karasözen, *Distributed Optimal Control of Diffusion-Convection-Reaction Equations Using Discontinuous Galerkin Methods*, The Proceedings of ENUMATH 2011 Conference, Leicester, England, 5-9 September 2011, (*To Appear*).

INTERNATIONAL SCIENTIFIC MEETINGS

A. Presentations in International Scientific Meetings:

A1. H. Yücel, M. Heikenschloss, and B. Karasözen, *Distributed Optimal Control of Diffusion-Convection-Reaction Equations Using Discontinuous Galerkin Methods*, ENUMATH 2011 Conference, Leicester, England 5-9 September 2011.

A2. H. Yücel, F. Yılmaz, Z. Seymen, and B. Karasözen, *Distributed Optimal Control of unsteady Burgers and Convection-Diffusion-Reaction Equations using COMSOL Multiphysics*,

Computational techniques for optimization problems subject to time-dependent PDEs, Brighton, England, 14-16 December 2009.

B. Participation in International Scientific Meetings:

B1. Finite Element Rodeo 2011, College Station, Texas, USA, 25-26 February, 2011.

B2. Winterschool on Hierarchical Matrices, Max-Planck-Institute for Mathematics in the Sciences, Leipzig, Germany, 2-6 March 2009.