ABSTRACT

EIGENVALUE PROBLEMS FOR FULLY NONLINEAR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS WITH TRANSPORT BOUNDARY CONDITIONS

by
Jacob Lesniewski

Fully nonlinear elliptic partial differential equations (PDEs) arise in a number of applications. From mathematical finance to astrophysics, there is a great deal of interest in solving them. Eigenvalue problems for fully nonlinear PDEs with transport boundary conditions are of particular interest as alternative formulations of PDEs that require data to satisfy a solvability condition, which may not be known explicitly or may be polluted by noisy data. Nevertheless, these have not yet been well-explored in the literature. In this dissertation, a convergence framework for numerically solving eigenvalue problems for fully nonlinear PDEs is introduced. In addition, existing two-dimensional methods for nonlinear equations are extended to handle transport boundary conditions and eigenvalue problems. Finally, new techniques are designed to enable appropriate discretization of a large range of fully nonlinear three-dimensional equations.

EIGENVALUE PROBLEMS FOR FULLY NONLINEAR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS WITH TRANSPORT BOUNDARY CONDITIONS

by
Jacob Lesniewski

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology and
Rutgers, The State University of New Jersey – Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mathematical Sciences

Department of Mathematical Sciences
Department of Mathematics and Computer Science, Rutgers-Newark

May 2021

APPROVAL PAGE

EIGENVALUE PROBLEMS FOR FULLY NONLINEAR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS WITH TRANSPORT BOUNDARY CONDITIONS

Jacob Lesniewski

Dr. Brittany D. Froese Hamfeldt, Dissertation Advisor Date
Associate Professor of Mathematics, New Jersey Institute of Technology

Dr. David G. Shirokoff, Committee Member Date
Associate Professor of Mathematics, New Jersey Institute of Technology

Dr. Michael S. Siegel, Committee Member Date
Professor of Mathematics, New Jersey Institute of Technology

Dr. Travis L. Askham, Committee Member Date
Assistant Professor of Mathematics, New Jersey Institute of Technology

Dr. Thomas L. Lewis, Committee Member Date
Associate Professor of Mathematics, The University of North Carolina at Greensboro

BIOGRAPHICAL SKETCH

Author: Jacob Lesniewski

Degree: Doctor of Philosophy

Date: May 2021

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mathematical Sciences,
  New Jersey Institute of Technology, Newark, NJ, 2021
- Master of Science in Applied Mathematics,
  New Jersey Institute of Technology, Newark, NJ, 2018
- Bachelor of Arts in Mathematics,
  Caldwell University, Caldwell, NJ, 2016

Major: Mathematical Sciences

Publications:

J. Lesniewski and B.F. Hamfeldt, “A convergent finite difference method for computing minimal Lagrangian graphs,” *Submitted.*

J. Lesniewski and B.F. Hamfeldt, “Convergent finite difference methods for fully nonlinear elliptic equations in three dimensions,” *Submitted.*

*I dedicate my dissertation work to my family and friends,
living and deceased.*

ACKNOWLEDGMENT

There are many people to thank for this work since I could not have done it without their support.

First, I am very grateful to my advisor, Dr. Brittany Hamfeldt for her patience, enthusiasm, and guidance. From the first year qualifying exams through all five years of my time here, she has always provided me ideas, inspiration, and encouragement. Without her, I could not have completed this work.

Next, I would like to thank Doctors David Shirokoff, Michael Siegel, Travis Askham and Thomas Lewis for being on my dissertation committee. Dr. Shirokoff taught me an excellent course in Convex Optimization which has been very helpful for this dissertation work. I would also like to thank the entire committee for their feedback throughout my time here.

I am also grateful to the Department of Mathematical Sciences for the teaching assistantship and the opportunity to pursue my Ph.D. at NJIT. This research was partially supported by the National Science Foundation under NSF DMS-1619807. I am thankful for the financial support which has allowed me to spend more time on this work.

I would also like to thank my family and friends at home, especially my wife Beth, for all of their support during my Ph.D. at NJIT. In addition, I would like to thank all of the friends and colleagues I met at NJIT, especially Tensae Andargachew, Yixuan Sun, and Guangyuan Liao, whose advice and collaboration helped me get through the most difficult times.

Finally, I am deeply grateful to God, since all of my accomplishments are only through His grace.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

List of Tables

List of Figures

Figure **Page**

List of Figures (Continued)

Figure **Page**

List of Figures (Continued)

CHAPTER 1

INTRODUCTION

There has been a rising interest in numerical techniques for solving fully nonlinear elliptic partial differential equations (PDEs) because of the frequency of their appearance in applications [33]. For example, they appear in optimal transport, meteorology [12], differential geometry [10], reflector/refractor design [28], seismology [17], astrophysics [23], mathematical finance [22], computer graphics [47], and mesh generation [8, 33]. When these problems arise in applications, they often include discontinuous or sharp jumps in the data and involve intricate domains. Many fully nonlinear elliptic PDEs can be expressed as eigenvalue problems, which have not yet been well-studied in this context.

The general form of the problem is

$$F\left(D^2u(x)\right) = c, \quad x \in X \tag{1.1}$$

$$\nabla u\left(X\right) \subset \bar{Y} \tag{1.2}$$

where c is an unknown eigenvalue and (5.9) corresponds to a transport boundary condition. Here $\nabla u(X)$ is the image of the domain X under the mapping ∇u.

Some examples include curvature equations used to construct minimal surfaces, the Monge-Ampére equation, and generated Jacobian equations arising in more general optimal transport problems [45, 26, 7]. We will focus on a particular example, a PDE for the computation of minimal Lagrangian graphs, with the goal of developing techniques that can be applied to more complicated PDEs:

$$\sum_{j=1}^{d}\left(\tan^{-1}\left(\lambda_j\right)\right) = c \tag{1.3}$$

where λ_j are the eigenvalues of the Hessian matrix and d is the number of dimensions.

The problem of constructing minimal surfaces is important in applications such as materials science [49] and molecular engineering [2]. There has also been recent interest in the use of mean curvature flows to generate minimal Lagrangian submanifolds of Calabi-Yau manifolds [48, 50].

Here the problem is solved with unusual implicit transport type boundary conditions which require the gradient to map one set into another. From this and the nature of the domains in practice, there is a need for efficient methods on non-uniform grids. Ultimately, the techniques demonstrated on the Lagrange curvature problem will be adapted to solve and analyze other more challenging PDEs.

Because of the interest in fully nonlinear elliptic PDEs, there have been many techniques developed for numerically solving them. These techniques include finite difference methods [19, 45], finite element methods [5, 35], and even least squares methods [9, 13]. However, simple implementations of these methods can fail to converge when ellipticity of the PDE degenerates or when the equation does not have a smooth solution. Simple extensions of standard methods usually break down when used on a degenerate elliptic PDE, or if a smooth solution does not exist. Those that have convergence guarantees are generally restricted to uniform Cartesian grids and find difficulty dealing with complex geometries.

Two frameworks for proving the convergence of numerical methods have been proposed in recent years [1, 20]. The results of this dissertation are motivated by the work of Barles and Souganidis, who were able to prove the convergence of monotone methods for fully nonlinear elliptic PDEs for which a comparison principle exists [1]. Since then, many techniques have been developed and improved upon such as wide stencil methods [45], meshfree methods [26], and filtered schemes [24]. Recently, the stencils required for these schemes have been made more accurate and more efficient by the introduction of modified piecewise Cartesian grids [33], but little has been done to address the solutions of eigenvalue problems with transport type boundary conditions. In general, these problems do not come with a comparison principle, so more analysis is needed to establish convergence for this boundary condition and the related eigenvalue problems.

In this dissertation, techniques for solving eigenvalue problems for fully nonlinear elliptic PDEs are presented. In particular, we propose a framework for discretizing and solving these problems, along with a proof of convergence. Additionally, we adapt existing methods to produce an efficient two-dimensional implementation which allows for weak solutions and complicated geometries. Finally, we present new monotone stencils for extending the schemes used in two dimensions to three dimensions. Natural generalizations of these schemes can be prohibitively expensive in three dimensions. Thus, we introduce new techniques for efficiently constructing the monotone stencils.

1.1 Outline of this Dissertation

There are multiple problems to address in solving fully nonlinear elliptic PDEs: criteria for convergence, efficient construction of the nonlinear operator, and efficiently solving the resulting nonlinear system. This dissertation focuses primarily on efficiently building and analyzing the operator. This is not an easy feat, especially in three dimensions or for general domains in two dimensions. Efficient solvers are future work.

In Chapter 2, we present an introduction to viscosity solutions specifying the weak solutions of elliptic PDEs, as well as existing results on convergence of numerical methods for these types of equations. In Chapter 3, we leverage this background in order to demonstrate convergence of monotone schemes for solving eigenvalue problems. In Chapter 4, we outline the construction of convergent monotone schemes in two dimensions. In Chapter 5, we extend these results to implement monotone schemes in three dimensions. These schemes are used to solve a large range of nonlinear PDEs including eigenvalue problems. Finally, in Chapter 6 we conclude and discuss future work.

CHAPTER 2

BACKGROUND

In this chapter, we discuss the existing theory available for fully nonlinear elliptic PDEs. We start with important theoretical results and then discuss more specifically the type of PDEs being solved in this dissertation. Finally, we review existing methods for solving fully nonlinear elliptic equations in two dimensions.

2.1 Elliptic Equations

A PDE

$$F(x, u, Du, D^2u) = 0 \tag{2.1}$$

is fully nonlinear and elliptic if it exhibits nonlinear dependence on the highest order derivative, and satisfies the ellipticity condition:

Definition 2.1 (elliptic operator)**.** Equation (2.1) is **(degenerate) elliptic** if

$$F(x, r, p, X) \leq F(x, s, p, Y)$$

for all $x \in \bar{X}$, $r, s \in \mathbb{R}$, $p \in \mathbb{R}^n$, $X, Y \in S^n$ with $X \geq Y$ and $r \leq s$, where $X \geq Y$ means $X - Y$ is a positive definite matrix, and S^n is the set of symmetric n by n matrices [24].

A desirable property that is shared by many elliptic operators is the comparison principle.

Definition 2.2 (comparison principle)**.** The PDE operator (2.1) satisfies a **comparison principle** if whenever $F(x, u(x), \nabla u(x), D^2u(x)) \leq F(x, v(x), \nabla v(x), D^2v(x))$ for all $x \in \bar{X}$ then $u(x) \leq v(x)$ for all $x \in \bar{X}$ [1].

A comparison principle can be used to establish existence and uniqueness of solutions to the PDE. A common technique for proving existence is Perron's method, which involves arguing that the maximal subsolution

$$u(x) \equiv \sup \left\{ v(x) \mid F(x, v(x), \nabla v(x), D^2 v(x)) \leq 0 \right\} \tag{2.2}$$

is actually a solution to the PDE. Uniqueness of solutions follows immediately from a comparison principle [1].

2.2 Viscosity Solutions

Many fully nonlinear elliptic equations do not possess a classical solution, and thus, some notion of weak solution is needed. A powerful approach is the viscosity solution, which relies on a maximum principle argument to transfer derivatives onto smooth test functions [11].

An equation can be shown to have a unique viscosity solution if it has a comparison principle [11]. In order to understand the weak solutions we solve for, the theory of viscosity solutions must be introduced.

Viscosity solutions were introduced by Lions and Crandall in the 1980s. These are a generalization of the classical solution to a PDE which can be found for degenerate elliptic PDEs. Let $X \subset \mathbb{R}^d$ and $F : X \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^{d \times d}$. Then the function $u \in C^2$ is considered a classical solution if

$$F(x, u, Du, D^2 u) = 0. \tag{2.3}$$

In contrast to classical solutions, an equation may have a viscosity solution even when Du or $D^2 u$ does not exist. For many PDEs we consider, classical solutions may not exist. This motivates the theory of viscosity solutions, particularly when dealing with degenerate elliptic PDEs, which need not have smooth solutions [24]. Viscosity solutions provide a framework which allows many comparison, uniqueness, existence, and continuous dependence theorems to be proved. A problem can be shown to have a unique viscosity solution if it has a comparison principle (Definition 2.10) [11]. Degenerate ellipticity (Definition 2.1) is related to the comparison principle, so we expect that many degenerate elliptic PDEs will have a comparison principle. In order to define viscosity solutions, we first must define the upper and lower semicontinuous envelopes [24]. These are not equivalent in general since there are second order linear problems that are uniformly elliptic but have multiple viscosity solutions [24].

Definition 2.3. (upper and lower semicontinuous envelopes) The **upper and lower semicontinuous envelopes** of a function u(x) are defined by

$$u^*(x) = \limsup_{y \to x} u(y), \tag{2.4}$$

and

$$u_*(x) = \liminf_{y \to x} u(y), \tag{2.5}$$

respectively [24].

Definition 2.4. A function $f(x)$ is called upper semicontinuous (USC) if for every point x_0 in its domain,

$$\limsup_{x \to x_0} f(x) \leq f(x_0)$$

Definition 2.5. A function $f(x)$ is called lower semicontinuous (LSC) if for every point x_0 in its domain,

$$\liminf_{x \to x_0} f(x) \geq f(x_0)$$

Definition 2.6. A lower semicontinuous function $u : X \to \mathbb{R}$ is a **(viscosity) supersolution** if for any $x \in X$ and any $\phi \in C^2(X)$ such that $u - \phi$ attains a (local) minimum at x,

$$F^* \left(x, u(x), D\phi(x), D^2\phi(x)\right) \geq 0. \tag{2.6}$$

Definition 2.7. Similarly, an upper semicontinous function $u : X \to \mathbb{R}$ is a **(viscosity) subsolution** if for any $x \in X$ and any $\phi \in C^2(X)$ such that $u - \phi$ attains a (local) maximum at x,

$$F_*\left(x, u(x), D\phi(x), D^2\phi(x)\right) \leq 0. \tag{2.7}$$

Definition 2.8. A continuous function u is called a **viscosity solution** if it is both a subsolution and a supersolution (See Figure 2.1).

When checking for viscosity solutions, we may assume the maximum or minimum of $u - \phi$ is unique, strict, and global with $u - \phi = 0$ at the extremum [24].

To illustrate how viscosity solutions work, we present the example of the Eikonal equation:

$$F(Du) = 1 - |Du| = 0 \tag{2.8}$$

$u(x) = |x|$ is a viscosity solution to this equation. Since $|x|$ is differentiable for $x \neq 0$ and the derivatives, ± 1, satisfy the equation,

$$1 - |Du| = 1 - |\pm 1| = 1 - 1 = 0$$

Since the derivative exists there, the local max (or min) of $u - \phi$ implies $u' - \phi' = 0$ or $\phi' = u' = \pm 1$. Thus,

$$F(x, u, D\phi) = F(D\phi) = 1 - |\pm 1| = 0$$

for $x \neq 0$. Hence, $F(D\phi) \leq 0$ and $F(D\phi) \geq 0$, respectively; so u is both a sub- and supersolution for $x \neq 0$.

To show this is a viscosity solution at the singularity, we first show that it is a supersolution. Suppose $u - \phi = |x| - \phi$ has a local minimum at $x = 0$. Then

$0 \in \partial(u - \phi)$ i.e., 0 is in the subdifferential of $u - \phi$ and ϕ for which there is a local minimum belong to the same set as the subdifferential of u at zero. Thus, $\partial u(0) = [-1, 1]$ and $\phi' \in [-1, 1]$. Hence,

$$F(D\phi) = 1 - |\phi'| \geq 0$$

Thus, u is a supersolution.

To verify that is a subsolution, note that we need only satisfy the inequality for any $\phi \in C^1$ such that $u - \phi$ has a local maximum at zero. We can see graphically that any ϕ that touched $|x|$ above so that $|x| - \phi$ is a local maximum would have a sharp turn and not be differentiable there and hence, not C^1. Since there is no such ϕ, there is nothing to check, and u is a subsolution. Therefore, $u = |x|$ is a viscosity solution of $1 - |Du| = 0$.

Remark 2.9. *u is not a viscosity solution of the seemingly equivalent equation $|Du| - 1 = 0$ since $|\phi'| - 1 \leq 0$ so we can't show that it is a supersolution. It is still a subsolution by default.*

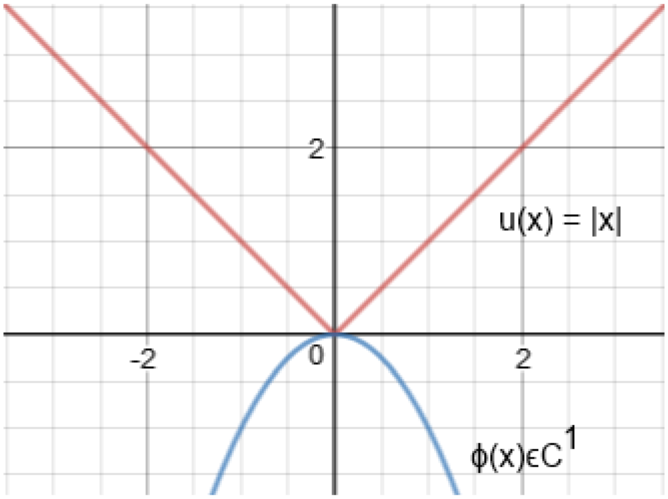

Figure 2.1 $u(x)$ (supersolution) is touched below by a smooth test function ϕ.

For many uniformly elliptic PDEs it can be proven that a comparison principle exists [37]. In order to make sense of this, we must define what it means to have a comparison principle.

Definition 2.10. A PDE has a **weak comparison principle** if given $u \in \text{USC}\left(\bar{X}\right)$ is a viscosity subsolution and $v \in \text{LSC}\left(\bar{X}\right)$ is a viscosity supersolution, if $u \leq v$ on ∂X then $u \leq v$ on $\bar{X}$ [37].

Additionally, we can derive uniqueness for the Dirichlet problem using a strong comparison principle which can be shown using the next two theorems.

Definition 2.11. A PDE has a **strong comparison principle** if given $u \in (\bar{X})$ is a viscosity subsolution and $v \in (\bar{X})$ is a viscosity supersolution, then $u \leq v$ on $\bar{X}$ [37].

Theorem 2.12. *If u is a subsolution and v is a supersolution, then $max(u - v)$ is the same on both $\bar{X}$ and ∂X [37].*

Theorem 2.13. *If u and v are viscosity solutions and $u = v$ on the boundary of X, then $u = v$ everywhere in $\bar{X}$ [37].*

The latter definition is the comparison principle required for much of the existing convergence framework. This differs from Definition 2.10 in that it does not require us to know *a priori* how u and v compare on the boundary, and it may apply to more general boundary conditions.

2.3 Optimal Transport Boundary Conditions

We consider a family of problems which are equipped with boundary conditions from optimal transport. These were initially used with the Monge-Ampére equation, but they are also relevant for a number of other problems. Normally, the Monge-Ampère optimal transportation problem is equipped with a condition on the sub-gradient of the solution:

$$\partial u(X) \subset Y \tag{2.9}$$

The Monge-Ampère PDE equipped with this condition is called the second boundary value problem for the Monge-Ampère equation [32]. One can express this as a nonlinear Neumann boundary condition. To do this, we require a defining function for the target set Y [16, 51], which should have the property that

$$H(y) = \begin{cases} < 0, & y \in Y \\ = 0, & y \in \partial Y \\ > 0, & y \notin \bar{Y}. \end{cases}$$

A natural choice is the signed distance function to the target boundary ∂Y. Note: (2.9) can be expressed as

$$H(\nabla u) \leq 0 \, [32]. \tag{2.10}$$

In this case, as in [3], we can rewrite the boundary condition as

$$H(\nabla u(x)) = 0, \quad x \in \partial X. \tag{2.11}$$

This simply requires that all points on the boundary of the domain X are mapped (via the gradient of u) onto the boundary of the target set Y. We know this is true for convex functions from the following lemma, proved in “On the Second Boundary Value Problem for Equations of Monge-Ampére Type” [51].

Lemma 2.14. Let $u \in C^2(\bar{X})$ be convex with X and Y convex. Then $\nabla u(\bar{X}) = \bar{Y}$ if and only if $\nabla u(\partial X) \subset \partial Y$ [51].

This and similar boundary conditions are relevant for a number of different PDEs including the Monge-Ampére equation.

As a model problem we focus on the problem of constructing minimal Lagrangian graphs, which can be recast as the following eigenvalue problem with nonlinear Neumann-type boundary conditions.

$$\begin{cases} F(D^2u(x)) + c = 0, & x \in X \\ H(\nabla u(x)) = 0, & x \in \partial X \end{cases} \tag{2.12}$$

where

$$F(D^2u(x)) = -\sum_{i=1}^{n} \arctan\left(\lambda_i(D^2u(x))\right), \tag{2.13}$$

$\lambda_i(D^2u(x))$ are the eigenvalues of the Hessian matrix $D^2u(x)$, and

$$H(\nabla u(x)) = \begin{cases} -\operatorname{dist}\left(\nabla u(x), \partial Y\right), & \nabla u(x) \in Y \\ 0, & \nabla u(x) \in \partial Y \\ \operatorname{dist}\left(\nabla u(x), \partial Y\right), & \nabla u(x) \notin \bar{Y}. \end{cases} \tag{2.14}$$

The eigenvalues $\lambda_i(D^2u(x))$ of the Hessian are the values such that $D^2u(x)y = \lambda y$ for all $y \in \mathbb{R}^{n\times 1}$.

From the convexity of Y, the signed distance function to its boundary is also convex, and thus, H is convex. We can rewrite H in terms of supporting hyperplanes to the convex target set

$$H(y) = \sup_{y_0 \in \partial Y} \{n(y_0) \cdot (y - y_0)\} \tag{2.15}$$

where $n(y)$ is the outward normal to ∂Y at y_0 [3]. By duality, this is equivalent to

$$H(y) = \sup_{||n||=1} \{n \cdot (y - y(n))\} \tag{2.16}$$

where $y(n)$ is the point on the boundary of Y with the normal n. Then if n is a unit outward normal to Y at y_0, the Legendre-Fenchel transform of $H(y)$ is

$$H^*(n) = \sup_{y \in \partial Y} \{n \cdot y - H(y)\} = \sup_{y \in \partial Y} \{n \cdot y\} = n \cdot y(n) \tag{2.17}$$

and we can rewrite the condition as

$$H(y) = \sup_{||n||=1} \{n \cdot y - H^*(n)\} \text{ [3]}. \tag{2.18}$$

This is equivalent to representing Y by supporting hyperplanes.

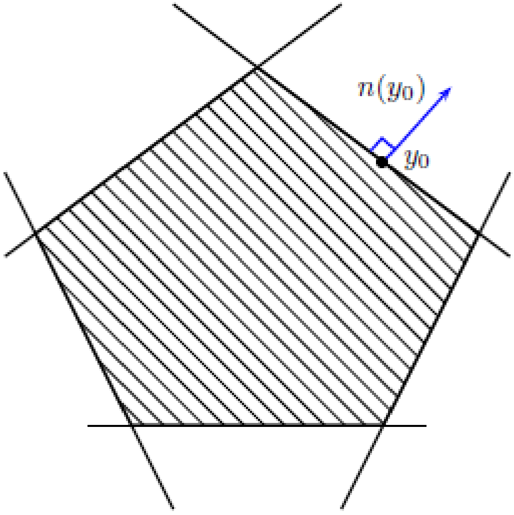

Figure 2.2 An example polyhedral target set with outward normal and (some) supporting hyperplanes.

Source: [4].

Theorem 2.15. *(Supporting Hyperplane Theorem) Let Y be a convex set and let $y_0 \in \partial Y$. Then y_0 has a (possibly non-unique) supporting hyperplane*

$$P = \{A(y) = 0 | A(y) \equiv n \cdot (y - y_0)\},$$

where $A(y) \leq 0$ whenever $y \in Y$ [4].

Lemma 2.16. Let $u \in C^2(\bar{X})$ be uniformly convex with X and Y convex. Then there exists $\ell > 0$ such that for all $x \in \partial X$

$$H(\nabla u(x)) = \max_{n \cdot n_x > \ell} \{\nabla u \cdot n - H^*(n)\} \tag{2.19}$$

From Section 2.3 of "Numerical Solution of the Second Boundary Value Problem for the Elliptic Monge-Ampère Equation" [4], we know that

$$n_x \cdot n_y = k\nabla H(\nabla u(x))^T D^2 u(x) \nabla H(\nabla u(x)),$$

where k is a normalization constant. This is positive for all x since $D^2u(x)$ is positive definite. Moreover, this is continuous on the compact set ∂X. Thus, it has a minimum, which must also be a positive value ℓ. From the same section in [4], we know that the maximum is attained when $n = n_y$. Since we know that $n_x \cdot n_y > \ell$, we can restrict the maximum to vectors n satisfying this constraint.

2.4 Theory of Convergence

We next turn our attention to the numerical solution of fully nonlinear PDEs and begin by reviewing existing convergence results that inspire the approach taken in this dissertation. In order to fully understand the convergence result derived by Barles and Souganidis [1], we must state the following definitions leading up to the theorem.

We consider a finite set of points $\mathcal{G}^h$ in $\bar{X}$ and focus on generalized finite difference approximations, which have the form:

$$F^h\left(x, u(x), u(x) - u(\cdot)\right) = 0 \quad x \in \mathcal{G}^h \tag{2.20}$$

where $F^h : \bar{X} \times \mathbb{R} \times B(\bar{X}) \to \mathbb{R}$ is locally bounded and $B(\bar{X})$ is the set of bounded functions on $\bar{X}$. The following definitions are given:

Definition 2.17 (Truncation error)**.** Define $\tau(h)$ to be the maximum **truncation error** of the scheme on the exact solution $u_{\text{ex}} \in C^2$ of (2.12):

$$\tau(h) = \max_{x \in \mathcal{G}^h} \left| F\left(D^2 u_{\text{ex}}(x)\right) - F^h\left(x, u_{\text{ex}}(x), u_{\text{ex}}(x) - u_{\text{ex}}(\cdot)\right)\right|.$$

Note that when smooth solution do not exist, we can alternately use Definition 2.18 instead.

Definition 2.18 (Truncation error)**.** The truncation error $\tau(h) > 0$ of the scheme is a quantity chosen so that for every smooth function ϕ,

$$\limsup_{h\to 0} \max_{x\in\mathcal{G}^h} \left(\frac{F^h(x, \phi(x), \phi(x) - \phi(\cdot)) - F(x, \nabla\phi(x), D^2\phi(x))}{\tau(h)} \right) < \infty \text{ [34].}$$

As above, the differences between a point and its neighbors which are used in the approximation are often denoted by $u(x) - u(\cdot)$ where $u(\cdot)$ represents the appropriate neighbor being used.

Definition 2.19. (Monotonicity): F^h is **monotone** if $F^h(x,t,u) \leq F^h(x,t,v)$ whenever $u \leq v \;\; \forall h \geq 0, x \in \bar{X}, t \in \mathbb{R}$, and $u, v \in B(\bar{X})$.

For example, let $F(x,u) = -u_{xx}(x)$. Then

$$F^h(x, u, u(x) - u(\cdot)) = -\frac{u(x+h) + u(x-h) - 2u(x)}{h^2}$$

is a monotone approximation since it is a non-decreasing function of the differences $u(x) - u(x+h)$ and $u(x) - u(x-h)$.

Remark 2.20. Montonicity is a discrete analogue of ellipticity.

Definition 2.21. (Stability): F^h is **stable** if $\forall h > 0$ there exists a solution $u^h \in B(\bar{X})$ and there exists some $M \in \mathbb{R}$ independent of h such that $\|u^h\|_\infty \leq M$.

Definition 2.22. (Consistency): F^h is **consistent** if $\forall x \in \bar{X}$ and $\forall \phi \in C_b^\infty(\bar{X})$

$$\limsup_{h\to 0, y \to x, \xi \to 0} F^h\left(y, \phi(y) + \xi, \phi(\cdot) + \xi\right) \leq F^*\left(x, \phi(x), \nabla\phi(x), D^2\phi(x)\right) \tag{2.21}$$

and

$$\liminf_{h\to 0, y \to x, \xi \to 0} F^h\left(y, \phi(y) + \xi, \phi(\cdot) + \xi\right) \geq F_*\left(x, \phi(x), \nabla\phi(x), D^2\phi(x)\right) \tag{2.22}$$

Monotone schemes automatically satisfy a discrete comparison principle, even if the underlying equation does not have a comparison principle.

Lemma 2.23 (discrete comparison principle [44, Theorem 5])**.** Let F^h be a monotone scheme and $F^h(x, u(x), u(x) - u(\cdot)) < F^h(x, v(x), v(x) - v(\cdot))$ for every $x \in \mathcal{G}^h$. Then $u(x) \leq v(x)$ for every $x \in \mathcal{G}^h$.

Montonicity, consistency, and stability, along with the PDE having a comparison principle, yield the convergence theorem:

Theorem 2.24. *Assume a scheme F^h is monotone, consistent, stable, and the underlying PDE has a strong comparison principle. Then, as $h \to 0$, the solution u^h converges locally uniformly to the unique continuous viscosity solution [1].*

This result gives a convergence proof for a wide class of schemes for nonlinear elliptic PDEs, which motivates the work of many in the field. However, often we cannot directly apply this result because many PDEs do not satisfy the required strong comparison principle result [4]. Additionally, the proof does not explain how to construct the required monotone schemes.

2.5 Numerical Methods for Nonlinear PDEs

In this section, we first describe some existing methods for solving 2nd order elliptic fully nonlinear PDEs. Then we discuss the most relevant methods in more detail, including one method for approximating directional derivatives on a rectangular grid. We then describe a monotone approximation of second directional derivatives on general point clouds in two dimensions and illustrate the advantage it has over uniform grids. We discuss the error in the approximation, how the scheme works, and why we expect convergence. In this dissertation, these approximations were used on convex sets, but they can be used with nonuniform meshes on complicated geometries as well. We show the approximations and give the conditions under which we expect a convergent method.

2.5.1 Existing Methods

Existing methods for solving 2nd order fully nonlinear elliptic PDEs can be divided into four categories: methods involving difference quotients, variational methods, finite basis expansions, or none of the above. Most methods fall into some combination of the first three. Some examples of each category are finite difference methods; Galerkin methods, finite element methods, least squares methods, spectral or discontinuous Galerkin methods; collocation or meshless methods; lattice Boltzmann, respectively. Variational formulations generally do not exist for fully nonlinear PDEs since the nonlinearity in the highest order term prevents shifting derivatives onto smooth test functions [18]. We briefly discuss some of the ideas behind a variety of methods and their advantages and disadvantages.

Geometric Methods These are a technique used for solving the Monge-Ampère equation which takes the geometric formulation of the problem and then discretizes that in order to solve the problem. The problem is reformulated by defining a measure for the Monge-Ampère operator which is dependent on the Lebesgue measure of the subdifferential of a Borel set, then finding when it is equal to a particular value depending on the boundary data. After that, it is straightforward to discretize the measure and the problem is to find a piecewise linear, continuous, and convex function u_N that takes the linear interpolant of g on the boundary such that the Monge-Ampère measure of u_N on each piece is equal to the prescribed measure of each piece [46].

Probabilistic Methods In many probabilistic methods such as those used for the Bellman equation, monotone finite difference schemes are created by approximating a controlled Markov diffusion process by a controlled Markov chain on a lattice. Then the dynamic programming equation leads to a discretized Bellman equation. Unfortunately, this method is computationally expensive [22].

Finite Difference Methods Standard finite difference methods can be used to approximate the Hessian by approximating each of its entries. Symmetric second difference stencils can be used to approximate the derivatives on the diagonal, and standard discretizations can be used for the mixed derivatives. These are easy to construct, but may not converge due to non-monotonicity. A variation on this approach is given by wide stencil methods, which directly approximate the eigenvalues of the Hessian. These are monotone and easier to analyze. The framework of Barles and Souganidis can be employed by monotone finite difference methods to prove the convergence of the wide stencil schemes introduced by Oberman. We will discuss wide stencil schemes in more detail in Chapter 4. These have been generalized to higher dimensions but retain the advantages and disadvantages described in Chapter 4 [44]. In "Discretization of the 3d Monge-Ampére operator, between Wide Stencils and Power Diagrams" [40], an alternate approach to producing wide stencil approximations was also used for the Monge-Ampére equation in three dimensions. In addition to these, convergent narrow stencil methods were proposed in "A Narrow-Stencil Finite Difference Method for Approximating Viscosity Solutions of Fully Nonlinear Elliptic Partial Differential Equations with Applications to Hamilton-Jacobi-Bellman Equations" [20], where they outline an alternate form of monotonicity, called generalized monotonicity. This is then used to build and analyze compact finite difference schemes for a class of fully nonlinear elliptic equations.

Finite Element Methods Finite element methods project the solution onto a finite-dimensional subspace. L^2 projection methods are one example of a finite element method. The idea is to pose the variational form $\int Fv = 0$ in a finite-dimensional setting with test functions and piecewise polynomials (such as finite element functions). Then the problem is broken into polyhedral cells, on which a linearized approximation to the solution is found. The linearization, along with the discretization, simplify the problem but introduce error. The advantage of these methods is that they make it easy to work with complicated geometries. The weakness of these methods is that often the weak formulations do not even make sense unless the original problem has a C^2 solution. An alternative method to these more traditional finite element methods is the two-scale finite element method proposed in “Two-Scale Method for the Monge-Ampère Equation: Convergence to the Viscosity Solution” [42]. This inherits the flavor of wide stencil finite difference methods, and is used to produce convergent piecewise linear approximations to the Monge-Ampére equation.

Augmented Lagrangian Method The augmented Lagrangian method is to solve a minimization problem with Lagrange multipliers. By recasting a fully nonlinear PDE into minimization problem, one can then use conjugate gradient or another algorithm to solve it. Some of the most common algorithms used are ALG2 or the closely related alternating direction methods of multipliers (ADMM) [14, 15].

Least Squares Methods Least squares methods are similar to the above method in that they solve a minimization problem. For example, $F\left(D^2u\right) = g$ could be converted to the problem minimize $j(u, g)$ where

$$j(u, g) = \frac{1}{2}\int_X |F\left(D^2u\right) - g|^2 dx \tag{2.23}$$

Remark 2.25. There is no analysis on if or when this is equivalent to other notions of weak solutions.

By minimizing this instead of directly solving the problem, we are able to capture the smooth convex solutions whenever they exist. In situations where there are no convex solutions, this yields the closest convex approximation which is good enough for many applications where the boundary matters more than the interior [14].

Vanishing Moment Method The vanishing moment method is a more general method for simplifying problems and can be paired with a variety of techniques. It follows and extends the idea of the vanishing viscosity method from first order fully nonlinear PDEs. The idea is to try to approximate the fully nonlinear PDE by a higher order quasi-linear PDE, making the PDE easier to solve and still converging to the original PDE solution. This is accomplished by adding a small multiple of the bilaplacian to the equation, then exploiting existing methods for fourth-order quasi-linear equations. Typically, this requires the addition of an artificial boundary condition, which can lead to boundary layers. Convergence results are not available except in a few simple cases where there is high regularity, and or radial symmetry [21].

CHAPTER 3

NONLINEAR EIGENVALUE PROBLEMS

3.1 Example Problems

The results described in Chapter 2 apply to a wide class of degenerate elliptic PDEs. Here, we discuss the type of problem being solved in this research. Specifically, the problems we solve are nonlinear eigenvalue problems with transport type boundary conditions. These are fully nonlinear elliptic partial differential equations, equipped with an extra unknown constant multiplier c, the eigenvalue. The boundary conditions are nonlinear and involve the re-configuring of the domain into some target set via a map defined by the gradient of the solution. Some examples include the Monge-Ampère equation, Pucci's maximal and minimal equations, the equation for the convex envelope, some obstacle problems, and the equation for a minimal Lagrangian submanifold [45, 26, 7]. The general form of these problems is

$$\begin{cases} F(x, \nabla u, D^2 u) = c\, f(x, \nabla u) & x \in X \\ H(\nabla u) = 0 & x \in \partial X. \end{cases} \tag{3.1}$$

Note that the problem cannot be well-posed since all expressions are in terms of derivatives of u, so at best it is unique up to additive constants. Moreover, there is typically a solvability condition that uniquely determines the eigenvalue c for which we get existence of solutions. Symbolically, some examples of these problems are:

$$\lambda_1 + \lambda_2 = c\, f \tag{3.2}$$

the equation for the minimal Lagrangian submanifold,

$$\tan^{-1}(\lambda_1) + \tan^{-1}(\lambda_2) = c \tag{3.3}$$

the Monge-Ampère equation,

$$\det(D^2u) = c\, f \tag{3.4}$$

or more generally in optimal transport

$$\det(\nabla T(x, u(x), \nabla u(x))) = c\, f(x, u(x), \nabla u(x)) \tag{3.5}$$

where T can be a much more complicated mapping [45].

3.2 Eigenvalue Problem for a PDE

The problem given by Equation (3.13) and the boundary condition from Equation (3.1), which we consider in this dissertation, is an example of an eigenvalue problem for a fully nonlinear elliptic operator. This equation is known to have a unique classical solution from “A Boundary Value Problem for Minimal Lagrangian Graphs” [7]. Abstractly, the problem statement is to find $u \in C^2(X) \cap C^1(\bar{X})$ and $x \in \mathbb{R}$ such that

$$\begin{cases} F(x, \nabla u(x), D^2u(x)) + cG(x, \nabla u(x)) = 0, & x \in X \\ H(x, \nabla u(x)) = 0, & x \in \partial X. \end{cases} \tag{3.6}$$

Remark 3.1. The solution is at best unique only up to additive constants.

In fact, this formulation of the problem is intricately connected to the solvability of a related PDE. As an example, we consider the Neumann problem for Poisson’s equation.

$$\begin{cases} \Delta u = f, & x \in X \\ \frac{\partial u}{\partial n} = g, & x \in \partial X \end{cases} \tag{3.7}$$

For a solution to exist, data must satisfy the solvability condition

$$\iiint_X f(x)\, dV = \iint_{\partial X} g(x)\, dS.$$

Often, the data f and g arising in applications are susceptible to noise, measurement error, etc. This can lead to a failure in the solvability condition. One approach to ensuring solvability in this case is to interpret this as an eigenvalue problem by introducing a constant c and solving

$$\begin{cases} \Delta u = cf, & x \in X \\ \frac{\partial u}{\partial n} = g, & x \in \partial X. \end{cases} \tag{3.8}$$

for the unknown pair (u, c).

The new solvability condition is

$$c \iiint_X f(x)\, dV = \iint_{\partial X} g(x)\, dS. \tag{3.9}$$

The solution of the eigenvalue problem (3.8) will select a value of c that satisfies this condition and forces the problem to be solvable. If f and g are close to satisfying the solvability condition, then the solution will choose $c \approx 1$ and produce a solution to a PDE close to the original (3.7), with the error due to errors in the input data f, g.

A similar issue arises in the solution of the second boundary value problem for the Monge-Ampère equation, which arises in the context of optimal transport.

$$\begin{cases} -g(\nabla u(x)) \det(D^2 u(x)) + f(x) = 0, & x \in X \\ u \text{ is convex} \\ \nabla u(X) \subset \bar{Y}. \end{cases} \tag{3.10}$$

This problem has a solution only if the following mass balance condition is satisfied,

$$\int_X f(x)\, dx = \int_Y g(y)\, dy. \tag{3.11}$$

However, in many applications (e.g., image processing [29], seismic full waveform inversion [17], mesh generation [8], etc.) the data is not expected to naturally satisfy the solvability condition. A proposed solution is to view the equation as an eigenvalue problem and seek a pair (u, c) satisfying

$$\begin{cases} -g(\nabla u(x)) \det(D^2u(x)) + cf(x) = 0, & x \in X \\ u \text{ is convex} \\ \nabla u(X) \subset \bar{Y}. \end{cases} \tag{3.12}$$

In fact, even when data does satisfy the relevant solvability condition, consistent discretizations of Equation (3.7) or Equation (3.10) cannot be expected to inherit this solvability.

To illustrate this, consider Poisson's equation in one dimension with Neumann boundary conditions:

$$\begin{cases} u'' = f(x) \equiv \cos\left(\frac{\pi}{2}x\right) & x \in (0,1) \\ u' = g(x) \equiv \frac{2}{\pi}\sin\left(\frac{\pi}{2}x\right) & x = 0, 1 \end{cases}$$

This has a solution, which is unique up to additive constants, since the data satisfies the solvability condition

$$\int_0^1 f(x)dx = g(1) - g(0).$$

Now consider the uniform grid $x_j = jh$, $j = 0, \ldots, N$ and discretize the equation using standard centered differences for the second derivative and a one-sided difference for the boundary condition. It is not hard to check that the resulting linear system has a solution only if the following discrete solvability condition is satisfied [38]:

$$h\sum_{j=1}^{N-1} f(x_j) = g(x_N) - g(x_0).$$

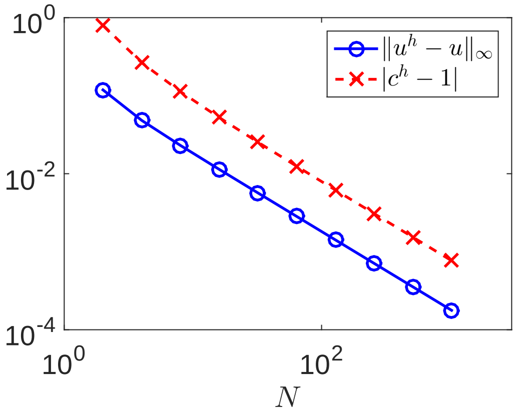

Figure 3.1 Discrete solution to Poisson's equation when viewed as an eigenvalue problem.

This is a natural discrete analogue of the continuous solvability condition, but it is not exactly satisfied at the discrete level and thus, the discrete problem fails to have a solution.

As an alternative, we view the Poisson equation as the following eigenvalue problem.

$$\begin{cases} u'' = cf(x) & x \in (0,1) \\ u' = g(x) & x = 0,1 \end{cases}$$

We discretize as before, including the eigenvalue c as an additional unknown, and supplementing the linear system with an additional equation $u(x_0) = 0$ in order to select a unique solution. This time, the discrete problem has a solution u^h with corresponding eigenvalue c^h. We verify that both $u^h \to u$ and $c^h \to 1$ as the grid is refined, so that the limiting problem is the original Poisson equation. See Figure 3.1.

There is certainly a need for numerical methods and convergence analysis that can be applied to eigenvalue problems for fully nonlinear elliptic equations. In addressing this issue for the construction of minimal Lagrangian graphs, we also begin the development of a framework for solving many other important nonlinear PDEs. While both Poisson's equation and the Monge-Ampére equation have known solvability conditions, the solvability condition for the problem we study is not known *a priori*.

3.2.1 The Equation for the Minimal Lagrangian Submanifold

The equation we use as a model problem in our study comes from Brendle and Warrens “A Boundary Value Problem for Minimal Lagrangian Graphs" [7]. In that paper, it is shown that there exists a diffeomorphism $f : X \to Y$ such that the graph $\Sigma = \{(x, f(x)) : x \in X\}$ is a minimal Lagrangian submanifold of $\mathbb{R}^n \times \mathbb{R}^n$,where X and Y are uniformly convex domains in $\mathbb{R}^n$ with smooth boundary. The problem is reduced to the solvability of the following nonlinear PDE,

$$F(D^2u(x)) = \sum_{j=1}^{n} \tan^{-1}(\lambda_j(D^2u)) = c \tag{3.13}$$

equipped with the transport boundary condition (2.9), where the $\lambda_j(D^2u)$ are the eigenvalues of $D^2u(x)$ and n is the number of dimensions. There are existence proofs for a similar boundary condition to that of the second boundary value problem for the Monge-Ampère equation [7]. Most precisely, the problem states

Find a convex function $u : X \to \mathbb{R}$ and a constant $c \in (0, \frac{n\pi}{2})$ such that ∇u is a diffeomorphism from X to Y and $F(D^2u(x)) = c$ for all $x \in X$ [7].

There are several challenges in solving this problem. The solvability condition, which may not be known *a priori*, may not be satisfied exactly because of noise in the data. Also, even if we know the solvability condition for the continuous problem, it may not be exactly equivalent to solvability of discretized problem.

3.2.2 Reformulation of the PDE

We begin by proposing a reformulation of the PDE (3.1), which will allow us to build more stability into our numerical schemes. Moreover, we demonstrate that viscosity solutions of this new equation (with the eigenvalue c_{ex} fixed) are equivalent to classical solutions of the original problem.

We remark first of all that solutions to the second boundary condition (2.9) will trivially satisfy *a priori* bounds on the solution gradient. That is, choose any $R > \max\{|p| \mid p \in \partial Y\}$ and let u satisfy the second boundary condition (2.9). Then

$$|\nabla u(x)| < R \tag{3.14}$$

for all $x \in \bar{X}$.

We also recall that any smooth convex solution of the second boundary condition, reformulated as in (2.11), will satisfy the constraints

$$\begin{aligned} -\lambda_1(D^2u(x)) &\leq 0 \\ H(\nabla u(x)) &\leq 0 \end{aligned} \tag{3.15}$$

for every $x \in X$. Here $\lambda_1(M)$ denotes the smallest eigenvalue of the symmetric positive definite matrix M.

We propose combining all of these constraints into a new PDE

$$\max\left\{F(D^2u(x)) + c_{\text{ex}}, -\lambda_1(D^2u(x)), H(\nabla u(x)), |\nabla u(x)| - R\right\} = 0, \quad x \in X. \tag{3.16}$$

We remark that this equation is posed only in the interior of the domain, and boundary conditions will not be required to select a unique (up to additive constants) solution. We also note that in the above equation, the eigenvalue c_{ex} will be interpreted as a known quantity.

Theorem 3.2 (Equivalence of PDEs)**.** Let $u : \bar{X} \to \mathbb{R}$ be continuous and $c_{\text{ex}} \in (0, n\pi/2)$ be the unique eigenvalue of (2.12). Then (u, c_{ex}) is a classical solution of (2.12) if and only if u is a viscosity solution of (3.16).

Proof. This result is an immediate consequence of Lemmas 3.3-3.4, proved below. □

Lemma 3.3 (Classical implies viscosity)**.** Let (u, c_{ex}) be a classical solution of (2.12). Then u is a viscosity solution of (3.16).

Proof. We remark that u trivially satisfies the constraints (3.14)-(3.15). Since additionally

$$F(D^2u(x)) + c = 0,$$

it is certainly true that the modified equation in Equation (3.16) holds in the classical sense. It is a simple consequence that (3.16) will also hold in the viscosity sense [11]. □

Lemma 3.4 (Viscosity implies classical)**.** Let $u : \bar{X} \to \mathbb{R}$ be continuous and $c_{\text{ex}} \in (0, n\pi/2)$ be the unique eigenvalue for (2.12). If u is a viscosity solution of (3.16) then (u, c_{ex}) is a classical solution of (2.12).

Proof. Let u_{ex} be any classical solution of (2.12). From [7], this is uniquely determined up to an additive constant.

We remark first of all that u is a viscosity subsolution of the equation

$$-\lambda_1(D^2u(x)) = 0.$$

From [43, Theorem 1], u is convex.

We also observe that u is a convex viscosity subsolution of the equation

$$H(\nabla u(x)) = 0.$$

From [31, Lemma 2.5], the subgradient of u satisfies

$$\partial u(X) \subset \bar{Y}.$$

As u is continuous up to the boundary, a consequence of this is that

$$\partial u(x) \cap \bar{Y} \neq \phi \tag{3.17}$$

for every $x \in \partial X$.

We now assume that $u - u_{\text{ex}}$ is not a constant and show that this leads to a contradiction. Since u_{ex} is a viscosity solution of the constrained PDE (3.16), it is also a subsolution of the uniformly elliptic component

$$F(D^2u(x)) + c_{\text{ex}} \leq 0.$$

Since u_{ex} is a classical solution of

$$F(D^2u(x)) + c_{\text{ex}} = 0,$$

it is also a viscosity solution [11] and a viscosity supersolution.

From [36, Theorem 3.1], the maximum of $u - u_{\text{ex}}$ must be attained at some point $x_0 \in \partial X$. Moreover, by a nonlinear version of the Hopf boundary lemma [39], we have that

$$\frac{\partial(u - u_{\text{ex}})(x_0)}{\partial n} > 0$$

for any exterior direction n satisfying $n \cdot n_x(x_0) > 0$. That is, taking any $p \in \partial u(x_0)$, we must have

$$(p - \nabla u_{\text{ex}}(x_0)) \cdot n > 0.$$

Now we consider in particular the choice of $n = \nabla H(\nabla u_{\text{ex}}(x_0))$, which does satisfy the requirement $n \cdot n_x(x_0) > 0$ as in Lemma 2.16. Hence,

$$(p - \nabla u_{\text{ex}}(x_0)) \cdot \nabla H(\nabla u_{\text{ex}}(x_0)) > 0$$

On the other hand, since H is convex, we know that

$$H(p) \geq H(\nabla u_{\text{ex}}(x_0)) + \nabla H(\nabla u_{\text{ex}}(x_0)) \cdot (p - \nabla u_{\text{ex}}(x_0)) > H(\nabla u_{\text{ex}}(x_0)) = 0.$$

The condition $H(p) > 0$ implies p is outside $\bar{Y}$ for any $p \in \partial u(x_0)$, which contradicts (3.17).

We conclude that $u - u_{\text{ex}}$ must be constant on X. Since the classical solution of (2.12) is unique up to additive constants, u is a classical solution. □

3.3 Numerical Approach

Our goal in this chapter is to find a framework for solving well-posed eigenvalue problems numerically in order to correctly approximate both the eigenvalue and the solution of the PDE. In order to build convergent methods for the eigenvalue problem in Equation (3.13), we wish to build upon recent developments in the approximation of fully nonlinear elliptic equations.

Classically, the convergence of numerical methods for linear equations is established via the Lax-Equivalence Theorem. Roughly speaking, a consistent, stable method will converge to the solution of the continuous equation. However, this does not immediately yield convergent methods for fully nonlinear equations for a couple of reasons. First, establishing the existence and stability of solutions to a discrete method can be a delicate problem in the case of nonlinear equations and secondly, it does not apply when the equation does not have classical solutions. Establishing error estimates for smooth solutions by linearizing the problem can be done fairly easily for Dirichlet boundary conditions, but is much harder for pure Neumann conditions.

A powerful contribution to the numerical approximation of elliptic equations was provided by the Barles-Souganidis framework, which states that the solution to a scheme that is consistent, monotone, and stable will converge to the viscosity solution, provided the underlying PDE satisfies a comparison principle [1]. In this setting, the Barles-Souganidis framework does not apply since we do not have a comparison principle. Nevertheless, it is still desirable to consider consistent, monotone schemes since they are more amenable to analysis.

In this dissertation, we consider finite difference schemes that have the form

$$F^h(x, u(x), u(x) - u(\cdot)) = 0, \quad x \in \mathcal{G}^h \tag{3.18}$$

where $u : \mathcal{G}^h \to \mathbb{R}$ is a grid function and $\mathcal{G}^h \subset \bar{X}$ is a set of discretization points, which can be a finite difference grid or a more general point cloud. Here h is a small parameter relating to the grid resolution. In particular, we expect that as $h \to 0$, the domain becomes fully resolved in the sense that

$$\lim_{h\to 0} \sup_{y\in X} \min_{x\in\mathcal{G}^h} |x - y| = 0. \tag{3.19}$$

The computational and convergence framework we employ involves a two-step approach. Let us first suppose that we have discrete approximations F^h, H^h, E^h, L^h of the PDE operators $F(D^2u), H(\nabla u), |\nabla u|, -\lambda_1(D^2u)$. The details of these discrete operators will be explained in the following subsections. These are assumed to have a maximum truncation error of $\tau(h)$ as defined in Definition 2.17. We also let $x_0 \in \bar{X}$ be any fixed point in the domain and choose a sequence $x_0^h \in \mathcal{G}^h$ such that $x_0^h \to x_0$. Finally, we choose some $\kappa(h) \geq 0$.

We now employ a two-step procedure to solve for an approximation (u^h, c^h) to the true solution $(u_{\text{ex}}, c_{\text{ex}})$. Existence can be shown using Perron's method.

1. Solve the discrete system

$$\begin{cases} F^h(x, v^h(x) - v^h(\cdot)) + c^h = 0, & x \in \mathcal{G}^h \cap X \\ H^h(x, v^h(x) - v^h(\cdot)) = 0, & x \in \mathcal{G}^h \cap \partial X \\ v^h(x_0^h) = 0 \end{cases} \tag{3.20}$$

for the grid function v^h and scalar c^h.

2. Solve the discrete system

$$\begin{cases} \max\big\{F^h(x, w^h(x) - w^h(\cdot)) + c^h, L^h(x, w^h(x) - w^h(\cdot))\,, \\ \qquad H^h(x, w^h(x) - w^h(\cdot)), E^h(x, w^h(x) - w^h(\cdot)) - R\big\} = 0, & x \in \mathcal{G}^h \cap X \\ \max\{H^h(x, w^h(x) - w^h(\cdot)) + \kappa(h) w^h(x), \\ \qquad E^h(x, w^h(x) - w^h(\cdot)) - R\} = 0, & x \in \mathcal{G}^h \cap \partial X \end{cases} \tag{3.21}$$

for the grid function w^h and set

$$u^h(x) = w^h(x) - w^h(x_0^h). \tag{3.22}$$

We remark that while the second step is important for the convergence analysis, we do not find it necessary to solve this second system in practice. Instead, we typically find that the solution v^h obtained in step 1 automatically satisfies the second system with $\kappa(h) = 0$. If this does not occur, solving the second system (3.21) becomes necessary. In that case, we should choose $\kappa(h) > 0$ to guarantee existence of a solution. This relaxation of the boundary condition is needed since the solvability conditions for Equations (3.20) and (3.21) may differ slightly. We do not *a priori* know the solvability conditions, so this is necessary. With $\kappa(h) > 0$, existence follows from a discrete version of Perron's method [30].

We also observe that the final candidate solution u^h that we compute satisfies the scheme

$$\begin{aligned} \max \{ \; & F^h(x, u^h(x) - u^h(\cdot)) + c^h, L^h(x, u^h(x) - u^h(\cdot)), \\ & H^h(x, u^h(x) - u^h(\cdot)), E^h(x, u^h(x) - u^h(\cdot)) - R \big\} = 0 \end{aligned} \tag{3.23}$$

at interior points $x \in \mathcal{G}^h \cap X$ and satisfies the inequality

$$E^h(x, u^h(x) - u^h(\cdot)) - R \leq 0 \tag{3.24}$$

at all points $x \in \mathcal{G}^h$.

The approximation schemes will have to satisfy consistency and monotonicity conditions in order to fit within the requirements of our ultimate convergence theorems (Theorems 3.5-3.6), with some additional structure built into the discrete Eikonal operator E^h.

The additional condition $|\nabla u| - R$ in Equation (3.16) must be discretized in a monotone way.

A choice of monotone approximation for $|\nabla u|$ that is convenient for the convergence analysis involves characterizing this as the maximum possible first directional derivative,

$$|\nabla u(x)| = \max_{|\nu|=1} \frac{\partial u(x)}{\partial \nu}.$$

Then a simple choice of discretization involves looking at all possible directions that can be approximated exactly within our search radius r.

$$|\nabla u| \approx E^h(x, u(x) - u(\cdot)) = \max \left\{ \frac{u(x) - u(y)}{|x - y|} \mid y \in \mathcal{G}^h \cap B(x, r) \right\}. \tag{3.25}$$

3.4 Convergence Analysis

In this section, we present a proof of convergence for the discrete eigenvalue problem in Equation (3.20). We separate this into two results: convergence of the eigenvalue c^h and convergence of the grid function u^h.

Theorem 3.5 (Convergence of the eigenvalue)**.** Let $(u_{\text{ex}}, c_{\text{ex}})$ be any smooth classical solution of the eigenvalue problem given by Equation (3.13) and Equation (2.9), and let (v^h, c^h) be any solution of the scheme (3.20). Then c^h converges to c_{ex} as $h \to 0$.

Theorem 3.6 (Convergence of the grid function)**.** Let $(u_{\text{ex}}, c_{\text{ex}})$ be a solution of the eigenvalue problem given by Equation (3.13) and Equation (2.9) satisfying $u_{\text{ex}}(x_0) = 0$ and let u^h be any solution of the scheme (3.21)-(3.22). Then $\tilde{u}^h$ converges uniformly to u_{ex} as $h \to 0$.

We also remark that, while our focus here is the construction of minimal Lagrangian graphs, this analysis could be readily adapted to more general eigenvalue problems of the form (3.1).

3.4.1 Convergence of the Eigenvalue

We begin by establishing convergence of the eigenvalue (Theorem 3.5). The proof of this result will require several short lemmas. In these we will use the shorthand notation

$$
\begin{aligned}
F_i^h[u] &= F^h(x_i, u(x_i) - u(\cdot)) \\
H_i^h[u] &= H^h(x_i, u(x_i) - u(\cdot)).
\end{aligned}
$$

We also define the following objects relating to sub- and super-solutions of the schemes.

$$U_c^h = \{u \mid F_i^h[u] + c \leq 0, x_i \in \mathcal{G}^h \cap X; H_i^h[u] < 0, x_i \in \mathcal{G}^h \cap \partial X\} \tag{3.26}$$

$$V_c^h = \{v \mid F_i^h[v] + c \geq 0, x_i \in \mathcal{G}^h \cap X; H_i^h[v] > 0, x_i \in \mathcal{G}^h \cap \partial X\} \tag{3.27}$$

We begin by establishing that these sets of sub(super)-solutions are non-empty for appropriate choices of c.

Lemma 3.7 (existence of sub(super) solutions)**.** There exist $u_+^h \in V^h_{c_{\text{ex}}+\omega(h)}$, $u_-^+h \in U_{c_{\text{ex}}-\omega(h)}$ where $\omega(h)$ is proportional to the maximum consistency error $\tau(h)$ of the scheme.

Proof. We begin by letting $G(x)$ be the signed distance function to the boundary of the domain ∂X. Through convolution with an appropriate mollifier, we obtain a smooth function w with the property that

$$\nabla w(x) = n_x, \quad x \in \partial X.$$

For some $\epsilon > 0$ (which will be fixed) we define

$$u_-^h = u_{\text{ex}} - \epsilon w, \quad u_+^h = u_{\text{ex}} + \epsilon w.$$

We will show that for suitable choices of $\epsilon = \epsilon(h)$ and $\omega(h) = \mathcal{O}(\tau(h))$ we have $u_-^h \in U^h_{c_{\text{ex}}-\omega(h)}$. The argument regarding u_+^h is similar.

Note that from Lemma (2.16), if $x \in \partial X$ then

$$\begin{aligned} H(\nabla u_-^h(x)) &= \sup_{n \cdot n_x > \ell} \{\nabla u_-^h(x) \cdot n - H^*(n)\} \\ &\leq \sup_{n \cdot n_x > \ell} \{\nabla u_{\text{ex}}(x) \cdot n - H^*(n)\} - \epsilon \ell \\ &= H(\nabla u_{\text{ex}}(x)) - \epsilon \ell \\ &= -\epsilon \ell. \end{aligned}$$

By consistency we have

$$H_i^h[u_-^h] \leq H(\nabla u_-^h(x_i)) + \tau(h) \leq -\epsilon \ell + \tau(h).$$

Choosing $\epsilon > \frac{1}{\ell}\tau(h)$, we obtain

$$H^h[u_-^h] < 0.$$

Since our PDE operator F is Lipschitz, we can also find some $L > 0$ so that

$$F(D^2 u_-^h(x)) \leq F(D^2 u_{\text{ex}}(x)) + L\epsilon = -c_{\text{ex}} + L\epsilon.$$

By consistency we have

$$F_i^h[u_-^h] \leq F(D^2 u_-^h(x_i)) + \tau(h) \leq -c_{\text{ex}} + L\epsilon + \tau(h).$$

Again choosing $\epsilon > \frac{1}{\ell}\tau(h)$ and defining $\omega(h) = L\epsilon(h) + \tau(h)$ we have

$$F^h[u_-^h] + c_{\text{ex}} - \omega(h) \leq 0.$$

We conclude that $u_-^h \in U^h_{c_{\text{ex}} - \omega(h)}$. □

Now using the discrete comparison principle, we can begin to see how the sets of sub(super)-solutions are related to each other, which will lead ultimately to constraints on our numerically computed eigenvalue.

Lemma 3.8 (Comparison of eigenvalues)**.** Suppose $u_1 \in U^h_{c_1}$ and $u_2 \in V^h_{c_2}$. Then $c_1 \leq c_2$.

Proof. Suppose instead that $c_1 > c_2$. Note that for any constant k we also have $u_1 + k \in U^h_{c_1}$. Thus, we can assume that $u_1 > u_2$. Now we estimate

$$F^h_i[u_1] + c_2 < F^h_i[u_1] + c_1 \leq 0 \leq F^h_i[u_2] + c_2, \quad x_i \in \mathcal{G}^h \cap X$$

and

$$H^h_i[u_1] < 0 < H^h_i[u_2], \quad x_i \in \mathcal{G}^h \cap \partial X.$$

By the discrete comparison principle (Lemma 2.23) we have $u_1 \leq u_2$, a contradiction. □

With these lemmas in place, we can now prove convergence of the numerically computed eigenvalue.

Proof of Theorem 3.5. Recall that

$$F^h_i[v^h] + c^h = 0, \quad x_i \in \mathcal{G}^h \cap X$$

and

$$H^h_i[v^h] = 0, \quad x_i \in \mathcal{G}^h \cap \partial X.$$

Following Lemmas 3.7-3.8 we conclude that

$$c_{\text{ex}} - \omega(h) \leq c^h \leq c_{\text{ex}} + \omega(h).$$ □

Having proved the convergence of the eigenvalue $c^h \to c_{\text{ex}}$, this reduces our task from the convergence of an eigenvalue problem to the convergence of a fully nonlinear elliptic PDE.

3.4.2 Convergence of the Grid Function

We now turn our attention to the convergence of the approximation u^h to the solution u_{ex} (Theorem 3.6).

In order to prove this theorem, we first need to construct a piecewise linear extension $\tilde{u}^h$ of the grid function u^h.

We assume the existence of a triangulation T^h of $\mathcal{G}^h$ such that

1. The maximal angle in any triangle is bounded uniformly away from π, independently of h.
2. The diameter of each triangle is $\mathcal{O}(h)$.

We note that since $\mathcal{G}^h \subset \bar{X}$, we need to extend triangles that intersect the boundary in order to obtain a decomposition $\tilde{T}^h$ that fully covers the domain. To do this, we define the regions $\tilde{t}_i$ as follows:

Definition 3.9 (Extension of triangulation)**.** Let $t \in T^h$, with the nodes x_0, x_1, x_2. Then we define the corresponding region $\tilde{t} \in \tilde{T}^h$ as follows:

(a) If at least two nodes of t are in X, set

$$\tilde{t} = t.$$

(b) If two nodes $x_1, x_2 \in \partial X$, set

$$\tilde{t} = \text{Conv}\{x_0, x_0 + 2(x_1 - x_0), x_0 + 2(x_2 - x_0)\} \cap \bar{X}.$$

where Conv denotes the convex hull of the set.

We remark that

$$\bigcup_{\tilde{t} \in \tilde{T}} \tilde{t} = \bar{X}.$$

Now we are able to define a continuous piecewise linear extension.

Definition 3.10 (Extension of grid function)**.** Define the unique continuous piecewise linear function $\tilde{u}^h$ satisfying:

(a) $\tilde{u}^h(x) = u^h(x)$ for all $x \in \mathcal{G}^h$.

(b) $\tilde{u}^h(x)$ is a linear function on each region $\tilde{t} \in \tilde{T}$.

We remark that $\tilde{u}^h$ will also satisfy the approximation scheme (3.22)-(3.24). An important element to our convergence proof will be to establish uniform Lipschitz bounds on the approximations $\tilde{u}^h$. Note that the search radius for the neighbors is assumed to be $\mathcal{O}(\sqrt{h})$ since this is what the schemes defined in Chapter 4 will be using.

Lemma 3.11 (Lipschitz bounds)**.** There exists a constant $L > 0$ such that the Lipschitz constant of $\tilde{u}^h$ is bounded by L for all sufficiently small $h > 0$.

Proof. We begin by considering the function $\tilde{u}^h$ restricted to some fixed region $\tilde{t} \in \tilde{T}^h$. Let x_0, x_1, x_2 be the nodes of $\tilde{t}$. Without loss of generality, we can assume that the maximal interior angle θ of $\tilde{t}$ occurs at the node x_0.

Now we know that $x_i \in \mathcal{G}^h$ for $i = 0, 1, 2$. Since $\tilde{u}^h$ satisfies the scheme (3.24), we know that

$$E^h(x_i, \tilde{u}^h(x_i) - \tilde{u}^h(\cdot)) - R \leq 0.$$

From the definition of E^h (3.25), we can conclude that

$$\tilde{u}^h(x_i) \leq \tilde{u}^h(y) + R|x_i - y|$$

for every $y \in \mathcal{G}^h \cap B(x_i, r)$. In particular, this holds for $y = x_0, x_1, x_2$ since the diameter of $\tilde{t}$ is bounded by $2h < r = \mathcal{O}(\sqrt{h})$ for small enough $h > 0$. Thus, we obtain the discrete Lipschitz bounds

$$|\tilde{u}^h(x_i) - \tilde{u}^h(x_j)| \leq R|x_i - x_j|, \quad i, j \in \{0, 1, 2\}.$$

We now use this to bound the gradient of $\tilde{u}^h$ over the region $\tilde{t}$. Notice that for $x \in \tilde{t}$ we can write

$$\tilde{u}^h(x) = \tilde{u}^h(x_0) + p \cdot (x - x_0)$$

where $p = \nabla \tilde{u}^h(x)$ is constant over this region. Since $x_1 - x_0$ and $x_2 - x_0$ span $\mathbb{R}^2$, we can find constants $a_1, a_2 \in \mathbb{R}$ such that

$$p = a_1(x_1 - x_0) + a_2(x_2 - x_0).$$

Now we use our discrete Lipschitz bounds to compute

$$R|x_1 - x_0| \geq |u_1 - u_0| = |x_1 - x_0||a_1|x_1 - x_0| + a_2|x_2 - x_0|\cos\theta|.$$

Simplifying and applying the bound on the maximal angle, we obtain

$$R \geq |a_1||x_1 - x_0| - M|a_2||x_2 - x_0|.$$

Similarly,

$$R \geq |a_2||x_2 - x_0| - M|a_1||x_1 - x_0|.$$

Combining the two above expressions, we find that

$$R(M+1) \geq (1 - M^2)|a_2||x_2 - x_0|$$

and thus

$$|a_2| \leq \frac{R}{(1-M)|x_2 - x_0|}.$$

An equivalent bound is available for $|a_1|$.

Now we can bound $p = \nabla \tilde{u}^h(x)$ over the region $\tilde{t}$ by

$$|p| \leq |a_1||x_1 - x_0| + |a_2||x_2 - x_0| \leq \frac{2R}{1-M} \equiv L.$$

Since $\tilde{u}^h$ is piecewise linear, its Lipschitz constant will be bounded by the maximum Lipschitz constant over each region $\tilde{t} \in \tilde{T}^h$, which is given by L. □

An immediate consequence of this is uniform bounds for $\tilde{u}^h$.

Lemma 3.12. There exists a constant $C > 0$ such that $\|\tilde{u}^h\|_\infty \leq C$ for all sufficiently small $h > 0$.

Proof. Since $\tilde{u}^h(x_0^h) = 0$ and $\tilde{u}^h$ has a bounded Lipschitz constant (Lemma 3.11), we have that

$$\begin{aligned}
|\tilde{u}^h(x)| &= |\tilde{u}^h(x) - \tilde{u}^h(x_0)| \\
&\leq L|x - x_0| \\
&\leq L \operatorname{diam}(X)
\end{aligned}$$

for every $x \in \bar{X}$. □

Next we adapt the usual Barles-Souganidis convergence proof [1] to begin to show how we can obtain viscosity solutions to (3.16) from our approximation scheme (3.21).

Lemma 3.13. Let h_n be any sequence such that $h_n \to 0$ and $\tilde{u}^{h_n}$ is uniformly continuous and converges uniformly to a continuous function v. Then v is a viscosity solution of (3.16).

Proof. We first demonstrate that v is a viscosity subsolution. Consider any $x_0 \in X$ and $\phi \in C^2$ such that $v - \phi$ has a strict local maximum at x_0 with $v(x_0) = \phi(x_0)$. Because $\tilde{u}^h$ and the limit function v are continuous, there exist sequences $y_n \in X$ and $z_n \in \mathcal{G}^h \cap X$ such that

$$y_n \to x_0, \quad |y_n - z_n| < h, \quad \tilde{u}^{h_n}(y_n) \to v(x_0)$$

where y_n maximizes $\tilde{u}^{h_n} - \phi$.

We note that from Lemma 3.11, we have that

$$|\tilde{u}^{h_n}(y_n) - \tilde{u}^{h_n}(z_n)| \leq L|y_n - z_n| < Lh.$$

From the definition of y_n as a maximizer of $\tilde{u}^{h_n} - \phi$, we also observe that

$$\tilde{u}^{h_n}(z_n) - \tilde{u}^{h_n}(\cdot) \geq \tilde{u}^{h_n}(y_n) - Lh - \tilde{u}^{h_n}(\cdot) \geq \phi(y_n) - \phi(\cdot) - Lh.$$

Let $G(\nabla u(x), D^2u(x))$ denote the PDE operator (3.16) and $G^h(x, u(x) - u(\cdot))$ at interior points $x \in \mathcal{G}^h \cap X$. Since $\tilde{u}^{h_n}$ is a solution of the scheme, we can use monotonicity to calculate

$$0 = G^{h_n}(z_n, \tilde{u}^{h_n}(z_n) - \tilde{u}^{h_n}(\cdot)) \geq G^{h_n}(z_n, \phi(y_n) - \phi(\cdot) - Lh).$$

As the scheme is consistent and continuous, we conclude that

$$0 \geq \lim_{n\to\infty} G^{h_n}(z_n, \phi(y_n) - \phi(\cdot) - Lh) = G(x_0, \nabla\phi(x_0), D^2\phi(x_0)).$$

Thus v is a subsolution of (3.16).

An identical argument shows that v is a supersolution and therefore a viscosity solution. □

With these lemmas in place, we can now complete the main convergence result.

Proof of Theorem 3.6. Let h_n be any sequence converging to 0. Since $\tilde{u}^{h_n}$ is uniformly bounded and Lipschitz continuous (Lemmas 3.11-3.12), we can apply the Arzela-Ascoli theorem to obtain a subsequence h_{n_k} such that $\tilde{u}^{h_{n_k}} \to v$ uniformly for some continuous function v.

By Lemma 3.13, v is a viscosity solution of (3.16) and therefore a classical solution of the eigenvalue problem (3.13)-(2.9). Moreover, since convergence is uniform and $u^{h_{n_k}}$ continuous we have that

$$v(x_0) = \lim_{k\to\infty} \tilde{u}^{h_{n_k}}(x_0^{h_{n_k}}) = 0.$$

Thus $v = u_{\text{ex}}$ is the unique solution of (3.13)-(2.9) satisfying $v(x_0) = 0$.

Since every sequence $\tilde{u}^{h_n}$ has a subsequence converging to u_{ex}, we conclude that $\tilde{u}^h$ converges to u_{ex}. □

Note that both the analysis of the convergence of the eigenvalue and of the convergence of the solution apply to a wider class of problems. Consider the class of PDE eigenvalue problems of the form:

$$\begin{cases} F(D^2u) = c, & x \in X \\ H(\nabla u) = 0, & x \in \partial X \end{cases} \tag{3.28}$$

under the assumptions that F is elliptic, H is oblique, and the problem has a unique (up to additive constants) classical solution. The entire argument, including correct computation of eigenvalue, also applies to show the convergence of $(u^h, c^h) \to (u_{\text{ex}}, c_{\text{ex}})$ for this wider class of problems.

CHAPTER 4

NUMERICAL METHODS IN TWO DIMENSIONS

In this chapter, we review existing schemes for second order operators, including wide stencil schemes from [45], meshfree schemes from [26], and quadtree methods from [33] for constructing higher order filtered schemes [24]. Then we extend the work that was done for Dirichlet boundary conditions to accommodate nonlinear transport boundary conditions and build higher-order schemes for these boundary conditions. We also extend this work to be able to solve eigenvalue problems. The extension of these schemes to accomodate nonlinear transport boundary conditions and the solution of eigenvalue problems are the main contributions of this chapter of the dissertation.

4.1 Wide Stencil Methods

We consider two existing ways to solve equations involving eigenvalues of the Hessian via approximating directional derivatives. The first uses a standard rectangular grid for approximating the eigenvalues using the directional derivatives in the directions aligned with the grid, which is limited in that it cannot do adaptivity and cannot work with complicated domains. Additionally, the schemes actually become inconsistent near the boundary. To approximate certain directions, looking at the nearest neighbors is not sufficient. Thus, for each direction, the directional derivative is approximated by a wide stencil finite difference approximation of the second derivative in that direction. Visually, the directions we are limited to using are illustrated in Figure 4.1.

The approximation is made by generalizing the standard centered difference approximation of the second derivative to accommodate directional derivatives in other directions $\theta \in [0, 2\pi)$. Recall,

$$U_{xx}^h = \frac{U_{i+1,j} + U_{i-1,j} - 2U_{i,j}}{h^2} + \mathcal{O}(h^2) \tag{4.1}$$

For a scheme of this type on a rectangular grid, we must often look further out to find neighbors that perfectly align in a given direction. Once found, we can approximate the second directional derivative $\frac{\partial^2 u}{\partial e_\theta^2}$ in certain directions θ aligned with the grid

$$\frac{\partial^2 u}{\partial e_\theta^2} = \frac{U_{(i+V_x(\theta),j+V_y(\theta))} + U_{(i-V_x(\theta),j-V_y(\theta))} - 2U_{(i,j)}}{\left(h \cdot \sqrt{\left(V_x(\theta)^2 + V_y(\theta)^2\right)}\right)^2}. \tag{4.2}$$

where $V_x(\theta), V_y(\theta)$ are the x and y components of the vector starting from the point of interest and terminating at the desired neighboring point expressed as relative indices in order to obtain the desired direction θ and $e_\theta = (\cos(\theta), \sin(\theta))$ is the unit vector in the direction of theta. This is the standard finite difference approximation of a second directional derivative. These approximations introduce a discretization error

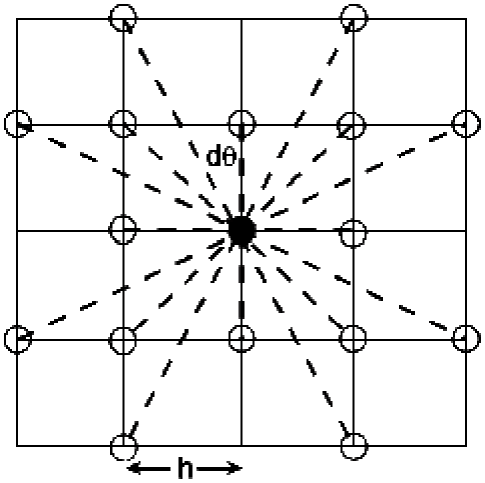

Figure 4.1 An example wide stencil finite difference approximation.

Source: [45, 26].

$d\theta$ corresponding to the size of the angles that can be resolved on the wide stencil [45, 26].

The basic approach to approximating the solutions of fully-nonlinear PDEs involving eigenvalues of the Hessian matrix is to then use the classical Rayleigh-Ritz characterization of the eigenvalues:

$$\lambda_+[u](x) = \max_{\theta\in[0,2\pi)} \frac{d^2u}{de_\theta^2}(x) \tag{4.3}$$

$$\lambda_-[u](x) = \min_{\theta\in[0,2\pi)} \frac{d^2u}{de_\theta^2}(x) \tag{4.4}$$

We approximate the minimum and maximum on the finite subset of directions given by

$$T_h = \left\{ k\, d\theta \;\middle|\; k = 0, ..., \left\lfloor \frac{2\pi}{d\theta} \right\rfloor \right\} \text{ [26]}$$

The minimum and maximum of the directional derivatives approximate the eigenvalues of the Hessian [45].

4.2 Meshfree (Generalized) Finite Difference Methods

Wide stencil methods have certain disadvantages, such as problems approximating directional derivatives near the boundary, being limited to uniform Cartesian grids, and an inability to deal with directions θ not aligned with the grid. In contrast, using meshfree methods, we approximate directional derivatives using point clouds where we can always pick out appropriate points within a given search radius of a point [45, 26].

We use generalized finite difference methods to approximate the directional derivatives of u. By using a point cloud and selecting the best aligned neighbors in a particular direction, with possibly varied distance between the points, we avoid the difficulty in approximating directions not aligned with a traditional grid. Using traditional wide stencils, to approximate certain directions may require searching further out on the grid, which leads to difficulty approximating derivatives at points near the boundary. Using meshfree methods, we can select the best aligned neighbors to minimize the error and then take an appropriate difference between them to approximate directional derivatives in any direction. Near the boundary we can still find appropriate points so that we do not lose accuracy there.

We introduce some important parameters for the mesh.

- $\mathcal{G} \subset \bar{X}$ is a point cloud consisting of the points x_i, $i = 1, ..., N$.
- $h = \sup_{x \in X} \min_{y \in \mathcal{G}} |x - y|$ is the spatial resolution of the point cloud.
- $h_B = \sup_{x \in \partial X} \min_{y \in \mathcal{G} \cap \partial X} |x - y|$ is the resolution of the point cloud on the boundary.
- $d\theta$ is the desired angular resolution for the approximation.
- $\delta = \min_{x \in X \cap \mathcal{G}} \inf_{y \in \partial X} |x - y|$ is the minimum pairwise distance between any interior point and any boundary point.

Finally,

$$r = h \left(1 + \sin\left(d\theta/2\right) + \cos\left(d\theta/2\right) \cot\left(d\theta/2\right)\right)$$

is the search radius associated with the point cloud [26].

Figure 4.2 shows a typical example of a search in the interior sufficiently far from the boundary. These points are guaranteed to exist provided we have a large enough search radius. This makes only a small error in the angular distance between the desired vector and the selected points. The error made by the approximation using the points selected in Figure 4.3 depends on size of the $d\theta_i$, which are the angular differences between the desired direction and the direction of the chosen neighbors. When near the boundary, finding neighbors in each direction is still possible if the point cloud is constructed correctly. These neighbors are also guaranteed to exist on and near the boundary provided that enough points are included along the boundary of the point cloud. A typical search near the boundary is shown in Figure 4.4. The boundary resolution h_b is the maximum of the distances between any two given

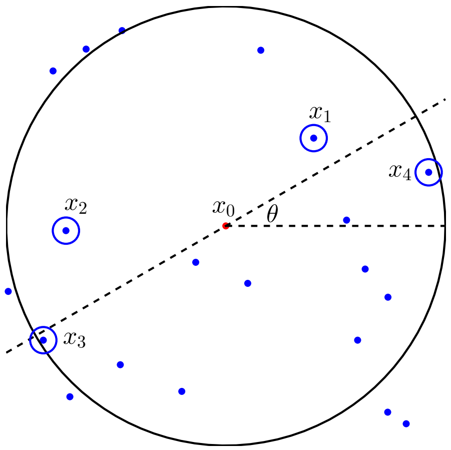

Figure 4.2 Finding appropriate points in a search radius (interior).

Source: [26].

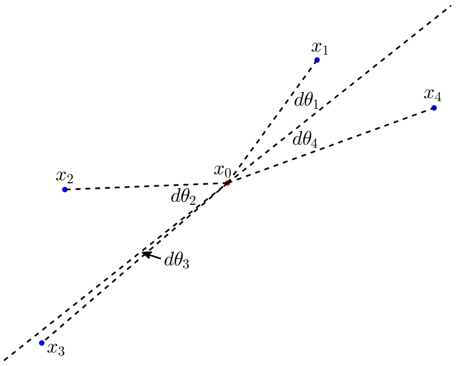

Figure 4.3 Angular resolution of the selected points.

Source: [26].

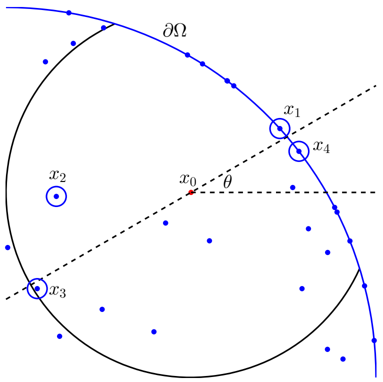

Figure 4.4 Finding appropriate points in a search radius (boundary).

Source: [26].

boundary points. δ is the minimal distance between the interior and the boundary. A node is guaranteed to exist on the boundary within the search radius given the boundary resolution $h_b \leq 2\delta \tan(d\theta/2)$ and $d\theta$ is sufficiently small, as illustrated in Figure 4.5.

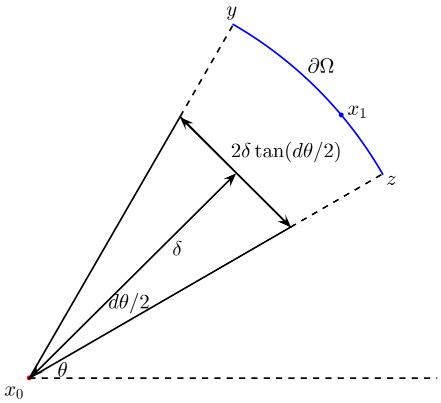

Figure 4.5 Existence of a boundary neighbor.

Source: [26].

Rather than be limited to the directions aligned with the grid, we may choose any desired direction or set of directions. Then each derivative is computed in much the same way as before, however we search within a certain radius for two points most closely matching the desired direction and for two most closely matching the opposite direction.

Error We are guaranteed to be able to find these points and this approach helps on both complicated geometries and near the boundary. $d\theta$ is the angular difference between the point and the given direction vector. As long as $d\theta$ is small, we make very little error. Unlike traditional wide stencil methods, this approach allows us to ensure that $d\theta \to 0$ globally as we refine the grid. The approximations are of the form

$$\frac{\partial^2 u}{\partial e_\theta^2} \approx \tilde{F}_\theta \left(x_i, u_i, u_i - u_j\right) \tag{4.5}$$

where j is the index of an appropriate neighboring point.

Scheme As already done in “Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian” [26], we look to approximate the directional derivatives by a scheme of the form:

$$\begin{aligned} \frac{\partial^2 u}{\partial e_\theta^2} &\approx \sum_{i=1}^{4} a_i \left(u(x_i) - u(x_0)\right) \\ &= \sum_{i=1}^{4} a_i [h_i \cos(\theta_i) u_\theta(x_0) + h_i \sin(\theta_i) u_{\theta+\pi/2}(x_0) \\ &+ \frac{1}{2} h_i^2 \cos^2(\theta_i) u_{\theta\theta}(x_0)] + \mathcal{O}\left(h_i^3 + h_i^2 \sin(d\theta_i)\right) \end{aligned} \tag{4.6}$$

Consistency and monotonicity require the following equations to be satisfied:

$$\begin{cases} \sum_{i=1}^{4} a_i h_i \cos\theta_i = 0 \\ \sum_{i=1}^{4} a_i h_i \sin\theta_i = 0 \\ \sum_{i=1}^{4} \frac{1}{2} a_i h_i^2 \cos^2\theta_i = 1 \\ a_i \geq 0. \end{cases} \quad [26] \tag{4.7}$$

In addition, the scheme must satisfy $\max(h_i) \leq r$ and $\max(d\theta_i) \leq d\theta$. This yields the scheme of the form:

$$D_{\theta\theta}u(x_0) \equiv \sum_{i=1}^{4} a_i \left(u(x_i) - u(x_0)\right) = \frac{\partial^2 u(x_0)}{\partial \theta^2} + \mathcal{O}(r + d\theta) \tag{4.8}$$

Without the $a_i \geq 0$ condition, we have three equations and four unknowns, so we choose an additional symmetry condition:

$$a_1 h_1 \sin\theta_1 + a_4 h_4 \sin\theta_4 = 0 \text{ [26]} \tag{4.9}$$

This allows the weights to be chosen appropriately when e.g., $\sin\theta_1 = 0$. It also implies another condition which allows the scheme to be solved explicitly:

$$a_2 h_2 \sin\theta_2 + a_3 h_3 \sin\theta_3 = 0 \text{ [26]} \tag{4.10}$$

Here, we use the following scheme to approximate the directional derivatives based on appropriate differences of points in the point cloud. Specifically, we use the scheme derived in "Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian" [26] which has negative monotonicity and consistency by construction. The scheme is of the form in Equation (4.8), with

$$a_1 = \frac{2S_4(C_3S_2 - C_2S_3)}{(C_3S_2 - C_2S_3)(C_1^2S_4 - C_4^2S_1) - (C_1S_4 - C_4S_1)(C_3^2S_2 - C_2^2S_3)} \tag{4.11}$$

$$a_2 = \frac{2S_3(C_1S_4 - C_4S_1)}{(C_3S_2 - C_2S_3)(C_1^2S_4 - C_4^2S_1) - (C_1S_4 - C_4S_1)(C_3^2S_2 - C_2^2S_3)} \tag{4.12}$$

$$a_3 = \frac{-2S_2(C_1S_4 - C_4S_1)}{(C_3S_2 - C_2S_3)(C_1^2S_4 - C_4^2S_1) - (C_1S_4 - C_4S_1)(C_3^2S_2 - C_2^2S_3)} \tag{4.13}$$

$$a_4 = \frac{-2S_1(C_3S_2 - C_2S_3)}{(C_3S_2 - C_2S_3)(C_1^2S_4 - C_4^2S_1) - (C_1S_4 - C_4S_1)(C_3^2S_2 - C_2^2S_3)} \tag{4.14}$$

where

$$C_i = h_i \cos(\theta_i) = \mathcal{O}(h_i), S_i = h_i \sin(\theta_i) = \mathcal{O}\left(h_i d\theta_i\right) \text{ [26].} \tag{4.15}$$

4.3 Quadtree Methods

We desire a mesh with the flexibility to resolve directional derivatives in many directions, deal with complicated geometries, and retain enough structure to build higher order schemes. One can only go so far on a Cartesian grid since the resolvable directions are limited, especially near the boundary. Unstructured point clouds have the flexibility to resolve every direction, but setting up the stencils requires an expensive search. It is better to have some structure in order to make things more efficient.

One approach to creating such a mesh is to use piecewise Cartesian grids stored as quadtrees augmented with additional boundary points, as was done in "Higher-Order Adaptive Finite Difference Methods for Fully Nonlinear Elliptic Equations" [33]. Using this approach we maintain enough structure to build higher order methods while still having the advantage of getting additional accuracy of our directional derivatives and adaptability to complex geometries.

4.3.1 Building the Quadtree

A quadtree is constructed from the basic idea of dividing squares into smaller squares [33].

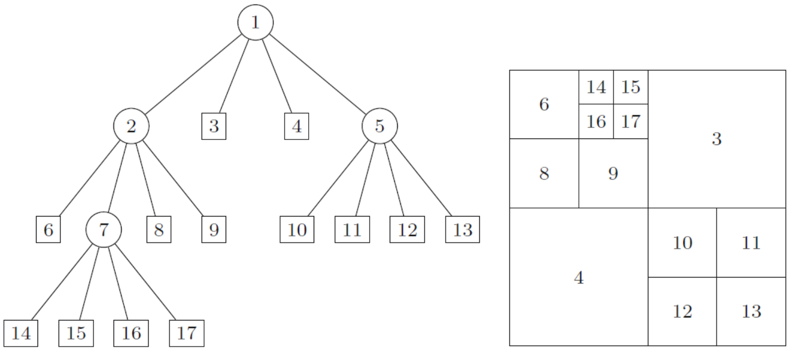

Figure 4.6 A quadtree and its corresponding subdivision. Internal nodes are represented by circles and leaves are represented with squares.

Source: [33].

We begin with a square that covers the domain, and subdivide it into four smaller squares. A square that has no squares within it is called a leaf square. Where higher resolution is needed or desired, we further subdivide these into four smaller squares. This process repeats as necessary and the vertices of the leaf squares which lie inside the domain are added to the point cloud.

To ensure existence of a stencil near the boundary, the point cloud must be augmented with additional boundary points. First, we include any points where an edge of a square intersects the boundary. This allows us to easily extend more accurate schemes from the interior to the boundary because these points align perfectly. Then we add more boundary points as needed so that more directions can be resolved even when close to the boundary. If there is a need for higher resolution near some part of the boundary, we add more boundary points to the cloud as needed.

In order for the scheme to be consistent, there must be some separation between the interior and the boundary. This should be $\mathcal{O}(h)$, which means that certain points too close to the boundary must be thrown out of the point cloud. In general, for schemes designed in this way to be consistent we require the boundary to be much more resolved than the interior ($h_B \ll h$). This is due to the fact that we search for appropriate neighbors in each direction allowing them to be much further away than h in the interior. To keep consistency, we need to be able to find and resolve the directions just as well near the boundary where the search radius must be smaller. Hence, a Cartesian mesh is not able to sustain a consistent scheme near the boundary [24].

4.4 Discretization in the Interior

In this section, we review the discretization of the PDE for second order operators using quadtrees, which was presented in “Higher-Order Adaptive Finite Difference

Methods for Fully Nonlinear Elliptic Equations" [33]. We then introduce new monotone discretizations of functions of the gradient of the solution.

4.4.1 Second Order Operators

We use approximations of the form

$$D_{\theta\theta}u(x_0) \equiv \sum_{i=1}^{4} a_i \left(u(x_i) - u(x_0)\right) = \frac{\partial^2 u(x_0)}{\partial \theta^2} + \mathcal{O}\left(r + d\theta\right)$$

to approximate the directional derivatives of u in each direction in order to approximate the eigenvalues of the Hessian matrix. These were established to be consistent and (negative) monotone in "Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian" [26], and were derived via Taylor expansion.

In order to find neighbors for the monotone scheme on a quadtree, we search each square in the quadtree along the line $x_0 + t\nu$ where x_0 is the point of interest and $\nu = (\cos\theta, \sin\theta)$. Both vertices and boundary points are included in this search. The best aligned neighbors in a particular direction are selected to be used in the approximation. Algorithm 4.1 describes the process for selecting neighbors in the first and fourth quadrant. The process is similar for other directions [33].

The eigenvalues of the Hessian, λ_1 and λ_2, can be approximated by the maximum and minimum of the directional derivatives of u. Thus, we first compute all directional derivatives in the directions given by an evenly divided partition of $[0, 2\pi)$, using more directions as n increases. These are computed by selecting the best neighbors for each point and then taking an appropriate difference, with weights described by Equation (4.14). This section applies much of what was done in "Higher-Order Adaptive Finite Difference Methods for Fully Nonlinear Elliptic Equations" [33]. In the next section, we derive a new scheme for discretizing functions of the gradient.

Algorithm 4.1 Potential Neighbors (Interior)

```
1: for each interior point do
2:     Identify the leaf square that has $x_0$ as its southwest vertex.
       This information is stored for each square so it is efficient to
       find.
3:     Find the first edge which intersects the line $x_0 + t\nu$.
4:     Identify the leaf squares that share this edge.
5:     Identify the edge of the square closest to $x_0$ along the line
       $x_0 + t\nu$.
6:     Consider the two endpoints of this edge as potential neighbors.
7:     Repeat 3-6 adding each node to the list of potential neighbors,
       until $x_0 + t\nu$ exits the search region or a boundary leaf square is
       intersected.
8:     If a boundary square is encountered, add all boundary nodes in
       that square to the list of potential neighbors.
9: end for[33]
```

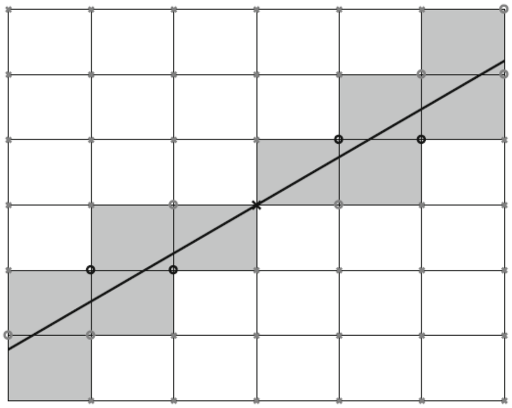

Figure 4.7 Potential neighbors are circled in gray. Examples of selected neighbors are circled in black. Gray squares are considered in Algorithm 4.1.

Source: [33].

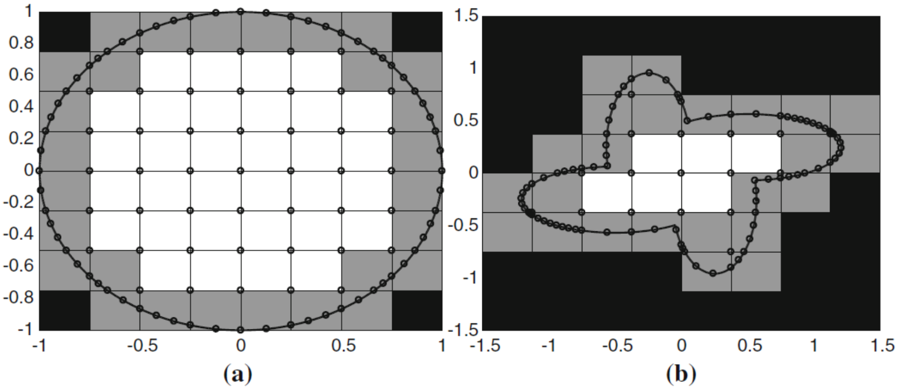

Figure 4.8 Blacked out squares are part of the quadtree but not used since they are not inside the domain. White squares are inside the domain, while gray squares intersect the boundary.

Source: [33].

In our problem, we apply the results of [33] to build a meshfree solver using quadtrees. In order to approximate the directional derivatives, we search for neighbors best aligned with each direction that lie in the square of the point at which we want to compute the derivative and in any square that is part of the quadtree that could be considered adjacent to it. This search pool includes up to eight neighboring squares by including those that touch on the diagonal, or as low as three, if the point is particularly close to the boundary.

4.4.2 Discretization of Functions of the Gradient

In order to approximate the function $f\left(x, \nabla u\right)$ on the right-hand side of the equation when there are gradient terms, we need a monotone discretization. In order to accomplish this, we first derive a consistent approximation via Taylor expansion using the same neighbors used in the interior to approximate the Laplacian operator (for reasons which will become clear).

Remark 4.1. This is a generalization of the Lax-Friedrichs approximation, which combines non-monotone centered schemes for the gradient with a small multiple of the Laplacian that enforces overall monotonicity.

By adding a carefully chosen value of ϵ times the discrete Laplacian to the approximation, we maintain monotonicity which allows the stability needed for convergence. For example, consider a discretization of $f(u') = u'$ using centered differences. Then

$$f(u') = u' \approx \frac{u_{i+1} - u_{i-1}}{2h} = \frac{u_{i+1} - u_i + u_i - u_{i-1}}{2h} = \frac{-(u_i - u_{i+1}) + (u_i - u_{i-1})}{2h}$$

is not a monotone scheme, since it is not a nondecreasing function of the differences $u_i - u_{i+1}$ and $u_i - u_{i-1}$. However, we can subtract ϵ times the second derivative's monotone approximation to get

$$\begin{aligned} f(u') \approx& \frac{-(u_i - u_{i+1}) + (u_i - u_{i-1})}{2h} - \epsilon \frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} \\ =& \frac{-(u_i - u_{i+1}) + (u_i - u_{i-1})}{2h} + \epsilon \frac{(u_i - u_{i+1}) + (u_i - u_{i-1})}{h^2} \\ =& \frac{(2\epsilon - h)(u_i - u_{i+1}) + (2\epsilon + h)(u_i - u_{i-1})}{2h^2} \end{aligned} \tag{4.16}$$

so this will be a monotone function if $\epsilon \geq h/2$.

Next, we derive the monotone discretization. Since each point uses four neighbors for the approximation, we need to solve a system of equations for the coefficients in each approximation. The differences $dx_i = (x_0 - x_i)$ and $dy_i = (y_0 - y_i)$ are used in the equations for u_x:

$$\begin{aligned} u_x &\approx \sum_{i=0}^{4} A_i u_i \\ &\approx \sum_{i=0}^{4} A_i u_0 + \sum_{i=1}^{4} \left[A_i \left(dx_i u_x + \frac{dx_i^2}{2} u_{xx} + dy_i u_y + \frac{dy_i^2}{2} u_{yy} + dx_i dy_i u_{xy} \right) \right] + \mathcal{O}(h^3) \end{aligned} \tag{4.17}$$

Then, we can derive the approximation with only four neighbors, since the neighbors chosen are from each cardinal direction; and thus, we do not care about the mixed derivative terms since one of dx_i or dy_i is equal to zero for each i.

$$\begin{cases} \sum_{i=0}^{4} A_i = 0 \\ \sum_{i=1}^{4} A_i dx_i = 1 \\ \sum_{i=1}^{4} A_i dy_i = 0 \\ \sum_{i=1}^{4} A_i dx_i^2/2 = 0 \\ \sum_{i=1}^{4} A_i dy_i^2/2 = 0 \end{cases} \tag{4.18}$$

For u_y, we similarly find coefficients that satisfy

$$\begin{cases} B_0 + \sum_{i=5}^{8} B_i = 0 \\ \sum_{i=5}^{8} B_i dx_i = 0 \\ \sum_{i=5}^{8} B_i dy_i = 1 \\ \sum_{i=5}^{8} B_i dx_i^2/2 = 0 \\ \sum_{i=5}^{8} B_i dy_i^2/2 = 0 \end{cases} \tag{4.19}$$

These two linear systems can be solved explicitly. We then have a first order approximation for the gradient terms, but it may not be monotone. In order to create a monotone approximation when we plug into f, we approximate

$$f\left(x, y, u_x, u_y\right) \approx f\left(x, y, \sum_{i=0}^{4} A_i u_i, B_0 u_0 + \sum_{i=5}^{8} B_i u_i\right) - \epsilon\left(\Delta_h u\right)$$

where $\Delta_h u$ is the discrete Laplacian depending on coefficients of the second derivatives. Writing this explicitly including neighbors for $\Delta_h u$, we get

$$f\left(x, y, u_x, u_y\right) \approx f\left(x, y, \sum_{i=0}^{4} A_i u_i, B_0 u_0 + \sum_{i=5}^{8} B_i u_i\right) - \epsilon\left(\sum_{i=0}^{4} C_i u_i + D_0 u_0 + \sum_{i=5}^{8} D_i u_i\right) \tag{4.20}$$

Since we have monotone approximations for the Laplacian operator, subtract it from the approximation using a weight $\epsilon = \mathcal{O}(h)$ so that the scheme becomes monotone, but as $h \to 0$, the extra terms drop out. This Lax-Friedrichs type approximation requires a choice of ϵ large enough so that the scheme is monotone to accomplish the result.

In practice, our problem requires epsilon to look like hK where K is the Lipschitz constant of f for u_x, u_y. Since we may not know this explicitly, we approximate the Lipschitz constant by 1.2 times the maximum of the derivatives with respect to u_x,u_y for the function f. It is better to make epsilon too large than too small because stability is our concern. The properties informally stated here are based on the following lemma.

Lemma 4.2. The approximation in Equation (4.20) is monotone if $\epsilon \geq \max_i \frac{|A_i|}{|C_i|} K$ where K is the Lipschitz constant of f.

Proof. Let $G\left(u_1 - u_0, u_2 - u_0, ..., u_8 - u_0\right)$ be the approximation scheme defined by

$$G\left(v_1, v_2, ..., v_8\right) = f\left(x, y, \sum_{i=1}^{4} A_i v_i, \sum_{i=5}^{8} B_i v_i\right) - \epsilon\left(\sum_{i=1}^{4} C_i v_i + \sum_{i=5}^{8} D_i v_i\right)$$

where $v_i = u_i - u_0$, A_i are the coefficients in the approximation for u_x, B_i are the coefficients in the approximation of u_y, and $C_i, D_i > 0$ are the coefficients in the approximation for $-\Delta u$ all derived via Taylor series expansion. Then, we want to show this is a monotone function in each argument; i.e., show that G is a non-increasing function of each v_i. Consider

$$G\left(v_1 + \delta, v_2, ..., v_8\right) - G\left(v_1, v_2, ..., v_8\right)$$

where $\delta > 0$. The other arguments of G being monotone will follow by a nearly identical argument.

$$
\begin{aligned}
&G\left(v_1+\delta, v_2, ..., v_8\right) - G\left(v_1, v_2, ..., v_8\right) \\
&= f\left(x, y, A_1\delta + \sum_{i=1}^{4} A_i v_i, \sum_{i=5}^{8} B_i v_i\right) - \epsilon\delta C_1 - \epsilon\left(\sum_{i=1}^{4} C_i v_i + \sum_{i=5}^{8} D_i v_i\right) \\
&- f\left(x, y, \sum_{i=1}^{4} A_i v_i, \sum_{i=5}^{8} B_i v_i\right) + \epsilon\left(\sum_{i=1}^{4} C_i v_i + \sum_{i=5}^{8} D_i v_i\right) \\
&= f\left(x, y, A_1\delta + \sum_{i=1}^{4} A_i v_i, \sum_{i=5}^{8} B_i v_i\right) - f\left(x, y, \sum_{i=1}^{4} A_i v_i, \sum_{i=5}^{8} B_i v_i\right) - \epsilon\delta C_1 \\
&= -\left[f\left(x, y, \sum_{i=1}^{4} A_i v_i, \sum_{i=5}^{8} B_i v_i\right) - f\left(x, y, A_1\delta + \sum_{i=1}^{4} A_i v_i, \sum_{i=5}^{8} B_i v_i\right)\right] - \epsilon\delta C_1 \\
&\leq K||\sum_{i=1}^{4} A_i v_i - A_1\delta - \sum_{i=1}^{4} A_i v_i||_\infty - \epsilon\delta C_1 \\
&= K||A_1\delta||_\infty - \epsilon\delta C_1 \leq 0
\end{aligned}
\tag{4.21}
$$

as long as $\epsilon > \frac{K\|A_1\|}{C_1}$. Thus, in order to show G is monotone, the argument can be repeated for all i, and taking $\epsilon > \max_i \frac{K\|A_i\|}{C_i}$ will yield a monotone function. □

In particular for this example, $\epsilon \geq \max_i \frac{K|A_i|}{C_i}$ which we know scales like

$$\epsilon \geq \frac{K\frac{1}{h}}{\frac{1}{h^2}} = Kh$$

4.5 Discretization of the Boundary Condition

The discretization of Dirichlet boundary conditions is simple. Here the contribution of this work is the discretization of functions of the gradient and of nonlinear Neumann-type boundary conditions. This section describes discretization of the possibly nonlinear Neumann-type boundary condition

$$H\left(\nabla u\right) = 0 \quad x \in \partial X$$

4.5.1 Discretization of the Boundary Condition

Recall the formulation we use for the boundary conditions,

$$H\left(\nabla u(x)\right) = 0, x \in \partial X$$

where H is the signed distance function:

$$H(y) = \begin{cases} +\text{dist}\left(y, \partial Y\right) & y \notin Y \\ -\text{dist}\left(y, \partial Y\right) & y \in Y \end{cases}$$

Convexity of H and u ensure the existence of a consistent, monotone discretization that uses only values in the domain.

The neighbors are selected for the boundary conditions based on searching for appropriate points within the same square as the boundary node or within neighboring squares. The points are divided into eight regions based on the direction they lie in, and points are chosen based on proximity to the node and direction. In Figure 4.9, we do not include the points very close to the boundary in the cloud, but include in the search all points both in the cloud and within a one square radius (including diagonally) of the point of interest.

To approximate the boundary condition in Equation (2.19), we first compute H^*, which can be done by using the formula in Equation (2.17) and approximating the sup by a max among a finite set of directions $\hat{N}$ which are determined for each point in the following way:

- Partition $[0, 2\pi)$ using a number of elements depending on h.
- $\hat{N}$ is the subset of directions from this partition which point outside the boundary at the given boundary point.

Then, we use a monotone approximation of the directional derivatives at each boundary point. We get a monotone approximation of the boundary condition by

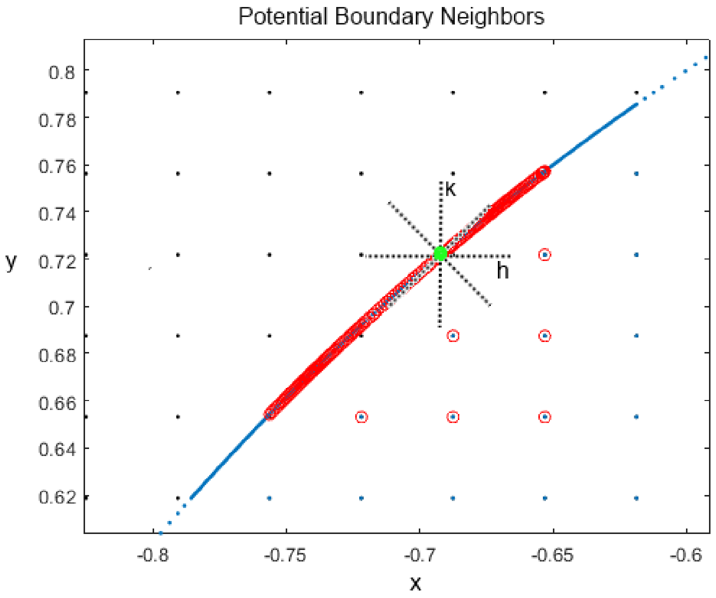

Figure 4.9 An example set of neighbors (in red) for the boundary point (in green) with points in the cloud labeled in blue and the points in the quadtree but not the point cloud labeled in black. The eight regions partition the h,k plane. Not all regions need to be represented.

computing

$$\max\left\{\mathcal{D}_n u - H^*\left(n\right)|n \in \hat{N}\right\}$$

where $\hat{N}$ is a finite subset of the directions such that $n \cdot n_x > 0$ and $\mathcal{D}_n u$ is an approximation of the directional derivative in the direction n. To approximate for a given n, we choose two neighbors such that the direction $-n$ is in the convex hull of the two neighbors. To accomplish this at a point $x_0 \in \partial x$, we need to identify points $x_1, x_2 \in \mathcal{G}^h$ such that for small $t > 0$, the line segment $x_0 - nt$ is contained in the convex hull of x_0, x_1, x_2 (which is a triangle). Given the structure of our mesh, this is easily accomplished for neighbors satisfying

$$|x_1 - x_0|, |x_2 - x_0| \leq \mathcal{O}\left(h\right).$$

See Figure 4.9 for a visual of this selection. Let u_1, u_2 be the chosen neighbors for u_0. Then let

- $h_i = x_i - x_0$
- $k_i = y_i - y_0$
- θ is the angle for the direction n

The scheme used to approximate the first directional derivatives is derived via Taylor expansion. We can set up the following system of equations, which leads to a monotone scheme

$$\begin{cases} \sum_{i=0}^{2} a_i = 0 \\ \sum_{i=1}^{2} a_i h_i = -\cos(\theta) \\ \sum_{i=1}^{2} a_i k_i = -\sin(\theta) \end{cases} \tag{4.22}$$

The scheme used is then

$$a_0 u_0 + a_1 u_1 + a_2 u_2$$

where we have the coefficients

$$\begin{cases} a_1 = \frac{-h_2 \sin(\theta) + k_2 \cos(\theta)}{h_2 k_1 - h_1 k_2} \\ a_2 = \frac{h_1 \sin(\theta) - k_1 \cos(\theta)}{h_2 k_1 - h_1 k_2} \\ a_0 = -(a_1 + a_2). \end{cases} \tag{4.23}$$

4.6 Solving the Eigenvalue Problem

We approximate the PDE operator as above, then use a damped Newton's method to solve for u^h and c^h.

Handling the Extra Unknown In our implementation, the point cloud includes boundary points, then the interior points, for a total of N points. Normally, the unknowns are the u_i such that $x_i \in \partial X$, and the u_i such that $x_i \in X$. The main characteristic that makes this an eigenvalue problem is an extra unknown c. In general, since we have introduced an additional unknown, we have N equations with

$N+1$ unknowns. This yields a solution that is only unique up to an additive constant, since both the PDE and the boundary condition depend only on derivatives of u. Thus, in order for the solution to be unique, we add another equation to the system. For now, we use the condition that the first value on the boundary has a value equal to zero. In practice, the choice of extra condition does not matter, since in most applications we only care about the gradient of u. In introducing an extra unknown, we also would like to be able to solve more general problems introducing more complicated functions on the right side. These variants have the form

$$F\left(\lambda_1(D^2u), \lambda_2(D^2u)\right) = cf\left(x, \nabla u\right)$$

in the interior. The full problem can then be written abstractly as

$$F[u, c] = 0$$

where

$$F[u, c] = \begin{cases} cf\left(x, \nabla u\right) - \tan^{-1}\left(\lambda_1\right) - \tan^{-1}\left(\lambda_2\right) & x \in X \\ H\left(\nabla u\right) & x \in \partial X \\ u_1 & \text{(Extra equation)} \end{cases} \tag{4.24}$$

and u_1 is the value of u at the first boundary point included in the point cloud.

We now replace Equation (4.24) with the discrete problem

$$\begin{cases} F^h[u^h] + c^h = 0, & x \in X \\ H^h[u] = 0, & x \in \partial X \\ u_1^h = 0 \end{cases} \tag{4.25}$$

Note that these approximations are consistent and monotone, so they fit into the convergence framework described by Theorem 3.5 and Theorem 3.6.

A more accurate, but not yet known to be convergent approximation is derived in the next section.

4.7 Higher Order Implementation

4.7.1 Filtered Schemes

The basic idea of a filtered scheme comes from [24]. This is to construct a scheme in a clever way that combines the stability of a monotone scheme (which is convergent from Barles and Souganidis [1]) with the accuracy of a higher order scheme that may or may not have stability issues.

We would like to use the accurate scheme where it works, and the monotone scheme where it has a singularity. A singularity can naturally be defined as when the difference between the accurate scheme and the monotone scheme is large. When the accurate scheme is not singular, the difference should be on the order of the discretization error. When it is singular, the difference will be very large since the accurate scheme would blow up. To convert this condition into a continuous filter function, we want the filter to be equal to the identity nearby the origin and zero outside, e.g.,

$$S(x) = \begin{cases} x & |x| \leq 1, \\ 0 & |x| \geq 2, \\ -x+2 & 1 \leq x \leq 2, \\ -x-2 & -2 \leq x \leq -1 \end{cases} \quad [24] \tag{4.26}$$

The scheme is then defined as:

$$F_S = F_M + \epsilon \cdot S\left(\frac{F_A - F_M}{\epsilon}\right) \tag{4.27}$$

where ϵ must be large in comparison to the discretization error of the scheme. This is consistent with what we are looking for since when the accurate scheme is too large, S is zero and the scheme reduces to the monotone scheme, and when it is not, S is the identity, which cancels the equation to the accurate scheme. Filtered schemes are almost monotone, and converge by similar principles to the proof of convergence of monotone schemes [1, 24].

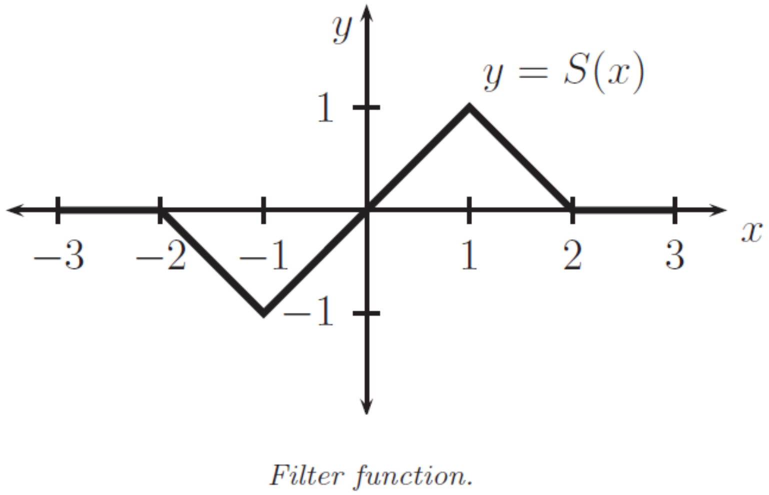

Figure 4.10 The filter function, a continuous function that is the identity near the origin, and decays to zero outside.

Source: [24].

Analysis The convergence of filtered schemes for nonlinear elliptic PDEs is established in "Convergent Filtered Schemes for the Monge-Ampère Partial Differential Equation" [24] in a similar way to the original convergence theorem put forth by Barles and Souganidis [1]. The basis for this is that we are working with a nearly monotone (or nearly elliptic) scheme:

Definition 4.3. (Nearly Monotone) A scheme $F^\epsilon[v]$ is **nearly monotone** if it can be written as

$$F^\epsilon[v] = F_M[v] + F_P^\epsilon[v] \tag{4.28}$$

where F_M is a monotone scheme and F_P^ϵ is a perturbation, which satisfies

$$\lim_{\epsilon \to 0} ||F_P^\epsilon|| = 0 \text{ [4]}.$$

This definition, along with the usual requirements for the convergence theorem, yield the following theorem from [24]:

Theorem 4.4. *(Convergence of Approximation Schemes) Let u be the unique viscosity solution of the PDE*

$$F\left(x, u(x), \nabla u(x), \nabla^2 u(x)\right) = 0, x \in \overline{X}$$

and let u^ϵ be a stable solution of the consistent, nearly monotone approximation scheme in Equation (4.28)*. Then,*

$$u^\epsilon \to u \textit{ locally uniformly, as } \epsilon \to 0 \textit{ [4].}$$

Also, if the non-monotone perturbation is continuous, u^ϵ exists and is stable.

Remark 4.5. This applies to PDEs satisfying a comparison principle. This does not fit our convergence framework at this time (in particular, for computing the correct eigenvalue). However, initial results show that this is an effective approach worthy of additional study in the context of eigenvalue problems.

The proof of the theorem is very similar to the proof of the Barles and Souganidis convergence theorem. What allows the result to carry through is that since S is bounded, Equation (4.27) is a small perturbation of the monotone scheme. This allows the result to go through even though the scheme itself is not monotone. This result provides motivation for using filtered schemes as well as a way of proving their convergence. In addition, the schemes are formally higher-order since on smooth examples it reduces to the accurate scheme, with reduction in order the more singular the example being solved [24].

Handling the Boundary Conditions We build a filtered scheme in our example to obtain formally second order convergence. In the interior, we can easily find an accurate approximation of the operator by using the structure of the quadtree. On the boundary, we also require an accurate scheme for u_x and u_y to approximate the boundary condition $H(\nabla u) = 0$, which we construct using appropriate neighbors. For this, we search the square and all neighboring squares for points closely aligned with the x and y directions, respectively; but these will generally not be perfectly aligned.

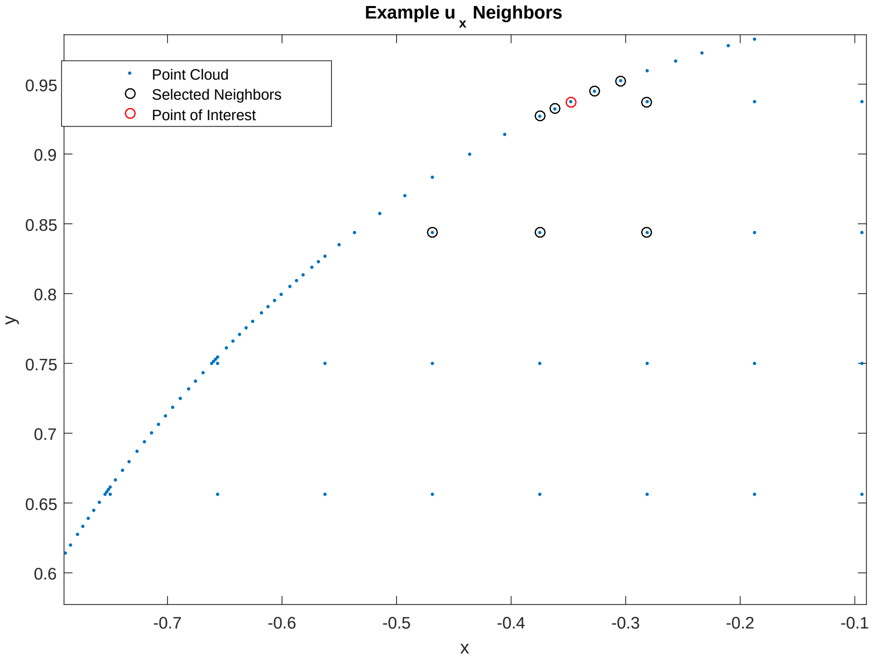

Figure 4.11 An example of selected neighbors for the higher order approximation to the x derivative.

We derive a second order scheme via Taylor expansion using five neighbors, then solving the system of equations numerically. For a scheme of the form $a_0u_0+\sum_{i=1}^{n} a_iu_i$ we require the following equations for u_x:

$$\begin{cases} \sum_{i=1}^{n} a_i = -a_0 \\ \sum_{i=1}^{n} h_ia_i = 1 \\ \sum_{i=1}^{n} k_ia_i = 0 \\ \sum_{i=1}^{n} \frac{h_i^2}{2}a_i = 0 \\ \sum_{i=1}^{n} \frac{k_i^2}{2}a_i = 0 \\ \sum_{i=1}^{n} h_ik_ia_i = 0 \end{cases} \tag{4.29}$$

where a_i are the coefficients, h_i are the difference in x coordinate between the point u_i and u_0, and k_i are the difference in y coordinate between the point u_i and u_0. Then we can solve the system numerically. When this happens, the system may or may not be well conditioned. To remedy the problem, we select eight neighbors and solve the five equations in a least squares sense. This can be included in the filtered scheme using the filter above and the monotone scheme used earlier.

4.7.2 Accurate Approximation of Distance Functions for Target Sets

In order to approximate the boundary condition $H\left(\nabla u\right) = 0$, we not only need accurate approximations for the gradient, but also a perfectly accurate signed distance function $H\left(x, y\right)$. This is handled on a case by case basis depending on the target set. Whenever possible, we find the exact function that does this. To find the correct $H\left(x, y\right)$, we must first solve the minimization problem:

$$\text{minimize } d\left(x, y\right)^2 = \left(x - x_0\right)^2 + \left(y - y_0\right)^2 \tag{4.30}$$

$$\text{subject to } x, y \text{ on the boundary of the target set.} \tag{4.31}$$

where (x_0, y_0) is the point that we are plugging into the distance function H. Note that we minimize the distance squared since this is a more smooth objective function and the problem is equivalent. To find the desired function, we take the actual distance on the coordinates found and then correct the sign to be negative if (x_0, y_0) is inside the target set and positive if it is outside. As a function of two variables, we expect the plot of the function on a square domain to look like a cone since the distance is negative inside and increasingly positive outside (see Figure 4.12). We also expect the level sets of $H\left(x, y\right)$ to be shaped like the target set. The zero level set of $H\left(x, y\right)$ should correspond to the actual target set (see Figure 4.12).

For many figures such as the skewed ellipse, the minimization problem must be done numerically. In that example, we represent the boundary of the ellipse in polar coordinates. Then for each point, we find the angle for which the distance to the ellipse in that direction is minimized. To do this, we use Newton's method with the derivative of the distance function. Unfortunately, in general there may be multiple places where the derivative is zero, some local minima and some local maxima. To ensure we converge quickly to the correct minimum, we do a crude search for the general location of the best angle by brute force with far less angles being considered. This puts us in a close enough spot for the initial guess that Newton's method can converge to the correct spot in very few iterations.

For the skewed ellipse, we see that the zero level set of $H\left(x, y\right)$ is the target set. In Figure 4.12, we see the expected type of figure which is a cone whose level sets are the shape of the target set. Other targets such as the unit circle have more obvious signed distance functions like $H\left(x, y\right) = \sqrt{x^2 + y^2} - 1$. Since any point on the boundary of the unit circle has a distance to the origin of one, and the unit circle is symmetric about the origin, the closest distance is a projection of the point in the direction of the origin onto the circle. Thus, since the distance to the origin for any point is given by $\sqrt{x^2 + y^2}$, the distance to the boundary of the circle would be

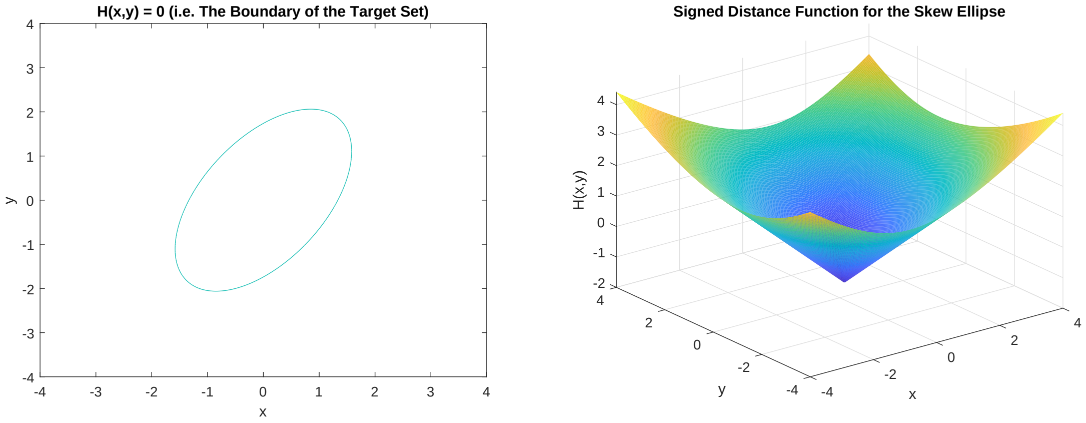

Figure 4.12 $H(x, y)$ for the skewed ellipse.

$\sqrt{x^2 + y^2} - 1$. Notice that for a value outside the circle, the distance is greater than one, so we have a positive distance. For a value inside the circle, the distance is less than one, so H is negative. Thus, we have the signed distance function as desired.

4.8 Solution Methods

Solution methods are readily available for the Dirichlet problem, however adaptions must be made for the solution of eigenvalue problems and for Neumann or nonlinear Neumann-type boundary conditions. The main difference is that we build in an extra condition in order to account for the extra unknown c. We can utilize existing solution methods by building the extra condition into the operator.

Once we have the approximations and everything is discretized appropriately, we solve for the solution using either a damped Newton's method or an explicit method such as Forward Euler.

For Newton's method we require a Jacobian matrix and expect convergence only for a good enough initial guess. For Euler, we instead require the time step to be sufficiently small, and we accomplish this by shrinking it to the exact size it needs to be to cause the residual to go down at each iteration. In both methods, we need a vector representing an operator $F^h[u]$ where the method used will find the solution to $F^h[u] = 0$.

Summary We solve a system of equations:

$$F[u] = \begin{bmatrix} H\left(\nabla u\right) \\ cf\left(x, \nabla u\right) - \tan^{-1}\left(\lambda_1\right) - \tan^{-1}\left(\lambda_2\right) \\ u\left(1\right) \end{bmatrix} \approx F^h[u] =$$

$$\begin{bmatrix} \max_{|n|=1}\left\{\nabla_h u\left(x\right) \cdot n - H^*\left(n\right) | n \in \hat{N}\right\} \\ cf\left(x, \nabla u\right) - \tan^{-1}\left(\max_{\theta \in [0,2\pi)} \frac{d^2u}{de_\theta^2}\left(x\right)\right) - \tan^{-1}\left(\min_{\theta \in [0,2\pi)} \frac{d^2u}{de_\theta^2}\left(x\right)\right) \\ u\left(1\right) \end{bmatrix} = \begin{bmatrix} 0 \\ ... \\ 0 \end{bmatrix}$$

where $\hat{N}$ is a finite subset of the directions such that $n \cdot n_x > 0$. This has the boundary conditions followed by the evaluation of the guess u in the PDE, followed by the condition that $u\left(1\right) = 0$. Specifically, the first entries of the vector correspond to the boundary points, where the desired condition is that $H\left(\nabla u\right) = 0$, so $H\left(\nabla u\right)$ is the value of F for u on the boundary. Next, we list the condition in the interior to solve, which is that

$$c\, f\left(x, \nabla u\right) - \tan^{-1}\left(\lambda_1\right) - \tan^{-1}\left(\lambda_2\right) = 0$$

Hence, the monotone approximation of the left-hand side is listed for interior points. Finally, we need to include the extra equation to ensure uniqueness, so the final equation is $u\left(1\right) = 0$. Again, this translates to the value of the last entry of $F^h[u]$ being $u\left(1\right)$.

A Jacobian matrix is then built using derivatives of the PDE with respect to each entry which contains all values of u followed by the value of c. These are used in a damped Newton solver which finds the u for which $F^h = 0$, i.e., for which u satisfies the PDE and boundary conditions. For the initial guesses, we can interpolate a solution computed on a more coarse grid or use a large quadratic such as

$$10\left(x^2+y^2\right).$$

Jacobian The Jacobian used for the method was made to enforce the boundary conditions by using the LF transform to capture the geometry of the target set. The top of the Jacobian matrix is the matrix M which is a differentiation matrix for the monotone approximation of the appropriate directional derivatives along the boundary.

$$M = D^h_\nu$$

where ν is the argmax of the discrete approximation to $\frac{\partial u}{\partial \nu} - H^*\left(\nu\right)$.

To solve for c, we needed an extra condition, $u\left(1\right) = 0$. This was included in the Jacobian as well as the partial derivatives with respect to c.

$$\nabla F^h = \begin{bmatrix} M & ... & 0 \\ -\text{diag}\left(\frac{1}{1+\lambda_1^2}\right) D_1^2 - \text{diag}\left(\frac{1}{1+\lambda_2^2}\right) D_2^2 & ... & 1 \\ 1 & 0... & 0 \end{bmatrix}$$

Using a non-linear damped Newton's method we solve for the solution based on the operator which includes the approximation of the eigenvalues.

For Euler's Method, we need only use the operator $F^h[u]$ with an initial guess, since we may solve for a steady state $(u_t = 0)$ of the equation $u_t = F^h[u]$. This leads to the method

$$U_{n+1} = U_n + dt \cdot F^h[U_n]$$

Then, to find the unknown c, we enforce the extra condition $u\left(1\right) = 0$ at each step and use the update

$$c = \max_i \left(F_i[U_{n+1}]\right)$$

directly after the update for U_{n+1}.

4.9 Computational Results, Examples, and Figures

We solve the problem for several different examples using the monotone scheme derived above. Some of these have exact solutions while others do not. The problem is solved for many different domains and target sets. Each of these examples uses the Newton solver. The Euler's method solver can be used for problems where the Newton solver is too expensive. On each example, we expect to see $\mathcal{O}(\sqrt{h})$ accuracy since this is the truncation error for the schemes. However, this rate of convergence is not rigorously proven.

4.9.1 Circle to Circle

For the first example, we simply map a circle into the same circle. Thus, the gradient of u would have to be the identity, and integrating up we find the exact solution to be

$$u(x,y) = \frac{1}{2}(x^2 + y^2)$$

$$c = \tan^{-1}(1) + \tan^{-1}(1)$$

This is a quadratic so we expect very high accuracy for low numbers of grid points. The map is pictured in Figure 4.13.

An error table, along with an error plot for the circle to circle example using the monotone scheme, are shown in Figure 4.14, and in Table 4.1. Note that we only expect $\mathcal{O}(\sqrt{h})$ accuracy but on this example we achieve order $\mathcal{O}(h)$.

4.9.2 Circle to Ellipse

Next, we map the unit circle to an ellipse with semi-major axis and semi-minor axis two and one, respectively. We also invert this map. In these examples, y should be

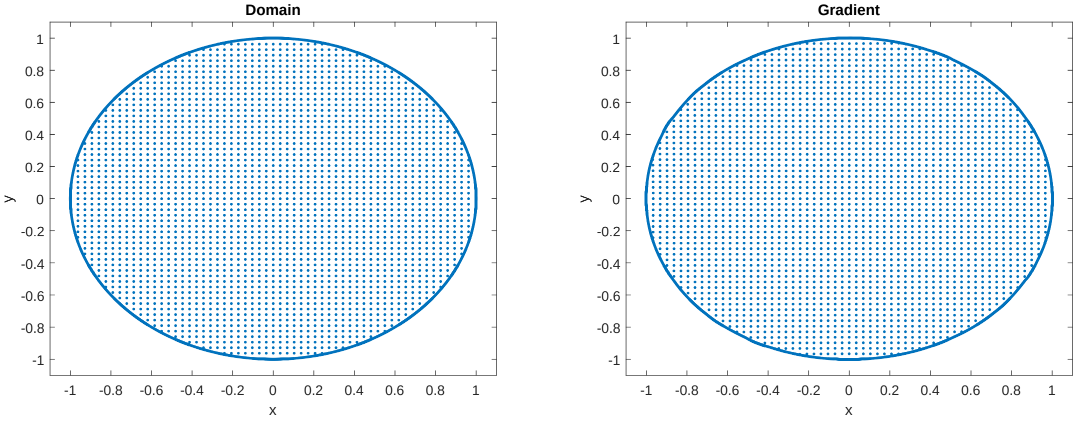

Figure 4.13 An example where ∇u maps a circle into another circle.

Table 4.1 Circle to Circle Error Table

h	Error	Ratio	Observed Order
0.1375	4.97923e-02		
0.06875	2.35620e-02	2.11325	1.07946
0.034375	1.15507e-02	2.03987	1.02848
0.0171875	5.61285e-03	2.05790	1.04118
0.00859375	2.75303e-03	2.03879	1.02771

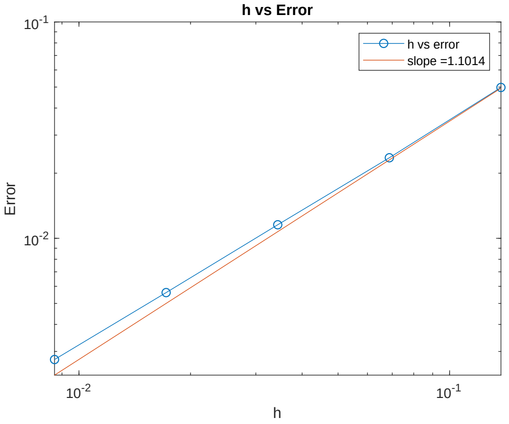

Figure 4.14 A convergence plot for the example mapping a circle into another circle.

the identity but x should either double or half its length. Hence for the circle to ellipse map, the gradient should be $< 2x, y >$ leading via integration to the exact solution

$$u(x, y) = x^2 + \frac{1}{2}y^2$$

$$c = \tan^{-1}(2) + \tan^{-1}(1)$$

For the ellipse to circle, we have the reverse, and the gradient should be $< \frac{1}{2}x, y >$ which leads via integration to the exact solution

$$u(x, y) = \frac{x^2}{4} + \frac{y^2}{2}$$

$$c = \tan^{-1}\left(\frac{1}{2}\right) + \tan^{-1}(1)$$

The two maps are pictured in Figures 4.15 and 4.16.

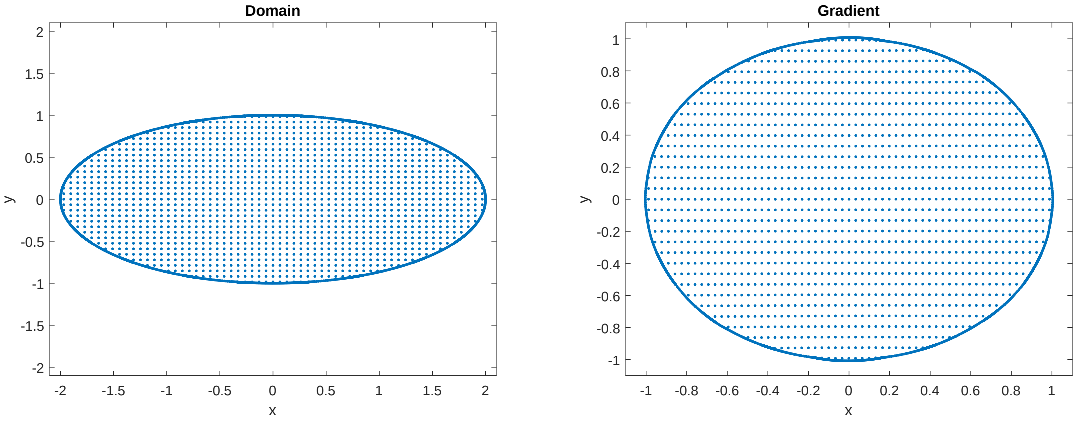

Figure 4.15 An example where ∇u maps an ellipse into a circle.

In addition to pictures of the maps, we present convergence plots (Figures 4.17 and 4.18) and error tables (Tables 4.2 and 4.3). On the circle to ellipse map, we observe $\mathcal{O}(h)$ accuracy, which is better than the expected $\mathcal{O}(\sqrt{h})$.

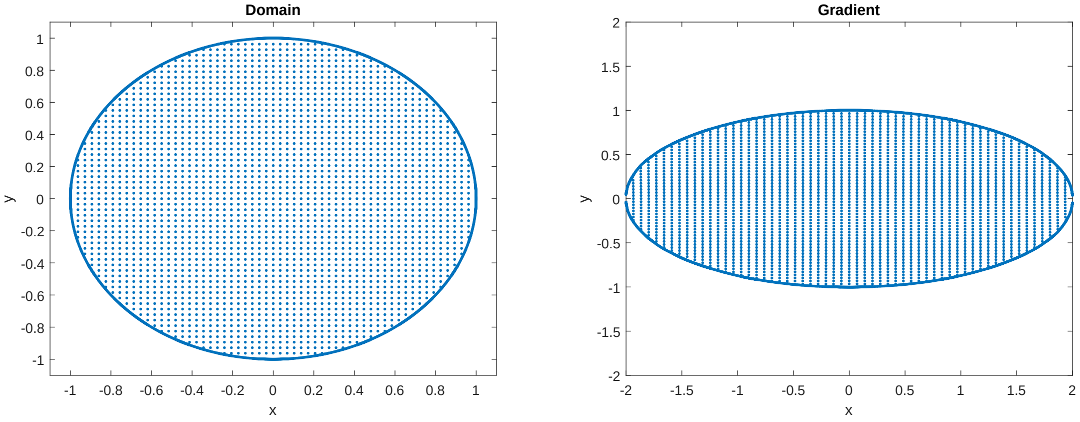

Figure 4.16 An example where ∇u maps a circle into an ellipse.

Table 4.2 Circle to Ellipse Error Table

h	Error	Ratio	Observed Order
0.13750	7.07007e-02		
0.06875	3.08281e-02	2.29338	1.19748
0.03438	1.70893e-02	1.80395	0.85116
0.01719	8.03814e-03	2.12602	1.08816
0.00859	3.41182e-03	2.35597	1.23632

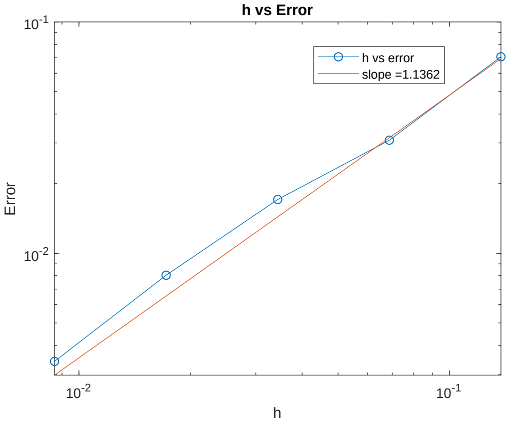

Figure 4.17 A convergence plot for the example mapping a circle into an ellipse.

4.9.3 Ellipse to Circle

Table 4.3 Ellipse to Circle Error Table

h	Error	Ratio	Observed Order
0.26250	9.31217e-02		
0.13125	4.87816e-02	1.90895	0.93278
0.06563	2.39944e-02	2.03304	1.02364
0.03281	1.13675e-02	2.11079	1.07778
0.01641	5.56250e-03	2.04359	1.03111

On the ellipse to circle map, we observe $\mathcal{O}(h)$ accuracy, which is better than the expected $\mathcal{O}(\sqrt{h})$.

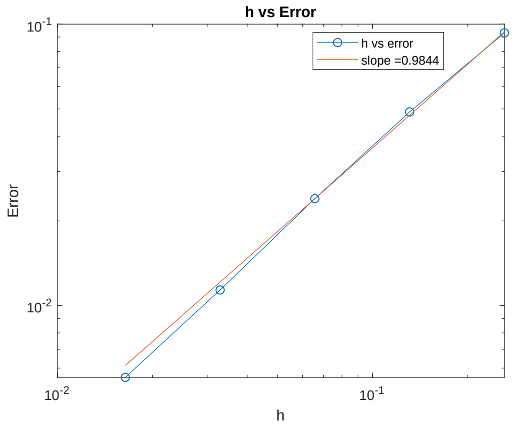

Figure 4.18 A convergence plot for the example mapping an ellipse into a circle.

4.9.4 Ellipse to Skewed Ellipse

The transformation from one figure into another is also illustrated by computing the transformation of an ellipse. The domain is the ellipse described above, but the target set is a skewed ellipse. Another characterization of the ellipse is as follows. Let B be the unit circle. Then the domain ellipse is given by $X = M_x B$ and the target skewed ellipse is given by $Y = M_y B$, where

$$M_x = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$M_y = \begin{bmatrix} 1.5 & .5 \\ .5 & 2 \end{bmatrix}$$

In $\mathbb{R}^2$ the optimal map can be found explicitly to be

$$\nabla u(x) = M_y R_\theta M_x^{-1} x$$

where R_θ is the rotation matrix

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

and the angle is given by

$$\theta = \tan^{-1}\left(\left(\operatorname{trace}\left(M_x^{-1} M_y^{-1} J\right) / \operatorname{trace}\left(M_x^{-1} M_y^{-1}\right)\right)\right)$$

where

$$J = R_{\pi/2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

The map and the convergence plot are pictured in Table 4.4 and Figure 4.19 [3].

We observe $\mathcal{O}(h)$ accuracy, which is better than the expected $\mathcal{O}(\sqrt{h})$.

4.9.5 Other Maps

For some maps, we do not have an exact solution, but we nonetheless compute the solution for them. For simplicity we map the square with sides of length 1.1 into

Table 4.4 Ellipse to Skewed Ellipse Convergence

h	Error	Ratio	Observed Order
0.26250	1.30436e-01		
0.13125	5.70280e-02	2.28723	1.19360
0.06563	2.69079e-02	2.11938	1.08364
0.03281	1.42328e-02	1.89055	0.91881
0.01641	6.76756e-03	2.10309	1.07251

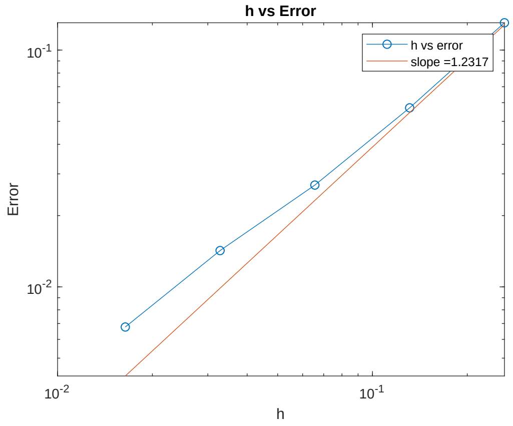

Figure 4.19 A convergence plot for the example mapping an ellipse into a skewed ellipse.

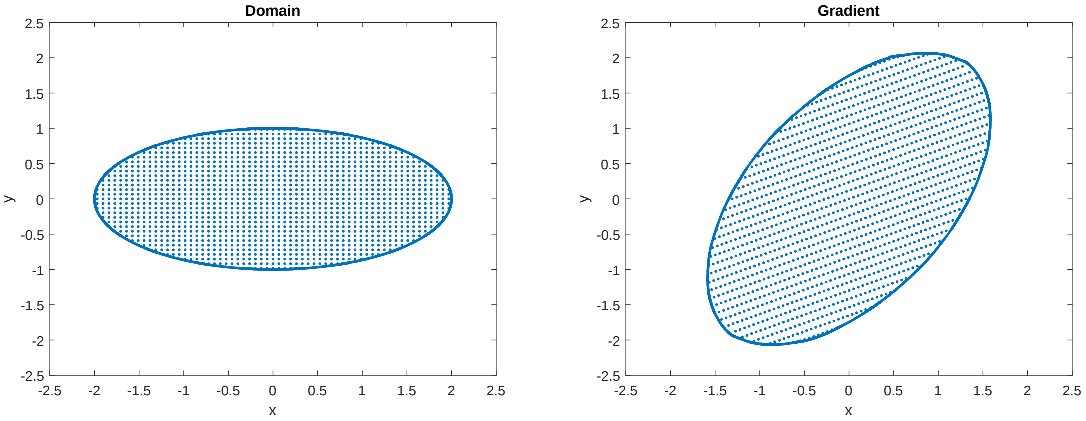

Figure 4.20 An example where ∇u maps an ellipse into a skewed ellipse.

the more complicated target sets. These include the bowl, an ice cream cone, and a pentagon. The maps are pictured in Figures 4.21, 4.22, and 4.23, respectively.

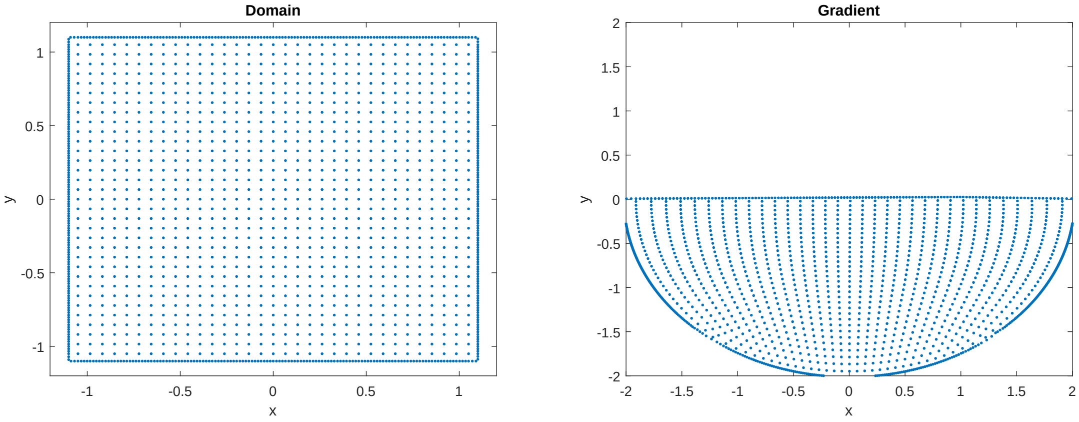

Figure 4.21 An example where ∇u maps a square into a bowl.

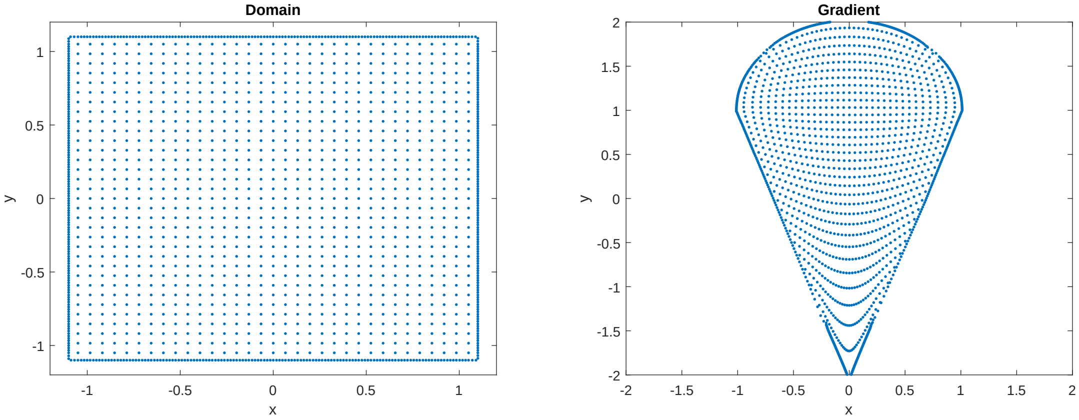

Figure 4.22 An example where ∇u maps a square into an ice cream cone.

One more map we do not have an exact solution for is the unit circle mapped into a square with side lengths 1.1. The map is shown in Figure 4.25. We also converge to a solution and show the map defined by the gradient. The final map we do not have an exact solution for is the unit circle mapped into a line segment. This is a very degenerate example since a line segment is one fewer dimension than the circle. The map is shown in Figure 4.26. We converge to some solution and display the map defined by its gradient.

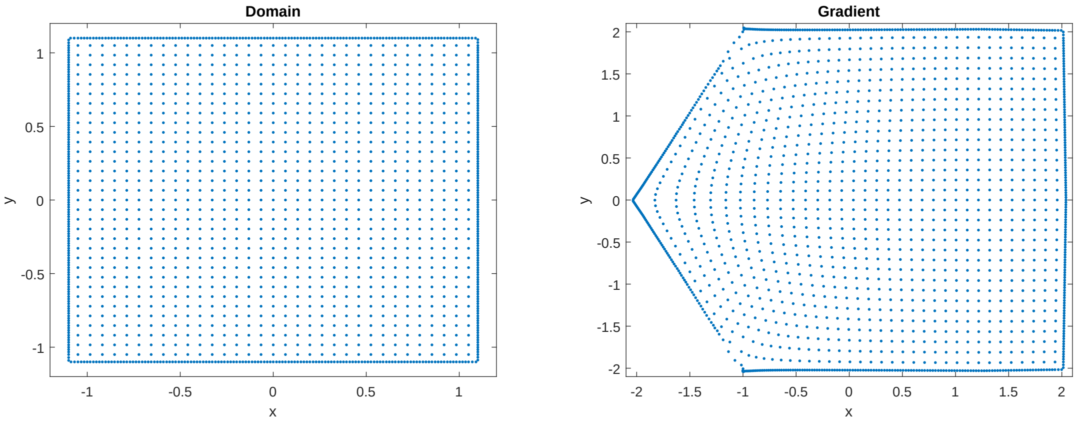

Figure 4.23 An example where ∇u maps a square into a pentagon.

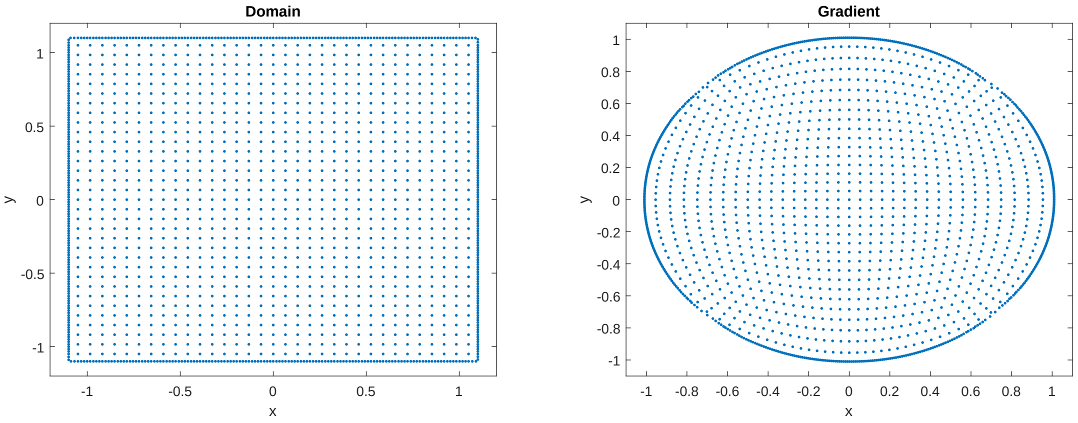

Figure 4.24 An example where ∇u maps a square into a circle.

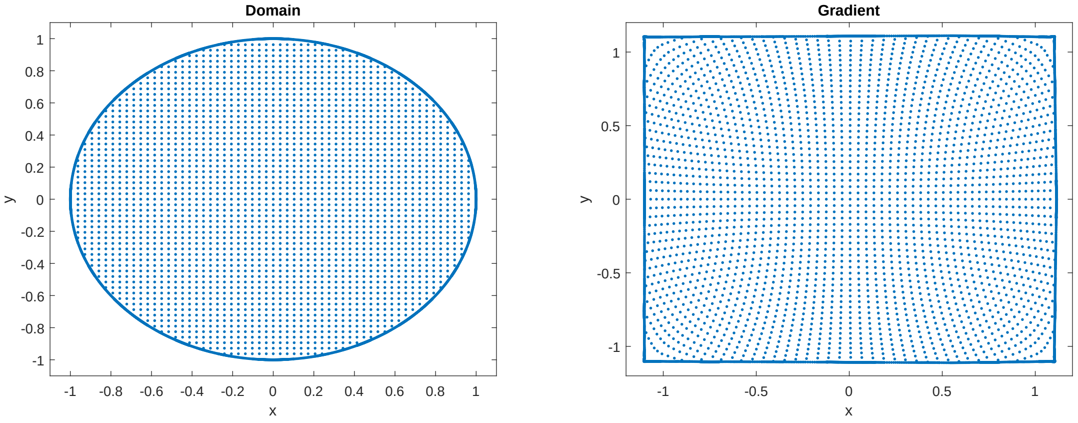

Figure 4.25 An example where ∇u maps a circle into a square.

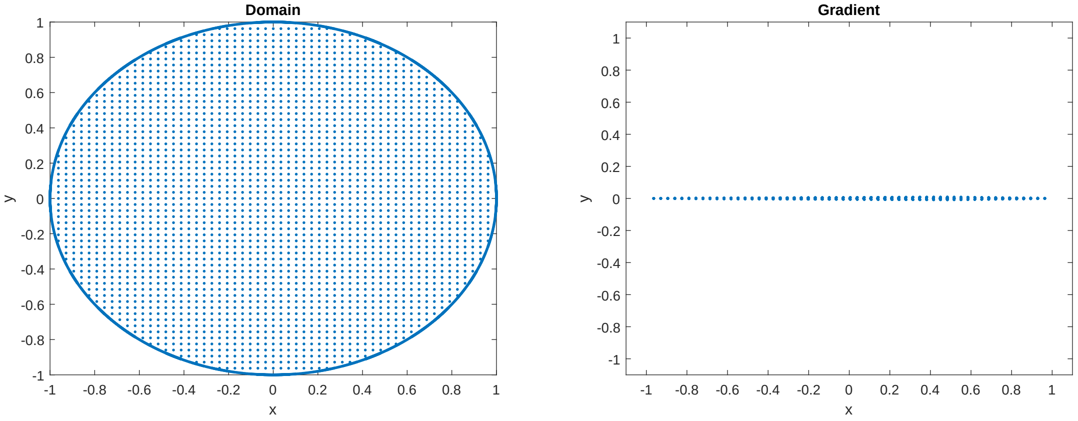

Figure 4.26 An example where ∇u maps a circle into a line segment.

4.9.6 Non-Constant f

One of the more complicated examples we compute involves the solution of the equation

$$\tan^{-1}(\lambda_1) + \tan^{-1}(\lambda_2) = c \cdot f(x,y)$$

where

$$f(x,y) = \tan^{-1}\left(e^{\frac{x^2+y^2-1}{2}}\right) + \tan^{-1}\left((1+x^2+y^2)\cdot e^{\frac{x^2+y^2-1}{2}}\right)$$

This example is contrived so that we have a nice exact solution which maps a unit circle into another unit circle with a different distribution in the interior than the previous example. The function f was chosen so that we have a nice solution $c = 1$, along with

$$u(x,y) = e^{\frac{x^2+y^2-1}{2}}.$$

The map is shown in Figure 4.27, along with an error table including the results in Table 4.5. Even on the non quadratic example, we get $\mathcal{O}(h)$ accuracy, which is better than the expected $\mathcal{O}(\sqrt{h})$.

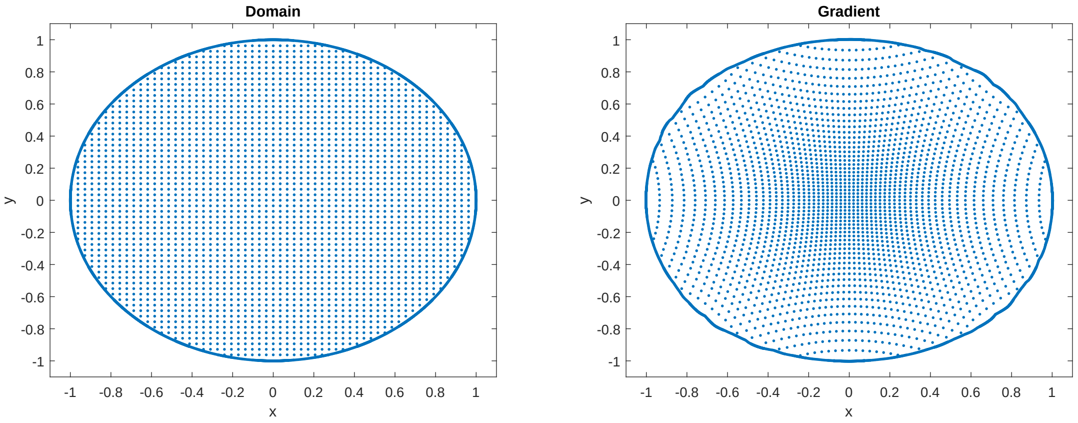

Figure 4.27 An example where ∇u maps a circle into a circle with a non-constant function $f(x)$.

Table 4.5 Circle to Circle (Non-Constant $f(x)$) Convergence

h	Error	Ratio	Observed Order
0.13750	4.85632e-02		
0.06875	2.47727e-02	1.96035	0.97111
0.03438	1.13290e-02	2.18667	1.12874
0.01719	5.91966e-03	1.91379	0.93643
0.00859	2.93863e-03	2.01443	1.01037

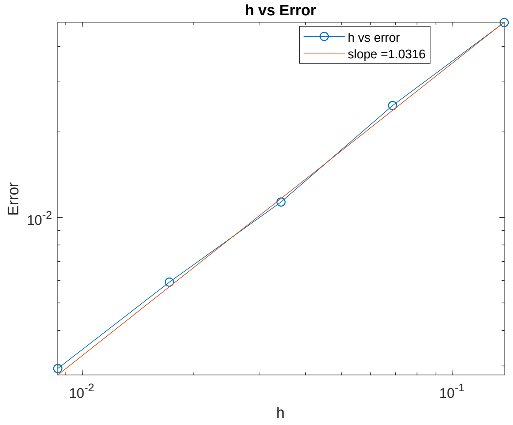

Figure 4.28 A convergence plot for the map from a circle to another circle with a non-constant function $f(x)$.

4.9.7 Functions of the Gradient

Finally, we compute an example where we solve

$$\tan^{-1}(\lambda_1) + \tan^{-1}(\lambda_2) = cf(x, \nabla u)$$

where c is unknown, and

$$f(x, \nabla u) = \frac{\pi(x^2 + y^2 + 1)}{2(u_x + u_y + 1)}$$

This utilizes the approximation for f including gradient terms. The exact solution is the same as the other examples, and we get better than expected accuracy on this example. The results are in Table 4.6 and Figure 4.29. For this example, we observe $\mathcal{O}(h)$ accuracy, which is better than the expected $\mathcal{O}(\sqrt{h})$.

Table 4.6 Circle to Circle (Non-Constant $f(x, \nabla u)$) Convergence

h	Error	Ratio	Observed Order
0.13750	6.60334e-02		
0.06875	3.04057e-02	2.17175	1.11886
0.03438	1.56964e-02	1.93711	0.95390
0.01719	7.72875e-03	2.03091	1.02213
0.00859	3.73824e-03	2.06748	1.04788

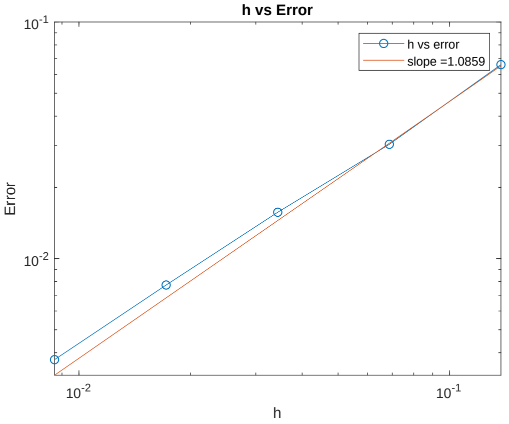

Figure 4.29 A convergence plot for the map from a circle to another circle with a non-constant function $f(x, \nabla u)$.

CHAPTER 5

NUMERICAL METHODS IN THREE DIMENSIONS

5.1 Approach

Building on the work that was done in two dimensions, we devise efficient methods for solving fully nonlinear elliptic partial differential equations in three dimensions. This work is an extension of the original generalized monotone finite difference schemes developed in "Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian" [26]. We still focus on building operators that are functions of second directional derivatives: $F(u_{\nu\nu})$ for ν in some admissible set, which could be finite or infinite.

There are also many additional challenges which we face as compared with the two-dimensional problem. Construction of the domain and boundary is also a non trivial problem since the resolution of the boundary needs to be higher than the resolution of the interior. This is addressed in Section 5.2. One challenge in approximating these is that the second derivatives must be characterized in a monotone way. Entirely new monotone schemes must be derived for three-dimensional problems. One important contribution of this dissertation is the creation of such schemes, which is addressed in Section 5.3. We also deal with functions of the eigenvalues of the Hessian, which involve an infinite admissible set and require a discretization of orthogonal coordinate frames. Solving problems involving eigenvalues of the Hessian requires a new method of approximating those eigenvalues. Since the third dimension adds a third eigenvalue, we cannot simply use the maximum and minimum of directional derivatives anymore. A framework for building monotone approximations of the eigenvalues of the Hessian into a monotone approximations of the full nonlinear operator is given in Section 5.6. Additionally, since we are working in three dimensions, practical concerns of computing resources such as memory and processing speed are brought to the forefront. Strategies to make the construction of the discrete approximation more efficient are presented throughout the chapter.

5.2 Building the Point Cloud

In order to build a point cloud where efficient schemes can be used, we follow an extension of the framework used in “Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian” [26]. Structured grids provide certain advantages in building the stencils, but as in two dimensions, the boundary needs to be more resolved than the interior. In two dimensions, where the boundary is a one-dimensional curve, this is fairly straightforward. However, it is much more difficult to find an optimal sampling of boundary points in three dimensions.

We begin by identifying interior points. One strategy is to begin with a uniform discretization of the minimal cube covering the domain, then reduce to only the interior points. Denote the point cloud by $\mathcal{G}$. Define x_{ijk} to be the nodes of the cube C. Let h be the space between adjacent nodes, and

$$n = \frac{1}{h} + 1.$$

Define

$$G(x) = \begin{cases} \operatorname{dist}(x, \partial X) & x \notin X \\ -\operatorname{dist}(x, \partial X) & x \in X \\ 0 & x \in \partial X \end{cases}$$

These can be found easily by looking at the values of the signed distance function to the boundary of the domain G. As in two dimensions, we require that there be some separation between the interior and the boundary in order to resolve directions there.

Thus, the interior points in $\mathcal{G}$ are

$$x_{ijk} \in C \text{ s.t. } G(x_{ijk}) + \frac{h}{2} < 0.$$

This ensures that there is a distance of at least $\frac{h}{2}$ between the boundary and the interior. Note that although we start with a cube, this restriction can be to arbitrarily complicated three-dimensional regions. This restriction also ensures that

$$\operatorname{dist}(x_{ijk}, \partial X) \geq h/2$$

for every $x_{ijk} \in \mathcal{G} \cap X$. A two-dimensional visualization of this process is shown in Figure 5.1.

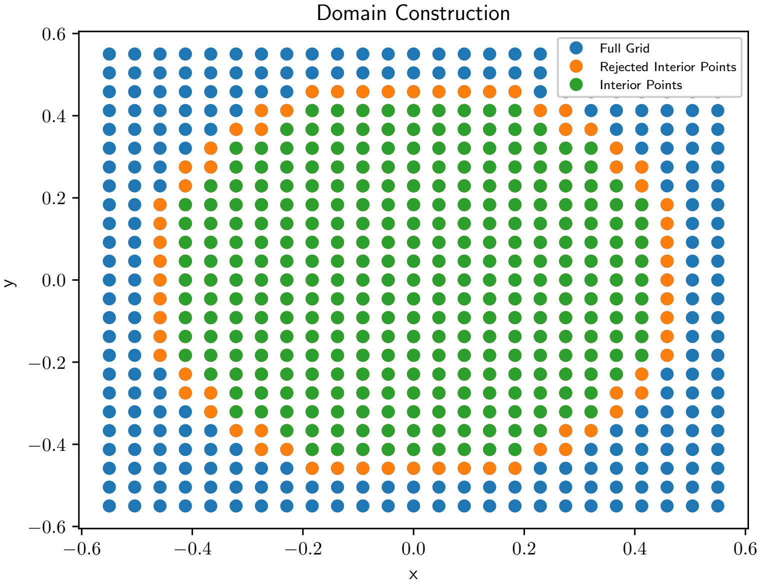

Figure 5.1 The $z = 0$ level set for the sphere. Only the points at least $\frac{h}{2}$ from the boundary are kept in the point cloud.

Next we describe the discretization of the boundary. Consider the following cubes C_{ijk} defined for each x_{ijk}:

$$C_{ijk} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]$$

Define any cube C_{ijk} such that $C_{ijk} \cap \partial X$ is non-empty to be a boundary cube. These can be found using G by taking the set of C_{ijk} such that there is at least one corner $x_- \in C_{ijk}$ with $G(x_-) < 0$ and one corner $x_+ \in C_{ijk}$ with $G(x_+) > 0$. Then boundary points are added to the cloud by further discretizing boundary cubes and taking points sufficiently close to the boundary of the domain. Let C_{ijk} be a boundary cube. We introduce the discretization

$$D_{ijk} = \left\{ (x_i + \tilde{i}\tilde{h}, y_j + \tilde{j}\tilde{h}, z_k + \tilde{k}\tilde{h}) \text{ s.t. } 0 \leq \tilde{i}, \tilde{j}, \tilde{k} \leq \tilde{n} \right\}$$

where $\tilde{n} \approx n^{1/4}$ and $\tilde{h} = \frac{1}{\tilde{n}-1}$. A two-dimensional visualization of the possible boundary points and a closer look at them are given in Figures 5.2 and 5.3.

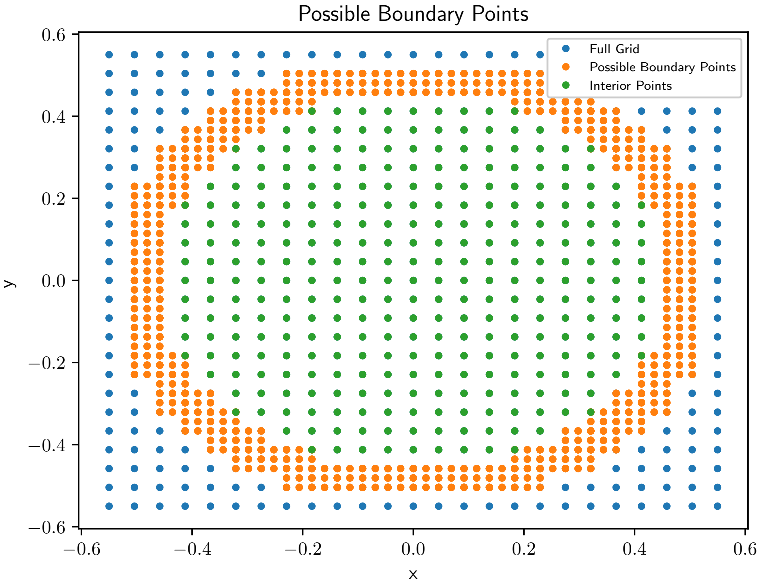

Figure 5.2 The candidate boundary points being considered for the $z = 0$ level set of the sphere example.

We keep points such that

$$G(x) < \frac{\tilde{h}}{2}$$

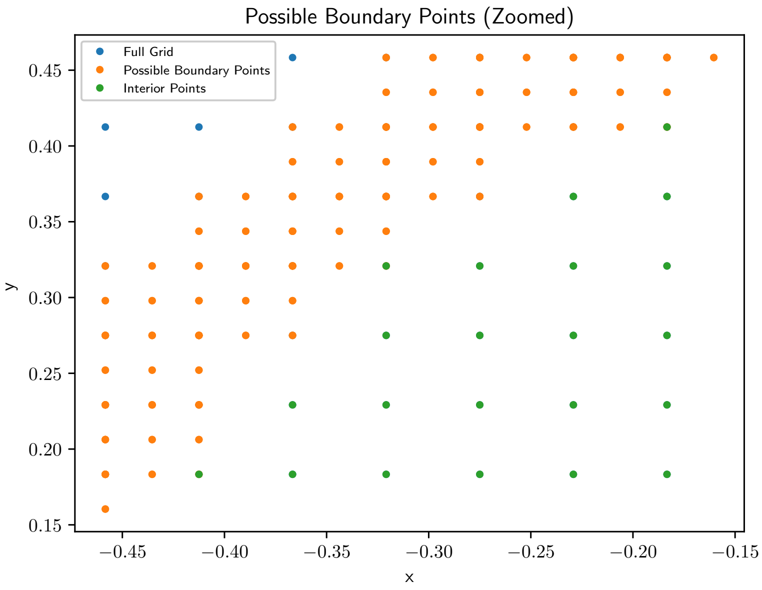

Figure 5.3 The candidate boundary points being considered for the $z = 0$ level set of the sphere example, zoomed in.

in the cloud as boundary points, then project them onto the true boundary. The chosen boundary points for the two-dimensional visualizations in Figures 5.2 and 5.3 are depicted in Figures 5.4 and 5.5.

There are $\mathcal{O}(n^2)$ boundary cubes, each of which are subdivided into $\tilde{n}^3$ boundary points, of which $\mathcal{O}(\tilde{n}^2)$ are taken and projected to the boundary. This is because $\mathcal{O}(\tilde{n}^2)$ points will be within $\tilde{h}/2$ of the boundary surface. This leaves us with

$$\mathcal{O}(n^2\tilde{n}^2) = \mathcal{O}(n^{2+2/4}) = \mathcal{O}(n^{5/2})$$

which is less than the total $\mathcal{O}(n^3)$ interior points. Note that although this resolution is higher than that of a traditional Cartesian grid, it will not significantly affect the overall computational cost, which continues to be dominated by the number of interior points.

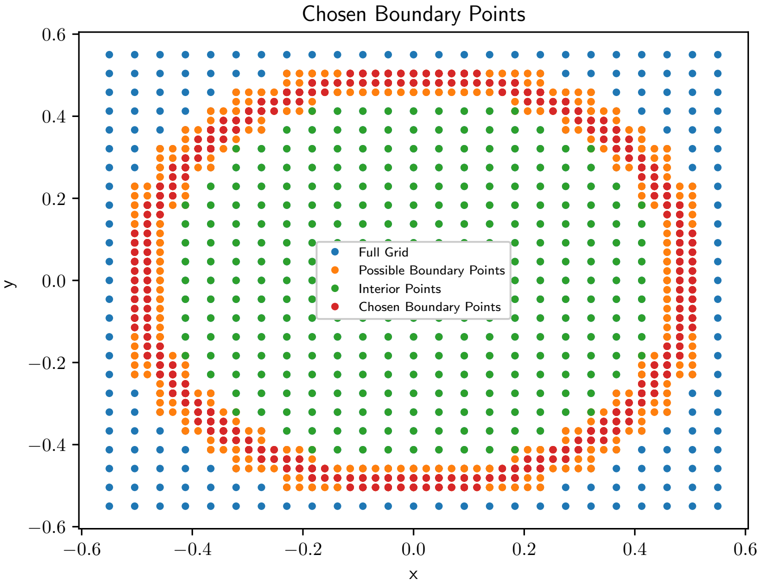

Figure 5.4 The chosen boundary points for the $z = 0$ level set of the sphere example.

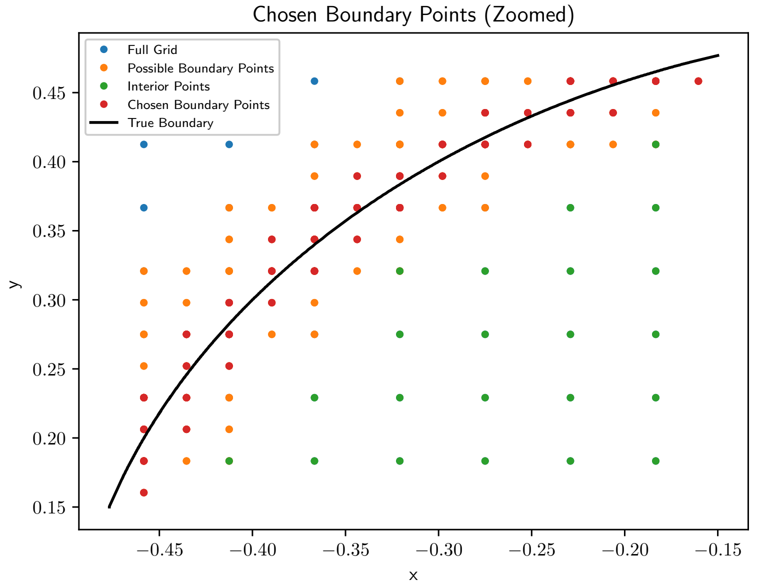

Figure 5.5 The chosen boundary points for the $z = 0$ level set of the sphere example, zoomed in.

5.3 Three-Dimensional Monotone Schemes

The schemes used in two dimensions do not extend readily to three dimensions. In order to find a (negative) monotone approximation, we must find a scheme which is a nondecreasing function of the differences $u_j - u_i$ where u_i is the point of interest and u_j are its neighbors. The simplest way to do this is via Taylor series expansions. Define $D_{\nu\nu}u_i$ to be the discrete approximation to the second directional derivative $u_{\nu\nu}$ in the direction ν at the point x_i. We look for schemes of the form

$$u_{\nu\nu}(x_i) \approx D_{\nu\nu}u_i = \sum_{j=1}^{n} a_j(u_j - u_i) \tag{5.1}$$

where a_j are the coefficients we seek and all $a_j \geq 0$ for (negative) monotonicity. Begin by Taylor expanding the function u about u_i in three dimensions. If we let $a_0 = -\sum_{j=1}^{n}(a_j)$, then we can reformulate the approximation as

$$u_{\nu\nu}(x_i) \approx a_0 u_i + \sum_{j=1}^{n} a_j(u_j). \tag{5.2}$$

In order to handle every possible situation, we derive schemes for both grid aligned and non grid aligned directions. An example of a perfectly grid aligned neighbor is shown in Figure 5.6 (left). An example of neighbors that are not perfectly aligned is given in Figure 5.6 (right).

5.4 Grid Aligned Approximation Scheme

One of the easiest (negative) monotone schemes in one dimension is the second difference operator:

$$u_{xx} \approx \frac{u(x+h) + u(x-h) - 2u(x)}{h^2} \tag{5.3}$$

where h is the space between the points. We would like to reduce the problem of approximating a directional derivative to that of approximating a one-dimensional second derivative. Thus, if we can find perfectly aligned neighbors along a particular

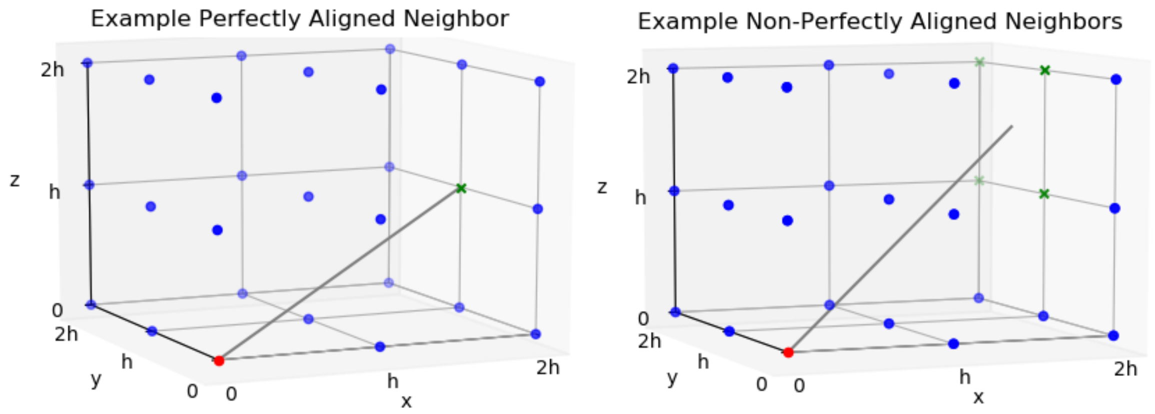

Figure 5.6 (Left) An example of a perfectly aligned neighbor. (Right) An example of four non-perfectly aligned neighbors.

direction, we can use a one-dimensional reduction. However, close to the boundary or otherwise, there may be instances when the neighbors are perfectly aligned but not equally spaced. In order to compensate for this, we derive a scheme via Taylor expansion. Let x_1 and x_2 be points on opposite sides of x_0 and perfectly aligned with the direction of interest. For simplicity, consider the x direction $(1, 0, ..., 0)$ for the derivation. Let $h_1 = x_1 - x_0 > 0$ and $h_2 = x_0 - x_2 > 0$, then

$$u_{xx} \approx a_1 \left(u_0 + h_1 u_x + \frac{h_1^2}{2} u_{xx} + \mathcal{O}(h^3) \right) + a_2 \left(u_0 - h_2 u_x + \frac{h_2^2}{2} u_{xx} + \mathcal{O}(h^3) \right) + a_0 u_0 \tag{5.4}$$

and we have the equations

$$a_0 + a_1 + a_2 = 0 \tag{5.5}$$

$$a_1 h_1 - a_2 h_2 = 0 \tag{5.6}$$

$$a_1 h_1^2/2 + a_2 h_2^2/2 = 1 \tag{5.7}$$

Solving this system leads to

$$a_1 = \frac{2}{h_1^2 + h_1 h_2}, a_2 = \frac{2}{h_2^2 + h_1 h_2}, \text{ and } a_0 = -\frac{2}{h_1^2 + h_1 h_2} - \frac{2}{h_2^2 + h_1 h_2} \tag{5.8}$$

This reduces to the classical second difference if $h_1 = h_2$. In general, we can rotate this scheme to work with any direction ν since it approximates the rate of change along any axis. If the points are perfectly aligned with a given direction ν, this is a second order accurate (negative) monotone approximation of the second directional derivative $u_{\nu\nu}$. Note that for points well away from the boundary we can easily use the underlying Cartesian structure to select neighbors for grid-aligned directions.

5.5 Generalized Finite Difference Approximation Scheme

Near the boundary and elsewhere, there are situations where a (negative) monotone scheme using neighbors that do not align with a direction ν is needed. This happens when there are no perfectly aligned points available for the direction of choice. In this case, we rotate and shift the coordinate frame such that the point of interest is the origin, and the direction of interest is the x-axis. This is accomplished by subtracting the point of interest from its neighbors and taking a dot product of the result with each ν in the frame to project onto the new axes. Then we Taylor expand about the point of interest x_0 which is the origin in this frame. Let O_i be the octants in the frame where x_0 is the origin and ν is the x-axis. Let x_{ji} be points such that

$$\|x_{ji} - x_0\| \leq r(h) = \sqrt{h}$$

which lie in a particular relative octant O_i. Good candidate neighbors are easy to find because of the structure imposed on the grid. Similar to what is done in "Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian" [26], we choose one neighbor x_i in each O_i such that

$$x_i = \text{argmin}_{x_{ji}}(\theta(x_{ji} - x_0) - \theta(\nu))^2 + (\phi(x_{ji} - x_0) - \phi(\nu))^2$$

where $\theta(\cdot)$ and $\phi(\cdot)$ are the θ and ϕ obtained by converting the argument to spherical coordinates. Thus, we take the points closest aligned to the direction as neighbors, so that the orthogonal derivatives are multiplied by small numbers and can be dropped.

We define a coordinate frame for each point:

$$\begin{cases} \bar{x_i} = (x_i - x_0) \cdot \nu_1, \\ \bar{y_i} = (y_i - y_0) \cdot \nu_2, \\ \bar{z_i} = (z_i - z_0) \cdot \nu_3, \end{cases}$$

Then in spherical coordinates for this frame, we have

$$\begin{cases} r_i = \sqrt{(\bar{x_i}^2 + \bar{y_i}^2 + \bar{z_i}^2)} = \sqrt{(((x_i - x_0) \cdot \nu_1)^2 + ((y_i - y_0) \cdot \nu_2)^2 + ((z_i - z_0) \cdot \nu_3)^2)}, \\ d\phi_i = \cos^{-1} (\frac{\bar{z_i}}{r_i}) = \cos^{-1} (\frac{(z_i - z_0) \cdot \nu_3}{r_i}), \\ d\theta_i = \tan^{-1} \left(\frac{\bar{y_i}}{\bar{x_i}}\right) \end{cases}$$

Via Taylor expansion we have for any orthogonal frame (x, y, z):

$$\begin{aligned} u(x, y, z) \approx & u(x_0) + u_x(x_0, y_0, z_0)(x - x_0) + u_y(x_0, y_0, z_0)(y - y_0) \\ & + u_z(x_0, y_0, z_0)(z - z_0) + u_{xx}(x_0, y_0, z_0)\frac{(x - x_0)^2}{2} \\ & + u_{yy}(x_0, y_0, z_0)\frac{(y - y_0)^2}{2} + u_{zz}(x_0, y_0, z_0)\frac{(z - z_0)^2}{2} \\ & + u_{xy}(x_0, y_0, z_0)(x - x_0)(y - y_0) + u_{xz}(x_0, y_0, z_0)(x - x_0)(z - z_0) \\ & + u_{yz}(x_0, y_0, z_0)(y - y_0)(z - z_0) + h.o.t. \end{aligned}$$

In an approximation for u_{xx}, if one chooses neighbors well aligned with this direction, then $|(y - y_0)|$ and $|(z - z_0)|$ are much smaller than $|x - x_0|$. Squaring these terms or multiplying them together, we have something even smaller. Thus, the dominant terms in the expansion would be

$$\begin{aligned} u(x, y, z) \approx & u(x_0) + u_x(x_0, y_0, z_0)(x - x_0) + u_y(x_0, y_0, z_0)(y - y_0) \\ & + u_z(x_0, y_0, z_0)(z - z_0) + u_{xx}(x_0, y_0, z_0)\frac{(x - x_0)^2}{2} \end{aligned}$$

Now consider in particular the orthogonal frame in which ν is the x-axis. If the neighbors are well aligned with the direction ν, the terms related to the orthogonal directions are

$$|y_i - y_0|, |z_i - z_0| \leq \mathcal{O}(\epsilon \sin d\theta_i \sin d\phi_i + \epsilon \cos d\phi_i) = \mathcal{O}(\epsilon d\phi) \ll \mathcal{O}(\epsilon) = |x_i - x_0|$$

where ϵ is the search radius. Then we also neglect the higher order terms which are $o(\epsilon^2)$ and in spherical coordinates we get the following system of equations:

$$\begin{cases} \sum_{i=1}^n a_i r_i \cos d\theta_i \sin d\phi_i = 0 \\ \sum_{i=1}^n a_i r_i \sin d\theta_i \sin d\phi_i = 0 \\ \sum_{i=1}^n a_i r_i \cos d\phi_i = 0 \\ \sum_{i=1}^n a_i \frac{r_i^2}{2} \cos d\theta_i{}^2 \sin d\phi_i{}^2 = 1. \end{cases} \tag{5.9}$$

We have four equations, and need to build a (negative) monotone scheme using some number of unknowns. This requires $a_i \geq 0$. In three dimensions, it is difficult to find extra equations that ensure that the system is well-posed and satisfies the monotonicity requirement. However, a solution can be easily found numerically if we know it exists. It turns out that the analytical solution of this is extremely complicated. Instead, we prove that a solution (and therefore a (negative) monotone scheme) exists for the underdetermined system and compute it using a least squares solver with the constraint that all of the coefficients must be nonnegative.

This involves solving a problem of the form

$$\begin{cases} \min \frac{1}{2} \|Ax - b\|_2 \\ \text{s.t. } x \geq 0 \end{cases} \tag{5.10}$$

where the rows of A and b are from the system of equations in Equation (5.9) and the x we are solving for is the set of coefficients $a_i \geq 0$.

This scheme is $\mathcal{O}(\sqrt{h})$ accurate as in two dimensions, since the angular error scales as the inverse tangent of the spacing h divided by the search radius $\sqrt{h}$, which is $\frac{h}{\sqrt{h}} = \sqrt{h}$. The angular resolution is discussed in Section 4.2.

5.5.1 Farkas' Lemma

In order to prove the existence of this scheme, we use Farkas' Lemma. This section will state and prove the lemma as presented in [52].

Lemma 5.1. Farkas' Lemma [52] Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^{m\times 1}$. Then exactly one of the following two conditions holds:

- $\exists x \in \mathbb{R}^{n\times 1}$ such that $Ax = b$ and all components of x are non-negative;
- $\exists y \in \mathbb{R}^{1\times m}$ such that $A^T y \geq 0, y^T b < 0$

Proof. First, we show that the two cannot happen at the same time. Suppose both conditions are true, then

$$y^T Ax = y^T b < 0$$

since $Ax = b$, and $y^T b < 0$. Note also that

$$y^T Ax = (y^T A)x = (A^T y)^T x \geq 0$$

since $A^T y \geq 0$ and $x \geq 0$. So we have a contradiction with $y^T Ax$ being both less than zero and greater than or equal to zero so the two are mutually exclusive [52].

Next show that if the first condition does not hold, then the second condition does. Let $v_1, v_2, ...v_n$ be the columns of A. Define

$$Q = \text{cone}(v_1, ..., v_n) \equiv \{s \in \mathbb{R}^m : s = \sum_{i=1}^{n} \lambda_i v_i, \lambda_i \geq 0,\ \forall i\}. \tag{5.11}$$

This is a conic combination of the columns of A, which unlike a convex combination does not require the λ_i to sum to one. Then

$$Ax = \sum_{i=1}^{n} x_i v_i$$

and there exists an x such that $Ax = b$ and $x \geq 0$ iff $b \in Q$ [52].

Suppose the first condition does not hold. Then $b \notin Q$. We know that Q is non-empty since $0 \in Q$, it is also closed and convex. Thus, we can apply the separating hyperplane theorem. This implies $\exists \alpha \in \mathbb{R}^m, \alpha \neq 0$, and β such that $\alpha^T b > \beta$ and $\alpha^T s < \beta$ for all $s \in Q$. Since $0 \in Q$, we know that $\beta > 0$. Note also that $\lambda v_i \in Q$ for all $\lambda > 0$. Then, since $\alpha^T s < \beta \, \forall s \in Q$, we have $\alpha^T(\lambda v_i) \in Q \, \forall \lambda > 0$.So $\alpha^T v_i < \frac{\beta}{\lambda} \, \forall \lambda > 0$. Since $\beta > 0$, as $\lambda \to \infty$, we have $\alpha^T v_i \leq 0$. If we set $y = -\alpha$, we get that $y^T b < 0$ and $y^T v_i \geq 0 \, \forall i$. Since v_i are the columns of A, we get that $A^T y \geq 0$. Thus, the second condition holds [52]. □

5.5.2 Existence of a Positive Solution

In this section, we use Farkas' Lemma to prove that there exists a positive solution to the system of equations in Equation (5.9), and therefore a (negative) monotone scheme exists and the numerical solver should succeed. We also show the existence of (negative) monotone schemes using some perfectly aligned neighbors and some non perfectly aligned neighbors. These are both important as either scheme may be useful near the boundary.

Eight Neighbors

Lemma 5.2. A positive solution to the system of equations in Equation (5.9) exists for $n = 8$ if for each neighbor x_i, $(x_i - x_0)$ is in a different octant of the coordinate frame in which the direction ν is the x-axis.

Proof. Consider the choice of neighbors such that each lies in a different octant of the new coordinate frame. Then each expression in the equations has a known, definite sign for each component. Indicating the signs of

$$(r_i \cos d\theta_i \sin d\phi_i, r_i \sin d\theta_i \sin d\phi_i, r_i \cos d\phi_i)$$

we have neighbors with

$$(+,+,+), (-,+,+), (-,-,+), (+,-,+)$$

$$(+,+,-), (-,+,-), (-,-,-), (+,-,-)$$

Then we know the signs of everything in the system we are solving. Each coefficient is some constant with known sign. Let $c_{ij} > 0$, then the system we are solving is $Ax = b$ where

$$A = \begin{bmatrix} c_{11} & -c_{12} & -c_{13} & c_{14} & c_{15} & -c_{16} & -c_{17} & c_{18} \\ c_{21} & c_{22} & -c_{23} & -c_{24} & c_{25} & c_{26} & -c_{27} & -c_{28} \\ c_{31} & c_{32} & c_{33} & c_{34} & -c_{35} & -c_{36} & -c_{37} & -c_{38} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} & c_{46} & c_{47} & c_{48} \end{bmatrix} \tag{5.12}$$

and

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.13}$$

As per Farkas' Lemma, consider $A^T y \geq 0$ with $b^T y < 0$.

$$A^T = \begin{array}{cc} \begin{bmatrix} c_{11} & c_{21} & c_{31} & c_{41} \\ -c_{12} & c_{22} & c_{32} & c_{42} \\ -c_{13} & -c_{23} & c_{33} & c_{43} \\ c_{14} & -c_{24} & c_{34} & c_{44} \\ c_{15} & c_{25} & -c_{35} & c_{45} \\ -c_{16} & c_{26} & -c_{36} & c_{46} \\ -c_{17} & -c_{27} & -c_{37} & c_{47} \\ c_{18} & -c_{28} & -c_{38} & c_{48} \end{bmatrix} & \begin{array}{c} (1r) \\ (2r) \\ (3r) \\ (4r) \\ (5r) \\ (6r) \\ (7r) \\ (8r) \end{array} \end{array} \tag{5.14}$$

and

$$b^T y =< 0, 0, 0, 1 > \cdot < y_1, y_2, y_3, y_4 > \tag{5.15}$$

so

$$b^T y < 0 \Rightarrow y_4 < 0. \tag{5.16}$$

Consider the solution set for this equation. If $y_4 < 0$, there are eight possibilities:

$$\begin{bmatrix} \text{(1b)}\ y_1, y_2, y_3 \geq 0 & \text{(2b)}\ y_2, y_3 \geq 0, y_1 \leq 0 \\ \text{(3b)}\ y_1, y_2, y_3 \leq 0 & \text{(4b)}\ y_1, y_2 \leq 0, y_3 \geq 0 \\ \text{(5b)}\ y_1, y_2 \geq 0, y_3 \leq 0 & \text{(6b)}\ y_1, y_3 \leq 0, y_2 \geq 0 \\ \text{(7b)}\ y_1, y_3 \geq 0, y_2 \leq 0 & \text{(8b)}\ y_2, y_3 \leq 0, y_1 \geq 0 \end{bmatrix} \tag{5.17}$$

Note that each possibility yields a solution with at least one negative component: (1b) would make Equation (7r) negative; (2b) would make Equation (8r) negative; (3b) would make Equation (1r) negative; (4b) would make Equation (5r) negative; (5b) would make Equation (3r) negative; (6b) would make Equation (4r) negative; (7b) would make Equation (6r) negative; (8b) would make Equation (2r) negative.

Thus, there is no solution to $A^T y \geq 0$ with $b^T y < 0$. Therefore, by Farkas' lemma, there exists a solution to $Ax = b$ with $x \geq 0$. □

It turns out that this neighbor picking strategy has the same effect in any dimension, and one can build positive or negative monotone approximations in this way for n dimensions using 2^n neighbors, one from each hyperoctant.

Five Neighbors

Lemma 5.3. Suppose at least one neighbor is perfectly aligned with the direction ν. A positive solution to the system of equations in Equation (5.9) exists for $n = 5$ if for each other neighbor x_j, $(x_j - x_0)$ is in a different octant opposite ν in the coordinate frame in which the direction ν is the x-axis.

Proof. Suppose the scheme is derived as above, but with five neighbors such that one neighbor aligns perfectly with the desired direction and the other four lie in the four octants opposite that neighbor. Then each expression in the equations has a known, definite sign for each component. Indicating the signs of

$$(r_i \cos d\theta_i \sin d\phi_i, r_i \sin d\theta_i \sin d\phi_i, r_i \cos d\phi_i)$$

we have neighbors with the following signs

$$(-,+,+),(-,-,+),(-,-,-),(-,+,-),$$

along with the fifth neighbor, which has $d\theta, d\phi = 0$. The system of equations is then

$$a_5 r_5 + \sum_{i=1}^{4} a_i r_i \cos d\theta_i \sin d\phi_i = 0$$

$$\sum_{i=1}^{4} a_i r_i \sin d\theta_i \sin d\phi_i = 0$$

$$\sum_{i=1}^{4} a_i r_i \cos d\phi_i = 0$$

$$a_5 r_5^2/2 + \sum_{i=1}^{4} a_i \frac{r_i^2}{2} \cos d\theta_i{}^2 \sin d\phi_i{}^2 = 1$$

Then we know the signs of everything in the system we are solving. Each coefficient is some constant with known sign. Let $c_{ij} > 0$, then the system we are solving is $Ax = b$ where

$$A = \begin{bmatrix} -c_{11} & -c_{12} & -c_{13} & -c_{14} & c_{15} \\ c_{21} & -c_{22} & -c_{23} & c_{24} & 0 \\ c_{31} & c_{32} & -c_{33} & -c_{34} & 0 \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \end{bmatrix} \tag{5.18}$$

and

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.19}$$

As per Farkas' Lemma, consider $A^T y \geq 0$ with $b^T y < 0$.

$$A^T = \left[\begin{array}{cccc} -c_{11} & c_{21} & c_{31} & c_{41} \\ -c_{12} & -c_{22} & c_{32} & c_{42} \\ -c_{13} & -c_{23} & -c_{33} & c_{43} \\ -c_{14} & c_{24} & -c_{34} & c_{44} \\ c_{15} & 0 & 0 & c_{45} \end{array}\right] \begin{array}{c} (1r) \\ (2r) \\ (3r) \\ (4r) \\ (5r) \end{array} \tag{5.20}$$

and

$$b^T y =< 0, 0, 0, 1 > \cdot < y_1, y_2, y_3, y_4 > \tag{5.21}$$

so

$$b^T y < 0 \Rightarrow y_4 < 0. \tag{5.22}$$

Consider the solution set for this equation. If $y_4 < 0$, there are eight possibilities:

$$\begin{bmatrix} \text{(1b)}\ y_1, y_2, y_3 \geq 0 & \text{(2b)}\ y_2, y_3 \geq 0, y_1 \leq 0 \\ \text{(3b)}\ y_1, y_2, y_3 \leq 0 & \text{(4b)}\ y_1, y_2 \leq 0, y_3 \geq 0 \\ \text{(5b)}\ y_1, y_2 \geq 0, y_3 \leq 0 & \text{(6b)}\ y_1, y_3 \leq 0, y_2 \geq 0 \\ \text{(7b)}\ y_1, y_3 \geq 0, y_2 \leq 0 & \text{(8b)}\ y_2, y_3 \leq 0, y_1 \geq 0 \end{bmatrix} \tag{5.23}$$

Then, for Equation (5r) to be non-negative, this implies $y_1 \geq 0$, which eliminates possibilities (2b),(3b),(4b), and (6b). This leaves us with

$$\begin{bmatrix} \text{(1b)}\ y_1, y_2, y_3 > 0 & - \\ - & - \\ \text{(5b)}\ y_1, y_2 > 0, y_3 < 0 & - \\ \text{(7b)}\ y_1, y_3 > 0, y_2 < 0 & \text{(8b)}\ y_2, y_3 < 0, y_1 > 0 \end{bmatrix} \tag{5.24}$$

Note that each possibility yields a solution with at least one negative component: (1b) would make Equation (3r) negative, and (7b) would make Equation (4r) negative, so we are left with

$$\begin{bmatrix} \text{(5b)}\ y_1, y_2 > 0, y_3 < 0 \\ \text{(8b)}\ y_2, y_3 < 0, y_1 > 0 \end{bmatrix} \tag{5.25}$$

(5b) would make Equation (2r) negative and (8b) would make Equation (1r) negative, so we are left with an empty solution set. Thus, there is no solution to $A^T y \geq 0$ with $b^T y < 0$. Therefore, by Farkas' lemma, there exists a solution to $Ax = b$ with $x \geq 0$. □

Corollary 5.4. Lemma 5.3 also applies with neighbors surrounding ν and a perfectly aligned neighbor in the $-\nu$ direction.

5.6 Approximating Eigenvalues in Three Dimensions

In order to approximate the eigenvalues of the Hessian matrix in three dimensions, we must use properties from linear algebra to provide a new formulation. We would like to simply use

$$\lambda_1 = \min_{\nu_1}\{u_{\nu_1\nu_1}\}, \quad \lambda_3 = \max_{\nu_3}\{u_{\nu_3\nu_3}\} \tag{5.26}$$

again, but this does not provide any insight into the third eigenvalue. We may consider using

$$\lambda_2 = \min_{\nu_2 \perp \nu_1}\{u_{\nu_2\nu_2}\} \tag{5.27}$$

but this does not necessarily yield a monotone scheme, since a perturbation of an individual directional derivative could change the minimum in a way that increases or decreases the operator. Focusing on the Lagrange curvature problem as a model operator, the following approximation can be used in three dimensions or higher:

$$G\left(\sum_{j=1}^{d} \phi(\lambda_j(D^2u))\right) = \max_{(\nu_1,\nu_2,\dots\nu_d)\in V}\left\{G\left(\sum_{j=1}^{d}\phi(u_{\nu_j\nu_j})\right)\right\} \tag{5.28}$$

where the directions are considered in orthonormal frames:

$$V = \{(\nu_1,...,\nu_d) | \nu_j \in \mathbb{R}^d, \nu_i \neq \nu_j \text{ if } i \neq j, ||\nu_j||_2 = 1, \nu_j \perp \nu_i \; \forall i \neq j\} \tag{5.29}$$

Lemma 5.5 (Functions of eigenvalues)**.** Let $G : \mathbb{R} \to \mathbb{R}$ be non-increasing, $\phi : \mathbb{R} \to \mathbb{R}$ concave, and A a symmetric real-valued $d \times d$ matrix. Then

$$G\left(\sum_{j=1}^{d} \phi(\lambda_j(A))\right) = \max_{(\nu_1,\nu_2,\dots\nu_d)\in V}\left\{G\left(\sum_{j=1}^{d}\phi(\nu_j^T A \nu_j)\right)\right\}. \tag{5.30}$$

Proof. Since A is a real-valued symmetric matrix, we can find d orthonormal eigenvectors $v_1, \dots, v_d$. Any $(\nu_1, \nu_2, ...\nu_d) \in V$ can be expressed as a linear combination of these eigenvectors:

$$\nu_j = \sum_{k=1}^{d} c_{jk} v_k = \sum_{k=1}^{d} (\nu_j^T v_k) v_k.$$

Since ν_j and v_j are both orthonormal, we can also compute

$$\sum_{k=1}^{d} c_{jk}^2 = \left(\sum_{k=1}^{d} c_{jk} v_k \right) \left(\sum_{l=1}^{d} c_{jl} v_l \right) = \nu_j^T \nu_j = 1,$$

$$\sum_{k=1}^{d} c_{jk}^2 = v_k^T \left(\sum_{j=1}^{d} \nu_j \nu_j^T \right) v_k = v_k^T I v_k = v_k^T v_k = 1.$$

Then note that

$$\nu_j^T A \nu_j = \left(\sum_{k=1}^{d} c_{jk} v_k^T \right) A \left(\sum_{k=1}^{d} c_{jk} v_k \right) = \left(\sum_{k=1}^{d} c_{jk} v_k^T \right) \left(\sum_{k=1}^{d} c_{jk} \lambda_k v_k \right) = \sum_{k=1}^{d} c_{jk}^2 \lambda_k.$$

Recall Jensen's Inequality for concave functions ϕ:

$$\phi \left(\frac{\sum_j c_j x_j}{\sum_j c_j} \right) \geq \frac{\sum_j c_j \phi(x_j)}{\sum_j c_j}$$

Now for any unit vector ν_j, we can use Jensen's inequality to estimate

$$\phi(\nu_j^T A \nu_j) = \phi \left(\sum_{k=1}^{d} c_{jk}^2 \lambda_k \right) \geq \sum_{k=1}^{d} c_{jk}^2 \phi(\lambda_k).$$

Summing these concave functions yields

$$\sum_{j=1}^{d} \phi(\nu_j^T A \nu_j) \geq \sum_{j=1}^{d} \sum_{k=1}^{d} c_{jk}^2 \phi(\lambda_k) = \sum_{k=1}^{d} \phi(\lambda_k)$$

with equality if the $(\nu_1, \ldots, \nu_d)$ coincide with the eigenvectors $(v_1, \ldots, v_d)$ of A.

Since G is non-increasing, we conclude that

$$G \left(\sum_{j=1}^{d} \phi(\lambda_j(A)) \right) = \max_{(\nu_1, \nu_2, \ldots \nu_d) \in V} \left\{ G \left(\sum_{j=1}^{d} \phi(\nu_j^T A \nu_j) \right) \right\}. \qquad \square$$

Remark 5.6. This also applies if G is non-decreasing and ϕ is convex, and if the maximum is replaced with a minimum, G can be non-decreasing with ϕ concave, or non-increasing with ϕ convex.

In particular, this can be applied to many important operators, such as the Lagrange curvature problem and the Monge-Ampére equation. In order to keep the solution to the Lagrange curvature problem in the space of convex functions, we propose the alternate operator

$$-\sum_{j=1}^{3}\left(\tan^{-1}\left(\max\{\lambda_j,0\}\right)+\min\{\lambda_j,0\}\right). \tag{5.31}$$

Let $G(x) = -x$ and $\phi(x) = \tan^{-1}(\max\{x,0\}) + \min\{x,0\}$. Then the Lagrange curvature problem can be expressed as

$$G\left(\sum_{j=1}^{d}(\phi_1(\lambda_j(D^2u)))\right) = c. \tag{5.32}$$

For the Monge-Ampére equation, since it is only elliptic on the space of convex functions, we follow "A Numerical Method for the Elliptic Monge-Ampère Equation with Transport Boundary Conditions" [25] and use the globally elliptic extension

$$-\max\{\lambda_1,0\}\max\{\lambda_2,0\}\max\{\lambda_3,0\}-(\min\{\lambda_1,0\}+\min\{\lambda_2,0\}+\min\{\lambda_3,0\})+f=0. \tag{5.33}$$

Let $\phi_1(x) = \log\max\{x,0\}$, $G_1(x) = -e^x$, $\phi_2(x) = \min\{x,0\}$, and $G_2(x) = -x$. Then the Monge-Ampére equation can be re-expressed as

$$G_1\left(\sum_{j=1}^{3}\phi_1(D^2u(\mathbf{x}))\right)+G_2\left(\sum_{j=1}^{3}\phi_2(D^2u(\mathbf{x}))\right)+f(\mathbf{x})=0,$$

similar to [27].

Next, we define the discrete operator. Due to the fact that we are working in three dimensions, approximating every directional derivative for every point becomes computationally infeasible very quickly. Instead, we must consider some subset of the orthogonal frames containing the true minimizing frame. Let

$$d\theta = \max_{v_1,\ldots,v_d \in V} \min_{(\nu_1,\ldots,\nu_d)\in V^h} \max_i \cos^{-1}(v_i \cdot \hat{\nu}_i)$$

where

$$V = \{(\nu_1, ..., \nu_d) | \nu_j \in \mathbb{R}^d, \nu_i \neq \nu_j \text{ if } i \neq j, ||\nu_j||_2 = 1, \nu_j \perp \nu_i \, \forall i \neq j\}$$

That is, for each frame in V, we first find the frame in V^h that minimizes the worst case angle between v_i and ν_i. Then, we maximize over all possible frames in V to find the worst case $d\theta$. Consider a representative finite set of orthonormal frames V^h such that $V^h \subset V$ and the $d\theta$ error approaches zero as $h \to 0$. Define the discrete operator:

$$F^h(u_i - u_j) = \max_{(\nu_1,\nu_2,\ldots\nu_d)\in V^h} \left\{ G\left(\sum_{j=1}^{d} \phi(D_{\nu_j\nu_j} u) \right) \right\} + f(x, y, z) \tag{5.34}$$

In order to use this approximation, we must show that it is monotone and consistent.

5.6.1 Monotonicity

Lemma 5.7.

$$G^h(u_i - u_j) = \max_{(\nu_1,\nu_2,\ldots\nu_d)\in V^h} \left\{ G\left(\sum_{k=1}^{d} \phi(D_{\nu_k\nu_k} u) \right) \right\}$$

is a monotone approximation of the operator given $D_{\nu_k\nu_k}u$ as defined by Equation (5.1).

Proof. Since $G\left(\sum_{k=1}^{d} \phi(D_{\nu_k\nu_k} u)\right)$ is an elliptic operator, it is a non-increasing function of the eigenvalues of the Hessian. Then since $D_{\nu_k\nu_k}u$ is negative monotone, it is a non-increasing function of the differences $u_i - u_j$. Substituting it in place of the eigenvalues leads to a non-decreasing function of each $u_i - u_j$. □

5.6.2 Consistency

In order to guarantee convergence, we also need the operator to be a consistent approximation of the operator and the eigenvalues involved.

Lemma 5.8. F^h as defined in Equation (5.34) is consistent.

Proof. Note that $D_{\nu_j\nu_j}u$ is a consistent approximation of $u_{\nu_j\nu_j}$. By continuity of the minimum function and its arguments, approximating V by V^h introduces a $d\theta$ error. We choose $d\theta \to 0$ as $h \to 0$ to achieve a consistent approximation for the eigenvalues. Then since $\tan^{-1}(\cdot)$ is continuous, $F^h[u] \to F(D^2u)$ as $h \to 0$. □

We now have consistency and monotonicity provided that the approximation for the directional derivatives is (negative) monotone. This scheme then fits within the Barles and Souganidis framework for Dirichlet boundary conditions, and within the framework from Chapter 3 for Transport boundary conditions and eigenvalue problems.

5.7 Construction of Orthogonal Frames

In this section, we discuss the evaluation the expression in Equation (5.34), which requires computing a minimum over many different orthogonal frames. This can be very expensive in three dimensions, so we define a multi-level approach to approximate the minimum. For each integer stencil width k, define

$$V_k = \{\nu \in \mathbb{Z}^3 st. \|\nu\|_\infty \leq k\}.$$

Our goal is to construct orthogonal frames out of these V_k.

For each stencil width k begin with a sampling $V_k^{(1)}$ of ν_1 directions in V_k such that for all distinct ν_{1i}, ν_{1j}, $\nu_{1i} \cdot \nu_{1j} \neq 0$.

We then define a set of ν_2 directions $V_k^{(2,\nu_1)}$ for each direction $\nu_1 \in V_k^{(1)}$ such that $\nu_2 \perp \nu_1$.

Finally, a ν_3 direction is computed for each ν_1 and ν_2 pairing as $\nu_3 = \nu_1 \times \nu_2$.

In order to define grid aligned neighbors for as many of the frames as possible, we discard any frame in which any $|\nu_{3i}| > k$.

Any ν_1 directions which have no associated ν_2 directions for which there is a valid ν_3 direction are also discarded.

The list of directions $V_k^{(1)}$ that can be formed from V_k for ν_1 can be created by brute force and saved for use in many different PDEs since it is computed once. An example list of ν_1 directions is given in Table 5.1.

Table 5.1 Example ν_1 Directions ($k = 2$)

(1,1,1),	(0,1,1),	(-1,1,1)
(0,0,1),	(1,2,2),	(-1,2,2)
(2,1,2),	(1,1,2),	(0,1,2)
(-1,1,2),	(-2,1,2),	(1,0,2)
(-1,0,2),	(0,-1,2)	

Then the lists of ν_2 directions $V_k^{(2,\nu_1)}$ such that $\nu_2 \perp \nu_1$ and $\max_i\{|\nu_i|\} \leq k$ can also be computed by brute force and saved for later use.

We next outline a strategy to use the lists $V_k^{(1)}$ and $V_k^{(2,\nu_1)}$ to determine appropriate sets of directions V^h for each h. We propose a multi-level approach where the stencil width widens after the solution of the problem at the previous level. This allows the angular error to decrease at each level and approach zero as $h \to 0$, while balancing memory and efficiency concerns. With a search radius of $\sqrt{h}$, a stencil width of

$$\text{round}\left(\frac{\sqrt{h}}{h}\right) = \text{round}(n^{1/2})$$

will lead to an angular error which scales like

$$\tan^{-1}\left(\frac{\text{distance between adjacent points}}{\text{search radius}}\right) = \tan^{-1}\left(\frac{h}{\sqrt{h}}\right) = \mathcal{O}(\sqrt{h}).$$

Hence, for consistency we require $k = \mathcal{O}(\sqrt{n})$.

After computing all possible coordinate frames, we build a hierarchy in order to reduce the memory cost of the number of directions being considered. In order to do this, we first create a map from each ν_1 in $V_k^{(1)}$ to the five closest aligned ν_1 directions in $V_{k+1}^{(1)}$.

Then, since each $\nu_1 \in V_{k+1}^{(1)}$ has multiple $\nu_2 \in V_{k+1}^{(2,\nu_1)}$ associated with it, we create a second map.

We would like to check a subset of the ν_2 directions in $V_{k+1}^{(2,\nu_1)}$ to reduce the memory and computational cost of approximating all directional derivatives for the given directions.

In order to determine the best candidate frames closest to the previous active frame, we also map the ν_2 for each $\nu_1 \in V_k^{(2,\nu_1)}$ to the closest five $\nu_2 \in V_{k+1}^{(2,\nu_1)}$ valid for each $\nu_1 \in V_{k+1}^{(1)}$ chosen as candidates by the ν_1 map.

The end result is a set of maps for each ν_1 direction that define the next ν_2 directions based on the previous ν_2 direction.

We can save these lists and maps, then compute each ν_3 using the cross product.

In order to utilize the precomputed hierarchy, first solve the full problem using $k = 1$ and record which frame achieved the minimum. Next, use the ν_1 map to generate a new list of five ν_1 directions considering a slightly wider stencil. Once this list is computed, substitute the previous active ν_2 direction into the ν_2 map appropriate for each ν_1 direction to get a list of five ν_2 directions per ν_1 direction. Finally, take the cross product $\nu_1 \times \nu_2$ to get a ν_3 direction for each pairing. This results in 25 total frames to check which lie in a cone around the previous active frames. An illustration is shown in Figure 5.7.

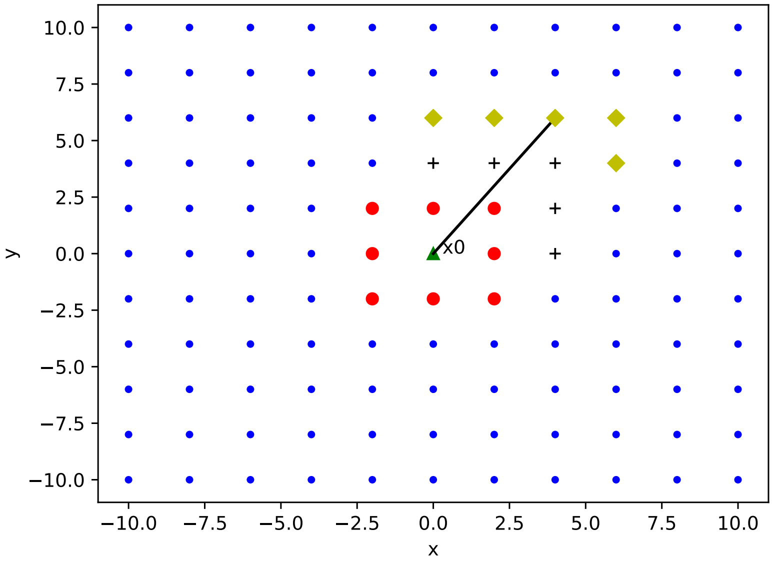

Figure 5.7 A two-dimensional illustration of the multi-level process for one direction ν_1 in the orthogonal frame. The true eigenvector direction is given by the black line. In the first level, we maximize over all the nearest (red dot) neighbors. We then identify the five (black plus) neighbors of stencil width two most closely aligned with the maximizer. After maximizing over these five neighbors, we continue the procedure by identifying the best five (yellow diamond) neighbors of stencil width three.

5.8 Neumann Boundary Conditions

In addition to solving problems with Dirichlet boundary conditions, it is desirable to solve problems with Neumann boundary conditions:

$$\frac{\partial u}{\partial \hat{n}} = g$$

where n is the unit outward normal to the boundary at each point. Given a smooth boundary and a smooth defining function for the boundary, we can compute n for each point of interest, or approximate it by using the signed distance function for the domain G.

$$n_1 = \left(\frac{G(x+\epsilon, y, z) - G(x-\epsilon, y, z)}{2\epsilon} \right)$$

$$n_2 = \left(\frac{G(x, y+\epsilon, z) - G(x, y-\epsilon, z)}{2\epsilon} \right)$$

$$n_3 = \left(\frac{G(x, y, z+\epsilon) - G(x, y, z-\epsilon)}{2\epsilon} \right)$$

where $\epsilon = 10^{-7}$ to get to machine precision. Once (n_1, n_2, n_3) is computed, normalizing it gives the unit outward normal $\hat{n}$ at each boundary point.

In order to use this framework, we must first derive a scheme to approximate this boundary condition at each boundary point. The simplest approach is through Taylor expansion:

$$u(x) \approx u(x_0) + \nabla u(x_0) \cdot (x - x_0) + \mathcal{O}(x - x_0)^2$$

Since $\frac{\partial u}{\partial \hat{n}} = \nabla u \cdot \hat{n}$, a scheme of the form

$$\nabla u \cdot \hat{n} \approx \sum_j a_j (u_j - u_i)$$

can be derived by setting the coefficients of the gradient terms to the components of $\hat{n}$.

$$\begin{aligned} \sum_j a_j h_j &= \hat{n}_1 \\ \sum_j a_j k_j &= \hat{n}_2 \\ \sum_j a_j l_j &= \hat{n}_3 \\ a_j &\leq 0 \end{aligned} \tag{5.35}$$

where $\hat{n}_i$ are the components of the unit outward normal $\hat{n}$, and the h_j, k_j, l_j are the x, y, and z spacing, respectively. In the next subsection, we simultaneously show a solution to this system and an efficient method of selecting neighbors.

5.8.1 Selection of Neighbors

We look for neighbors in the interior to use for a monotone approximation of the first directional derivatives at boundary points. Due to the structure of the grid, the interior is composed of small cubes

$$C_{ijk} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]. \tag{5.36}$$

As a simple way of selecting appropriate neighbors, we let C_{ijk} be the first such cube (5.36) entered by the ray $\mathbf{x}_0 - t\mathbf{n}$. We choose as neighbors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ the four vertices of the face through which this ray enters the small cube. Using these neighbors, we can show that a solution to the system (5.35) exists.

Lemma 5.9 (Existence of a negative solution)**.** A negative solution to the system of equations in Equation (5.35) exists if $\mathbf{x}_0 - t\hat{n}$ lies in the convex hull of the four vertices $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ of a square for some $t > 0$.

Proof. Since $\mathbf{x}_0 - t\hat{n}$ lies in the convex hull of the four corners of a square, then it also lies in the convex hull of three of these points. Without loss of generality, let these be $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$. Then there exist $\lambda_1, \lambda_2, \lambda_3 \in [0, 1]$ with $\lambda_1 + \lambda_2 + \lambda_3 = 1$ such that

$$\mathbf{x}_0 - tn = \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \lambda_3 \mathbf{x}_3.$$

Now we let $v(x)$ be the piecewise linear interpolant of the values of $u(x)$ at the points $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^3$. Since v is linear, we can compute its first directional derivative in the direction n via

$$\begin{aligned} \frac{\partial v}{\partial n} &= \frac{v(\mathbf{x}_0) - v(\mathbf{x}_0 - t\hat{n})}{t} \\ &= \frac{v(\mathbf{x}_0) - \lambda_1 v(\mathbf{x}_1) - \lambda_2 v(\mathbf{x}_2) - \lambda_3 v(\mathbf{x}_3)}{t}. \end{aligned}$$

Then we can easily verify that

$$a_1 = -\frac{\lambda_1}{t},\ a_2 = -\frac{\lambda_2}{t},\ a_3 = -\frac{\lambda_3}{t},\ a_4 = 0$$

is a solution of (5.35). □

5.9 Transport Boundary Conditions

Having outlined what is needed to solve problems with Neumann boundary conditions, we can utilize the framework used in two dimensions to solve problems with transport boundary conditions:

$$H(\nabla u) = 0$$

where H is the signed distance function for the target set. This can be done using the same Legendre-Fenchel transform formulation as in two dimensions:

$$\sup_{n \cdot n_x > 0} \{\nabla u \cdot n - H^*(n)\}$$

By approximating this as a maximum over a discrete set of n_d values, we can approximate the boundary condition. To discretize the directions, choose $\tilde{n} = 9 + \sqrt{n}$.

Then let θ be a partition of $(0, 2\pi)$ into $\tilde{n}$ values, and ϕ be a partition of $(0, \pi)$ into $\tilde{n}/2$ values. Define the n_d directions as the all unique directions given by

$$(\cos(\theta)\sin(\phi), \sin(\theta)\sin(\phi), \cos(\phi)).$$

As with Neumann boundary conditions, it is necessary to use the eigenvalue formulation with an extra unknown c in order to satisfy solvability conditions. This is currently memory intensive, since we consider a larger and larger number of directions for each boundary point as n increases.

5.10 Parallelization

The run time of this work can be easily reduced using parallel computing. Indeed, the current implementation used by the author uses multiprocessing in Python. Since the computation of neighbors and coefficients for each orthogonal frame are independent, each of the three orthogonal directions can be assigned to a different processor. This was done in order to speed up the algorithm by assigning one processor to each of the three frames to be computed simultaneously. Through this, the stencils can be computed faster than the solution can be.

5.11 Solution Methods

We also describe the techniques used in this implementation to solve the discrete system of nonlinear equations arising from our approximations. In three dimensions, it is not practical to explicitly build Jacobian matrices due to their prohibitively large sizes. Moreover, many of the PDEs we consider are degenerate and/or have singular solutions. Thus, Newton's method is not immediately suitable for these problems.

Our approach here is to use a combination of an active set approach [6], which has excellent stability properties for the nonlinear systems we consider, and a simple Gauss-Seidel iteration. Depending on the particular PDE of interest, the solver may collapse into only one of these methods or it may involve a combination. In the future, this approach could be accelerated using a nonlinear multigrid method.

Recall that we are trying to solve systems of the form

$$\max_{\boldsymbol{\nu}\in V^h} F(\mathbf{x}_i, u_i, \mathcal{D}_{\boldsymbol{\nu}} u_i, \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u_i) = 0, \quad \mathbf{x}_i \in \mathcal{G}. \tag{5.37}$$

The same approach works if the maximum above is replaced by a minimum.

The basic approach is to iterate through a two step process. First, for a given input u, we identify the directions $\boldsymbol{\nu}_i \in V^h$ that maximize (5.37) at each point in the domain. Secondly, we fix this direction and seek an approximate solution of

$$F(\mathbf{x}_i, u_i, \mathcal{D}_{\boldsymbol{\nu}_i} u_i, \mathcal{D}_{\boldsymbol{\nu}_i\boldsymbol{\nu}_i} u_i) = 0.$$

In order to implement the second part of the procedure, we recall that our finite difference systems can be written in the form

$$F(\mathbf{x}_i, u_i, \mathcal{D}_{\boldsymbol{\nu}} u_i, \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u_i) = G_{\boldsymbol{\nu}}(\mathbf{x}_i, u_i, u_j)$$

where the $\mathbf{x}_j$ are points designated as neighbors of $\mathbf{x}_i$. We design a Gauss-Seidel iteration for this by solving for the reference value u_i in terms of the values at neighboring grid points. That is, we identify a function $G_{\boldsymbol{\nu}}^{-1}(\mathbf{x}_i, u_j)$ such that

$$G_{\boldsymbol{\nu}}(\mathbf{x}_i, G_{\boldsymbol{\nu}}^{-1}(\mathbf{x}_i, u_j), u_j) = G_{\boldsymbol{\nu}}(\mathbf{x}_i, u_i, u_j).$$

In some cases (for linear or simple nonlinear operators), this function $G_{\boldsymbol{\nu}}^{-1}$ can be identified explicitly. For example if

$$G(\mathbf{x}_i, u_i, u_j) = -\frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} + f(\mathbf{x}_i)$$

approximates a simple second derivative then

$$G^{-1}(\mathbf{x}_i, u_j) = \frac{u_{i+1} + u_{i-1}}{2} - \frac{1}{2} f(\mathbf{x}_i) h^2.$$

In more complicated examples, this inverse can be obtained (or approximated) through several iterations of a nonlinear solver such as a scalar Newton's method. This would involve using the update

$$u_i = u_i - \frac{G_{\boldsymbol{\nu}}(\mathbf{x}_i, u_i, u_j)}{\frac{\partial}{\partial u_i} G_{\boldsymbol{\nu}}(\mathbf{x}_i, u_i, u_j)}.$$

The resulting solution method is described in Algorithm 5.1. This solver is simple to implement and memory efficient since there is no need to construct the Jacobian matrix. In practice, we can initialize the method with the solution computed on a less refined grid.

Algorithm 5.1 Solution method for (5.37)

```
while Residual > Tolerance do
    for $\mathbf{x}_i \in \mathcal{G}$ do
        $\boldsymbol{\nu}_i = \text{argmax}_{\boldsymbol{\nu} \in V^h} G_{\boldsymbol{\nu}}(\mathbf{x}_i, u_i, u_j)$.
    end for
    for $k = 1, \dots, 10$ do
        for $\mathbf{x}_i \in \mathcal{G}$ do
            $u_i = G_{\boldsymbol{\nu}_i}^{-1}(\mathbf{x}_i, u_j)$.
        end for
    end for
end while
```

In order to establish an appropriate stopping criterion, we use the infinity norm of the difference between sets of ten iterations. It turns out that the using the residual or the norm of the operator as a stopping criterion does not always lead to convergence for these examples. Instead, stopping when the difference between iterations is small leads to more accurate solutions.

When finished, we save the active direction set as well as the solution in order to improve initial guesses on a finer grid.

5.12 Eigenvalue Problems

In order to be able to use the approximations from the previous section to solve general problems involving Neumann or Transport boundary conditions, a compatibility condition must be satisfied. In general, this is not exactly satisfied at the discrete level even when satisfied at the continuous level. As in two dimensions, we can deal with this problem by introducing an extra unknown constant c and solving an eigenvalue problem with an extra condition that $u_0 = 0$. We can do this without changing the problem since in general, all parts of the problems we are solving involve only derivatives of the function u. Thus, a constant shift will still be a solution to the problem. The introduction of the extra constant works exactly the same as it does in two dimensions.

5.13 Computational Results, Examples, and Figures

We demonstrate the effectiveness of the method by solving a variety of computational examples including a range of challenging nonlinear PDEs and different boundary conditions. For each result, we give the distance between two neighboring points along one axis h, the L^∞ error of the approximation for each h, and the log scale slope between successive h values.

5.13.1 Linear Degenerate Equation

Consider a linear degenerate equation similar to the one in "Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian" [26].

$$\begin{cases} -u_{\nu\nu}(x,y,z) = 0, & x^2+y^2+z^2 < 1 \\ u(x,y,z) = \sin\left(2\pi(x - \sqrt{2}y - \sqrt{3}z)\right), & x^2+y^2+z^2 = 1 \end{cases} \tag{5.38}$$

where $\nu = (1, -1, \frac{-(\sqrt{3}+\sqrt{6})}{3})$. Note that the direction ν is not aligned with any Cartesian grid. For this example, neither the grid aligned scheme we derived nor any other grid aligned scheme can be used for a consistent monotone approximation [26, 41]. For this example, the eight neighbor meshfree schemes are used exclusively since the other scheme cannot find perfectly aligned neighbors. The exact solution is

$$u(x, y, z) = \sin\left(2\pi(x - \sqrt{2}y - \sqrt{3}z)\right)$$

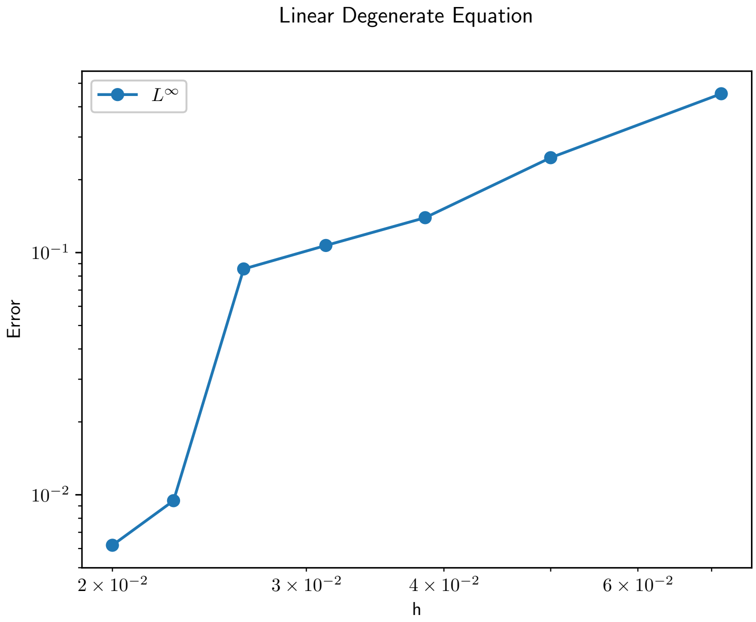

Figure 5.8 A convergence plot for the Linear Degenerate Equation on a sphere.

We present the convergence plot in Figure 5.8 and a data table in Table 5.2. Note that the jump between $h = .0263158$ and $h = .0227273$ is due to the scaling of the number of boundary points being based on round$(n^{1/4})$. This is when round$(n^{1/4})$ increases from 2 to 3 and the boundary becomes much more resolved as a result. The eight neighbor scheme is set to use the best available points, so extra resolution on the boundary can lead to a great increase in accuracy for grids where the angular

Table 5.2 Linear Degenerate Equation Error Table

h	L^∞ Error	Observed Order
0.0714286	0.452733	
0.05	0.246499	1.70447
0.0384615	0.139358	2.17376
0.03125	0.107001	1.27244
0.0263158	0.0855902	1.29917
0.0227273	0.00945624	15.0262
0.02	0.00619057	3.31406

error is dominant over the spatial discretization error. We observe better accuracy than the expected $\mathcal{O}(\sqrt{h})$.

5.13.2 Two Operators

For a second example, consider the following fully nonlinear PDE:

$$\begin{cases} \max\{-u_{\nu_1\nu_1}, -u_{\nu_2\nu_2}\} = f(x,y,z) & x^2+y^2+z^2<1 \\ \nu_1 = (1,1,0) & \\ \nu_2 = (-1,0,1) & \\ u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}} & x^2+y^2+z^2=1 \end{cases} \tag{5.39}$$

where

$$f(x,y,z) = \max\left\{-\frac{1}{2}e^{\frac{x^2+y^2+z^2}{2}}(2+x^2+2xy+y^2), -\frac{1}{2}e^{\frac{x^2+y^2+z^2}{2}}(2+x^2-2xz+z^2))\right\}$$

This is a more complicated operator similar to the linear degenerate equation. Since the directions chosen for the problem are integer directions, we can approximate the directional derivatives accurately using the grid aligned scheme in the interior. In general there will not be perfectly aligned points near the boundary, so the eight neighbor meshfree scheme is used near the boundary. The exact solution is

$$u(x, y, z) = e^{\frac{x^2+y^2+z^2}{2}}$$

The convergence plot and the data table are presented in Figure 5.9 and Table 5.3,

$$\max\left(-u_{\nu_1\nu_1}, -u_{\nu_2\nu_2}\right) = f(x, y, z)$$

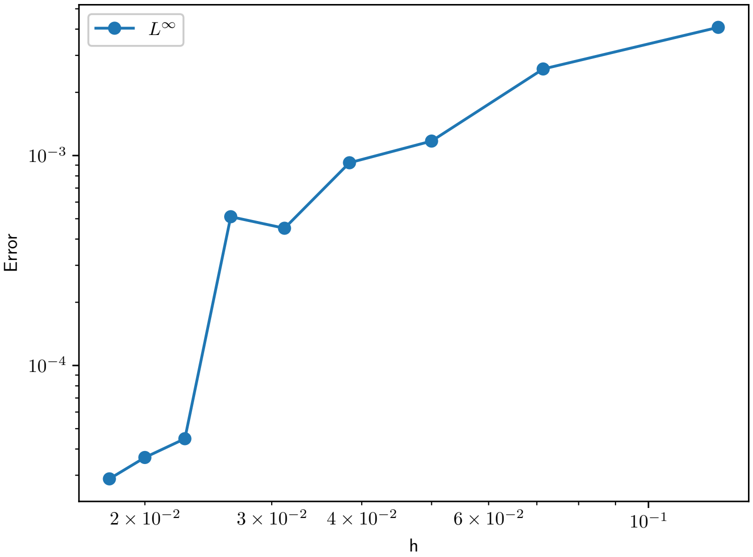

Figure 5.9 A convergence plot for the two-operator problem on a sphere.

respectively. As with the linear degenerate equation, the jump between $h = .0263158$ and $h = .0227273$ is due to the scaling of the number of boundary points being based on round$(n^{1/4})$. We also observe better accuracy than the expected $\mathcal{O}(\sqrt{h})$ for this problem.

5.13.3 Convex Envelope Equation

Consider as another example the convex envelope equation.

$$\begin{cases} \max\left\{-\lambda_1(D^2u), u - g\right\} = 0 & x^2 + y^2 + z^2 < 0.25 \\ u = 0.2 & x^2 + y^2 + z^2 = 0.25 \end{cases} \tag{5.40}$$

Table 5.3 Two-Operator Error Table

h	L^∞ Error	Observed Order
0.125	4.07982e-03	
0.0714286	2.58436e-03	0.815875
0.05	1.17185e-03	2.21739
0.0384615	9.23987e-04	0.905781
0.03125	4.5052e-04	3.45935
0.0263158	5.11281e-04	-0.736213
0.0227273	4.47889e-05	16.6091
0.02	3.64554e-05	1.61047
0.0178571	2.88353e-05	2.0691

where

$$g(x, y, z) = \min\left\{2\sqrt{(x^2 + y^2 + z^2)}, 0.2\right\}$$

This is a fully nonlinear elliptic partial differential equation. Additionally, the solution must be understood in a weak sense because the solution is only Lipschitz continuous and not differentiable at $(0, 0, 0)$. The exact solution for this problem is

$$u(x, y, z) = 0.4\sqrt{(x^2 + y^2 + z^2)}$$

For each step in the Gauss-Seidel type update in the solver, we evaluate

$$\min\left\{\lambda_1(D^2u), u - g\right\}$$

and update using the Newton update for the active operator. This is effectively the same as a policy iteration, but involves no matrices.

The convergence plot and the data table are presented in Figure 5.10 and Table 5.4, respectively.

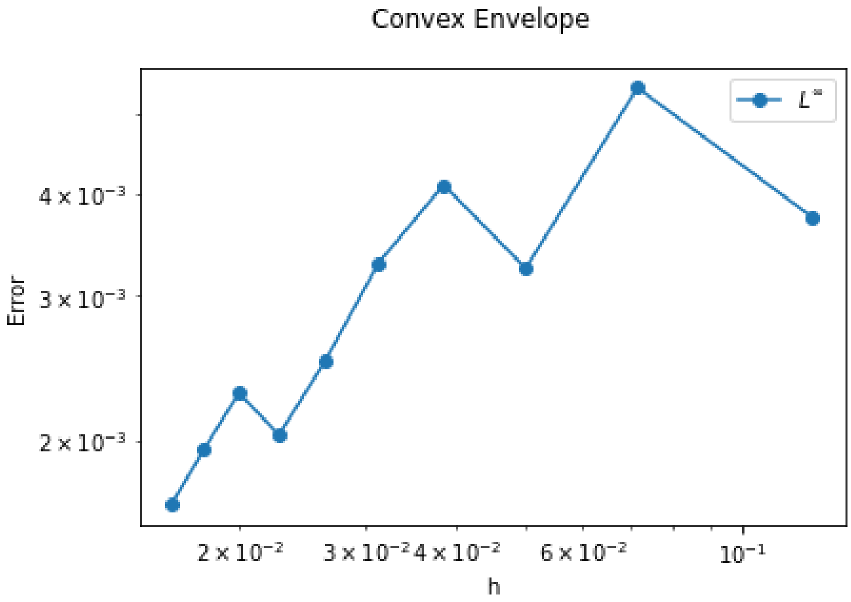

Figure 5.10 A convergence plot for the Convex Envelope Equation on a sphere.

Table 5.4 Convex Envelope Error Table

h	L^∞ Error	Observed Order
0.125	0.0037424	
0.0714286	0.00538204	-0.649265
0.05	0.00324458	1.41889
0.0384615	0.00409974	-0.891659
0.03125	0.00329443	1.05322
0.0263158	0.00250006	1.60556
0.0227273	0.00203438	1.40601
0.02	0.00228924	-0.923316
0.0178571	0.00195022	1.41431
0.016129	0.00167566	1.49076

Note that this is not monotone convergence due to effects of variations in the alignment of the grid points for different n. Despite this, we observe overall convergence close to $\mathcal{O}(\sqrt{h})$ even on this problem where the regularity is low.

5.13.4 Poisson's Equation

For the next three examples, we demonstrate the effectiveness of the method for solving functions of the eigenvalues of the Hessian. First, consider something as simple as Poisson's equation, which can be expressed as

$$\lambda_1(D^2u(x,y,z)) + \lambda_2(D^2u(x,y,z)) + \lambda_3(D^2u(x,y,z)) = f(x,y,z) \tag{5.41}$$

In order to use an example where the method will not be exact, we choose

$$\begin{cases} \sum_i \lambda_i(D^2u(x,y,z)) = e^{\frac{x^2+y^2+z^2}{2}}(3+x^2+y^2+z^2) & x^2+y^2+z^2<1 \\ u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}} & x^2+y^2+z^2=1 \end{cases} \tag{5.42}$$

In order to approximate the eigenvalues, we needed to minimize

$$\min\left(u_{\nu_1\nu_1} + u_{\nu_2\nu_2} + u_{\nu_3\nu_3}\right)$$

over all frames such that ν_1, ν_2, ν_3 are orthogonal. Thus, both the grid aligned scheme and the meshfree scheme are used at various points to approximate a sampling of those frames, and then we take the discrete minimum. The results of this approximation are in Figure 5.11 and Table 5.5. The exact solution is

$$u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}}$$

5.13.5 Monge-Ampére Equation

While Poisson's equation is linear and can be expressed as a sum of the eigenvalues of the Hessian, the Monge-Ampére equation is fully nonlinear and can be expressed

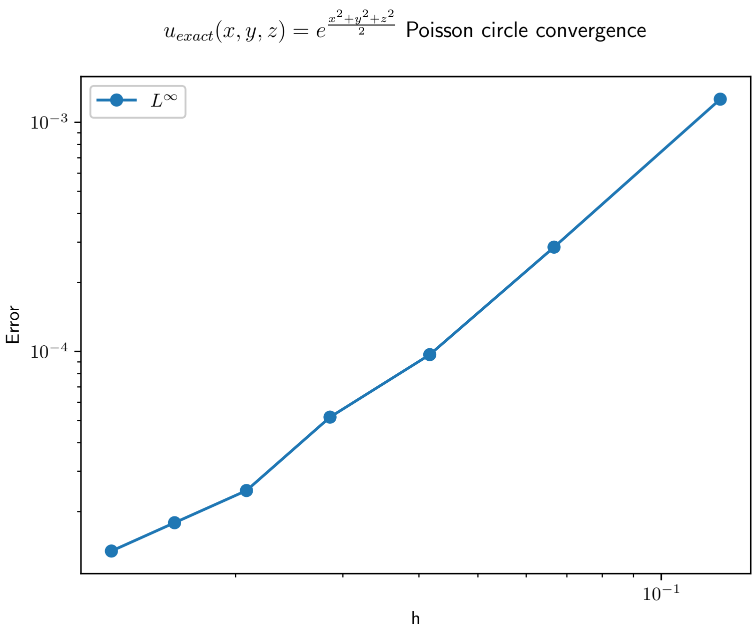

Figure 5.11 A convergence plot for Poisson's Equation on a sphere.

Table 5.5 Poisson's Equation Error Table

h	L^∞ Error	Observed Order
0.125	0.00126191	
0.0666667	0.000285287	2.36535
0.0416667	9.68666e-05	2.2982
0.0285714	5.16758e-05	1.6654
0.0208333	2.47164e-05	2.33503
0.015873	1.78873e-05	1.18918
0.0125	1.34561e-05	1.19156

as their product. Normally, the Monge-Ampére equation can be expressed as:

$$\begin{cases} -\det\left(D^2u\right)+f=0, & x\in X \\ u=g & x\in\partial X \\ u \text{ is convex.} \end{cases} \tag{5.43}$$

The determinant can be expressed as a product of the eigenvalues:

$$-\det\left(D^2u\right)=-\lambda_1\lambda_2\lambda_3$$

As in "Meshfree Finite Difference Approximations for Functions of the Eigenvalues of the Hessian" [26], since the equation is only elliptic in the space of convex functions, we can use the globally elliptic extension from "A Numerical Method for the Elliptic Monge-Ampère Equation with Transport Boundary Conditions" [25]:

$$-(\max\{\lambda_1,0\}\max\{\lambda_2,0\}\max\{\lambda_3,0\}+\min\{\lambda_1,0\}+\min\{\lambda_2,0\}+\min\{\lambda_3,0\})+f=0 \tag{5.44}$$

This formulation of the Monge-Ampére equation leads us to approximate $-\det\left(D^2u\right)$ by

$$\begin{aligned} -\min\{&\max\{u_{\nu_1\nu_1},0\}\max\{u_{\nu_2\nu_2},0\}\max\{u_{\nu_3\nu_3},0\} \\ &+\min\{u_{\nu_1\nu_1},0\}+\min\{u_{\nu_2\nu_2},0\}+\min\{u_{\nu_3\nu_3},0\}\} \end{aligned} \tag{5.45}$$

where ν_1, ν_2, and ν_3 are mutually orthogonal. We can use this to approximate the operator by applying the grid aligned and meshfree schemes to approximate the second directional derivatives for a discrete subset of the orthogonal frames, and then taking the discrete minimum.

Consider the specific example analogous to the one used for the other problems:

$$\begin{cases} -\det\left(D^2u\right)+e^{\frac{3}{2}(x^2+y^2+z^2)}(1+x^2+y^2+z^2)=0, & x^2+y^2+z^2<.25 \\ u=e^{\frac{x^2+y^2+z^2}{2}} & x^2+y^2+z^2=.25 \\ u \text{ is convex.} \end{cases} \tag{5.46}$$

with the exact solution being

$$u(x, y, z) = e^{\frac{x^2+y^2+z^2}{2}}$$

The results of the approximation are included in Figure 5.12 and Table 5.6. On this example, we also observe better than the expected $\mathcal{O}(\sqrt{h})$ convergence.

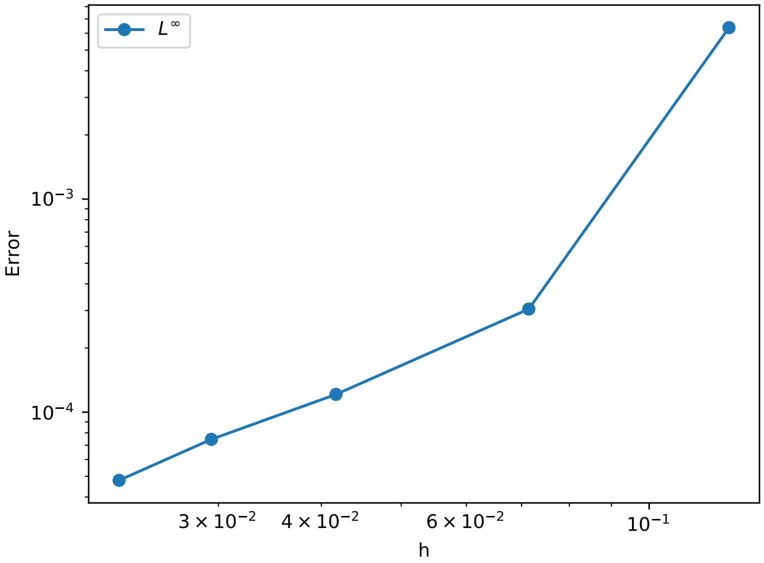

Figure 5.12 A convergence plot for the Monge-Ampére Equation on a sphere.

Table 5.6 Monge-Ampére Equation Error Table

h	L^∞ Error	Observed Order
0.125	0.00637493	
0.0714286	0.000304859	5.43279
0.0416667	0.000121219	1.71105
0.0294118	7.45852e-05	1.39435
0.0227273	4.7917e-05	1.71614

5.13.6 Lagrangian Curvature Problem

Finally, we consider a variation of the Lagrangian Curvature problem from [7]. Specifically, we solve

$$\begin{cases} \sum_{j=1}^{3} \tan^{-1}\left(\lambda_j(D^2u)\right) = f(x,y,z) & x^2+y^2+z^2<1 \\ u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}} & x^2+y^2+z^2=1 \end{cases} \tag{5.47}$$

where

$$f(x,y,z) = 2\tan^{-1}\left(e^{\frac{x^2+y^2+z^2}{2}}\right) + \tan^{-1}\left((1+x^2+y^2+z^2)e^{\frac{x^2+y^2+z^2}{2}}\right)$$

The exact solution is

$$u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}}$$

This is also a fully nonlinear PDE. As we proved in Section 5.6.2, we can create a consistent monotone approximation of this operator by minimizing the directional derivatives over all orthogonal frames. Thus, we approximate the operator by

$$\min_{\nu_1,\nu_2,\nu_3 \text{ orthogonal}} \left(\tan^{-1}\left(u_{\nu_1\nu_1}\right) + \tan^{-1}\left(u_{\nu_2\nu_2}\right) + \tan^{-1}\left(u_{\nu_3\nu_3}\right)\right)$$

where the $u_{\nu_i\nu_i}$ are approximated by the monotone stencils we derived earlier. As with the other problems, we approximate the directional derivatives for a subset of orthogonal frames, then take a discrete minimum. The results are in Figure 5.13 and Table 5.7. We observe accuracy ranging from the expected $\mathcal{O}(\sqrt{h})$ to $\mathcal{O}(h^2)$.

Filtered Schemes Although filtered schemes have not yet been analyzed in this case, their implementation is a promising avenue for future work. We can already demonstrate some preliminary numerical results by solving the Lagrange Curvature problem with Dirichlet boundary conditions using a filtered scheme. The results from solving this equation using filtered schemes are shown in Figure 5.14 and Table 5.8 As suspected, we observe second order accuracy for this example.

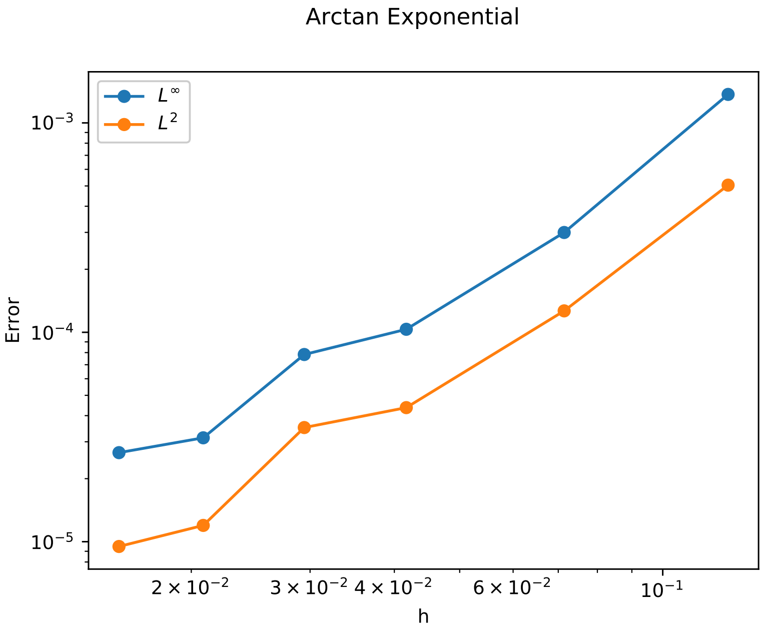

Figure 5.13 A convergence plot for the Lagrangian Curvature Equation.

Table 5.7 Lagrangian Curvature Error Table

h	L^∞ Error	Observed Order
0.125	0.00136792	
0.0714286	0.000299701	2.71305
0.0416667	0.00010337	1.97491
0.0294118	7.83304e-05	0.796364
0.0208333	3.13434e-05	2.6561
0.015625	2.66247e-05	0.567172

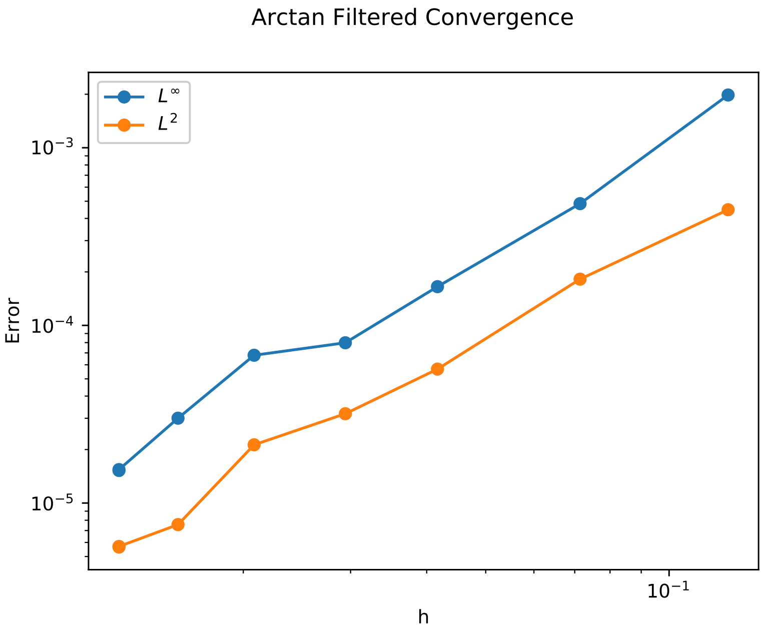

Figure 5.14 A convergence plot for the Lagrangian Curvature Equation with Dirichlet boundary conditions using filtered schemes.

Table 5.8 Lagrangian Curvature Filtered Error Table

h	L^∞ Error	Observed Order
0.125	0.00198083	
0.0714286	0.000484545	2.51612
0.0416667	0.000165188	1.99654
0.0294118	7.98305e-05	2.08775
0.0208333	6.78961e-05	0.469571
0.015625	3.00262e-05	2.83615
0.0125	1.52504e-05	3.03602

5.13.7 Neumann Boundary Conditions

Next, we consider a variation of the Lagrangian Curvature problem from “A Boundary Value Problem for Minimal Lagrangian Graphs” [7] with Neumann boundary conditions. Specifically, we solve

$$\begin{cases} \sum_{j=1}^{3} \tan^{-1}\left(\lambda_j(D^2u)\right) = c\, f(x,y,z) & x^2+y^2+z^2<1 \\ \frac{\partial u(x,y,z)}{\partial \hat{n}} = e^{\frac{1}{2}} & x^2+y^2+z^2=1 \end{cases} \tag{5.48}$$

where

$$f(x,y,z) = 2\tan^{-1}\left(e^{\frac{x^2+y^2+z^2}{2}}\right) + \tan^{-1}\left((1+x^2+y^2+z^2)e^{\frac{x^2+y^2+z^2}{2}}\right)$$

The exact solution is

$$u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}}$$

As with the other problems, we approximate the second directional derivatives for a subset of orthogonal frames, then take a discrete minimum. We also approximate the normal derivatives using the scheme derived in Section 5.8. The results are in Figure 5.15 and Table 5.9. We observe close to the expected $\mathcal{O}(\sqrt{h})$ accuracy on this example.

In addition, we consider Poisson’s equation with Neumann boundary conditions. Specifically, we solve the eigenvalue formulation

$$\begin{cases} \lambda_1(D^2u)+\lambda_2(D^2u)+\lambda_3(D^2u) = cf(x,y,z) & x^2+y^2+z^2<1 \\ \frac{\partial u(x,y,z)}{\partial \hat{n}} = e^{\frac{1}{2}} & x^2+y^2+z^2=1 \end{cases} \tag{5.49}$$

where

$$f(x,y,z) = (3+x^2+y^2+z^2)e^{\frac{x^2+y^2+z^2}{2}}$$

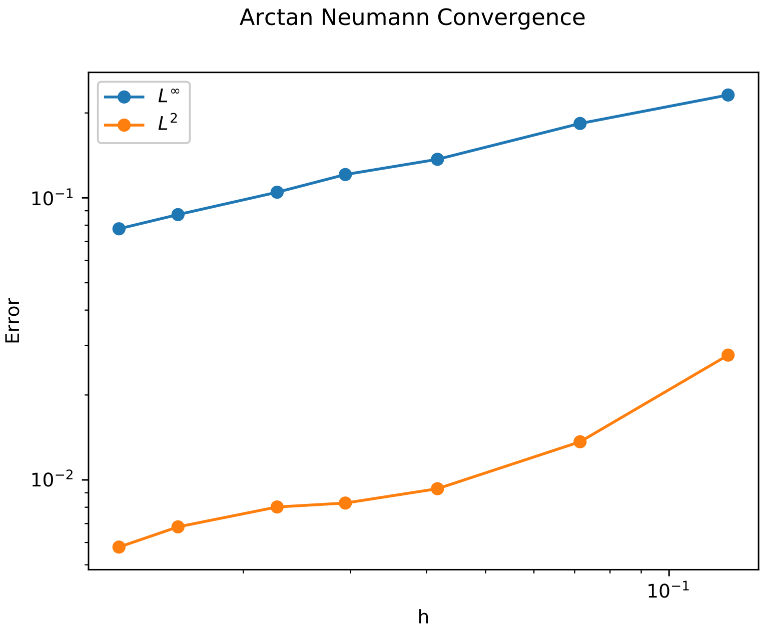

Figure 5.15 A convergence plot for the Lagrangian Curvature Equation with Neumann boundary conditions.

Table 5.9 Lagrangian Curvature with Neumann Boundary Conditions Error Table

h	L^∞ Error	Observed Order
0.125	0.231545	
0.0714286	0.183496	0.41561
0.0416667	0.136972	0.542525
0.0294118	0.120954	0.357059
0.0227273	0.104701	0.559656
0.015625	0.0872153	0.487688
0.0125	0.0776716	0.519349

The exact solution is

$$u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}}$$

For the interior, we approximate the second directional derivatives for a subset of orthogonal frames, then take a discrete minimum. For the boundary, we compute the normal derivative using the exact outward normal vector and we also approximate the normal derivatives using the scheme derived in Section 5.8. The results are in Figure 5.16 and Table 5.10. On average we observe the expected $\mathcal{O}(\sqrt{h})$ accuracy on this example.

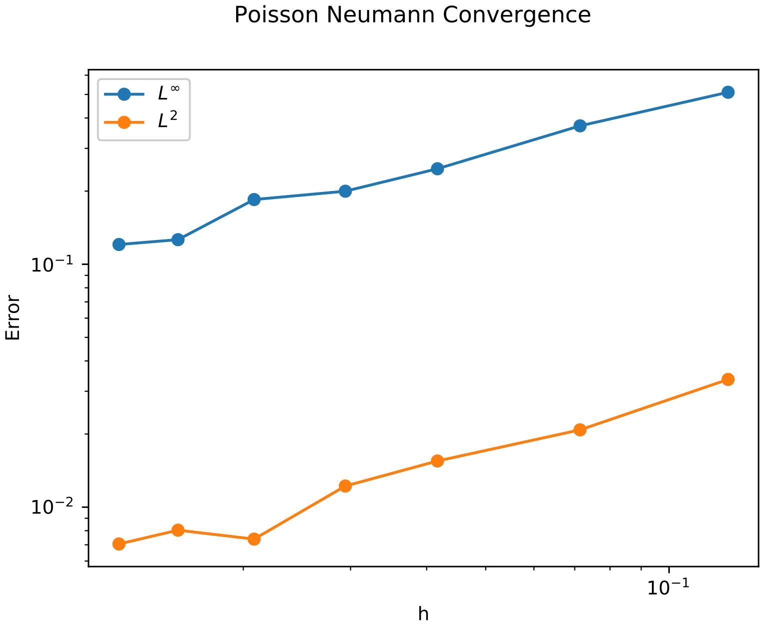

Figure 5.16 A convergence plot for Poisson's Equation with Neumann boundary conditions.

5.13.8 Transport Boundary Conditions

Next, we consider a variation of the Lagrangian Curvature problem from "A Boundary Value Problem for Minimal Lagrangian Graphs" [7] with Transport-type boundary

Table 5.10 Poisson's Equation Neumann Boundary Conditions Error Table

h	L^∞ Error	Observed Order
0.125	0.510503	
0.0714286	0.371374	0.56858
0.0416667	0.247244	0.7548
0.0294118	0.199854	0.610916
0.0208333	0.184646	0.229521
0.015625	0.126245	1.32166
0.0125	0.120595	0.205195

conditions. Specifically, we solve

$$\begin{cases} \tan^{-1}\left(\lambda_1(D^2u)\right) + \tan^{-1}\left(\lambda_2(D^2u)\right) + \tan^{-1}\left(\lambda_3(D^2u)\right) = c\,\frac{3\pi}{4} & x^2+y^2+z^2 < 1 \\ \nabla u(\mathbb{S}^2) \subset T(\mathbb{S}^2). \end{cases} \tag{5.50}$$

where $T(x, y, z) = (x + 2, y + 1, z - 1)$ is an affine shift. The exact solution is

$$u(x, y, z) = \frac{(x+2)^2 + (y+1)^2 + (z-1)^2}{2}$$

In order to discretize the function, we introduce the modification in Equation (5.31) which agrees with (5.50) on the space of convex functions (which is where the desired solution lives). It also fits into our framework since it can be expressed as in Equation (5.32). We approximate the boundary condition using the same scheme as in two dimensions as stated in Section 5.9. The results are in Figure 5.17 and Table 5.11. We observe slightly better than the expected $\mathcal{O}(\sqrt{h})$ accuracy overall.

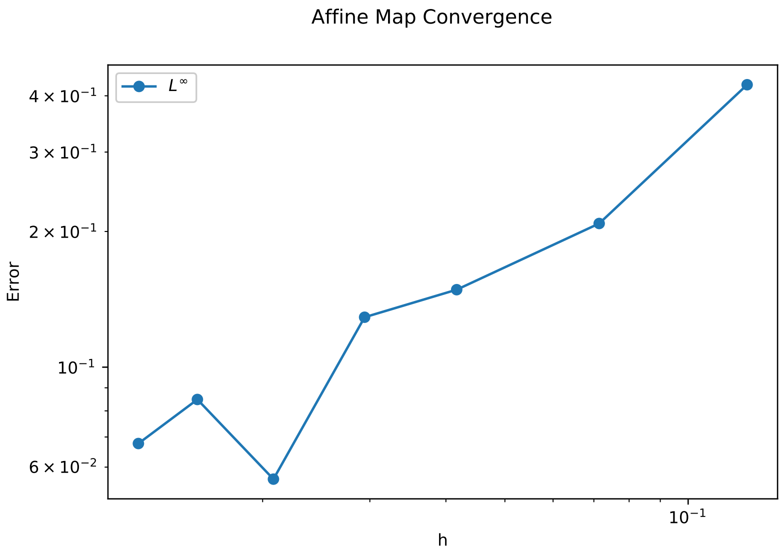

Figure 5.17 A convergence plot for the Lagrangian Curvature Equation with transport boundary conditions.

Table 5.11 Lagrangian Curvature with Transport Boundary Conditions Error Table

h	L^∞ Error	Observed Order
0.125	0.423231	
0.0714286	0.208314	1.26671
0.0416667	0.148553	0.627286
0.0294118	0.129053	0.404012
0.0208333	0.0564395	2.39837
0.015625	0.0847806	-1.4144
0.0125	0.0676967	1.00845

CHAPTER 6

CONCLUSIONS

6.1 Summary

This dissertation focused primarily on efficiently building and analyzing approximations of fully nonlinear elliptic operators in two- and three-dimensions. Our particular focus was on eigenvalue problems involving fully nonlinear PDEs.

In general dimensions, we produced a new framework for numerically solving eigenvalue problems involving fully nonlinear PDEs. The proof was demonstrated on a model problem: the construction of minimal Lagrangian graphs. Convergence to both the correct solution and the correct eigenvalue allows the solution of problems where the solvability condition is not satisfied exactly at the discrete level. In addition, this analysis provides a framework for the design of other convergent methods for eigenvalue problems involving fully nonlinear elliptic PDEs.

In two dimensions, techniques for solving eigenvalue problems for fully nonlinear elliptic PDEs were presented. In particular, we proposed a framework for discretizing and solving fully nonlinear eigenvalue problems with nonlinear transport-type boundary conditions. We also provided numerical results using the Lagrangian curvature problem as a test problem. The implementation allows for the solution of this problem on complicated geometries by adapting the quadtree methods used for simpler PDEs and Dirichlet boundary conditions. Additionally, higher order schemes were derived for functions of the gradient. Finally, we derived new monotone stencils for extending the schemes used in two dimensions to three dimensions. The new schemes can also be easily extended to n dimensions.

In three dimensions, we implemented a new generalized finite difference method for solving a large range of fully nonlinear elliptic equations. One challenge was in devising and implementing a framework for approximation of the eigenvalues of the Hessian in higher dimensions, where one cannot use the maximum and minimum of the second directional derivatives anymore. In addition, it was challenging to implement these schemes in three dimensions using limited memory and computational resources. We overcame these challenges in order to extend some of the two-dimensional work to three dimensions. We have built monotone schemes for solving problems involving eigenvalues of the Hessian in dimensions three and higher. In addition, we have extended the eigenvalue problem framework from our two-dimensional work in order to satisfy compatibility conditions at the discrete level in three or higher dimensions. This allows us to solve the problems with nonlinear transport-type boundary conditions in dimensions three and higher. The techniques derived for three dimensions can be readily applied to higher dimensions and to complicated domains.

6.2 Future Work

There are multiple problems to deal with in order to numerically solve fully nonlinear elliptic PDEs in three dimensions. One problem is the criteria for convergence, which we studied in this research. The next difficult problem was to efficiently construct a monotone approximation to the fully nonlinear operator, which was another focus of this dissertation. However, although the solution of these discrete systems were found in this work, this was not done very efficiently. One avenue of future work is the completion of the analysis and implementation of higher-order methods. Filtered schemes can be extended for higher dimensional operators and eigenvalue problems. Further analysis is needed to show the convergence of these higher-order methods when used with eigenvalue problems and in higher dimensions.

Equipped with the monotone approximations from this work, the next step is to build efficient solvers. This problem is nontrivial because with three dimensions, there are a large number of points, and the Jacobian matrix required for Newton's method is often infeasible computationally and from a memory perspective. Without directly building the system of equations, information must propagate some other way. One way to efficiently propagate this information is by using multigrid methods.

In addition, this eigenvalue framework can be extended to other problems. Of particular interest are general optimal transport problems where the data does not naturally satisfy the solvability condition at the discrete level. This framework holds great promise for these applications.

After efficient solvers are built, many other three-dimensional problems in industry can be solved. For example, optimal transportation problems in geophysics can solved efficiently using the techniques in this dissertation paired with new efficient solvers. Additionally, the efficient solution of three-dimensional optimal transportation problems can also be used for image registration. This is the problem of aligning two data sets with each other, which is relevant for brain MRIs. Since these deal with information about proton density it is natural to use optimal transport techniques to solve them. With efficient three-dimensional solvers, optimal transport can be used to create a better three-dimensional image of the brain. The specifics may depend on the imaging techniques used, but the ability to efficiently solve three-dimensional Monge-Ampére problems is a good starting point for these applications.

REFERENCES

[1] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asym. Anal.*, 4(3):271–283, 1991.

[2] P. W. Bates, G.-W. Wei, and S. Zhao. Minimal molecular surfaces and their applications. *J. Comp. Chem.*, 29(3):380–391, 2008.

[3] J.-D. Benamou, B. D. Froese, and A. M. Oberman. Numerical solution of the optimal transportation problem using the monge-ampère equation. *J. Comput. Phys.*, 260:107–126, March 2014.

[4] J.-D. Benamou, A. Oberman, and B. Froese. Numerical solution of the second boundary value problem for the Elliptic Monge-Ampère equation. Inria research report, June 2012.

[5] K. Böhmer. On finite element methods for fully nonlinear elliptic equations of second order. *SIAM J. Numer. Anal.*, 46(3):1212–1249, 2008.

[6] O. Bokanowski, S. Maroso, and H. Zidani. Some convergence results for Howard's algorithm. *SIAM J. Numer. Anal.*, 47(4):3001–3026, 2009.

[7] S. Brendle and M. Warren. A boundary value problem for minimal lagrangian graphs. *J. Diff. Geom.*, 84(2):267–287, 02 2010.

[8] C. Budd and J. Williams. Moving mesh generation using the parabolic mongeampère equation. *SIAM J. Sci. Comput.*, 31(5):3438–3465, 2009.

[9] A. Caboussat, R. Glowinski, and D. Sorensen. A least-squares method for the numerical solution of the dirichlet problem for the elliptic monge-ampère equation in dimension two. *ESAIM: Contr. Opt. Calc. Var.*, 19, 07 2013.

[10] L.A. Caffarelli and M. Milman. *Monge Ampère Equation: Applications to Geometry and Optimization : NSF-CBMS Conference on the Monge Ampère Equation, Applications to Geometry and Optimization, July 9-13, 1997, Florida Atlantic University.* Cont. Math. - Amer. Math. Soc. Bull. Amer. Math. Soc., 1999.

[11] M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc. (N.S.)*, 27(1):1–67, 1992.

[12] M. Cullen, J. Norbury, and R. Purser. Generalised lagrangian solutions for atmospheric and oceanic flows. *SIAM J. Appl. Math.*, 51(1):20–31, 1991.

[13] E. Dean and R. Glowinski. Numerical solution of the two-dimensional elliptic monge-ampère equation with dirichlet boundary conditions: A least-squares approach. *C. R. Math.*, 339:887–892, 12 2004.

[14] E. J. Dean and R. Glowinski. Numerical solution of the two-dimensional elliptic monge-ampére equation with dirichlet boundary conditions: An augmented lagrangian approach. *C. R. Math. Acad. Sci. Paris*, 336:779784, 2003.

[15] E. J. Dean and R. Glowinski. An augmented lagrangian approach to the numerical solution of the dirichlet problem for the elliptic monge-ampére equation in two dimensions. *Electron. Trans. Numer. Anal.*, 22:7196, 2006.

[16] P. Delanoë. Classical solvability in dimension two of the second boundary-value problem associated with the monge-ampère operator. *Ann. Inst. H. P. Nonlin. Anal.*, 8(5):443–457, September 1991.

[17] B. Engquist and B. D. Froese. Application of the Wasserstein metric to seismic signals. *Comm. Math. Sci.*, 12(5):979–988, 2014.

[18] X. Feng, R. Glowinski, and M. Neilan. Recent developments in numerical methods for fully nonlinear second order partial differential equations. *SIAM Rev.*, 55(2):205–267, 2013.

[19] X. Feng, C.-Y Kao, and T. Lewis. Convergent finite difference methods for one-dimensional fully nonlinear second order partial differential equations. *J. Comput. Appl. Math.*, 254, 12 2012.

[20] X. Feng and T. Lewis. A Narrow-stencil finite difference method for approximating viscosity solutions of fully nonlinear elliptic partial differential equations with applications to Hamilton-Jacobi-Bellman equations. *arXiv e-prints*, July 2019.

[21] X. Feng and M. Neilan. Vanishing moment method and moment solutions for fully nonlinear second order partial differential equations. *SIAM J. Sci. Comput.*, 38(1):74–98, 2009.

[22] W.H. Fleming and H.M. Soner. *Controlled Markov Processes and Viscosity Solutions.* Sto. Model. Appl. Prob. New York, NY: Springer, 2006.

[23] U. Frisch, S. Matarrese, R. Mohayaee, and A. Sobolevski. A reconstruction of the initial conditions of the Universe by optimal mass transportation. *Nat.*, 417:260–262, May 2002.

[24] B. Froese and A. Oberman. Convergent filtered schemes for the Monge-Ampère partial differential equation. *SIAM J. Numer. Anal.*, 51(1):423–444, 2013.

[25] B. D. Froese. A numerical method for the elliptic Monge-Ampère equation with transport boundary conditions. *SIAM J. Sci. Comput.*, 34(3):A1432–A1459, 2012.

[26] B. D. Froese. Meshfree finite difference approximations for functions of the eigenvalues of the Hessian. *Numer. Math.*, 138(1):75–99, 2018.

[27] B. D. Froese and A. M. Oberman. Convergent finite difference solvers for viscosity solutions of the elliptic Monge-Ampère equation in dimensions two and higher. *SIAM J. Numer. Anal.*, 49(4):1692–1714, 2011.

[28] T. Glimm and V. Oliker. Optical design of single reflector systems and the monge–kantorovich mass transfer problem. *J. Math. Sci.*, 117(3):4096–4108, September 2003.

[29] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent. Optimal mass transport for registration and warping. *Int. J. Comp. Vis.*, 60(3):225–240, 2004.

[30] B. Hamfeldt. Convergent approximation of non-continuous surfaces of prescribed Gaussian curvature. *Comm. Pure Appl. Anal.*, 17(2):671–707, 2018.

[31] B. Hamfeldt. Convergence framework for the second boundary value problem for the monge–ampère equation. *SIAM J. Numer. Anal.*, 57(2):945–971, 2019.

[32] B. F. Hamfeldt. Viscosity subsolutions of the second boundary value problem for the Monge-Ampére equation. *arXiv*, July 2018.

[33] B. F. Hamfeldt and T. Salvador. Higher-order adaptive finite difference methods for fully nonlinear elliptic equations. *SIAM J. Sci. Comput.*, 75(3):1282–1306, June 2018.

[34] B. F. Hamfeldt and A. G. R. Turnquist. A convergence framework for optimal transport on the sphere. *arXiv e-prints*, 2021.

[35] M. Jensen and I. Smears. On the convergence of finite element methods for Hamilton–Jacobi–Bellman equations. *SIAM J. Numer. Anal.*, 51(1):137–162, 2013.

[36] R. Jensen. The maximum principle for viscosity solutions of fully nonlinear second order partial differential equations. *Arch. Rat. Mech. Anal.*, 101(1):127, 1988.

[37] S. Koike. *A Beginner's Guide to the Theory of Viscosity Solutions.* MSJ memoirs. Tokyo, JP: Mathematical Society of Japan, 2004.

[38] R. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (Classics in Applied Mathematics).* Philadelphia, PA: SIAM, 2007.

[39] Y. Lian and K. Zhang. Boundary lipschitz regularity and the Hopf lemma for fully nonlinear elliptic equations. *arXiv e-prints*, December 2018.

[40] J.-M. Mirebeau. Discretization of the 3d Monge-Ampére operator, between wide stencils and power diagrams. *ESAIM: Math. Model. Numer. Anal.*, 49(5):1511–1523, 2015.

[41] T. S. Motzkin and W. Wasow. On the approximation of linear elliptic differential equations by difference equations with positive coefficients. *J. Math. and Phys.*, 31(1-4):253–259, 1952.

[42] R. Nochetto, D. Ntogkas, and W. Zhang. Two-scale method for the Monge-Ampère equation: Convergence to the viscosity solution. *Math. Comput.*, 88, 06 2017.

[43] A. Oberman. The convex envelope is the solution of a nonlinear obstacle problem. *Proc. Amer. Math. Soc.*, 135(6):1689–1694, 2007.

[44] A. M. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–Jacobi equations and free boundary problems. *SIAM J. Numer. Anal.*, 44(2):879–895, 2006.

[45] A. M. Oberman. Wide stencil finite difference schemes for the elliptic Monge-Ampère equation and functions of the eigenvalues of the Hessian. *Disc. Cont. Dynam. Syst. Ser. B*, 10(1):221–238, 2008.

[46] V. I. Oliker and L. D. Prussner. On the numerical solution of the equation $(\frac{\partial^2 z}{\partial x^2})(\frac{\partial^2 z}{\partial y^2})-(\frac{\partial^2 z}{\partial x \partial y})^2 = f$ and its discretizations. *I, Numer. Math.*, 54:271–293, 1988.

[47] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics.* New York, NY: Springer, 2003.

[48] K. Smoczyk and M.-T. Wang. Mean curvature flows of Lagrangian submanifolds with convex potentials. *J. Diff. Geom.*, 62(2):243–257, 2002.

[49] E. L. Thomas, D. M. Anderson, C. S. Henkee, and D. Hoffman. Periodic area-minimizing surfaces in block copolymers. *Nat.*, 334(6183):598, 1988.

[50] R. P. Thomas and S.-T. Yau. Special Lagrangians, stable bundles and mean curvature flow. *Comm. Anal. Geom.*, 10(5):1075–1113, 2002.

[51] J. Urbas. On the second boundary value problem for equations of Monge-Ampére type. *J.R.A.M.*, 487:115–124, 1997.

[52] D. Williamson. *Lecture notes in Mathematical Programming I.* Lecture Notes, Cornell University, September 2008.