MULTIGRID METHODS FOR OPTIMAL CONTROL PROBLEMS GOVERNED BY CONVECTION-DIFFUSION EQUATIONS

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

BY

ÖZGÜN MURAT ARSLANTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

JUNE 2015

Approval of the thesis:

MULTIGRID METHODS FOR OPTIMAL CONTROL PROBLEMS GOVERNED BY CONVECTION-DIFFUSION EQUATIONS

submitted by **ÖZGÜN MURAT ARSLANTAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing, Middle East Technical University** by,

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

Assoc. Prof. Dr. Ömür Uğur
Head of Department, **Scientific Computing** ______________

Prof. Dr. Bülent Karasözen
Supervisor, **Scientific Computing, METU** ______________

Dr. Hamdullah Yücel
Co-supervisor, **Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg** ______________

Examining Committee Members:

Prof. Dr. Bülent Karasözen
Scientific Computing, METU ______________

Prof. Dr. Gerhard Wilhelm Weber
Scientific Computing, METU ______________

Assoc. Prof. Dr. Ömür Uğur
Scientific Computing, METU ______________

Assist. Prof. Fikriye Nuray Yılmaz
Department of Mathematics, Gazi University ______________

Dr. Hamdullah Yücel
Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg ______________

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:	ÖZGÜN MURAT ARSLANTAŞ
Signature :	

ABSTRACT

MULTIGRID METHODS FOR OPTIMAL CONTROL PROBLEMS GOVERNED BY CONVECTION-DIFFUSION EQUATIONS

Arslantaş, Özgün Murat
M.S., Department of Scientific Computing
Supervisor : Prof. Dr. Bülent Karasözen
Co-Supervisor : Dr. Hamdullah Yücel

June 2015, 46 pages

Linear-quadratic optimal control problems governed by partial differential equations proved themselves important through their use in many real life applications. In order to solve the large scale linear system of equations that results from optimality conditions of the optimization problem, efficient solvers are required. For this purpose, multigrid methods, with an ordering technique to deal with the dominating convection, can be good candidates. This thesis investigates an application of the multigrid methods for the linear-quadratic optimal control problems governed by convection-diffusion equation, discretized by a discontinuous Galerkin method, namely, symmetric interior penalty Galerkin (SIPG) method. Further, an ordering technique called Downwind Numbering is proposed to reduce the number of iteration in multigrid approach.

Keywords : Optimal control problems, discontinuous Galerkin method, convection-diffusion equations, multigrid methods, ordering

ÖZ

KISITLARI KONVEKSİYON-DİFÜZYON EŞİTLİKLERİ OLAN ENİYİLEMELİ KONTROL PROBLEMLERİ İÇİN ÇOKLU AĞ YÖNTEMLERİ

Arslantaş, Özgün Murat
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. Bülent Karasözen
Ortak Tez Yöneticisi : Dr. Hamdullah Yücel

Haziran 2015, 46 sayfa

Kısıtları kısmi türevli diferansiyel denklemlerden oluşan ikinci dereceden doğrusal eniyilemeli kontrol problemleri, kendi önemlerini gerçek hayat uygulamalarındaki kullanımları yoluyla ispatladılar. Optimizasyon problemininin eniyileme koşullarından elde edilen büyük ölçekli lineer eşitlikler sistemini çözerken verimli çözüm yöntemlerinin kullanılması gerekmektedir. Bu amaç için, ağır basan konveksiyon teriminin üstesinden gelmesi amacıyla bir yeniden sıralama tekniği ile kullanılan çoklu ağ yöntemleri, iyi bir aday olabilirler. Bu tez, çoklu ağ yöntemlerinin, bir süreksiz Galerkin yöntemi olan simetrik içten cezalandırma Galerkin yöntemi ile ayrıklaştırılan kısıtları kısmi türevli diferansiyel denklemlerden oluşan ikinci dereceden doğrusal eniyilemeli kontrol problemlerine bir uygulamasını incelemektedir. İlaveten, bir sıralama tekniği olan Akış yönü Sıralaması yöntemini, çoklu ağ yaklaşımının tekrarlama sayısını düşürmesi için öneriyoruz.

Anahtar Kelimeler : Eniyilemeli kontrol problemleri, süreksiz Galerkin yöntemi, konveksiyon-difüzyon eşitlikleri, çoklu ağ yöntemi, sıralama

To My Family

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Bülent Karasözen for his valuable guidance, the opportunities he provided, his kindness and understanding.

I would like to thank my co-supervisor Dr. Hamdullah Yücel for his guidance and advices. Without him and his unlimited support, I would have not been able to succeed in completing my studies.

I would like to thank Prof. Dr. Peter Benner for providing me the possibility to research at Max Planck Institute for Dynamics of Complex Technical Systems, and the members of the institute for kindly hosting me for three months.

I would like to thank Prof. Dr. Gerhard-Wilhelm Weber for his endless kindness and proofreading this thesis.

I would like to thank Marie Nguyen-Khuong, Dr. Lilí Esmeralda Gallo Ramirez, Pawan Goyal, Dr. Terry Nguyen-Khuong, and Dr. Mislav Novokmet for their precious company and support during my days in Magdeburg.

I would like to thank Selin Özokcu for her continuous support, invaluable company and delicious cooking.

Last but not least, I would like to thank my parents for their unending support.

TABLE OF CONTENTS

CHAPTERS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In recent years, linear-quadratic optimal control problems proved themselves important through their use in many real life applications such as the optimal control of systems [39], the shape optimization of technological devices [58] and flow control problems [30, 37, 62]. In order to solve the large scale linear system of equations that results from the optimality conditions of such kind of problems, it is preferable to use efficient iterative methods instead of direct methods. For this purpose, we consider an application of the multigrid methods to the linear-quadratic optimal control problems of the form

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

subject to

$$\begin{aligned} -\epsilon\Delta y(x) + \beta(x)\cdot\nabla y(x) + r(x)y(x) &= f(x) + u(x), && x \in \Omega, \\ y(x) &= g_D(x), && x \in \Gamma^D, \\ \epsilon\nabla y(x)\cdot\mathbf{n}(x) &= g_N(x), && x \in \Gamma^N. \end{aligned} \tag{1.2}$$

Simple iterative methods such as Gauss-Seidel method and Jacobi method, fail at smoothing the low-frequency error components of the approximation even though they are are known to be effective for smoothing the high-frequency error components in the approximation. The elegant idea of multigrid methods is to overcome this obstacle by introducing a sequence of meshes [12, 17, 19, 22, 23, 24, 38, 41, 44, 51, 59, 74, 76]. By doing so, multigrid methods aim to effectively smooth the high-frequency components of the approximation on a mesh that correspond to low-frequency components of the approximation on a coarser mesh. Remaining main ingredients of a multigrid method are the smoothing operators that are used to remove the high frequency components of the error on the mesh, and the transfer operators that provide the connection between coarse and fine meshes.

The numerical solution of the convection-diffusion PDEs is particularly challenging and requires special numerical techniques, which take into account the structure of the convection; for which purpose there mainly exist two approaches. The first approach mainly considers modifying the transfer operators according to the convection [15], whereas the second approach, which we also focus on in this thesis, considers modifying the smoothing operators, e.g. Gauss-Seidel or Jacobi, according to the convection. The modification mainly depends on the ordering of the vertices or elements in the mesh. However, for PDEs with a general convection term it is difficult to determine appropriate smoothers and vertex orderings [11, 40, 42, 52, 53, 60, 68].

In the optimal control context multigrid solvers are applied successfully to solve elliptic optimal control problems in [7, 14, 15, 16, 18, 19, 36, 49, 72], and parabolic optimal control problems in [12]. One reason why the solution of optimization problems governed by convection-diffusion PDEs provides additional challenges is that the optimality system involves two PDEs, the state equation with convection β and the so-called adjoint equation with convection $-\beta$. 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. Further, when node ordering strategies are applied to the state PDE, the node ordering of the adjoint PDE would be the reverse of the state PDE.

In this thesis, we investigate the analysis and application of multigrid methods for linear-quadratic convection dominated optimal control problems using a discontinuous Galerkin (DG) method. Especially, we choose a symmetric interior penalty Galerkin (SIPG) method due to the its symmetric property [46, 69]. For convection problems, DG methods produce stable discretizations without the need for stabilization strategies, they work on non-conforming meshes, and they allow different orders of approximation to be used on each element in a very straightforward manner. These are some of the features that motivate us to use DG methods for the numerical solution of optimization problems governed by convection dominated PDEs. DG methods have been successfully applied to a variety of PDEs, in particular convection-diffusion PDEs in [5, 9, 38, 48], and to optimal control problems governed by convection-diffusion PDEs in [2, 75, 80, 82, 83, 84, 85]. Further, multigrid methods using DG discretizations have been applied for single elliptic equations in [38, 61, 65, 77] and for Navier-Stokes equations in [50].

The rest of the thesis is organized as follows: In Chapter 2 we introduce discontinuous Galerkin methods, especially the SIPG method, for the convection-diffusion equations. Multigrid algorithm, in terms of the general context, and then transfer operators and smoothing techniques are described in Chapter 3. In addition, an ordering technique called Downwind Numbering algorithm is given to handle the difficulty of the convection term. Further, some numerical results are demonstrated for single convection-diffusion PDEs. In Chapter 4, we solve the optimal control problem governed by convection-diffusion PDE, discretized by the SIPG method by applying multigrid algorithm, and then give some numerical results. Throughout this thesis, numerical results are obtained using the software MATLAB®.

CHAPTER 2

DISCONTINUOUS GALERKIN METHODS

Discontinuous Galerkin (DG) methods were first introduced in 1973 by Reed and Hill [66], when they used a method to solve first-order hyperbolic problems. Later on, towards the end of seventies, Wheeler [79] and Douglas & Dupont [35] used DG methods for elliptic and parabolic problems. As the method proved itself efficient and began to be used widely, works using DG have multiplied; among the works on DG methods, Arnold, Brezzi, Cockburn & Marini [4], Bassi & Rebay [8], Cockburn & Shu [28] and Peraire & Persson [63] can be seen as the cornerstones for their contributions in their respective areas of study. Up to the present day, various DG methods have been introduced and used for the elliptic problems [6, 25, 70] and for convection-diffusion problems [5, 9, 38, 48].

As the DG method was introduced to the literature, numerical methods such as the finite difference method (FDM), the finite volume method (FVM) and the finite element method (FEM) were being used to find approximate solutions to the boundary value problems for partial differential equations (PDEs). Having advantages on their own, each one of these methods had some drawbacks also. The FDM, which is the historically oldest method, excels for its simplicity, i.e., the discretization of general problems and operators is mostly intuitive, leading to efficient schemes. Additionally, when needed, time stepping methods can be chosen freely via the flexibility that comes from the explicit semi-discrete form that the FDM yields. However, the drawback of the FDM reveals when faced with complex geometries, that is, the method cannot cope with the local order and grid size changes while reflecting these local features to the general solution. The method that overcomes this obstacles is the FVM. The scheme yielded by the FVM has no conditions about the grid structure as it is purely local. This property allows the use of elements with different sizes in a grid. Even though the use of elements with different sizes is allowed, a grid structure is required for high-dimensional problems; negating the superiority of the FVM over the FDM by means of the geometric flexibility. On general unstructured grids, the FVM fails to reach high-order accuracy, which is its biggest drawback. The method that shines for its ability to put together the desired properties of both the FDM and the FVM is the FEM. The FEM not only allows the use of elements with varying sizes, but also has the relatively simple extension for higher-order approximations. In particular, the FEM possesses the property known as the *hp*-adaptivity [32, 33], that is the allowance of different orders of approximations in each element, that results to changes in size and order locally. However, the drawbacks for the FEM are basis functions being global and implicit structure of the semi-discrete scheme. Shortly, a method rises on the point where another one fails, but there is no method to overcome all the drawbacks at once. As we have already seen, the geometric flexibility is achieved through the use of element-based methods. Adding to this information we must look for a method in which the high-order accuracy is obtained through the local elements, as in the FEM, where the basis functions are defined locally, instead of globally, as in the FVM. DG method is the elegant combination of the FEM and the FVM, that offers the desired properties of every method mentioned above. The price that has to be paid for using the DG method reveals itself as the increase in the total degrees of freedom. However, the system resulting from the DG method is generally more sparse than the system resulting from the FEM, albeit being larger, that would potentially lead to a faster solution procedure when tackled in a smart way. In Table 2.1, we summarize the properties that we have mentioned so far. It reflects the drawbacks of the methods, however, one can overcome these drawbacks in different ways.

Table 2.1: The summary of properties of mostly used methods to find approximate solutions to the boundary value problems. $\checkmark$ and $\times$ stand for the success and the drawbacks of the methods, respectively.

	Complex geometries	High-order accuracy and *hp*-adaptivity	Explicit semi-discrete form
FDM	×	✓	✓
FVM	✓	×	✓
FEM	✓	✓	×
DG	✓	✓	✓

In literature there exists various DG methods [6, 8, 9, 25, 28, 35, 63]. Among these methods, local discontinuous Galerkin (LDG), compact discontinuous Galerkin (CDG) and interior penalty Galerkin (IPG) methods, which are completely consistent and stable, converge with optimal order with respect to the L^2 and H^1 norms. On the other hand, the inconsistent pure penalty methods, i.e., Babuška-Zlámal [6], Brezzi et al. [25], fail to achieve the optimal convergence property with respect to the L^2 norm. The method of Baumann-Oden ($k \geq 2$) [9], where k is the order of polynomial approximation, and its stabilized form, nonsymmetric interior penalty Galerkin (NIPG) [70], while lacking adjoint consistency, stand between the aforementioned two classes of methods as they have a suboptimal convergence rate. The cure for the suboptimal rate of convergence of the NIPG is to use a superpenalization approach [69]. However the drawback of using superpenalization approach reveals itself as the increase in the condition number of the stiffness matrix [27]. Table 2.2 summarizes the results regarding the consistency, adjoint consistency, stability and rate of convergence in H^1 and L^2 norms, for various DG methods.

In this chapter we explain how to discretize an elliptic problem using one of the IPG methods; the symmetric interior penalty Galerkin (SIPG) method, introduced by Arnold [3] and Wheeler [79], then we explain how to discretize convection part of a convection-diffusion problem using the upwind discretization method [54, 66].

Table 2.2: The summary of properties of various DG methods, where k is the order of polynomial approximation. ✓ and × stand for the success and the drawbacks of the methods, respectively.

Method	Consistency	Adjoint Consistency	Stability	H^1	L^2
Brezzi et al. [25]	✓	✓	✓	h^k	h^{k+1}
LDG [28]	✓	✓	✓	h^k	h^{k+1}
CDG [63]	✓	✓	✓	h^k	h^{k+1}
IPG [35]	✓	✓	✓	h^k	h^{k+1}
Bassi et al. [8]	✓	✓	✓	h^k	h^{k+1}
NIPG [70]	✓	×	✓	h^k	h^k
Babuška-Zlámal [6]	×	×	✓	h^k	h^{k+1}
Baumann-Oden ($k = 1$) [9]	✓	×	×	×	×
Baumann-Oden ($k \geq 2$) [9]	✓	×	×	h^k	h^k
Bassi-Rebay [8]	✓	✓	×	$[h^k]$	$[h^{k+1}]$

2.1 Preliminaries

We first present some definitions which are required to construct DG methods in the rest of the thesis.

In this thesis, we denote a bounded polygonal domain in $\mathbb{R}^d$ by Ω. For $1 \leq p < \infty$, the spaces $L^p(\Omega)$ that denotes the p-integrable functions are defined as

$$L^p(\Omega) = \{v \text{ measurable} : \|v\|^2_{L^p(\Omega)} < \infty\},$$

where $||\cdot||_{L^p(\Omega)}$ is defined as

$$\begin{aligned} L^p(\Omega) &= \left(\int_\Omega |v(x)|^p \, dx\right)^{1/p}, \quad 1 \leq p < \infty, \\ L^\infty(\Omega) &= \text{ess sup}\{|v(x)| : x \in \Omega\}. \end{aligned}$$

The space $L^2(\Omega)$, which is the mainly considered space throughout this thesis, is a Hilbert space with respect to the inner product and the norm that are defined as

$$(u, v)_\Omega = \int_\Omega u(x)v(x) \, dx, \text{ and } \quad ||v||_{L^2(\Omega)} = (v, v)^{1/2}_\Omega,$$

respectively.

Definition 2.1. For a continuous function v defined on $\mathbb{R}^d$, the support of v is the closure of the set of points at which the function is not equal to zero. v is said to have compact support in the domain Ω, if the support of v is bounded and included in the interior of Ω.

Let $\mathcal{D}(\Omega) := \{v \in C^{\infty} : v \, \text{has compact support in} \, \Omega\}$. Then for any multi-index $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$, where $|\alpha| = \sum_{i=1}^{d} \alpha_i$, the distributional derivative $D^{\alpha}v$ is given as

$$D^{\alpha}v(\psi) = (-1)^{|\alpha|} \int_{\Omega} v(x) \frac{\partial^{\alpha}\psi}{\partial x_1^{\alpha_1}, \ldots, \partial x_d^{\alpha_d}}, \qquad \forall \psi \in \mathcal{D}(\Omega).$$

Then, the Sobolev space $W^{(k,p)}$ is

$$W^{(k,p)}(\Omega) = \{v \in L^p(\Omega) : D^{\alpha}v \in L^p(\Omega), \forall \, 0 \leq |\alpha| \leq k\}.$$

Our interest here focuses on the Sobolev space defined as $H^s(\Omega) := W^{(s,2)}(\Omega)$, for $s \in \mathbb{Z}$. Then, the Sobolev norm $||\cdot||_{H^s(\Omega)}$ and the Sobolev seminorm $|\cdot|_{H^s(\Omega)}$ associated with the Sobolev space $H^s(\Omega)$ are given as

$$||v||_{H^s(\Omega)} = \left(\sum_{0 \leq |\alpha| \leq s} ||D^{\alpha}v||_{L^2(\Omega)}^2 \right)^{1/2}, \; |v|_{H^s(\Omega)} = \left(\sum_{|\alpha| = s} ||D^{\alpha}v||_{L^2(\Omega)}^2 \right)^{1/2}.$$

However, the DG methods also make use of the space $H^s(\Omega)$ for $s \in \mathbb{R}$, called the broken Sobolev space, which depend on the partition of the domain Ω. Let Ω be divided into triangular elements T with the intersection of any two elements is either empty, a vertex or an edge, and denote this division into elements T by $\mathcal{T}_h$. Then, the broken Sobolev space for $s \in \mathbb{R}$ is defined as

$$H^s(\mathcal{T}_h) = \{v \in L^2(\Omega) : v|_T \in H^s(T), \; \forall T \in \mathcal{T}_h\}$$

The broken Sobolev norm $|| \cdot ||_{H^s(\mathcal{T}_h)}$ and the broken gradient seminorm $| \cdot |_{H^0(\mathcal{T}_h)}$ associated with the broken Sobolev space $H^s(\mathcal{T}_h)$ are also given as

$$||v||_{H^s(\mathcal{T}_h)} = \left(\sum_{T \in \mathcal{T}_h} ||v||_{H^s(T)}^2 \right)^{1/2}, \qquad |v|_{H^0(\mathcal{T}_h)} = \left(\sum_{T \in \mathcal{T}_h} ||\nabla v||_{L^2(T)}^2 \right)^{1/2}.$$

2.2 Symmetric Interior Penalty Galerkin (SIPG) Method for Convection-Diffusion Equations

We here consider the following convection-diffusion-reaction problem:

$$\begin{aligned} -\epsilon \Delta y(x) + \beta(x) \cdot \nabla y(x) + r(x) y(x) &= f(x), \qquad & x \in \Omega, \\ y(x) &= g^D(x), \qquad & x \in \Gamma^D, \\ \epsilon \nabla y(x) \cdot \mathbf{n}(x) &= g^N(x), \qquad & x \in \Gamma^N, \end{aligned} \tag{2.1}$$

where

$$f \in L^2(\Omega), \; g^D \in H^{3/2}(\Gamma^D), \; g^N \in H^{1/2}(\Gamma^N), \; 0 < \epsilon, \; \beta(x) \in W^{1,\infty}(\Omega)^2, \; r \in L^{\infty}(\Omega),$$

with the normal vector $\mathbf{n}$ to the boundary. Furthermore, for the well-posedness of the problem (2.1), we need the following coercivity condition:

$$r - \frac{1}{2}\nabla \cdot \beta(x) \geq r_0 \geq 0. \tag{2.2}$$

We now begin with the classical weak formulation of the model problem (2.1): find $y \in Y$ such that

$$\int_{\Omega} (\epsilon \nabla y \cdot \nabla v + \beta \cdot \nabla y v)\, dx = \int_{\Omega} f v \, dx + \int_{\Gamma^N} g^N v \, ds, \ \forall v \in V, \tag{2.3}$$

where the solution space Y and the test function space V are defined as

$$\begin{aligned} Y &= \{y \in H^1(\Omega) : y = g^D \ \text{ on } \Gamma^D\}, \\ V &= \{v \in H^1(\Omega) : v = 0 \ \text{ on } \Gamma^D\}, \end{aligned} \tag{2.4}$$

respectively.

Let $\{\mathcal{T}_h\}_h$ be a family of conforming triangulation of the polygonal domain Ω. Each mesh $\mathcal{T}_h$ consists of triangular elements $T_{1,\ldots,k} \in \mathcal{T}_h$ such that $\overline{\Omega} = \bigcup_{i=1}^{k} \overline{T_i}$, and the intersection of two distinct triangles is either empty or consists of two nodes and the edge connecting these nodes. Let $\mathcal{E}_h := \mathcal{E}_h^0 \cup \mathcal{E}_h^D \cup \mathcal{E}_h^N$ and $\mathcal{E}_h^i \cap \mathcal{E}_h^j = \emptyset$, where $\mathcal{E}_h^0, \mathcal{E}_h^D$ and $\mathcal{E}_h^N$ are the interior edges, Dirichlet boundary edges and Neumann boundary edges, respectively. The diameter of an element T and the length of an edge E are denoted by h_T and h_E, respectively. Furthermore, the mesh size h of the triangulation $\mathcal{T}_h$ is defined by:

$$h = \max_{T \in \mathcal{T}_h} h_T.$$

The discrete solution space Y_h and the discrete test function space V_h are defined as

$$V_h = Y_h = \{v \in L^2(\Omega) : v|_T \in \mathbb{P}_k(T), \forall T \in \mathcal{T}_h\}, \tag{2.5}$$

where, for any $T \in \mathcal{T}_h$, the set $\mathbb{P}_k(T)$ denotes the set of polynomials of degree at most k on T. Note that the space Y_h of discrete solutions and the space V_h of test functions are chosen to be identical due to the weak treatment of the boundary conditions in DG framework.

As the functions in V_h are not necessarily continuous, we have two different traces for two adjacent triangles which share an interior edge. Let T_i and $T_j \in \mathcal{T}_h$ be two adjacent triangles sharing $e \in \mathcal{E}_h^0$, see Figure 2.1, v_i and v_j denote the traces of a scalar function v on T_i and on T_j, respectively. Then, the jump and average values of v on the edge e are

$$\begin{aligned} [v] &= v_i \mathbf{n}_e - v_j \mathbf{n}_e, \\ \{v\} &= \frac{1}{2}(v_i + v_j), \end{aligned}$$

respectively, where $\mathbf{n}_e$ is the unit normal vector to the edge e oriented from T_i to T_j. Similarly, let $\mathbf{w}_i$ and $\mathbf{w}_j$ denote the traces of a vector valued function $\mathbf{w}$ on T_i and on T_j, respectively. Then, the jump and average values of $\mathbf{w}$ on the edge e are given by

$$\begin{aligned} [\mathbf{w}] &= \mathbf{w}_i \cdot \mathbf{n}_e - \mathbf{w}_j \cdot \mathbf{n}_e, \\ \{\mathbf{w}\} &= \frac{1}{2}(\mathbf{w}_i + \mathbf{w}_j), \end{aligned}$$

respectively. Note that on $e \in \mathcal{E}^0$, the jump $[v]$ of the scalar function v is a vector which is parallel to the normal vector of e, and the jump $[\mathbf{w}]$ of the vector valued function $\mathbf{w}$ is a scalar value.

For any $e \in T_i \cap \partial\Omega$, we also define

$$\begin{aligned} [v] &= v_i\mathbf{n}, \\ \{v\} &= v_i, \end{aligned}$$

and

$$\begin{aligned} [\mathbf{w}] &= \mathbf{w}_i \cdot \mathbf{n}, \\ \{\mathbf{w}\} &= \mathbf{w}_i, \end{aligned}$$

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

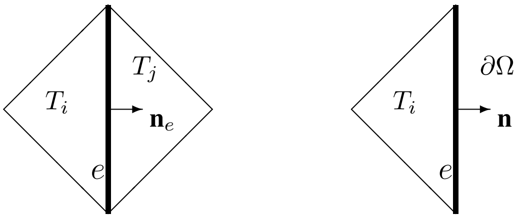

Figure 2.1: Two adjacent elements sharing an edge (left); an element with an edge on the boundary of domain (right).

We will now present the derivation of the symmetric discontinuous interior penalty Galerkin (SIPG) method for the diffusion part of the problem (2.1). Let the following elliptic problem be given by

$$\begin{aligned} -\epsilon\Delta y &= f && \text{in } \Omega, \\ y &= g^D && \text{on } \Gamma^D, \\ \nabla y \cdot \mathbf{n} &= g^N && \text{on } \Gamma^N. \end{aligned} \tag{2.6}$$

Let us multiply (2.6) by a test function $v \in V_h$ and split the integral for each triangle in T.

$$-\sum_{T\in\mathcal{T}_h} \int_T \epsilon\Delta yv \, dx = \sum_{T\in\mathcal{T}_h} \int_T fv \, dx.$$

An application of the divergence theorem on every element integral yields

$$\sum_{T\in\mathcal{T}_h} \int_T \epsilon\nabla y \cdot \nabla v \, dx - \sum_{T\in\mathcal{T}_h} \int_{\partial T} \epsilon(\nabla y \cdot \mathbf{n})v \, ds = \sum_{T\in\mathcal{T}_h} \int_T fv \, dx + \sum_{e\in\mathcal{E}_h^N} \int_e g^N v \, ds.$$

Using the definition of the jump operator and assuming that $v \in V_h$ are discontinuous on the interior edges, we obtain

$$\sum_{T\in\mathcal{T}_h} \int_T \epsilon\nabla y \cdot \nabla v \, dx - \sum_{e\in\mathcal{E}_h^0\cup\mathcal{E}_h^D} \int_e [\epsilon v\nabla y] \, ds = \sum_{T\in\mathcal{T}_h} \int_T fv \, dx + \sum_{e\in\mathcal{E}_h^N} \int_e g^N v \, ds.$$

As y is assumed to be smooth enough so that ∇y is continuous, we get $[\nabla y] = 0$. Combining this with the fact that $[v\nabla y] = \{\nabla y\} \cdot [v] + [\nabla y] \cdot \{v\}$, we derive

$$\sum_{T\in\mathcal{T}_h} \int_T \epsilon\nabla y \cdot \nabla v \, dx - \sum_{e\in\mathcal{E}_h^0\cup\mathcal{E}_h^D} \int_e \{\epsilon\nabla y\} \cdot [v] \, ds = \sum_{T\in\mathcal{T}_h} \int_T fv \, dx + \sum_{e\in\mathcal{E}_h^N} \int_e g^N v \, ds.$$

Since y is continuous on the interior edges, i.e., $[y] = 0$, we can manipulate the equation by adding integrals involving $[y]$, as a multiplier, as shown below:

$$\begin{aligned}\sum_{T\in\mathcal{T}_h} \int_T \epsilon\nabla y \cdot \nabla v \, dx &- \sum_{e\in\mathcal{E}_h^0\cup\mathcal{E}_h^D} \int_e \{\epsilon\nabla y\} \cdot [v] \, ds - \sum_{e\in\mathcal{E}_h^0\cup\mathcal{E}_h^D} \int_e \{\epsilon\nabla v\} \cdot [y] \, ds \\ + \sum_{e\in\mathcal{E}_h^0\cup\mathcal{E}_h^D} \frac{\sigma\epsilon}{h_e} \int_e [y] \cdot [v] \, ds &= \sum_{T\in\mathcal{T}_h} \int_T fv \, dx + \sum_{e\in\mathcal{E}_h^D} \int_e g^D (\frac{\sigma\epsilon}{h_e} v - \nabla v \cdot \mathbf{n}) \, ds \\ &+ \sum_{e\in\mathcal{E}_h^N} \int_e g^N v \, ds,\end{aligned}$$

which is called the SIPG formulation of the problem (2.6), where σ is the sufficiently large nonnegative penalty parameter that guarantees coercivity. The choice of σ affects the jumps on interior edges, i.e., larger σ values yield to a smaller jump value, that would eventually have an effect on the DG approximation of the solution. If σ goes to infinity, then the jump values goes to zero, implying that the discontinuities on the interior edges of numerical DG approximation start to diminish and the approximation would converge to the continuous Galerkin approximation [26].

Let **n** denote the unit outward normal vector to $\partial\Omega$. The inflow boundary edges Γ^- and the outflow boundary edges Γ^+ of $\partial\Omega$ are defined as

$$\Gamma^- := \{x \in \partial\Omega : \beta(x) \cdot \mathbf{n}(x) < 0\}, \text{ and } \Gamma^+ := \{x \in \partial\Omega : \beta(x) \cdot \mathbf{n}(x) \geq 0\}.$$

Similarly, let $\mathbf{n}_T$ denote the unit normal vector on the boundary ∂T of an element T. Then, the inflow boundary edges ∂T^- and the outflow boundary edges ∂T^+ of an element T are defined as

$$\partial T^- := \{x \in \partial T : \beta(x) \cdot \mathbf{n}_T(x) < 0\}, \text{ and } \partial T^+ := \{x \in \partial T : \beta(x) \cdot \mathbf{n}_T(x) \geq 0\}.$$

Additionally, on an interior edge ∂T, the traces of a piecewise continuous scalar function y from inside and outside of the element T are defined as

$$y^{in}(x) := \lim_{\delta\to 0+} y(x + \delta\beta), \text{ and } y^{out}(x) := \lim_{\delta\to 0+} y(x - \delta\beta), \text{ respectively.}$$

Then, with the use of the definitions above, the SIPG discretization of the convection-diffusion problem (2.1), where the convection part is discretized by upwind discretization, is given as follows: find $y_h \in Y_h$ such that

$$a_h(y_h, v_h) = l_h(v_h) \qquad \forall v_h \in V_h, \tag{2.7}$$

where

$$
\begin{aligned}
a_h(y_h, v_h) = & \sum_{T\in\mathcal{T}_h} \int_T \epsilon \nabla y_h \cdot \nabla v_h \, dx + \sum_{T\in\mathcal{T}_h} \int_T (\beta \cdot \nabla y_h v_h + r y_h v_h) \, dx \\
& - \sum_{e\in\mathcal{E}_h^0 \cup \mathcal{E}_h^D} \int_e \{\epsilon \nabla v_h\} \cdot [y_h] \, ds - \sum_{e\in\mathcal{E}_h^0 \cup \mathcal{E}_h^D} \int_e \{\epsilon \nabla y_h\} \cdot [v_h] \, ds \\
& + \sum_{T\in\mathcal{T}_h} \int_{\partial T^- \setminus \partial\Omega} (\beta \cdot \mathbf{n})(y_h^{out} - y_h^{in}) v_h^{in} \, ds - \sum_{T\in\mathcal{T}_h} \int_{\partial T^- \cap \Gamma^-} (\beta \cdot \mathbf{n}) y_h^{in} v_h^{in} \, ds \\
& + \sum_{e\in\mathcal{E}_h^0 \cup \mathcal{E}_h^D} \frac{\sigma\epsilon}{h_e} \int_e [y_h] \cdot [v_h] \, ds \\
l_h(v_h) = & \sum_{T\in\mathcal{T}_h} \int_T f v_h \, dx + \sum_{e\in\mathcal{E}_h^D} \int_e g^D (\frac{\sigma\epsilon}{h_e} v_h - \epsilon \nabla v_h \cdot \mathbf{n}) \, ds + \sum_{e\in\mathcal{E}_h^N} \int_e g^N v_h \, ds \\
& - \sum_{T\in\mathcal{T}_h} \int_{\partial T^- \cap \Gamma^-} (\beta \cdot \mathbf{n}) g^D v_h^{in} \, ds.
\end{aligned}
$$

The existence and uniqueness of the solution y_h to the problem (2.7) is guaranteed by Theorem 4.2 in [48], and an error estimate for this solution is given by Theorem 5.1 in [38], with respect to the norms that are defined in the aforementioned papers, respectively.

CHAPTER 3

MULTIGRID METHODS

After the discretization of the problem (2.1), a linear system remains to be solved. For this purpose, the direct solvers are generally costly and inefficient, and iterative solvers, such as Gauss-Seidel, Jacobi or Successive Over-Relaxation, lose their efficiency as the mesh size is decreased, hence, reaching higher accuracy of the approximate solution solely with an iterative solver is a rough task. However, there are also methods that can preserve their convergence speed for decreasing mesh size, one of these methods being multigrid methods [22, 41, 44, 76], which this thesis focuses on.

We here consider the geometric multigrid approach, based on the hierarchy of meshes for our linear model. However, algebraic multigrid method (AMG) [67, 74, 76] can also be thought of especially for nonlinear problems, where the application of the geometric multigrid method is too difficult. For a (geometric) multigrid scheme, we need a hierarchy of nested meshes and the two main ingredients: the smoothing operators and the coarse-grid correction step.

The simple iterative solvers, such as Gauss-Seidel, Jacobi or Successive Over Relaxation, are known to be effective solvers for smoothing the high frequency error in the approximation, however these methods lack the desired convergence rates for low frequency components of the error [49]. The simple yet elegant idea of multigrid methods is to eliminate the high frequency and the low frequency error components via the smoothing operators and coarse-grid correction steps, respectively.

The traditional way to understand the multigrid method is to first realize how a two-grid method works, as the former is the recursive implementation of the latter. Main ingredients of a two-grid algorithm are the two meshes, namely a coarse mesh and a fine mesh, the transfer operations between these two meshes, and the smoothing operators on these meshes. The fine grid is obtained by the uniform triangulation of the coarse grid. The operator that transfers information from the fine grid to coarse grid is called the restriction operator, and the operator that transfers information from the coarse grid to fine grid is called the prolongation operator. Note that, in a multigrid scheme, two-grid scheme is named as the aforementioned coarse-grid correction step.

In this chapter, we firstly define the operators needed for coarse-grid correction step and the smoothing operators. Afterwards, we present an ordering algorithm introduced by Bey and Wittum [11], that we will use to fasten our multigrid scheme. We conclude this chapter by giving some numerical examples.

3.1 Restriction and Prolongation operators

The aim of the transfer operators, namely restriction and prolongation operators, is to form a relation between two nested interpolation spaces, V_h and V_H, where V_H is coarser than V_h, satisfying $V_H \subseteq V_h$. These spaces are defined in terms of equation (2.5) as

$$V_h = \{v_h \in L^2(\Omega) : v_h|_T \in \mathbb{P}_k(T),\ \forall T \in \mathcal{T}^h\},$$
$$V_H = \{v_H \in L^2(\Omega) : v_H|_T \in \mathbb{P}_k(T),\ \forall T \in \mathcal{T}^H\},$$

respectively. Additionally, $\mathcal{T}^h$ and $\mathcal{T}^H$ are the fine and coarse triangulations of Ω with the set of triangular elements $\{T_i^h\}$ and $\{T_I^H\}$, respectively. Figure 3.1 illustrates two different triangulations of a domain $\Omega = [0,1]^2$ with 8 and 32 triangular elements.

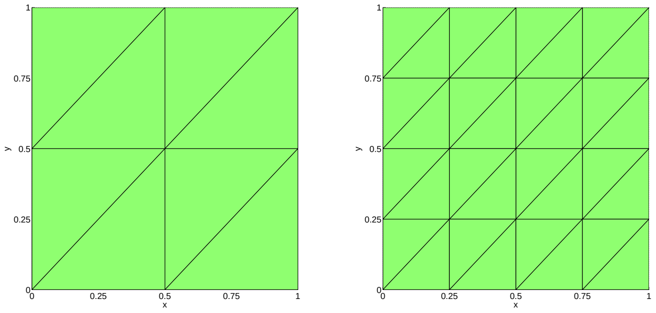

Figure 3.1: A coarse triangulation with 8 triangular elements (left), and a fine triangulation with 32 triangular elements (right) of a domain $\Omega = [0,1]^2$.

Let the equation (2.1) be discretized on $\mathcal{T}^h$ and let y_h be the discrete solution that belongs to V_h. Then, the discrete solution can be defined by

$$y_h = \sum_{i=1}^{p} \sum_{m=1}^{nloc} \mathrm{Y}_{i,m}^h \varphi_{i,m}^h, \tag{3.1}$$

where $p = |\{T_i^h\}|$, i.e., the number of elements in the fine space triangulation $\mathcal{T}^h$, $\{\{\varphi_{i,m}\}_{m=1}^{nloc}\}_{i=1}^{p}$ is the basis functions of the space V_h, and $nloc$ is the local dimension.

Similarly, let the equation (2.1) be discretized on $\mathcal{T}^H$ and let y_H be the discrete solution that belongs to V_H. Then, we have

$$y_H = \sum_{I=1}^{P} \sum_{M=1}^{nloc} \mathrm{Y}_{I,M}^H \varphi_{I,M}^H, \tag{3.2}$$

where $P = |\{T_I^H\}|$, i.e., the number of elements in the coarse space triangulation $\mathcal{T}^H$, and and $\{\{\varphi_{I,M}\}_{M=1}^{nloc}\}_{I=1}^{P}$ is the basis for the space V_H.

The purpose of the restriction operator is to project $y_h \in V_h$ onto V_H, such that

$$(y_h - y_H, \varphi^H_{I,M})_{T^H_I} = \mathbf{0}, \quad \forall I \in \{1, \dots, P\} \quad \text{and} \quad \forall M \in \{1, \dots, nloc\}, \tag{3.3}$$

Suppose a triangle T^H_I in the coarse triangulation of Ω is decomposed into n triangles in the fine triangulation of the same domain Ω. Then it follows from (3.3) that

$$\sum_{i=1}^{n} \sum_{m=1}^{nloc} \mathrm{Y}^h_{i,m} (\varphi^h_{i,m} \varphi^H_{I,S})_{T^h_i} = \sum_{M=1}^{nloc} \mathrm{Y}^H_{I,M} (\varphi^H_{I,M}, \varphi^H_{I,S})_{T^H_I} \quad \forall S \in \{1, \dots, nloc\}. \tag{3.4}$$

Writing the above equality in matrix form gives

$$\mathrm{Y}^H_I = \sum_{i=1}^{n} \mathbf{M}^{-1}_{H_I} \mathbf{M}^{H_I}_{h_i} \mathrm{Y}^h_i, \;\; \text{where} \;\; \left(\mathbf{M}^{H_I}_{h_i}\right)_{S,M} = (\varphi^h_{i,m}, \varphi^H_{I,S})_{T^h_i} \tag{3.5}$$

with $\mathbf{M}_{H_I}$ being the coarse triangle mass matrix. Moving from one triangle of the coarse triangulation to the all triangles gives the global restriction operator I^H_h that satisfies

$$\mathrm{Y}^H = I^H_h \mathrm{Y}^h, \tag{3.6}$$

where U^h and U^H are the vectors of degrees of freedom in the fine and coarse spaces, respectively, and I^H_h is given explicitly as

$$I^H_h = \begin{bmatrix} \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I^{H_I}_h & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots \end{bmatrix}, \tag{3.7}$$

where $I^{H_I}_h = \left[\dots \mathbf{M}^{-1}_{H_I} \mathbf{M}^{H_I}_{h_i} \dots\right]$, which is called the local restriction operator. Notice that the global restriction operator is a $p \times P$ block matrix where each block is of size $nloc \times nloc$.

Once the computation of the global restriction matrix, restriction matrix from now on, is complete, it is easy to get the global prolongation matrix, prolongation matrix from now on, as it is defined to be the transpose of the restriction matrix:

$$I^h_H := \left(I^H_h\right)^T. \tag{3.8}$$

3.2 Smoothing Operators

As we have already mentioned, the second important ingredient of a multigrid algorithm is the smoothing operators. Splitting methods such as Jacobi method and Gauss-Seidel method [71, 78] can be used as the smoothing operators.

3.2.1 Splitting Methods

Let the following linear system be given

$$Ax = b, \tag{3.9}$$

where $A \in \mathbb{R}^{n\times n}$, $x, b \in \mathbb{R}^{n\times 1}$, and assume that A is nonsingular with all diagonal entries being nonzero.

The idea of a splitting method is to split the any given matrix A such that for $D, L, U \in \mathbb{R}^{n\times n}$, $A = D + L + U$, where D is the diagonal part of A, and U, L are the strictly upper triangular and the strictly lower triangular parts of A, respectively.

Then, (3.9) can be written as

$$(D + L + U)x = b \tag{3.10}$$

or,

$$Dx = -(L + U)x + b. \tag{3.11}$$

Furthermore, (3.11) can be explicitly written as

```
For $i = 1 : n$
    $D(i,i)x^{(m+1)}(i) = b(i) - \sum_{j=1}^{i-1} L(i,j)x^{(m)}(j) - \sum_{j=i+1}^{n} U(i,j)x^{(m)}(j)$
End
```

(3.12)

where $m \geq 0$ is the iteration number and $x^{(0)}$ is the initial guess for the solution of the system (3.9). As the diagonal entries of A are assumed to be nonzero, D is nonsingular, then (3.12) can be written as

```
For $i = 1 : n$
    $x^{(m+1)}(i) = \frac{1}{D(i,i)}\left(b(i) - \sum_{j=1}^{i-1} L(i,j)x^{(m)}(j) - \sum_{j=i+1}^{n} U(i,j)x^{(m)}(j)\right)$
End
```

(3.13)

which in matrix form corresponds to

$$x^{(m+1)} = D^{-1}(b - (L+U)x^{(m)}). \tag{3.14}$$

This iterative method is called the *pointwise Jacobi method.* As seen from (3.13), at $(m+1)^{st}$ smoothing step, we need to keep all the components of the initial vector $x^{(m)}$ until the calculation of the last entry of the updated vector $x^{(m+1)}$, and the updated entries of $x^{(m+1)}$ are not used until the $(m+2)^{nd}$ smoothing step. However, if we were to use the entries of $x^{(m+1)}$ as soon as they are updated, we would get the iterative method

For $i = 1 : n$

$$x^{(m+1)}(i) = \frac{1}{D(i,i)}\left(b(i) - \sum_{j=1}^{i-1} L(i,j)x^{(m+1)}(j) - \sum_{j=i+1}^{n} U(i,j)x^{(m)}(j)\right)$$

End

(3.15)

which in matrix form corresponds to

$$x^{(m+1)} = (D+L)^{-1}(b - Ux^{(m)}). \tag{3.16}$$

Notice that, as D is nonsingular, $D+L$ is also nonsingular, hence $(D+L)^{-1}$ exists. Further, the iterative method (3.15) requires only one vector to be kept for any smoothing step, which makes it more efficient than the pointwise Jacobi method.

The iterative method (3.15) defined by the splitting (3.16) is called the *pointwise forward Gauss-Seidel* iteration, as the iteration starts from the 1^{st} entry and ends at the n^{th} entry of the vector $x^{(m+1)}$. Alternatively, if the iteration were to start from the n^{th} entry and end at the 1^{st} entry of $x^{(m+1)}$, it would be called the *pointwise backward Gauss-Seidel* iteration defined as

For $i = n : 1$

$$x^{(m+1)}(i) = \frac{1}{D(i,i)}\left(b(i) - \sum_{j=1}^{i-1} L(i,j)x^{(m)}(j) - \sum_{j=i+1}^{n} U(i,j)x^{(m+1)}(j)\right)$$

End

(3.17)

3.3 Multigrid V-cycle

As we have already defined the transfer operators and smoothing operators, we are ready to define the multigrid method.

Let equation (2.1) be discretized by SIPG method and the resulting linear system be

$$A_h y_h = b_h, \tag{3.18}$$

where $A_h \in \mathbb{R}^{(nel*nloc \times nel*nloc)}$, $y_h, b_h \in \mathbb{R}^{(nel*nloc)}$, and nel is the number of elements in $\mathcal{T}^h$, i.e., $|T^h|$.

The core of a multigrid method is the so called two-grid method, which involves a coarse level mesh and a fine level mesh, see Algorithm 1.

Algorithm 1 Two-Grid Method

```
Given an initial guess $y_0$, smooth $y_0$ vpre times: $y = S(A_h, b_h, y_0, vpre)$.
Compute the residual with the smoothed approximation $y$: $r_h = b_h - A_h y$.
Restrict the residual $r_h$ to the coarse mesh: $r_H = I_h^H * r_h$.
Restrict the stiffness matrix $A_h$ to the coarse mesh: $A_H = I_h^H * A_h * I_H^h$.
Solve the coarse grid linear system: $A_H e = r_H$.
Prolongate the error and update the approximation u: $y = y + I_H^h * e$.
Smooth the updated approximation vpost times: $y = S(A_h, b_h, y, vpost)$.
```

As soon as the principle idea of two-grid method is understood fully, first thing that comes to mind is to add more levels, coarser or finer meshes, to the method. Instead of naming the method according to the number of levels, e.g. three-grid method, four-grid method, the name ”multigrid” is used to define such a method. We will here use the V-cycle type of multigrid methods, that the algorithm starts from the finest level, say *level* ℓ, for $\ell > 1$, and step by step goes to the coarsest one, say *level* 1, after directly solving the problem on the coarsest level, the algorithm starts to climb up until reaching the finest level again. This process is visualized at Figure 3.2. For the multigrid algorithm, we will denote the stiffness matrix, the approximate solution vector and the right hand side vector on level k by A_k, y_k and b_k, respectively. Additionally, let nel_k denote the number of triangles in the triangulation of level k. Then, for a given initial approximate solution $y_\ell \in \mathbb{R}^{nel_\ell * nloc}$, the single iteration of a multigrid V-cycle algorithm computes the updated approximate solution y_ℓ^* for the linear system $A_\ell y_\ell = b_\ell$ as

$$y_\ell^* = MG_\ell(A_\ell, b_\ell, y_\ell),$$

where $MG_\ell : \mathbb{R}^{nel_\ell * nloc \times nel_\ell * nloc} \times (\mathbb{R}^{nel_\ell * nloc})^2 \to \mathbb{R}^{nel_\ell * nloc}$ is defined by the following recursive algorithm:

Algorithm 2 Multigrid V-cycle: $y^* = MG_k(A, b, y)$

Define $MG_1(A, b, \cdot) = A^{-1}F$, where $MG_1 : \mathbb{R}^{nel_1 * nloc \times nel_1 * nloc} \times (\mathbb{R}^{nel_1 * nloc})^2 \to \mathbb{R}^{nel_1 * nloc}$. For $k > 1$ define $MG_k : \mathbb{R}^{nel_k * nloc \times nel_k * nloc} \times (\mathbb{R}^{nel_k * nloc})^2 \to \mathbb{R}^{nel_k * nloc}$ as:

```
Given an initial guess $y_k$, smooth $y_k$ vpre times: $y_k = S(A_k, b_k, y_k, vpre)$.
Compute the residual with the smoothed approximation $y_k$: $r_k = b_k - A_k y_k$.
Restrict the residual $r_k$: $r_{k-1} = I_k^{k-1} * r_k$.
Restrict the stiffness matrix $A_k$: $A_{k-1} = I_k^{k-1} * A_k * I_{k-1}^k$.
$e_{k-1} := 0$.
Apply multigrid to the restricted residual: $e_{k-1} = MG_{k-1}(A_{k-1}, r_{k-1}, e_{k-1})$.
Prolongate the error and update $y_k$: $y_k = y_k + I_{k-1}^k * e_{k-1}$.
Smooth the updated approximation vpost times: $y_k = S(A_k, b_k, y_k, vpost)$.
```

When a multigrid method is used as an iteration solver for a linear system, the iteration described on Algorithm 2 is used consecutively until the residual drops below the prescribed tolerance. However, similar to any other iterative solvers, the multigrid

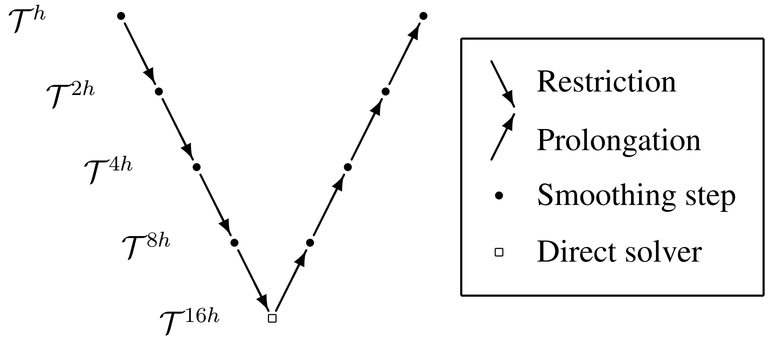

Figure 3.2: Multigrid V-cycle scheme, where $\mathcal{T}^h$ and $\mathcal{T}^{16h}$ are the coarsest and finest triangulations of the domain Ω.

method is not always convergent. In order to give the theorem for the convergence of the multigrid method, we need to rewrite the multigrid iteration $y_k^* = MG_k(A_k, b_k, y_k)$ as

$$y_k^* = G_k y_k + B_k b_k,$$

where G_k and B_k are called the iteration matrix and the preconditioner for the matrix A_k, respectively. These matrices are defined by Howard in [49, Theorem 3.1.1.] as given below:

Theorem 3.1. *Let $y_k \in \mathbb{R}^{(nel_k*nloc)}$ be an initial approximation of the solution to the system $A_k y_k = b_k$, where $A_k \in \mathbb{R}^{(nel_k*nloc\times nel_k*nloc)}$ and $b_k \in \mathbb{R}^{(nel_k*nloc)}$. Then, Algorithm 2 can be written in the form*

$$y_k^* = G_k y_k + B_k b_k,$$

where $G_k, B_k \in \mathbb{R}^{(nel_k*nloc\times nel_k*nloc)}$ are defined as

$$\begin{aligned} G_k y_k &= MG_k(A_k, 0, y_k), \\ B_k b_k &= MG_k(A_k, b_k, 0), \end{aligned}$$

for $k = 2, \ldots, \ell$, and

$$\begin{aligned} G_1 y_1 &= MG_1(A_1, 0, y_1), \\ &= 0, \\ B_1 b_1 &= MG(A_1, b_1, 0), \\ &= A_1^{-1} b_1. \end{aligned}$$

Then, using the above theorem, the multigrid iteration on the finest level ℓ is given as

$$y_\ell^* = G_\ell y_\ell + B_\ell b_\ell. \tag{3.19}$$

For a given tolerance, we can use the multigrid algorithm as an iterative solver to solve any problem. However, it can not imply the convergence for any problem. The following theorem states that the convergence of multigrid algorithm requires that the iteration matrix G_ℓ has a spectral radius strictly less than one [34, 71].

Theorem 3.2. *For any right hand side b_ℓ and any initial guess y_ℓ, the iteration (3.19) converges to the solution $\overline{y_\ell}$, if and only if the spectral radius of the iteration matrix G_ℓ is strictly less than one. Additionally, if for any matrix norm $||\cdot||$, the iteration matrix satisfies $||G_\ell|| < 1$, then the iteration (3.19) converges.*

3.4 Ordering

It is a well known fact that the performance of an iterative solution method applied on a system $Ax = b$, depends greatly on the placement of the nonzero elements of the matrix A. For the finite elements, finite difference or finite volume discretization of the convection-diffusion problem, this placement depends on the numbering of the vertices, however for the DG discretization, this placement depends on the numbering of the elements. The sparsity structure of the stiffness matrix for a triangulation with 4 elements, where the order of the polynomial approximation is 2, is illustrated in Figure 3.3.

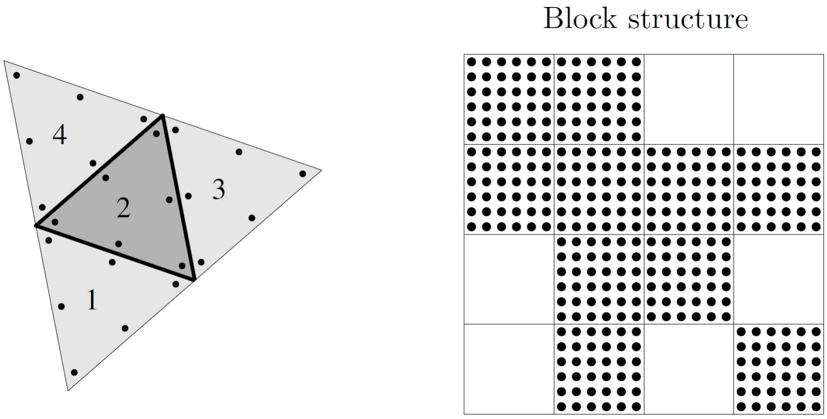

Figure 3.3: Sparsity structure of the stiffness matrix for a triangulation with 4 elements, where the order of the polynomial approximation is 2. [64, Fig. 2.1.]

As the convection term starts to dominate the diffusion term, the dominating terms of the stiffness matrix can be reordered in a way such that they appear in the lower triangular part of the matrix. In the extreme case where $\epsilon = 0$, this reordering leads to a lower triangular matrix for which the *Gauss-Seidel* method is an exact solver. For the convection term $(a, 0)\big(\text{or } (0, a)\big)$, where $a > 0$, the horizontal (or vertical) lexicographical numbering is the proper ordering for this purpose, on the other hand for the convection term $(-a, 0)\big(\text{or } (0, -a)\big)$, the horizontal (or vertical) anti-lexicographical numbering is the desired one [43].

In [11], Bey and Wittum proposed a robust algorithm called *Downwind Numbering*. The main goal of this algorithm is to reorder the vertices in such a way that the terms for convection part in the stiffness matrix is lower triangular. By doing so, this ordering yields to a stiffness matrix that has its dominating entries in the lower triangular part, as desired.

We here propose a *Downwind Numbering* for DG method as done in [11] for FVM. Denote by $C = (c_{ij})_{i,j\in\{1,\ldots,nel\}}$ and $D = (d_{ij})_{i,j\in\{1,\ldots,nel\}}$, the convection and diffusion components of the corresponding stiffness matrix $A = D + C$, which results from the DG discretization of a convection-diffusion equation, where nel is the number of triangles of the underlying triangulation $\mathcal{T}$. We say that "triangle i (T_i) is in the downwind of triangle j (T_j)" if the flow moves from T_j to T_i, i.e., if the edge e is the common edge for T_i and T_j, we have

$$\beta(x) \cdot \mathbf{n}(x) < 0, \; x \in e, \tag{3.20}$$

where **n** is the unit normal vector to the edge e oriented from T_i to T_j. This is illustrated in Figure 3.4.

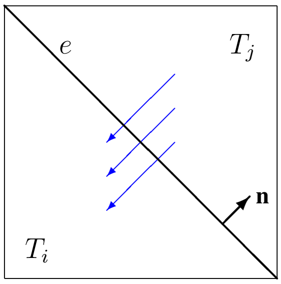

Figure 3.4: Flow direction.

In the convection component C of the stiffness matrix A, the condition (3.20) corresponds to c_{ij} being different from the zero matrix ($c_{ij} \in \mathbb{R}^{nloc\times nloc} \;\; \forall i, j \in \{1, \ldots, nel\}$). Furthermore, let $I(j)$ denote the set of triangles such that for any $k \in I(j)$, "T_j is in the downwind of T_k". Then, we can list the main steps of the algorithm as follows: For every triangle $T \in \mathcal{T}$, compute $I(T)$. The first list L_1 of triangles consists of the triangles T with $|I(T)| = 0$. If there is no cycle[1] in the convection graph, it is guaranteed that there is at least one triangle without incoming flow, implying $|I(T)| = 0$. Next step is to find the triangles that are in the downwind of the triangles in the list L_k and not in any L_j for $j < k$. Each time a triangle T is in the downwind of a triangle from the list L_k, $|I(T)|$ is decreased by 1, and if $|I(T)| = 0$ during the check on list L_k, the triangle T goes into the list L_{k+1}. The process stops when for all triangles $|I(T)| = 0$. Then, each list is sorted and concatenated to give the new ordering. This process is shown in Algorithm 3.

In the case of complicated convection, it is not easy to pick the best ordering, especially when there exists cycles in the convection matrix. In order to deal with the cycles, techniques using graph theory are presented in [40, 42, 52, 53].

[1] For a given triangle T_m, list all the triangles that are in the downwind of T_m. Then, for each of the triangles in this list, make new lists of triangles that are in the downwind of these triangles. Continue this process until no triangle has a triangle in its downwind or T_m appears in one of these lists. Notice that these two cases are mutually exclusive. Then, the convection graph is said to have "no cycle" if the first case is satisfied.

Algorithm 3 Downwind Numbering

```
for $T \in \mathcal{T}$ do
    compute $I(T)$.
end for
let $L = L_0$ be the list of all triangles with $|I(T)| = 0$.
k=0.
while $L_k \neq \emptyset$ do
    $L_{k+1} = \emptyset$.
    for $T \in L_k$ do
        for all neighbors $G$ of $T$ that are in the downwind of $T$ do
            if $|I(G)| \neq 0$ then
                $|I(G)| = |I(G)| - 1$.
                if $|I(G)| = 0$ then
                    $L_{k+1} = L_{k+1} \cup \{G\}$.
                end if
            end if
        end for
    end for
    sort $L_{k+1}$ and add it to the list $L$.
    $k = k + 1$.
end while
if $|L| \neq nel$ then
    display error: Convection graph contains cycles!
end if
```

3.5 Numerical Results

We now present some numerical demonstration to test the effectiveness of the *Downwind Numbering* algorithm. In numerical examples, we choose the pointwise forward Gauss-Seidel method as both pre-smoother and post-smoother. The penalty parameter in the SIPG method is chosen as $\sigma = 6$ on the interior edges and $\sigma = 12$ on the boundary edges.

We consider the following convection-diffusion equation:

$$\begin{aligned} -\epsilon\Delta y + \beta \cdot \nabla y = f \quad & in \quad \Omega, \\ y = g \quad & on \quad \partial\Omega, \end{aligned} \tag{3.21}$$

where $\epsilon = 10^{-3}$, $\beta = (1, 0)^T$ and $\Omega = (0, 1)^2$. Further, the analytical solution is given by

$$y(x_1, x_2) = x_1^2 + x_2^2. \tag{3.22}$$

Figures 3.5-3.8 present the positioning of the nonzero elements of the stiffness and convection matrices, without and with ordering for various mesh sizes $2^{-1}, 2^{-2}$ and 2^{-5}. Notice that after reordering, convection matrices become lower triangular as intended.

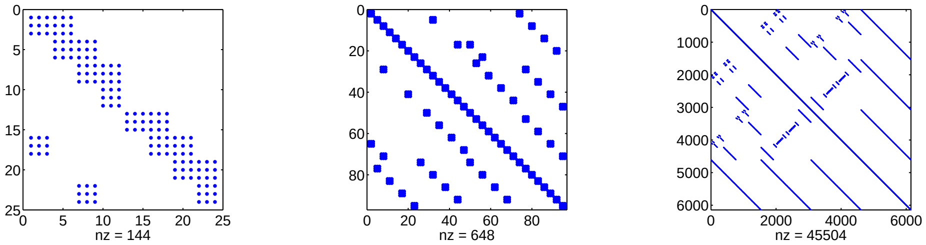

Figure 3.5: Convection matrices on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

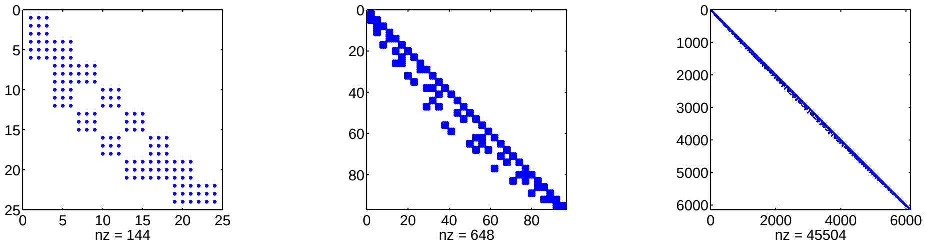

Figure 3.6: Convection matrices after *Downwind Numbering* on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

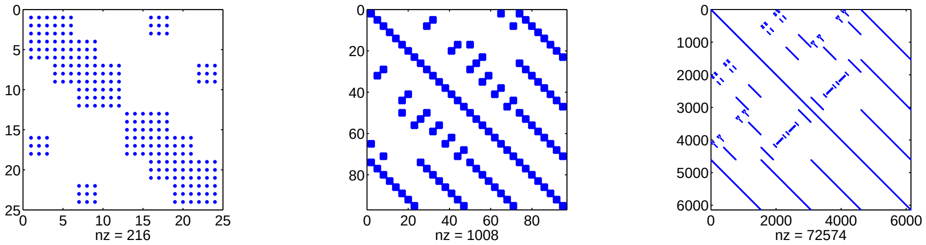

Figure 3.7: Stiffness matrices on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

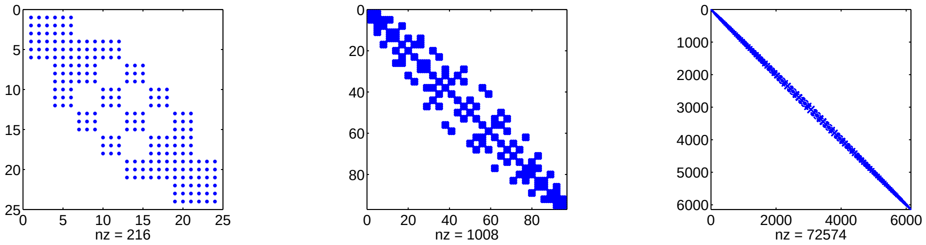

Figure 3.8: Stiffness matrices after *Downwind Numbering* on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

To check the efficiency of the *Downwind Numbering* algorithm, we give the number of V-cycles required to reach a ratio of 10^{-8} in terms of relative residual in Tables 3.1-3.2, where k stands for the number of levels used aside from the finest one, and m stands for the number of pre- and post-smoothing steps. Further, Figure 3.9 illustrates the

Table 3.1: Number of iterations required to reach a relative residual of 10^{-8} with (top) and without (bottom) reordering.

	m=2	*m*=4	*m*=6	*m*=8	*m*=10	*m*=12	*m*=14	*m*=16	*m*=18	*m*=20
k=1	10	5	4	3	2	2	2	2	1	1
k=2	10	5	4	3	2	2	2	2	2	1
k=3	10	6	4	3	3	2	2	2	2	2
k=4	19	11	7	6	5	4	3	3	3	3

	m=2	*m*=4	*m*=6	*m*=8	*m*=10	*m*=12	*m*=14	*m*=16	*m*=18	*m*=20
k=1	15	6	4	3	3	2	2	2	2	1
k=2	14	7	4	3	3	2	2	2	2	2
k=3	17	8	6	4	4	3	3	2	2	2
k=4	25	14	10	7	6	5	4	4	4	3

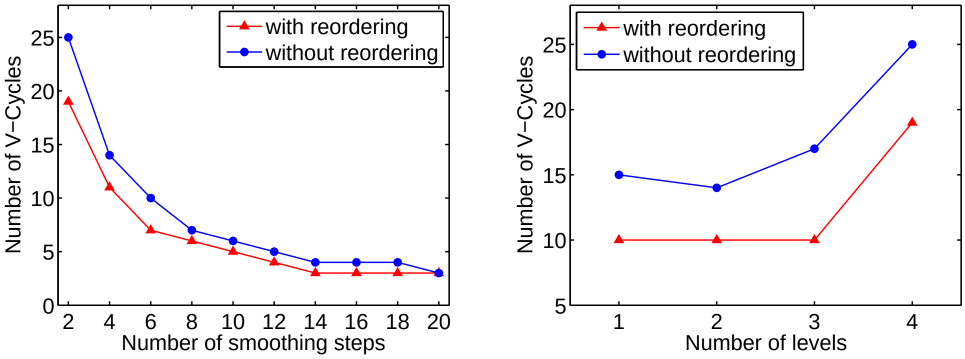

Figure 3.9: The required number of V-cycles to reach the desired relative residual of 10^{-8} for varying number of smoothing steps with fixed mesh size of 2^{-5} (left), and for varying number of levels aside from the finest one with fixed number of smoothing steps that is given as 2 (right), for $\beta = (1, 0)^T$.

required number of V-cycles to reach the desired relative residual for varying number of smoothing steps with fixed mesh size of 2^{-5}, and for varying number of levels aside from the finest one with fixed number of smoothing steps that is given as 2, left and right, respectively. As seen in the Table 3.1, the effect *Downwind Numbering* algorithm is more evident for smaller number of smoothing steps.

Let us change the convection term as $\beta = (-2, -1)^T$ for the same problem (3.21). Similarly, the sparsity patterns of the convection and stiffness matrices and the number of iterations, for with and without the application of *Downwind Algorithm* are displayed in Figures 3.10-3.13 and Table 3.2, respectively. Similarly, Figure 3.14 illustrates the required number of V-cycles to reach the desired relative residual for varying number of smoothing steps with fixed mesh size of 2^{-5}, and for varying number of levels aside from the finest one with fixed number of smoothing steps that is given as 2, left and right, respectively.

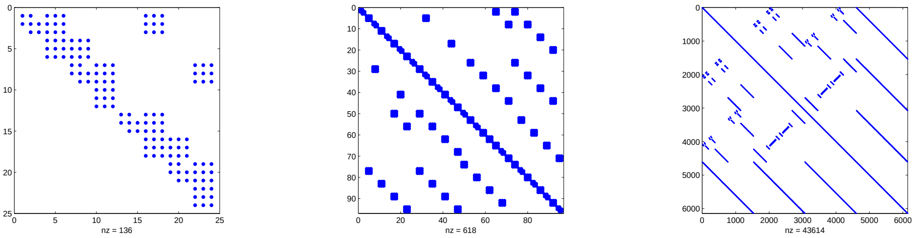

Figure 3.10: Convection matrices on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

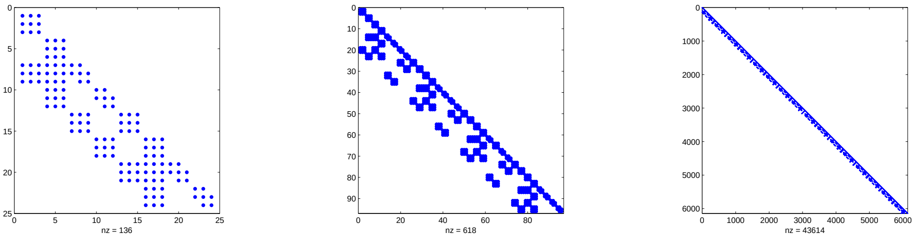

Figure 3.11: Convection matrices after *Downwind Numbering* on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

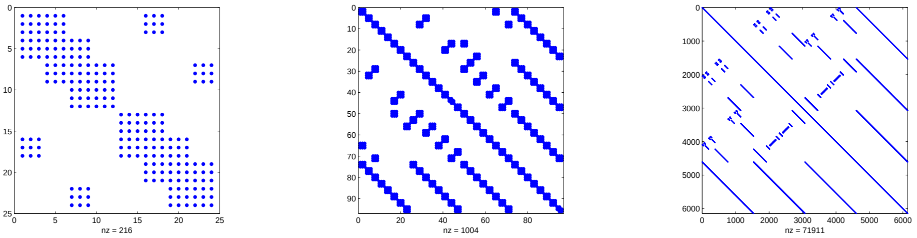

Figure 3.12: Stiffness matrices on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

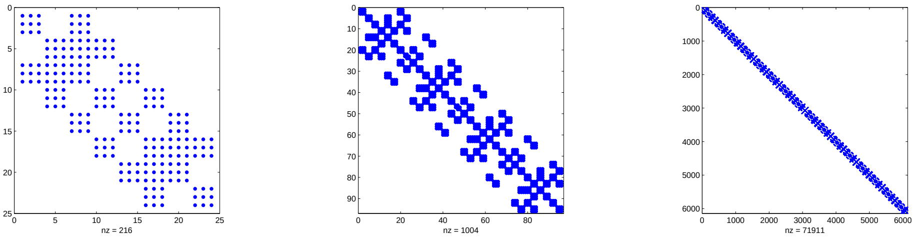

Figure 3.13: Stiffness matrices after *Downwind Numbering* on meshes of size $h = 2^{-1}$, $h = 2^{-2}$ and $h = 2^{-5}$, from left to right.

In the above applications of multigrid method to solution of the convection-diffusion equation (3.21) with dominating convection, the *Downwind Numbering* algorithm was efficient in decreasing the number of multigrid V-cycle iterations. This was an expected result as the *Downwind Numbering* algorithm managed to gather the domi-

Table 3.2: Number of iterations required to reach a relative residual of 10^{-8} with (top) and without (bottom) reordering.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	7	4	3	2	2	2	2	1	1	1
k=2	7	4	3	2	2	2	2	1	1	1
k=3	10	5	4	3	2	2	2	2	2	1
k=4	15	9	6	5	4	3	3	3	2	2

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	9	5	3	2	2	2	2	1	1	1
k=2	11	5	3	3	2	2	2	2	1	1
k=3	16	7	5	4	3	3	2	2	2	2
k=4	25	13	9	7	5	4	4	3	3	3

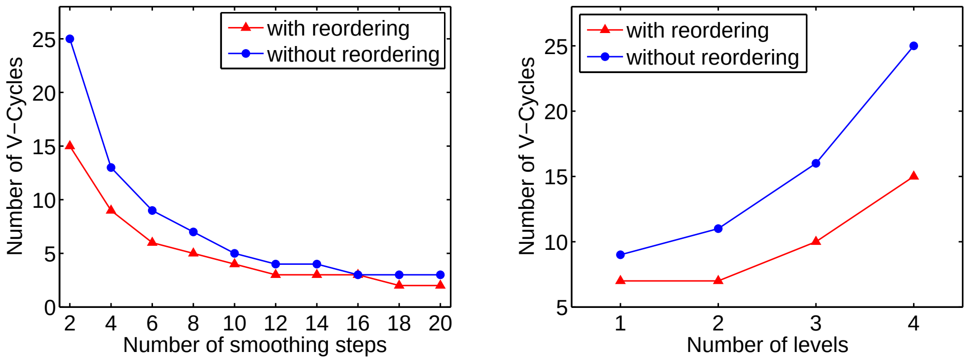

Figure 3.14: The required number of V-cycles to reach the desired relative residual of 10^{-8} for varying number of smoothing steps with fixed mesh size of 2^{-5} (left), and for varying number of levels aside from the finest one with fixed number of smoothing steps that is given as 2 (right), for $\beta = (-2, -1)^T$.

nating entries in the lower triangular part of the stiffness matrices on each mesh, via transforming the convection matrices into lower triangular matrices, causing the pointwise forward Gauss-Seidel smoother to perform more efficiently, especially for smaller number of smoothing steps.

CHAPTER 4

OPTIMAL CONTROL PROBLEMS

In recent years, linear-quadratic optimal control problems (OCPs) proved themselves important through their use in many real life applications such as the optimal control of systems [39], the shape optimization of technological devices [58] and flow control problems [30, 37, 62]. In order to solve the linear system of equations that results from the optimality conditions for these problems, a combined usage of the theory of optimization, the theory of PDEs and the tools of numerical mathematics, for the implementation part, is required.

In the last decade, discontinuous Galerkin (DG) methods have became popular for the optimal control problems governed by convection-diffusion equations due to their superiority over other discretization methods, as discussed in Chapter 2. The results in [56] show that DG discretizations enjoy a better convergence behavior for convection dominated optimal control problems since optimal convergence orders are obtained if the error is computed away from boundary or interior layers, in contrast to the streamline upwind Petrov Galerkin (SUPG) stabilized finite element discretization [45]. Also, a DG discretization have been used in [2, 75, 80, 82, 83, 84, 85] for the linear-quadratic optimal control problems .

In this chapter, we state that the optimal control problem governed by convection-diffusion-reaction PDE has a unique solution and then discuss the derivation of the optimality conditions of the optimization problem. Further, we provide the symmetric interior penalty Galerkin (SIPG) discretization of the optimality conditions. Later on, we redefine the transfer operators and smoothing operators of a multigrid method for the optimal control problems governed by convection-diffusion-reaction PDEs, and in the last part we present numerical results.

4.1 Introduction

We consider the following linear-quadratic optimal control problem:

$$\text{minimize } J(y,u) := \frac{1}{2}\int_\Omega (y(x) - y_d(x))^2 \, dx + \frac{\omega}{2}\int_\Omega u(x)^2 \, dx \qquad (4.1)$$

subject to

$$\begin{aligned} -\epsilon\Delta y(x) + \beta(x)\cdot\nabla y(x) + r(x)y(x) &= f(x) + u(x), && x \in \Omega \\ y(x) &= g_D(x), && x \in \Gamma^D, \\ \epsilon\nabla y(x)\cdot\mathbf{n}(x) &= g_N(x), && x \in \Gamma^N, \end{aligned} \tag{4.2}$$

where Ω is a bounded , open and convex domain in $\mathbb{R}^2$ with boundary $\partial\Omega = \Gamma^D \cup \Gamma^N$, such that $\Gamma^D \cap \Gamma^N = \emptyset$, and Γ^D and Γ^N stand for Dirichlet boundary and Neumann boundary, respectively. Further, let f, β, y_d, g_D, and g_N be given functions, $\epsilon, \omega > 0$ be given scalars and **n** be the normal vector to the boundary. The unknowns, y and u, in (4.1)-(4.2) are referred to as the state and control variables. The equation (4.2) is also referred as the state equation.

We first define the state space Y, the control space U, and the test functions space V as

$$\begin{aligned} Y &= \{y \in H^1(\Omega) : y = g_D \text{ on } \Gamma^D\}, \\ U &= L^2(\Omega), \\ V &= \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma^D\}. \end{aligned}$$

Then, the weak formulation of the state equation (4.2) can be stated such that: find $y \in Y$ for any fixed $u \in U$

$$a(y, v) + b(u, v) = (f, v) + (g_N, v)_{\Gamma^N}, \quad \forall v \in V, \tag{4.3}$$

where the bi-(linear) forms are defined by

$$\begin{aligned} a(y, v) &:= \int_\Omega (\epsilon\nabla y\cdot\nabla v + \beta\cdot\nabla y v + ryv)\, dx, \\ b(u, v) &:= -\int_\Omega uv\, dx, \\ (f, v) &:= \int_\Omega fv\, dx, \quad (g_N, v)_{\Gamma^N} := \int_{\Gamma^N} g_N v\, ds. \end{aligned}$$

We can now rewrite the optimization problem (4.1)-(4.2) using the weak formulation (4.3) of the state equation (4.2) as

$$\text{minimize } J(y, u) := \frac{1}{2}||y - y_d||^2_{L^2(\Omega)} + \frac{\omega}{2}||u||^2_{L^2(\Omega)} \tag{4.4}$$

subject to

$$\begin{aligned} &a(y, v) + b(u, v) = (f, v) + (g_N, v)_{\Gamma^N}, \qquad \forall v \in V, \\ &(y, u) \in Y \times U. \end{aligned} \tag{4.5}$$

In order to ensure the existence and uniqueness of the linear-quadratic optimal control problem governed by convection-diffusion equation, we make the following assumptions for the functions and parameters in the optimization problem (4.1)-(4.2):

$$\begin{aligned} f, y_d \in L^2(\Omega), g_D \in H^{3/2}(\Gamma^D), g_N \in H^{1/2}(\Gamma^N), 0 < \epsilon, \\ \beta(x) \in W^{1,\infty}(\Omega)^2, 0 < \omega, r \in L^\infty(\Omega), \end{aligned} \tag{4.6}$$

$$\Gamma^N \subset \{x \in \partial\Omega : \beta(x) \cdot \mathbf{n}(x) \geq 0\}, \tag{4.7}$$

and

$$r - \frac{1}{2}\nabla \cdot \beta(x) \geq r_0 \geq 0, \qquad x \in \Omega. \tag{4.8}$$

It can be derived by the standard techniques (see, e.g., [57, Sect. II.1]) that the control problem (4.4)-(4.5) has a unique solution $(y, u) \in (Y, U)$ under the assumptions (4.6-4.8).

We need the following Lagrangian functional

$$\mathcal{L}(y, u, p) = \frac{1}{2}||y - y_d||^2_{L^2(\Omega)} + \frac{\omega}{2}||u||^2_{L^2(\Omega)} + a(y, p) + b(u, p) - (f, p) - (g_N, v)_{\Gamma^N}, \tag{4.9}$$

to provide the necessary and sufficient optimality conditions of the optimization problem (4.4)-(4.5) as derived in [57, Sect. II.1].

Then, setting the partial Fréchet-derivatives of Lagrangian functional (4.9) with respect to the state y, control u, and adjoint p equal to zero, we obtain the following optimality system:

$$a(\psi, p) = -((y - y_d), \psi) \qquad \forall \psi \in V, \tag{4.10a}$$

$$b(w, p) + \omega(u, w) = 0 \qquad \forall w \in U, \tag{4.10b}$$

$$a(y, v) + b(u, v) = (f, v) + (g_N, v)_{\Gamma^N} \qquad \forall v \in V, \tag{4.10c}$$

which consists of the adjoint equation, the optimality condition, and the state equation, respectively.

Applying Green's Identities [73] and integration by parts to the adjoint equation (4.10a), we obtain

$$\begin{aligned} -((y - y_d), \psi) &= a(\psi, p) \\ &= \int_\Omega (\epsilon \nabla\psi \cdot \nabla p + \beta \cdot \nabla\psi p + r\psi p)\, dx \\ &= -\int_\Omega \epsilon p \Delta\psi\, dx + \int_{\partial\Omega} \epsilon p(\nabla\psi \cdot \mathbf{n})\, ds + \int_\Omega (\beta \cdot \nabla\psi p + r\psi p) dx \\ &= -\int_\Omega \epsilon\psi \Delta p\, dx + \int_{\partial\Omega} \epsilon\psi(\nabla p \cdot \mathbf{n})\, ds + \int_\Omega (\beta \cdot \nabla\psi p + r\psi p) dx \\ &= -\int_\Omega \epsilon\psi \Delta p\, dx + \int_{\partial\Omega} \epsilon\psi(\nabla p \cdot \mathbf{n})\, ds + \int_{\partial\Omega} (\beta \cdot \mathbf{n})\psi p\, ds \\ &\quad - \int_\Omega \psi\beta \cdot \nabla p\, dx - \int_\Omega \nabla \cdot \beta p\psi\, dx + \int_\Omega r\psi p\, dx, \quad \forall \psi \in V, \end{aligned} \tag{4.11}$$

which is the weak form of a convection-diffusion-reaction equation with the convection term $-\beta$ given as

$$
\begin{aligned}
-\epsilon\Delta p(x) - \beta(x)\cdot\nabla p(x) + (r(x) - \nabla\cdot\beta(x))p(x) &= -(y(x) - y_d(x)) && x \in \Omega, \\
p(x) &= 0, && x \in \Gamma^D, \\
\epsilon\nabla p(x)\cdot\mathbf{n}(x) + \beta(x)\cdot\mathbf{n}(x)p(x) &= 0, && x \in \Gamma^N.
\end{aligned}
$$

Further, the optimality condition (4.10b) corresponds to

$$p(x) = \omega u(x), \qquad x \in \Omega.$$

Finally, we can state the following theorem given in [56, Sect. 2]:

Theorem 4.1. *If the assumptions (4.6)-(4.8) are satisfied, then the optimal control problem (4.4)-(4.5) has a unique solution* $(y,u) \in Y \times U$*. Further, the pair* $(y,u) \in Y \times U$ *solves (4.4)-(4.5) if and only if the triple* $(y,u,p) \in Y \times U \times V$ *is the unique solution of the optimality system*

$$
\begin{aligned}
a(\psi,p) &= -((y-y_d),\psi) && \forall\psi \in V, \\
b(w,p) + \omega(u,w) &= 0 && \forall w \in U, \\
a(y,v) + b(u,v) &= (f,v) + (g_N,v)_{\Gamma^N} && \forall v \in V.
\end{aligned}
$$

To solve an optimal control problem, there mainly exist two approaches, which are called the discretize-then-optimize (DO) and the optimize-then-discretize (OD). For the DO approach, as the name suggests, the optimal control problem is discretized first and then the finite dimensional optimality system is formed. On the other hand, for the OD approach, the system of optimality conditions consisting of the state equation, the adjoint equation, and the optimality condition are firstly formed on the continuous level, then the resulting system is discretized. Both approaches are studied and compared in [1, 10, 20, 21, 29, 31, 47, 81, 83] for the optimal control problems governed by the convection-diffusion equation by using different kind of discretization methods.

It is known that for local projection based stabilization [10] and edge stabilization [81] of continuous Galerkin discretization, the linear systems of equations resulting from both approaches commute. On the other, there exist other discretization schemes, such as the streamline upwind Petrov-Galerkin method [29, 56], where the resulting linear system of equations do not commute. It is recently shown in [83] that the symmetric discontinuous Galerkin methods, for example SIPG method, yield the same scheme for both approaches, while the nonsymmetric discontinuous Galerkin methods yield different schemes. This motivates us to use only the discretize-then-optimize approach with the SIPG discretization, in the rest of the thesis.

4.2 Discretize-then-Optimize Approach

The discretization of the state equation, based on the symmetric interior penalty Galerkin (SIPG) method, is described in Chapter 2. We now extend the discussion to the optimal control problem by following the discretize-then-optimize approach.

We first define the discrete state space Y_h, the discrete control space U_h and the space of test functions V_h as

$$\begin{aligned} V_h = Y_h &= \{y \in L^2(\Omega) : y|_T \in \mathbb{P}_n(T), \forall T \in \mathcal{T}_h\}, \\ U_h &= \{u \in L^2(\Omega) : u|_T \in \mathbb{P}_m(T), \forall T \in \mathcal{T}_h\}. \end{aligned}$$

Notice that the orders $n, m \in \mathbb{N}$ of the finite element approximation need not be equal, however, in this thesis we will be using $n = m$, hence, $V_h = Y_h = U_h$. Note that, as in the Section 2.2, the state function space Y_h and the test function space V_h are chosen to be identical due to the weak treatment of the boundary conditions in DG framework. Then, the discretized optimal control problem is given as

$$\text{minimize } J(y_h, u_h) := \frac{1}{2} \sum_{T \in \mathcal{T}_h} ||y_h - y_d||_T^2 + \frac{\omega}{2} \sum_{T \in \mathcal{T}_h} ||u||_T^2 \tag{4.12}$$

subject to

$$\begin{aligned} &a_h(y_h, v_h) + b_h(u_h, v_h) = l_h(v_h) + (g_N, v_h)_{\Gamma^N}, \qquad \forall v_h \in V_h, \\ &(y_h, u_h) \in Y_h \times U_h, \end{aligned} \tag{4.13}$$

where

$$\begin{aligned} a_h(y_h, v_h) =& \sum_{T \in \mathcal{T}_h} \int_T \epsilon \nabla y_h \cdot \nabla v_h \, dx + \sum_{T \in \mathcal{T}_h} \int_T (\beta \cdot \nabla y_h v_h + r y_h v_h) \, dx \\ &- \sum_{e \in \mathcal{E}_h} \int_e \{\epsilon \nabla v_h\} \cdot [y_h] \, ds - \sum_{e \in \mathcal{E}_h} \int_e \{\epsilon \nabla y_h\} \cdot [v_h] \, ds \\ &+ \sum_{T \in \mathcal{T}_h} \int_{\partial T^- \setminus \partial\Omega} (\beta \cdot \mathbf{n})(y_h^{out} - y_h^{in}) v_h^{in} \, ds - \sum_{T \in \mathcal{T}_h} \int_{\partial T^- \cap \Gamma^-} (\beta \cdot \mathbf{n}) y_h^{in} v_h^{in} \, ds, \\ &+ \sum_{e \in \mathcal{E}_h} \frac{\sigma\epsilon}{h_e} \int_e [y_h] \cdot [v_h] \, ds \\ b_h(u_h, v_h) =& - \sum_{T \in \mathcal{T}_h} \int_T u_h v_h \, dx, \\ l_h(v_h) =& \sum_{T \in \mathcal{T}_h} \int_T f v_h \, dx + \sum_{e \in \mathcal{E}_h^D} \int_e g^D (\frac{\sigma\epsilon}{h_e} v_h - \epsilon \nabla v_h \cdot \mathbf{n}) \, ds \\ &- \sum_{T \in \mathcal{T}_h} \int_{\partial T^- \cap \Gamma^-} (\beta \cdot \mathbf{n}) g^D v_h^{in} \, ds. \end{aligned}$$

Setting the partial Frechét derivatives of the Lagrangian of the discretized problem (4.12)-(4.13)

$$\begin{aligned} \mathcal{L}_h(y_h, u_h, p_h) =& \frac{1}{2} \sum_{T \in \mathcal{T}_h} ||y_h - y_d||_T^2 + \frac{\omega}{2} \sum_{T \in \mathcal{T}_h} ||u_h||_T^2 + a_h(y_h, p_h) + b_h(u_h, p_h) \\ &- l_h(p_h) - (g_N, p_h)_{\Gamma^N}, \end{aligned}$$

with respect to discrete state, control and adjoint variables equal to zero,

$$\begin{aligned}\nabla_y \mathcal{L}_h(y_h, u_h, p_h) = 0,\\ \nabla_u \mathcal{L}_h(y_h, u_h, p_h) = 0,\\ \nabla_p \mathcal{L}_h(y_h, u_h, p_h) = 0,\end{aligned}$$

we obtain the following discrete optimality system

$$\begin{aligned} a_h(\psi_h, p_h) &= -(y_h - y_d, \psi_h), && \forall \psi_h \in V_h,\\ b_h(w_h, p_h) + \omega(u_h, w_h) &= 0, && \forall w_h \in U_h,\\ a_h(y_h, u_h) + b_h(u_h, v_h) &= l_h(v_h) + (g_N, v_h)_{\Gamma^N}, && \forall v_h \in V_h,\end{aligned}$$

that consists of the discrete adjoint equation, the discrete optimality condition and the discrete state equation, respectively.

To reformulate the optimization problem (4.12)-(4.13) in terms of matrix-vector form, we can write the discrete state and control spaces as: for each element $T \in \mathcal{T}_h$,

$$\begin{aligned} Y_{h,T} &= span\{\varphi_i^T : 1 \leq i \leq nloc, T \in \mathcal{T}_h\},\\ U_{h,T} &= span\{\psi_i^T : 1 \leq i \leq nloc, T \in \mathcal{T}_h\},\end{aligned}$$

where $\left\{\varphi_i^T\right\}_{i=1}^{nloc}$ is the basis functions for the state space, $\left\{\psi_i^T\right\}_{i=1}^{nloc}$ is the basis functions for the control space, and *nloc* is the local dimension. This way of defining the state and control spaces lets us to rewrite the state y and the control u such that

$$\begin{aligned} y(x) &= \sum_{m=1}^{nel}\sum_{j=1}^{nloc} y_j^m \varphi_j^m(x),\\ u(x) &= \sum_{m=1}^{nel}\sum_{j=1}^{nloc} y_j^m \psi_j^m(x),\end{aligned} \tag{4.14}$$

where *nel* is the number of triangles in the underlying triangulation $\mathcal{T}_h$ of Ω. Inserting (4.14) into (4.12)-(4.13), we obtain the following discretized optimization problem:

$$\text{minimize } J(\overrightarrow{y}, \overrightarrow{u}) := \frac{1}{2}\overrightarrow{y}^T \mathbb{M}\overrightarrow{y} - \overrightarrow{b}^T \overrightarrow{y} + \frac{\omega}{2}\overrightarrow{u}^T \mathbb{Q}\overrightarrow{u} + \int_\Omega \frac{1}{2} y_d^2 \, dx \tag{4.15}$$

subject to

$$\mathbb{A}\overrightarrow{y} + \mathbb{B}\overrightarrow{u} = \overrightarrow{f}, \tag{4.16}$$

where $\mathbb{A}$ and $\overrightarrow{f}$ correspond to the bilinear form $a_h(y_h, v_h)$ and the linear form $l_h(v_h)$ in (4.13), $\mathbb{M}, \mathbb{Q}, \mathbb{B} \in \mathbb{R}^{(nloc*nel)\times(nloc*nel)}$, $\overrightarrow{b} \in \mathbb{R}^{nloc*nel}$, $\overrightarrow{y}$ and $\overrightarrow{u}$ are given as

$$\begin{aligned} \overrightarrow{y} &= (y_1^1, y_2^1, \ldots, y_{nloc}^1, y_1^2, y_2^2, \ldots, y_{nloc}^2, \ldots, y_1^{nel}, y_2^{nel}, \ldots, y_{nloc}^{nel})^T,\\ \overrightarrow{u} &= (u_1^1, u_2^1, \ldots, u_{nloc}^1, u_1^2, u_2^2, \ldots, u_{nloc}^2, \ldots, u_1^{nel}, u_2^{nel}, \ldots, u_{nloc}^{nel})^T.\end{aligned}$$

Additionally, the matrices $\mathbb{M}$, $\mathbb{Q}$, $\mathbb{B}$ and the vector $\overrightarrow{b}$ are defined as

$$(\mathbb{M})_{ij} = \int_T \varphi_j \varphi_i \, dx, \quad (\mathbb{B})_{ij} = -\int_T \varphi_j \psi_i \, dx, \quad (\mathbb{Q})_{ij} = \int_T \psi_j \psi_i \, dx,$$
$$(\overrightarrow{b})_i = \int_T y_d \varphi_i \, dx.$$

Then, the necessary and sufficient optimality conditions for the optimal control problem (4.15)-(4.16) are

$$\begin{aligned} \mathbb{M}\overrightarrow{y} + \mathbb{A}^T \overrightarrow{p} &= \overrightarrow{b}, \\ \omega \mathbb{Q} \overrightarrow{u} + \mathbb{B}^T \overrightarrow{p} &= 0, \\ \mathbb{A}\overrightarrow{y} + \mathbb{B}\overrightarrow{u} &= \overrightarrow{f}, \end{aligned}$$

which can be written in the matrix form as

$$\begin{bmatrix} \mathbb{M} & 0 & \mathbb{A}^T \\ 0 & \omega\mathbb{Q} & \mathbb{B}^T \\ \mathbb{A} & \mathbb{B} & 0 \end{bmatrix} \begin{bmatrix} \overrightarrow{y} \\ \overrightarrow{u} \\ \overrightarrow{p} \end{bmatrix} = \begin{bmatrix} \overrightarrow{b} \\ 0 \\ \overrightarrow{f} \end{bmatrix}. \tag{4.17}$$

Such large systems can not be solved using direct numerical techniques. Therefore, it is crucial to have efficient iterative methods for solving these optimality systems. We here investigate multigrid methods applied to the saddle system (4.17) obtained from a symmetric interior penalty Galerkin discretization of the optimal control problems governed by convection-diffusion equation.

Multigrid methods have been investigated in [7, 12, 13, 15, 17, 18, 19, 36, 49, 51, 55, 59, 72] for several different types of optimal control problems. However, it is not always easy to derive an efficient approach for the optimal control problems governed by convection-diffusion equations due to the structure of the optimality systems. The optimality system involves two PDEs; the state equation with convection β and the adjoint equation with convection $-\beta$. In the following section, we discuss the multigrid method for the optimal control problems.

4.3 Multigrid Methods for Optimal Control Problems

We here solve the optimality system (4.17) by using the multigrid approach, as done in the previous section for the state equation. However, due to the structure of the optimality system we need to modify the transfer and smoothing operators.

The vector x consists of 3 blocks of size $(nloc \times nel) \times 1$, where $nloc$ is the local dimension and nel is the number of triangles in the underlying triangulation of the domain. The block on top corresponds to the state variable, the block in the middle corresponds to the control variable and the block on the bottom corresponds to the adjoint variable, which are all discretized on the same mesh. As defined in Section 3.1, the restriction operator for each of these blocks is I_h^H, where the subscript h and the superscript H stand for the fine and the coarse triangulations of the domain, respectively. Hence, the optimal control restriction operator can be defined as

$$\mathcal{I}_h^H = \begin{bmatrix} I_h^H & 0 & 0 \\ 0 & I_h^H & 0 \\ 0 & 0 & I_h^H \end{bmatrix}.$$

As we have done for the prolongation operator, the optimal control prolongation operator is defined to be the transpose of the optimal control restriction operator, that is,

$$\mathcal{I}_H^h = (\mathcal{I}_h^H)^T.$$

The system (4.17) that we are trying to solve has zero elements on its diagonal, thus, we cannot apply the pointwise smoothing operators that were defined in the Section 3.2. In order to overcome this obstacle, we adopt the idea of a block Gauss-Seidel method described in [49].

We begin with grouping the state, the control and the adjoint variables for any given triangle to a vector. For the k^{th} triangle, the corresponding state variables start at the entry $(k-1)*nloc+1$ and end at the entry $k*nloc$, the corresponding control variables start at the entry $nloc*N+(k-1)*nloc+1$ and end at the entry $nloc*nel+k*nloc$ and the corresponding adjoint variables start at the entry $2*nloc*nel+(k-1)*nloc+1$ and end at the entry $2*nloc*nel+k*nloc$. For any triangle,we group the corresponding entries such that they form a vector of size $3*nloc \times 1$, where the state variables are on the top, the control variables are in the middle and the adjoint variables are on the bottom, and name this vector x_k. Repeating this procedure for all nel triangles, we get the vectors $x_1, \dots, x_N$, and we can form the new block vector x^{block} defined as

$$x^{block} = (x_1, x_2, \dots, x_{nel})^T.$$

Similarly, we can form the block vector g^{block} defined as

$$g^{block} = (g_1, g_2, \dots, g_{nel})^T,$$

and the block matrix K^{block} defined as

$$K^{block} = \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1nel} \\ K_{21} & K_{22} & \dots & K_{2nel} \\ \vdots & \vdots & \ddots & \vdots \\ K_{nel1} & K_{nel2} & \dots & K_{nelnel} \end{bmatrix},$$

where $K_{i,j}$ is a $3*nloc \times 3*nloc$ matrix that is obtained via the same procedure that we used to define x^{block} and g^{block}, but this time for two dimensions at the same time.

Then, we can define the *block forward Gauss-Seidel* iteration as

```
For $i = 1 : nel$
    $x_i^{(m+1)} = \frac{1}{K_{ii}} \left( g_i - \sum_{j=1}^{i-1} K_{ij} x_j^{(m+1)} - \sum_{j=i+1}^{nel} K_{ij} x_j^{(m)} \right)$
End
```

As we have done in section 3.2, we can define the *block backward Gauss-Seidel* iteration as

```
For $i = N : 1$
    $x_i^{(m+1)} = \frac{1}{K_{ii}} \left( g_i - \sum_{j=1}^{i-1} K_{ij} x_j^{(m)} - \sum_{j=i+1}^{nel} K_{ij} x_j^{(m+1)} \right)$
End
```

4.4 Numerical Results

In this section we discuss some numerical results for the optimal control problem governed by convection-diffusion PDEs by applying the multigrid methods. Then, we will also apply the *Downwind Numbering* algorithm discussed in chapter 3 for single state equations. Although the algorithm works well for single equations, especially for small smoothing iteration numbers, it does not address the same effectivity for the optimal control problems due to the structure of the optimality system.

In numerical examples, we use five levels of meshes where the coarsest and finest meshes have 8 and 2048 triangular elements, respectively. The penalty parameter in the SIPG method is chosen as $\sigma = 6$ on the interior edges and $\sigma = 12$ on the boundary edges. The stopping criterion is chosen as the ratio of the difference in error of the last two consecutive state variable estimates to the error of the initial state variable being less than 10^{-8}.

We consider an optimal control problem of the form:

$$\min \ \frac{1}{2} ||y - y_d||^2_{L^2(\Omega)} + \frac{\omega}{2} ||u||^2_{L^2(\Omega)}$$

subject to

$$\begin{aligned} -\epsilon \Delta y + \beta \cdot \nabla y + ry &= f + u, && \text{in} \quad \Omega, \\ y &= 0, && \text{on} \quad \partial\Omega, \end{aligned}$$

where $\Omega = (0, 1)^2$, $\beta = (1/\sqrt{2}, 1/\sqrt{2})^T$, $r = 0$, and $\omega = 10^{-2}$. The source function f and the desired state y_d are chosen such that the optimal state y, the optimal control u, and the optimal adjoint p satisfy

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

where

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

We test the multigrid algorithm for various diffusion parameter ϵ. In the first scenario, we chose the parameter as $\epsilon = 1$. We compare the results from applying different types of smoothers in the multigrid algorithm. Table (4.1)-(4.2 display the iteration number of multigrid for different pre- and post-smoothing steps. The refinement level and number of smoothing steps are denoted by k and m. Block forward Gauss-Seidel as a pre-smoother and block backward Gauss-Seidel as a post-smoother yields slightly better results. It is an expected result for symmetric smoothing, i.e., a forward block Gauss-Seidel as a pre-smoother and a backward block Gauss-Seidel as a post-smoother [49]. Figure 4.1 illustrates the optimal state solution and the optimal control solution obtained via multigrid method with 2 steps of forward block Gauss-Seidel as a pre-smoother and 2 steps of backward block Gauss-Seidel as a post-smoother on the mesh with 2048 triangular elements, where $\epsilon = 1$.

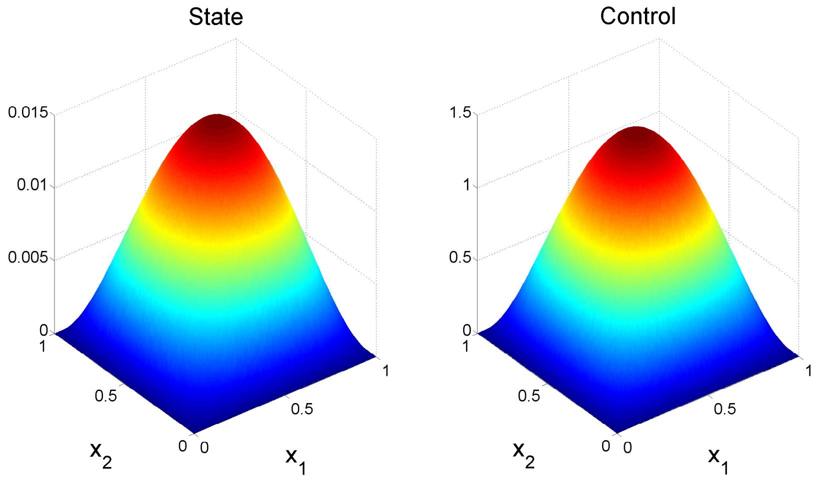

Figure 4.1: The optimal state solution and the optimal control solution obtained via multigrid method with 2 steps of forward block Gauss-Seidel as a pre-smoother and 2 steps of backward block Gauss-Seidel as a post-smoother on the mesh with 2048 triangular elements.

Table 4.1: The number of V-cycle iterations as functions of refinement level k and number of smoothing steps m. The pre-smoothing and post-smoothing are chosen as block forward Gauss-Seidel and block backward Gauss-Seidel, respectively.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	20	13	11	9	8	7	8	7	7	6
k=2	23	14	12	10	9	9	8	8	8	7
k=3	14	14	11	10	9	9	8	8	8	7
k=4	16	12	11	9	9	8	8	7	7	7

Table 4.2: The number of V-cycle iterations as functions of refinement level k and number of smoothing steps m. Both pre- and post-smoothing are block forward Gauss-Seidel.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	23	16	15	13	12	10	9	9	8	7
k=2	31	19	16	14	13	12	11	10	9	9
k=3	34	20	16	14	13	12	11	11	10	10
k=4	35	19	16	14	12	11	11	10	10	10

We now apply the Downwind Numbering algorithm [11] for the optimal control problems. Table (4.3)-(4.4) show the numerical results of the algorithm for different pre-post-smoothing. However, we observe that there is no gain of the downwind algorithm for optimal control problems.

Table 4.3: The number of V-cycle iterations for *Downwind Numbering* as functions of refinement level k and number of smoothing steps m. The pre-smoothing and post-smoothing are chosen as block forward Gauss-Seidel and block backward Gauss-Seidel, respectively.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	27	17	15	13	12	10	9	9	8	7
k=2	37	21	16	13	12	11	10	10	10	9
k=3	43	22	17	14	13	12	11	11	10	10
k=4	44	21	16	14	13	12	11	10	10	10

Table 4.4: The number of V-cycle iterations for *Downwind Numbering* as functions of refinement level k and number of smoothing steps m. Both pre- and post-smoothing are block forward Gauss-Seidel.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	24	15	15	13	12	10	9	9	8	7
k=2	28	19	16	14	13	10	11	10	10	9
k=3	28	16	15	13	12	12	11	11	10	10
k=4	22	14	14	13	12	11	11	10	10	10

In the second scenario, we chose the diffusion parameter as $\epsilon = 10^{-3}$. Similar results are displayed in Table (4.5-4.8) for various pre- post-smoothing. We observe that the iteration number of multigrid algorithm decreases for small number of diffusion parameter. There is also a slight benefit of preserving the symmetry in smoothers applied within multigrid solvers. Further, the proposed downwind algorithm in chapter 3 does not produce better results as expected for the optimal control problems. Figure 4.2 illustrates the optimal state solution and the optimal control solution obtained via multigrid method with 2 steps of forward block Gauss-Seidel as a pre-smoother and 2 steps of backward block Gauss-Seidel as a post-smoother on the mesh with 2048 triangular elements, where $\epsilon = 10^{-3}$. Notice that, both the state and the control solutions exhibit boundary layers in Figure 4.2, while being smoother in the interior part. In order to achieve smoother approximations on the boundary, we need to refine our mesh even more; however, using uniform refinement also results in refinement for the triangles that the approximation is close enough to the exact solution, which is unnecessary. That is why, for problems with boundary layers, the adaptive mesh refinement [82, 84] can be an efficient alternative to uniform refinement, in order to prevent the unnecessary increase in the degrees of freedom.

Table 4.5: The number of V-cycle iterations as functions of refinement level k and number of smoothing steps m. The pre-smoothing and post-smoothing are chosen as block forward Gauss-Seidel and block backward Gauss-Seidel, respectively.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	11	7	5	4	4	3	3	3	3	3
k=2	12	9	7	6	5	5	4	4	4	3
k=3	11	6	6	7	7	6	6	5	5	4
k=4	10	6	7	5	7	6	5	6	5	6

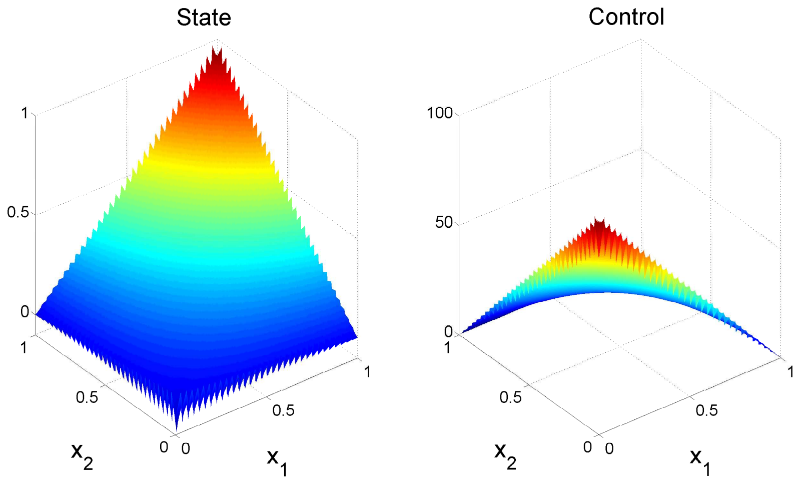

Figure 4.2: The optimal state solution and the optimal control solution obtained via multigrid method with 2 steps of forward block Gauss-Seidel as a pre-smoother and 2 steps of backward block Gauss-Seidel as a post-smoother on the mesh with 2048 triangular elements, where $\epsilon = 10^{-3}$.

Table 4.6: The number of V-cycle iterations as functions of refinement level k and number of smoothing steps m. Both pre- and post-smoothing are block forward Gauss-Seidel.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	16	9	6	5	4	3	4	3	3	3
k=2	17	12	10	8	6	6	6	5	5	4
k=3	22	12	9	9	9	8	7	7	7	6
k=4	23	14	10	9	9	8	8	8	7	8

Table 4.7: The number of V-cycle iterations for *Downwind Numbering* as functions of refinement level k and number of smoothing steps m. The pre-smoothing and post-smoothing are chosen as block forward Gauss-Seidel and block backward Gauss-Seidel, respectively.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	13	8	5	5	4	3	3	3	3	2
k=2	19	8	7	6	5	4	4	4	4	4
k=3	100	7	7	6	5	5	5	5	5	5
k=4	30	7	8	6	5	5	5	5	5	4

Table 4.8: The number of V-cycle iterations for *Downwind Numbering* as functions of refinement level k and number of smoothing steps m. Both pre- and post-smoothing are block forward Gauss-Seidel.

	m=2	m=4	m=6	m=8	m=10	m=12	m=14	m=16	m=18	m=20
k=1	17	9	6	5	4	4	4	3	3	3
k=2	30	12	8	7	6	6	5	5	4	4
k=3	22	10	9	7	7	6	5	5	4	4
k=4	16	11	7	7	7	6	6	6	6	6

CHAPTER 5

CONCLUSION AND OUTLOOK

In this thesis we have studied the development and application of multigrid methods for linear-quadratic elliptic convection dominated optimal control problems using discontinuous Galerkin (DG) methods. We chose DG methods for their superiority over other discretization methods in terms of geometric flexibility, the allowance for different order of approximation to be used on different elements in the mesh in a straightforward manner, and achieving higher-order accuracy through the local elements. However, the drawback of the DG methods is the increase in the total degrees of freedom, i.e., larger systems to be solved. In order to deal with this drawback, we chose multigrid methods as our solver, as such methods proved themselves as efficient iterative solvers for elliptic partial differential equations (PDEs).

In order to reduce the required number of multigrid V-cycle iterations for solutions of convection-diffusion PDEs with dominating convection, we adopted a reordering technique from [11], called *Downwind Numbering*. We presented the implementation of *Downwind Numbering* algorithm for systems resulting from DG discretizations. Using *Downwind Numbering* algorithm we ordered the stiffness matrix of convection-diffusion equations such that the dominating entries appeared exclusively in the lower triangular part. It is well known that for a system $Ax = b$, where A is a lower triangular matrix, one step of Gauss-Seidel iteration is an exact solver for any initial approximation. That is why *Downwind Numbering* algorithm performed efficiently for the solutions of the convection-diffusion equations, as shown in Section 3.5. However, we did not discuss convection-diffusion equations with cycles in convection matrices since they also require an implementation of graph theory.

We then extended the discussion of multigrid methods to optimal control problems by redefining the transfer operators and smoothing operators to suit discrete optimality systems that result from discretizations of the optimal control problems governed by convection-diffusion equations. However, for the optimal control problems, the multigrid method coupled with *Downwind Numbering* algorithm did not work as efficiently as they did for single convection-diffusion equations. One of the reason for this inefficiency was two PDEs appearing in the optimality systems; the state equation with convection β and the so-called adjoint equation with convection $-\beta$, which led to two different triangular element orderings that are the reverses of each other. Further, preserving symmetry in smoothing operators within multigrid iterations, i.e., using block forward Gauss-Seidel as pre-smoother and block backward Gauss-Seidel as post-smoother, proved itself slightly favorable over using block forward Gauss-Seidel as both pre- and post-smoother. The reason why for this result can be the fact that the optimality system being symmetric.

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., 41(3), pp. 229–251, 2004.

[2] T. Akman, H. Yücel, and B. Karasözen, A priori error analysis of the upwind symmetric interior penalty Galerkin (SIPG) method for the optimal control problems governed by unsteady convection diffusion equations, Comput. Optim. Appl., 57(3), pp. 703–729, 2014.

[3] D. N. Arnold, An interior penalty finite element method with discontinuous elements, SIAM J. Numer. Anal., 19(4), pp. 742–760, 1982.

[4] 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(5), pp. 1749–1779, 2002.

[5] B. Ayuso and L. D. Marini, Discontinuous Galerkin methods for advection-diffusion-reaction problems, SIAM J. Numer. Anal., 47(2), pp. 1391–1420, 2009.

[6] I. Babuška and M. Zlámal, Nonconforming elements in the finite element method with penalty, SIAM J. Numer. Anal., 10(5), pp. 863–875, 1973.

[7] H. Baek, S. D. Kim, and H.-C. Lee, A multigrid method for an optimal control problem of a diffusion-convection equation, J. Korean Math. Soc, 47(1), pp. 83–100, 2010.

[8] 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(2), pp. 267–279, 1997.

[9] C. E. Baumann and J. T. Oden, A discontinuous hp finite element method for convection-diffusion problems, Comput. Methods Appl. Mech. Engrg., 175(3), pp. 311–341, 1999.

[10] R. Becker and B. Vexler, Optimal control of the convection-diffusion equation using stabilized finite element methods, Numer. Math., 106(3), pp. 349–367, 2007.

[11] J. Bey and G. Wittum, Downwind numbering: Robust multigrid for convection-diffusion problems, Appl. Numer. Math., 23(1), pp. 177–192, 1997.

[12] A. Borzì, Multigrid methods for parabolic distributed optimal control problems, J. Comput. Appl. Math., 157(2), pp. 365–382, 2003.

[13] A. Borzì, Smoothers for control-and state-constrained optimal control problems, Comput. Vis. Sci., 11(1), pp. 59–66, 2008.

[14] A. Borzì and G. Borzì, An algebraic multigrid method for a class of elliptic differential systems, SIAM J. Sci. Comput., 25(1), pp. 302–323, 2003.

[15] A. Borzì and G. Borzì, An efficient algebraic multigrid method for solving optimality systems, Comput. Vis. Sci., 7(3-4), pp. 183–188, 2004.

[16] A. Borzì and K. Kunisch, The numerical solution of the steady state solid fuel ignition model and its optimal control, SIAM J. Sci. Comput., 22(1), pp. 263–284, 2000.

[17] A. Borzì and K. Kunisch, A multigrid scheme for elliptic constrained optimal control problems, Comput. Optim. Appl., 31(3), pp. 309–333, 2005.

[18] A. Borzì, K. Kunisch, and D. Y. Kwak, Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system, J. Control Optim., 41(5), pp. 1477–1497, 2002.

[19] A. Borzì and V. Schulz, Multigrid methods for pde optimization, SIAM review, 51(2), pp. 361–395, 2009.

[20] M. Braack, Optimal control in fluid mechanics by finite elements with symmetric stabilization, SIAM J. Control Optim., 48(2), pp. 672–687, 2009.

[21] M. Braack, E. Burman, V. John, and G. Lube, Stabilized finite element methods for the generalized Oseen problem, Comput. Methods Appl. Mech. Engrg., 196(4), pp. 853–866, 2007.

[22] J. H. Bramble, *Multigrid methods*, volume 294, CRC Press, 1993.

[23] J. H. Bramble, J. E. Pasciak, and J. Xu, The analysis of multigrid algorithms for nonsymmetric and indefinite elliptic problems, Math. Comp., 51(184), pp. 389–414, 1988.

[24] S. Brenner, Convergence of the multigrid V-cycle algorithm for second-order boundary value problems without full elliptic regularity, Math. Comput., 71(238), pp. 507–525, 2002.

[25] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo, Discontinuous Galerkin approximations for elliptic problems, Numer. Meth. Part. D. E., 16(4), pp. 365–378, 2000.

[26] A. Cangiani, J. Chapman, E. H. Georgoulis, and M. Jensen, On local superpenalization of interior penalty discontinuous Galerkin methods, Int. J. Numer. Anal. Model., 11(3), pp. 478–495, 2014.

[27] P. Castillo, Performance of discontinuous Galerkin methods for elliptic pdes, SIAM J. Numer. Anal., 24(2), pp. 524–547, 2002.

[28] B. Cockburn and C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal., 35(6), pp. 2440–2463, 1998.

[29] S. S. Collis and M. Heinkenschloss, Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems, CAAM TR02-01, 2002.

[30] L. Dedè, S. Micheletti, and S. Perotto, Anisotropic error control for environmental applications, Appl. Numer. Math., 58(9), pp. 1320–1339, 2008.

[31] L. Dede and A. Quarteroni, Optimal control and numerical adaptivity for advection–diffusion equations, ESAIM: Math. Model. Num., 39(05), pp. 1019–1040, 2005.

[32] L. Demkowicz, *Computing with hp-adaptive finite elements: Volume 1 One and Two Dimensional Elliptic and Maxwell Problems*, Chapman & Hall/CRC Applied Mathematics & Nonlinear Science, CRC Press, 2006.

[33] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszenski, W. Rachowicz, and A. Zdunek, *Computing with hp-adaptive finite elements: Volume II Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*, Chapman & Hall/CRC Applied Mathematics & Nonlinear Science, CRC Press, 2007.

[34] J. Demmel, *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, 1997.

[35] J. Douglas and T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, in *Computing methods in applied sciences*, pp. 207–216, Springer, 1976.

[36] M. Engel and M. Griebel, A multigrid method for constrained optimal control problems, J. Comput. Appl. Math., 235(15), pp. 4368–4388, 2011.

[37] H. Fernando, S. Lee, J. Anderson, M. Princevac, E. Pardyjak, and S. Grossman-Clarke, Urban fluid mechanics: air circulation and contaminant dispersion in cities, Environ. Fluid Mech., 1(1), pp. 107–164, 2001.

[38] J. Gopalakrishnan and G. Kanschat, A multilevel discontinuous Galerkin method, Numer. Math., 95(3), pp. 527–550, 2003.

[39] M. D. Gunzburger, *Perspectives in flow control and optimization*, volume 5, SIAM, 2003.

[40] W. Hackbusch, On the feedback vertex set problem for a planar graph, Computing, 58(2), pp. 129–155, 1997.

[41] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 2003, ISBN 9783540127611.

[42] W. Hackbusch, R. Kriemann, S. Le Borne, and J.-F. Maitre, Cd2d3d-a package to solve convection dominated problems employing ordering techniques, in *Numerical Flow Simulation II*, pp. 34–48, Springer, 2001.

[43] W. Hackbusch and T. Probst, Downwind Gauss-Seidel smoothing for convection dominated problems, Numer. Linear Algebra Appl., 4(2), pp. 85–102, 1997.

[44] W. Hackbusch, U. Trottenberg, and editors, *Multigrid Methods*, volume 960 of *Lecture Notes in Mathematics*, Springer Berlin Heidelberg, 1982.

[45] 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(6), pp. 4607–4638, 2010.

[46] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54, Springer Science & Business Media, 2007.

[47] M. Hinze, N. Yan, and Z. Zhou, Variational discretization for optimal control governed by convection dominated diffusion equations, J. Comput. Math, 27(2-3), pp. 237–253, 2009.

[48] P. Houston, C. Schwab, and E. Süli, Discontinuous hp-finite element methods for advection-diffusion-reaction problems, SIAM J. Numer. Anal., 39(6), pp. 2133–2163, 2002.

[49] P. A. Howard, *Multigrid methods for elliptic optimal control problems*, Master's thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2006.

[50] C. M. Klaij, M. H. van Raalte, H. van der Ven, and J. J. van der Vegt, h-multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys., 227(2), pp. 1024–1045, 2007.

[51] O. Lass, M. Vallejos, A. Borzì, and C. Douglas, Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing, 84(1-2), pp. 27–48, 2009.

[52] S. Le Borne, *Multigrid methods for convection-dominated problems*, Ph.D. thesis, University of Kiel, Kiel, 1999.

[53] S. Le Borne, Ordering techniques for two-and three-dimensional convection-dominated elliptic boundary value problems, Computing, 64(2), pp. 123–155, 2000.

[54] P. Lesaint and P.-A. Raviart, 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, (33), pp. 89–123, 1974.

[55] R. M. Lewis and S. G. Nash, Model problems for the multigrid optimization of systems governed by differential equations, SIAM J. Sci. Comput., 26(6), pp. 1811–1837, 2005.

[56] D. Leykekhman and M. Heinkenschloss, Local error analysis of discontinuous Galerkin methods for advection-dominated elliptic linear-quadratic optimal control problems, SIAM J. Numer. Anal., 50(4), pp. 2012–2038, 2012.

[57] J. L. Lions, Optimal control of systems governed by partial differential equations problèmes aux limites, 1971.

[58] B. Mohammadi and Pironneau, *Applied shape optimization for fluids*, volume 28, Oxford University Press Oxford, 2001.

[59] S. G. Nash, A multigrid approach to discretized optimization problems, Optim. Method Softw., 14(1-2), pp. 99–116, 2000.

[60] M. A. Olshanskii and A. Reusken, Convergence analysis of a multigrid method for a convection-dominated model problem, SIAM J. Numer. Anal., 42(3), pp. 1261–1291, 2004.

[61] L. N. Olson and J. B. Schroder, Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems, J. Comput. Phys., 230(18), pp. 6959–6976, 2011.

[62] D. Parra-Guevara and Y. N. Skiba, On optimal solution of an inverse air pollution problem: Theory and numerical approach, Math. Comput. Model., 43(7), pp. 766–778, 2006.

[63] J. Peraire and P.-O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Numer. Anal., 30(4), pp. 1806–1824, 2008.

[64] P.-O. Persson and J. Peraire, Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations, J. Sci. Comput., 30(6), pp. 2709–2733, 2008.

[65] F. Prill, M. Lukácová-Medvid̆ová, and R. Hartmann, Smoothed aggregation multigrid for the discontinuous Galerkin method, J. Sci. Comput., 31(5), pp. 3503–3528, 2009.

[66] W. H. Reed and T. R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[67] S. Reitzinger, *Algebraic Multigrid Methods for Large Scale Finite Element Equations*, Schriften der Johannes-Kepler-Universität Linz, Rudolf Trauner, 2001.

[68] A. Reusken, Convergence analysis of a multigrid method for convection–diffusion equations, Numer. Math., 91(2), pp. 323–349, 2002.

[69] B. Rivière, *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*, Society for Industrial and Applied Mathematics, 2008.

[70] B. Rivière, M. F. Wheeler, and V. Girault, Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. part i, Computational Geosciences, 3(3-4), pp. 337–360, 1999.

[71] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003.

[72] J. Schöberl, R. Simon, and W. Zulehner, A robust multigrid method for elliptic optimal control problems, SIAM J. Numer. Anal., 49(4), pp. 1482–1503, 2011.

[73] W. A. Strauss, *Partial differential equations: An introduction*, John Wiley, 2008, ISBN 9780470054567.

[74] K. Stüben, An introduction to algebraic multigrid, in *Multigrid*, pp. 413–533, Academic Press, London, 2001.

[75] T. Sun, Discontinuous Galerkin finite element method with interior penalties for convection diffusion optimal control problem, Int. J. Numer. Anal. Model, 7(1), pp. 87–107, 2010.

[76] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, 2001.

[77] M. Van Raalte and P. Hemker, Two-level multigrid analysis for the convection–diffusion equation discretized by a discontinuous Galerkin method, Numer. Linear Algebra Appl., 12(5-6), pp. 563–584, 2005.

[78] R. S. Varga, *Matrix iterative analysis*, volume 27, Springer Science & Business Media, 2009.

[79] M. F. Wheeler, An elliptic collocation-finite element method with interior penalties, SIAM J. Numer. Anal., 15, pp. 152–161, 1978.

[80] C. Xiong and Y. Li, Error analysis for optimal control problem governed by convection diffusion equations: DG method, J. Comput. Appl. Math., 235(10), pp. 3163–3177, 2011.

[81] N. Yan and Z. Zhou, A priori and a posteriori error analysis of edge stabilization Galerkin method for the optimal control problem governed by convection-dominated diffusion equation, J. Comput. Appl. Math., 223(1), pp. 198–217, 2009.

[82] H. Yücel and P. Benner, Adaptive discontinuous Galerkin methods for state constrained optimal control problems governed by convection diffusion equations, Comput. Optim. Appl., pp. 1–31, 2014.

[83] H. Yücel, M. Heinkenschloss, and B. Karasözen, Distributed optimal control of diffusion-convection-reaction equations using discontinuous Galerkin methods, in *Numerical Mathematics and Advanced Applications 2011*, pp. 389–397, Springer, 2013.

[84] H. Yücel and B. Karasözen, Adaptive symmetric interior penalty Galerkin (SIPG) method for optimal control of convection diffusion equations with control constraints, Optimization, 63(1), pp. 145–166, 2014.

[85] Z. Zhou and N. Yan, The local discontinuous Galerkin method for optimal control problem governed by convection diffusion equations, Int. J. Numer. Anal. Model, 7(4), pp. 681–699, 2010.