A Multivariate Spline Method for Numerical Solution of Partial Differential Equations

by

Jinsil Lee

(Under the Direction of Ming-Jun Lai

Abstract

This thesis presents a new collocation method using multivariate splines over triangulation or tetrahedralization for solving partial differential equations. The method is applied to the Poisson equation and extended to the second-order elliptic PDE in non-divergence form, with numerical experiments demonstrating its accuracy and efficiency in both 2D and 3D settings compared to existing spline methods. The thesis also explores solving the Dirichlet problem of the 2D and 3D elliptic Monge-Ampére equation and addresses optimal control design for suppressing singularity formation in chemotaxis governed by the parabolic-elliptic Patlak-Keller-Segel system via flow advection, with the spline collocation method employed to solve the optimality conditions and numerical experiments demonstrating effectiveness.

Index words: Collocation method, Multivariate splines, elliptic PDE in nondivergence form, Monge-Ampére equation, Keller-Segel equation, Optimal control problem

A Multivariate Spline Method for Numerical Solution of Partial Differential Equations

by

Jinsil Lee

B.S., Pusan National University, Republic of Korea, Feb 2014

M.S., Pusan National University, Republic of Korea, Feb 2016

A Dissertation Submitted to the Graduate Faculty of the

University of Georgia in Partial Fulfillment of the Requirements for the Degree.

DOCTOR of PHILOSOPHY

Athens, Georgia

2023

A Multivariate Spline Method for Numerical Solution of Partial Differential Equations

by

Jinsil Lee

Major Professor: Ming-Jun Lai

Co-Advisor: Weiwei Hu

Committee: Lin Mu

Jingzhi Tie

Electronic Version Approved:

Ron Walcott

Dean of the Graduate School

The University of Georgia

August 2023

Dedication

I am deeply grateful for the unwavering support, guidance, and encouragement that Ming-Jun Lai, my advisor, has provided throughout my doctoral studies. His insightful advice and continuous motivation have played an indispensable role in the successful completion of this thesis.

Furthermore, I would like to express my sincere appreciation to Weiwei Hu, my co-advisor, and Yong-Hoon Lee, my master's thesis advisor, for their invaluable mentorship. Their encouragement to participate in conferences and their assistance in expanding my research interests have been crucial in shaping both this dissertation and my personal academic development.

I would also like to extend my gratitude to my colleagues and friends for their unwavering support and camaraderie during my academic journey.

Lastly, I am deeply thankful to my family members, Byeongyoung Lee, Pilsoo Choi, and Yeaseul Lee, for their constant support and encouragement. Their unwavering belief in me has been an enduring source of strength and motivation.

Acknowledgments

I would like to express my deepest gratitude to Ming-Jun Lai, my advisor, for his unwavering support, guidance, and encouragement throughout my doctoral studies. His insightful advice and continuous motivation have been indispensable in the successful completion of this thesis. I am truly grateful for his mentorship and the impact he has had on my academic growth.

I would also like to sincerely thank Weiwei Hu, my co-advisor, and Yong-Hoon Lee, my master's thesis advisor, for their invaluable mentorship. Their encouragement to participate in conferences and their assistance in expanding my research interests have been pivotal in shaping not only this dissertation but also my personal and intellectual development. Their expertise and dedication have been invaluable assets to my research journey.

I would also like to extend my gratitude to my colleagues and friends for their unwavering support and camaraderie. Their presence and encouragement have provided a sense of community and inspiration during the challenges of my academic pursuit.

Lastly, I want to express my heartfelt appreciation to my family members, Byeongyoung Lee, Pilsoo Choi, and Yeaseul Lee, for their constant support and unwavering belief in me. Their encouragement and faith in my abilities have been a constant source of strength and motivation.

Furthermore, I would like to acknowledge and thank Dr. Awanou for inviting me to present my research and for providing valuable advice. His support and guidance have been instrumental in elevating the quality of my work.

Additionally, I am grateful to Dr. Mu and Dr. Tie, who served as members of my thesis committee. Their expertise and constructive feedback have significantly contributed to the refinement and success of this research endeavor. I deeply appreciate their time, knowledge, and dedication in evaluating my work and providing invaluable insights.

The mentorship and contributions of all these individuals have profoundly enriched my academic journey, and I am truly honored to have had the privilege to collaborate with them.

CONTENTS

Acknowledgments **v**

List of Figures **ix**

List of Tables **xii**

1 Introduction **1**

1.1 PDEs 3

1.2 Review of Numerical Methods for PDEs 6

1.3 Domains with positive reach 13

1.4 Multivariate Splines 15

2 A Spline-Based Collocation Method for second order elliptic equations **22**

2.1 A Spline Based Collocation Method for the Poisson Equation 28

2.2 General Second Order Elliptic Equations 37

2.3 Implementation of the Spline-based Collocation Method 45

2.4 Numerical results for the Poisson Equation 49

2.5 Numerical Results for General Second Order Elliptic PDE 55

2.6 The Rate of Convergence of the LL method 60

3 A Splined Based Collocation Method for Monge Amphére equations 65

3.1 Our Proposed Algorithms and Their Convergence Analysis 73

3.2 Numerical Results for 3D Monge-Ampère Equations 90

4 Spline Collocation Method for Chemotaxis with Flow Advection 104

4.1 Well-posedness of the PKS system and existence of an optimal control 109

4.2 First-order optimality conditions 118

4.3 Numerical Implementation 125

5 Conclusion 142

Appendices 144

A Convergence of Algorithm 1 144

Bibliography 150

List of Figures

1.1 Domains with positive reach 13

2.1 Several 2D domains used for Numerical Experiments 46

2.2 Several 3D domains used for Numerical Experiments 46

2.3 The accuracies of the solutions $|e_s|_{l_2}, |e_s|_{h_1}$ and the smoothness $|H_0 c|_{l_\infty}, |Hc|_{l_\infty}$ based on testing functions u^{s5}(left) and u^{s8}(right) with $\alpha = \gamma = 1$ for various β . . . 50

2.4 The accuracies of the solutions $|e_s|_{l_2}, |e_s|_{h_1}$ and the smoothness $|H_0 c|_{l_\infty}, |Hc|_{l_\infty}$ based on testing functions u^{s5}(left) and u^{s8}(right) with $\beta = \gamma = 1$ for various α . . . 50

2.5 The RMSE in L^2 and H^1 norm of $u - u_s$ for testing functions u^{s2}(left) and u^{s4}(right) versus the size h of triangulation with $D = 8, r = 2$ where $e_s := u - u_s$ 61

2.6 The RMSE in L^2 and H^1 norm of $u - u_s$ for testing functions u^{s2}(left) and u^{s4}(right) versus the DOFs with $D = 8, r = 2$ where $e_s := u - u_s$ 62

2.7 The RMSE in L^2 and H^1 norm of $u - u_s$ for testing functions u^{s2}(left) and u^{s4}(right) versus the degrees with $r = 2$ where $e_s := u - u_s$ 62

2.8 The RMSE in L^2 and H^1 norm of $u - u_s$ with $D = 9, r = 1$ for testing functions u^{3ds3}(left) and u^{3ds5}(right) versus the mesh size h 63

2.9 The RMSE in L^2 and H^1 norm of $u - u_s$ with $D = 9, r = 1$ for testing functions u^{3ds3}(left) and u^{3ds5}(right) versus the DOFs 64

3.1 $\log(\|u - u_s^{(k)}\|_\infty)$($y$-axis) for u^{3d3}(left) and u^{3d5}(right) for each iteration(x-axis) with different $a = 6, 9, 27$ 75

3.2 Errors $\log(|e_s|_{l_2}), \log(|e_s|_{h_1})$ for u^{3ds3}(Top) and u^{3ds8}(Bottom) 79

3.3 ρ_k and $\|\Delta u_k\|_\infty$ for 80 iterations for smooth solutions s_1, s_2, s_3 and non-smooth solutions ns_1, ns_2 87

3.4 An enlarged graphs in Figure 3.3 87

3.5 Convergence rates of l_2, h_1 errors for solutions u^{3ds3}(Left) and u^{3ds5}(Right) with respect to $|\triangle|$ based on the LL method 92

3.6 Several 3D domains (Top: Cube, Letter L, Letter C, Bottom: Letter S, Subset of the unit ball) 95

3.7 Several 3D domains (Cube, Letter L, Torus) 98

3.8 Function f, g and error $\|g(\nabla u_s)\det(D^2 u_s) - f\|_\infty$ in Example 12 102

3.9 Graph of f, g, and $g(\nabla u_s)\det(D^2 u_s)$ in Example 13 103

3.10 Plot of the vector field $(u_x - x, u_y - y)$ overlaid with a color map representing the function f (the color map shows the value of f at each point) 103

4.2 Initial condition of θ (Left), numerical solution c_0 from Step 1 (Middle) at $t = 0$, RMSE for solution θ, c in Example 15 131

4.5 Density θ with $(x_0, y_0) = (0.5, 0.5)$ for various initial mass 137

4.6 Uncontrolled evolution of density θ with $(x_0, y_0) = (0.5, 0.5)$ at various time steps . 138

4.7 $\mathbf{v}_1$(Left) and $\mathbf{v}_2$(Right) 139

4.8 Controlled evolution of density θ with $(x_0, y_0) = (0.5, 0.5), N = 2$ at various time steps $t = 0, 0.05, 0.1, 0.4$ 140

4.9 Control input and cost functional for $(x_0, y_0) = (0.5, 0.5), N = 4$, and $t \in [0, 0.4]$. 141

4.10 Density θ for $(x_0, y_0) = (0.7, 0.7)$ with different $\bar{u}$ at $t = 0.0188$ and $t = 0.5$ 141

List of Tables

2.1 Times in seconds for generating necessary matrices for each 2D domain in Figure 1.1 and 2.1. 47

2.2 Times in seconds for generating necessary matrices for each 3D domain in Figure 2.2. 48

2.3 Times in seconds for generating necessary matrices for each 3D domain in Figure 2.2 with degree $D = 5, 9$ 48

2.4 Times in seconds for finding solutions of 3D Poisson equation(P), general second-order elliptic equation with smooth PDE coefficients (SG) or with non-smooth PDE coefficients (NSG1, NSG2) for each domain in Figure 2.2. 48

2.5 The RMS of errors $u - u_s$ and $\nabla u - \nabla u_s$ for Poisson equations for four domains showed in Figure 1.1 when $r = 2$ and $D = 8$. 52

2.6 RMSE of spline solutions for the Poisson equation over the four domains in Figure 1.1 when $r = 2$ and $D = 8$ for both the AWL method and the LL method. 53

2.7 The number of vertices, triangles and the averaged time for solving the 2D Poisson equation for each domain in Figure 1.1. 53

2.8 RMS of error vectors $u - u_s$ and $\nabla u - \nabla u_s$ for the 3D Poisson equation over the four domains in Figure 2.2 based on trivariate spline functions of smoothness $r = 1$ and degree $D = 9$ 54

2.9 The RMSE of spline solutions for the 3D Poisson equation over the two domains in Figure 2.2 based on trivariate spline functions of smoothness $r = 1$ and degree $D = 9$ for the AWL method and LL method. 55

2.10 Comparison of the RMS of vectors $u - u_s, \nabla u - \nabla u_s$ for general elliptic equations with smooth coefficients in Example 1 and CPU time for different mesh sizes and degrees 57

2.11 RMSE of spline solutions for general second order elliptic equations with smooth coefficients over the four domains in Figure 1.1 and 2.1 when $r = 2$ and $D = 8$. . . 57

2.12 RMSE $u - u_s$ and $\nabla u - \nabla u_s$ for the general elliptic equation with the non-smooth coefficients in Example 14 over the four domains in Figure 1.1 and 2.1 when $r = 2$ and $D = 8$. 58

2.13 The RMS of vectors $u - u_s, \nabla u - \nabla u_s$ for general elliptic equations with non-smooth coefficients in Example 15 over the four domains when $r = 2$ and $D = 8$. 59

2.14 The RMSE of spline solutions for general elliptic equations in Example 1, PDE with non-smooth coefficients in Example 14 and in Example 15 over the Circle with 3 holes when $r = 2$ and $D = 8$ for the LW method and the LL method, respectively. 60

2.15 The number of vertices, triangles and the averaged time in seconds for solving 2D general second order equations over the four domains in Figure 1.1 by the LW and LL methods. 60

3.1 Comparison of computational efficiency and accuracy of different numerical methods with the number of vertices $N_v = 81$ and the number of triangles $N_T = 128$, the exact solution $u = e^{\frac{x^2+y^2}{2}}$ for solving the 2D Monge-Ampére equation 74

3.2 Errors of numerical solutions u^{3ds1} for the Monge Ampère equation over $[0,1]^3$ with $D = 9$, $r = 1$ over the same tetrahedralization for various initial values p by two algorithms 79

3.3 Errors of numerical solutions u^{3ds1}, u^{3ds2} for Monge Ampère equation over $[0,1]^3$ for LL methods with $D = 5$, $r = 1$ and LR method in [CGG18] 91

3.4 Errors of numerical solutions u^{3ds3} and CPU time(s) for Monge Ampère equation over $[0,1]^3$ for LL methods with $D = 5$, $r = 1$ and LR method in [CGG18] 92

3.5 Errors of numerical solutions u^{3ds4} for Monge Ampère equation over $[0,1]^3$ for LL methods with $D = 3,4,5,6$, $r = 1$, $h = 1$ and SE method in [Awa15] 93

3.6 Errors of numerical solutions u^{3ds4} for Monge Ampère equation over $[0,1]^3$ for LL methods with $D = 3,4,5,6$, $r = 1$, $h = 1/2$ and SE method in [Awa15] 93

3.7 Errors of numerical approximation of the solution u^{3ds5} for Monge Ampère equation over $[0,1]^3$ by the and LR method and by the LL method with $D = 5$, $r = 1$ 94

3.8 Errors of numerical approximation of the solution u^{3ds6} for Monge Ampère equation over $[0,1]^3$ by the and LR method and by the LL method with $D = 5$, $r = 1$ 95

3.9 CPU time results and numbers of vertices and tetrahedrons over domains in Figure 3.6 when $D = 9, r = 1$ 96

3.10 Errors of numerical solutions $u^{3ds3} - u^{3ds5}$ for Monge Ampère equations over several domains in Figure 3.6 for LL methods with $D = 9$, $r = 1$ 96

3.11 Errors of numerical approximation of solution u^{3ds7} for Monge Ampère equation over the unit ball for the LR method and the LL method with $D = 5, r = 1$ 97

3.12 CPU time and errors of our spline solution to u^{3ds7} for Monge Ampère equation over the domain in Figure 3.6 with $D = 5$, $r = 1$, the number of vertices=585, the number of tetrahedrons=2304 97

3.13 The CPU time, DOFs, errors $|e_s|_{l_2}, |e_s|_{h_1}$ using LL method with $D = 6, r = 1$ and $|e_s|_{l_2}$ using the Cascadic method in [Ben+20] over the cube $[0, 1]^3$ 98

3.14 The CPU time, DOFs, errors $|e_s|_{l_2}, |e_s|_{h_1}$ using LL method with $D = 6, r = 1$ and $|e_s|_{l_2}$ using the Cascadic method in [Ben+20] over L-shaped domain 99

3.15 The CPU time, DOFs, errors $|e_s|_{l_2}, |e_s|_{h_1}$ using LL method with $D = 3, r = 1$ and $|e_s|_{l_2}$ using the Cascadic method in [Ben+20] over Torus 99

3.16 Numerical results for example 12 102

Chapter 1

Introduction

In recent years, there has been a growing interest in employing numerical techniques to approximate solutions for partial differential equations (PDEs) using triangulation or tetrahedral structures. Numerous numerical methods, such as the finite element method, finite difference method, meshless method, discontinuous/continuous Galerkin methods, and neural network methods, have emerged as promising alternatives to tackle this complex problem. This trend underscores the increasing demand for faster and more accurate numerical solutions to PDEs.

This dissertation focuses on the multivariate spline collocation method as a novel approach for solving PDEs with their boundary conditions, which frequently arise in various fields like engineering, mathematics, physics, and biology. Spline functions offer several advantages in the context of PDE numerical approximations, including degree flexibility, customizable smoothness, and the inherent partition of unity property found in Bernstein-Bézier polynomials. For theoretical properties and numerical implementation of bivariate/trivariate spline functions, refer to [LS07a], [ALW06], [Sch15], [LL22], and [LL23]. Additionally, multiple dissertations explain the implemen-

tation and usage of multivariate splines for the numerical solution of Helmholtz equations, Maxwell equations, and 3D surface reconstruction; see [Awa03], [Mer19], and [Xu19]. These dissertation works greatly improved the performance of computation of multivariate splines for applications such as numerical solutions of PDEs in comparison to traditional techniques, e.g. finite element methods, discontinuous Galerkin methods, finite difference methods, and others.

One significant advantage of the spline-based collocation method is its ability to eliminate weak PDE solution formulations, thereby eliminating the need for numerical quadrature in computations. This reduction in complexity potentially leads to enhanced computational efficiency. Additionally, the spline-based collocation technique demonstrates enhanced adaptability in handling PDE coefficient-induced discontinuities. Such flexibility enables convenient adjustment of collocation points in the vicinity of both sides of disjoint curves or surfaces, thus ensuring a dependable numerical approximation that faithfully captures the underlying natural phenomena.

The multivariate spline-based collocation method also offers an effective approach to enhancing approximation accuracy by increasing the degree of polynomials or augmenting the number of collocation points. This technique is more cost-effective than attempting to find solutions through uniform refinement of the underlying triangulation or tetrahedralization, especially when limited by computer memory constraints. By utilizing a higher density of collocation points, the spline-based method achieves superior accuracy while maintaining reasonable computational costs.

Additionally, our spline collocation method incorporates tuning parameters that allow for control over the accuracy and the smoothness of spline solutions. By adjusting these parameters, users can achieve an optimal balance between the approximation's fidelity and the computational resources required to obtain it.

In Chapter 1 of this thesis, we begin by presenting relevant definitions and notations pertaining to Partial Differential Equations (PDEs), as well as different boundary conditions. Next, we provide a comprehensive review of various numerical methods for solving PDEs, discussing their respective advantages and disadvantages. We also provide a brief summary of the fundamental concepts of multivariate splines, which will be utilized in later sections.

The rest of the work is organized as follows. Chapter 2 presents a brief introduction of spline collocation method and how to solve the 2D and 3D poisson equation and 2nd order ellipic PDEs. Chapter 3 describes how to solve the Monge-Ampére equation using iterative method and displays numerical results. Chapter 4 describes how to apply the spline collocation method to the Keller-Segel method.

1.1 PDEs

Partial differential equations describe physical phenomena such as sound, heat, diffusion, electrostatics, electrodynamics, fluid dynamics, elasticity, general relativity, and quantum mechanics. They also arise from many purely mathematical considerations, such as differential geometry and the calculus of variations. For a real multivariate function u and a domain Ω, we call a k^{th} order partial differential equation of u as

$$\mathcal{L}(D^k u(x), D^{k-1}u(x), \cdots, Du(x), u(x), x) = 0,\ x \in \Omega \tag{1.1}$$

where

$$\mathcal{L} : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \cdots \times \mathbb{R} \times \Omega \to \mathbb{R}$$

is given. A PDE is called linear if $\mathcal{L}$ is a linear function of $u(x)$ and its derivatives. For a second-order linear PDE, the general form of PDE in two dimensions is given as follows

$$a^{11}\frac{\partial^2 u}{\partial x_1 \partial x_1} + a^{12}\frac{\partial^2 u}{\partial x_1 \partial x_2} + a^{22}\frac{\partial^2 u}{\partial x_2 \partial x_2} + b^1\frac{\partial u}{\partial x_1} + b^2\frac{\partial u}{\partial x_2} + cu = f, \quad (x_1, x_2) \in \Omega \subset \mathbb{R}^2 \tag{1.2}$$

The PDE properties depend on the sign of the discriminant $D(x) = (a^{12}(x))^2 - 4a^{11}(x)a^{22}(x)$.

- if $D(x) < 0$, the PDE is called elliptic,
- if $D(x) = 0$, the PDE is called parabolic,
- if $D(x) > 0$, the PDE is called hyperbolic.

A given PDE may be of one type at a specific point $\hat{x}$ and of another type at some other point x^*. From a physical standpoint, elliptic, parabolic, and hyperbolic PDEs represent steady-state or equilibrium processes, time-dependent diffusion processes, and wave propagation, respectively. Elliptic equations describe systems in their minimal energy state, while parabolic equations describe evolutionary phenomena that ultimately lead to a steady state described by an elliptic equation. Finally, hyperbolic equations often model the transport of some physical quantity, such as mass transfer in fluids.

1.1.1 Sobolev Spaces

Let us introduce Sobolev spaces which play an important role in PDE theory. Let p be a real number, $p \geq 1$. We define the set of all real-valued integrable functions on a domain $\Omega \subseteq \mathbb{R}^n$ as follows:

$$L^p(\Omega) = \{u : \Omega \to \mathbb{R}^n | u \text{ is measurable and } \int_\Omega |u|^p < \infty\}.$$

This $L^p(\Omega)$ is a normed Banach space equipped with the norm

$$\|u\|_{L^p(\Omega)} = (\int_\Omega |u|^p)^{\frac{1}{p}}.$$

That is, $\|\cdot\|_{L^p(\Omega)}$ satisfies the norm properties:

- $\|u + v\|_{L^p(\Omega)} \leq \|u\|_{L^p(\Omega)} + \|v\|_{L^p(\Omega)}$ for all $u, v \in L^p(\Omega)$
- $\|au\|_{L^p(\Omega)} = |a|\|u\|_{L^p(\Omega)}$ for all $u \in L^p(\Omega)$ and all scalars a.
- for all $u \in L^p(\Omega)$, if $\|u\|_{L^p(\Omega)} = 0$, then $u = 0$.

and for every Cauchy sequence $u_n \in L^p(\Omega), n \in N$ converges to an element $u \in L^p(\Omega)$, i.e. $\lim_{n\to\infty} \|u_n - u\| = 0$. Now we introduce a Sobolev space of order k, W_p^k, defined by

$$W^{k,p}(\Omega) = \{u \in L^p(\Omega) | D^\alpha u \in L^p(\Omega), |\alpha| \leq k\}$$

equipped with the Sobolev norm

$$\|u\|_{W^{k,p}(\Omega)} := (\sum_{|\alpha|\leq k} \|D^\alpha u\|^p_{L^p(\Omega)})^{\frac{1}{p}}$$

where D^α is a weak derivative of order $|\alpha| \leq k$ which coincides with the corresponding partial derivative in the classical pointwise sense,

$$\frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}.$$

In particular, when $p = 2, k = m$, we denote $W^{m,2}(\Omega)$ by $H^m(\Omega)$ and this space is Hilbert space. Last, we define the closure of $C_0^\infty(\Omega)$ by $H_0^1(\Omega)$. If $\partial\Omega$, then $u = 0$ on $\partial\Omega$ for any $u \in H_0^1(\Omega)$.

1.2 Review of Numerical Methods for PDEs

We first introduce the concept of weak solutions of PDEs. For convenience, we consider the Poisson equation

$$-\Delta u = f, \quad \text{in } \Omega, \quad u = g, \text{on } \partial\Omega$$

we called a function $u \in C^2(\Omega) \cap C(\bar{\Omega})$ satisfying the above Poisson equation as a classical solution of this equation. However, it is hard to find such a C^2 smooth solution u without using the spline collecation method discussed in this dissertation. To deal with this limitation, we consider a weak solution for PDEs. For any $v \in C_0^1(\Omega)$ and a classical solution of the Poisson equation, we have

$$-\int \Delta u v = \int f v.$$

By the Green's theorem and the fact that $v = 0$ on $\partial\Omega$, we obtain

$$\int \nabla u \nabla v = \int fv \ \ \forall v \in C_0^1(\Omega).$$

In this integration equation, we can assume that $u \in H_0^1(\Omega)$, not $C^2(\Omega)$. We call a function $u \in H_0^1(\Omega)$ as a weak solution of satisfying

$$-\int \Delta u v = \int fv.$$

All partial derivatives should be understood as weak derivatives. Most researchers try to find a weak solution of PDEs using various numerical methods.

1.2.1 Finite Element Method

The finite element method (FEM) is widely used to solve two or three-dimensional PDEs arising in engineering and mathematical modeling. The basic idea of FEM is to approximate a function u satisfying a given PDE using linear combinations of basis functions. For simplicity, let us consider the Poisson equation

$$-\Delta u = f$$

over the rectangular domain $\Omega = [0, 1]^2$ with a given data f where

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

It starts by taking the inputted bounded domain with a polygonal boundary $\partial\Omega$ and performing calculations to make the domain into finitely many elements such as triangles, and rectangles. Next, we define a finite elements subspace V_h consisting of continuous piecewise linear functions. With each interior node, we associate a basis function ϕ which is equal to 1 at that node and equal to 0 at all the other nodes. The finite element approximation of the problem is to find $u_h = \sum_{i=1}^{N} c_i\phi_i \in V_h$ such that

$$\int_\Omega (\frac{\partial u_h}{\partial x}\frac{\partial v_h}{\partial x} + \frac{\partial u_h}{\partial y}\frac{\partial v_h}{\partial y})dxdy = \int_\Omega f v_h dxdy \ \ \forall v_h \in V_h$$

where N is the number of nodes. That is, the goal of this method is to find $c = (c_1, \cdots, c_N)$ such that

$$\sum_{i=1}^{N} c_i \int_\Omega (\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} + \frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y})dxdy = \int_\Omega f\phi_j dxdy \ \ \forall \phi_j \in V_h.$$

It can be rewritten as a system of linear equations

$$Ac = F$$

where $A = (\int_\Omega (\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} + \frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y})dxdy)_{i,j}$ and $F = (\int_\Omega f\phi_j dxdy)_j$. We can apply this method to different kinds of PDE such as general elliptic PDEs, and Biharmonic equations in a similar way.

1.2.2 Finite Difference Method

Another useful numerical method for PDE equations is the finite difference method(FDM). For example, we consider the Poisson equation

$$-\Delta u = f$$

over the rectangular domain $\Omega = [0,1]^2$ with Dirichlet boundary conditions. Given two integers $m, n \geq 2$, we construct a rectangular grid $\mathcal{T}_h$ by the tensor product of two uniform grids of $[0,1]$: $\{(x_i, y_i) = ((i-1)h_x, (j-1)h_y)|, 1 \leq i \leq m, 1 \leq j \leq n$ with $h_x = \frac{1}{m-1}, h_y = \frac{1}{n-1}$. Let $h = max\{h_x, h_y\}$ denote the size of $\mathcal{T}_h$. Denote by $\Omega_h = \{(x_i, y_i) \in \Omega\}$ and boundary $\Gamma_h = \{(x_i, y_i) \in \partial\Omega\}$. We consider the discrete function space given by $V_h = \{u_h(x_i, y_j), 1 \leq i \leq m, 1 \leq j \leq n\}$. For convenicence, we denote $u_{i,j} = u(x_i, y_j)$. For a continuous function $u \in C(\Omega)$, the interpolation operator $I_h : C(\Omega) \to V_h$ maps u to a discrete function u_I. By the definition, $(u_I)_{i,j} = u(x_i, y_j)$. Note that the value of a discrete function is only defined at grid points. Values inside each cell can be obtained by the convex combination of values at grid points. Similar definitions can be applied to the one-dimensional case. We choose a mesh size h and $u \in V_h(0,1)$. There are several different formulas at a node x_j for a discrete function $u \in V_h$ as follows:

- Backward difference : $(D^- u)_j = \frac{u_j - u_{j-1}}{h}$
- Forward difference : $(D^+ u)_j = \frac{u_{j+1} - u_j}{h}$
- Central difference : $(D^c u)_j = \frac{u_{j+1} - u_{j-1}}{2h}$
- Second central difference : $(D^{2c} u)_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2}$

By using the second central difference to approximate the Laplace operator at an interior node (x_i, y_j) :

$$\begin{aligned}(\Delta_h u)_{i,j} &= (D_{xx}^{2c} u)_{i,j} + (D_{yy}^{2c} u)_{i,j} \\ &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}\end{aligned}$$

When $h_x = h_y$, it can be simplified to

$$-(\Delta_h u)_{i,j} = \frac{-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1}}{h^2}$$

and can be denoted by the following stencil symbol

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Hence, the finite difference method for solving the Poisson equation is simply

$$\frac{-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1}}{h^2} = f_{i,j}, 1 \le i \le m, 1 \le j \le n, \tag{1.3}$$

with appropriate processing of a boundary condition where $f_{i,j} = f(x_i, y_j)$. We use (1.3) for all grid points including boundary points but simply drop terms involving grid points outside of the domain. Let us give an ordering of $N = m \times n$ grids and use a single index $k = 1$ to N for $u_k = u_{i(k),j(k)}$ which is called a linear indexing. With any choice of linear indexing, (1.3) can be written as a linear

algebraic equation:

$$Ku = f,$$

where $K \in \mathbb{R}^{N\times N}, u, f \in \mathbb{R}^n$.

1.2.3 Meshless Method using Radial Basis Functions

The Radial Basis Function (RBF) method has been widely researched in various fields of science and engineering as a means of interpolation and approximation. RBFs have been used in different methods such as the collocation method, meshless local Petrov-Galerkin method, dual reciprocity method, method of fundamental solution, and others. In 1991, Kansa was the first to apply RBFs to solve PDEs in fluid dynamics problems through a direct collocation procedure. Kansa's method can be applied to boundary value PDE problems and results in an asymmetric linear system of equations. The primary concept of Kansa's method is to approximate the solution using a linear combination of RBFs as represented in the equation:

$$u_{rbf}(x) = \sum_{k=1}^{N} c_k \phi(\|x - x_k\|).$$

Here, c_j are undetermined coefficients, and the Euclidean norm is commonly used for the norm $|\cdot|$. A simple boundary value problem can be represented as:

$$\mathcal{L}u(x) = f(x), x \in \Omega,\ Bu(x) = g(x), x \in \partial\Omega,$$

where $\mathcal{L}$ is a differential operator, B is a boundary differential operator, and f, g are known functions. For this problem, N_i distinct interior collocation points in Ω are represented by $(x_j, f(x_j))_{j=1}^{N_i}$, and $(x_j, g(x_j))_{j=N_i+1}^{N}$ are boundary points. Using the collocation technique, the boundary value problem can be reduced to a discrete problem by imposing a finite number of conditions as given by:

$$\mathcal{L}u(x_j) = f(x_j), j = 1, 2, \cdots, N_i,$$

$$Bu(x_j) = g(x_j), j = N_i + 1, \cdots, N.$$

This results in a square linear system where the c_j can be obtained using any appropriate linear system solver, given by:

$$\sum_{k=1}^{N} c_k \mathcal{L}\phi(|x_j - x_k|) = f(x_j), j = 1, 2, \cdots, N_i,$$

$$\sum_{k=1}^{N} c_k B\phi(|x_j - x_k|) = g(x_j), j = N_i + 1, \cdots, N.$$

Some commonly used RBFs are $\phi(r) = r, r^3, e^{-r^2}, r^{2n-1}, r^{2n} \ln r$, etc. These functions are considered to satisfy the invertibility of the linear system for any finite distinct points.

However, one disadvantage of Kansa's method is that it is a global method that generates a dense matrix, which can be impossible for large-scale problems. To address this issue, several local meshless methods have been developed, such as the Compactly Supported RBFs (CS-RBF), the Local Multiquadrics Approximation (LMQ), and the Local RBF Collocation Method (LRBFCM).

These methods construct local domains for each collocation point, wherein each collocation point considers its nearest neighbors. This approach results in smaller linear systems in which the dimensions of the matrices are equal to the number of collocation points that fall within each local domain.

1.3 Domains with positive reach

Let us introduce a concept on domains of interest explained in [GL20].

Definition 1. *Let $K \subseteq \mathbb{R}^d$ be a non-empty set. Let r_K be the supremum of the number r such that every point in*

$$P = \{x \in \mathbb{R}^d : dist(x, K) < r\}$$

has a unique projection in K. The set K is said to have a positive reach if $r_K > 0$.

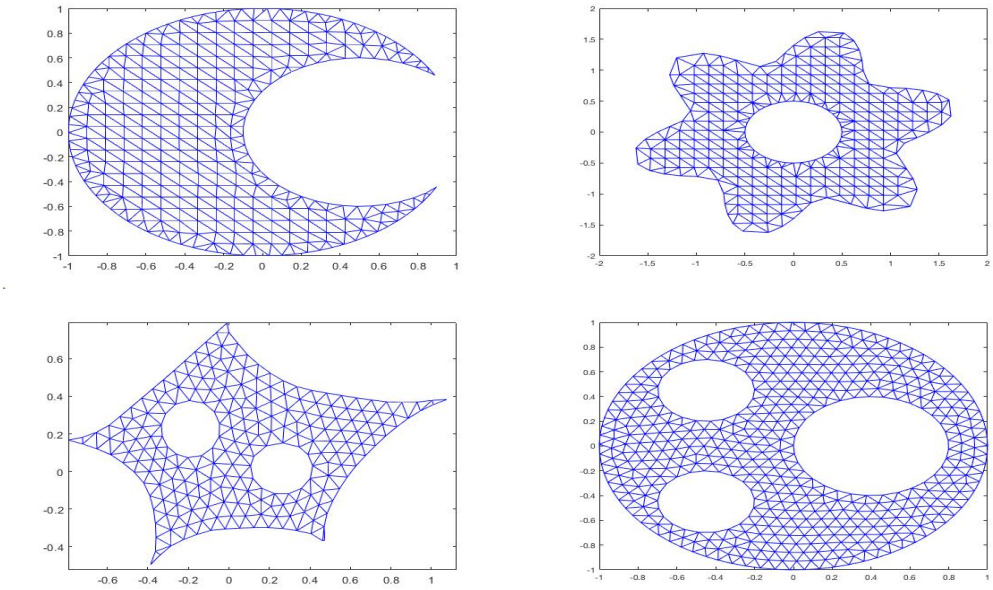


Figure 1.1: Domains with positive reach

A domain with a C^2 boundary has a positive reach. As Figure 1.1 illustrates, the domains with positive reach are much more general than convex domains. See Figure 2.2 for domains with positive reach in the 3D setting. Let $B(0,\epsilon)$ be the closed ball centering at 0 with radius $\epsilon > 0$, and let K^c stand for the complement of the set $K \in \mathbb{R}^d$. For any $\epsilon > 0$, the set

$$E_\epsilon(K) := (K^c + B(0,\epsilon))^c \subseteq K$$

is called an ϵ-erosion of K.

Definition 2. *A set $K \subseteq \mathbb{R}^d$ is said to have a uniformly positive reach r_0 if there exists some $\epsilon_0 > 0$ such that for all $\epsilon \in [0,\epsilon_0]$, $E_\epsilon(K)$ has a positive reach at least r_0.*

And we have the following property about these domains

Lemma 1. *If $\Omega \subset \mathbb{R}^d$ is of positive reach r_0, then for any $0 < \epsilon < r_0$, the boundary of $\Omega_\epsilon := \Omega + B(0,\epsilon)$ containing Ω is of $C^{1,1}$. Furthermore, Ω_ϵ has a positive reach $\geq r_0 - \epsilon$.*

In [GL20], Gao and Lai proved the following regularity theorem which will be used to prove Theorem 4 in the next chapter.

Theorem 1. *Let Ω be a bounded domain. Suppose the closure of Ω is of uniformly positive reach r_Ω. For any $f \in L^2(\Omega)$, let $u \in H_0^1(\Omega)$ be the unique weak solution of the Dirichlet problem:*

$$\begin{cases} -\Delta u &= f \ \ in\ \Omega \\ u &= 0 \ \ on\ \partial\Omega \end{cases}$$

Then $u \in H^2(\Omega)$ in the sense that

$$\sum_{i,j=1}^{n} \int_{\Omega} (\frac{\partial^2 u}{\partial x_i \partial x_j})^2 \leq C_0 \int_{\Omega} f^2 dx \tag{1.4}$$

for a positive constant C_0 depending only on r_Ω.

1.4 Multivariate Splines

We will begin by discussing bivariate spline functions. For a bounded domain $\Omega \subset \mathbb{R}^d$ with $d = 2$, let $\triangle := \{T_1, \cdots, T_n\}$ be a triangulation of Ω which is a collection of triangles and $\mathcal{V}$ be the set of vertices of $\triangle$. For a triangle $T \in \Omega, T_i = (v_1, v_2, v_3)$, we define the barycentric coordinates (b_1, b_2, b_3) of a point $(x, y) \in \Omega$. These coordinates are the solution to the following system of equations

$$b_1 + b_2 + b_3 = 1$$

$$b_1 v_{1,x} + b_2 v_{2,x} + b_3 v_{3,x} = x$$

$$b_1 v_{1,y} + b_2 v_{2,y} + b_3 v_{3,y} = y$$

and are nonnegative if $(x, y) \in T$. We use the barycentric coordinates to define the Bernstein polynomials of degree D:

$$B^T_{i,j,k}(x, y) := \frac{k!}{i!j!k!} b_1^i b_2^j b_3^k, \; i + j + k = D,$$

which form a basis for the space $\mathcal{P}_k$ of polynomials of degree D. Therefore, we can represent all $s \in \mathcal{P}_D$ in B-form:

$$s|_T = \sum_{i+j+k=D} c_{ijk} B^T_{ijk}, \tag{1.5}$$

where the B-coefficients $c_{i,j,k}$ are uniquely determined by s. Moreover, for given $T = (v_1, v_2, v_3) \in \triangle$,we define the associated set of domain points to be

$$\mathcal{D}_{D',T} := \{\eta_{ijk} = \frac{iv_1 + jv_2 + kv_3}{D'}\}_{i+j+k=D'}. \tag{1.6}$$

We need to agree on an ordering for the $N = \begin{pmatrix} d+2 \\ 2 \end{pmatrix}$ coefficients in (1.5). In the sequel, we assume that they are in lexicographical order,i.e., $c_{\hat{i}\hat{j}\hat{k}}$ comes before c_{ijk} provided $\hat{i} > i$ or ($\hat{i} = i$ and $\hat{j} > j$) or ($\hat{i} = i$, $\hat{j} = j$ and $\hat{k} > k$). For the domain points given by (1.8) and the Bernstein basis polynomials of degree d arranged in lexicographical order, the matrix $M = [B^T_{ijk}(\eta_{ijk})] \in \mathbb{R}^{N\times N}$ is nonsingular. In order to store a Bernstein polynomial, we only need to store its coefficient vector c.

We present an algorithm for efficiently evaluating a polynomial $s|_T$ at a given point η. The algorithm is based on a recurrence relation given by

$$B^{T,D}_{ijk} = b_1 B^{T,D-1}_{i-1jk} + b_2 B^{T,D-1}_{ij-1k} + b_3 B^{T,D-1}_{ijk-1}, \quad \text{all } i+j+k = D$$

and we denote $B^{T,D}_{D00} = b_1 B^{T,D-1}_{D-100}$. Then we have the following theorem

Theorem 2. *(de Casteljau Algorithm) Let s be a polynomial written in the B-form* (1.5) *with coefficients*

$$c_{ijk}^{(0)} = c_{ijk}, \quad i+j+k = D.$$

Suppose η has barycentric coordinates $b := (b_1, b_2, b_3)$, and for all $l = 1, \cdots, D$, let

$$c_{ijk}^{(l)} := b_1 c_{i+1jk}^{(l-1)} + b_2 c_{ij+1k}^{(l-1)} + b_3 c_{ijk+1}^{(l-1)}, \tag{1.7}$$

for $i+j+k = D-l$. Then

$$s(\eta) = \sum_{i+j+k=D-l} c_{ijk}^{(l)} B_{ijk}^{T,D-l}(\eta),$$

for all $0 \le l \le D$. In particular, $s(\eta) = c_{000}^{(D)}$.

We also present formulae for directional derivatives of a polynomial written in B-form. Let p is a point in $\mathbb{R}^2$ and $u := w - \tilde{w}$ is a vector where $w, \tilde{w}$ are points. Then, there are barycentric coordinates $(b_1, b_2, b_3), (\alpha_1, \alpha_2, \alpha_3)$ and $(\beta_1, \beta_2, \beta_3)$ corresponding to the points $p, w, \tilde{w}$, respectively. Then we define the directional derivative at p with respect to u by

$$\begin{aligned} D_u B_{ijk}^{T,D}(p) &:= \frac{d}{dt} B_{ijk}^{T,D}(p+tu)|_{t=0} \\ &= D[(\alpha_1 - \beta_1) B_{i-1jk}^{T,D-1} + (\alpha_2 - \beta_2) B_{ij-1k}^{T,D-1} + (\alpha_3 - \beta_3) B_{ijk-1}^{T,D-1}]. \end{aligned}$$

The following theorem can be easily derived from the B-form.

Theorem 3. *Let s be a polynomial of degree D written in the B-form* (1.5) *relative to a triangle T, and let u be a vector with directional coordinates $a := (a_1, a_2, a_3)$. Then the directional derivative*

at v of s in the direction u is given by

$$D_u s(v) = D \sum_{i+j+k=d-1} c^{(1)}_{ijk}(a) B^{D-1}_{ijk}(v),$$

where $c^{(1)}_{ijk}(a)$ *are the quantities obtained in the first step of the de Casteljau algorithm based on the triple* a.

In addition, we show that the above equations can be used to calculate the first or high-order derivatives of a Bernstein polynomial, as well as its integration and inner products. For more details, refer to [LS07a].

We define the spline space $S_D^{-1}(\triangle) := \{s|_T \in \mathcal{P}_D, T \in \triangle\}$, where T is a triangle in a triangulation $\triangle$ of Ω. We use this piecewise polynomial space to define the space $\mathcal{S}_D^r := C^r(\Omega) \cap S_D^{-1}(\triangle)$. This can be achieved through the smoothness conditions on the coefficients of $s \in S_D^{-1}(\triangle)$. Let H be the matrix which consists of the smoothness conditions across each interior edge. Let $\mathbf{s}$ be the coefficient vector of s. It is known that $H\mathbf{s} = 0$ if and only if $s \in C^r(\Omega)$ (cf. [LS07a]). It is known a multivariate spline space is a linear vector space that is spanned by a set of basis functions. However, it is difficult to construct locally supported basis functions in $C^r(\Omega)$ with $r \geq 1$ due to the complication of the smoothness conditions over $\triangle$. Typically, any small perturbation of a vertex in $\triangle$ may change the dimension of $S_D^r(\triangle)$. To overcome this difficulty of constructing locally supported basis spline functions, we will begin with a discontinuous spline space $s \in S_D^{-1}(\triangle)$ and then add the smoothness conditions $H\mathbf{c} = 0$ as constraints in addition to the constraint of boundary condition.

Computations involving splines written in B-form can be performed recursively using the de Casteljautes algorithm. In fact, these spline functions have numerically stable, closed-form formulas

for differentiation, integration, and inner products. If $D \geq 3r+2$, spline functions on quasi-uniform triangulations have optimal approximation power.

Lemma 2. *([Lai and Schumaker, 2007[LS07a]]) Let $k \geq 3r+2$ with $r \geq 1$. Suppose $\triangle$ is a quasi-uniform triangulation of Ω. Then for every $u \in W_q^{k+1}(\Omega)$, there exists a quasi-interpolatory spline $s_u \in \mathcal{S}_k^r(\triangle)$ such that*

$$||D_x^\alpha D_y^\beta (u - s_u)||_{q,\Omega} \leq C|\triangle|^{k+1-\alpha-\beta}|u|_{k+1,q,\Omega}$$

for a positive constant C dependent on u, r, k and the smallest angle of $\triangle$, and for all $0 \leq \alpha+\beta \leq k$ with

$$|u|_{k,q,\Omega} := (\sum_{a+b=k} ||D_x^a D_y^b u||_{L^q(\Omega)}^q)^{\frac{1}{q}}.$$

Similarly, for trivariate splines, let $\Omega \subset \mathbb{R}^3$ and $\triangle$ be a tetrahedralization of Ω. We define a trivariate spline just like bivariate splines by using Bernstein-Bźier polynomials defined on each tetrahedron $t \in \triangle$. Let us quickly summarize the essentials of trivariate splines in this section. Given a tetrahedron T, we write $|T|$ for the length of its longest edge, and ρ_T for the radius of the largest inscribed ball in T. For any polygonal domain $\Omega \subset \mathbb{R}^3$, let $\triangle := \{T_1, \cdots, T_n\}$ be a tetrahedralization of Ω which is a collection of tetrahedra and $\mathcal{V}$ be the set of vertices of $\triangle$. We called a tetrahedralization as a quasi-uniform tetrahedralization if all tetrahera T in $\triangle$ have comparable sizes in the sense that

$$\frac{|T|}{\rho_T} \leq C < \infty, \quad \text{all tetrahera } T \in \triangle,$$

where ρ_T is the inradius of T. Let $|\triangle|$ be the length of the longest edge in $\triangle$.

Next for a tetrahedron $T = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4) \in \triangle$, we define the barycentric coordinates (b_1, b_2, b_3, b_4) of a point $(x, y, z) \in \Omega$ as the solution to the following system of equations

$$b_1 + b_2 + b_3 + b_4 = 1$$

$$b_1 v_{1,x} + b_2 v_{2,x} + b_3 v_{3,x} + b_4 v_{4,x} = x$$

$$b_1 v_{1,y} + b_2 v_{2,y} + b_3 v_{3,y} + b_4 v_{4,y} = y$$

$$b_1 v_{1,z} + b_2 v_{2,z} + b_3 v_{3,z} + b_4 v_{4,z} = z,$$

where the vertices $\mathbf{v}_i = (v_{i,x}, v_{i,y}, v_{i,z})$ for $i = 1, 2, 3, 4$. $b_1, \cdots, b_4$ are nonnegative if $(x, y, z) \in T$. Next we use the barycentric coordinates to define the Bernstein polynomials of degree D:

$$B^T_{i,j,k,\ell}(x, y, z) := \frac{D!}{i!j!k!\ell!} b_1^i b_2^j b_3^k b_4^\ell, \; i + j + k + \ell = D,$$

which form a basis for the space $\mathcal{P}_D$ of polynomials of total degree D. Therefore, we can represent all $s \in \mathcal{P}_D$ in B-form:

$$s|_T = \sum_{i+j+k+\ell=D} c^T_{ijk\ell} B^T_{ijk\ell}, \forall T \in \triangle,$$

where the B-coefficients $c^T_{i,j,k,\ell}$ are uniquely determined by s. Let $\mathbf{c} = \{c^T_{ijk\ell}, i+j+k+\ell = D, T \in \triangle\}$ be the coefficient vector associated with the spline function s.

Moreover, for given $T = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4) \in \triangle$, we define the associated set of domain points to be

$$\mathcal{D}_{D,T} := \{\frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3 + \ell\mathbf{v}_4}{D}\}_{i+j+k=D}. \tag{1.8}$$

Let $\mathcal{D}_{D,\triangle} = \cup_{T\in\triangle}\mathcal{D}_{D,T}$ be the domain points of tetrahedral $\triangle$ and degree D. Letting

$$\mathbb{S}_D^r(\triangle) = \{s \in C^r(\Omega) : s|_t \in \mathbb{P}_D, t \in \triangle\} = C^r(\Omega) \cap S_D^{-1}(\triangle)$$

be the spline space of degree D and smoothness $r \geq 0$, each $s \in S_D^r(\triangle)$ can be rewritten as

$$s(x)|_t = \sum_{i+j+k+\ell=D} c_{ijk\ell}^t \mathcal{B}_{ijk\ell}^t(x), \quad \forall t \in \triangle,$$

where $\mathcal{B}_{ijk\ell}^t$ are Bernstein-Bźier polynomials (cf. [ALW06], [LS07a], [Sch15]) which are nonzero on t and zero otherwise. Approximation properties of trivariate splines can be found in [LS07b] and [Lai89].

How to use them to solve partial differential equations based on weak formulations like the finite element method has been discussed in [ALW06] and [Sch15]. We leave the detail to these references.

Chapter 2

A Spline-Based Collocation Method for second order elliptic equations

In this chapter, we propose and study a new collocation method based on multivariate splines for the numerical solution of partial differential equations over the polygonal domain in $\mathbb{R}^d$ for $d \geq 2$. Instead of using a second-order elliptic equation in divergence form:

$$\begin{cases} -\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}(a^{ij}(x)\frac{\partial}{\partial x_j}u) + \sum_{i=1}^{d} b^i(x)\frac{\partial}{\partial x_i}u + c^1(x)u &= f, \quad x \in \Omega \subset \mathbb{R}^d, \\ u &= g, \quad \text{on } \partial\Omega \end{cases} \tag{2.1}$$

which is often used for various finite element methods, as we discuss in this chapter a more general form of second-order elliptic PDE in the non-divergence form:

$$\begin{cases} \sum_{i,j=1}^{d} a^{ij}(x)\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}u + \sum_{i=1}^{d} b^i(x)\frac{\partial}{\partial x_i}u + c(x)u &= f, \quad x \in \Omega \subset \mathbb{R}^d, \\ u &= g, \quad \text{on } \partial\Omega, \end{cases} \tag{2.2}$$

where the PDE coefficient functions $a^{ij}(x), i, j = 1, \cdots, d$ are in $L^\infty(\Omega)$ and satisfy the standard elliptic condition. In addition, when $d \geq 2$, we shall assume the so-called Cordés condition, see (2.26) in a later section or see [SS13].

Numerical solutions to the 2nd order PDE in the non-divergence form has been studied extensively recently. See some studies in [SS13], [LW18], [MY17], [WW19], [Sch19], and etc.. The method in this chapter provides a new and more effective approach. We mainly use the Sobolev space $H^2(\Omega)$. It is known when Ω is convex (cf. [Gri85]), the solution to the Poisson equation with zero boundary condition, i.e. $g = 0$ will be in $H^2(\Omega)$. Recently, the researchers in [GL20] showed that when Ω has an uniformly positive reach, the solution of (2.2) with zero boundary condition will be in $H^2(\Omega)$. Various domains of uniformly positive reach, e.g. star-shaped domain and domains with holes are shown in [GL20]. Many more domains other than convex domains can have H^2 solution.

For any $u \in H^2(\Omega)$, we use the standard H^2 norm

$$\|u\|_{H^2} = \|u\|_{L^2(\Omega)} + \|\nabla u\|_{L^2(\Omega)} + \sum_{i,j=1}^{d} \|\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}u\|_{L^2(\Omega)} \tag{2.3}$$

for all u on $H^2(\Omega)$ and the semi-norm

$$|u|_{H^2} = \sum_{i,j=1}^{d} \|\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}u\|_{L^2(\Omega)}. \tag{2.4}$$

Since we will use multivariate spline functions to approximate the solution $u \in H^2(\Omega)$, we use C^r smooth spline functions with $r \geq 1$ and the degree D of splines sufficiently large satisfying $D \geq 3r+2$ in $\mathbb{R}^2$ and $D \geq 6r+3$ in $\mathbb{R}^3$. Let $S_D^r(\triangle)$ be the spline space of degree D and smoothness r over triangulation or tetrahedralization $\triangle$ of Ω. We now explain our spline-based collocation method. For simplicity, we use the standard Poisson equation which is a special case of the PDE (2.2).

$$\begin{cases} -\Delta u &= f, \quad \text{in } \Omega \subset \mathbb{R}^d, \\ u &= g, \quad \text{on } \partial\Omega. \end{cases} \tag{2.5}$$

When Ω has a uniform positive reach, the solution to the Poisson equation will be in $H^2(\Omega)$. We shall use C^r spline functions with $r \geq 2$ to approximate the solution u. In addition, we shall use the so-called domain points (cf. [LS07a] or the next section) to be the collocation points. Letting $\xi_i, i = 1, \cdots, N$ be the domain points of $\triangle$ and degree $D' > 0$, where D' will be different from D, our multivariate spline-based collocation method is to seek a spline function $s \in S_D^r(\triangle)$ satisfying

$$\begin{cases} -\Delta s(\xi_i) &= f(\xi_i), \quad \forall \xi_i \in \Omega \subset \mathbb{R}^d, \\ s(\xi_i) &= g(\xi_i), \quad \forall \xi_i \in \partial\Omega, \end{cases} \tag{2.6}$$

where $i = 1, \cdots, N$.

One of the key ideas is to let a computer decide how to choose $\mathbf{c}$ to satisfy the smoothness condition $H\mathbf{c} = 0$ and (2.6) above simultaneously. Clearly, (2.6) leads to a linear system that may not have a unique solution. It may be an over-determined linear system if $D' > D$ or an under-determined linear system if $D' < D$. Our method is to use a least squares solution if the system is overdetermined or a sparse solution if the system is under-determined (cf. [LW21]).

To establish the convergence of the spline-based collocation solution as the size of $\triangle$ goes to zero, we define a new norm $\|u\|_L$ on $H^2(\Omega) \cap H_0^1(\Omega)$ for the Poisson equation as follows.

$$\|u\|_L = \|\Delta u\|_{L^2(\Omega)}. \tag{2.7}$$

We will show that the new norm is equivalent to the standard norm on Banach space $H^2(\Omega) \cap H_0^1(\Omega)$. That is,

Theorem 4. *Suppose $\Omega \subset \mathbb{R}^d$ be a bounded domain and the closure of Ω is of uniformly positive reach $r_\Omega > 0$. Then there exist two positive constants A and B such that*

$$A\|u\|_{H^2} \le \|u\|_L \le B\|u\|_{H^2}, \quad \forall u \in H^2(\Omega) \cap H_0^1(\Omega). \tag{2.8}$$

See the proof of Theorem 6 in a later section. Letting $u \in H^2(\Omega) \cap H_0^1(\Omega)$ be the solution of (2.5) with $g = 0$ and u_s be the spline solution of (2.6), we use the first inequality above to have

$$A\|u - u_s\|_{H^2} \le \|u - u_s\|_L.$$

It can be seen from (2.6) that the first equation can be written as

$$\Delta(u_s(\xi_i) - u(\xi_i)) = 0, i = 1, \cdots, N \tag{2.9}$$

which is a discretization of $\|u - u_s\|_L^2$. Let $|\triangle|$ be the size of triangulation or tetrahedralization $\triangle$. Since we can use a spline function to approximate u if u is sufficiently smooth when the size $|\triangle|$ goes to zero (cf. [LS07a]), we seek the minimizer u_s of minimization to be explained in a later section. Then the root mean square error (RMSE) will be small for a sufficiently large amount of collocation points and distributed evenly when the size $|\triangle|$ of $\triangle$ is small. Then our Theorem 4 implies that $\|u - u_s\|_{H^2}$ is small. Furthermore, we will show

$$\|u - u_s\|_{L^2(\Omega)} \leq C|\triangle|^2\|u - u_s\|_L \text{ and } \|\nabla(u - u_s)\|_{L^2(\Omega)} \leq C|\triangle|\|u - u_s\|_L \tag{2.10}$$

for a positive constant C, under the assumption that $u - u_s = 0$ on $\partial\Omega$. These will establish the multivariate spline-based collocation method for the Poisson equation.

In general, we let $\mathcal{L}$ be the PDE operator in (2.11). Note that we begin with the second-order term of the PDE just for convenience.

$$\begin{cases} \displaystyle\sum_{i,j=1}^{d} a^{ij}(x)\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}u & = f, \quad x \in \Omega \subset \mathbb{R}^d, \\ u & = g, \quad \text{on } \partial\Omega, \end{cases} \tag{2.11}$$

We shall similarly define a new norm associated with the PDE (2.11):

$$\|u\|_{\mathcal{L}} = \|\mathcal{L}(u)\|_{L^2(\Omega)}. \tag{2.12}$$

Similarly, we will show the following.

Theorem 5. *Suppose $\Omega \subset \mathbb{R}^d$ be a bounded domain and the closure of Ω is of uniformly positive reach $r_\Omega > 0$. Suppose that the second-order partial differential equation in (2.11) is elliptic, i.e. satisfying (2.25) and satisfies the Cordés condition if $d \geq 2$. There exist two positive constants A_1 and B_1 such that*

$$A_1\|u\|_{H^2} \leq \|u\|_{\mathcal{L}} \leq B_1\|u\|_{H^2}, \quad \forall u \in H^2(\Omega) \cap H_0^1(\Omega). \tag{2.13}$$

See proof in a section later. This result allows us to establish the convergence of the spline-based collocation method for second-order elliptic PDEs in non-divergence form, similarly to the Poisson equation setting. Also, we will have an improved convergence similar to (2.10). Spline-based collocation methods offer several advantages over traditional finite element methods, discontinuous Galerkin methods, virtual element methods, and others. For instance, they do not require a weak formulation of the PDE solution, eliminating the need for numerical quadrature. Additionally, they can easily handle discontinuities in PDE coefficients by adjusting the location of collocation points near the affected areas. Moreover, the multivariate spline-based collocation method allows for increased approximation accuracy by increasing degree of polynomial or adding more collocation points, which can be more cost-effective than uniform refinement of the underlying triangulation or

tetrahedralization. Finally, our spline collocation method includes tuning parameters that provide control over the accuracy and smoothness of the spline solution.

We shall provide many numerical results in 2D and 3D to demonstrate how well the spline-based collocation methods can perform. Mainly, we would like to show the performance of solutions under the various settings: (1) the PDE coefficients are smooth or not very smooth, (2) the PDE solutions are smooth or not very smooth, (3) the domain of interest may not be a uniformly positive reach, even very complicated a domain such as the human head used in the numerical experiment in this thesis, and (4) the dimension d can be 2 or 3. In addition, we shall compare with the existing methods in [ALW06] and [LW18] to demonstrate that the multivariate spline-based collocation method can be better in the sense that it is more accurate and more efficient under the assumption that the associated collocation matrices are generated beforehand. Finally, we remark that we have extended our study to the Biharmonic equation, Navier-Stokes equations and the optimal transport problem.

2.1 A Spline Based Collocation Method for the Poisson Equation

For convenience, we simply explain our method when $d = 2$ in this section. Numerical results in the settings of $d = 2$ and $d = 3$ will be given in later sections.

For a given triangulation $\triangle$, we use a spline space $S_D^r(\triangle)$ to find the coefficient vector $\mathbf{c}$ of spline function $s = \sum_{t\in\triangle}\sum_{i+j+k=D} c_{ijk}^t B_{ijk}^t \in S_D^r(\triangle)$ satisfying the following equations

$$\begin{cases} -\sum_{t\in\triangle}\sum_{i+j+k=D} c_{ijk}^t \Delta B_{ijk}^t(\xi_i) &= f(\xi_i), \quad \xi_i \in \Omega \subset \mathbb{R}^2 \\ s(\xi_i) &= g(\xi_i), \quad \text{on } \partial\Omega, \end{cases} \tag{2.14}$$

where $\{\xi_i\}_{i=1,\cdots,N} \in \mathcal{D}_{D',\triangle}$ are the domain points of $\triangle$ of degree D' as explained in (1.8) in the previous section and $D' > D$. Using these points, we have the following matrix equation:

$$-K\mathbf{c} := \left[-\Delta(B_{ijk}^t(\xi_i))\right]\mathbf{c} = [f(\xi_i)] = \mathbf{f},$$

where $\mathbf{c}$ is the vector consisting of all spline coefficients $c_{ijk}^t, i+j+k = D, t \in \triangle$. In general, the spline s with coefficients in $\mathbf{c}$ is a discontinuous function. In order to make $s \in \mathcal{S}_D^r(\triangle)$, its coefficient vector $\mathbf{c}$ must satisfy the constraints $H\mathbf{c} = 0$ for the smoothness conditions that the $\mathcal{S}_D^r(\triangle)$ functions possess (cf. [LS07a]).

Based on the smoothness conditions (cf. Theorem 2.28 or Theorem 15.38 in [LS07a]), we can construct matrices H_0 for the C^0 smoothness conditions of spline functions and H_r for the C^r smoothness conditions for $r \geq 2$, respectively. Our collocation method is to find $\mathbf{c}^*$ by solving the following constrained minimization:

$$\min_{\mathbf{c}} J(c) = \frac{1}{2}(\alpha\|B\mathbf{c} - \mathbf{g}\|^2 + \beta\|H_r\mathbf{c}\|^2 + \gamma\|H_0\mathbf{c}\|^2) \quad \text{subject to } \|K\mathbf{c} + \mathbf{f}\| \leq \epsilon_1, \tag{2.15}$$

where $B, \mathbf{g}$ are associated with the boundary condition, H_r is associated with the smoothness condition with $r = 2$ and H_0 is associated with the smoothness condition with $r = 0$, $\alpha > 0, \beta > 0, \gamma > 0$ are fixed parameters, and $\epsilon_1 > 0$ is a given tolerance. It is easy to see that the minimization (2.15) is a convex minimization problem over a convex feasible set. The problem (2.15) will have a unique solution if the feasible set is not empty. We shall use the following iterative method to solve the minimization problem (2.15). See Appendix for a derivation and a proof of the convergence of Algorithm 1.

Algorithm 1: Iterative Method

Let I be the identity matrix of $\mathbb{R}^m$. Fix $\epsilon > 0$. Given an initial guess $\lambda^{(0)} \in \mathrm{Im}(K)$, we first compute

$$\mathbf{z}^{(1)} = (\alpha B^\top B + \beta H_r^\top H_r + \gamma H_0^\top H_0 + \frac{1}{\epsilon} K^\top K)^{-1} (\alpha B^\top G + \frac{1}{\epsilon} K^\top \mathbf{f} - K^\top \lambda^{(0)})$$

and iteratively compute

$$(\alpha B^\top B + \beta H_r^\top H_r + \gamma H_0^\top H_0 + \frac{1}{\epsilon} K^\top K)\mathbf{z}^{(k+1)} = (\alpha B^\top B + \beta H_r^\top H_r + \gamma H_0^\top H_0)\mathbf{z}^{(k)} + \frac{1}{\epsilon} K^\top \mathbf{f}$$

for $k = 1, 2, \cdots$, where $\mathrm{Im}(K)$ is the range of K.

Let u_s be the solution of Algorithm 1. We would like to show

$$\|u - u_s\|_{L^2(\Omega)} \leq C|\triangle|^2 \epsilon_1 \tag{2.16}$$

for some constant $C > 0$, where $|\triangle|$ is the size of the underlying triangulation or tetrahedralization $\triangle$ of the domain Ω. To do so, we first show

Lemma 3. *Suppose that Ω is a polygonal domain. Suppose that $u \in H^3(\Omega)$. Then there exists a positive constant $\hat{C}$ depending on $D \geq 1$ and $D' > D$ such that*

$$\|\Delta u(x,y) - \Delta u_s(x,y)\|_{L^2(\Omega)} \leq \epsilon_1 \hat{C}.$$

Proof. Indeed, by Lemma 8, we have a quasi-interpolatory spline s_u satisfying

$$|\Delta u(x,y) - \Delta s_u(x,y)| \leq \epsilon, \forall (x,y) \in \Omega$$

for a triangulation $\triangle$ with $|\triangle|$ small enough. Since $\Delta u(x,y) = -f(x,y)$, we have

$$\|\Delta s_u + \mathbf{f}\| \leq \epsilon_1 \tag{2.17}$$

if ϵ small enough. That is, the feasible set is not empty.

Next we use the minimization (2.15) to have the minimizer u_s satisfying

$$|\Delta u(x_i,y_i) - \Delta u_s(x_i,y_i)| \leq \epsilon_1$$

with sufficiently small $|\triangle|$ for any domain points (x_i, y_i) which construct the collocation matrix K. Now, these two inequalities imply that

$$|\Delta u_s(x_i,y_i) - \Delta s_u(x_i,y_i)| \leq \epsilon_1 + \epsilon_1.$$

Note that $\Delta u_s - \Delta s_u$ is a polynomial over each triangle $t \in \triangle$ which has small values at the domain points. This implies that the polynomial $\Delta u_s - \Delta s_u$ is small over t. That is,

$$|\Delta u_s(x, y) - \Delta s_u(x, y)| \leq C(\epsilon_1 + \epsilon_1) = 2C\epsilon_1 \tag{2.18}$$

by using Theorem 2.27 in [LS07a]. Finally, we can use (2.18) to prove

$$|\Delta u(x, y) - \Delta u_s(x, y)| = |\Delta u(x, y) - \Delta s_u(x, y) + \Delta s_u(x, y) - \Delta u_s(x, y)| \leq \epsilon_1 + 2C\epsilon_1.$$

and then

$$\|\Delta u(x, y) - \Delta u_s(x, y)\|_{L^2(\Omega)} \leq \epsilon_1 \hat{C}$$

for a constant $\hat{C}$ depending on the bounded domain Ω and D, D', but independent of $|\triangle|$. □

Now, let us consider the convergence of our method. Without loss of generality, we may assume $g = 0$. Indeed, for any general g, let $u_g \in H^2(\Omega)$ be a function satisfying the boundary condition, i.e. $u_g|_{\partial\Omega} = g$ and we consider the Poisson equation with solution $w = u - u_g$ and the new right-hand side $f_w = f + \Delta u_g$. Recall the standard norm on $H^2(\Omega)$ defined in (2.3). It is also a norm of $H^2(\Omega) \cap H_0^1(\Omega)$. It is easy to see that the space $H^2(\Omega) \cap H_0^1(\Omega)$ is a Banach space with the norm $\|\cdot\|_{H^2(\Omega)}$. In addition, let us define a new norm $\|u\|_L$ on $H^2(\Omega) \cap H_0^1(\Omega)$ as follows.

$$\|u\|_L = \|\Delta u\|_{L^2(\Omega)} \tag{2.19}$$

We can easily show that $\|\cdot\|_L$ is a norm on $H^2(\Omega) \cap H_0^1(\Omega)$ as follows: Indeed, if $\|u\|_L = 0$, then $\Delta u = 0$ in Ω and $u = 0$ on the boundary $\partial\Omega$. By the Green theorem, we get

$$\int_\Omega |\nabla u|^2 = -\int_\Omega u\Delta u + \int_{\partial\Omega} u\frac{\partial u}{\partial n} = 0.$$

By Poincaré's inequality, we get

$$\|u\|_{L^2(\Omega)} \leq C\|\nabla u\|_{L^2(\Omega)} = 0.$$

Hence, we know that $u = 0$. Next for any scalar a, it is trivial to have

$$\|au\|_L = \|\Delta(au)\|_{L^2(\Omega)} = |a|\|\Delta u\|_{L^2(\Omega)} = |a|\|u\|_L.$$

Finally, the triangular inequality is also trivial.

$$\|u+v\|_L \quad = \quad \|\Delta(u+v)\|_{L^2(\Omega)} \leq \|\Delta u\|_{L^2(\Omega)} + \|\Delta v\|_{L^2(\Omega)} = \|u\|_L + \|v\|_L$$

by the linearity of the Laplacian operator.

We now show that the new norm is equivalent to the standard norm on $H^2(\Omega) \cap H_0^1(\Omega)$. We are now ready to establish the following

Theorem 6. *Suppose $\Omega \subset \mathbb{R}^d$ is a bounded domain and the closure of Ω is of uniformly positive reach $r_\Omega > 0$. There exist two positive constants A and B such that*

$$A\|u\|_{H^2} \leq \|u\|_L \leq B\|u\|_{H^2}, \quad \forall u \in H^2(\Omega) \cap H_0^1(\Omega). \tag{2.20}$$

Proof. We first show that $H^2(\Omega) \cap H_0^1(\Omega)$ is the Banach space with the norm $\|u\|_L$. Assume that $\{u_n\}$ is the Cauchy sequence in $H^2(\Omega) \cap H_0^1(\Omega)$. We know that $\{\Delta u_n\}$ is a Cauchy sequence in $L^2(\Omega)$. Then there exists $U^* \in L^2(\Omega)$ such that Δu_n converges to U^*. It is known there exists a unique u^* satisfying the Dirichlet problem:

$$\begin{cases} \Delta u &= U^* \\ u &= 0. \end{cases}$$

By Theorem 1, we know $u^* \in H^2(\Omega) \cap H_0^1(\Omega)$. Thus, we can say that there exists the unique u^* satisfying $\|u_n - u^*\|_L \to 0$ as $n \to \infty$. It is easy to get the following inequality

$$\|u\|_L \;=\; \|\Delta u\|_{L^2(\Omega)} \leq \sum_{i,j=1}^{d} \|\frac{\partial^2}{\partial x_i \partial x_j} u\|_{L^2(\Omega)} \leq \|u\|_{H^2} \tag{2.21}$$

for all $u \in H^2(\Omega) \cap H_0^1(\Omega)$.

Next, by Theorem 1, more precisely, by (1.4), we have

$$\|u\|_{H^2} \leq C\|\Delta u\|_{L^2(\Omega)} = C\|u\|_L$$

for a constant C dependent on C_0 in (1.4). Therefore, we choose $A = \frac{1}{C}$ to finish the proof. □

Using Theorem 6, we immediately obtain the following theorem

Theorem 7. *Suppose f and g are continuous over bounded domain $\Omega \subseteq \mathbb{R}^d$ for $d \geq 2$. Suppose $\Omega \subset \mathbb{R}^d$ be a bounded domain and the closure of Ω is of uniformly positive reach $r_\Omega > 0$. Suppose that $u \in H^3(\Omega)$ and $(u - u_s)|_{\partial\Omega} = 0$. We have the following inequality*

$$\|u - u_s\|_{L^2(\Omega)} \leq C\epsilon_1, \|\nabla(u - u_s)\|_{L^2(\Omega)} \leq C\epsilon_1$$

and

$$\sum_{i+j=2} \|\frac{\partial^2}{\partial x^i \partial y^j} u\|_{L^2(\Omega)} \leq C\epsilon_1$$

for a positive constant C depending on A and Ω, where A is one of the constants in Theorem 6.

Proof. Using Lemma 3 and the assumption on the approximation on the boundary, we have

$$\|u - u_s\|_{H^2(\Omega)} \quad \leq \frac{1}{A}\|\Delta(u - u_s)\|_{L^2(\Omega)} \leq \frac{1}{A}\epsilon_1\hat{C}.$$

We choose $C = \frac{\hat{C}}{A}$ to finish the proof. □

Finally we show that the convergence of $\|u - u_s\|_{L^2(\Omega)}$ and $\|\nabla(u - u_s)\|_{L^2(\Omega)}$ can be better.

Theorem 8. *Suppose that* $(u - u_s)|_{\partial\Omega} = 0$. *Under the assumptions in Theorem 21, we have the following inequality*

$$\|u - u_s\|_{L^2(\Omega)} \leq C|\triangle|^2\epsilon_1 \text{ and } \|\nabla(u - u_s)\|_{L^2(\Omega)} \leq C|\triangle|\epsilon_1$$

for a positive constant $C = 1/A$, *where* A *is one of the constants in Theorem 6 and* $|\triangle|$ *is the size of the underlying triangulation* $\triangle$.

Proof. First of all, it is known for any $w \in H^2(\Omega) \cap H_0^1(\Omega)$, there is a continuous linear spline L_w over the triangulation $\triangle$ such that

$$\|D_x^\alpha D_y^\beta (w - L_w)\|_{L^2(\Omega)} \leq C|\triangle|^{2-\alpha-\beta}|w|_{H^2(\Omega)} \tag{2.22}$$

for nonnegative integers $\alpha \geq 0, \beta \geq 0$ and $\alpha + \beta \leq 2$, where $|w|_{H^2(\Omega)}$ is the semi-norm of w in $H^2(\Omega)$. Indeed, we can use the same construction method for quasi-interpolatory splines used for the proof of Lemma 8 to establish the above estimate. The above estimate will be used twice below.

By the assumption that $u - u_s = 0$ on $\partial\Omega$, it is easy to see

$$\begin{aligned}
& \|\nabla(u - u_s)\|^2_{L^2(\Omega)} = -\int_\Omega \Delta(u - u_s)(u - u_s) = -\int_\Omega \Delta(u - u_s - L_{u-u_s})(u - u_s) \\
= & \int_\Omega \nabla(u - u_s - L_{u-u_s})\nabla(u - u_s) \leq \|\nabla(u - u_s)\|_{L^2(\Omega)}\|\nabla(u - u_s - L_{u-u_s})\|_{L^2(\Omega)} \\
\leq & \|\nabla(u - u_s)\|_{L^2(\Omega)} C|\triangle| \cdot |u - u_s|_{H^2(\Omega)} \\
\leq & \|\nabla(u - u_s)\|_{L^2(\Omega)}|\triangle|\frac{C}{A}\|\Delta(u - u_s)\|_{L^2(\Omega)}.
\end{aligned}$$

where we have used the first inequality in Theorem 6. It follows that $\|\nabla(u - u_s)\|_{L^2(\Omega)} \leq |\triangle|\frac{C}{A}\epsilon_1$.

Next we let $w \in H^2(\Omega) \cap H_0^1(\Omega)$ be the solution to the following Poisson equation:

$$\begin{cases} -\Delta w &= u - u_s \quad \text{in } \Omega \subset \mathbb{R}^d \\ w &= 0 \quad \text{on } \partial\Omega, \end{cases} \tag{2.23}$$

Then we use the continuous linear spline L_w to have

$$\begin{aligned} \|u - u_s\|^2_{L^2(\Omega)} &= -\int_\Omega \Delta w(u - u_s) = -\int_\Omega \Delta(w - L_w)(u - u_s) \\ &= \int_\Omega \nabla(w - L_w)\nabla(u - u_s) \le \|\nabla(u - u_s)\|_{L^2(\Omega)}\|\nabla(w - L_w)\|_{L^2(\Omega)} \\ &\le \|\nabla(u - u_s)\|_{L^2(\Omega)} C|\triangle| \cdot |w|_{H^2(\Omega)} \le \frac{C}{A}|\triangle|\epsilon_1|\triangle|\frac{C}{A}\|\Delta w\|_{L^2(\Omega)} \\ &= \frac{C}{A}|\triangle|\epsilon_1|\triangle|\frac{C}{A}\|u - u_s\|_{L^2(\Omega)}. \end{aligned}$$

where we have used the first inequality in Theorem 6 and the estimate of $\|\nabla(u - u_s)\|_{L^2(\Omega)}$ above. Hence, we have $\|u - u_s\|_{L^2(\Omega)} \le \frac{C^2}{A^2}|\triangle|^2\epsilon_1$ as $|\triangle| \to 0$. □

2.2 General Second Order Elliptic Equations

In this section we consider a collocation method based on bivariate/trivariate splines for a solution of the general second order elliptic equation in (2.2). For the PDE coefficient functions $a^{ij}, b^i, c^1 \in L^\infty(\Omega)$, we assume that

$$a_{ij} = a_{ji} \in L^\infty(\Omega) \ \ \forall i, j = 1, \cdots, d \tag{2.24}$$

and there exist λ, Λ such that

$$\lambda \sum_{i=1}^{d} \eta_i^2 \leq \sum_{i,j}^{d} a^{ij}(x)\eta_i\eta_j \leq \Lambda \sum_{i=1}^{d} \eta_i^2, \forall \eta \in \mathbb{R}^d \backslash \{0\} \tag{2.25}$$

for all i, j and $x \in \Omega$. For convenience, we first assume that b^i=0 and $c^1 = 0$ in this section. In addition to the elliptic condition, we add the Cordés condition for the well-posedness of the problem. We assume that there is an $\epsilon \in (0, 1]$ such that

$$\frac{\sum_{i,j=1}^{d}(a^{i,j})^2}{(\sum_{i=1}^{d} a^{ii})^2} \leq \frac{1}{d-1+\epsilon} \ \ a.e.\ in\ \Omega \tag{2.26}$$

Next let $\theta \in L^\infty(\Omega)$ be defined by

$$\theta := \frac{\sum_{i=1}^{d} a^{ii}}{\sum_{i,j=1}^{d}(a^{i,j})^2}.$$

Under these conditions, the researchers in [SS13] proved the following lemma

Lemma 4. *Let the operator* $\mathcal{L}(u) := \sum_{i,j=1}^{d} a^{ij}(x)\frac{\partial^2}{\partial x_i \partial x_j}u$ *satisfy* (2.24)*,* (2.25) *and* (2.26)*. Then for any open set* $U \subseteq \Omega$ *and* $v \in H^2(U)$*, we have*

$$|\theta \mathcal{L}v - \Delta v| \leq \sqrt{1-\epsilon}|D^2 v| \ \ a.e.\ in\ U, \tag{2.27}$$

where $\epsilon \in (0, 1]$ *is as in* (2.26)*.*

Instead of using the convexity to ensure the existence of the strong solution of (2.2) in [SS13], we shall use the concept of uniformly positive reach in [GL20]. The following is just the restatement of Theorem 3.3 in [GL20].

Theorem 9. *Suppose that $\Omega \subset \mathbb{R}^d$ with $d \geq 2$ is a bounded domain with uniformly positive reach. Then the second order elliptic PDE in (2.2) satisfying (2.26) has a unique strong solution in $H^2(\Omega)$.*

We now extend the collocation method in the previous section to find a numerical solution of (2.2). Similar to the discussion in the previous section, we can construct the following matrix for the PDE in (2.2):

$$\mathcal{K} := \left[\sum_{i,j=1}^{d} a^{ij}(\xi_i) \frac{\partial^2}{\partial x_i \partial x_j} (B^t_{ijk}(\xi_i)) \right].$$

Similar to (2.15), consider the following minimization problem:

$$\min_{\mathbf{c}} J(c) = \frac{1}{2}(\alpha \|B\mathbf{c} - \mathbf{g}\|^2 + \beta \|H\mathbf{c}\|^2 + \gamma \|H_0\mathbf{c}\|^2) \quad \text{subject to } -\mathcal{K}\mathbf{c} = \mathbf{f}, \tag{2.28}$$

Again we will solve a nearby minimization problem as in the previous section. Like the Poisson equation, we let $\epsilon_1 = \|\mathcal{K}\mathbf{c} + \mathbf{f}\|$ for the minimizer $\mathbf{c}$ of (2.28). To study the convergence, we may assume that $g = 0$ as in the previous section so that the solution u_s with the coefficient vector $\mathbf{c}$ which is the minimizer of (2.28) satisfies $u_s = 0$ on $\partial\Omega$ and hence, $\|u - u_s\|_{L^2(\partial\Omega)} = 0$. Also, we have that $\|\mathcal{L}u_s + f\|_{L^2(\Omega)} \leq \epsilon_1$. To show u_s approximate u over Ω, let us define a new norm $\|u\|_{\mathcal{L}}$ on $H^2(\Omega) \cap H^1_0(\Omega)$ as follows.

$$\|u\|_{\mathcal{L}} = \|\mathcal{L}u\|_{L^2(\Omega)} \tag{2.29}$$

We can show that $\|\cdot\|_{\mathcal{L}}$ is a norm on $H^2(\Omega)\cap H_0^1(\Omega)$ as follows if $\epsilon\in(0,1]$ is large enough. Indeed, if $\|u\|_{\mathcal{L}}=0$, then $\mathcal{L}u=0$ in Ω and $u=0$ on the boundary $\partial\Omega$. Using this Lemma 4 and Theorem 6, we get

$$\int_\Omega \Delta u\Delta u-\int_\Omega(\Delta-\theta\mathcal{L})u\Delta u=\int_\Omega\theta\mathcal{L}(u)\Delta u=0 \tag{2.30}$$

and

$$\begin{aligned}&\textstyle\int_\Omega\Delta u\Delta u-\int_\Omega(\Delta-\theta\mathcal{L})u\Delta u\geq\int_\Omega|\Delta u|^2-\int_\Omega\sqrt{1-\epsilon}|D^2u|\cdot|\Delta u|\\&=\textstyle\int_\Omega|\Delta u|^2-\int_\Omega\sqrt{1-\epsilon}|D^2u|\cdot|\Delta u|\geq\|\Delta u\|^2-\frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|\|\Delta u\|.\end{aligned}$$

Therefore, if $\epsilon>1-A^2$, then

$$(1-\frac{\sqrt{1-\epsilon}}{A})\|\Delta u\|\leq 0.$$

Hence, we know that $u=0$. The other two properties of the norm can be proved easily.

We mainly show that the above norm is equivalent to the standard norm on $H^2(\Omega)\cap H_0^1(\Omega)$. Indeed, recall a well-known property about the norm equivalence.

Lemma 5. *([Brezis, 2011 [Bre11]]) Let E be a vector space equipped with two norms, $\|\cdot\|_1$ and $\|\cdot\|_2$. Assume that E is a Banach space for both norms and that there exists a constant $C>0$ such that*

$$\|x\|_2\leq C\|x\|_1,\ \forall x\in E. \tag{2.31}$$

Then the two norms are equivalent, i.e., there is a constant $c > 0$ such that

$$\|x\|_1 \leq c_1 \|x\|_2, \ \forall x \in E.$$

Using Lemma 5, we can prove the following theorem

Theorem 10. *Suppose that Ω is bounded and has uniformly positive reach $r_\Omega > 0$. Then there exist two positive constants A_1 and B_1 such that*

$$A_1 \|u\|_{H^2(\Omega)} \leq \|u\|_{\mathcal{L}} \leq B_1 \|u\|_{H^2(\Omega)}, \quad \forall u \in H^2(\Omega) \cap H_0^1(\Omega). \tag{2.32}$$

Proof. It follows that

$$\|u\|_{\mathcal{L}} \quad \leq \quad \max_{i,j=1\cdots,d} \|a^{ij}\|_\infty \sum_{i,j=1}^{d} \|\frac{\partial^2}{\partial x_i \partial x_j} u\|_{L^2(\Omega)} \leq B_1 \|u\|_{H^2(\Omega)}$$

for all $u \in H^2(\Omega) \cap H_0^1(\Omega)$, where B_1 depending on d, Λ and C. Using Lemma 4 and the above inequality, there exist $\alpha_1 > 0$ satisfying

$$\|u\|_{H^2} \leq \alpha_1 \|u\|_{\mathcal{L}}.$$

Therefore, we choose $A_1 = \frac{1}{\alpha_1}$ to finish the proof. □

Theorem 11. *Let Ω be a bounded and closed set satisfying the uniformly positive reach condition. Assume that $a^{ij} \in L^\infty(\Omega)$ satisfy* (2.24), (2.25) *and* (2.26) *and $\epsilon > 1 - A^2$. Suppose that $u \in H^3(\Omega)$ and $u - u_s = 0$ on $\partial\Omega$. For the solution u of equation* (2.11) *and the corresponding minimizer u_s,*

we have the following inequality

$$\|u - u_s\|_{L^2(\Omega)} \leq C\epsilon_1$$

for a positive constant C depending on Ω *and* A_1 *which is one of the constants in Theorem 7. Similar for* $\|\nabla(u - u_s)\|_{L^2(\Omega)}$ *and* $|u - u_s|_{H^2}$.

Next we consider the case that b^i and c^1 are not zero. Assume that $\|a^{ij}\|_\infty$, $\|b^i\|_\infty$, $\|c^1\|_\infty \leq \Lambda_1$ and we denote that $\mathcal{L}_1(u) := \sum_{i,j=1}^d a^{ij}(x)\frac{\partial^2}{\partial x_i \partial x_j}u + \sum_{i=1}^d b^i(x)\frac{\partial}{\partial x_i}u + c^1(x)u$ and define a new norm $\|u\|_{\mathcal{L}_1}$ on $H^2(\Omega) \cap H_0^1(\Omega)$ as follows.

$$\|u\|_{\mathcal{L}_1} = \|\mathcal{L}_1 u\|_{L^2(\Omega)}. \tag{2.33}$$

Assume that $\|u\|_{\mathcal{L}_1} = 0$, i.e., $\mathcal{L}_1 u = 0$ over Ω and $u = 0$ on $\partial\Omega$. From (2.27), we have

$$\int_\Omega \theta \mathcal{L}(u)\Delta u \geq \|\Delta u\|^2 - \frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|^2.$$

Then by the above inequality we get

$$\begin{aligned}
0 &= \int_\Omega \theta \mathcal{L}_1(u)\Delta u = \int_\Omega \theta \mathcal{L}(u)\Delta u + \sum_{i=1}^{d} \theta b^i(x)\frac{\partial}{\partial x_i} u\Delta u + \theta c^1(x) u\Delta u \\
&\geq \|\Delta u\|^2 - \frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|^2 + \int_\Omega \sum_{i=1}^{d} \theta b^i(x)\frac{\partial}{\partial x_i} u\Delta u + \theta c^1(x) u\Delta u \\
&\geq \|\Delta u\|^2_{L^2(\Omega)} - \frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|^2_{L^2(\Omega)} - \|\theta\|_\infty \max_i \|b^i\|_\infty \sqrt{d}\|\nabla u\|_{L^2(\Omega)}\|\Delta u\|_{L^2(\Omega)} \\
&\quad -\|\theta\|_\infty \|c^1\|_\infty \|u\|_{L^2(\Omega)}\|\Delta u\|_{L^2(\Omega)} \\
&= \|\Delta u\|^2_{L^2(\Omega)} - \frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|^2_{L^2(\Omega)} - C_m(\|\nabla u\|_{L^2(\Omega)}\|\Delta u\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)}\|\Delta u\|_{L^2(\Omega)}),
\end{aligned}$$

where $C_m = \max\{\|\theta\|_\infty \max_i \|b^i\|_\infty \sqrt{d}, \|\theta\|_\infty \|c^1\|_\infty\}$. Dividing $\|\Delta u\|_{L^2(\Omega)}$ both sides of the inequality above and using Theorem 4, it is followed that

$$\begin{aligned}
0 &\geq \|\Delta u\|_{L^2(\Omega)} - \frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|_{L^2(\Omega)} - C_m(\|\nabla u\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)}) \\
&\geq \|\Delta u\|_{L^2(\Omega)} - \frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|_{L^2(\Omega)} - C_m\|u\|_{H^2(\Omega)} \\
&\geq \|\Delta u\|_{L^2(\Omega)} - \frac{\sqrt{1-\epsilon}}{A}\|\Delta u\|_{L^2(\Omega)} - \frac{C_m}{A}\|\Delta u\|_{L^2(\Omega)} \\
&= \|\Delta u\|_{L^2(\Omega)}(1 - \frac{\sqrt{1-\epsilon}}{A} - \frac{C_m}{A}).
\end{aligned}$$

If the constant $(1 - \frac{\sqrt{1-\epsilon}}{A} - \frac{C_m}{A})$ is positive, then we can conclude that $\Delta u = 0$. Together with the fact $u = 0$ on $\partial\Omega$, we know $u = 0$. The other properties $\|u + v\|_{\mathcal{L}_1} \leq \|u\|_{\mathcal{L}_1} + \|v\|_{\mathcal{L}_1}$ and $\|au\|_{\mathcal{L}_1} = |a|\|u\|_{\mathcal{L}_1}$ can be easily proved. The detail is omitted.

Theorem 12. *Assume that* $(1 - \frac{\sqrt{1-\epsilon}}{A} - \frac{C_m}{A}) > 0$. *There exist two positive constants* A_2 *and* B_2 *such that*

$$A_2\|u\|_{H^2(\Omega)} \leq \|u\|_{\mathcal{L}_1} \leq B_2\|u\|_{H^2(\Omega)}, \quad \forall u \in H^2(\Omega) \cap H_0^1(\Omega). \tag{2.34}$$

Proof. The proof can be done by using Lemma 5. We leave it to the interested reader. □

Therefore, we can get the following theorem for the general elliptic PDE:

Theorem 13. *Suppose* $\Omega \subset \mathbb{R}^d$ *be a bounded domain and the closure of* Ω *is of uniformly positive reach* $r_\Omega > 0$. *Assume that* $a^{ij}, b^i, c^1 \in L^\infty(\Omega)$ *satisfy* (2.24), (2.25), (2.26) *and* $(1 - \frac{\sqrt{1-\epsilon}}{A} - \frac{C_m}{A}) > 0$. *Suppose that* $u \in H^3(\Omega)$ *and* $u - u_s = 0$ *on* $\partial\Omega$. *For the solution* u *of equation* (2.2) *and the corresponding minimizer* u_s, *we have the following inequality*

$$\|u - u_s\|_{L^2(\Omega)} \leq C\epsilon_1$$

for a positive constant C *depending on* Ω *and a constant* A_2 *in Theorem 12.*

Finally we show that the convergence of $\|u - u_s\|_{L^2(\Omega)}$ and $\|\nabla(u - u_s)\|_{L^2(\Omega)}$ can be better

Theorem 14. *Suppose that the bounded domain* Ω *has a uniformly positive reach. Suppose* f *and* g *are continuous over bounded domain* $\Omega \subseteq \mathbb{R}^d$ *for* $d = 2, 3$. *Let* u *be the solution of the general second order PDE* (2.2) *with differential operator* $\mathcal{L}$. *Suppose that* $u \in H^3(\Omega)$. *If* $u - u_s|_{\partial\Omega} = 0$, *we further have the following inequality*

$$\|u - u_s\|_{L^2(\Omega)} \leq C|\triangle|^2\epsilon_1 \text{ and } \|\nabla(u - u_s)\|_{L^2(\Omega)} \leq C|\triangle|\epsilon_1$$

for a positive constant $C = 1/A_2$, where A_2 is one of the constants in Theorem 6 and $|\triangle|$ is the size of the underlying triangulation $\triangle$.

Proof. The proof is similar to Theorem 8. We leave the detail to the interested reader. □

2.3 Implementation of the Spline-based Collocation Method

Before presenting our computational results for the Poisson equation and general second-order elliptic equations, let us first describe the implementation of our spline-based collocation method. We divide the implementation into two parts.

The first part of the implementation involves constructing the collocation matrices K for the Poisson equation and $\mathcal{K}$ for the general second-order PDE in non-divergence form over triangulation/tetrahedralization, based on the degree D of spline functions, smoothness $r \geq 1$, and the domain points $\mathcal{D}_{D',\triangle}$ associated with the triangulation/tetrahedralization. This part also generates the smoothness matrix H_r, H_0. More specifically, for the Poisson equation, we construct collocation matrices:

$$MxxV := [(B^t_{ijk}(\mathbf{x})_{xx}|_{\mathbf{x}=\xi_\ell}, \xi_\ell \in \mathcal{D}_{D',\triangle}] \text{ and } MyyV := [(B^t_{ijk}(\mathbf{x})_{yy}|_{\mathbf{x}=\xi_\ell}, \xi_\ell \in \mathcal{D}_{D',\triangle}]. \quad (2.35)$$

We also choose additional points, beyond the domain points, to build $MxxV$ and $MyyV$ to improve accuracy. For example, we select $D' = D + 3$ to generate domain points. Then, $K = MxxV + MyyV$ has a size of $M \times m$ for the Poisson equation, where $m = \dim(S^{-1}D(\triangle))$ and $M = \dim(S^{-1}D'(\triangle))$. After generating matrices, we save them for later use in solving the Poisson equation for various right-hand side functions and boundary conditions.

For the general elliptic equations, we generate all related matrices $MxxV$, $MxyV$, $MyyV$, $MxV, MyV, \cdots$ similarly to the matrices $MxxV, MyyV$ for the Poisson equation. Then, we generate the collocation matrix $\mathcal{K}$ associated with the PDE coefficients at the same domain points using all the generated matrices $MxxV, MxyV, MyyV, MxV, MyV, \cdots$. This step is the most time-consuming. See Tables 2.1 and 2.2 for the 2D and 3D settings.

The second part, Part 2, involves constructing the right-hand side vector $\mathbf{f}$ for each given PDE problem and the matrix B and vector G associated with the boundary condition. We then use Algorithm 1 to solve the minimization problems (2.15) and (2.28). We test the performance of our collocation method using four different domains in 2D, as shown in Figure 1.1 and 2.1, and four different domains in 3D, as shown in Figure 2.2. Furthermore, the spline-based collocation method has been tested on numerous other domains of interest. In particular, many domains that may not have a positive reach are used for testing, and their numerical results can be found in this thesis.

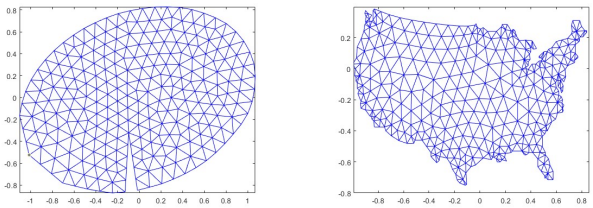


Figure 2.1: Several 2D domains used for Numerical Experiments

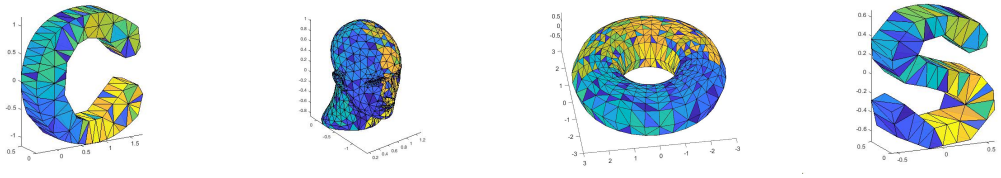


Figure 2.2: Several 3D domains used for Numerical Experiments

Table 2.1: Times in seconds for generating necessary matrices for each 2D domain in Figure 1.1 and 2.1.

Domains	Number of vertices	Number of triangles	degree	Time (P)	Time (G)	Time (UGA P)	Time (UGA G)
Moon	325	531	8	0.48	4.51	1.23	1.80
Flower	297	494	8	0.38	2.47	0.28	0.72
Star	231	366	8	0.30	1.49	0.30	0.71
Circle	525	895	8	0.85	5.83	0.32	1.86
Crack	258	453	8	0.34	2.00	0.21	0.66
US	396	591	8	0.93	2.97	0.20	0.84

In our computational experiments, we utilize a cluster computer at the University of Georgia to generate the related collocation matrices for various degrees of splines and domain points, as described in Part I. We employ multiple CPUs to enable the simultaneous processing of multiple operations. For the 2D case, we use 10 processors on a parallel computer equipped with a 12th Gen Intel(R) Core(TM) i7-12650H processor running at 2.30 GHz and 16.0 GB of installed RAM for both Part 1 and Part 2. Additionally, we use a high-memory (512GB) node from the Sapelo 2 cluster at the University of Georgia, which features four AMD Opteron 6344 2.6 GHz processors. By using 48 processors on the UGA cluster, we can generate the necessary matrices, and the computational times for Part 1 are listed in Table 2.1. For the 3D case, we employ 48 processors for Part 1 and 12 processors for Part 2 to perform the computations. Tables 2.2 and 2.4 display the computational times for generating collocation matrices, where (P) and (UGA P) indicate the time for the Poisson equation with 10 and 48 processors, respectively, and (G) and (UGA G) represent the general second-order PDE using 10 and 48 processors, respectively. Also, we compare the CPU times with the degree $D = 5$ and 9 in Table 2.3.

Another aspect of our computation is determining the values for α, β, and γ in our Algorithm 1. Since there are multiple numerical solutions, we need to decide which one to select. If we prioritize

Table 2.2: Times in seconds for generating necessary matrices for each 3D domain in Figure 2.2.

Domains	Number of vertices	Number of tetrahedron	Degree of splines	Time (P)	Time (G)	Time (UGA P)	Time (UGA G)
Letter C	190	431	9	7.83	69.0	3.17	15.8
Letter S	115	171	9	2.4	25.8	0.72	5.37
Torus	773	2911	9	41.0	451.0	8.39	82.0
Human head	913	1588	9	21.9	243.3	4.53	44.7

Table 2.3: Times in seconds for generating necessary matrices for each 3D domain in Figure 2.2 with degree $D = 5, 9$

Domains	Number of		Degree = 5		Degree = 9	
	Vertices	Tetrahedra	Time (P)	Time (G)	Time (P)	Time (G)
Letter C	190	431	1.30	4.53	5.9	69.0
Letter S	115	171	0.24	0.76	2.4	25.8
Torus	773	2911	1.87	9.89	41.0	451.0
Human head	913	1588	0.81	5.52	21.9	243.3

the accuracy of the numerical solutions over the smoothness of the spline solutions, we use $\alpha \geq 100$ while setting $\beta = \gamma = 1$. Conversely, if we are more interested in the smoothness of the spline solutions, such as in computer-aided geometric design, we use $\alpha = 1$ while setting $\beta = \gamma \geq 10^5$ or $\beta = 1$ and $\gamma = 10^5$.

To demonstrate these phenomena, let's consider Figure 2.3 and 2.4. The numerical results in these figures are based on spline functions of degree $D = 8$ and smoothness $r = 2$ over one of the four domains in Figure 1.1 for all the testing functions listed in the next subsection. In this section, the errors are calculated based on $NI = 1001 \times 1001$ equally-spaced points $(\eta_i)_{i=1}^{NI}$ within

Table 2.4: Times in seconds for finding solutions of 3D Poisson equation(P), general second-order elliptic equation with smooth PDE coefficients (SG) or with non-smooth PDE coefficients (NSG1, NSG2) for each domain in Figure 2.2.

Domain	Time (P)	Time (SG)	Time (NSG1)	Time (NSG2)
Letter C	3.71	6.04	6.07	5.88
Letter S	2.10	2.41	2.33	2.44
Torus	402.13	595.74	285.42	181.83
Human head	27.21	48.10	48.96	48.96

the different domains. The errors have been computed according to the norms:

$$\begin{cases} |u|_{l_2} & = \sqrt{\frac{\sum_{i=1}^{NI}(u(i))^2}{NI}} \\ |u|_{h_1} & = \sqrt{\frac{\sum_{i=1}^{NI}(u(i))^2+(u_x(i))^2+(u_y(i))^2}{NI}} \\ |u|_{l_\infty} & = \max|u(i)|, \end{cases}$$

where $u(i) := u(\eta_i), u_x(i) := u_x(\eta_i), u_y(i) := u_y(\eta_i)$ for given functions u, u_x, u_y. The rooted mean square (RMS) of vectors $e_s = u - u_s$ and the maximum error of $e_s, H_0\mathbf{c}, H_r\mathbf{c}$ are calculated based on those 1001^2 equally-spaced points within the bounding box of the domain.

As β increases from 1 to 10^5, the accuracies of the smoothness $|H_0c|_{l_\infty}$ and $|Hc|_{l_\infty}$ decrease, meaning that the smoothness relations can be enforced exactly. However, the errors $|e_s|_{l_2}$ and $|e_s|_{h_1}$ increase. Figure 2.4 shows that we obtain better numerical solutions when $\alpha = 100 > 1 = \beta = \gamma$ for some testing functions, but a worse approximation for others. Our method provides the advantage of controlling the output to produce a smoother but less accurate numerical solution or a more accurate but slightly bumpy solution. In this chapter, we emphasize the accuracy of spline solutions when reporting our numerical results, which can be compared with standard FEM or DC methods. For the numerical experiments in subsequent sections, we choose $\alpha = 10^2, \beta = 1$, and $\gamma = 1$ to achieve better l_2, h_1 errors.

2.4 Numerical results for the Poisson Equation

We shall present computational results for 2D Poisson equation and 3D Poisson equations separately in the following two subsections. In each section, we first present the computational results from

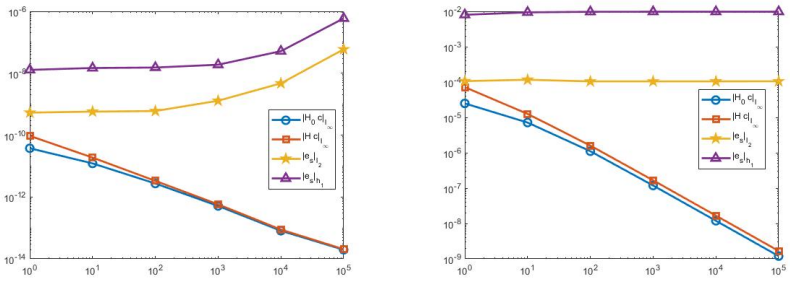


Figure 2.3: The accuracies of the solutions $|e_s|_{l_2}, |e_s|_{h_1}$ and the smoothness $|H_0 c|_{l_\infty}, |Hc|_{l_\infty}$ based on testing functions u^{s5}(left) and u^{s8}(right) with $\alpha = \gamma = 1$ for various β

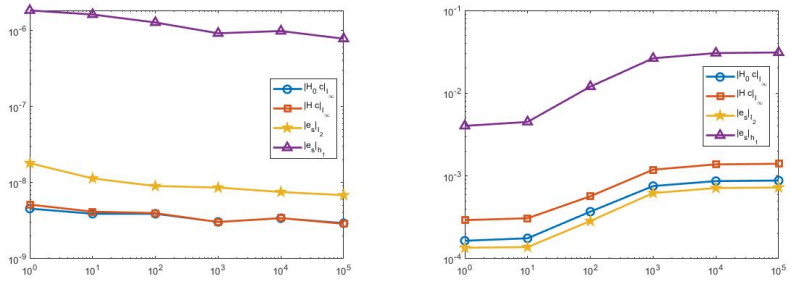


Figure 2.4: The accuracies of the solutions $|e_s|_{l_2}, |e_s|_{h_1}$ and the smoothness $|H_0 c|_{l_\infty}, |Hc|_{l_\infty}$ based on testing functions u^{s5}(left) and u^{s8}(right) with $\beta = \gamma = 1$ for various α

the spline based collocation method to demonstrate the accuracy the method can achieve. Then we present a comparison of our collocation method with the numerical method proposed in [ALW06] which uses multivariate splines to find the weak solution like finite element method. For convenience, we shall call our spline based collocation method the LL method and the numerical method in [ALW06] the AWL method.

2.4.1 Numerical examples for 2D Poisson equations

We have used various triangulations over various bounded domains to experiment the performance of our Algorithm 1. and tested many solutions to the Poisson equation to see the accuracy that the LL method can do. For convenience, we shall only present a few of the computational results based on the domains in Figure 1.1. The following is a list of 10 testing functions (8 smooth solutions and 2 not very smooth)

$$
\begin{aligned}
u^{s1} &= e^{\frac{(x^2+y^2)}{2}}, \\
u^{s2} &= \cos(xy) + \cos(\pi(x^2+y^2)), \\
u^{s3} &= \frac{1}{1+x^2+y^2}, \\
u^{s4} &= \sin(\pi(x^2+y^2)) + 1, \\
u^{s5} &= \sin(3\pi x)\sin(3\pi y), \\
u^{s6} &= \arctan(x^2-y^2), \\
u^{s7} &= -\cos(x)\cos(y)e^{-(x-\pi)^2-(y-\pi)^2} \\
u^{s8} &= \tanh(20y-20x^2) - \tanh(20x-20y^2), \\
u^{ns1} &= |x^2+y^2|^{0.8} \text{ and}
\end{aligned}
$$

$$u^{ns2} = (xe^{1-|x|} - x)(ye^{1-|y|} - y).$$

Note that the testing function in u^{s8} is notoriously difficult to solve. One has to use a good adaptive triangulation method (cf. [LM17]). The rooted mean square (RMS) of $u - u_s$ and $\nabla(u - u_s)$ of approximate spline solution u_s against the exact solution u are given in Table 2.5. These errors are computed based on 1001×1001 equally-spaced points of the bounding box of a domain in in Figure 1.1 which fell inside the domain. We chose collocation points to create $M \times m$ matrix K, where m is the number of Bernstein basis functions (the dimension of spline space $S_D^{-1}(\triangle)$) and Algorithm 1 is used to find the numerical solutions.

Table 2.5: The RMS of errors $u - u_s$ and $\nabla u - \nabla u_s$ for Poisson equations for four domains showed in Figure 1.1 when $r = 2$ and $D = 8$.

	Moon		Flower with a hole		Star with 2 holes		Circle with 3 holes	
Solution	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$
u^{s1}	6.95e-11	4.15e-10	1.23e-11	1.54e-10	1.67e-12	6.57e-11	1.63e-11	1.68e-10
u^{s2}	3.53e-10	4.81e-09	1.83e-11	8.79e-10	2.46e-12	9.77e-11	2.65e-11	2.55e-10
u^{s3}	2.58e-11	1.81e-10	6.96e-12	9.48e-11	1.48e-12	5.66e-11	8.03e-12	8.73e-11
u^{s4}	2.53e-10	3.57e-09	2.19e-11	6.80e-10	1.45e-12	8.41e-11	1.92e-11	2.00e-10
u^{s5}	6.16e-08	1.44e-06	7.73e-09	2.57e-07	3.02e-10	1.36e-08	5.33e-10	1.87e-08
u^{s6}	1.75e-11	2.86e-10	3.23e-12	8.71e-11	2.97e-13	7.23e-12	7.51e-12	7.85e-11
u^{s7}	3.07e-12	2.27e-11	1.15e-12	1.51e-11	2.81e-13	6.69e-12	1.10e-12	1.28e-11
u^{s8}	1.06e-03	9.32e-02	8.65e-04	8.38e-02	4.84e-05	3.36e-03	5.21e-04	2.09e-02
u^{ns1}	7.31e-10	3.68e-08	5.18e-06	4.94e-04	2.62e-06	3.89e-04	1.80e-05	3.22e-04
u^{ns2}	3.16e-04	2.61e-03	7.39e-05	1.51e-03	2.75e-05	9.76e-04	1.91e-05	6.25e-04

From Table 2.5, we can see that the performance of our method is excellent. Next let us compare with the numerical method in [ALW06] for the same degree, the same smoothness, and the same triangulation. The comparison results are shown in Table 2.6. One can see that both methods perform very well. Our method can achieve a better accuracy due to the reason the more number of collocation points is used than the dimension of spline space $S_D^{-1}(\triangle)$.

Finally, we summarize the computational times for both methods in Table 2.7. One can see the LL method can be more efficient if the collocation matrices are already generated. The LL method

Table 2.6: RMSE of spline solutions for the Poisson equation over the four domains in Figure 1.1 when $r = 2$ and $D = 8$ for both the AWL method and the LL method.

	Moon		Flower with a hole		Star with 2 holes		Circle with 3 holes	
Solution	AWL	LL	AWL	LL	AWL	LL	AWL	LL
u^{s1}	1.51e-07	6.95e-11	1.14e-07	1.23e-11	2.08e-07	1.67e-12	5.22e-08	1.63e-11
u^{s2}	1.33e-07	3.53e-10	3.79e-07	1.83e-11	8.93e-07	2.46e-12	2.35e-08	2.65e-11
u^{s3}	4.94e-08	2.58e-11	8.07e-08	6.96e-12	1.44e-07	1.48e-12	1.62e-08	8.03e-12
u^{s4}	5.77e-07	2.53e-10	3.89e-07	2.19e-11	4.51e-07	1.45e-12	2.02e-07	1.92e-11
u^{s5}	1.58e-06	6.16e-08	1.43e-06	7.73e-09	1.67e-06	3.02e-10	2.65e-07	5.33e-10
u^{s6}	5.00e-07	1.75e-11	1.44e-07	3.23e-12	4.03e-07	2.97e-13	9.47e-08	7.51e-12
u^{s7}	1.99e-08	3.07e-12	2.20e-08	1.15e-12	3.30e-08	2.81e-13	4.97e-09	1.10e-12
u^{s8}	1.31e-03	1.06e-03	1.19e-03	8.65e-04	1.49e-04	4.84e-05	7.96e-04	5.21e-04
u^{ns1}	1.50e-07	7.31e-10	2.39e-04	5.18e-06	2.26e-05	2.62e-06	1.43e-05	1.80e-05
u^{ns2}	1.38e-03	3.16e-04	4.55e-04	7.39e-05	9.87e-05	2.75e-05	8.57e-05	1.91e-05

Table 2.7: The number of vertices, triangles and the averaged time for solving the 2D Poisson equation for each domain in Figure 1.1.

Domain	Number of vertices	Number of triangles	Average time for AWL method	Average time for LL method (part 2)
Moon	325	531	9.61e-01	6.28e-01
Flower with a hole	297	494	8.05e-01	5.39e-01
Star with 2 holes	231	366	5.53e-01	3.97e-01
Circle with 3 holes	525	895	1.44e+00	9.74e-01

can be useful for time dependent PDE such as the heat equation. We only need to generate the collocation matrix once and use it repeatedly for many time step iterations.

2.4.2 Numerical results for the 3D Poisson equation

We have used our collocation method to solve the 3D Poisson equation and the tested 10 smooth and non-smooth solution over various domains. For convenience, we only show a few computational results to demonstrate that our collocation method works very well. Our testing solutions are as follows:

$$
\begin{aligned}
u^{3ds1} &= \sin(2x + 2y)\tanh(\frac{xz}{2}) \\
u^{3ds2} &= e^{\frac{x^2+y^2+z^2}{2}}
\end{aligned}
$$

Table 2.8: RMS of error vectors $u - u_s$ and $\nabla u - \nabla u_s$ for the 3D Poisson equation over the four domains in Figure 2.2 based on trivariate spline functions of smoothness $r = 1$ and degree $D = 9$

	Letter C		Letter S		Torus		Human head	
Solution	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$
u^{3ds1}	2.31e-11	2.52e-10	3.01e-12	4.58e-11	7.87e-11	1.40e-09	4.12e-10	5.02e-09
u^{3ds2}	5.47e-10	4.86e-09	7.53e-12	7.31e-11	4.52e-09	3.24e-08	1.66e-08	1.29e-07
u^{3ds3}	5.49e-07	8.40e-06	8.87e-08	7.80e-07	3.32e-09	3.21e-08	9.96e-06	1.65e-04
u^{3ds4}	4.83e-09	5.09e-08	4.29e-09	3.85e-08	2.16e-09	1.61e-08	1.13e-08	2.21e-07
u^{3ds5}	6.49e-07	1.67e-05	1.17e-07	9.47e-07	7.07e-09	5.78e-08	3.62e-06	5.88e-05
u^{3ds6}	3.52e-09	3.99e-08	8.39e-10	6.53e-09	2.03e-08	1.72e-07	6.69e-08	6.90e-07
u^{3ds7}	9.14e-06	8.63e-05	3.20e-06	2.44e-05	1.40e-07	4.75e-06	4.31e-05	8.24e-04
u^{3ds8}	2.05e-08	2.79e-07	3.30e-09	3.35e-08	1.76e-10	2.98e-09	1.90e-08	3.94e-07
u^{3dns1}	8.80e-06	4.66e-04	3.17e-05	1.14e-03	2.23e-09	1.80e-08	8.28e-06	2.18e-03
u^{3dns2}	8.39e-05	1.20e-03	4.30e-05	4.65e-04	1.20e-04	2.49e-03	8.90e-04	5.18e-02

$$
\begin{aligned}
u^{3ds3} &= \cos(xyz) + \cos(\pi(x^2 + y^2 + z^2)) \\
u^{3ds4} &= \frac{1}{1 + x^2 + y^2 + z^2} \\
u^{3ds5} &= \sin(\pi(x^2 + y^2 + z^2)) + 1 \\
u^{3ds6} &= 10e^{-x^2-y^2-z^2} \\
u^{3ds7} &= \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \\
u^{3ds8} &= z\tanh((-\sin(x) + y^2)) \\
u^{3dns1} &= |x^2 + y^2 + z^2|^{0.8} \\
u^{3dns2} &= (xe^{1-|x|} - x)(ye^{1-|y|} - y)(ze^{1-|z|} - z).
\end{aligned}
$$

The rooted mean squared errors of approximate spline solutions against the exact solution are computed based on $501 \times 501 \times 501$ equally-spaced points of the bounding box of a domain shown in Figure 2.2 which fall into the domain.

We choose collocation points to create $M \times m$ matrix K, where m is the dimension of spline space $S_D^{-1}(\triangle)$ and apply Algorithm 1 to find the numerical solutions. We tested 10 functions over the domains in Figure 2.2. Their root mean square errors are presented in Table 2.8. We also

Table 2.9: The RMSE of spline solutions for the 3D Poisson equation over the two domains in Figure 2.2 based on trivariate spline functions of smoothness $r = 1$ and degree $D = 9$ for the AWL method and LL method.

	Torus				Human head			
	AWL		LL		AWL		LL	
Solution	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$
u^{3ds1}	3.55e-09	5.74e-07	1.79e-10	2.04e-09	2.83e-09	7.56e-07	5.83e-12	6.45e-11
u^{3ds2}	2.92e-08	1.98e-06	1.14e-08	8.50e-08	5.21e-07	2.72e-06	3.45e-10	2.95e-09
u^{3ds3}	1.07e-07	8.90e-06	5.34e-09	3.31e-08	6.44e-08	1.21e-05	7.26e-10	8.21e-09
u^{3ds4}	1.88e-08	1.46e-06	3.57e-09	2.29e-08	1.83e-08	2.72e-06	2.68e-10	2.76e-09
u^{3ds5}	8.25e-08	5.50e-06	1.33e-08	8.95e-08	6.09e-08	8.43e-06	9.75e-10	5.78e-09
u^{3ds6}	2.50e-07	1.80e-05	3.39e-08	1.90e-07	1.31e-07	1.35e-05	2.35e-09	2.47e-08
u^{3ds7}	8.07e-08	5.83e-06	1.01e-07	2.34e-06	1.88e-08	2.72e-06	4.19e-08	5.21e-07
u^{3ds8}	8.16e-09	7.24e-07	6.42e-10	4.32e-09	8.16e-09	3.41e-07	2.69e-11	1.66e-10
u^{3dns1}	3.92e-08	2.67e-06	5.07e-09	3.22e-08	3.63e-08	2.67e-06	3.82e-06	6.23e-04
u^{3dns2}	6.30e-04	2.29e-03	1.09e-04	1.58e-03	3.42e-04	2.49e-03	2.30e-04	4.84e-03

compare the AWL method with LL method for the numerical solution of the 3D Poisson equation. See numerical results in Table 2.9 which show that the LL method is more accurate than the AWL method when the solutions are smooth and is similar to the AWL method when the solutions are not very smooth.

2.5 Numerical Results for General Second Order Elliptic PDE

We shall present computational results for 2D and 3D general second order PDEs separately in the following two subsections. In each section, we first present the computational results from the spline based collocation method to demonstrate the accuracy the method can achieve. Then we present a comparison of our collocation method with the numerical method based on [LW18]. For convenience, we shall call our spline based collocation method the LL method and the numerical method in [LW18] the LW method.

2.5.1 Numerical examples for 2D general second order equations

We have used the same triangulations over various bounded domains as shown in Figure 1.1 and 2.1, and tested the same solutions which we used for the Poisson equation for the general second order equation to see the accuracy that the LL method can have. The root mean squared error(RMSE) $u - u_s$ and $\nabla u - \nabla u_s$ of approximate spline solutions $u_s, \nabla u_s$ against the exact solutions $u, \nabla u$ are given in Tables in this section. The RMSE is computed based on 1001×1001 equally-spaced points of the bounding box of a domain in Figure 1.1 and 2.1 which fell inside the domain. We chose additional collocation points to create $M \times m$ matrix $\mathcal{K}$, where m, M are the dimension of spline space $S_D^{-1}(\triangle)$ and $S_{D'}^{-1}(\triangle)$, respectively.

2D general second order equations with smooth coefficients

Example 1. *We first tested our computational method to solve the 2nd order elliptic equation with smooth PDE coefficients:* $a_{11} = x^2 + y^2, a_{12} = \cos(xy), a_{21} = e^{xy}, a_{22} = x^3 + y^2 - \sin(x^2 + y^2), b_1 = 3\cos(x)y^2, b_2 = e^{-x^2-y^2}, c = 0$. *Our testing functions are 2 non-smooth solutions* u^{ns1}, u^{ns2}, *and 8 smooth solutions* u^{s1} — u^{s8} *given in the previous section. The RMS of error vectors* $u - u_s$ *and* $\nabla(u - u_s)$ *over the four domains in Figure 1.1 and 2.1 is presented in Table 2.11. The numerical results show that the LL method works very well.*

Table 2.10 presents the numerical results for various mesh sizes and degrees. The table indicates that increasing the degree leads to a more accurate solution without refining the triangle. This approach is faster and can achieve the same level of accuracy.

Table 2.10: Comparison of the RMS of vectors $u - u_s, \nabla u - \nabla u_s$ for general elliptic equations with smooth coefficients in Example 1 and CPU time for different mesh sizes and degrees

	Mesh size=1/4				Mesh size=1/8			
	Degree 5	Time(s) 0.91	Degree 11	Time(s) 1.52	Degree 5	Time(s) 1.21	Degree 8	Time(s) 3.08
solution	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$
u^{s1}	2.55e-06	3.12e-05	1.26e-11	3.66e-10	1.05e-07	1.55e-06	5.55e-11	2.65e-10
u^{s2}	3.03e-03	2.49e-02	5.63e-09	4.79e-08	1.19e-04	1.24e-03	1.72e-08	3.87e-07
u^{s3}	4.52e-06	7.26e-05	1.11e-11	2.52e-10	2.04e-07	5.30e-06	3.05e-11	3.18e-10
u^{s4}	2.71e-03	2.50e-02	3.58e-09	5.58e-08	1.43e-04	1.27e-03	1.77e-08	4.03e-07
u^{s5}	3.21e-02	3.72e-01	4.25e-07	4.49e-06	2.51e-03	2.81e-02	5.00e-07	1.49e-05
u^{s6}	1.72e-05	1.74e-04	6.85e-11	8.03e-10	9.31e-07	1.03e-05	1.69e-10	2.09e-09
u^{s7}	7.17e-08	8.41e-07	1.86e-12	5.56e-11	2.50e-09	3.90e-08	7.50e-12	3.81e-11
u^{s8}	5.54e-01	5.72e+00	7.07e-02	1.44e+00	1.36e-01	2.29e+00	2.45e-02	5.24e-01
u^{ns1}	2.89e-05	6.47e-03	4.26e-06	1.33e-03	8.70e-06	2.22e-03	3.65e-06	8.28e-04
u^{ns2}	2.76e-04	3.33e-03	2.13e-05	3.85e-04	5.57e-05	1.16e-03	5.98e-06	2.56e-04

Table 2.11: RMSE of spline solutions for general second order elliptic equations with smooth coefficients over the four domains in Figure 1.1 and 2.1 when $r = 2$ and $D = 8$.

	Cracked domain		US		Star with 2 holes		Circle with 3 holes	
Solution	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$
u^{s1}	1.78e-09	6.84e-09	3.13e-10	3.14e-09	7.10e-11	1.15e-09	2.86e-10	3.60e-09
u^{s2}	3.03e-09	5.39e-08	2.14e-10	3.55e-09	5.13e-11	1.19e-09	2.52e-10	4.26e-09
u^{s3}	9.66e-10	4.05e-09	1.34e-10	1.72e-09	4.79e-11	8.98e-10	1.57e-10	1.96e-09
u^{s4}	2.32e-09	4.63e-08	3.57e-10	3.20e-09	6.37e-11	1.24e-09	4.35e-10	5.06e-09
u^{s5}	3.19e-07	5.59e-06	6.25e-09	2.08e-07	8.07e-10	3.41e-08	2.82e-09	9.36e-08
u^{s6}	9.64e-10	1.81e-08	7.10e-11	7.98e-10	5.09e-12	8.74e-11	1.43e-10	9.79e-10
u^{s7}	2.55e-10	8.38e-10	1.34e-11	2.45e-10	7.48e-12	1.33e-10	3.33e-11	4.04e-10
u^{s8}	8.93e-02	1.30e+00	8.74e-04	1.82e-02	2.10e-04	6.27e-03	2.10e-03	5.90e-02
u^{ns1}	4.63e-06	2.80e-04	6.57e-06	5.04e-04	1.68e-06	2.47e-04	5.79e-05	3.06e-03
u^{ns2}	2.14e-04	3.30e-03	4.88e-05	1.73e-03	1.12e-04	2.89e-03	6.23e-05	1.17e-03

2D general second order equations with non-smooth coefficients

Example 2. *In [SS13], the researchers experimented their numerical methods for the second order PDE as follows:*

$$\sum_{i,j=1}^{2} (1 + \delta_{ij}) \frac{x_i}{|x_i|} \frac{x_j}{|x_j|} u_{x_i x_j} = f \quad in\ \Omega, \quad u = 0\ on\ \partial\Omega,$$

Table 2.12: RMSE $u - u_s$ and $\nabla u - \nabla u_s$ for the general elliptic equation with the non-smooth coefficients in Example 14 over the four domains in Figure 1.1 and 2.1 when $r = 2$ and $D = 8$.

	Cracked domain		US		Star with 2 holes		Circle with 3 holes	
Solution	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$
u^{s1}	1.24e-11	3.98e-11	2.65e-10	4.14e-09	1.35e-12	8.90e-11	7.29e-12	3.57e-11
u^{s2}	2.61e-10	2.62e-09	3.61e-10	5.43e-09	3.36e-12	1.22e-10	8.62e-12	5.34e-11
u^{s3}	5.03e-12	1.91e-11	2.29e-10	2.39e-09	1.41e-12	6.60e-11	2.75e-12	1.59e-11
u^{s4}	2.90e-10	3.26e-09	5.44e-10	4.87e-09	2.68e-12	1.57e-10	1.17e-11	6.71e-11
u^{s5}	3.51e-08	5.30e-07	4.06e-09	7.29e-08	2.59e-10	6.56e-09	3.74e-10	7.53e-09
u^{s6}	4.96e-11	9.41e-10	6.26e-11	1.08e-09	2.40e-13	7.46e-12	1.64e-12	1.11e-11
u^{s7}	2.48e-12	5.95e-12	3.01e-11	3.96e-10	1.82e-13	1.16e-11	3.24e-13	2.20e-12
u^{s8}	2.24e-03	7.41e-02	5.85e-04	7.71e-03	5.28e-05	1.55e-03	1.25e-04	8.78e-03
u^{ns1}	1.83e-04	6.66e-04	2.47e-04	1.09e-03	2.20e-04	1.91e-03	2.61e-05	2.74e-04
u^{ns2}	3.64e-04	1.44e-03	3.20e-04	2.06e-03	3.04e-04	2.61e-03	8.46e-05	6.99e-04

where $\Omega = (-1, 1)^2$ *and the solution* u *is* $u(x, y) = (xe^{1-|x|} - x)(ye^{1-|y|} - y)$ *which is one of our testing functions. It is easy to see those coefficients satisfy the Cordes condition*

$$\frac{\sum_{i,j=1}^{d}(a_{i,j})^2}{(\sum_{i=1}^{2} a_{ii})^2} = \frac{2^2 + 1 + 1 + 2^2}{(2+2)^2} = \frac{10}{16} \leq \frac{1}{2 - 1 + \epsilon}$$

when $\epsilon = \frac{3}{5}$. *This equation was also numerically experimented in [LW18] and [WW19].*

Let us test our method on this 2nd-order elliptic equation with non-smooth coefficients for the 2 non-smooth solutions u^{ns1}, u^{ns2}, *and 8 smooth solutions* $u^{s1} - u^{s8}$ *over the four domains used in the previous section. We use bivariate splines of degree* $D = 8$ *and smoothness* $r = 2$ *for the experiment. And the RMSE of the solutions for the four domains in Figure 1.1 and 2.1 are reported in Table 2.12. It is clear to see that our method works very well.*

Example 3. *The second example in the paper [SS13] is another second-order PDE:*

$$\sum_{i,j=1}^{2}(\delta_{ij} + \frac{x_i x_j}{|x|^2})u_{x_i x_j} = f \ \ in\ \Omega, \quad u = 0\ on\ \partial\Omega,$$

where $\Omega = (-1, 1)^2$ *and the solution* u *is* $u(x, y) = |x^2 + y^2|^{\frac{\alpha}{2}}$ *which is on the list of our testing functions. Then those coefficients satisfy the Cordes condition when* $\epsilon = \frac{4}{5}$*. Similar to Example 14,*

Table 2.13: The RMS of vectors $u - u_s, \nabla u - \nabla u_s$ for general elliptic equations with non-smooth coefficients in Example 15 over the four domains when $r = 2$ and $D = 8$.

	Cracked domain		US		Star with 2 holes		Circle with 3 holes	
Solution	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$	$u - u_s$	$\nabla(u - u_s)$
u^{s1}	2.74e-11	5.99e-11	1.31e-10	2.23e-09	2.88e-12	8.28e-11	7.63e-12	3.81e-11
u^{s2}	2.55e-10	2.58e-09	1.34e-10	3.64e-09	5.23e-12	1.27e-10	1.23e-11	6.76e-11
u^{s3}	2.98e-11	6.61e-11	5.89e-11	1.12e-09	2.59e-12	8.28e-11	6.89e-12	3.21e-11
u^{s4}	2.61e-10	3.18e-09	1.76e-10	3.08e-09	4.00e-12	1.51e-10	1.24e-11	7.15e-11
u^{s5}	3.42e-08	5.26e-07	5.25e-09	7.25e-08	2.76e-10	6.23e-09	3.73e-10	7.44e-09
u^{s6}	5.13e-11	9.28e-10	1.97e-11	3.28e-10	3.30e-13	8.66e-12	2.08e-12	1.20e-11
u^{s7}	6.31e-12	1.38e-11	1.52e-11	2.98e-10	3.93e-13	1.08e-11	4.93e-13	2.40e-12
u^{s8}	2.08e-03	6.82e-02	2.45e-04	7.03e-03	3.68e-05	1.56e-03	1.72e-04	8.73e-03
u^{ns1}	1.56e-04	6.09e-04	2.75e-04	1.26e-03	8.13e-05	8.08e-04	2.76e-05	2.59e-04
u^{ns2}	2.76e-05	5.01e-04	6.59e-05	5.25e-04	1.70e-05	3.94e-04	6.63e-06	2.60e-04

we use the LL method to solve the PDE above using the 10 testing functions based on bivariate splines of degree $D = 8$ *and smoothness* $r = 2$*. See Table 2.13 for the RMS of error vectors.*

2.5.2 Comparison with Numerical Method in [LW18]

In this subsection, we compare our method (LL) with the method from the paper by LW [LW18] for solving three PDEs given in Examples 1, 14, and 15. We will present the Root Mean Squared Errors (RMSEs) from both methods in Table 2.14. For simplicity, we only show the numerical results from both methods for the Circle with 3 holes domain in Table 2.14. We obtained similar results for the other 2D domains in Figure 1.1. As seen in Table 2.14, our LL method provides more accurate results.

In Table 2.15, we show that the average computational time for our LL method is shorter than the LW method. This table includes the number of vertices, triangles, and the average time in seconds for solving 2D general second-order equations over the four domains in Figure 1.1 using both the LW and LL methods.

Table 2.14: The RMSE of spline solutions for general elliptic equations in Example 1, PDE with non-smooth coefficients in Example 14 and in Example 15 over the Circle with 3 holes when $r = 2$ and $D = 8$ for the LW method and the LL method, respectively.

	PDE in Example 1		PDE in Example 14		PDE in Example 15	
Method	LW	LL	LW	LL	LW	LL
u^{s1}	2.01e-09	1.49e-10	2.15e-06	1.03e-11	7.47e-09	6.64e-12
u^{s2}	2.22e-08	4.31e-11	2.97e-05	1.66e-11	3.86e-08	7.34e-12
u^{s3}	1.70e-09	8.85e-11	4.96e-06	5.20e-12	2.97e-09	4.03e-12
u^{s4}	2.29e-08	2.12e-10	6.13e-05	1.61e-11	7.66e-08	9.03e-12
u^{s5}	8.24e-08	3.37e-09	4.19e-04	7.58e-10	1.20e-06	5.95e-10
u^{s6}	2.63e-09	3.72e-11	4.11e-06	3.25e-12	3.15e-09	1.77e-12
u^{s7}	3.06e-14	1.05e-11	2.10e-11	1.22e-12	2.66e-14	4.61e-13
u^{s8}	8.50e-04	2.26e-03	2.54e-03	1.87e-04	1.78e-04	1.46e-04
u^{ns1}	1.35e-05	4.83e-05	3.57e-05	1.50e-05	1.52e-05	3.26e-05
u^{ns2}	2.45e-04	4.56e-05	1.60e-04	2.56e-04	5.92e-05	1.60e-05

Table 2.15: The number of vertices, triangles and the averaged time in seconds for solving 2D general second order equations over the four domains in Figure 1.1 by the LW and LL methods.

Domain	Number of vertices	Number of triangles	Average time for LW method	Average time for Part 2 of LL method
Flower with a hole	297	494	1.3236e+02	3.521e-01
Circle with 3 holes	525	895	4.4387e+03	8.313e-01

By considering the computational results in Table 2.15 and the computational times in Table 2.14, we conclude that our LL method is more effective and efficient than the LW method.

2.6 The Rate of Convergence of the LL method

Finally, we will discuss the convergence rate of the LL method. First, in Example 4, we perform an experiment to analyze the convergence rate using numerical solutions of 2D general elliptic PDEs in Example 1 over the domain $[0, 1]^2$. We display the convergence rate in relation to the size $h = |\triangle|$ of the triangulation $\triangle$ in Figure 2.5. Moreover, we demonstrate the convergence rate with respect to the Degrees of Freedom (DOF) in Figure 2.6.

Similarly, in Example 5, we first show the convergence rate based on the numerical solutions of 3D general elliptic PDE with smooth coefficients, with respect to the size h of the triangulations, in Figure 2.8. Then, we present the convergence rate in relation to the DOFs.

Example 4. *We carried out a numerical solution of the general elliptic equations in Example 1 with $D = 8, r = 2$ for testing functions u^{s2} and u^{s4} on various refinement levels to showcase the convergence behavior. The L^2, H^1 error vectors $u - u_s$ based on 1001^2 evenly spaced points over $[0, 1]^2$ concerning the size $h = |\triangle|$ are depicted in Figure 2.5. We observe that the convergence rate is approximately $O(h^7)$. According to Theorem 8, the $|u - u_s|_2 \leq Ch^2\epsilon$. This indicates that the numerical computation aligns with, and even surpasses, our theoretical expectations.*

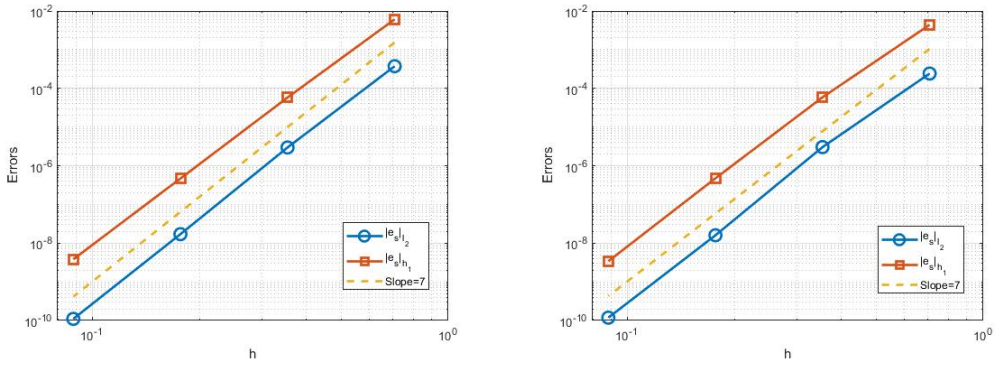


Figure 2.5: The RMSE in L^2 and H^1 norm of $u - u_s$ for testing functions u^{s2}(left) and u^{s4}(right) versus the size h of triangulation with $D = 8, r = 2$ where $e_s := u - u_s$

Also, Table 2.7 shows the RMSE for each degree $D = 5, 6, 7, 8$.

Subsequently, convergence results are displayed in Figure 2.6, based on the DOF (equal to the number of triangles $\times \frac{(D+1)(D+2)}{2}$). The RMSEs between the numerical and exact solutions are asymptotically proportional to $(DOF)^{-3.5}$. Thus, the asymptotic rate is $(DOF)^{-1/(d+1)}$, where $d = 2$. See the next example for $d = 3$.

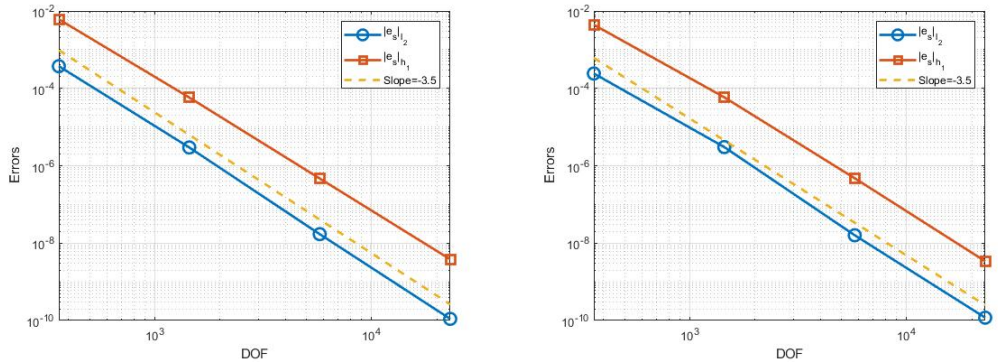


Figure 2.6: The RMSE in L^2 and H^1 norm of $u - u_s$ for testing functions u^{s2}(left) and u^{s4}(right) versus the DOFs with $D = 8, r = 2$ where $e_s := u - u_s$

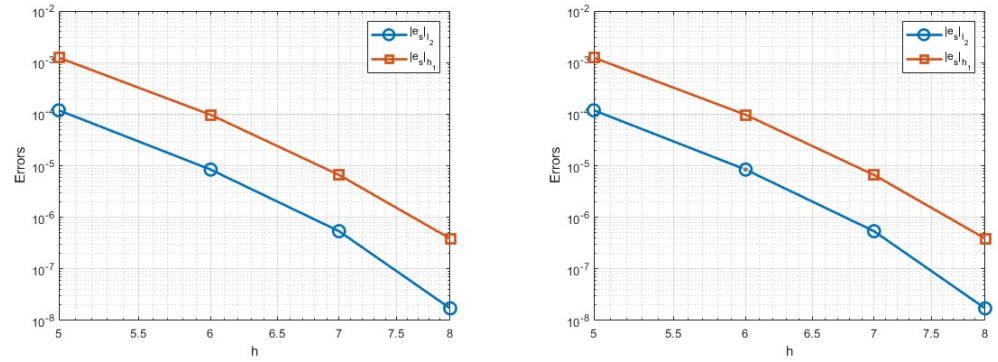


Figure 2.7: The RMSE in L^2 and H^1 norm of $u - u_s$ for testing functions u^{s2}(left) and u^{s4}(right) versus the degrees with $r = 2$ where $e_s := u - u_s$

Example 5. *We evaluated a 2nd-order elliptic equation (2.2) with smooth PDE coefficients* $a_{11} = x^2 + y^2, a^{22} = \cos(xy - z), a^{33} = \exp(\frac{1}{x^2+y^2+z^2+1}), a^{12} + a^{21} = x^2 - y^2 - z, a^{23} + a^{32} = \cos(xy - z)\sin(x - y), a^{13} + a^{31} = \frac{1}{y^2+z^2+1}, b_1 = 0, b_2 = -1, b_3 = \tan^{-1}(x^3 - y^2 + \cos(z)), c = x + y + z$. *Here,* $a^{12} = a^{21}, a^{32} = a^{23}$, *and* $a^{13} = a^{31}$. *The testing functions include the two smooth solutions* u^{3ds3} *and* u^{3ds5} *over the standard cube* $[0, 1]^3$. *The* L^2, H^1 *error vectors* $u - u_s$ *based on* 501^3 *evenly spaced points over* $[0, 1]^3$ *are presented in Figure 2.8. The errors between the numerical solution and exact solutions are asymptotically proportional to* $O(h^7)$. *We observe that the convergence rate is consistent with our theory for these smooth testing functions. Hence, we conclude that the LL methods perform exceptionally well.*

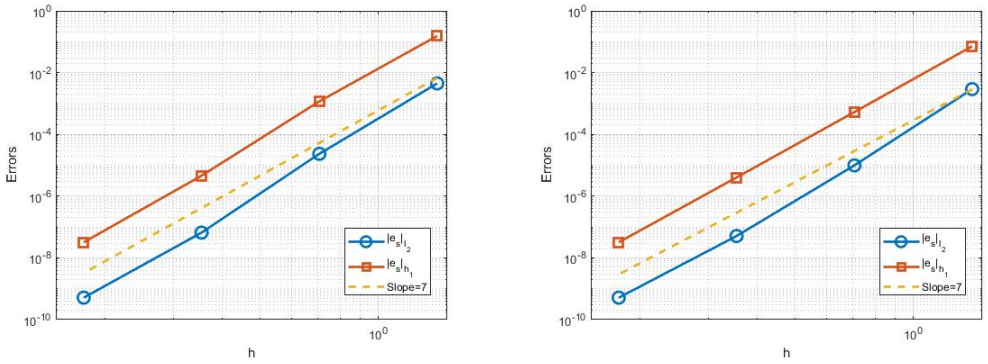


Figure 2.8: The RMSE in L^2 and H^1 norm of $u - u_s$ with $D = 9, r = 1$ for testing functions u^{3ds3}(left) and u^{3ds5}(right) versus the mesh size h

Additionally, we demonstrate the rate of convergence concerning the DOFs in Figure 2.9 based on the DOF (equal to the number of triangles $\times \frac{(D+1)(D+2)(D+3)}{6}$*).*

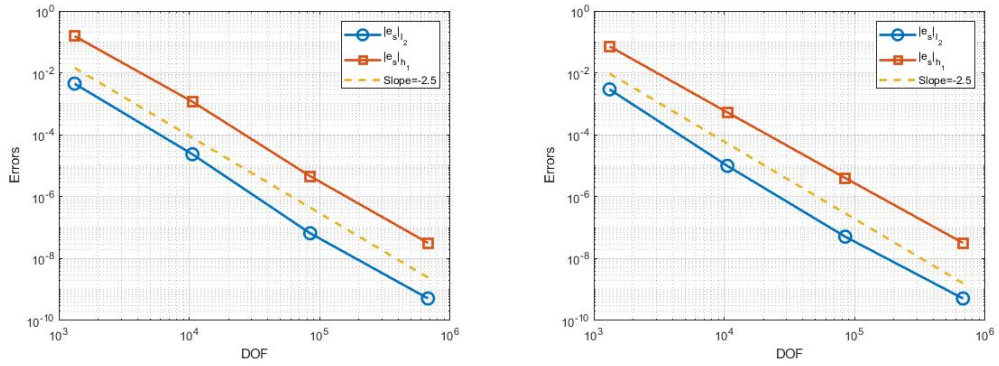


Figure 2.9: The RMSE in L^2 and H^1 norm of $u - u_s$ with $D = 9, r = 1$ for testing functions u^{3ds3}(left) and u^{3ds5}(right) versus the DOFs

Chapter 3

A Splined Based Collocation Method for Monge Amphére equations

We are interested in numerically solving the Monge-Ampére equation with Dirichlet boundary condition:

$$\det(D^2u(\mathbf{x})) = f(\mathbf{x}), \text{ in } \Omega \subset \mathbb{R}^d, d = 2, 3 \tag{3.1}$$

$$u(\mathbf{x}) = g(\mathbf{x}), \text{ on } \partial\Omega, \tag{3.2}$$

where $\mathbf{x}$ has d independent variables in a bounded domain $\Omega \subset \mathbb{R}^d$ and D^2u is the Hessian of the function u. More precisely, the Hessian of the function u, denoted as D^2u, can be expressed as

follows:

$$\det(D^2u) = u_{xx}u_{yy} - u_{xy}^2$$

for $d = 2$ and

$$\det(D^2u) = u_{xx}u_{yy}u_{zz} + 2u_{xy}u_{yz}u_{xz} - u_{xx}(u_{yz})^2 - u_{yy}(u_{xz})^2 - u_{zz}(u_{xy})^2, \tag{3.3}$$

for $d = 3$. This is the first step toward solving the fully nonlinear Monge-Ampére equation

$$\det(D^2u(\mathbf{x})) = \frac{f(\mathbf{x})}{g(\nabla u(\mathbf{x}))}, \quad \mathbf{x} \text{ in } \Omega \subset \mathbb{R}^d \tag{3.4}$$

$$\nabla u(\mathbf{x})|_{\partial\Omega} = \partial W, \tag{3.5}$$

where the boundary condition is called the oblique boundary condition. Such a partial differential equation arises from the optimal transportation problem (cf. e.g. [Eve98] and [Vil03]). More specifically, given a density function $f(\mathbf{x})$ on the domain Ω and another density function $g(\mathbf{w})$ on a separate domain W, the goal is to find the optimal plan T which transports f over Ω to g over W under the cost functional $c(\mathbf{x}, \mathbf{w}) = \frac{1}{2}\|\mathbf{x} - \mathbf{w}\|^2$, with $\int_\Omega f(\mathbf{x})d\mathbf{x} = \int_W g(\mathbf{w})d\mathbf{w}$. It is Y. Brenier who discovered a characterization of the optimal transportation problem.

Theorem 15. *(Brenier, 1988[BB00]) Suppose that the transport cost is the quadratic Euclidean distance, $c(x, y) = \frac{1}{2}\|x - y\|^2$ and suppose that W is a convex domain. Then there exists a convex function $u : \Omega \mapsto \mathbb{R}$ satisfying the Monge-Ampére equation (3.4), unique up to a constant, such that the gradient map $m = \nabla u$ is the unique optimal transport map satisfying the oblique boundary condition $\nabla u|_{\partial\Omega} = \partial W$.*

Although it is hard to determine the oblique boundary condition mentioned above, once we specify a map from the boundary of Ω to the boundary of W, the problem (3.4) becomes a Neumann boundary problem of the Monge-Ampére equation. In particular, if u is C^2 function whose gradient ∇u transforms Ω onto W, we can move the density $f(\mathbf{x})$ at $\mathbf{x} \in \Omega$ to the location $\nabla u(\mathbf{x}) \in W$ to become the density $g(\nabla u(\mathbf{x}))$. Such a problem is called the free movement problem which will be addressed at the end of this chapter.

Instead of considering the Neumann or oblique boundary value problem, this chapter will focus on the Monge-Ampére equation with a Dirichlet boundary condition. Note that this PDE has been studied for many years. In addition to the mathematical community, the Monge-Ampére equation has also been broadly studied in many applied fields such as elasticity, geometric optics, and image processing. See [Ber06] and [MY01]. Today such free-form optics are important in illumination applications. For example, they are used in the automotive industry for the construction of headlights that use the full light emitted by the lamp to illuminate the road but at the same time do not glare oncoming traffic [ZNC11]. There are multiple ways to solve this inverse reflector problem; brute-force approaches, methods of supporting ellipsoids, simultaneous multiple surfaces approach, and Monge-Ampére approaches. Also, the Monge-Ampére equation finds applications in finance, seismic wave propagation, geostrophic flows, in differential geometry as explained in [CGG18]. In this chapter, we shall explain a spline-based collocation method to solve the nonlinear PDE (3.1).

Let us begin recalling some existence, uniqueness, and regularity property of the Monge-Ampére equation (3.1). When f, g are sufficiently smooth, the solution of (3.1) is very smooth explained in the following

Theorem 16. *(Theorem 1 in [CNS84]) Suppose that a bounded domain $\Omega \in \mathbb{R}^n$ is strictly convex, where $n \geq 2$. For any strictly positive right-hand side $f \in C^\infty(\overline{\Omega})$ with the boundary condition g which has an extension $g \in C^\infty(\overline{\Omega})$, there exists a unique strictly convex solution u is in $C^\infty(\Omega)$ satisfying (3.1).*

There are several weaker versions of the existence results with regularity properties in the literature. For example,

Theorem 17. *(Figalli, 2017 [Fig17]) Let Ω be a uniformly convex domain, $k \geq 2, \alpha \in (0,1)$, and assume that $\partial\Omega$ is of class $C^{k+2,\alpha}$. Let $f \in C^{k,\alpha}(\bar{\Omega})$ with $f \geq c_0 > 0$. Then for any $g \in C^{k+2,\alpha}(\partial\Omega)$, there exists a unique solution $u \in C^{k+2,\alpha}(\bar{\Omega})$ to the Dirichlet problem* (3.1).

In [Awa13], Awanou introduced another weaker version of the existence theorem:

Theorem 18. *(Awanou, 2013[Awa13]) Let Ω be a uniformly convex domain in $\mathbb{R}^n$ with boundary in C^3. Suppose $g \in C^3(\bar{\Omega}), \inf f > 0$, and $f \in C^\alpha(\bar{\Omega})$ for some $\alpha \in (0,1)$. Then* (3.1) *has a convex solution u which satisfies the a priori estimate*

$$\|u\|_{C^{2,\alpha}(\bar{\Omega})} \leq C,$$

where C depends only on $n \geq 2, \alpha, \inf f, \Omega, \|f\|_{C^\alpha(\bar{\Omega})}$ and $\|g\|_{C^3}$.

In general, there are at least three different notions of solutions which have been studied in the literature besides the classic solution: one is called Aleksandrov solution, another one is viscosity solution, and the next one is Brainer's solution, according to the monograph by Villani, 2003, see page 129 in [Vil03]. The theory for the Monge-Ampére equation is deep (cf. [CC95], [Eve98],

[Vil03] and [Vil08]). In particular, the regularity of the solution has been extensively studied (cf. e.g. [Caf90], [Wan96], [CLW21]). In a landmark paper [Caf90], Cafferelli showed that the solution of the Monge-Ampére equation has an interior regularity over $\Omega' \subset \Omega$, i.e. $u \in H^{2,p}(\Omega')$ for any open set Ω' inside Ω. Furthermore, the solution has H^2 regularity over the entire domain, as established in [Wan96]:

Theorem 19. *(Wang, 1996[Wan96]) Let Ω be a strictly convex domain in $\mathbb{R}^n$. If $\partial\Omega$ and g in the equation (2) are C^3 smooth, and $f(x) \in C^{1,1}(\bar{\Omega})$, then the solution $u \in C^{2+\alpha}(\bar{\Omega})$.*

Due to these regularity results, we can use C^2 smooth multivariate splines to approximate the solution u under the conditions $f \in C^{1,1}(\bar{\Omega})$, $g \in C^3(\partial\Omega)$ and Ω being a strictly convex domain. In our computation, we are able to solve the Monge-Ampére equation over domains with uniform positive reach (cf. [GL20]) which include strictly convex domains as a special case. Additionally, we can use our method to experiment with the solution (3.1) even when f is not in $C^{1,1}(\bar{\Omega})$.

The numerical solution of the Monge-Ampére equation (MAE) is an active area of research, with many researchers developing different numerical methods and analyzing their theoretical convergence. As mentioned in [BFO10], the MAE poses several challenges for numerical solutions. The first challenge is that the equation is fully nonlinear, which means that geometric solutions or viscosity solutions must be used as weak solutions. The second challenge is the convex constraint, as the equation might not have a unique solution without it.

The popular finite element method is not directly applicable because of the involvement of the Hessian of the solution. This restricts the use of the Finite Element Method (FEM) or general Galerkin projection methods, discontinuous Galerkin method or continuous Galerkin method.

However, there are several remedy approaches based on the finite element method such as a mixed finite element method, vanishing moment method, etc. See [**A13**], [Awa14], [AL14] and [FN09].

Besides of finite element type methods, there are many finite difference methods, as seen in [BFO10], [Awa16], [Liu+16], [Ben+20], [LG21]. By simple calculation, we get

$$|\Delta u| = \sqrt{(\Delta u)^2} = \sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xx}u_{yy}} = \sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2 + 2f}$$

where $u_{xx}u_{yy} - u_{xy}^2 = f$ This leads to a semi-implicit scheme for solving the Monge-Ampére equation, used in [BFO10]. Similarly, for d dimensional space, we can get

$$(\Delta u)^d = d!f + P(\lambda_1, \ldots, \lambda_d) \tag{3.6}$$

where $P(\lambda)$ is a d-homogeneous polynomial. Moreover, many interesting approaches are based on the classic finite difference method as demonstrated in [BS19], [LX20], [Ben+20]. However, these methods have a weakness: they do not have analytic form of solution over the entire domain. In addition, we can find time marching methods in [Awa15], [Awa13], and least squares relaxation methods in [DG03], [DG04], [CGG18].

Let us be more precise on the numerical methods mentioned above. The least square notion of the solution was proposed and studied in [DG03], [DG04], and [CGG18]. Especially, this least square approach using a relaxation algorithm of the Gauss-Seidel-type iterations to decouple differential operators in [CGG18]. The approximation relies on mixed low order finite element methods with regularization techniques. Several 3D examples were demonstrated to show the performance of this method. In this chapter, we will compare the numerical results from our method to those to

in [CGG18] to show that our method produces more accurate results. These comparisons will be presented in the last section.

In [Awa15], a time marching approach is used to solve the Monge-Ampére equation. Given $\nu > 0$, the researcher considered the sequence of iterates

$$-\nu\Delta u_{k+1} = -\nu\Delta u_k + \det D^2 u_k - f,\ u_{k+1} = g \ \text{ on } \partial\Omega. \tag{3.7}$$

He used the discrete version of Newton's method in the vanishing moment methodology. And he showed the convergence of the iterative method for solving the nonlinear system. We shall also compare his numerical results with our results in the last section to show that our proposed method is also more accurate.

In [Ben+20], the researchers introduced the meshless Generalized Finite Difference Method (GFDM) in both 2D and 3D settings. They tested several examples using the Cascadic iterative algorithm over convex and non-convex domains. We will compare our proposed method with the results from the Cascadic iterative algorithm in the last section to demonstrate that our method is also better.

We now present our numerical method for solving (3.1) using multivariate spline functions over a tetrahedralization of Ω. We choose to use multivariate splines for several reasons. Firstly, multivariate splines with smoothness $r \geq 2$ can accurately approximate the solution u of the Monge-Ampére equation over any convex polyhedral domain. Additionally, since spline functions have C^2 smoothness, we can calculate the Hessian of the solution. This allows us to use the collocation method instead of the weak formulations presented in previous works, such as [**A13**] and [Awa14].

To solve the Monge-Ampére equation numerically, many researchers have adopted an iterative algorithm known as the fixed point algorithm, which was introduced in [BFO10]. The fixed point algorithm is given by:

$$\Delta u_{k+1} = ((\Delta u_k)^d + a(f - \det D^2 u_k))^{\frac{1}{d}} \tag{3.8}$$

along with the prescribed Dirichlet boundary conditions with $a = 2$ and $d = 2$. The researchers in [BFO10] explained that this is a fixed point method as the true solution u satisfies (3.8) trivially.

In [Awa15], this iterative algorithm is generalized to the 3D setting with $d \geq 3$ for various $a > 0$. In particular, the researcher in [Awa15] explained that the iteration (3.8) is well-defined for $a \leq d^d$ as $\det(D^2 u_k) \leq \frac{1}{d^2}(\Delta u)^d$. Numerical results in [Awa15] are demonstrated in the framework of the spline element method with $a = 2$ for the 2D case and $a = 9$ for the 3D case.

In this chapter, we shall use the following iterative method:

$$\Delta u_{k+1} = ((\Delta u_k)^d + d^d(f - \det D^2 u_k))^{\frac{1}{d}} \tag{3.9}$$

to handle the nonlinearity of the Monge-Ampére equation where $d = 3$.

However, another requirement of the solution of the Monge-Ampére equation is that u must be convex in order for the equation to be elliptic. Without this constraint, the equation does not have a unique solution. (For example, taking boundary data $g = 0$, if u is a solution, then $-u$ is also a solution in $\mathbb{R}^d$.) Many numerical methods mentioned above failed to enforce this convexity constraint. The convexity of u is equivalent to the positive definiteness of the Hessian matrix $D^2 u$. In terms of the eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ of $D^2 u$, we will ensure that three eigenvalues

$\lambda_1(k) \geq \lambda_2(k) \geq \lambda_3(k)$ of the kth iteration u_k in a spline space satisfy $\lambda_1(k) + \lambda_2(k) + \lambda_3(k) \geq 0$ as well as $\lambda_1(k)\lambda_2(k)\lambda_3(k) > 0$, although they are not enough to ensure the convexity of kth spline solution u_k.

In Section 3.1, we introduce the spline collocation method for the Monge-Ampére equation and its average algorithm, and establish three different versions of convergence results. Finally, in the last section 3.2, we present numerical results for several 3D examples of smooth and convex solutions, as well as nonsmooth convex solution over convex and nonconvex bounded domains to demonstrate the effectiveness of our proposed method. We compare our results with those of several existing numerical methods to show the accuracy and efficiency of our method. Finally, we shall present some examples of free movement in 2D and 3D settings to show how the density from one place is moved to another place. This will demonstrate further that our proposed method is versatile enough.

3.1 Our Proposed Algorithms and Their Convergence Analysis

3.1.1 A Spline-Based Collocation Method for 3D Monge-Ampère Equation

In the previous chapter, we showed a method for solving the Poisson equation. In this chapter, we will extend this method to solve the Monge-Ampére equation (Monge-Ampére equation) in 2D and 3D using a similar collocation approach.

To commence, we will employ the same set of domain points as collocation points to construct a spline function that satisfies the Monge-Ampére equation at these points. However, we will need to make adjustments to the matrix K utilized in the previous chapter and introduce additional

constraints on the coefficient vector to account for the differences in the equations. The Monge-Ampére equation will be solved through iterative methods, including the semi-implicit method(SI method) in (3.6), the time marching method(TM method) in (3.7), and the fixed-point algorithm(FP method) in (3.8). Table 3.1 presents the numerical results of the exact solution $u = e^{\frac{x^2+y^2}{2}}$ for these methods with the number of vertices $N_v = 81$ and the number of triangles $N_T = 128$. Based on the results in Table 3.1, the FP method appears to be the most efficient and accurate method for solving the 2D Monge-Ampére equation. Similarly, FP method is faster and more accurate for the 3D Monge-Ampére equation. Therefore, we propose to solve the Monge-Ampére equation using

Table 3.1: Comparison of computational efficiency and accuracy of different numerical methods with the number of vertices $N_v = 81$ and the number of triangles $N_T = 128$, the exact solution $u = e^{\frac{x^2+y^2}{2}}$ for solving the 2D Monge-Ampére equation

Method	CPU time (s)	l_2 norm	Number of iterations
FP method	0.491	2.63e-10	32
TM method	1.806	8.28e-10	53
SI method	2.496	3.52e-10	37

the FP method. This method was first introduced in [Awa15] and is an iterative algorithm given as follows:

$$\Delta u_{k+1} = \sqrt[d]{(\Delta u_k)^d + a(f - \det(D^2 u_k))}, \quad k = 0, 1, \cdots, . \tag{3.10}$$

In this chapter, we employ this method to achieve accurate and efficient solutions to the 3D Monge-Ampére equation. We begin by solving for an initial u_0 using the equation:

$$\Delta u_0 = \sqrt[3]{27 f}$$

together with the given boundary conditions.

It is important to note that the choice of initial u_0 is based on an assumption regarding the eigenvalues of $\det(D^2u)$. Specifically, if these eigenvalues are close to each other, then a good initial guess is $\Delta u = 3\sqrt[3]{f}$. However, if the eigenvalues are quite different, this may not be a good choice. We will explain our approach to addressing this issue later in this section.

Additionally, we have experimented with different values of the parameter a in (3.10), and our tests have shown that using $a = 27$ leads to more accurate results. Hence, we will use $a = 27$ throughout the chapter. This can be seen in Figure 3.1, where we plot the $\log(|u - u_s^{(k)}|_\infty)$ for different a values in the case of u^{3d3} and u^{3d5}.

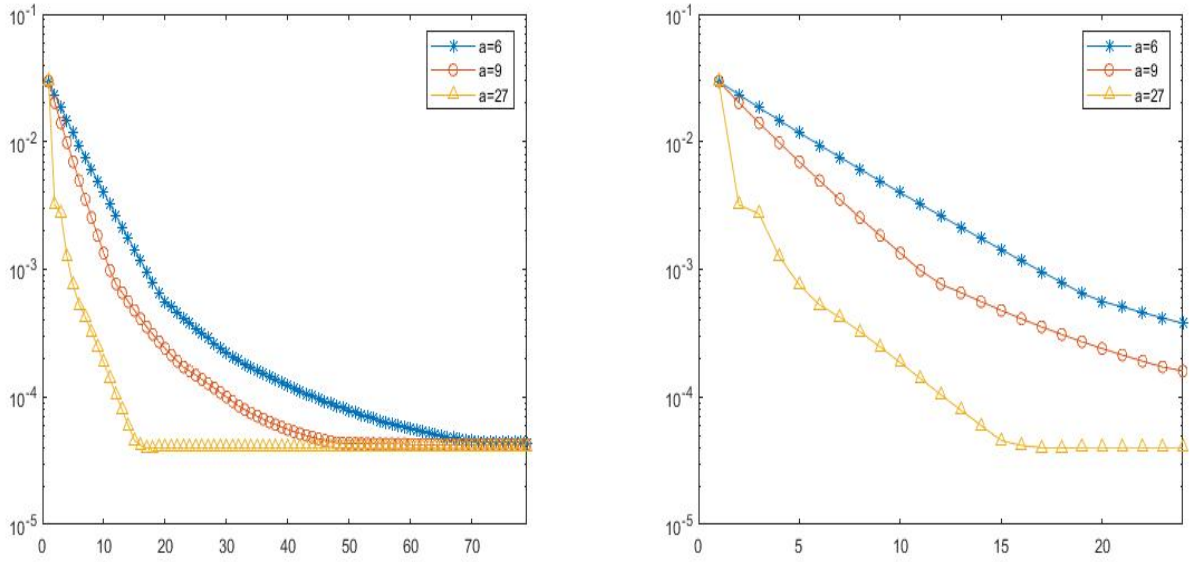


Figure 3.1: $\log(\|u - u_s^{(k)}\|_\infty)$($y$-axis) for u^{3d3}(left) and u^{3d5}(right) for each iteration(x-axis) with different $a = 6, 9, 27$

To explain our numerical method, we will use the Dirichlet boundary condition $u|_{\partial\Omega} = g$ for simplicity.

3.1.2 Two Computational Algorithms

We will present two computational algorithms for solving the Monge-Ampére equation. The first one is a standard approach that has been widely used in the literature based on finite differences and finite element discretizations. However, we will use multivariate spline functions to discretize the function space $H^2(\Omega)$ to demonstrate the efficiency and effectiveness of our approach and compare our numerical results with those of other methods in the literature. The details of our computational results can be found in the next section.

Algorithm 2: An Iterative Poisson Equation Algorithm

Start with an initial solution u_0 by solving the following Poisson equation using our collocation method discussed in the previous section:

$$\Delta u_0 = \sqrt[3]{27f} \tag{3.11}$$

and $u_0 = g$ on the boundary $\partial\Omega$. We then iteratively solve the Poisson equation

$$\Delta u_{k+1} = \sqrt[3]{(\Delta u_k)^3 + 27(f - \det(D^2 u_k))}, \quad k = 0, 1, \cdots. \tag{3.12}$$

That is, we find $u_{k+1} \in S_D^r(\triangle)$ satisfying the following equations approximately:

$$\begin{cases} \Delta u_{k+1}(\xi_i) & = \sqrt[3]{(\Delta u_k(\xi_i))^3 + 27(f(\xi_i) - det(D^2 u_k(\xi_i)))} \quad \xi_i \in \Omega \subset \mathbb{R}^3, \\ u_{k+1}(\xi_i) & = g(\xi_i), \quad \xi_i \in \partial\Omega \end{cases} \tag{3.13}$$

Then we use the iterative algorithm 1 in chapter 2.
Terminate the iteration when $\|f - \det(D^2 u_{k+1})\|_{l_\infty} > \|f - \det(D^2 u_k)\|_{l_\infty}$.

Next we explain an averaged iterative algorithm. Assume Ω is bounded and the closure of Ω is of uniformly positive reach as explained in the previous section. For any $f \in L^2(\Omega)$, the solution of the Poisson equation with zero boundary condition is in $H^2(\Omega)$ by Theorem 1. Furthermore, the solution of the Poisson equation with boundary condition g is in $H^2(\Omega)$ if $g \in H^{1/2}(\partial\Omega)$. Indeed, we consider a function $v \in H^2(\Omega)$ whose trace on $\partial\Omega$ is $g \in H^{1/2}(\partial\Omega)$. Define $w = u - v$ and we

have

$$\int_\Omega \nabla w \cdot \nabla \phi = \int_\Omega \nabla u \cdot \nabla \phi - \int_\Omega \nabla v \cdot \nabla \phi$$
$$= \int_\Omega -f \cdot \phi - \int_\Omega \Delta v \cdot \phi = \int_\Omega (\Delta v - f) \cdot \phi.$$

for every $\phi \in H_0^1(\Omega)$. Then the solution w satisfies the weak formulation of

$$\Delta w = f - \Delta v \text{ in } \Omega, w = 0 \text{ on } \partial\Omega$$

is in $H^2(\Omega)$ (cf. [GL20]). Therefore, $u = w + v$ is in $H^2(\Omega)$.

Let T be an operator which maps $H^2(\Omega) \to H^2(\Omega)$ in the following sense: for any $v \in H^2(\Omega)$, let $u = T(v)$ be the solution of the Poisson equation:

$$\Delta u = \sqrt[3]{(\Delta v)^3 + 27(f - \det(D^2 v))} \text{ over } \Omega$$

and $u|_{\partial\Omega} = g$ with $g \in H^{1/2}(\partial\Omega)$. In other words, the operator T on $H^2(\Omega)$ is defined by

$$T(u) = \Delta^{-1}[\sqrt[3]{(\Delta u)^3 + 27(f - \det(D^2 u))}].$$

It is easy to see that T is a nonlinear operator T maps $H^2(\Omega)$ to $H^2(\Omega)$. Also, we can see that the exact solution u^* satisfying $\det(D^2 u^*) = f$ is a fixed point of T.

Now we are ready to define an averaged iterative algorithm. In this way, we can find more accurate solutions than the one using Algorithm 2 only.

Algorithm 3: The Averaged Iterative Algorithm

Start with an initial u_0, where $\Delta u_0 = \sqrt[3]{27f}$ over Ω and $u_0 = g$ on $\partial\Omega$.
We iteratively solve the Poisson equations

$$\Delta u_{k+\frac{1}{2}} = \sqrt[3]{(\Delta u_k)^3 + 27(f - \det(D^2 u_k))}, \tag{3.14}$$

together with the boundary condition $u_{k+\frac{1}{2}} = g$ on $\partial\Omega$ by using the minimization in (**??**) and then take

$$u_{k+1} = \frac{1}{2}u_{k+\frac{1}{2}} + \frac{1}{2}u_k. \tag{3.15}$$

Stop the iteration if $\|f - \det(D^2 u_{k+1})\|_{l_\infty} > \|f - \det(D^2 u_k)\|_{l_\infty}$.

Let us present some performance of these two algorithms to show that Algorithm 3 indeed very useful. Consider a testing function u^{3ds1} as in Section 3.2, the eigenvalues of the Hessian matrix $D^2 u^{3d1}$ are $1, 5, 15$. Although these three eigenvalues are not close to any real positive number, we use various positive numbers p for the right-hand side of the Poisson equation $\Delta u_0 = p$ to solve u_0 as an initial solution and then apply Algorithm 2 and Algorithm 3. In Table 3.2, the results from both Algorithms are shown after the same number of iterations. We can see that Algorithm 3 produces more accurate solution than Algorithm 2 from various initial values except for p which is close to $21 = \Delta u^{3ds1}$, i.e. $p \in [17.7, 26]$. Also, the ℓ_2 and h_1 errors from Algorithm 3 are better than the errors from Algorithm 1 for testing functions u^{3ds3} and u^{3ds8} as shown in Figure 3.2.

Table 3.2: Errors of numerical solutions u^{3ds1} for the Monge Ampère equation over $[0,1]^3$ with $D = 9$, $r = 1$ over the same tetrahedralization for various initial values p by two algorithms

$\Delta u_0 =$ p	Algorithm 1		Algorithm 2	
	$\|e_s\|_{l_2}$	$\|e_s\|_{h_1}$	$\|e_s\|_{l_2}$	$\|e_s\|_{h_1}$
12.6	2.1291e-02	1.6315e-01	1.9230e-02	1.3670e-01
15.1	3.4135e-03	5.8902e-02	5.0004e-03	5.1503e-02
16.4	2.0124e-03	3.1978e-02	1.2081e-08	5.4356e-07
17.1	7.9130e-04	1.4517e-02	4.1401e-08	1.6448e-06
17.7	1.3980e-09	3.9929e-08	3.6103e-08	2.7446e-06
26.0	6.4074e-09	2.4003e-07	4.8324e-07	1.8097e-05
26.5	2.0899e-04	6.3176e-03	5.0149e-07	1.8781e-05
27.0	5.6423e-04	1.7172e-02	3.9592e-04	1.2049e-02
27.5	8.9037e-04	2.5381e-02	7.0701e-04	2.0696e-02

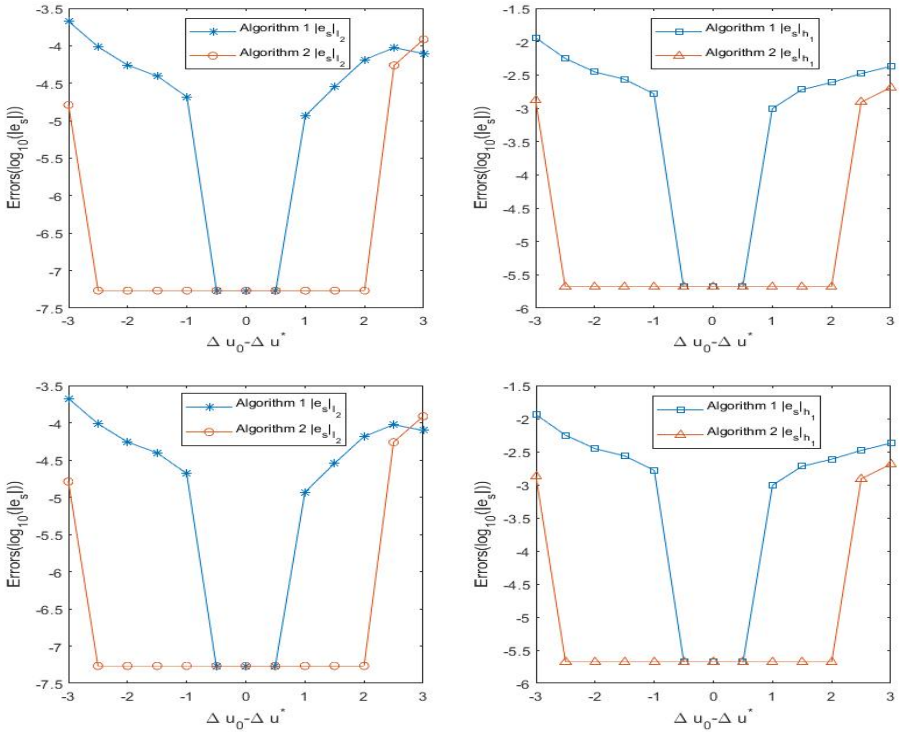


Figure 3.2: Errors $\log(|e_s|_{l_2}), \log(|e_s|_{h_1})$ for u^{3ds3}(Top) and u^{3ds8}(Bottom)

3.1.3 Convergence Analysis

According to [Awa14], it is known that if $\det(D^2u^*) = f > 0$ and u^* is convex, then there exist constants $m, M > 0$, independent of the mesh size $|\triangle|$ such that

$$0 < m \leq \lambda_3 \leq \lambda_2 \leq \lambda_1 \leq M,$$

where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of $(D^2u(x)), \forall x \in \Omega$. The following result is also known (cf. [Awa15]). For clarity, we provide proof below.

Lemma 6. *Suppose that the convex solution $u^* \in W^{2,\infty}$ satisfies $det(D^2u^*) = f > 0$. There exists a $\delta > 0$ such that for any u which is close enough to the exact solution u^* in the sense that $|u - u^*|_{2,\infty} \leq \delta$, we have*

$$det(D^2u) \leq \frac{1}{27}(\Delta u)^3 < \frac{1}{a}(\Delta u)^3$$

for any $a < 27$.

Proof. Recall that the eigenvalues of a symmetric matrix are continuous functions of its entries, as roots of the characteristic equation (cf. Ostrowski (1960) Appendix K [Ost60]). Thus, for a given $u^* \in W^{2,\infty}(\Omega)$, there exists $\delta > 0$ such that for $u \in W^{2,\infty}(\Omega), |u - u^*|_{2,\infty} \leq \delta$ implies $M \geq \lambda_1(D^2u(x)) \geq \lambda_2(D^2u(x)) \geq \lambda_3(D^2u(x)) > 0$. Now, we use the property that $\det(D^2u)$ is

the multiplication of all eigenvalues to have

$$\begin{aligned}
&\det(D^2u) = \lambda_1\lambda_2\lambda_3 = \frac{1}{27}(3\lambda_1\lambda_2\lambda_3 + 6\lambda_1\lambda_2\lambda_3 + 18\lambda_1\lambda_2\lambda_3) \\
&\leq \frac{1}{27}(\lambda_1^3 + \lambda_2^3 + \lambda_3^3 + 6\lambda_1\lambda_2\lambda_3 + 3\lambda_3(\lambda_1^2 + \lambda_2^2) + 3\lambda_1(\lambda_2^2 + \lambda_3^2) + 3\lambda_2(\lambda_1^2 + \lambda_3^2)) \\
&= \frac{1}{27}(\lambda_1 + \lambda_2 + \lambda_3)^3 = \frac{1}{27}(\Delta u)^3.
\end{aligned}$$

This completes all the proof. □ □

We first consider the point-wise convergence of the sequence from Algorithm 2.

Theorem 20. *Fix a spline space $\mathcal{S}_D^r(\triangle)$ with $\triangle$ being a tetrahedralization of the domain Ω. Let $u_k \in \mathcal{S}_D^r(\triangle), k \geq 1$ be the sequence from Algorithm 1. Then, any average values of $f - \det(D^2u_k), k \geq 1$ are nonnegative in the following senses:*

$$\frac{1}{n+1}\sum_{k=0}^{n}(f(\boldsymbol{x}) - \det(D^2u_k)(\boldsymbol{x})) \geq 0, \quad \boldsymbol{x} \in \Omega \tag{3.16}$$

for all $n \geq 1$. Furthermore, suppose that there exists a bound $M > 0$ such that $|u_k(\boldsymbol{x})| \leq M$ over Ω for all $k \geq 0$. Then

$$\frac{1}{n+1}\sum_{k=0}^{n}(f(\boldsymbol{x}) - \det(D^2u_k)(\boldsymbol{x})) \to 0 \tag{3.17}$$

when $n \to \infty$ for all $\boldsymbol{x} \in \Omega$.

We remark that the condition that $|u_k(\mathbf{x})| \leq M$ above is a computational condition one can check during the iterative computation of Algorithm 1. Our numerical experiments show that for

some testing functions u, this condition does satisfy while for other testing functions, the condition does not satisfy. See Figure 3.3 for these numerical phenomena.

Proof. By (3.13) and Lemma 6, we get

$$\begin{aligned}
27\det(D^2u_{k+1})) \le (\Delta u_{k+1})^3 &= (\Delta u_k)^3 + 27(f - \det(D^2u_k)) \\
&= (\Delta u_{k-1})^3 + 27(f - \det(D^2u_{k-1})) + 27(f - \det(D^2u_k)) \\
&= (\Delta u_{k-1})^3 + 2 \cdot 27f - 27\det(D^2u_{k-1}) - 27\det(D^2u_k) \\
&= \cdots \\
&= (\Delta u_0)^3 + 27(k+1)f - 27\sum_{j=0}^{k} \det(D^2u_j) \\
&= 27f + 27(k+1)f - 27\sum_{j=0}^{k} \det(D^2u_j).
\end{aligned}$$

Hence, we have

$$0 \le 27f + 27(k+1)f - 27\sum_{j=0}^{k} \det(D^2u_j) - 27\det(D^2u_{k+1})) = 27\sum_{j=0}^{k+1}(f - \det(D^2u_j)).$$

which leads to (3.16). In addition, we also have

$$(\Delta u_{k+1})^3 - (\Delta u_0)^3 = 27\sum_{j=0}^{k}(f - \det(D^2u_j)).$$

By the assumption of this theorem, u_{k+1} has a bound, i.e. $\|u_{k+1}\|_{\infty,\Omega} \leq M$. Then we can use the Markov inequality to have

$$\|\Delta u_{k+1}\|_{\infty,\Omega} \leq \frac{C}{|\triangle|^2}\|u_{k+1}\|_{\infty,\Omega} \leq \frac{CM}{|\triangle|^2} < \infty \tag{3.18}$$

for a constant $C > 0$ independent of u_{k+1}. It thus follows

$$\frac{27\sum_{j=0}^{k}(f - \det(D^2u_j))}{k+1} = \frac{(\Delta u_{k+1})^3 - (\Delta u_0)^3}{k+1} \to 0.$$

Therefore, we finished a proof of Theorem 20. □ □

Furthermore, we denote $w(u, f) := \sqrt[3]{(\Delta u)^3 + 27(f - \det(D^2u))}$. We have

$$\begin{aligned}\|\Delta u_{k+1} - \Delta u\|_{L^2(\Omega)} &= \|\frac{(\Delta u_k)^3 + 27(f - \det(D^2u_k)) - (\Delta u)^3}{(w(u_k, f))^2 + w(u_k, f)w(u, f) + (w(u, f))^2}\|_{L^2(\Omega)} \\ &= \|\frac{(\Delta u_k)^3 - (\Delta u)^3 + 27(\det(D^2u) - \det(D^2u_k))}{(w(u_k, f))^2 + w(u_k, f)w(u, f) + (w(u, f))^2}\|_{L^2(\Omega)}\end{aligned}$$

By simple calculations, we get

$$(\Delta u_k)^3 - (\Delta u)^3 = (\Delta u_k - \Delta u)((\Delta u_k)^2 + \Delta u_k \cdot \Delta u + (\Delta u)^2)$$

and by Lemmas 2.1, 2.2, and 2.3 in [Awa14]

$$\det(D^2u) - \det(D^2u_k) = \mathrm{cof}((1-t)D^2u_k + tD^2u) : (D^2u_k - D^2u)$$

for some $t \in [0,1]$. By simple calculation and Lemma 7, we have

$$\begin{aligned}
&\|\frac{(\Delta u_k)^3 - (\Delta u)^3}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^2(\Omega)} \\
&\leq \|\frac{(\Delta u_k - \Delta u)((\Delta u_k)^2 + \Delta u_k \cdot \Delta u + (\Delta u)^2)}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^2(\Omega)} \\
&\leq \|\frac{(\Delta u_k)^2 + \Delta u_k \cdot \Delta u + (\Delta u)^2}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^\infty(\Omega)} \|\Delta u_k - \Delta u\|_{L^2(\Omega)}.
\end{aligned}$$

Let $M = (M_{ij}), N = (N_{ij})$ be matrix fields and c be a real number. Then we get

$$\begin{aligned}
\frac{M : N}{c} &= \frac{1}{c}\sum_{i,j=1}^{3} M_{ij}N_{ij} = \sum_{i,j=1}^{3} \frac{M_{ij}}{c}N_{ij} \\
&\leq \|\frac{M}{c}\|_\infty \sum_{i,j=1}^{3} |N_{ij}| \leq \|\frac{M}{c}\|_\infty \Big(\sum_{i,j=1}^{3} N_{ij}^2 \Big)^{1/2} \cdot 3,
\end{aligned} \tag{3.19}$$

where $\|\frac{M}{c}\|_\infty = \max_{1\leq i\leq 3} \sum_{j=1}^{3} |\frac{M_{ij}}{c}|$.

By (3.19) with , we have

$$\begin{aligned}
&\|\frac{27(\det(D^2u) - \det(D^2u_k))}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^2(\Omega)} \\
&\leq 27\|\frac{(\mathrm{cof}((1-t)D^2u_k + tD^2u) : (D^2u_k - D^2u))}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^2(\Omega)} \\
&= 27\Big[\int \Big(\frac{(\mathrm{cof}((1-t)D^2u_k + tD^2u) : (D^2u_k - D^2u))}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2} \Big)^2 \Big]^{\frac{1}{2}} \\
&= 81\|\frac{\mathrm{cof}((1-t)D^2u_k + tD^2u)}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_\infty |u_k - u|_{H^2(\Omega)} \\
&\leq 81\|\frac{(1-t)D^2u_k + tD^2u}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_\infty^2 |u_k - u|_{H^2(\Omega)} \\
&\leq 81\|\frac{(1-t)D^2u_k + tD^2u}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_\infty^2 \|u_k - u\|_{H^2(\Omega)}
\end{aligned}$$

for some $t \in [0,1]$. By these two equations, we can have

$$\begin{aligned}
\|\Delta u_{k+1} - \Delta u\|_{L^2(\Omega)} &= \|\frac{(\Delta u_k)^3 - (\Delta u)^3 + 27(\det(D^2u) - \det(D^2u_k))}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^2(\Omega)} \\
&\leq \|\frac{(\Delta u_k - \Delta u)((\Delta u_k)^2 + \Delta u_k \cdot \Delta u + (\Delta u)^2)}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^2(\Omega)} \\
&+ \|\frac{27(\det(D^2u) - \det(D^2u_k))}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^2(\Omega)} \\
&\leq \|\frac{(\Delta u_k)^2 + \Delta u_k \cdot \Delta u + (\Delta u)^2}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^\infty(\Omega)}\|\Delta u_k - \Delta u\|_{L^2(\Omega)} \\
&+ 81\|\frac{(1-t)D^2u_k + tD^2u}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_\infty^2\|u_k - u\|_{H^2(\Omega)}
\end{aligned}$$

for some $t \in [0,1]$. Now, we need the following lemma from [LL22] to prove one of the main convergence results In this chapter.

Lemma 7. *Suppose that Ω is bounded and has uniformly positive reach $r_\Omega > 0$. Then there exist two positive constants A and B such that*

$$A\|u\|_{H^2(\Omega)} \leq \|\Delta u\|_{L^2(\Omega)} \leq B\|u\|_{H^2(\Omega)}, \quad \forall u \in H^2(\Omega) \cap H_0^1(\Omega). \tag{3.20}$$

By Lemma 7, we have

$$\begin{aligned}
\|\Delta u_{k+1} - \Delta u\|_{L^2(\Omega)} &\leq \|\frac{(\Delta u_k)^2 + \Delta u_k \cdot \Delta u + (\Delta u)^2}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_{L^\infty(\Omega)}\|\Delta u_k - \Delta u\|_{L^2(\Omega)} \\
&+ 81\|\frac{(1-t)D^2u_k + tD^2u}{(w(u_k,f))^2 + w(u_k,f)w(u,f) + (w(u,f))^2}\|_\infty^2 \frac{1}{A}\|\Delta u_k - \Delta u\|_{L^2(\Omega)}
\end{aligned}$$

and therefore

$$\|\Delta u_{k+1} - \Delta u\|_{L^2(\Omega)} \leq \rho_k \|\Delta u_k - \Delta u\|_{L^2(\Omega)},$$

where

$$\begin{aligned} \rho_k := \; & \| \frac{(\Delta u_k)^2 + \Delta u_k \cdot \Delta u + (\Delta u)^2}{(w(u_k, f))^2 + w(u_k, f) w(u, f) + (w(u, f))^2} \|_{L^\infty(\Omega)} + \\ & \frac{81}{A} \| \frac{(1-t) D^2 u_k + t D^2 u}{(w(u_k, f))^2 + w(u_k, f) w(u, f) + (w(u, f))^2} \|_\infty^2. \end{aligned} \tag{3.21}$$

We are now ready to conclude the following result

Theorem 21. *Suppose that* Ω *is bounded and has uniformly positive reach. If* $\rho_k \leq \gamma < 1$ *for all* $k \geq 1$*, then the sequence* $\{u_k\}$ *from Algorithm 2 converges.*

Note that our numerical experiments show that for some testing function u, we have indeed $\rho_k < 1$ while there is other testing function u which gives $\rho_k > 1$. See Figures 3.3 and 3.4. Also, it is hard to estimate ρ_k from the formula (3.21).

In Figures 3.3 and 3.4, we plot ρ_k corresponding to the numerical solution for smooth solutions s_1, s_2, s_3, s_4 and non-smooth solutions ns_1, ns_2. They are defined as follows.

- s_1: polynomial function $(x^2 + 5y^2 + 15z^2)/2$;
- s_2: exponential function $\exp((x^2 + y^2 + z^2)/2)$;
- s_3: radical function $-\sqrt{6 - (x^2 + y^2 + z^2)}$;
- s_4: $(x^2 + y^2 + z^2)/2 - \sin(x) - \sin(y) - \sin(z)$;

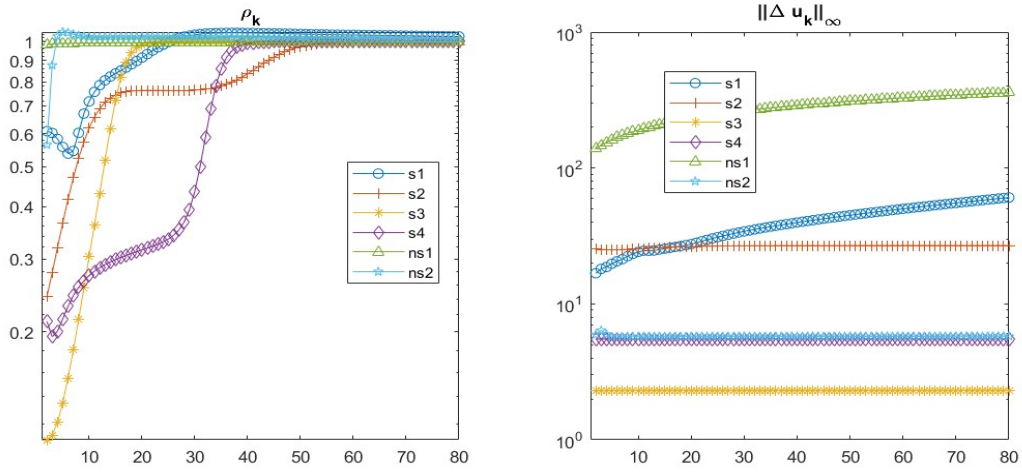


Figure 3.3: ρ_k and $\|\Delta u_k\|_\infty$ for 80 iterations for smooth solutions s_1, s_2, s_3 and non-smooth solutions ns_1, ns_2

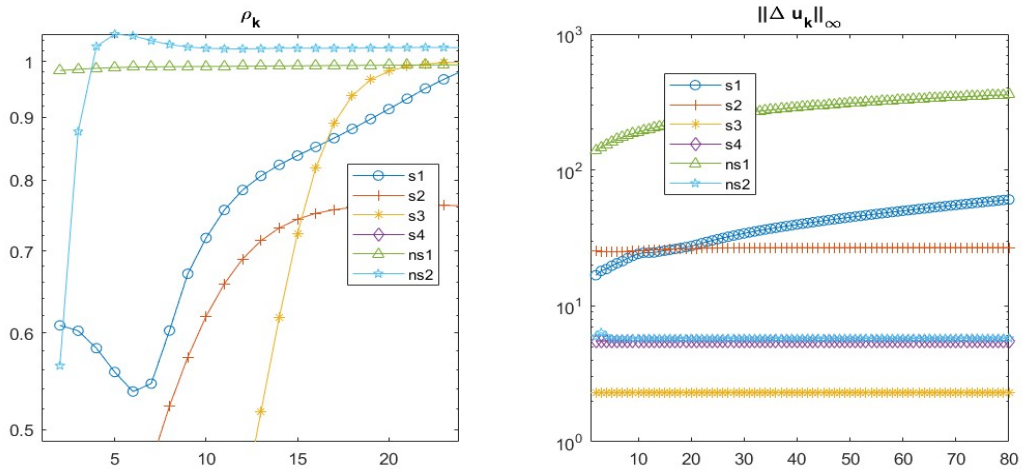


Figure 3.4: An enlarged graphs in Figure 3.3

- ns_1: $-\sqrt{3-(x^2+y^2+z^2)}$ where f is ∞ at $(1,1,1)$;
- ns_2: $\dfrac{(x^2+y^2+z^2)^{3/4}}{3}$ where f is ∞ at $(0,0,0)$.

The graphs in this figure above show that $\rho_k < 1$ and $\|\Delta u_k\|_\infty$ are bounded for smooth testing solutions. However, ρ_k may be bigger than 1 and $\|\Delta u_k\|_\infty$ may increase which maybe unbounded for nonsmooth testing functions.

When $\rho_k > 1$, the above analysis will not be useful to see the convergence of the sequence $\{u_k\}$. The remaining case is $\rho_k \leq 1$. In this case, we need Algorithm 3. That is, we now study the convergence of our Algorithm 3. Letting $u^k, k \geq 1$ be the sequence from Algorithm 3, it is easy to see that

$$u_{k+1} - u^* = \frac{1}{2}(u_k - u^*) + \frac{1}{2}(T(u_k) - T(u^*)) \tag{3.22}$$

for all $k \geq 1$ since u^* is a fixed point of T. Since $\rho_k \leq 1$, we have $\|T(u_k) - T(u^*)\| \leq \|u_k - u^*\|$ and hence, $\|u_{k+1} - u^*\| \leq \|u_k - u^*\|$ for all $k \geq 1$. It follows that $u_k, k \geq 1$ are bounded in a $H^2(\Omega)$ norm. We now show the averaged iterative algorithm converges.

Theorem 22. *Suppose that Ω is a bounded domain which has a uniformly positive reach. Suppose that $g \in H^{1/2}(\partial\Omega)$. Suppose that $\rho_k \leq 1$. Then Averaged Iterative Algorithm 3 converges.*

Proof. Let $S = H^2(\Omega)$. By the assumptions, the operator T defined above from S to S is a continuous and nonexpansive operator. We first recall the following equality: For any $x, y, z \in S$ and a real number $\lambda \in [0,1]$, we have the following identity

$$\lambda\|x-z\|^2 + (1-\lambda)\|y-z\|^2 - \lambda(1-\lambda)\|x-z\|^2 = \|\lambda x + (1-\lambda)y - z\|^2.$$

The proof is left to the interested reader. Let $\lambda = 1/2$ and $x = u^k$, $y = T(u^k)$, and $z = u^*$ which is a fixed point or the solution. Then we have

$$\begin{aligned}\|u_{k+1} - u^*\|^2 &= \frac{1}{2}\|u_k - u^*\|^2 + \frac{1}{2}\|T(u_k) - u^*\|^2 - \frac{1}{4}\|u_k - T(u_k)\|^2 \\ &= \frac{1}{2}\|u_k - u^*\|^2 + \frac{1}{2}\|T(u_k) - T(u^*)\|^2 - \frac{1}{4}\|u_k - T(u_k)\|^2 \\ &\leq \|u_k - u^*\|^2 - \frac{1}{4}\|u_k - T(u_k)\|^2.\end{aligned}$$

It follows that

$$\sum_{k=1}^{N} \frac{1}{4}\|u_k - T(u^k)\|^2 + \|u_{N+1} - u^*\|^2 \leq \|u_0 - u^*\|^2$$

for any integer $N > 1$. That is, $\|u_k - T(u_k)\| \to 0$ when $k \to \infty$.

We now claim that the sequence $u_k, k \geq 1$ converges. Note that due to the nonexpansiveness, $\|u_k\|, k \geq 1$ are bounded as explained above. Let $\hat{u}$ be the limit of a subsequence of $u^k, k \geq 1$. Then we have $\hat{u} = T(\hat{u})$ by the continuity of the operator T. So $\hat{u}$ is a fixed point of T. By the definition of T, we have

$$\Delta\hat{u} = \sqrt[3]{(\Delta\hat{u})^3 + 27(f - \det(D^2\hat{u}))}$$

or $(\Delta\hat{u})^3 = (\Delta\hat{u})^3 + 27(f - \det(D^2\hat{u}))$. It follows that $f = \det(D^2\hat{u})$. Since the Monge-Ampére equation has a unique solution, we have $\hat{u} = u^*$. If there exists another $\tilde{u}$ which is the limit of another subsequence of $u^k, k \geq 1$, we also have $\tilde{u} = T(\tilde{u})$. Then $\tilde{u} = u^*$. Hence, the sequence $\{u_k, k \geq 1\}$ from Algorithm 3 converges. □ □

3.2 Numerical Results for 3D Monge-Ampère Equations

In this section, we present numerical results from various computational experiments. We will first test several smooth and nonsmooth solutions over convex domains, such as $[0,1]^3$. Next, we show the numerical results over non-convex domains such as C, L, S-shaped domains. For all the experiments, we use 10 processors on a parallel computer equipped with a 12th Gen Intel(R) Core(TM) i7-12650H processor running at 2.30 GHz and 16.0 GB of installed RAM. All the cases, the errors are computed based on $NI = 351 \times 351 \times 351$ equally spaced points $\{(\eta_i)\}_{i=1}^{NI}$ fell inside the domain of computation. The errors will be calculated according to the norms

$$\begin{cases} |u|_{l_2} &= \sqrt{\frac{\sum_{i=1}^{NI}(u(i))^2}{NI}} \\ |u|_{h_1} &= \sqrt{\frac{\sum_{i=1}^{NI}(u(i))^2+(u_x(i))^2+(u_y(i))^2+(u_z(i))^2}{NI}} \\ |u|_{l_\infty} &= \max|u(i)|, \end{cases}$$

where $u(i) := u(\eta_i), u_x(i) := u_x(\eta_i)$, $u_y(i) := u_y(\eta_i)$ and $u_z(i) := u_z(\eta_i)$ for given functions u, u_x, u_y, u_z. Tables in this section are the numerical results of $|e_s|_{l_2}$ and $|e_s|_{h_1}$, where $e_s := u - u_s$.

3.2.1 Smooth Testing Functions

Example 6 (Polynomial Examples)**.** *In [CGG18], the researchers experimented with the following two smooth exact solutions:*

- $f^{3d1} = 75$ *such that an exact solution is* $u^{3ds1} = \frac{1}{2}(x^2 + 5y^2 + 15z^2)$
- $f^{3d2} = 1000$ *such that an exact solution is* $u^{3ds2} = \frac{1}{2}(x^2 + 10y^2 + 100z^2)$

The numerical results of their least squares/relaxation method (called LR method in this chapter) are shown in Table 3.3. Together we present numerical results based on our spline collocation method by Algorithms 2. Table 3.3 shows our spline collocation method (called LL method) produces more

Table 3.3: Errors of numerical solutions u^{3ds1}, u^{3ds2} for Monge Ampère equation over $[0,1]^3$ for LL methods with $D = 5$, $r = 1$ and LR method in [CGG18]

LR method				
h	u^{3ds1}		u^{3ds2}	
	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$
0.2	7.19e-02	1.58e-00	2.74e-02	5.16e-01
0.1	1.80e-02	7.91e-01	7.52e-03	2.81e-01
0.0625	7.06e-03	4.95e-01	3.06e-03	1.83e-01
0.04	2.89e-03	3.16e-01	1.26e-03	1.20e-01
LL method				
h	u^{3ds1}		u^{3ds2}	
	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$
0.25	2.68e-07	1.01e-05	2.48e-04	4.63e-03

accurate solutions than those presented in [CGG18]. The eigenvalues of the Hessian matrix are $1, 5, 15$ *and therefore* $det(D^2u^{3ds1}) = \lambda_1\lambda_2\lambda_3 = 75$ *and* $\Delta u^{3ds1} = \lambda_1 + \lambda_2 + \lambda_3 = 21$. *In Algorithm 3, we choose an initial value* $\Delta u_0 = 14.55$ *to approximate the exact solution* u^{3ds1}. *This choice of initial value leads to converging iterations since 14.55 is close to* $\Delta u^{3ds1} = \sqrt[3]{27f} = \sqrt[3]{27 \cdot 75} = 12.65$. *Similarly, we choose our initial value* u_0 *for* u^{3ds2} *satisfying* $\Delta u_0 = 106.2$ *which makes the iterations from Algorithm 3 converge. By choosing a good initial value* u_0 *we achieve the numerical results shown in Table 3.3.*

We also test other smooth solutions which were experimented in the literature, e.g., [Awa13], [Awa15], [CGG18], [Ben+20], and etc..

Example 7 (Smooth Exponential Functions)**.** *Consider a smooth exponential exact solution* $u^{3ds3} = e^{\frac{(x^2+y^2+z^2)}{2}}$ *associated with* $f^{3d3} = (1 + x^2 + y^2 + z^2)e^{\frac{3(x^2+y^2+z^2)}{2}}$. *We compare our methods with the least squares/relaxation method(LR method) in [CGG18]. Table 3.4 shows comparison results including* l_2, h_1 *norm of these two methods for each mesh size* h.

Table 3.4: Errors of numerical solutions u^{3ds3} and CPU time(s) for Monge Ampère equation over $[0, 1]^3$ for LL methods with $D = 5$, $r = 1$ and LR method in [CGG18]

LR method					LL method					
h	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate	h	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate	Time(s)
0.2	7.19e-02	-	1.58e-00	-	1	1.17e-03	-	1.24e-02	-	0.06
0.1	1.80e-02	1.99	7.91e-01	0.99	0.5	4.36e-05	4.74	7.82e-04	3.99	0.41
0.0625	7.06e-03	1.99	4.95e-01	1.00	0.25	1.42e-06	4.94	2.72e-05	4.84	3.05
0.04	2.89e-03	1.99	3.16e-01	0.99	0.125	1.10e-07	3.69	1.36e-06	4.32	62.4

We can see that better convergence results using LL methods with $D = 5, r = 1$. *In Figure 3.5, we show plots of the* $|e_s|_{l_2}, |e_s|_{h^1}$ *with respect to the mesh size* h. *We can see that the rate of convergences is about* $O(h^{4.82})$.

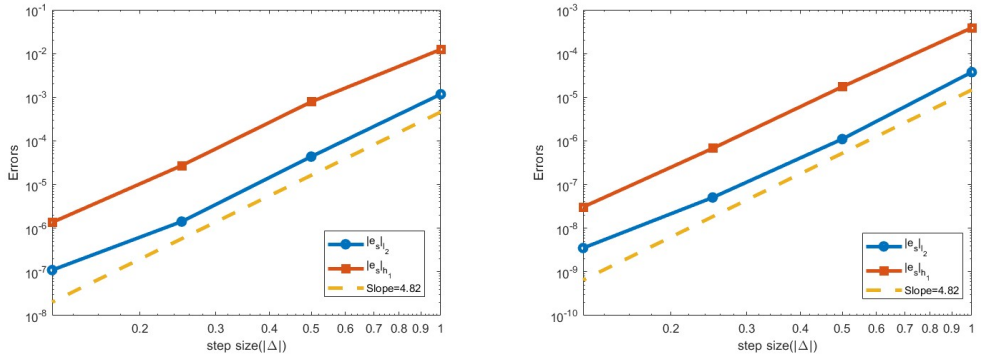


Figure 3.5: Convergence rates of l_2, h_1 errors for solutions u^{3ds3}(Left) and u^{3ds5}(Right) with respect to $|\triangle|$ based on the LL method

Example 8. *In [Awa13] and [Awa15], Awanou introduced the pseudo transient continuation, time marching methods and the spline element methods for Monge-Ampére equations. He presented*

several 2D and 3D numerical examples by his methods. For testing function $u^{3ds4} = e^{\frac{(x^2+y^2+z^2)}{3}}$*, it seems that the numerical results using the spline element method (SE method) in [Awa15] is the best. We use our spline collocation method(LL method) and compare* L^2, H^1, H^2 *errors of our method and the SE method. Table 3.5 and 3.6 show that we can get better accuracy when* $h = 1$ *and* $h = 1/2$ *for each degree* $D = 4, 5, 6$*.*

Table 3.5: Errors of numerical solutions u^{3ds4} for Monge Ampère equation over $[0, 1]^3$ for LL methods with $D = 3, 4, 5, 6$, $r = 1$, $h = 1$ and SE method in [Awa15]

D	SE method			LL method		
	L^2 norm	H^1 norm	H^2 norm	L^2 norm	H^1 norm	H^2 norm
3	1.2338e-02	7.6984e-02	4.4411e-01	1.6916e-02	1.0879e-01	3.8250e-01
4	1.6289e-03	1.4719e-02	1.3983e-01	6.4696e-04	6.1874e-03	3.7146e-02
5	1.5333e-03	8.7312e-03	6.0412e-02	1.7440e-04	2.2203e-03	1.7392e-02
6	1.2324e-04	9.7171e-04	1.0584e-02	4.6740e-05	6.2257e-04	3.5432e-03

Table 3.6: Errors of numerical solutions u^{3ds4} for Monge Ampère equation over $[0, 1]^3$ for LL methods with $D = 3, 4, 5, 6$, $r = 1$, $h = 1/2$ and SE method in [Awa15]

D	SE method			LL method		
	L^2 norm	H^1 norm	H^2 norm	L^2 norm	H^1 norm	H^2 norm
3	3.1739e-03	2.3005e-02	2.4496e-01	2.4294e-03	1.5806e-02	1.0139e-01
4	3.2786e-04	3.5626e-03	5.2079e-02	9.5591e-05	1.1644e-03	9.8077e-03
5	2.4027e-05	3.9210e-04	8.8868e-03	5.8750e-06	1.2214e-04	1.4292e-03
6	1.3821e-06	2.2369e-05	6.0918e-04	6.0635e-07	1.4198e-05	1.6487e-04

3.2.2 Non-smooth Testing Functions

Example 9. *In [CGG18], the researchers considered the following problem which does not have an exact solution with the* $H^2(\Omega)-$ *regularity or may have no solution at all. For* $R \geq \sqrt{3}$*, let* $u = -\sqrt{R^2 - (x^2 + y^2 + z^2)}$ *be a testing function. When* $R > \sqrt{3}$*, this function belongs to* $C^\infty(\bar{\Omega})$*,*

while $u \in C^0(\bar{\Omega}) \cap W^{1,s}(\Omega)$, *with* $1 \leq s < 2$, *if* $R = \sqrt{3}$. *More precisely, let us consider the following two solutions*

$$u^{3ds5} = -\sqrt{6 - (x^2 + y^2 + z^2)} \ \ with \ \ f^{3d5} = 6(6 - (x^2 + y^2 + z^2))^{-\frac{5}{2}}$$

and

$$u^{3ds6} = -\sqrt{3 - (x^2 + y^2 + z^2)} \ \ with \ \ f^{3d6} = 3(3 - (x^2 + y^2 + z^2))^{-\frac{5}{2}}.$$

The numerical results from the least squares/relaxation method in [CGG18] (called LR method) are shown in Table 3.7. In Figure 3.5, we can see that the rate of convergences of u^{3ds5} *are about* $O(h^{4.82})$. *In addition, we show our spline collocation method (called LL method) in the same table for comparison.*

Table 3.7: Errors of numerical approximation of the solution u^{3ds5} for Monge Ampère equation over $[0, 1]^3$ by the and LR method and by the LL method with $D = 5, r = 1$

LR method					LL method				
$\|\triangle\|$	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate	$\|\triangle\|$	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate
0.2	4.96e-03	-	8.60e-02	-	1	3.75e-05	-	3.88e-04	-
0.1	1.28e-03	1.95	4.41e-02	0.96	0.5	1.10e-06	5.09	1.73e-05	4.49
0.0625	5.09e-04	1.96	2.78e-02	0.97	0.25	5.05e-08	4.45	6.73e-07	4.69
0.04	2.10e-04	1.97	1.79e-02	0.98	0.125	3.52e-09	3.84	3.06e-08	4.46

It is clear to see that when the solution u^{3ds5} *is smooth, both methods work nicely, and our collocation method is much more accurate.*

Next, let us consider the non-smooth solution u^{3ds6} *in Table 3.8. Table 3.8 shows numerical results such as* l_2, h_1 *errors of these two methods for various mesh sizes. Our method can get a more accurate solution with* $D = 5, r = 1$ *with the large mesh size* $|\triangle|$. *However, it is clear that an*

Table 3.8: Errors of numerical approximation of the solution u^{3ds6} for Monge Ampère equation over $[0,1]^3$ by the and LR method and by the LL method with $D = 5$, $r = 1$

LR method				
$\|\triangle\|$	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate
0.2	1.15e-02	-	6.60e-01	-
0.1	3.06e-03	1.91	6.31e-01	-
0.0625	1.24e-03	1.92	6.25e-01	-
0.04	5.17e-04	1.96	6.22e-01	-

LL method				
$\|\triangle\|$	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate
1	8.07e-02	-	7.02e-01	-
0.5	7.06e-03	3.52	1.63e-01	2.10
0.25	4.78e-04	3.88	2.21e-02	2.89
0.125	3.85e-04	0.31	2.54e-02	-0.20
0.0625	3.57e-04	0.11	1.98e-02	0.35

adaptive method is needed to improve the approximation since the maximal error, $e_s = u - u_s$, is worst near the point $(1, 1, 1)$.

3.2.3 Numerical Results over Nonconvex Domains

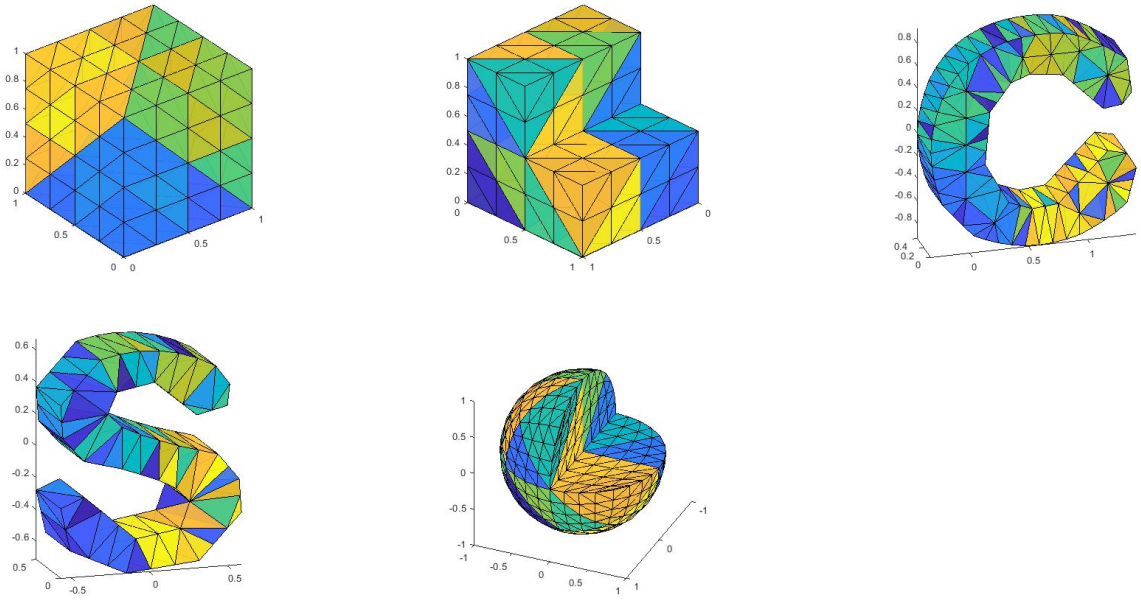


Figure 3.6: Several 3D domains (Top: Cube, Letter L, Letter C, Bottom: Letter S, Subset of the unit ball)

In this section, we test various solutions for each domain in Figure 3.6. We display CPU times versus the number of vertices and triangles in Table 3.9 for each domain in Figure 3.6 when $D = 9, r = 1$.

Table 3.9: CPU time results and numbers of vertices and tetrahedrons over domains in Figure 3.6 when $D = 9, r = 1$

Domain	No. of Vetices	No. of Tetrahedrons	Total CPU(s)
Cube	125	384	45.3
Letter L	105	288	32.2
Letter C	190	431	105.4
Letter S	115	171	36.3

Example 10. *We use our method to numerically solve three smooth testing functions* $u^{3ds3}, u^{3ds4}, u^{3ds5}$ *over 5 solids which are not strictly convex or not convex. They even do not have a uniformly positive reach. Table 3.10 shows our method performs very well.*

Table 3.10: Errors of numerical solutions $u^{3ds3} - u^{3ds5}$ for Monge Ampère equations over several domains in Figure 3.6 for LL methods with $D = 9$, $r = 1$

Solution	Cube		Letter L		Letter C		Letter S	
	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$
u^{3ds3}	1.76e-09	1.64e-07	6.63e-10	2.58e-08	1.48e-08	8.67e-07	3.39e-11	1.75e-09
u^{3ds4}	2.82e-11	1.91e-10	3.90e-11	1.04e-09	2.31e-08	3.56e-07	5.84e-11	2.70e-09
u^{3ds5}	5.05e-02	3.61e-01	2.47e-08	9.56e-07	3.87e-08	3.32e-06	6.03e-10	5.03e-08

Example 11. *In [CGG18], the researchers considered the problem over the unit ball* $\Omega = \{(x, y, z)|x^2 + y^2 + z^2 < 1\}$ *and a convex solution*

$$u^{3ds7} = -\frac{1}{2\sqrt{3}}(1 - x^2 - y^2 - z^2)$$

of the Monge-Ampére-Dirichlet problem with $f = \frac{1}{3\sqrt{3}}$. *They experimented with their numerical solutions (called LR method) over the unit ball as well as the 3/4 ball as shown in Figure 3.6.*

In Table 3.11, we first include the numerical results from [CGG18] and then compare the $L^2(\Omega), H^1(\Omega)$ *norms of the computed approximation error* $u^{3ds7} - u_s$ *by our spline collocation method. In addition, we tested the solution* u^{3ds7} *over the subset of the unit ball as shown in Figure*

Table 3.11: Errors of numerical approximation of solution u^{3ds7} for Monge Ampère equation over the unit ball for the LR method and the LL method with $D = 5$, $r = 1$

LR method				
$\|\triangle\|$	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate
2.98e-01	3.26e-02	-	2.60e-01	-
1.61e-01	1.11e-02	1.74	1.28e-01	1.14
8.32e-02	3.22e-03	1.88	6.16e-02	1.11
4.34e-02	9.89e-04	1.80	2.86e-02	1.17

LL method				
$\|\triangle\|$	$\|e_s\|_{l_2}$	rate	$\|e_s\|_{h^1}$	rate
1	3.71e-13	-	3.15e-12	-
0.5	2.97e-14	3.64	1.39e-13	4.51

3.6. The numerical results we obtained are displayed in Table 3.12.

Table 3.12: CPU time and errors of our spline solution to u^{3ds7} for Monge Ampère equation over the domain in Figure 3.6 with $D = 5$, $r = 1$, the number of vertices=585, the number of tetrahedrons=2304

LL method		
CPU time	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$
174.70	1.90e-08	2.49e-06

3.2.4 Comparison with Numerical Method in [Ben+20]

In this section, we compare our LL method with the Cascadic method in [Ben+20]. The researchers presented several examples in [Ben+20] over the irregular domains in Figure 3.7 by using the

following test functions

$$
\begin{aligned}
u^{3ds3} &= e^{\frac{(x^2+y^2+z^2)}{2}}, \\
u^{3ds6} &= -\sqrt{3-(x^2+y^2+z^2)}, \\
u^{3ds8} &= \frac{x^2+y^2+z^2}{2} - \sin(x) - \sin(y) - \sin(z), \\
u^{3ds9} &= \frac{(x^2+y^2+z^2)^{\frac{3}{4}}}{3}.
\end{aligned}
$$

We use our method (LL method) to compute numerical solutions based on the same testing functions over the same testing domains. Our numerical results are shown in Tables 3.13, 3.14 and 3.15.

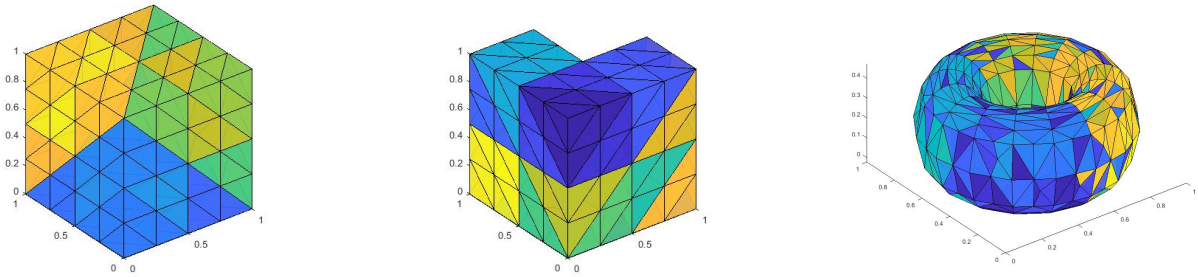


Figure 3.7: Several 3D domains (Cube, Letter L, Torus)

Table 3.13: The CPU time, DOFs, errors $|e_s|_{l_2}$, $|e_s|_{h_1}$ using LL method with $D = 6, r = 1$ and $|e_s|_{l_2}$ using the Cascadic method in [Ben+20] over the cube $[0, 1]^3$

	LL method				Cascadic method
solution	CPU time	DOFs	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$
u^{3ds3}	4.85	32256	4.40e-08	1.61e-06	9.86e-04
u^{3ds6}	2.29	32256	4.02e-04	2.67e-02	1.78e-04
u^{3ds8}	4.40	32256	1.81e-11	6.89e-10	3.50e-04
u^{3ds9}	8.45	32256	1.32e-04	1.13e-03	2.23e-04

Table 3.14: The CPU time, DOFs, errors $|e_s|_{l_2}, |e_s|_{h_1}$ using LL method with $D = 6, r = 1$ and $|e_s|_{l_2}$ using the Cascadic method in [Ben+20] over L-shaped domain

	LL method				Cascadic method
solution	CPU time	DOFs	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$
u^{3ds3}	3.18	24192	2.16e-07	9.71e-06	4.8655e-03
u^{3ds6}	3.15	24192	7.42e-06	3.82e-04	1.6238e-04
u^{3ds8}	2.58	24192	3.90e-11	2.00e-09	1.2240e-04
u^{3ds9}	7.37	32256	5.48e-06	4.59e-04	4.2183e-04

Table 3.15: The CPU time, DOFs, errors $|e_s|_{l_2}, |e_s|_{h_1}$ using LL method with $D = 3, r = 1$ and $|e_s|_{l_2}$ using the Cascadic method in [Ben+20] over Torus

	LL method				Cascadic method
solution	CPU time	DOFs	$\|e_s\|_{l_2}$	$\|e_s\|_{h^1}$	$\|e_s\|_{l_2}$
u^{3ds3}	28.24	46280	3.45e-05	5.93e-04	3.1914e-04
u^{3ds6}	26.75	46280	8.56e-06	1.52e-04	1.9850e-04
u^{3ds8}	26.07	46280	1.17e-06	1.97e-05	2.1182e-04
u^{3ds9}	26.64	46280	1.96e-06	9.48e-05	1.7504e-04

3.2.5 Optimal Transport with the Monge-Ampére Equation and Neumann Boundary Condition

Finally, we consider the optimal transport problem (3.4)

$$\det(D^2 u(\mathbf{x})) = \frac{f(\mathbf{x})}{g(\nabla u(\mathbf{x}))}, \quad \mathbf{x} \text{ in } \Omega \subset \mathbb{R}^2 \tag{3.23}$$

$$\nabla u(\mathbf{x}) = \partial W, \quad \mathbf{x} \text{ on } \partial\Omega. \tag{3.24}$$

In [Fro12], the researchers solve the following projection method

$$\det(D^2u^{k+1}(x)) = \frac{f(x)}{g(\nabla u^{k+1}(x))}, x \in V$$

$$\nabla u^{k+1} \cdot n_x = p_k \cdot n_x,\ x \in \partial V$$

$$u^{k+1} \text{ is convex}$$

$$\int_X u(x)dx = 0$$

where p^k is the projection of ∇u^k onto the boundary of the target domain W, i.e. ∂W. This system is solved efficiently with Newton's method.

In this section, we let the function g be a constant. Then, the Monge-Ampére equation (3.4) becomes

$$\det(D^2u(\mathbf{x})) = Cf(\mathbf{x}),\ \ \mathbf{x} \text{ in } \Omega \subset \mathbb{R}^3 \tag{3.25}$$

$$\nabla u(\mathbf{x}) = \partial W,\ \ \mathbf{x} \text{ on } \partial\Omega. \tag{3.26}$$

And then we consider the following algorithm

$$\det(D^2u^{k+1}(x)) = Cf(x), x \in V \tag{3.27}$$

$$\nabla u^{k+1} \cdot n_x = p_k \cdot n_x,\ x \in \partial V \tag{3.28}$$

$$u^{k+1} \text{ is convex and } \int_X u(x)dx = 0 \tag{3.29}$$

where $C > 0$ is a constant and p^k is the projection of ∇u^k onto the boundary of the target domain W.

Now we can apply our computational approach to find a solution of u and form a transportation map from Ω to W.

Example 12. *In [***BFO14***], the researchers introduced a mapping with an exact solution that involves mapping a square onto another square. To set up this example, we define the function*

$$q(z) = \left(-\frac{1}{8\pi}z^2 + \frac{1}{256\pi^3} + \frac{1}{32\pi}\right)\cos(8\pi z) + \frac{1}{32\pi^2}z\sin(8\pi z).$$

Now we map the density

$$f(x_1, x_2) = 1 + 4(q''(x_1)q(x_2) + q(x_1)q''(x_2)) + 16(q(x_1)q(x_2)q''(x_1)q''(x_2) - q'(x_1)^2q'(x_2)^2)$$

in the square $(-0.5, 0.5) \times (-0.5, 0.5)$ *onto a uniform density* $g = 1$ *in the same square. This transport problem has the exact solution*

$$u_{x_1}(x_1, x_2) = x_1 + 4q'(x_1)q(x_2), \quad u_{x_2}(x_1, x_2) = x_2 + 4q(x_1)q'(x_2).$$

Using our algorithm (3.27) *with initial value* $\nabla u_0 = (x_1, x_2)$*, we can get the numerical results in Table 3.16 and this shows that our algorithms works well. And Figure 3.8 shows that the graph of* f, g *and* $g(\nabla u_s)det(D^2u_s)$.

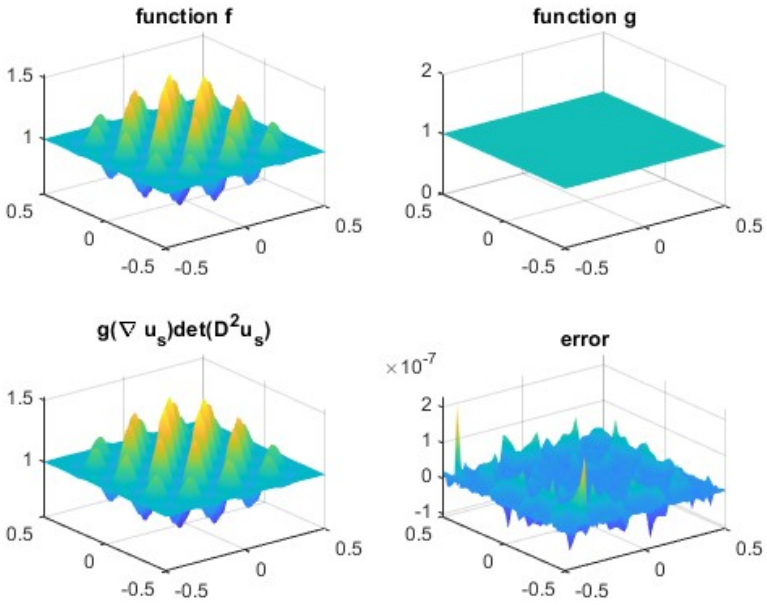


Figure 3.8: Function f, g and error $\|g(\nabla u_s)\det(D^2 u_s) - f\|_\infty$ in Example 12

Table 3.16: Numerical results for example 12

Degree	CPU time	Max error	L^2 error	$\|\det(D^2 u_k) - f/g(\nabla u)\|_\infty$
5	1.75	3.49e-03	4.48e-04	1.12e-01
8	3.21	3.37e-05	3.35e-06	1.17e-03
11	38.0	3.37e-07	3.11e-08	1.62e-05
14	95.9	2.96e-09	4.87e-10	2.98e-08

Example 13. *Let* $\Omega = [-1, 1]^2$ *and let* $f(x_1, x_2) = 2 + \tanh(10(x_1 - 1/2))/2$ *and* $g = 7/4$. *We use our optimal transport algorithm* (3.27) *with initial value* $\nabla u_0 = (x_1, x_2)$. *Figure 3.9 illustrates the graph of* f, g, *and* $g(\nabla u_s)\det(D^2 u_s)$.

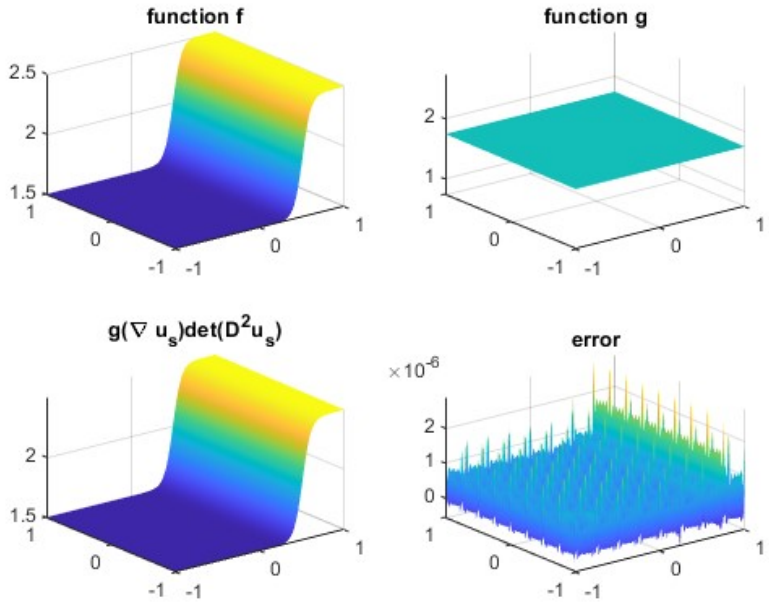


Figure 3.9: Graph of f, g, and $g(\nabla u_s)\det(D^2 u_s)$ in Example 13

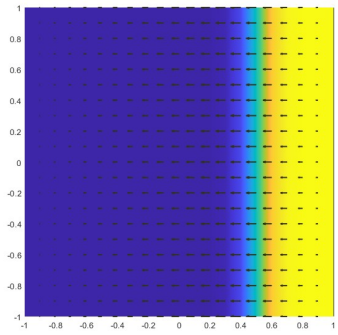


Figure 3.10: Plot of the vector field $(u_x - x, u_y - y)$ overlaid with a color map representing the function f (the color map shows the value of f at each point)

Chapter 4

Spline Collocation Method for Chemotaxis with Flow Advection

In this chapter, we explore the problem of optimal control design for the suppression of singularity formation in chemotaxis via flow advection. Chemotaxis refers to the movement of cells in response to a chemical stimulus. The process is typically modeled using a coupled parabolic system, which was first introduced by Patlak [Pat53] and further developed by Keller and Segel [KS70; KS71] to describe the evolving densities of one or more chemotactic populations and their attractants/repellents. Several related studies and reviews can be found in the literature, such as [Hor03; Hor04; PD09; SS11; Suz05].

The focus of our work is a simplified parabolic-elliptic Patlak-Keller-Segel (PKS) equation with flow advection, which was introduced by Nagai in [Nag95] (also see [JL92]). The flow advection, induced by the movement of the ambient fluid, occurs in an open bounded and connected domain $\Omega \subset \mathbb{R}^d$, where $d = 2, 3$, with a smooth boundary Γ. Our approach involves designing an optimal

control strategy that can effectively suppress the formation of singularities in the chemotactic population, which can arise due to excessive aggregation or depletion of cells. This problem has significant applications in the field of tissue engineering and cancer treatment, where controlling the movement and proliferation of cells is crucial for achieving desirable outcomes.

Let $\theta \geq 0$ be the density of the cells and $c \geq 0$ be the concentration of a chemoattractant produced by the cells. Further, let $\mathbf{v} = \mathbf{v}(x)$ be a predetermined time-independent incompressible flow and $u = u(t)$ be a time-dependent control input regulating the strength of the flow. The system with controlled flow advection is governed by

$$\frac{\partial \theta}{\partial t} = \Delta\theta - u(t)\mathbf{v} \cdot \nabla\theta - \nabla \cdot (\theta\chi\nabla c)) \quad \text{in} \quad \Omega, \tag{4.1}$$

$$-\Delta c + c = \theta \quad \text{in} \quad \Omega, \tag{4.2}$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in} \quad \Omega, \tag{4.3}$$

with Neumann boundary conditions for both $\theta \geq 0$ and $c \geq 0$, no-penetration condition for $\mathbf{v}$

$$\frac{\partial \theta}{\partial n} = \frac{\partial c}{\partial n} = 0 \quad \text{and} \quad \mathbf{v} \cdot n = 0 \quad \text{on} \quad \Gamma, \tag{4.4}$$

and the initial condition

$$\theta(x, 0) = \theta_0(x) \quad \text{in} \quad \Omega, \tag{4.5}$$

where $\chi > 0$ is a sensitivity parameter of the cells to the chemo-attractant c and n is the outward unit normal vector to the domain boundary Γ. The objective is to seek an optimal regulating function

$u(t)$ for the ambient fluid so as to suppress the possible finite time blow-up. It is well-studied that in the absence of flow advection or the drift if the initial condition is above a certain critical threshold, the solution of the PKS equations may blow up in finite time by concentrating positive mass at a single point (e.g. [JL92; HV97; HMV97; HMV98; Her00]). With flow advection, however, for any initial distribution there exists an ambient velocity field $\mathbf{v}$, either time-independent or dependent, such that the solution to (4.1)–(4.5) is globally regular for all positive time [KX16]. Singularity formation can be prevented via flow advection by mixing the cell in the direction that mitigates concentration. Iyer, Xu, and Zlatoš in [IXZ21] showed that if the flow has small dissipation times, the global well-posedness result can be obtained in the torus $\mathbb{T}^d$, $d = 2, 3$. Without loss of generality, we may assume that $u(t)$ is a positive constant. Indeed, if $u < 0$, it can be treated similarly by letting $\mathbf{v}$ be $-\mathbf{v}$. The small dissipation times can be obtained by increasing u, if the operator $\mathbf{v} \cdot \nabla$ has no eigenfunctions in $H^1(\Omega)$ other than the constant function. In this case, the incompressible flow $\mathbf{v}$ is so-called *relaxation enhancing* [Con+08, Def. 1.1]. However, such flows are rather complex to construct and many flows in real-world applications do not necessarily possess this property. The authors in [IXZ21] further showed that the flows with arbitrarily small dissipation times can be constructed by rescaling a general class of smooth (time-independent) cellular flows. The proof is based on the probabilistic method. A recent work by Hu in [Hu22] considered the PKS system in a general bounded domain and showed that if the semigroup generated by the advection-diffusion operator has a *rapid decay property* (defined in Section 4.1), then the global well-posedness can be established. A direct estimate showed that for cellular flows in rectangle-domains, rescaling the cell size and the flow amplitude can make the decay rate of the semigroup arbitrarily fast. Other related work on suppression of singularity by shear flows can be also found in (e.g. [BH17]).

In the rest of our discussion, we set $\chi = 1$. Let $\theta \geq 0$ be the global in time solution to the PKS system (4.1)–(4.5). With boundary conditions in (4.4), it is easy to verify that

$$\bar{\theta}(t) = \frac{1}{|\Omega|}\int_\Omega \theta \, dx = \bar{\theta}_0,$$

for any $t > 0$. In fact, by Stokes formula, it is easy to see that

$$\begin{aligned}\frac{\partial \int_\Omega \theta \, dx}{\partial t} &= \int_\Omega \Delta\theta \, dx - u\int_\Omega \mathbf{v}\cdot\nabla\theta \, dx - \int_\Omega \nabla\cdot(\theta\chi\nabla c)) \, dx \\ &= \int_\Gamma \frac{\partial\theta}{\partial n} \, dx - u(\int_\Gamma \mathbf{v}\cdot n\theta \, dx - \int_\Omega \nabla\cdot\mathbf{v}\theta \, dx) - \int_\Gamma (\theta\chi\nabla c)\cdot n \, dx = 0,\end{aligned}$$

thus $\int_\Omega \theta \, dx = \int_\Omega \theta_0 \, dx$ for any $t > 0$. Let $\vartheta = \theta - \bar{\theta}$, then $\bar{\vartheta}(t) = 0$ for $t \geq 0$, and ϑ satisfies

$$\frac{\partial\vartheta}{\partial t} = \Delta\vartheta - A\mathbf{v}\cdot\nabla\vartheta - \nabla\cdot((\vartheta + \bar{\theta})\nabla c) \quad \text{in} \quad \Omega, \tag{4.6}$$

$$-\Delta c + c = \vartheta + \bar{\theta} \quad \text{in} \quad \Omega, \tag{4.7}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in} \quad \Omega, \tag{4.8}$$

$$\frac{\partial\vartheta}{\partial n} = \frac{\partial c}{\partial n} = 0 \quad \text{and} \quad \mathbf{v}\cdot n = 0 \quad \text{on} \quad \Gamma, \tag{4.9}$$

$$\vartheta(x, 0) = \theta_0(x) - \bar{\theta} \quad \text{in} \quad \Omega. \tag{4.10}$$

4.0.1 Control design via flow advection

Although there is quite a few literature on optimal control for chemotaxis, it predominantly focuses on the linear distributed control of the chemoattractant (e.g. [RY01; Ryu08; RRV18]) or bilinear

control of the cells or chemoattractant of the form $u\vartheta$ or uc (e.g. [GMR20b; GMR20a; FM03]), where $u = u(x,t)$ is the control input.

This is the first work, to our best knowledge, to control the PKS system via flow advection. We aim at designing an optimal input $u(t)$ for regulating the flow so that the possible blow-up in solution can be suppressed for any initial data. Here assume that $u(t)$ has both upper and lower bounds. Let the set of admissible control be

$$U_{ad} = \{u \in L^2(0,t_f) : \underline{u} < 0 \le u \le \bar{u}\}$$

and seek for $u \in U_{ad}$ that minimizes the following cost functional

$$J(u) = \frac{\alpha}{2}\|\vartheta(t_f)\|_{L^2}^2 + \frac{\beta}{2}\int_0^{t_f} \|\vartheta(t)\|_{L^2}^2\, dt + \frac{\gamma}{2}\int_0^{t_f} |u|^2\, dt, \tag{P}$$

for a given $t_f > 0$, subject to (4.6)–(4.10), where $\alpha, \beta \ge 0$ and $\gamma > 0$ are the state and control weight parameters, respectively. The parameters α and β do not vanish simultaneously.

We first show that problem (P) is well-posed, that is, for any given ϑ_0, there exits a control $u \in U_{ad}$ such that $J(u) < \infty$. To this end, it is critical to understand the well-posedness of the PKS system in the presence of flow advection. Then we proceed to prove the existence of an optimal solution. Since we have a nonlinear system associated with a nonlinear control input, problem (P) is no longer convex. As a result, the optimal solution may not be unique. Note that if $\mathbf{v}$ is chosen such that (4.20) is satisfied, then the rapid decay property (4.16) holds and hence the global well-posedness and stability can be established [Hu22]. In this case, $u(t)$ can be always set as a constant $A = A(\vartheta_0, \bar{\theta})$ large enough so that $\vartheta \in C([0,\infty); L^2(\Omega)) \cap L^2_{loc}(0,\infty; H^1(\Omega))$, and

therefore $J(u) < \infty$. On the other hand, as long as we let $\bar{u} \geq A$, the set of admissible controls is nonempty.

The rest of this work is organized as follows. In Section 4.1, we first recall the global well-posed result of the PKS system in a bounded domain, which paves a way for establishing the well-posedness of the optimal control problem (P) and proving the existence of an optimal solution. In Section 4.2, we derive the first-order optimality conditions for solving the optimal solution using the variational inequality (e.g. [Lio71]). Finally, in Section 4.3 we use the spline collocation method for implementing the optimality system (e.g. [Lai89; ALW06; GLS15; LL22]). Numerical experiments based on 2D cellular flows for suppression of singularity in rectangle domains will be presented to demonstrate the effectiveness of our control design.

4.1 Well-posedness of the PKS system and existence of an optimal control

To start with, we introduce the well-posedness of the PKS system, where $\mathbf{v}$ is assumed to be divergence-free and time-independent. Let $X = L^2_0(\Omega) = \{\psi \in L^2(\Omega) : \int_\Omega \psi \, dx = 0\}$ be the subspace of mean zero functions. Define

$$\mathcal{L}_A = \Delta - A\mathbf{v} \cdot \nabla$$

with $D(\mathcal{L}_A) = \{\phi \in H^2(\Omega) \cap X : \frac{\partial \phi}{\partial n}|_\Gamma = 0\}$. Then $\mathcal{L}_A$ is strictly negative. Moreover, it is m-accretive as the left open half-plane is contained in its resolvent set $\varrho(L_A)$ and

$$(\mathcal{L}_A + \lambda)^{-1} \in \mathscr{L}(L^2(\Omega)), \quad \|(\mathcal{L}_A + \lambda)^{-1}\| \leq \Re\lambda^{-1} \quad \text{for } \Re\lambda > 0.$$

Further define the nonlinear operator $\mathcal{N} : H^1(\Omega) \to X$ by

$$\mathcal{N}\vartheta = -\nabla \cdot ((\vartheta + \bar{\theta})\nabla c). \tag{4.11}$$

The system can be rewritten as an abstract Cauchy problem on a state space X

$$\dot{\vartheta} = \mathcal{L}_A\vartheta + \mathcal{N}\vartheta, \tag{4.12}$$

$$\vartheta(0) = \vartheta_0, \tag{4.13}$$

where the system operator $\mathcal{L}_A$ generates an analytic semigroup, denoted by $e^{\mathcal{L}_A t}, t \geq 0$, on X and the nonlinearity of operator $\mathcal{N}$ can be characterized by the following results (e.g. [Hu22, Lemma 2.4], [IXZ21, (H2), p. 2]).

Lemma 8. *For $\vartheta \in H^1(\Omega)$, then there is a constant $C_1 > 0$ such that*

$$||\mathcal{N}\vartheta||_{L^2} \leq C_1(||\vartheta||^2_{H^1} + |\bar{\theta}|\|\vartheta\|_{L^2}). \tag{4.14}$$

Moreover, for $\vartheta \in L^2(\Omega)$, there is a constant $C_2 > 0$ such that

$$\|(-\mathcal{L}_A)^{-\frac{3}{4}}(\mathcal{N}\vartheta)\|_{L^2} \leq C_2(\|\vartheta\|_{L^2}^2 + |\bar{\theta}|\|\vartheta\|_{L^2}). \tag{4.15}$$

We call that a semigroup has a *rapid decay property* if there exist $M_0 > 0$ and $\omega_A > 0$ such that

$$\|e^{\mathcal{L}_A t}\|_{\mathscr{L}(X)} \leq M_0 e^{-\omega_A t}, \quad t \geq 0, \tag{4.16}$$

where ω_A can be made arbitrarily large and M_0 is independent of ω_A. Here $\mathscr{L}(X)$ stands for the set of bounded linear operators on X and $\|\cdot\|_{\mathscr{L}(X)}$ stands for the operator norm.

To replace c in terms of ϑ, we define the operator

$$\mathcal{A} = -\Delta + I \tag{4.17}$$

with domain $D(\mathcal{A}) = \{\phi \in H^2(\Omega) \colon \frac{\partial \phi}{\partial n}|_\Gamma = 0\}$. Then $\mathcal{A}$ is strictly positive and self-adjoint, and $c = \mathcal{A}^{-1}(\vartheta + \bar{\theta}) = \mathcal{A}^{-1}\vartheta + \bar{\theta}$.

Let

$$\Psi(\mathcal{L}_A) = \inf\{\|(\mathcal{L}_A - i\lambda)\phi\|_{L^2} \colon \phi \in D(\mathcal{L}_A), \lambda \in \mathbb{R}, \|\phi\|_{L^2} = 1\}.$$

The following Gearhart-Prüss type theorem is proven in [Wei21, Theorems 1.3] for m-accretive operators

$$\|e^{\mathcal{L}_A t}\|_{\mathscr{L}(L^2(\Omega))} \leq M_0 e^{-\Psi(\mathcal{L}_A)t}, \quad t \geq 0, \tag{4.18}$$

where $M_0 = e^{\pi/2}$, and

$$\Psi(\mathcal{L}_A) \to +\infty, \quad \text{as} \quad A \to +\infty, \tag{4.19}$$

if and only if

$$\mathbf{v} \cdot \nabla \text{ has no eigenfunctions in } H^1(\Omega) \cap X. \tag{4.20}$$

In other words, the rapid decay property can be obtained if (4.20) holds. The following theorem on the global well-posedness and stability of the nonlinear system (4.12)–(4.13) for a given velocity field in a bounded domain has been established in [Hu22, Theorem 2.2], based on the classic tools of analytic semigroup theory for semilinear equations.

Theorem 23. *Let $\vartheta_0 \in X$ and $\boldsymbol{v} \in L^\infty(\Omega)$ be a divergence-free vector field satisfying $\boldsymbol{v} \cdot n|_\Gamma = 0$. If $\Psi_A = \Psi_A(\vartheta_0, \bar{\theta}) > 0$ is sufficiently large, then there exists a unique mild (weak) solution ϑ to* (4.12)–(4.13) *satisfying*

$$\vartheta \in C([0,\infty); X) \cap L^2_{loc}(0,\infty; H^1(\Omega)) \tag{4.21}$$

and

$$\sup_{t\geq 0}\|\theta\|_{L^2} \leq 2\|\vartheta_0\|_{L^2} + 1. \tag{4.22}$$

Moreover, there exist constants $M_* > 0$ *and* $\omega_0 > 0$ *such that*

$$\|\vartheta\|_{L^2} \leq M_* e^{-\omega_0 t}\|\vartheta_0\|_{L^2}. \tag{4.23}$$

Using the variation of parameters formula we can express the solution to (4.12)–(4.13) as

$$\vartheta(t) = e^{\mathcal{L}_A t}\vartheta_0 + \int_0^t e^{\mathcal{L}_A(t-\tau)}(\mathcal{N}\vartheta)(\tau)\, d\tau. \tag{4.24}$$

Furthermore, by (4.14)–(4.15) and (4.21)-(4.22) we have $\Delta\theta \in L^2(0,t_f;(H^1(\Omega))')$, $\mathbf{v}\cdot\nabla\vartheta \in L^2(0,t_f;L^2(\Omega))$, and

$$\mathcal{N}\vartheta \in L^1(0,t_f;L^2(\Omega)) \cap L^2(0,t_f;(D(-\mathcal{L}_A)^{3/4})'),$$

for any $0 < t_f < \infty$. Therefore, we can derive that

$$\frac{\partial\vartheta}{\partial t} \in L^2(0,t_f;(D(-\mathcal{L}_A)^{3/4})'). \tag{4.25}$$

Next we study the regularity of the solution c as the regularity of ϑ is discussed in Theorem 23. It is easy to see that $\|f\| = \|\nabla f\|_{L^2} + \|f\|_{L^2}$ is a norm on $H^2(\Omega)$. Let $\|f\|_{H^2}$ be the standard H^2 norm on Sobolev space $H^2(\Omega)$. Then these two norms $\|f\|$ and $\|f\|_{H^2}$ are equivalent for $f \in H^2(\Omega)$.

That is, there exist positive constants A_1 and A_2 such that

$$A_1\|f\|_{H^2} \leq \|f\| \leq A_2\|f\|_{H^2}, \quad f \in H^2(\Omega).$$

From (4.7) and the boundedness of ϑ in (4.23), we have

$$\begin{aligned} A_1^2\|c\|_{H^2}^2 \leq \|c\|^2 &= \|\nabla c\|_{L^2}^2 + \|c\|_{L^2}^2 \\ &= -\int_\Omega (\Delta c)c + \int_\Omega c^2 = \int_\Omega (\vartheta + \bar{\theta})c \\ &\leq \|\vartheta + \bar{\theta}\|_{L^2}\|c\|_{L^2} \leq \|\vartheta + \bar{\theta}\|_{L^2}\|c\|_{H^2} \end{aligned}$$

It follows that $A_1^2\|c\|_{H^2} \leq \|\vartheta + \bar{\theta}\|_{L^2} \leq \|\vartheta\|_{L^2} + \|\bar{\theta}\|_{L^2}$. Thus, $c \in H^2(\Omega)$.

4.1.1 Existence of an optimal control

With Theorem 23 at our disposal, we are in a position to show the existence of optimal control to problem (P) subject to (4.6)–(4.10).

Definition 3. *Let $\vartheta_0 \in L^\infty(\Omega)$, $u \in U_{ad}$, and $\mathbf{v} \in L^\infty(\Omega)$ be divergence free with $\mathbf{v} \cdot n|_\Gamma = 0$. ϑ is said to be a weak solution of system* (4.6)–(4.10)*, if θ satisfies*

$$(\frac{\partial \vartheta}{\partial t}, \phi) + (\nabla\vartheta, \nabla\phi) - u(\boldsymbol{v}\vartheta, \nabla\phi) - ((\vartheta + \bar{\theta})\nabla c, \nabla\phi) = 0, \quad \forall\phi \in H^1(\Omega), \tag{4.26}$$

in the distribution sense on $(0, t_f)$.

Theorem 24. *Let $\vartheta_0 \in L^\infty(\Omega)$. Assume that $\mathbf{v} \in L^\infty(\Omega)$ is divergence-free with $\mathbf{v} \cdot n|_\Gamma = 0$, which is chosen so that the rapid decay property* (4.16) *holds. Then there exists an optimal solution $u(t) \in U_{ad}$ to the problem (P).*

Proof. To show the existence of an optimal solution, we employ the direct method. Since $J \geq 0$ is bounded from below, we may choose a minimizing sequence $\{u_m\} \subset L^2(0, t_f)$ such that

$$\lim_{m\to\infty} J(u_m) = \inf_{u\in L^2(0,t_f)} J(u).$$

By the definition of J, the sequence $\{u_m\}$ is uniformly bounded in U_{ad}, and hence there exists a weakly convergent subsequence, still denoted by $\{u_m\}$, such that

$$u_m \to u^* \quad \text{weakly in} \quad L^2(0, t_f).$$

For $\vartheta_0 \in L^\infty(\Omega)$, based on Theorem 23 there exists a corresponding sequence $\{\vartheta_m\}$ bounded in $C([0, t_f]; L^2(\Omega)) \cap L^2(0, t_f; H^1(\Omega))$. Together with (4.25) we may extract a subsequence, still denoted by $\{\theta_m\}$, such that

$$\vartheta_m \to \vartheta^* \quad \text{weakly in} \quad L^2(0, t_f; H^1(\Omega)) \tag{4.27}$$

$$\frac{\partial \vartheta_m}{\partial t} \to \frac{\partial \vartheta^*}{\partial t} \quad \text{weakly in} \quad L^2(0, t_f; (D(-\mathcal{L}_A)^{3/4})'). \tag{4.28}$$

By Aubin-Lions lemma, (4.27)–(4.28) indicate that

$$\vartheta_m \to \vartheta^* \quad \text{strongly in} \quad L^2(0, t_f; L^2(\Omega)). \tag{4.29}$$

It remains to show that ϑ^* is the solution corresponding to u^* based on Definition 3. Note that u_m and ϑ_m satisfy

$$(\frac{\partial \vartheta_m}{\partial t}, \phi) + (\nabla \vartheta_m, \nabla \phi) - u_m(\mathbf{v}\vartheta_m, \nabla \phi) - ((\vartheta_m + \bar{\theta})\nabla c_m, \nabla \phi) = 0, \quad \forall \phi \in H^1(\Omega). \tag{4.30}$$

Let ψ be a continuously differentiable function on $[0, t_f]$ with $\psi(t_f) = 0$. For each $\phi \in H^1(\Omega)$, we multiply (4.30) by ψ and integrate by parts. After integrating the first term by parts, we get

$$\begin{aligned} &- \int_0^{t_f} (\vartheta_m, \phi\dot{\psi})\, dt + \int_0^{t_f} (\nabla \vartheta_m, \nabla \phi)\psi\, dt - \int_0^{t_f} u_m(\mathbf{v}\theta_m, \nabla \phi)\psi\, dt \\ &\quad - \int_0^{t_f} (\vartheta_m \nabla c_m, \nabla \phi)\psi\, dt - \int_0^{t_f} (\bar{\theta}\nabla c_m, \nabla \phi)\psi\, dt = (\vartheta_0, \phi\psi(0)). \end{aligned} \tag{4.31}$$

Since $\phi\dot{\psi} \in L^2(0, T; L^2(\Omega))$ and $\nabla\phi \in L^2(\Omega)$, it is straightforward to pass to the limit in the first two terms and the last term of the left hand side of (4.31) with the help of (4.27). To estimate the second term, using the convergence results (4.27)–(4.29) we have

$$\begin{aligned} &| \int_0^T (\int_\Omega u_m \mathbf{v}\vartheta_m \cdot \nabla\phi\, dx dt - \int_0^T \int_\Omega u^* \mathbf{v}\vartheta^* \cdot \nabla\phi\, dx)\psi(t) dt| \\ &\quad \le | \int_0^T (\int_\Omega u_m \mathbf{v}\vartheta_m \cdot \nabla\phi - u_m \mathbf{v}\vartheta^* \cdot \nabla\phi\, dx)\psi(t) dt| \\ &\qquad + | \int_0^T (\int_\Omega u_m \mathbf{v}\vartheta^* \cdot \nabla\phi - u\mathbf{v}\vartheta^* \cdot \nabla\phi\, dx)\psi(t) dt| \\ &\quad \le | \int_0^T u_m (\int_\Omega \mathbf{v}(\vartheta_m - \vartheta) \cdot \nabla\phi\, dx)\psi(t) dt + | \int_0^T (u - u_m)(\int_\Omega \mathbf{v}\vartheta^* \cdot \nabla\phi\, dx)\psi(t)\, dt| \\ &\quad \le \|u_m\|_{L^\infty(0,t_f)} \|\mathbf{v}\|_{L^\infty} \|\vartheta_m - \vartheta^*\|_{L^2(0,t_f;L^2(\Omega))} \|\nabla\phi\|_{L^2} \|\psi\|_{L^2(0,t_f)} \\ &\qquad + | \int_0^T (u^* - u_m)(\int_\Omega \mathbf{v}\vartheta^* \cdot \nabla\phi\, dx)\psi(t)\, dt| \to 0, \end{aligned}$$

where the last term converges to zero is because

$$\sup_{t\in[0,t_f]}\int_\Omega \mathbf{v}\vartheta^*\cdot\nabla\phi\,dx \le \sup_{t\in[0,t_f]}\|\mathbf{v}\|_{L^\infty}\|\vartheta^*\|_{L^2}\|\nabla\phi\|_{L^2} \le \|\mathbf{v}\|_{L^\infty}\|\vartheta^*\|_{L^\infty(0,t_f;L^2(\Omega))}\|\nabla\phi\|_{L^2},$$

so $\int_\Omega \mathbf{v}\vartheta^*\cdot\nabla\phi\,dx \in L^\infty(0,t_f)$ and $(\int_\Omega \mathbf{v}\vartheta^*\cdot\nabla\phi\,dx)\psi(t) \in L^2(0,t_f)$. Due to the weak convergence of u_m, the last terms converges to zero. Moreover,

$$\begin{aligned}
&|\int_0^{t_f}[(\vartheta_m\nabla c_m,\nabla\phi)-(\vartheta\nabla c,\nabla\phi)]\psi(t)\,dt| \\
&\quad= |\int_0^{t_f}(\int_\Omega \vartheta_m(\nabla c_m-\nabla c)\cdot\nabla\phi\,dx+\int_\Omega(\vartheta_m-\vartheta)\nabla c\cdot\nabla\phi\,dx)\psi(t)\,dt| \\
&\quad\le \|\vartheta_m\|_{L^2(0,t_f;H^1(\Omega))}\|\nabla c_m-\nabla c\|_{L^2(0,t_f;H^1(\Omega))}\|\nabla\phi\|_{L^2}\|\psi\|_{L^\infty(0,t_f)} \\
&\qquad+|\int_0^{t_f}(\int_\Omega(\vartheta_m-\vartheta)\nabla c\cdot\nabla\phi\,dx)\psi(t)\,dt| \to 0.
\end{aligned} \tag{4.32}$$

where for the last term we have $\nabla c\cdot\nabla\phi\psi(t) \in L^2(0,t_f;(H^1(\Omega))')$. In fact, for any $g \in L^2(0,t_f;H^1(\Omega))$

$$\begin{aligned}
|\int_0^{t_f}\int_\Omega \nabla c\cdot\nabla\phi\psi(t)g\,dx\,dt| &\le \int_0^{t_f}\|\nabla c\|_{H^1}\|\nabla\phi\|_{L^2}\|g\|_{H^1}|\psi(t)|\,dt \\
&\le \|\nabla c\|_{L^2(0,t_f;H^1(\Omega))}\|\nabla\phi\|_{L^2}\|g\|_{L^2(0,t_f;H^1(\Omega))}\|\psi\|_{L^\infty(0,t_f)},
\end{aligned}$$

which follows

$$\begin{aligned}
\|\nabla c\cdot\nabla\phi\psi\|_{L^2(0,t_f;(H^1(\Omega))')} &\le \|\nabla c\|_{L^2(0,t_f;H^1(\Omega))}\|\nabla\phi\|_{L^2}\|\psi\|_{L^\infty(0,t_f)} \\
&\le C\|\vartheta\|_{L^2(0,t_f;L^2(\Omega))}\|\nabla\phi\|_{L^2}\|\psi\|_{L^\infty(0,t_f)}.
\end{aligned}$$

By (4.27), the last term of (4.32) converges to zero. Therefore, ϑ^* is the solution corresponding to u^*.

Finally, using the weakly lower semicontinuity property of norms in J yields

$$J(u^*) \leq \varliminf_{m\to\infty} J(u_m) = \inf_{u\in U_{\text{ad}}} J(u),$$

which indicates that u^* is an optimal solution to problem (P). □

4.2 First-order optimality conditions

Since (P) is non-convex, we will deal with local solutions. We now derive the first-order necessary optimality conditions for problem (P) by using a variational inequality (e.g. [Lio71]), that is, if J is Gâteaux differentiable with respect to u and g is an optimal solution of problem (P), then

$$J'(u) \cdot (h - u) \geq 0, \quad h \in U_{ad}. \tag{4.33}$$

A direct calculation follows that

$$J'(u) \cdot h = \alpha(\vartheta(t_f), z) + \beta \int_0^{t_f} (\vartheta, z)\, dt + \gamma \int_0^{t_f} uh\, dt. \tag{4.34}$$

To justify that J is indeed Gâteaux differentiable with respect to u, it suffices to show that ϑ is Gâteaux differentiable with respect to u.

Lemma 9. *Let $z = \vartheta'(u) \cdot h$ for $h \in U_{ad}$, be the first variation of ϑ. Then z satisfies $\int_\Omega z\, dx = 0$ and*

$$\frac{\partial z}{\partial t} = \Delta z - u\mathbf{v} \cdot \nabla z - h\mathbf{v} \cdot \nabla \vartheta - \nabla \cdot (z \nabla c) - \nabla \cdot ((\vartheta + \bar{\theta}) \nabla \mathcal{A}^{-1} z), \tag{4.35}$$

$$\frac{\partial z}{\partial n}\Big|_\Gamma = 0, \tag{4.36}$$

with initial condition

$$z(x, 0) = 0. \tag{4.37}$$

Moreover, there exists a unique solution to (4.35)–(4.37) *and for any $0 < t_f < \infty$,*

$$z \in C([0, t_f]; X) \cap L^2(0, t_f; H^1(\Omega)). \tag{4.38}$$

Proof. Applying an L^2-estimate together with Poincaré and Hölder's inequalities and the boundary conditions in (4.9) and (4.36) yields

$$\begin{aligned}
\frac{1}{2}\frac{d\|z\|_{L^2}^2}{dt} + \|\nabla z\|_{L^2}^2 =& -\frac{1}{2}u\int_\Omega \mathbf{v}\cdot\nabla z^2\,dx - \int_\Omega h\mathbf{v}\cdot\nabla\vartheta z\,dx - \int_\Gamma z\nabla c\cdot nz\,dx + \int_\Omega z\nabla c\cdot\nabla z\,dx \\
&- \int_\Gamma((\vartheta+\bar\theta)\nabla\mathcal{A}^{-1}z)\cdot nz\,dx + \int_\Omega(\vartheta+\bar\theta)\nabla\mathcal{A}^{-1}z\cdot\nabla z\,dx \\
=& h\int_\Omega \vartheta\mathbf{v}\cdot\nabla z\,dx + \int_\Omega z\nabla c\cdot\nabla z\,dx + \int_\Omega(\vartheta+\bar\theta)\nabla\mathcal{A}^{-1}z\cdot\nabla z\,dx \\
\le& |h|_{L^\infty}\|\vartheta\|_{L^2}\|\mathbf{v}\|_{L^\infty}\|\nabla z\|_{L^2} + \|z\|_{L^4}\|\nabla c\|_{L^4}\|\nabla z\|_{L^2} \\
&+ \|\vartheta+\bar\theta\|_{L^4}\|\nabla\mathcal{A}^{-1}z\|_{L^4}\|\nabla z\|_{L^2} \\
\le& C|h|_{L^\infty}^2\|\vartheta\|_{L^2}^2\|\mathbf{v}\|_{L^\infty}^2 + \frac{1}{8}\|\nabla z\|_{L^2}^2 + C\|z\|_{L^2}^{1-d/4}\|\nabla z\|_{L^2}^{d/4}\|\nabla\mathcal{A}^{-1}\vartheta\|_{H^1}^2\|\nabla z\|_{L^2} + \frac{1}{8}\|\nabla z\|_{L^2}^2 \qquad (4.39)\\
&+ C\|\vartheta+\bar\theta\|_{H^1}^2\|\nabla\mathcal{A}^{-1}z\|_{H^1}^2 + \frac{1}{8}\|\nabla z\|_{L^2}^2. \qquad (4.40)
\end{aligned}$$

For the second term in (4.39) we use Young's inequality and obtain

$$\begin{aligned}
\|z\|_{L^2}^{1-d/4}\|\nabla z\|_{L^2}^{d/4}\|\nabla\mathcal{A}^{-1}\vartheta\|_{H^1}^2\|\nabla z\|_{L^2} &\le C\|z\|_{L^2}^2\|\nabla\mathcal{A}^{-1}\vartheta\|_{H^1}^{\frac{16}{4-d}} + \frac{1}{8}\|\nabla z\|_{L^2}^2 \\
&\le C\|z\|_{L^2}^2\|\vartheta\|_{L^2}^{\frac{16}{4-d}} + \frac{1}{8}\|\nabla z\|_{L^2}^2. \qquad (4.41)
\end{aligned}$$

Combining (4.39) with (4.40) and (4.41) follows

$$\frac{d\|z\|_{L^2}^2}{dt} + \|\nabla z\|_{L^2}^2 \le C|h|_{L^\infty}^2\|\vartheta\|_{L^2}^2\|\mathbf{v}\|_{L^\infty}^2 + C\|z\|_{L^2}^2\|\vartheta\|_{L^2}^{\frac{16}{4-d}} + C(\|\vartheta\|_{H^1}^2 + |\bar\theta|^2)\|z\|_{L^2}^2,$$

and hence by Grönwall's inequality we obtain

$$\sup_{t\in[0,t_f]} \|z\|_{L^2}^2 \leq C\|\mathbf{v}\|_{L^\infty}^2 \int_0^{t_f} |h|_{L^\infty}^2 \|\vartheta\|_{L^2}^2 \, dt \cdot e^{C\int_0^{t_f} (\|\vartheta\|_{L^2}^{\frac{16}{4-d}} + \|\vartheta\|_{H^1}^2 + |\bar{\theta}|^2)\, dt} \tag{4.42}$$

It is clear that by (4.41)–(4.42),

$$\int_0^{t_f} \|\nabla z\|_{L^2}^2 < \infty. \tag{4.43}$$

With the help of *a priori* estimates (4.42)–(4.43) and the Galerkin approximation, one can show that there exists a unique solution to (4.35)–(4.37). In other words, z is well-defined and this completes the proof. □

Lemma 9 indicates that ϑ is Gâteaux differentiable with respect to $u \in U_{ad}$, so is J.

Theorem 25. *Assume that $\vartheta_0 \in L^\infty(\Omega)$. Let $\mathbf{v} \in L^\infty(\Omega)$ be divergence-free and $\mathbf{v}\cdot n|_\Gamma = 0$. If $u(t)$ is the optimal solution to problem (P) and ϑ is the corresponding solution to the state equations.*

Then there exists an adjoint state ρ such that the optimal triplet (u, ϑ, ρ) satisfies

$$\textbf{\textit{State Equations}} \quad \begin{cases} \frac{\partial \vartheta}{\partial t} = \Delta \vartheta - u\mathbf{v} \cdot \nabla \vartheta - \nabla \cdot ((\vartheta + \bar{\vartheta})\nabla c), \\ -\Delta c + c = \vartheta + \bar{\theta}, \\ \frac{\partial \theta}{\partial n}\Big|_{\Gamma} = \frac{\partial c}{\partial n}\Big|_{\Gamma} = 0, \\ \theta(0) = \theta_0, \end{cases} \tag{4.44}$$

$$\textbf{\textit{Adjoint Equations}} \quad \begin{cases} -\frac{\partial \rho}{\partial t} = \Delta \rho + u\mathbf{v} \cdot \nabla \rho + \chi \nabla c \cdot \nabla \rho + \mathcal{A}^{-1}(\nabla \cdot (\theta \nabla \rho)) + \beta \vartheta, \\ \frac{\partial \rho}{\partial n}\Big|_{\Gamma} = 0, \\ \rho(t_f) = \alpha \vartheta(t_f), \end{cases} \tag{4.45}$$

$$\textbf{\textit{Optimality condition:}} \quad u^{opt}(t) = \mathbb{P}_{[\underline{u}, \overline{u}]}\left(-\frac{1}{\gamma}\int_{\Omega}(\theta \nabla \rho) \cdot \mathbf{v}\, dx\right), \tag{4.46}$$

where for real numbers $c \leq d$, $\mathbb{P}_{[c,d]}$ denotes the projection of $\mathbb{R}$ onto $[c, d]$, that is, $\mathbb{P}_{[c,d]}(f) := \min\{d, \max\{c, f\}\}$.

Proof. To define the adjoint state, we take the inner product of (4.35) with ρ

$$\begin{aligned} \int_0^{t_f} (\frac{\partial z}{\partial t}, \rho)\, dt = &\int_0^{t_f} (\Delta z, \rho)\, dt - \int_0^{t_f} u(\mathbf{v} \cdot \nabla z, \rho)\, dt - \int_0^{t_f} (h\mathbf{v} \cdot \nabla \theta, \rho)\, dt \\ &- \int_0^{t_f} (\nabla \cdot (z \nabla c), \rho)\, dt - \int_0^{t_f} (\nabla \cdot ((\vartheta + \bar{\theta})\nabla \mathcal{A}^{-1} z), \rho)\, dt, \end{aligned} \tag{4.47}$$

which follows

$$
\begin{aligned}
(\rho(t_f), z(t_f)) - (\rho(0), z(0)) + \int_0^{t_f} (-\frac{\partial \rho}{\partial t}, z)\, dt & \\
= \int_0^{t_f} (\Delta\rho, z)\, dt + \int_0^{t_f} u(\mathbf{v}\cdot\nabla\rho, z)\, dt + \int_0^{t_f} (\theta\nabla\rho, \mathbf{v}) h\, dt & \\
+ \int_0^{t_f} (z\nabla c, \nabla\rho)\, dt + \int_0^{t_f} (z, \mathcal{A}^{-1}(\nabla\cdot((\vartheta+\bar{\theta})\nabla\rho)))\, dt. & \qquad (4.48)
\end{aligned}
$$

Let ρ satisfy

$$
-\frac{\partial \rho}{\partial t} = \Delta\rho + \mathbf{v}\cdot\nabla\rho + \chi\nabla c\cdot\nabla\rho + \mathcal{A}^{-1}(\nabla\cdot((\vartheta+\bar{\theta})\nabla\rho)) + \beta\vartheta, \qquad (4.49)
$$

$$
\frac{\partial \rho}{\partial n}|_{\partial\Omega} = 0, \qquad (4.50)
$$

with final time condition

$$
\rho(t_f) = \alpha\vartheta(t_f). \qquad (4.51)
$$

Then combining (4.51) with (4.49) and (4.50) follows

$$
(\alpha\vartheta(t_f), z(t_f)) = \int_0^{t_f} (\theta\nabla\rho, \mathbf{v}) h\, dt - \int_0^{t_f} (\beta\vartheta, z)\, dt. \qquad (4.52)
$$

As a result, if u is an optimal solution, then

$$
J'(u)\cdot h = \int_0^{t_f} (\,\theta\nabla\rho, \mathbf{v}) h\, dt + \gamma \int_0^{t_f} uh\, dt \geq 0, \qquad (4.53)
$$

for any $h \in U_{ad}$, which establishes (4.46). □

4.2.1 Control of cellular flows in rectangle-like domains

In real life applications, however, many flows are not necessarily relaxation-enhancing, i.e., (4.20) may not be satisfied. The semigroup generated by the advection-diffusion operator can still have the rapid decay property. As shown in (e.g. [IXZ21; Hu22]), for the velocity field generated by cellular flows in rectangle-like domains (rectangles ($d = 2$) and parallelepipeds ($d = 3$)), rescaling both the cell size and the flow amplitude can make $\Psi(\mathcal{L}_A)$ arbitrarily large, and hence the rapid decay property of the semigroup $e^{\mathcal{L}_A t}$ can be achieved. It is important to point out that in rectangle-like domains our main theorems in this work still hold.

To demonstrate the idea, we consider the following prototypical example of a 2D cellular flow for our numerical tests

$$\mathbf{v}(x, y) = \nabla^{\perp} \sin(2\pi x) \sin(2\pi y) = 2\pi \begin{bmatrix} -\sin(2\pi x)\cos(2\pi y) \\ \cos(2\pi x)\sin(2\pi y) \end{bmatrix} \tag{4.54}$$

in a two-dimensional domain $\Omega = (0, 1)^2$. For $d = 3$, one can utilize the cubic cells given by (e.g. [Bis60; IXZ21; RZ07]) Let $v_{i_N}(x) = v_i(Nx)$ for $x \in \Omega$, $N \in \mathbb{N}^+$, and $\mathbf{v}_N = (v_{1_N}, \ldots, v_{d_N})$, $d = 2, 3$. According to (4.54), the rescaled cellular flow velocity $\mathbf{v}_N$ is still sufficiently smooth and periodic, yet with higher frequency compared to $\mathbf{v}$. Now define

$$\mathcal{L}_N = \Delta - N\mathbf{v}_N(x) \cdot \nabla$$

with $D(\mathcal{L}_N) = D(\mathcal{L}_1)$. One can show that

$$\Psi(\mathcal{L}_N) = N^2\Psi(\mathcal{L}_1). \tag{4.55}$$

The detailed proof is given by [Hu22, Section 3]. Thus,

$$\|e^{\mathcal{L}_N t}\|_{\mathscr{L}(X)} \leq M_0 e^{-\Psi(\mathcal{L}_N)t} = M_0 e^{-N^2\Psi(\mathcal{L}_1)t}, \quad t \geq 0,$$

which indicates that the decay rate of the semigroup $e^{\mathcal{L}_N t}$ can be made arbitrarily fast if N is sufficiently large. In other words, for a given initial condition θ_0, as long as the amplitude $N(\theta_0)$ and the frequency $2\pi N(\theta_0)$ of the cellular flows are large enough, the system is well-posed and exponentially stable. For a fixed flow frequency, the objective of the optimal control design using cellular flows is to determine an optimal time-dependent flow amplitude $u(t)$ to regulate the strength of the flow. The existence of such a solution follows from Theorem 24. It is clear that the upper bound of $u(t)$ can be set as $N(\theta_0)$. Numerical experiments based on cellular flows will be presented in the next section to demonstrate our ideas and designs.

4.3 Numerical Implementation

In this section, we discuss our numerical experiments aimed at suppressing the singularity of the PKS system in rectangular domains using the 2D cellular flows introduced in section 4.2.1. To solve the optimality system (4.44)–(4.46) over time, we employ the spline collocation method in conjunction with the Euler forward method. The bivariate spline functions are utilized to approximate the system

based on the weak formulation for parabolic-elliptic PDEs. Interested readers can refer to [ALW06; GLS15; LL23; LL22; LS07a] for a detailed study.

First, we provide some examples of the PKS system without flow advection, along with the exact solution, to demonstrate the accuracy of our approach. We calculate the root mean square error (RMSE) to evaluate the quality of our results. Additionally, we present examples of the PKS system without flow advection and showcase the corresponding numerical results.

Finally, we utilize the Gaussian function, which experiences unbounded growth without flow advection, to determine the optimal control u(t) capable of suppressing the blow-up phenomenon.

4.3.1 A Spline-Based Collocation Method for the Keller Segel Equation without Flow Advection

This section presents a spline collocation method for solving the following system of partial differential equations (PDEs), known as the PKS system:

$$\frac{\partial \theta}{\partial t} = \Delta\theta - \nabla \cdot (\theta \chi \nabla c)) + f \quad \text{in} \quad \Omega, \tag{4.56}$$

$$-\Delta c + c = \theta \quad \text{in} \quad \Omega, \tag{4.57}$$

$$\frac{\partial \theta}{\partial n} = \frac{\partial c}{\partial n} = 0 \quad \text{on} \quad \Gamma \tag{4.58}$$

$$\theta(x, 0) = \theta_0(x) \quad \text{in} \quad \Omega, \tag{4.59}$$

$$\int_\Omega \theta dx = \int_\Omega \theta_0 dx \tag{4.60}$$

where Ω is a polygonal domain in $\mathbb{R}^2$, f is a nonzero force term, θ and c are the unknown functions, and χ is a given coefficient. To solve the equations (4.56)–(4.59), we propose a computational scheme that uses the Euler forward method and consists of the following steps.

Step 1: Initialization. Given an integer N_T, a terminal time $t_f > 0$, we define a constant time step $\triangle t := \frac{t_f}{N_T}$ and $t_k := k\triangle t$, for $k = 0, \ldots, N_T - 1$. We start by initializing the concentration field $c_0(x)$ using the initial density function $\theta_0(x)$ through the following PDE:

$$-\Delta c_0(x) + c_0(x) = \theta_0(x), \qquad c_0(x) \geq 0, \quad \frac{\partial c_0}{\partial \nu}|_\Gamma = 0. \tag{4.61}$$

Step 2: Time Stepping. We use a forward Euler method to approximate the time derivative of the density function:

$$\frac{\theta_k - \theta_{k-1}}{\triangle t} = \Delta\theta_{k-1} - \nabla \cdot (\theta_{k-1}\nabla c_{k-1}) + f_{k-1}, \text{ in } \Omega \tag{4.62}$$

$$\theta_k \geq 0, \text{ in } \Omega \tag{4.63}$$

$$\frac{\partial \theta_k}{\partial \nu} = 0, \text{ on } \partial\Omega, \tag{4.64}$$

where $f_{k-1}(x) = f(t_{k-1}, x)$.

Step 3: Calculation of Concentration. We calculate $c_k = c(t_k, x)$ at time step t_k, based on the value of θ_k, using the following PDE on the domain Ω:

$$-\Delta c_k(x) + c_k(x) = \theta_k, c_k(x) \geq 0, \text{ and } \frac{\partial c_k}{\partial \nu}|_{\partial\Omega} = 0 \tag{4.65}$$

Step 4: By repeating Steps 2 and 3 for each $k = 1, \cdots, N_T$, we can obtain the numerical solution (θ_k, c_k) of the PKS system (4.56)–(4.58) at each time step t_k.

Let us focus on how to solve the PDE in (4.61) and (4.65). We first choose a set of domain points $\{\xi_i\}_{i=1,\cdots,N}$ as collocation points and then use the spline basis functions $\mathcal{B}_\alpha^t$ just like the previous sections. Using these functions, we find the spline solution $\theta_k = \sum_{t\in\triangle}\sum_{|\alpha|=D} a_\alpha^{t,k}\mathcal{B}_\alpha^t$ in spline space $S_D^r(\triangle)$, $c_k = \sum_{t\in\triangle}\sum_{|\alpha|=D} b_\alpha^{t,k}\mathcal{B}_\alpha^t$ in the same spline space $S_D^r(\triangle)$ with the coefficient vector $\mathbf{a}^k \geq 0, \mathbf{b}^k \geq 0$, respectively, satisfying

$$-\Delta c_k(\xi_i) + c_k(\xi_i) = \theta_k(\xi_i), \text{and } \frac{\partial c_k(\xi_i)}{\partial \nu}|_{\partial\Omega} = 0,\ k = 0, 1, \cdots N_T,$$

where $\{\xi_i\}_{i=1,\cdots,N} \in \mathcal{D}_{D,\triangle}$ are the domain points of $\triangle$ of degree D. We can rewrite the above equation as the matrix equation:

$$(-K + MV)\mathbf{b}^k = MV\mathbf{a}^k, \text{and } B\mathbf{b}^k = 0$$

where $K = MxxV + MyyV = \left[\partial_{xx}\mathcal{B}_\alpha^t(x_i, y_i)\right] + \left[\partial_{yy}\mathcal{B}_\alpha^t(x_i, y_i)\right]$, $MV = \left[\mathcal{B}_\alpha^t(x_i, y_i)\right]$ and $B\mathbf{b}^k = 0$ is associated with the boundary condition. Also, we add the smoothness conditions in terms of smooth matrix equation: $H\mathbf{b}^k = 0$. Hence, our spline collocation method is to find $\mathbf{b}^k$ by solving the following constrained minimization:

$$\min_{\mathbf{b}^k} J(\mathbf{b}^k) = \frac{1}{2}\|(-K + MV)\mathbf{b}^k - MV\mathbf{a}^k\|^2 \tag{4.66}$$

$$\text{subject to } B\mathbf{b}^k = 0, H\mathbf{b}^k = 0, \mathbf{b}^k \geq 0. \tag{4.67}$$

It is easy to see that such a constrained minimization has a unique solution as the feasible set is convex and the minimizing functional is convex. We shall use the iterative method in [**LL22**] to solve the above constrained minimization problem.

Similarly, we can solve the PDE in Step 2, i.e. (4.62) by finding the spline approximation θ_k which satisfies the following collocation equations

$$\begin{cases} \theta_k(\xi_i) = \theta_{k-1}(\xi_i) + \triangle t(\Delta\theta_{k-1}(\xi_i) - \nabla \cdot (\theta_{k-1}(\xi_i)\nabla c_{k-1}(\xi_i)) + f_{k-1}(t_{k-1}, \xi_i)) \\ \dfrac{\partial \theta_k}{\partial \nu}(\xi_i) = 0, \quad \xi_i \in \Gamma \\ \bar{\theta}_k = \bar{\theta}_0, \end{cases} \tag{4.68}$$

where $\{\xi_i\}_{i=1,\cdots,N} \in \mathcal{D}_{D,\triangle}$ are the domain points of $\triangle$ of degree D. That is, we can find the θ_k by solving the following constrained minimization problem:

$$\min_{\mathbf{a}^k} J(\mathbf{a}^k) = \frac{1}{2}\|MV\mathbf{a}^k - \mathbf{f}_u\|^2 \tag{4.69}$$

$$\text{subject to } B\mathbf{a}^k = 0, H\mathbf{a}^k = 0, I\mathbf{a}^k = \bar{\theta}_0, \mathbf{a}^k \geq 0, \tag{4.70}$$

where $I = [\int_\Omega \mathcal{B}_\alpha^t]$ is a row vector, $B, 0$ are from the boundary condition, $\mathbf{f}_u$ is from the right side of the equation (4.68) and H is the matrix of all the smoothness conditions across each interior edge of triangulation $\triangle$. Again, we see that the above-constrained minimization has a unique solution. We use the iterative method in [**LL22**] to solve the above minimization problem.

4.3.2 Numerical Examples for the PKS System without Flow Advection

This section presents numerical results obtained using the spline collocation method for the Keller-Segel equations to demonstrate the accuracy of our method. We calculate the root mean squared errors (RMSE) of the approximate spline solutions θ_s and c_s against the exact solutions θ and c based on 201×201 equally-spaced points over the bounding box of the domain Ω. In particular, we consider the following solutions satisfying (4.56)–(4.58).

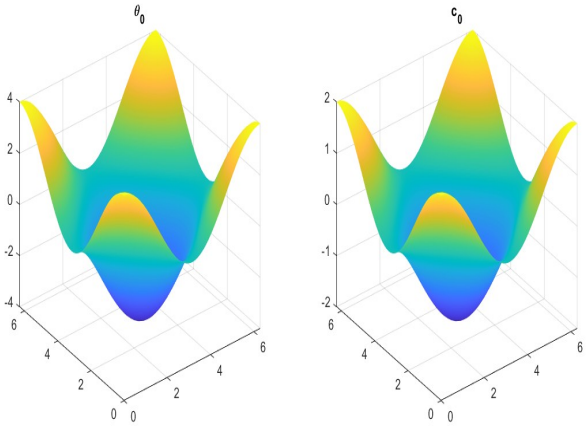
Example 14. *Let* $\Omega = [0,1] \times [-\pi, \pi]^2$*, and let the exact solutions be given by*

$$\begin{cases} \theta &= 2e^{-t}(\cos(x) + \cos(y)) \\ c &= e^{-t}(\cos(x) + \cos(y)) \end{cases}$$

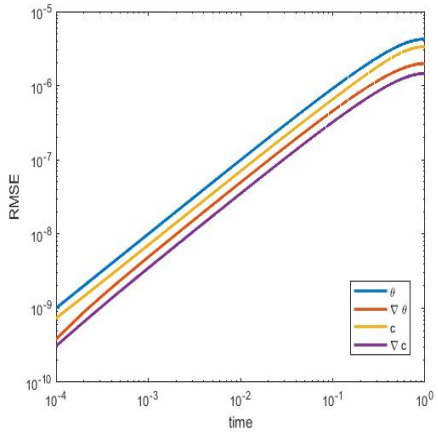
where an appropriate f *is chosen. We choose the terminal time* $t_f = 1$*, time step* $\triangle t = 1e-05$*, degree* $D = 10$*, and spline space parameter* $r = 2$*. The mesh size is set to* $h = \pi/8$*. Figure 4.1a shows the graphs of the exact solutions* θ *and* c *at* $t = 0$*, as well as the numerical solution* c_0 *obtained from Step 1. Figure 4.1b shows the RMSEs of* $\theta - \theta_s$*,* $\nabla(\theta - \theta_s)$*,* $c - c_s$*, and* $\nabla(c - c_s)$*, which are all close to* $1e-09$ *at the beginning and decrease to between* $1e-05$ *and* $1e-09$*. These results demonstrate the excellent performance of our method.*

Example 15. *Next, let* $\Omega = [0,1] \times [-\frac{3\pi}{2}, \frac{3\pi}{2}]^2$*, and the exact solutions for* (4.56)–(4.58) *are chosen as follows:*

$$\begin{cases} \theta &= 2e^{-t}(2\sin(x) + \sin(y)) \\ c &= e^{-t}(2\sin(x) + \sin(y)). \end{cases}$$

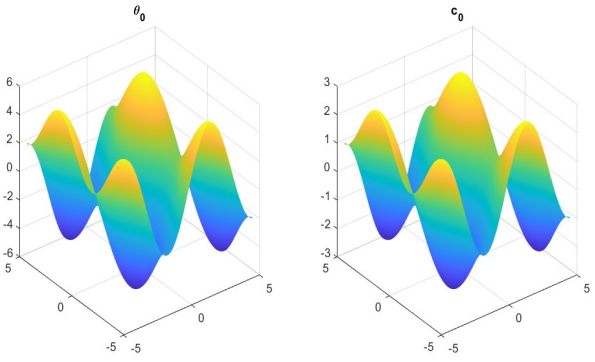


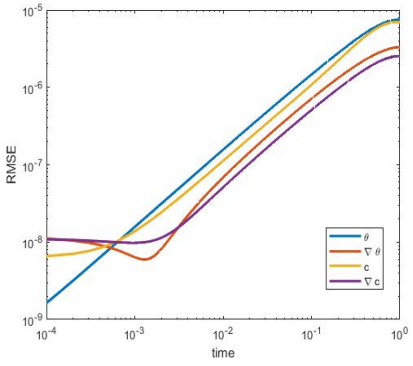
(a) Initial condition of θ (left), numerical solution c_0 from Step 1 (right) at $t = 0$ in Example 14



(b) RMSE for solution θ in Example 14

Let the terminal time $t_f = 1$ *and time step* $\triangle t = 1e-05, D = 10, r = 2$. *Also let the mesh size of triangulation of* $[-\frac{3\pi}{2}, \frac{3\pi}{2}]^2$ *be* $h = \pi/8$. *Then the RMS of* $\theta - \theta_s, \nabla(\theta - \theta_s), c - c_s, \nabla(c - c_s)$ *based on the* 201×201 *equally-spaced points over the domain* $[-\frac{3\pi}{2}, \frac{3\pi}{2}]^2$ *are shown in Figure 4.2. Similar to the results of Example 14, the RMSEs are between* $1e-05$ *and* $1e-09$.





(a)

Figure 4.2: Initial condition of θ (Left), numerical solution c_0 from Step 1 (Middle) at $t = 0$, RMSE for solution θ, c in Example 15

4.3.3 A Spline Based Collocation Method for the Keller Segel Equation with Flow Advection

To validate the accuracy of our algorithms for solving the PKS system without flow advection, we present numerical tests in section 4.3.1. In particular, we verify the validity of the algorithm for conservation of mass and non-negativity of the solutions θ and c. Following the main results in Theorem 25, we implement the optimality system (4.44)–(4.46) using Algorithm 4.1 presented below.

Algorithm 4.1: Adjoint Method for Optimal Control Problem

Given a final time $t_f > 0$, a tolerance $\epsilon > 0$, and constants $\alpha, \beta, \gamma > 0$, for $j = 1, 2, \ldots$, we compute u^j using the following iterations with known u^{j-1}:

- **Step 1**: First, we use the initial function $u^{j-1}(t)$ and the given $\mathbf{v}(x, y)$. Let $\theta(0, x) = \theta_0(x) \geq 0$ and $c_0(x) = c(0, x)$. Solve

$$-\Delta c_0(x) + c_0(x) = \theta_0 \quad \text{and} \quad \left.\frac{\partial c_0}{\partial \nu}\right|_{\Gamma} = 0.$$

Let $u_k = u^{j-1}(t_k)$. Solve $\theta_k = \theta(t_k, x)$ and $c_k = c(t_k, x)$ at each time step $t_k, k = 1, 2, \cdots, n$, where $n = \frac{t_f}{\Delta t}$, from the following equations

$$\frac{\theta_k - \theta_{k-1}}{\triangle t} = \Delta\theta_{k-1} - u_k \mathbf{v} \cdot \nabla\theta_{k-1} - \nabla \cdot (\theta_{k-1} \nabla c_{k-1}),$$
$$\frac{\partial \theta_k}{\partial \nu}\Big|_{\Gamma} = 0,$$

and

$$-\Delta c_k(x) + c_k(x) = \theta_k \text{ and } \frac{\partial c_k}{\partial \nu}\Big|_{\Gamma} = 0.$$

Stop if $|\theta|_\infty > 5,000$ and set $t_f = t_n$.

- **Step 2** : Compute $\rho_n = \alpha(\theta_n - \bar{\theta}_n)$ using $t_f = t_n$ and $w_n = \mathcal{A}^{-1}(\nabla \cdot (\theta_n \nabla \rho_n))$, that is, we solve

$$(-\Delta + I)w_n = \nabla \cdot (\theta_n \nabla \rho_n) \quad \text{and} \quad \frac{\partial w_n}{\partial \nu}\Big|_\Gamma = 0.$$

- **Step 3** : Set $t_f = t_n$. Use ρ_n, θ_n and c_n to solve ρ_{n-1} backward in time from the adjoint equations

$$\begin{aligned} &-\frac{\rho_n - \rho_{n-1}}{\triangle t} = \Delta \rho_n + u_n \mathbf{v} \cdot \nabla \rho_n + \nabla c_n \cdot \nabla \rho_n + w_n + \beta(\theta_n - \bar{\theta}_n), \\ &\frac{\partial \rho_{n-1}}{\partial \nu}\Big|_\Gamma = 0. \end{aligned}$$

- **Step 4** : For given ρ_{n-1} and θ_{n-1}, compute w_{n-1} from

$$(-\Delta + I)w_{n-1} = \nabla \cdot (\theta_{n-1} \nabla \rho_{n-1}).$$

- **Step 5** : Find ρ_{n-2} that satisfies

$$\begin{aligned} &\rho_{n-2} = \rho_{n-1} + \triangle t(\Delta \rho_{n-1} + u_{n-1} \mathbf{v} \cdot \nabla \rho_{n-1} + \nabla c_{n-1} \cdot \nabla \rho_{n-1} + w_{n-1} + \beta(\theta_{n-1} - \bar{\theta}_{n-1})), \\ &\frac{\partial \rho_{n-2}}{\partial \nu}\Big|_\Gamma = 0. \end{aligned}$$

  Repeat **Steps 4–5** to solve $\rho_{n-3}, \cdots, \rho_0$.

- **Step 6** : Compute

$$u_j(t_k) = \mathbb{P}_{[\underline{u}, \overline{u}]}\left(-\frac{1}{\gamma}\int_\Omega (\theta_k \nabla \rho_k)) \cdot \mathbf{v}\, dx\right), \quad k = 0, 1, \ldots, n.$$

  where $\mathbb{P}_{[\underline{u}, \overline{u}]}$ denotes the projection of $\mathbb{R}$ onto $[\underline{u}, \overline{u}]$, that is, $\mathbb{P}_{[\underline{u}, \overline{u}]}(f) := \min\{\overline{u}, \max\{\underline{u}, f\}\}$.

- Stop, if $\frac{|J(u^j) - J(u^{j-1})|}{J(u^{j-1})} < \epsilon$.

We begin with demonstrating the development of singularity in the solution to the PKS system in finite time when there is no flow advection and the initial condition is above a certain threshold. We then present our optimal control design for suppressing such singularity formation. Nagai in [Nag95] considered the system (4.1)–(4.5) without advection in a 2D disk and showed that under the condition $\bar{\theta}_0 > 8\pi$, the radial solution blows up in finite time if $\int_\Omega \theta_0 |x|^2 dx$ is sufficiently small. However, under the condition $\bar{\theta}_0 < 8\pi$, the radial solution exists globally in time. In our numerical simulations, we first consider different Gaussian functions as the initial data as in [Fil06] to demonstrate the density evolution, in response to the influence of a chemoattractant governed by the PKS system (4.1)–(4.5) without flow advection. We set $\Delta t = 1e\text{-}5$ and $h = \frac{1}{8}$ as the time step and mesh size, respectively.

Example 16. *In this numerical example, we investigate the behavior of the advection term in the system. The system is given by:*

$$\frac{\partial \theta}{\partial t} = \Delta\theta - u(t)\boldsymbol{v}_N \cdot \nabla\theta - \nabla \cdot (\theta\chi\nabla c) + f \quad in \quad \Omega,$$

$$-\Delta c + c = \theta \quad in \quad \Omega,$$

$$\nabla \cdot \boldsymbol{v} = 0 \quad in \quad \Omega,$$

$$\frac{\partial \theta}{\partial \nu}\Big|_\Gamma = 0 = \frac{\partial c}{\partial \nu}\Big|_\Gamma$$

where the cellular flow is generated by the vector field:

$$\boldsymbol{v}_N(x, y) = 2\pi \begin{pmatrix} -\sin(2N\pi x)\cos(2N\pi y) \\ \cos(2N\pi x)\sin(2N\pi y) \end{pmatrix}, \quad N \in \mathbb{N}^+, \tag{4.71}$$

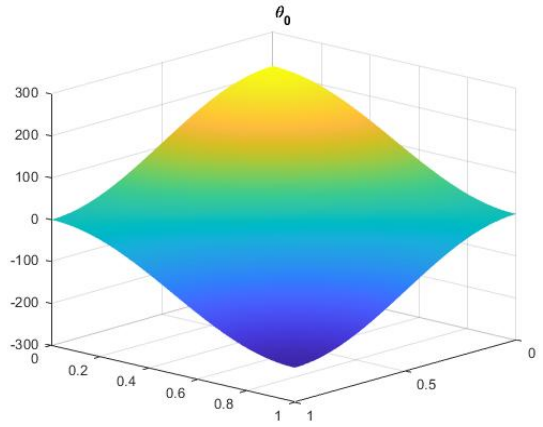
as shown in Figure 4.7 for $N = 1, 2$. *The domain is defined as* $\Omega = [0, 1] \times [0, 1]^2$, *and the exact solutions are given by:*

$$\begin{cases} \theta &= 10(\pi^2 + 1)e^{-t}(\cos(\pi x) + \cos(\pi y)) \\ c &= 10e^{-t}(\cos(\pi x) + \cos(\pi y)) \end{cases}$$

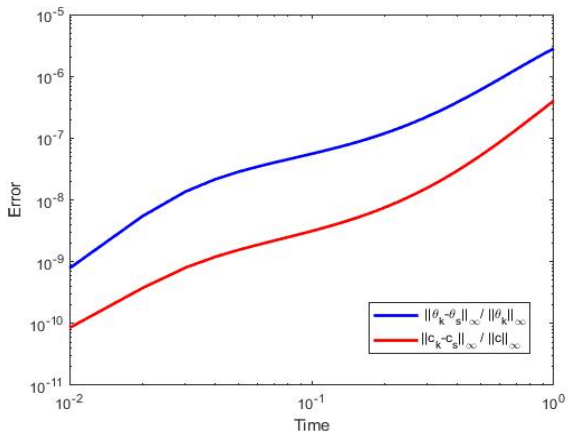
where an appropriate f *is chosen. We perform tests using these exact solutions for various cases, with* u *taking on the values of* $0, 3, \min 4\sin(4\pi t), 3$, *and* $\max 3\cos(3\pi t), -2$ *for each* $N = 1, 2$. *The numerical simulations are carried out with a terminal time of* $t_f = 1$, *a time step of* $\Delta t = 1e-05$, *a degree of* $D = 8$, *and a smoothness parameter of* $r = 2$. *Furthermore, the mesh size is set to* $h = 1/8$.

In Figure 4.3a, we present the graph of the exact solution θ *at* $t = 0$. *The relative errors* $\frac{|\theta - \theta_s|_\infty}{|\theta|_\infty}$ *and* $\frac{|c - c_s|_\infty}{|c|_\infty}$ *are shown in Figure 4.3b when* $u = 0$. *These errors are both close to* $1e-09$ *initially and decrease to values between* $1e-05$ *and* $1e-09$.

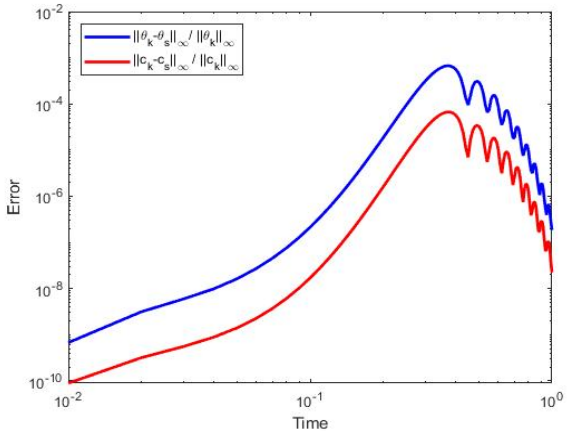
Furthermore, in Figure 4.4a to Figure 4.4f, we display the relative errors for different values of u *and* $N = 1, 2$. *It is observed that the method performs excellently in all cases, and notably, we achieve better results when* $N = 2$ *compared to* $N = 1$.



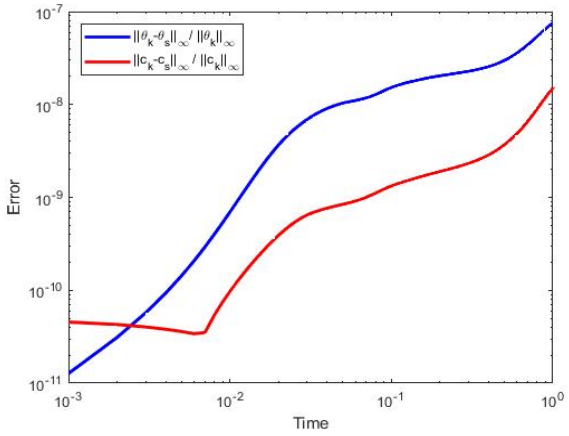
(a) Initial condition of θ at $t = 0$ in Example 16



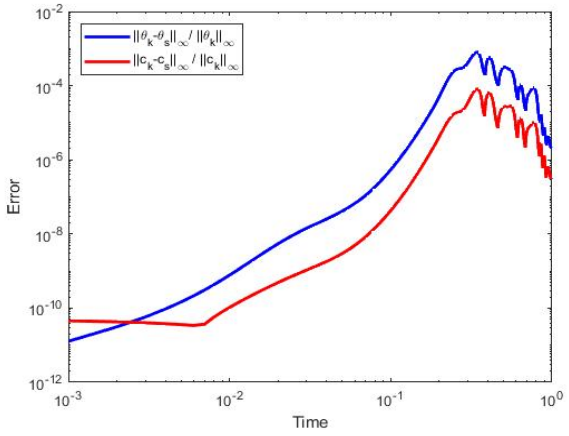
(b) Relative error of θ and c in Example 16 when $u = 0$



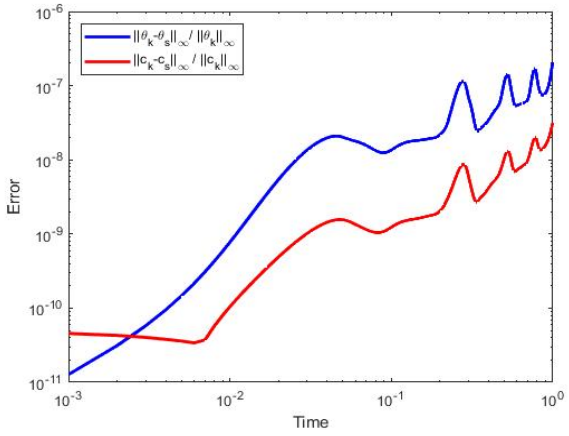
(a) Relative error of θ and c in Example 16 when $u = 3$ and $N = 1$



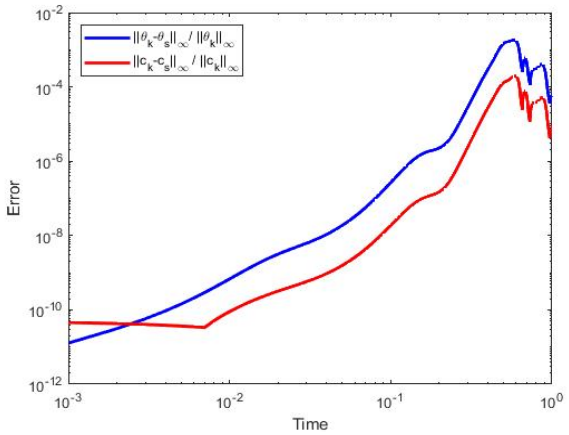
(b) Relative error of θ and c in Example 16 when $u = 3$ and $N = 2$



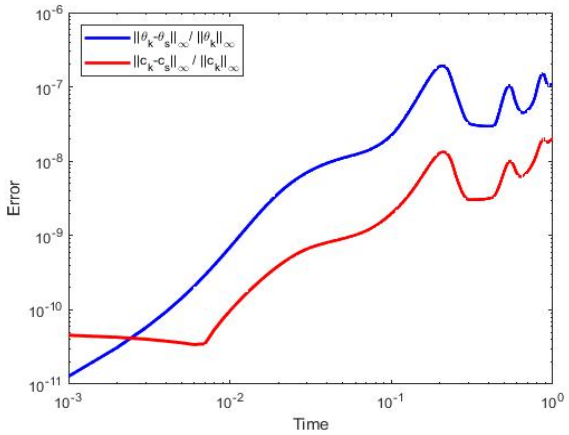
(c) Relative error of θ and c in Example 16 when $u = \min(4\sin(4\pi t), 3)$ and $N = 1$



(d) Relative error of θ and c in Example 16 when $u = \min(4\sin(4\pi t), 3)$ and $N = 2$



(e) Relative error of θ and c in Example 16 when $u = \max(4\cos(14\pi t), -3)$ and $N = 1$

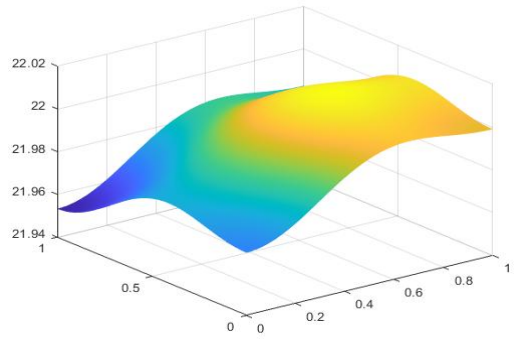


(f) Relative error of θ and c in Example 16 when $u = \max(4\cos(14\pi t), -3)$ and $N = 2$

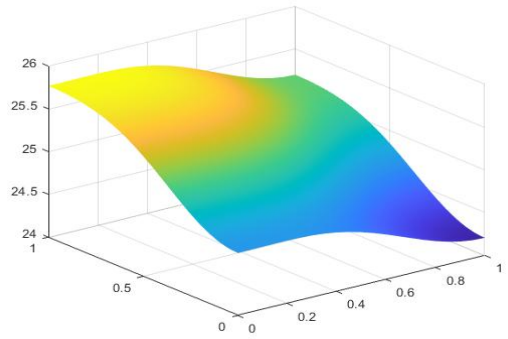
Example 17. *Let* $\Omega = (0,1) \times (0,1)$ *and consider the following Gaussian function:*

$$\theta_0 = \frac{N_0}{2\pi\rho} \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\rho}\right), \quad N_0 \in \mathbb{N}^+, \quad \rho > 0,$$

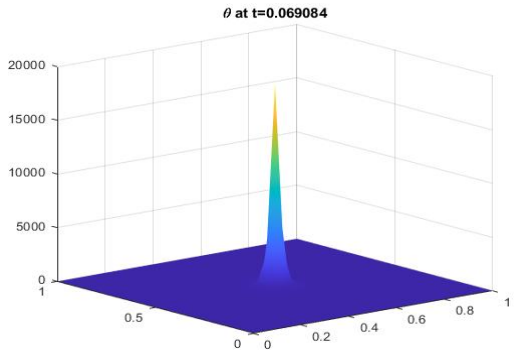
where $(x_0, y_0) = (0.5, 0.5)$. *It is easy to check that* $N_0(1 - e^{-\frac{1}{8\rho}}) < \int_\Omega \theta_0 \, dx < N_0(1 - e^{-\frac{1}{4\rho}})$. *Thus* $\int_\Omega \theta_0 \, dx \approx N_0$ *if* ρ *is small. Set* $\rho = 10^{-2}$ *and the final time* $t_f = 1$. *We test different initial mass by letting* N_0 *be* $7\pi, 8\pi, 9\pi$, *and* 10π, *respectively.*



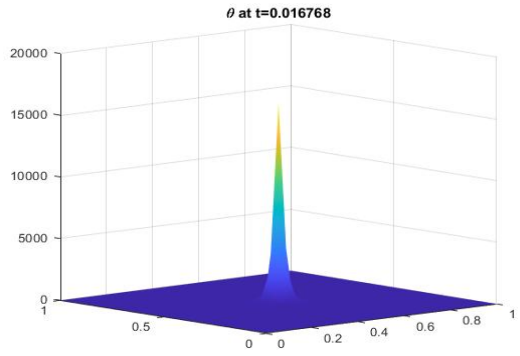
(a) Density θ at $t = 1$ for $N_0 = 7\pi$



(b) Density θ at $t = 1$ for $N_0 = 8\pi$



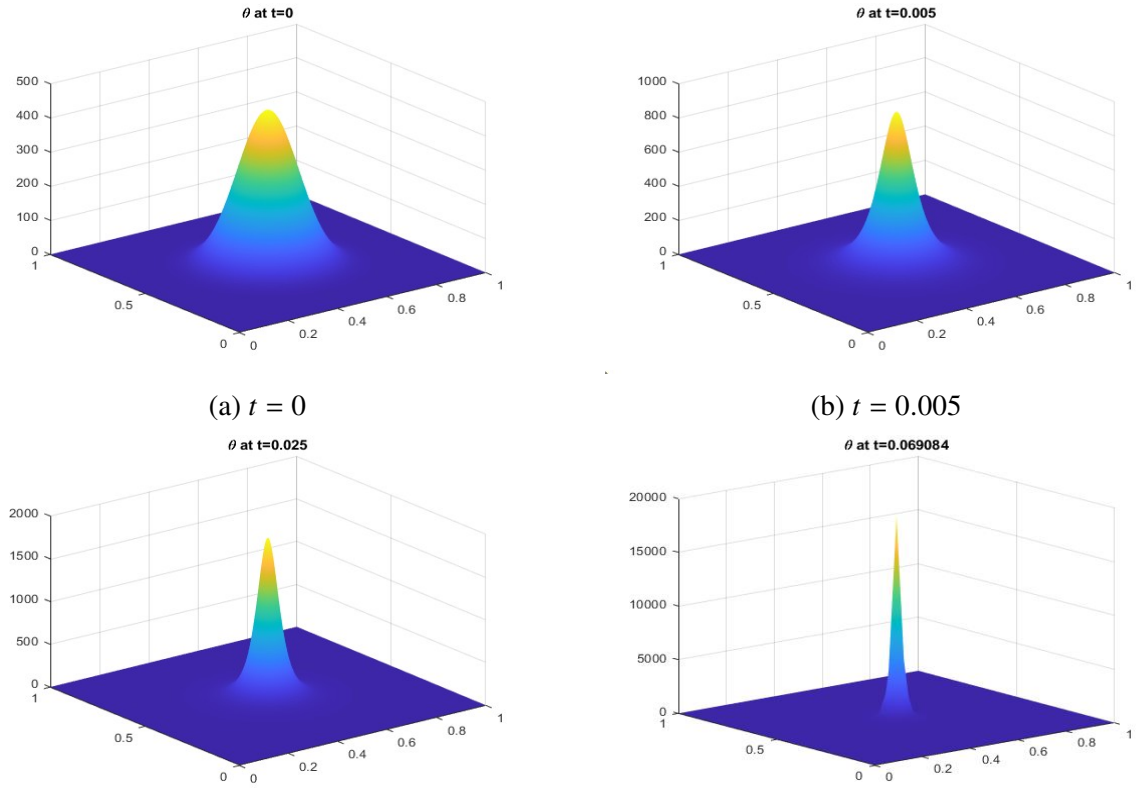
(c) Density θ at $t = 0.069084$ for $N_0 = 9\pi$



(d) Density θ at $t = 0.016768$ for $N_0 = 10\pi$

Figure 4.5: Density θ with $(x_0, y_0) = (0.5, 0.5)$ for various initial mass

Without flow advection, our results show that for $N_0 \leq 8\pi$ the density distribution approaches its average over time as shown in Figures 4.5a–4.5b. However, for $N_0 > 8\pi$, the bump in the density function develops rapidly during evolution as shown in Figures 4.5c–4.5d. For the latter, we refine the mesh near the center to



(a) $t = 0$ (b) $t = 0.005$

(c) $t = 0.025$ (d) $t = 0.069084$

Figure 4.6: Uncontrolled evolution of density θ with $(x_0, y_0) = (0.5, 0.5)$ at various time steps

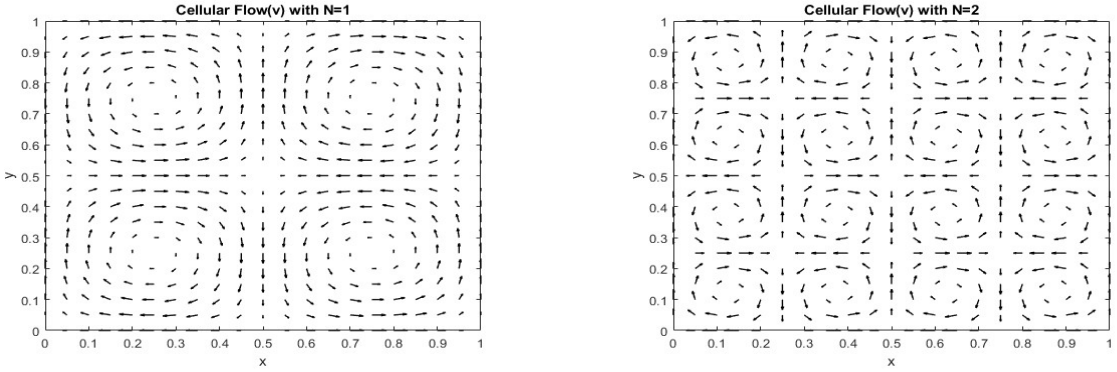
get more accurate results. Figure 4.6 demonstrates the density θ at various time steps for an initial mass of 9π with $(x_0, y_0) = (0.5, 0.5)$. We observe the rapid accumulation of mass toward the center, which indicates a possible singularity formation in finite time.

In the next example, we continue to study the PKS system with controlled flow advection, using the same settings as in Example 17. We present two examples to show how the behavior of the Gaussian function changes with different centers, using Algorithm 4.1

Example 18. *Consider the same domain and initial data with $N_0 = 9\pi$ as in Example 17. We further consider the cellular flow generated by*

$$\boldsymbol{v}_N(x,y) = 2\pi \begin{pmatrix} -\sin(2N\pi x)\cos(2N\pi y) \\ \cos(2N\pi x)\sin(2N\pi y), \end{pmatrix}, \quad N \in \mathbb{N}^+, \tag{4.72}$$

as plotted in Figure 4.7 for $N = 1, 2$. When $u(t) = N$ is a constant, the density distribution θ blows up for

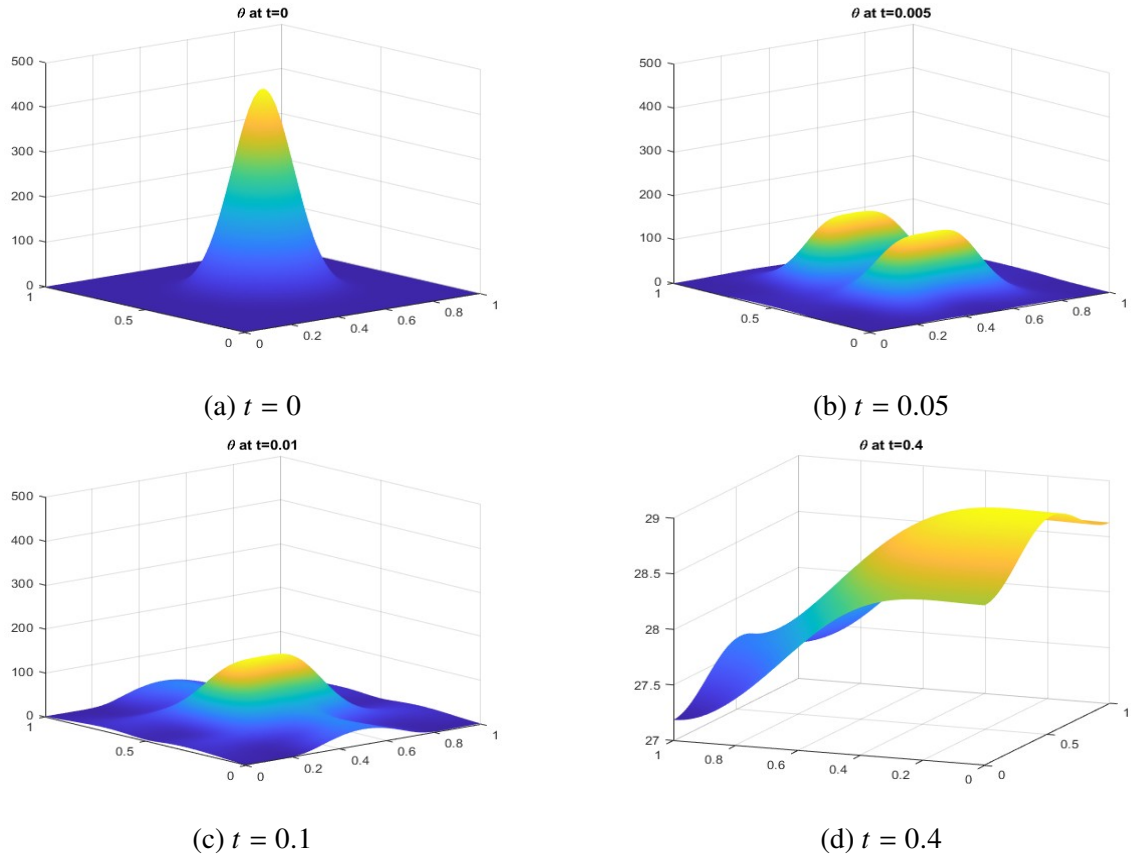


(a) Cellular flow when $N = 1$ (b) Cellular flow when $N = 2$

Figure 4.7: $\mathbf{v}_1$(Left) and $\mathbf{v}_2$(Right)

$N = 1$ but converges to its average for $N \geq 2$. Here we set $N = 2, t_f = 0.4$, $\alpha = \beta = 1$, $\gamma = 1$, $\underline{u} = -N$, $\overline{u} = N$, and the stop criterion in Algorithm 4.1 is set to be $\epsilon = 1e\text{-}08$.

Figures 4.8a–4.8d show the density evolution with controlled flow advection with $N = 2$ at $t = 0, 0.05, 0.1, 0.4$, respectively. The maximum value of θ decreases from 450 to 9π and the sharp peak is suppressed by flow advection. Moreover, θ approaches 9π as times evolved. The behavior of the control input $u(t)$ is presented in Figure 4.9a, which first oscillates and switches between positive and negative values, and then converges to zero. This suggests that adjusting flow orientation is important in improving the efficiency of flow advection, which may be more effective than simply increasing the flow strength for preventing the accumulation of mass. Figure 4.9b shows that the cost functional $J(u)$ decreases and converges.



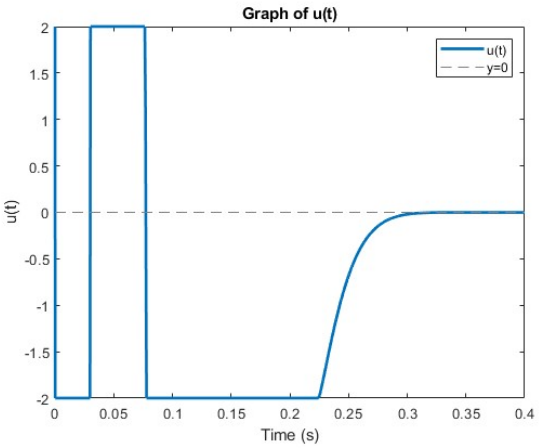
(a) $t = 0$ (b) $t = 0.05$

(c) $t = 0.1$ (d) $t = 0.4$

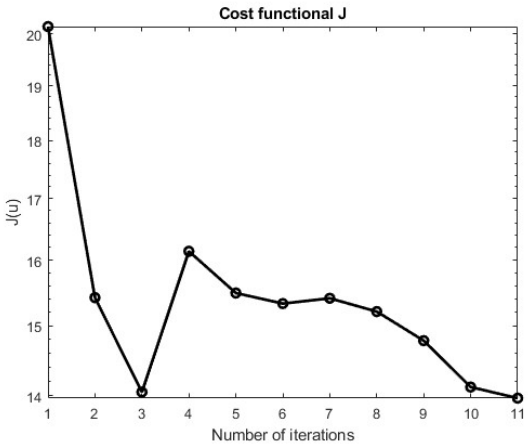
Figure 4.8: Controlled evolution of density θ with $(x_0, y_0) = (0.5, 0.5)$, $N = 2$ at various time steps $t = 0, 0.05, 0.1, 0.4$

Next, we investigate the effectiveness of flow advection with Gaussian initial data centered at different locations. It is important to note that when the streamlines of the velocity coincide with the level sets of θ, then $v \cdot \nabla\theta = 0$, and hence flow advection has no effect on enhancing diffusion.

We repeat the experiments for the Gaussian initial data with different centers (x_0, y_0), where x_0 and y_0 take on the values of $0.4, 0.5, 0.6, 0.7$. Initially, we set $N = 4$ in (4.72) and $\bar{u} = N$, but find that θ blows up in all cases except for $(x_0, y_0) = (0.5, 0.5)$. To address this issue, we increase $\bar{u}$ to $20N$ and find that θ does not blow up in any of these cases. As time progresses, the maximum value of θ decreases, and we observe that θ converges to its average, which is about 9π for different centers $(x_0, y_0) = (0.4, 0.6), (0.5, 0.6)$, and so on.



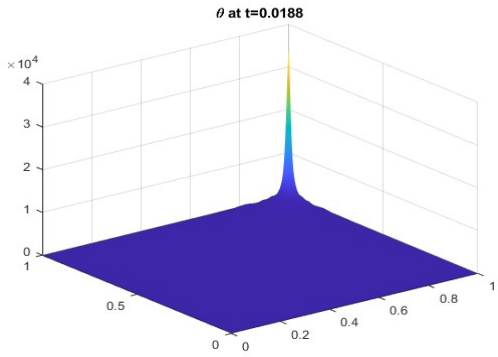
(a) Control input $u(t)$ for $t \in [0, 0.4]$



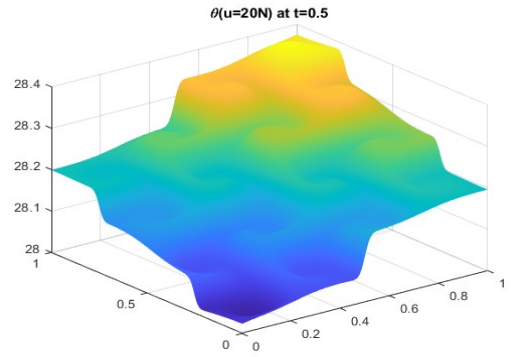
(b) Cost functional $J(u_j)$ for $j = 1, \cdots, 11$

Figure 4.9: Control input and cost functional for $(x_0, y_0) = (0.5, 0.5)$, $N = 4$, and $t \in [0, 0.4]$

Furthermore, we observe that a large $\bar{u}$ also allows a smaller frequency N. Figures 4.10a–4.10b demonstrate the density evolution for Gaussian initial datum centered at $(0.7, 0.7)$ when $t = 0.0188$ and $t = 0.5$, with $\bar{u} = N$ and $\bar{u} = 20N$ for $N = 2$, respectively. It is shown that increasing the upper bound $\bar{u}$ for the control input leads to a much faster decrease in the maximum value of θ and a convergence to its average value.



(a) Density θ at $t = 0.0188$ when $\bar{u} = N$



(b) Density θ at $t = 0.5$ when $\bar{u} = 20N$

Figure 4.10: Density θ for $(x_0, y_0) = (0.7, 0.7)$ with different $\bar{u}$ at $t = 0.0188$ and $t = 0.5$

Chapter 5

Conclusion

In summary, this thesis proposes and develops numerical methods for solving partial differential equations (PDEs) in a variety of settings, ranging from polygonal domains in $\mathbb{R}^d$ to bounded and connected domains with complex boundary conditions. Chapter 2 introduces a collocation method based on multivariate splines for solving second-order elliptic PDEs in non-divergence form. The method is shown to be accurate and efficient and can handle a wide range of elliptic PDEs. The proposed method is extended to the Poisson equation and numerical results are presented to demonstrate its effectiveness. Chapter 3 focuses on the Monge-Ampère equation, which arises from the optimal transportation problem and is fully nonlinear and subject to convexity constraints. The proposed method uses trivariate spline functions to approximate the solution of the Monge-Ampère equation over an arbitrary convex polyhedral domain. The collocation method is used to handle the fully nonlinear equation, and an iterative algorithm is employed to enforce the convexity constraint. Numerical results demonstrate the accuracy and efficiency of the proposed method, and the chapter highlights areas for future research. Chapter 4 tackles the problem of optimal control design for

the suppression of singularity formation in chemotaxis via flow advection. The authors propose a simplified parabolic-elliptic Patlak-Keller-Segel (PKS) equation with flow advection induced by the movement of the ambient fluid in a bounded and connected domain. The objective is to seek an optimal regulating function for the ambient fluid as to suppress the possible finite time blow-up. The authors show that the flows with arbitrarily small dissipation times can be constructed by rescaling a general class of smooth cellular flows, and the global well-posedness of the PKS system can be established under certain conditions. Overall, the thesis contributes to the development of numerical methods for solving PDEs in a variety of settings. The proposed methods are accurate, efficient, and can handle complex boundary conditions. The thesis highlights areas for future research, including the extension of methods to higher dimensions and the development of fast solvers for large-scale problems. The proposed methods have potential applications in various fields, including fluid dynamics, material science, and biological systems.

Appendix A

Convergence of Algorithm 1

We explain Algorithm 1 which is used to solve the minimization problem (2.15). In fact, Algorithm 1 is derived based on the solution to the following minimization

$$\min_{\mathbf{c}} J(c) = \frac{1}{2}(\alpha\|B\mathbf{c} - \mathbf{g}\|^2 + \beta\|H_r\mathbf{c}\|^2 + \gamma\|H_0\mathbf{c}\|^2) \quad \text{subject to } -K\mathbf{c} = \mathbf{f}, \tag{A.1}$$

where $B, \mathbf{g}$ are associated with the boundary condition, H_r is associated with the smoothness condition $\alpha > 0, \beta > 0$ are fixed parameters. Let us give a reason why we use (A.1) to replace (2.15). By Lemma 8, we know spline functions can approximate the solution of the PDE very well when the solution u is in $H^3(\Omega)$. When the size $|\triangle|$ is small enough, for the quasi-interpolatory spline S_u can approximate u such that $\|\Delta(u - S_u)\|_{L^2(\Omega)} \leq \epsilon_1$. That is the feasible set of (2.15) is not empty. Thus, two minimization problems (2.15) and (A.1) are closely related to each other. Even though there is not $\mathbf{c}$ satisfying $-K\mathbf{c} = \mathbf{f}$ exactly, a numerical computation in a computer will give a nearby solution $\mathbf{c}$ such that $\|K\mathbf{c} + \mathbf{f}\| \leq \epsilon_1$. We thus seek a spline solution u_s satisfying (A.1).

We use a similar technique in [AL05] and [ALW06]. For convenience, we first consider the problem

$$\min_{\mathbf{c}} J(c) = \frac{1}{2}(\alpha\|B\mathbf{z} - \mathbf{g}\|^2 + \beta\|H\mathbf{z}\|^2) \quad \text{subject to } -K\mathbf{z} = \mathbf{f}, \tag{A.2}$$

where $B, \mathbf{g}$ are from the boundary condition, H is from the smoothness condition. By the theory of Lagrange multipliers, letting

$$\mathcal{U}(z, \lambda) = \frac{1}{2}(\alpha z^\intercal B^\intercal B z - \alpha z^\intercal B^\intercal G - \alpha G^\intercal B z + \alpha G^\intercal G + \beta z^\intercal H^\intercal H z) + \lambda^\intercal (K\mathbf{z} + \mathbf{f}),$$

there exist λ such that

$$\frac{\partial \mathcal{U}}{\partial z} = \alpha B^\intercal B z - \alpha B^\intercal G + \beta H^\intercal H z + K^\intercal \lambda = 0 \tag{A.3}$$

$$\frac{\partial \mathcal{U}}{\partial \lambda} = K\mathbf{z} + \mathbf{f} = 0 \tag{A.4}$$

We can rewrite the above linear equations as follow:

$$\begin{bmatrix} K^\intercal & \alpha B^\intercal B + \beta H^\intercal H \\ O & K \end{bmatrix} \begin{bmatrix} \lambda \\ z \end{bmatrix} = \begin{bmatrix} \alpha B^\intercal G \\ -\mathbf{f} \end{bmatrix} \tag{A.5}$$

To solve (A.5), we consider the following sequence of problems for a fixed $\epsilon > 0$:

$$\begin{bmatrix} K^\intercal & \alpha B^\intercal B + \beta H^\intercal H \\ -\epsilon I & K \end{bmatrix} \begin{bmatrix} \lambda^{(k+1)} \\ z^{(k+1)} \end{bmatrix} = \begin{bmatrix} \alpha B^\intercal G \\ -\mathbf{f} - \epsilon \lambda^{(k)} \end{bmatrix} \tag{A.6}$$

for $k = 0, 1, \cdots$, with an initial guess $\lambda^{(0)} = 0$. Note that (A.6) reads

$$(\alpha B^\intercal B + \beta H^\intercal H) z^{(k+1)} + K^\intercal \lambda^{(k+1)} = \alpha B^\intercal G$$

$$K z^{(k+1)} - \epsilon \lambda^{(k+1)} = -\mathbf{f} - \epsilon \lambda^{(k)}$$

Multiplying on the both sides of the second equation in (A.6) by $K^\intercal$, we get

$$K^\intercal K z^{(k+1)} - \epsilon K^\intercal \lambda^{(k+1)} = -K^\intercal \mathbf{f} - \epsilon K^\intercal \lambda^{(k)}$$

or

$$K^\intercal \lambda^{(k+1)} = \frac{1}{\epsilon} K^\intercal K z^{(k+1)} - \frac{1}{\epsilon} K^\intercal \mathbf{f} + K^\intercal \lambda^{(k)}$$

and substitute it into the first equation in (A.6) to get

$$(\alpha B^\intercal B + \beta H^\intercal H) z^{(k+1)} + \frac{1}{\epsilon} K^\intercal K z^{(k+1)} - \frac{1}{\epsilon} K^\intercal \mathbf{f} + K^\intercal \lambda^{(k)} = \alpha B^\intercal G.$$

Simplifying the above equation leads to

$$(\alpha B^\intercal B + \beta H^\intercal H + \frac{1}{\epsilon} K^\intercal K) z^{(k+1)} = \alpha B^\intercal G + \frac{1}{\epsilon} K^\intercal \mathbf{f} - K^\intercal \lambda^{(k)} \tag{A.7}$$

It follows that

$$z^{(1)} = (\alpha B^\intercal B + \beta H^\intercal H + \frac{1}{\epsilon} K^\intercal K)^{-1} (\alpha B^\intercal G + \frac{1}{\epsilon} K^\intercal \mathbf{f} - K^\intercal \lambda^{(0)}) \tag{A.8}$$

Using the first equation in (A.6),i.e., $(\alpha B^\intercal B + \beta H^\intercal H)z^{(k+1)} = \alpha B^\intercal G - K^\intercal \lambda^{(k+1)}$ to replace $\alpha B^\intercal G$ in (A.7), we have

$$(\alpha B^\intercal B + \beta H^\intercal H + \frac{1}{\epsilon} K^\intercal K) z^{(k+1)} = (\alpha B^\intercal B + \beta H^\intercal H) z^{(k)} + \frac{1}{\epsilon} K^\intercal \mathbf{f}. \tag{A.9}$$

We get the minimizer using (A.8) and (A.9). These lead to Algorithm 1.

Next, we show the convergence of the above iterative algorithm. Since the minimization problem (2.15) is convex over a convex feasible set, we know that the minimization has a unique solution. We may assume that the linear system from Lagrange multiplier method has a solution pair (λ, z) with a unique solution z if the size $|\triangle|$ of triangulation $\triangle$ is small enough and the spline space $S_D^r(\triangle)$ is dense enough in $H^2(\Omega) \cap H_0^1(\Omega)$.

Theorem 26. *Suppose that the matrices K, H, B satisfy the following consistent condition: if $Kz = 0, Hz = 0$, and $Bz = 0$, one has $z = 0$. Then there exists a constant $\tilde{C}(\epsilon)$ depending on ϵ but independent of the iteration number k such that*

$$\|z^{(k+1)} - z\| \leq \|\tilde{K}^{-1}\| \|K^\intercal\| \Big(\frac{\tilde{C}\epsilon}{1 + \tilde{C}\epsilon}\Big)^{k+1}$$

for $k \geq 1$, where $\tilde{C} = \|K^+\|^2 \|\alpha B^\intercal B + \beta H^\intercal H\|$ and K^+ stands for the pseudo inverse of K and $\tilde{K} = \alpha B^\intercal B + \beta H^\intercal H + \frac{1}{\epsilon} K^\intercal K$.

Proof. First, we show that $\tilde{K}$ is invertible for $\alpha, \beta > 0$. If $\tilde{K}z = 0$, we have

$$c^\intercal \tilde{K} c = \alpha \|Bc\|^2 + \beta \|Hc\|^2 + \frac{1}{\epsilon} \|Kc\|^2 = 0$$

which implies that $Kz = 0, Hz = 0, Bz = 0$. By the assumption, $z = 0$. Thus, $\tilde{K}$ is invertible and hence the sequence $\{z^{(k)}\}$ is well-defined. Let $\tilde{C}_1 = \|\tilde{K}\|$ which depends on ϵ.

From (A.5) and (A.7) ,

$$\tilde{K}z^{(k+1)} = \alpha B^\intercal G + \frac{1}{\epsilon}K^\intercal \mathbf{f} - K^\intercal \lambda^{(k)}.$$

Hence, we have

$$z^{(k+1)} - z = \tilde{K}^{-1}K^\intercal(\lambda - \lambda^{(k)}). \tag{A.10}$$

By using (A.6) and (A.7), we get

$$-\epsilon(\lambda^{(k+1)} - \lambda) = -\epsilon(\lambda^{(k)} - \lambda) - \mathbf{f} - Kz^{(k+1)}$$

and

$$z^{(k+1)} = \tilde{K}^{-1}(\alpha B^\intercal G + \frac{1}{\epsilon}K^\intercal \mathbf{f} - K^\intercal \lambda^{(k)}).$$

It follows that

$$\begin{aligned}
-\epsilon(\lambda^{(k+1)} - \lambda) &= -\epsilon(\lambda^{(k)} - \lambda) - \mathbf{f} - Kz^{(k+1)} \\
&= -\epsilon(\lambda^{(k)} - \lambda) - \mathbf{f} - K\tilde{K}^{-1}(\alpha B^\intercal G + \frac{1}{\epsilon}K^\intercal \mathbf{f} - K^\intercal \lambda^{(k)}) \\
&= -\epsilon(\lambda^{(k)} - \lambda) - \mathbf{f} - K\tilde{K}^{-1}(\tilde{K}z + K^\intercal \lambda - K^\intercal \lambda^{(k)}) \\
&= -\epsilon(\lambda^{(k)} - \lambda) - \mathbf{f} - Kz - K\tilde{K}^{-1}K^\intercal(\lambda - \lambda^{(k)}).
\end{aligned}$$

As a result, we get

$$(\lambda^{(k+1)} - \lambda) = (\lambda^{(k)} - \lambda)(I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\intercal). \tag{A.11}$$

In order to show the next step, we use Lemma 7 in [ALW06], i.e., $\mathbb{R}^m = \mathrm{Ker}(K^\intercal) \oplus \mathrm{Im}(K)$ where $\mathrm{Ker}(K^\intercal)$ is the kernel of $K^\intercal$. Assume that $\lambda \in \mathrm{Im}(K)$. By the second equation in (A.6) that

$$K(z^{(k+1)} - z) = \epsilon(\lambda^{(k)} - \lambda^{(k+1)}).$$

That is, $\lambda^{(k)} - \lambda^{(k+1)}$ is in the $\mathrm{Im}(K)$ and therefore

$$\lambda^{(k)} - \lambda = \sum_{j=1}^{k} (\lambda^{(j)} - \lambda^{(j-1)}) + (\lambda^{(0)} - \lambda),$$

we have $\lambda^{(k)} - \lambda \in \mathrm{Im}(K)$ for each k. From (A.11), we need to estimate the norm of $I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\intercal$ restricted to $\mathrm{Im}(K)$ in order to estimate the norm of $\lambda^{(k+1)} - \lambda$. We write $\|I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\intercal\|$ for $\|(I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\intercal)|_{\mathrm{Im}(K)}\|$ and we have:

$$\|\lambda^{(k+1)} - \lambda\| \leq \|I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\intercal\| \|\lambda^{(k)} - \lambda\|.$$

We claim that

$$\|I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\intercal\| \leq \frac{\tilde{C}_2\epsilon}{1 + \tilde{C}_2\epsilon},$$

for some constant $\tilde{C}_2 > 0$. Indeed, by the Rayleigh-Ritz quotient, we have

$$\|\lambda^{(k+1)} - \lambda\| \leq \|I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\top\| = \max_{0\neq v\in \mathrm{Im}(K)} (1 - \frac{1}{\epsilon}\frac{v^\top K\tilde{K}^{-1}K^\top v}{v^\top v}).$$

By using a technique from [ALW06], we can get

$$\frac{1}{\epsilon}\frac{v^\top K\tilde{K}^{-1}K^\top v}{v^\top v} > \frac{1}{1+\tilde{C}_2\epsilon}, \quad \forall v \in \mathrm{Im}(K)$$

where $\tilde{C}_2 = \|K^+\|^2\|\alpha B^\top B + \beta H^\top H\|$. It follows that

$$\|I - \frac{1}{\epsilon} K\tilde{K}^{-1}K^\top\| \leq 1 - \frac{1}{1+\tilde{C}_2\epsilon} = \frac{\tilde{C}_2\epsilon}{1+\tilde{C}_2\epsilon}.$$

As a results, we obtain

$$\|\lambda^{(k+1)} - \lambda\| \leq \frac{\tilde{C}_2\epsilon}{1+\tilde{C}_2\epsilon}\|\lambda^{(k)} - \lambda\|$$

and from (A.10)

$$\|z^{(k+1)} - z\| \leq \|\tilde{K}^{-1}\|\|K^\top\|\Big(\frac{\tilde{C}_2\epsilon}{1+\tilde{C}_2\epsilon}\Big)^{k+1}\|\lambda^{(0)} - \lambda\|.$$

□

BIBLIOGRAPHY

[AL05] G. Awanou and M.-J. Lai. "On Convergence Rate of the Augmented Lagrangian Algorithm for Nonsymmetric Saddle Point Problems". In: *Journal of Applied Numerical Mathematics* 54.1-2 (2005), pp. 122–134.

[AL14] Geard Awanou and H. Li. "Error analysis of a mixed finite element method for the Monge-Ampére equation". In: *International Journal of Numerical Analysis and Modeling* 11.4 (2014), pp. 745–761.

[ALW06] G. Awanou, M.-J. Lai, and P. Wenston. "The multivariate spline method for scattered data fitting and numerical solution of partial differential equations". In: *Wavelets and splines: Athens 2005*. Nashboro Press, 2006, pp. 24–74.

[Awa03] Gerard Awanou. "Energy methods in 3D spline approximations". PhD thesis. University of Georgia, 2003.

[Awa13] Gerard Awanou. "Pseudo time continuation and time marching methods for Monge-Ampére type equations". In: *Advances in Computational Mathematics* 41.4 (2013), pp. 741–766.

[Awa14] Gerard Awanou. “Standard finite elements for the numerical resolution of the elliptic Monge-Ampére equation: classical solutions”. In: *IMA Journal of Numerical Analysis* 34.3 (2014), pp. 828–846.

[Awa15] Gerard Awanou. “Spline element method for Monge-Ampére equations”. In: *BIT Numerical Mathematics* 55.2 (2015), pp. 625–646.

[Awa16] Gerard Awanou. “On standard finite difference discretizations of the elliptic Monge-Ampére equation”. In: *Journal of Scientific Computing* 69.2 (2016), pp. 892–904.

[BB00] J. Benamou and Y. Brenier. “A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem”. In: *Numerische Mathematik* 84 (2000), pp. 375–393.

[Ben+20] JJ Benitoa et al. “Solving Monge-Ampére equation in 2D and 3D by Generalized Finite Difference Method”. In: *Engineering Analysis with Boundary Elements* 124C (2020), pp. 52–63.

[Ber06] MV Berry. “Oriental magic mirrors and the Laplacian image”. In: *European Journal of Physics* 27 (2006), pp. 109–118.

[BFO10] J-D Benamou, BD Froese, and AM Oberman. “Two Numerical Methods for the elliptic Monge-Ampére equation”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 44 (2010), pp. 737–758.

[BH17] Jacob Bedrossian and Siming He. “Suppression of blow-up in Patlak–Keller–Segel via shear flows”. In: *SIAM Journal on Mathematical Analysis* 49.6 (2017), pp. 4722–4766.

[Bis60] FE Bisshopp. "On two-dimensional cell patterns". In: *Journal of Mathematical Analysis and Applications* 1.3-4 (1960), pp. 373–385.

[Bre11] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer, 2011.

[BS19] K. Bohmer and R. Schaback. "A meshfree method for solving the Monge-Ampére equation". In: *Numerical Algorithms* 82 (2019), pp. 539–551.

[Caf90] L. A. Caffarelli. "Interior $W^{2,p}$ estimates for solutions of the Monge-Ampére equation". In: *Ann. of Math.* 131.1 (1990), pp. 135–150.

[CC95] L. A. Caffarelli and X. Cabr'e. *Fully nonlinear elliptic equations*. Vol. 43. American Mathematical Society Colloquium Publications. American Mathematical Society, 1995.

[CGG18] Alexandre Caboussat, Roland Glowinski, and Dimitrios Gourzoulidis. "A Least-Squares/Relaxation Method for the Numerical Solution of the Three-Dimensional Elliptic Monge-Ampére Equation". In: *Journal of Scientific Computing* 77 (2018), pp. 53–78.

[CLW21] S. Chen, J. Liu, and X.-J. Wang. "Global regularity for the Monge-Ampére equation with natural boundary condition". In: *Ann. of Math.* 194.3 (2021), pp. 745–793.

[CNS84] L. A. Caffarelli, L. Nirenberg, and J. Spruck. "The Dirichlet problem for nonlinear second order elliptic equations I. Monge-Ampére equation". In: *Commun. Pure Appl. Math.* 37.3 (1984), pp. 369–402.

[Con+08] Peter Constantin et al. “Diffusion and mixing in fluid flow”. In: *Annals of Mathematics* (2008), pp. 643–674.

[DG03] E. J. Dean and R. Glowinski. “Numerical solution of the two-dimensional elliptic Monge-Ampére equation with Dirichlet boundary conditions: an augmented Lagrangian approach”. In: *C. R. Math. Acad. Sci. Paris* 336.9 (2003), pp. 779–784.

[DG04] E. J. Dean and R. Glowinski. “Numerical solution of the two-dimensional elliptic Monge-Ampére equation with Dirichlet boundary conditions: a least-squares approach”. In: *C. R. Math. Acad. Sci. Paris* 339.12 (2004), pp. 887–892.

[Eve98] L. Evens. *Partial Differential Equation*. American Mathematical Society, 1998.

[Fig17] A. Figalli. *The Monge-Amp‘ere Equation and Its Applications*. European Mathematical Society, 2017.

[Fil06] F. Filbet. “A finite volume scheme for the Patlak–Keller–Segel chemotaxis model”. In: *Numerische Mathematik* 104.4 (2006), pp. 457–488.

[FM03] K Renee Fister and C Maeve McCarthy. “Optimal control of a chemotaxis system”. In: *Quarterly of Applied Mathematics* (2003), pp. 193–211.

[FN09] X. Feng and M. Neilan. “Mixed finite element methods for the fully nonlinear Monge-Amp‘ere equation based on the vanishing moment method”. In: *SIAM Journal on Numerical Analysis* 47.2 (2009), pp. 1226–1250.

[Fro12] B. D. Froese. “A numerical method for the elliptic Monge–Amp‘ere equation with transport boundary condition”. In: *SIAM Journal on Scientific Computing* 34.3 (2012), A1432–A1459.

[GL20] F. Gao and M.-J. Lai. “A new H^2 regularity condition of the solution to Dirichlet problem of the Poisson equation and its applications”. In: *Acta Mathematica Sinica* 36.1 (2020), pp. 21–39.

[GLS15] Juan B Gutierrez, Ming-Jun Lai, and George Slavov. “Bivariate spline solution of time dependent nonlinear PDE for a population density over irregular domains”. In: *Mathematical biosciences* 270 (2015), pp. 263–277.

[GMR20a] Francisco Guillen-Gonzalez, Exequiel Mallea-Zepeda, and Maria Angeles Rodriguez-Bellido. “A regularity criterion for a 3D chemo-repulsion system and its application to a bilinear optimal control problem”. In: *SIAM Journal on Control and Optimization* 58.3 (2020), pp. 1457–1490.

[GMR20b] Francisco Guillén-González, Exequiel Mallea-Zepeda, and Marıa Ángeles Rodrıguez-Bellido. “Optimal bilinear control problem related to a chemo-repulsion system in 2D domains”. In: *ESAIM: Control, Optimisation and Calculus of Variations* 26 (2020), p. 29.

[Gri85] P. Grisvard. *Elliptic Problems in Nonsmooth Domains*. Pitman, 1985.

[Her00] Miguel A Herrero. “Asymptotic properties of reaction-diffusion systems modeling chemotaxis”. In: *Applied and Industrial Mathematics, Venice—2, 1998*. Springer, 2000, pp. 89–108.

[HMV97] Miguel A Herrero, E Medina, and J. J. L Velázquez. “Finite-time aggregation into a single point in a reaction-diffusion system”. In: *Nonlinearity* 10.6 (1997), pp. 1739–1754.

[HMV98] Miguel A Herrero, E Medina, and JJL Velázquez. “Self-similar blow-up for a reaction-diffusion system”. In: *Journal of Computational and Applied Mathematics* 97.1-2 (1998), pp. 99–119.

[Hor03] Dirk Horstmann. “From 1970 until present: the Keller-Segel model in chemotaxis and its consequences. I”. In: *Jahresber. Deutsch. Math.-Verein.* 105 (2003), pp. 103–165.

[Hor04] Dirk Horstmann. “From 1970 until present: the Keller-Segel model in chemotaxis and its consequences. II, Jahresber”. In: *Deutsch. Math.-Verein.* 106 (2004), pp. 51–69.

[Hu22] W. Hu. “Global regularity and stability analysis of the Patlak-Keller-Segel system with flow advection in a bounded domain: a semigroup approach”. In: *preprint* (2022).

[HV97] M. A. Herrero and JJL Velázquez. “A blow-up mechanism for a chemotaxis model”. In: *Annali Della Scuola Normale Superiore Di Pisa Classe Di Scienze* 24.4 (1997).

[IXZ21] Gautam Iyer, Xiaoqian Xu, and Andrej Zlatoš. “Convection-induced singularity suppression in the Keller-Segel and other non-linear PDEs”. In: *Transactions of the American Mathematical Society* 374.09 (2021), pp. 6039–6058.

[JL92] Willi Jäger and Stephan Luckhaus. “On explosions of solutions to a system of partial differential equations modelling chemotaxis”. In: *Transactions of the american mathematical society* 329.2 (1992), pp. 819–824.

[KS70] Evelyn F Keller and Lee A Segel. “Initiation of slime mold aggregation viewed as an instability”. In: *Journal of theoretical biology* 26.3 (1970), pp. 399–415.

[KS71] Evelyn F Keller and Lee A Segel. “Model for chemotaxis”. In: *Journal of theoretical biology* 30.2 (1971), pp. 225–234.

[KX16] Alexander Kiselev and Xiaoqian Xu. “Suppression of chemotactic explosion by mixing”. In: *Archive for Rational Mechanics and Analysis* 222.2 (2016), pp. 1077–1112.

[Lai89] M.-J. Lai. “On Construction of Bivariate and Trivariate Vertex Splines on Arbitrary Mixed Grid Partitions”. PhD thesis. Texas A&M University, 1989.

[LG21] N. Lei and X. Gu. “FFT-OT: A Fast Algorithm for Optimal Transportation”. In: *ICCV*. 2021.

[Lio71] J.-L. Lions. *Optimal control of systems governed by partial differential equations*. Springer, New York, 1971.

[Liu+16] J. Liu et al. “A multigrid scheme for 3D Monge-Ampére equations”. In: *International Journal of Computer Mathematics* (2016).

[LL22] M.J. Lai and J. Lee. “A Multivariate Spline based Collocation Method for Numerical Solution of PDEs”. In: *SIAM Journal on Numerical Analysis* 60.4 (2022), pp. 2405–2434.

[LL23] M.-J. Lai and J. Lee. “Trivariate Spline Collocation Methods for Numerical Solution to 3D Monge-Ampére Equation”. In: *Journal of Scientific Computing* 95 (Apr. 2023). DOI: 10.1007/s10915-023-02183-9.

[LM17] M.-J. Lai and C. Mersmann. "Adaptive Triangulation Methods for Bivariate Spline Solutions of PDEs". In: *Approximation Theory XV: San Antonio, 2016*. Ed. by G. Fasshauer and L.L. Schumaker. Springer Verlag, 2017, pp. 155–175.

[LS07a] M.-J. Lai and L.L. Schumaker. *Spline Functions over Triangulations*. Cambridge University Press, 2007.

[LS07b] M.-J. Lai and L.L. Schumaker. "Trivariate C^r polynomial macro-elements". In: *Constructive Approximation* 26.1 (2007), pp. 11–28.

[LW18] M.-J. Lai and C.M. Wang. "A bivariate spline method for 2nd order elliptic equations in non-divergence form". In: *Journal of Scientific Computing* 74.2 (2018), pp. 803–829.

[LW21] M.-J. Lai and Y. Wang. "Sparse Solutions to Underdetermined Linear Systems". In: *Publication, Philadelphia* (2021).

[LX20] Z. Liu and Q. Xu. "On Multiscale RBF Collocation Methods for Solving the Monge-Ampére Equation". In: *Mathematical Problems in Engineering* 2020 (2020), p. 1748037.

[Mer19] C. Mersmann. "Numerical Solution of Helmholtz equation and Maxwell equations". University of Georgia, 2019.

[MY01] S.-Y. Mak and D.-Y. Yip. "Secrets of the chinese magic mirror replica". In: *Phys. Ed.* 36 (2001), pp. 102–107.

[MY17] L. Mu and X. Ye. “A simple finite element method for non-divergence form elliptic equations”. In: *International Journal of Numerical Analysis and Modeling* 14.2 (2017), pp. 306–311.

[Nag95] Toshitaka Nagai. “Blow-up of radially symmetric solutions to a chemotaxis system”. In: *Advances in Mathematical Sciences and Applications* 5 (1995), pp. 581–601.

[Ost60] A.M. Ostrowski. *Solution of Equations and Systems of Equations*. Vol. IX. Pure and Applied Mathematics. New York-London: Academic Press, 1960, pp. ix +202.

[Pat53] Clifford S Patlak. “Random walk with persistence and external bias”. In: *The bulletin of mathematical biophysics* 15.3 (1953), pp. 311–338.

[PD09] Benoıt Perthame and Anne-Laure Dalibard. “Existence of solutions of the hyperbolic Keller-Segel model”. In: *Transactions of the american mathematical society* 361.5 (2009), pp. 2319–2335.

[RRV18] M Ángeles Rodrıguez-Bellido, Diego A Rueda-Gómez, and Élder J Villamizar-Roa. “On a distributed control problem for a coupled chemotaxis-fluid model”. In: *Discrete & Continuous Dynamical Systems-B* 23.2 (2018), p. 557.

[RY01] Sang Uk Ryu and Atsushi Yagi. “Optimal control of Keller-Segel equations”. In: *Journal of mathematical analysis and applications* 256.1 (2001), pp. 45–66.

[Ryu08] Sang-Uk Ryu. “Boundary control of chemotaxis reaction diffusion system”. In: *Honam Mathematical Journal* 30.3 (2008), pp. 469–478.

[RZ07] Lenya Ryzhik and Andrej Zlatoš. “KPP pulsating front speed-up by flows”. In: *Communications in Mathematical Sciences* 5.3 (2007), pp. 575–593.

[Sch15] L.L. Schumaker. *Spline Functions: Computational Methods*. SIAM, 2015.

[Sch19] L.L. Schumaker. "Solving elliptic PDE's on domains with curved boundaries with an immersed penalized boundary method". In: *Journal of Scientific Computing* 80.3 (2019), pp. 1369–1394.

[SS11] Takasi Senba and Takashi Suzuki. *Applied analysis: mathematical methods in natural science*. World Scientific, 2011.

[SS13] I. Smears and E. S"uli. "Discontinuous Galerkin finite element approximation of nondivergence form elliptic equations with Cordes coefficients". In: *SIAM Journal on Numerical Analysis* 51.4 (2013), pp. 2088–2106.

[Suz05] Takashi Suzuki. *Free energy and self-interacting particles*. Springer, 2005.

[Vil03] C'edric Villani. *Topics in Optimal Transportation*. AMS, 2003.

[Vil08] C'edric Villani. *Optimal Transport: Old and New*. Vol. 338. Springer Science & Business Media, 2008.

[Wan96] Xu-Jia Wang. "Regularity for Monge-Amp‘ere equation near the boundary". In: *Analysis* 16 (1996), pp. 101–107.

[Wei21] Dongyi Wei. "Diffusion and mixing in fluid flow via the resolvent estimate". In: *Science China Mathematics* 64.3 (2021), pp. 507–518.

[WW19] C. Wang and J. Wang. "A primal dual weak Galerkin finite element method for second order elliptic equations in non-divergence form". In: *Mathematics of Computation* 88.319 (2019), pp. 189–215.

[Xu19] Y. D. Xu. “Multivariate Splines for Scattered Data Fitting. Eigenvalue Problems, and Numerical Solution to Poisson equations”. PhD thesis. University of Georgia, Athens, GA, 2019.

[ZNC11] X. Zhu, J. Ni, and Q. Chen. “An optical design and simulation of LED low-beam headlamps”. In: *Journal of Physics: Conference Series* 276 (2011), p. 012201.