NUMERICAL METHODS FOR THE ELLIPTIC MONGE-AMPÈRE EQUATION AND OPTIMAL TRANSPORT

by

Brittany Dawn Froese

M.Sc., Simon Fraser University, 2009

B.Sc., Trinity Western University, 2007

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2012

APPROVAL

Name: Brittany Dawn Froese

Degree: Doctor of Philosophy

Title of Thesis: Numerical Methods for the Elliptic Monge-Ampère Equation and Optimal Transport

Examining Committee: Dr. JF Williams, Associate Professor
Chair

Dr. Adam Oberman, Associate Professor
Senior Supervisor

Dr. Steven Ruuth, Professor
Supervisor

Dr. Nilima Nigam, Associate Professor
Internal Examiner

Dr. Panagiotis Souganidis, Professor
Department of Mathematics
The University of Chicago
External Examiner

Date Defended: June 8, 2012

Abstract

The problem of optimal transport, which involves finding the most cost-efficient way of transporting mass from one location to another, has been widely-studied, going back to the late eighteenth century. Recent years have revealed numerous applications in areas such as medical imaging, meteorology, cosmology, oceanography, and economics. Despite the importance of optimal transport, the computation of solutions remains extremely challenging. In the simplest case, where the cost function is quadratic, the problem takes on additional structure. In this setting, the constraint that mass must be conserved can be expressed as a fully non-linear partial differential equation known as the elliptic Monge-Ampère equation.

The numerical solution of the Monge-Ampère equation has received a great deal of attention in recent years, yet the correct and efficient computation of solutions remains a challenge. Because of the nonlinearity of the equation, solutions can be singular and standard numerical approaches can fail. This means that novel solution techniques are needed to correctly capture the behaviour of weak solutions. We describe a monotone finite difference discretisation, which provably converges to the viscosity solution of the Monge-Ampère equation. The accuracy of the discretisation is improved by combining higher-order schemes with the monotone scheme needed to capture the correct behaviour of solutions near singularities. In doing this, we provide a general result about the convergence of higher-order finite difference methods for elliptic equations. The resulting nonlinear equations are solved efficiently using Newton's method.

To ensure that mass is mapped into the desired region, the Monge-Ampère equation must be coupled to a transport boundary condition. This type of boundary condition is non-standard, and previously has been implemented only in very simple cases (such as transporting a square to a square). We propose a new method for implementing the transport condition by solving a sequence of more tractable Monge-Ampère equations with Neumann

boundary conditions. To demonstrate the effectiveness and efficiency of the resulting methods, we provide computational results for a number of challenging problems including the recovery of inverse maps, mapping onto unbounded density functions, mapping from a disconnected domain, and mapping onto non-convex sets.

Keywords: Monge-Ampère; optimal transport; partial differential equations; viscosity solutions; boundary conditions; finite difference methods

Soli Deo Gloria!

Acknowledgments

Writing a thesis is not a solo activity, and there are many people who deserve credit for this work.

First of all, I am deeply grateful to my supervisor, Dr. Adam Oberman. During the past five years, he has been the source of a great deal of instruction, encouragement, and enthusiasm. I have also benefited from my conversations with Dr. JD Benamou, who has always been generous with his time and expertise. I offer my thanks to the many other faculty and graduate students who have taught me, challenged me, and shared their passion with me during my time at SFU. In particular, I wish to thank Dr. JF Williams, Dr. Steve Ruuth, Dr. Nilima Nigam, and Dr. Panagiotis Souganidis for participating in my thesis defence.

The work described in this thesis was supported by a Pacific Century Graduate Scholarship, as well as a Canadian Graduate Scholarship from the National Science and Engineering Research Council. I am grateful for this financial support, which has enabled me to devote so much of my time to research.

It would be impossible to overstate how thankful I am for my parents and sister Joelle. My dad's early instruction that "math is happiness" has carried me a long way. All three of them have provided me with much love, encouragement, and comic relief to help sustain me through the ups and downs of graduate school.

Finally, I simply cannot find the words to express the depth of my gratitude to my God and Saviour Jesus Christ. If there is anything good that I have accomplished, it is only due to His grace.

> "For from him and through him and to him are all things. To him be the glory forever! Amen."
>
> Romans 11:36, NIV

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The elliptic Monge-Ampère equation is a fully nonlinear Partial Differential Equation (PDE) first described in the late eighteenth century. Since then, the equation has arisen in a number of important applications and the associated regularity theory has received a great deal of attention. Despite the importance of the Monge-Ampère equation, until recently, very little progress had been made in actually solving the equation numerically.

The last several years have seen an explosion of interest in numerical methods for solving this and other fully nonlinear PDEs. For example, this topic was the focus of an invited lecture at the 2007 International Congress on Industrial and Applied Mathematics (ICIAM) [47]. Several methods have been developed for approximating solutions of the Monge-Ampère equation. However, the richness and complexity of the equation also lead to a number of important challenges that place limitations on these numerical methods. Moreover, the type of boundary conditions that can be enforced using currently available methods are typically quite different from the boundary conditions that arise naturally in applications. Consequently, the development of numerical methods for this PDE remains a challenging problem. The development of methods powerful enough to handle these challenges would have important implications for several interesting applications.

The goal of this thesis is to construct efficient and robust numerical methods for the Monge-Ampère equation by bringing this PDE into the framework of modern finite difference techniques and convergence theory. We are also interested in using the Monge-Ampère equation to numerically compute solutions to the optimal mass transport problem. With this in mind, we develop a novel method for implementing the unique transport boundary condition that occurs naturally in many applications.

1.1 The Monge-Ampère Equation

The Monge-Ampère operator is given by

$$\det(D^2u(x))$$

where D^2u is the Hessian of the function u.

We consider the equation in a convex bounded subset $X \subset \mathbb{R}^d$ with boundary ∂X. The general form of a Monge-Ampère type equation is

$$\det(D^2u(x)) = F(x, u(x), \nabla u(x)), \quad \text{in } \Omega.$$

In order for the equation to be elliptic, which is important both for uniqueness and to ensure that solutions have a meaningful physical interpretation, we must also impose the convexity constraint

$$u \text{ is convex.} \tag{1.1}$$

In the simplest case $F(x, u(x), \nabla u(x)) \equiv f(x)$ is a continuous function, $f \in C(X)$, and f is bounded away from zero, $f(x) \geq \mu > 0$. These conditions, combined with suitable conditions on the domain, ensure uniform ellipticity of the PDE and improve the regularity of solutions. In the most general case, the right-hand side can be a measure [48]. We consider the case

$$F(x, \nabla u(x)) \geq 0,$$

which permits singular solutions.

Much of the work on numerics for the Monge-Ampère equation has concentrated on the simple Monge-Ampère equation of the form

$$\det(D^2u(x)) = f(x). \tag{1.2}$$

To keep the key ideas of this thesis clear, we will begin by considering only this simple form. However, later in the thesis, we will also consider the numerical solution of the more general equation

$$\det(D^2u(x)) = F(x, \nabla u(x)). \tag{1.3}$$

The dependence of the right-hand side on gradients introduces additional challenges in correctly approximating this equation.

1.2 Applications

Part of the beauty of the Monge-Ampère equation lies in its relationship to so many different applications.

The most direct application, and the one considered by Monge and Ampère, is the problem of optimal mass transport [2, 21, 33, 88]. The problem here is to find a mapping $s(x)$ that transports the source density $f(x)$ to the target density $g(y)$ and minimises the cost functional

$$\int_{\mathbb{R}^d} c(x, s(x)) f(x)\, dx$$

where $c : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty)$ is some cost function. When the cost is quadratic, the optimal mapping is simply the gradient of a convex function that satisfies the Monge-Ampère equation

$$\det(D^2 u(x)) = f(x)/g(\nabla u(x)).$$

One recent application of the Monge-Ampère equation and optimal mass transport is in the generation of equidistributing meshes [13, 82]. This enables other equations to be solved on meshes that concentrate grid points in regions of high activity, which allows sharp fronts to be more accurately and inexpensively resolved.

The Monge-Ampère operator can also appear in inequality constraints in other variational problems for optimal mappings, where the cost may not be the usual transportation cost. For example, mapping problems arising in areas such as image registration [49, 50, 51, 85] and computer graphics [24, 53, 63, 64] involve the minimisation of some metric on

$$dist(X, s(X))$$

subject to the constraint that $s(x)$ is a mapping between the sets X and Y. Here $dist$ is some metric between images; an example is the disparity of grayscale levels. For a combination of modeling and mathematical reasons, it is often natural to restrict to diffeomorphisms with prescribed or bounded Jacobian

$$\lambda \leq \det(\nabla s) \leq \Lambda.$$

In the case where the mapping is cyclically monotone [7, 78], this mapping $s(x)$ is the gradient of a convex potential function $u(x)$ and we recover an inequality involving the Monge-Ampère operator,

$$\lambda \leq \det(D^2 u) \leq \Lambda.$$

Another natural application of Monge-Ampère equations is in geometric problems involving the construction of surfaces with prescribed metrics or curvatures, as well as the associated existence and uniqueness results. For example, the equation describing a convex surface $(x, u(x))$ in $\mathbb{R}^{d+1}$ with prescribed Gaussian curvature $\kappa(x)$ is of Monge-Ampère type [88]:

$$\det(D^2u(x)) = \kappa(x)(1 + \left|\nabla u(x)\right|^2)^{\frac{d+2}{2}}. \tag{1.4}$$

Other recent applications include dynamic meteorology [52, 56], oceanography [27], astrophysics [37], elasticity [80], economics and traffic flow [79], and geometric optics [44, 45, 89, 90, 91].

1.3 Boundary Conditions

Perhaps the simplest setting for a Monge-Ampère equation is in a periodic domain $\mathbb{T}^d$. In this case, the PDE is expressed in the form:

$$\begin{cases} \det(I + D^2u(x)) = f(x) \quad x \in \mathbb{T}^d \\ \frac{|x|^2}{2} + u \text{ is convex.} \end{cases}$$

Here the function u must be periodic; aside from this constraint, there are no boundary effects to worry about.

In the non-periodic setting, some type of boundary condition is necessary to ensure that the equation is well-posed. In addition to introducing extra constraints that must be satisfied, boundary effects can cause solution regularity to break down, which can in turn result in the poor performance of failure of many numerical methods.

The simplest boundary condition, and the one considered in the first part of this thesis, is the Dirichlet boundary condition

$$u(x) = \phi(x), \quad x \in \partial\Omega. \tag{1.5}$$

Alternatively, we could choose to specify normal derivatives at the boundary using a Neumann boundary condition

$$\nabla u(x) \cdot \mathbf{n}(x) = \phi(x), \quad x \in \partial\Omega \tag{1.6}$$

where $\mathbf{n}(x)$ denotes the unit outward normal to the boundary ∂X. In very simple mapping problems, this boundary condition is natural.

In many applications, the gradient of the solution defines a map from its domain to its range. Often, the range of the map is pre-specified, which leads to the transport boundary condition

$$\nabla u : X \to Y. \tag{1.7}$$

Although this condition is very natural in mapping applications, very little progress has been made on its numerical implementation.

1.4 Related Work

In the last several years, a number of methods have been proposed for the numerical solution of the Monge-Ampère equation. We make a distinction between methods for the PDE itself and methods that address the transport boundary condition, which has received much less attention from a numerical standpoint. We also distinguish between methods that provably converge to a weak solution of the equation, and other methods that provide no guarantee of convergence.

An early work by Oliker and Prussner [76] introduced a discretisation based on a geometric interpretation of the solutions [3]. In two dimensions, this method converges to the Aleksandrov solution.

Another convergent method for the equation was presented by Oberman [74]; this involves a wide-stencil finite difference discretisation of the two-dimensional Monge-Ampère equation. While this method is proven to converge to the viscosity solution of (1.2), the scheme introduces an additional discretisation error related to the stencil width. The associated CFL condition also limits the speed of this method.

Dean and Glowinski et al. [28, 29, 30, 31, 46] have investigated Lagrangian and least squares methods for the numerical solution of the Monge-Ampère equation. These methods perform well when the solutions are in H^2. However, the authors point out that for solutions with less regularity the methods may fail. In [29] they give an example of a solution that is not in H^2, for which their method diverges.

Feng and Neilan [34, 35] solve second order equations (including the Monge-Ampère equation) by adding a small multiple of the bilaplacian. The bilaplacian term introduces an additional discretisation error and additional boundary conditions, which may not be compatible with the weak solution of the equation; this introduces a boundary layer into the computed solution.

Sulman et. al. [83] and Budd and Williams [12] have solved the Monge-Ampère equation by seeking the steady-state solution of different parabolic forms of the equation. If the data is smooth enough, the (continuous) parabolic equation will converge to the solution of the elliptic Monge-Ampère equation. Convergence of the discretised problem is not addressed.

Böhmer has studied the consistency and stability of certain finite element approximations to fully nonlinear elliptic equations [8]. Stable finite element approximations of the Monge-Ampère equation in two- and three-dimensions have been also been constructed by Brenner et. al. [10, 11]. These finite element approximations all require solutions to be smoother than H^2.

In the periodic setting, Loeper and Rapetti [65] solve the equation using Newton's method. They prove convergence of the Newton algorithm for the continuous problem (though not the discretised problem) to the solution of (1.2). However, they restrict themselves to the case where the source term f is strictly positive. The work of Frisch et. al. [92] studies the case of periodic boundary conditions in an odd dimensional space. Several different formulations of the equation appear, including a Fourier integral form.

Much less work has been done on the full L^2 optimal transport problem. An early work by Knott and Smith used techniques from complex analysis to construct optimal maps between uniform densities in two dimensions [59]. Another approach to the optimal transport problem involves re-framing it as a fluid flow problem. This approach was introduced by Benamou and Brenier [5] and has been further developed by Haber et. al. [49]. However, it is computationally expensive as it requires introducing an additional dimension to the problem. We also mention the work of Delzanno et. al. [32, 36], which involves discretising (1.2) using the natural finite difference discretisation and solving the resulting system using an inexact Newton-Krylov method. Under some assumptions about the boundary conditions, they are able to map from a rectangular domain into a region with four (possibly curved) sides.

1.5 Outline of This Thesis

In this thesis, we are concerned with building finite difference methods for solving Monge-Ampère type equations and the related L^2 optimal transport problem. The main contributions of this thesis are also presented in [6, 39, 40, 41, 42].

We begin by providing important background information about the theory of the

Monge-Ampère equation. This material, presented in Chapter 2, serves to motivate and guide the methods constructed in the remainder of the thesis. In view of our interest in developing fast solution methods, we also describe Newton's method for solving the PDE and provide a convergence proof in the continuous setting.

In Chapter 3, we investigate the use of standard finite difference discretisations of the Monge-Ampère equation. We describe several different solution methods and discuss both their usefulness and their limitations.

In Chapter 4, we move on to the theory of convergent finite difference methods. We describe a new characterisation of the Monge-Ampère equation which, together with the general convergence theory, enables us to construct a new monotone finite difference method and prove that it convergences to the weak (viscosity) solution of the equation.

In Chapter 5, we construct two hybrid finite difference methods that combine the best features of the standard and convergent finite difference schemes. This allows us to improve solution accuracy without sacrificing correctness and stability near singularities. For one of these schemes, we prove convergence to the viscosity solution of the Monge-Ampère equation. To accomplish this, we also prove a very general result about the convergence of higher order finite difference methods for a class of degenerate elliptic PDEs.

In Chapter 6, we turn our attention to the problem of optimal transport. We propose a new method for implementing the transport boundary condition by solving a sequence of more tractable Monge-Ampère equations with Neumann boundary conditions. By solving these sub-problems using the finite difference methods developed in this thesis, we produce an efficient method for solving a wide range of challenging L^2 optimal transport problems.

In Chapter 7, we summarise the contributions of this thesis. We also suggest several possible directions for future research that extend naturally from this work.

Chapter 2

The Monge-Ampère Equation

The main purpose of this chapter is to present important background material on the Monge-Ampère equation. To motivate our study of optimal transport boundary conditions, we present a derivation of the Monge-Ampère equation in the setting of L^2 optimal mass transport.

Much of this chapter focuses on the theoretical background of the Monge-Ampère equation. In order to motivate and guide the methods developed in this thesis, we review several important properties of the equation and its solutions. We go on to describe the use of Newton's method to solve this equation. As part of this discussion, we provide a proof that Newton's method converges (for the Dirichlet problem) if the problem is sufficiently regular.

We conclude this chapter by describing several numerical challenges associated with the Monge-Ampère equation, which we intend to address in this thesis.

2.1 Optimal Transport

We begin by describing the problem of optimal mass transport and explaining how the Monge-Ampère equation arises in this context.

2.1.1 Monge-Kantorovich Mass Transport

The problem originally considered by Monge is how to transport a given pile of sand into a hole with minimum cost, where the original cost is simply the magnitude of the distance the sand is transported (Figure 2.1). That is, the problem is to find a mapping $s(x)$ from

the original set X to the target set Y that minimises the cost functional

$$I[s] = \int_X \left| x - s(x) \right| \, dx. \tag{2.1}$$

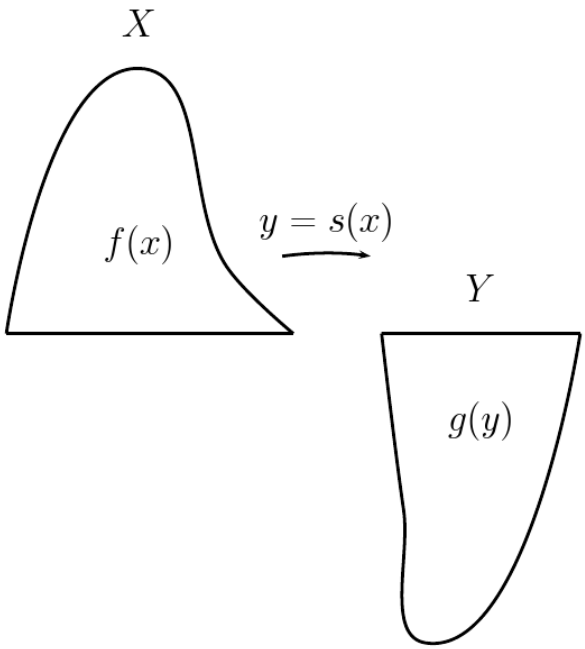


Figure 2.1: The mass transport problem.

The more general Monge-Kantorovich problem describes the transportation of mass densities using more general cost functions. That is, we want to find a mapping that takes the density $f(x)$ in the space X into the density $g(y)$ in space Y. We denote the set of such functions as the admissible set $\mathcal{A}$. We are also given a cost function $c(x, y)$, which gives the cost of transporting a unit of mass from location x to location y. The problem is then to find a mapping $s(x) \in \mathcal{A}$ that minimises the cost functional

$$I[s] = \int_X c(x, s(x)) f(x) \, dx. \tag{2.2}$$

Kantorovich contributed to the understanding of optimal transport by reformulating the problem as a linear program and describing a simple dual formulation [54, 55]. While this has made many theoretical questions easier to answer, this approach also effectively doubles the dimension of the problem. Consequently, computing the solution to even a small-scale problem is prohibitively expensive. This motivates the development of more sophisticated methods that will enable the efficient computation of optimal maps.

2.1.2 Conservation of Mass

It is useful to consider in more detail the requirement that the minimiser of the cost (2.2) must push the density $f(x)$ in X entirely onto the density $g(y)$ in Y. Since we require that mass be conserved, the following equality must hold for any continuous function $h(y)$:

$$\int_X h(s(x))f(x)\,dx = \int_Y h(y)g(y)\,dy.$$

By introducing the change of variables $y = s(x)$ into the right-hand side of this equation we obtain

$$\int_X h(s(x))f(x)\,dx = \int_X h(s(x))g(s(x))\det(\nabla s(x))\,dx.$$

Rearranged, this becomes

$$\int_X \big(f(x) - g(s(x))\det(\nabla s(x))\big)\,h(s(x))\,dx = 0.$$

Again, this holds for every continuous function $h(x)$. Consequently, we obtain the equation

$$\det(\nabla s(x)) = f(x)/g(s(x)).$$

2.1.3 Cyclical Monotonicity

The simplest and most widely studied cost function is the quadratic cost function

$$c(x,y) = \frac{1}{2}|x-y|^2\,.$$

With this cost, the Monge-Kantorovich problem becomes

$$\begin{aligned} \text{minimise} \qquad & \int_X \frac{1}{2}\big|x - s(x)\big|^2 f(x)\,dx \\ \text{subject to} \qquad & \det\big(\nabla s(x)\big) = f(x)/g(s(x)). \end{aligned} \tag{2.3}$$

It turns out that a solution of this problem must be cyclically monotone. Intuitively, this means that mass is not being "twisted." To see why, we assume that a minimiser $s(x)$ exists and choose any finite number $N \in \mathbb{N}$ of distinct points $x_k \in X$. Then we denote by E_k the ball of radius r_k centred at x_k. Here the r_k are chosen so that all of the balls are disjoint and contain the same total mass ϵ. That is, for every $1 \leq k \leq N$,

$$\int_{E_k} f(x)\,dx = \epsilon. \tag{2.4}$$

We also define the points and regions that the x_k, E_k are mapped onto by

$$y_k = s(x_k), \quad F_k = s(E_k).$$

We observe that the new regions F_k also contain mass ϵ since the mapping $s(x)$ conserves mass:

$$\begin{aligned}\int_{F_k} g(y)\,dy &= \int_{E_k} f(x)\,dx \\ &= \epsilon.\end{aligned}$$

We can now define a new mapping $s'(x)$ by cyclically permuting the images of E_k and leaving the remainder of the mapping $s(x)$ unchanged (Figure 2.2).

$$s'(x) = \begin{cases} s(x + x_{k+1} - x_k) & x \in E_k, 1 \leq k < N \\ s(x + x_1 - x_N) & x \in E_N \\ s(x) & x \in X \setminus \bigcup_{k=1}^{N} E_k. \end{cases}$$

By design, this new mapping will also push the density $f(x)$ entirely onto $g(y)$.

We recall that $s(x)$ is a minimiser of the cost functional in (2.3). This means that

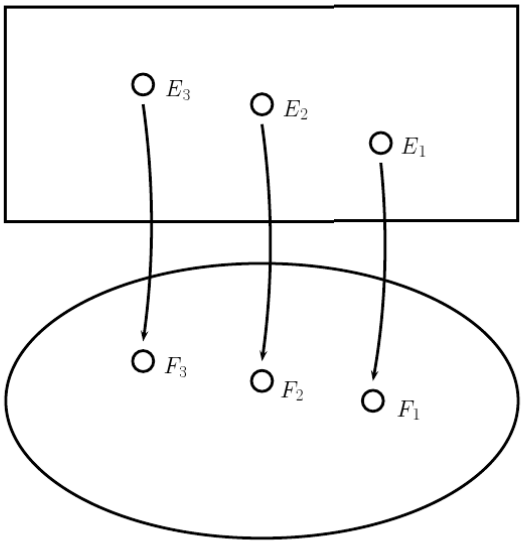
$$I[s] \leq I[s'].$$

Substituting in the quadratic cost we see that

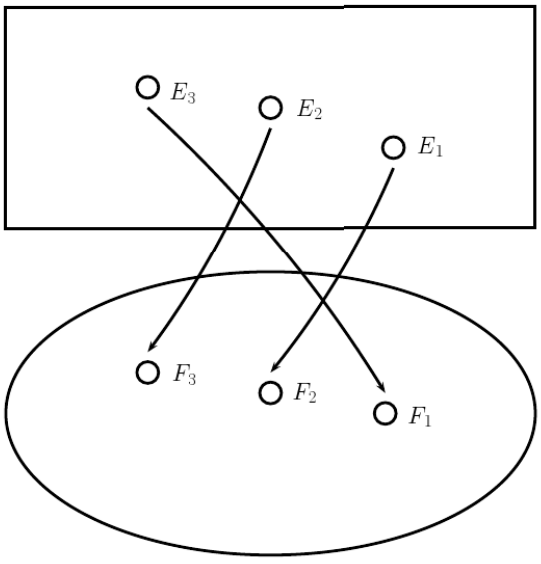
$$\int_X \left|x - s(x)\right|^2 f(x)\,dx \leq \int_X \left|x - s'(x)\right|^2 f(x)\,dx.$$

Expanding the quadratic term, we obtain

$$\int_X \left(\left|s(x)\right|^2 - 2x \cdot s(x)\right) f(x)\,dx \leq \int_X \left(\left|s'(x)\right|^2 - 2x \cdot s'(x)\right) f(x)\,dx.$$



(a)



(b)

Figure 2.2: (a) A mapping that minimises (2.3). (b) A cyclical permutation of the minimiser.

Since both $s(x)$ and $s'(x)$ push the density $f(x)$ onto $g(y)$, and since these two mappings are identical over much of the domain, this simplifies to

$$\sum_{k=1}^{N} \int_{E_k} x \cdot (s'(x) - s(x)) f(x)\, dx \leq 0.$$

Dividing both sides by ϵ (that is, replacing the integrals by averages over the balls E_k) we obtain

$$\sum_{k=1}^{N} \frac{1}{\epsilon} \int_{E_k} x \cdot (s'(x) - s(x)) f(x)\, dx \leq 0.$$

In the limit as $\epsilon \to 0$ this becomes

$$\sum_{k=1}^{N} x_k \cdot (y_{k+1} - y_k) \leq 0.$$

This is exactly the statement that the mapping $s(x)$ is cyclically monotone.

2.1.4 The Monge-Ampère Equation

The Monge-Ampère equation emerges from the Monge-Kantorovich mass transport problem with quadratic cost function via a result proved by Rockafellar [78].

Theorem 2.1. *Every cyclically monotone subset of $\mathbb{R}^n \times \mathbb{R}^n$ lies in the subdifferential of a convex mapping of $\mathbb{R}^n \to \mathbb{R}$.*

This means that the solution to the transport problem (2.3) can almost everywhere be expressed as

$$s(x) = \nabla u(x)$$

where u is a convex function [67]. Given the constraints on $s(x)$ in (2.3), this convex function must satisfy the Monge-Ampère equation

$$\begin{cases} \det\left(D^2 u(x)\right) = f(x)/g(\nabla u(x)) \quad x \in X \\ \nabla u : X \to Y \\ u \text{ is convex.} \end{cases} \tag{2.5}$$

2.2 Analysis and Weak Solutions

Although the Monge-Ampère equation is a second order PDE, there is no guarantee that it will possess a classical C^2 solution. Consequently, it is necessary to use some notion of weak solution (either the viscosity or the Aleksandrov solution). In this section we present regularity results and background analysis that inform the numerical approach taken in this thesis.

2.2.1 Regularity

We begin by reviewing regularity results for the Monge-Ampère equation and the related problem of L^2 optimal transport.

Solutions of the optimal transport problem need not be smooth. An example of a singular solution (see Figure 2.3) is the problem of mapping the circle

$$X = \{(x_1, x_2) \mid x_1^2 + x_2^2 \leq 1\}$$

onto the disconnected set

$$Y = \{(x_1, x_2) \mid x_1 \leq -0.25, (x_1 + 0.25)^2 + x_2^2 \leq 1\} \cup \{(x_1, x_2) \mid x_1 \geq 0.25, (x_1 - 0.25)^2 + x_2^2 \leq 1\}.$$

In fact, the solution remains singular even if the disconnected region Y is approximated by a connected region Y_ϵ [18].

While we do not solve the problem of mapping onto a disconnected region, we are able to solve for the inverse mapping (which takes the disconnected set Y to the connected set X) in §6.4.2.

As long as the sets X,Y are bounded, we are at least guaranteed that the solution of the Monge-Ampère equation is differentiable almost everywhere with bounded gradient.

Remark. When the solution to the Monge-Ampère equation is not differentiable, the map is given by the sub-gradient rather than the gradient. This allows a single point to be mapped onto a region rather than a single point.

More regularity is guaranteed if we restrict ourselves to convex target sets Y.

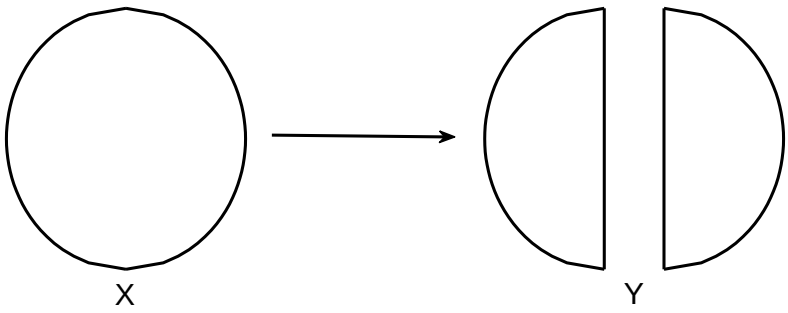


Figure 2.3: A transport problem with a singular solution.

Theorem 2.2 (Interior Regularity [16, 18])**.** *Suppose that X, Y are bounded, connected, open sets and Y is convex. Suppose also that the density functions*

$$f : X \to (0, \infty),\ g : Y \to (0, \infty)$$

are bounded away from 0 and ∞. Then the solution of the Monge-Ampère equation (1.2), (1.5), (1.1) *belongs to $C^{1,\alpha}_{loc}(X)$ for some $0 < \alpha < 1$.*

If, in addition, the density functions $f, g \in C^{\beta}$ for some $0 < \beta < 1$ then the solution of Monge-Ampère belongs to $C^{2,\alpha}_{loc}(X)$ for every $0 < \alpha < \beta$.

If both sets X, Y are uniformly convex, we can obtain regularity up to the boundary as well.

Theorem 2.3 (Boundary Regularity [17, 19])**.** *Suppose, in addition to the hypotheses of Theorem 2.2, that the sets X and Y are uniformly convex. Then the solution of Monge-Ampère is in $C^{2,\alpha}(\bar{X})$ for some $0 < \alpha < 1$.*

One of the primary concerns of this thesis is to correctly approximate the Monge-Ampère operator in the interior of the domain. For simplicity and concreteness, therefore, much of this thesis will focus on the Monge-Ampère equation with a simple right-hand side that is independent of the solution u. The system will be augmented with a simple Dirichlet

boundary condition:

$$\begin{cases} \det(D^2u(x)) = f(x), & \text{in } X \\ u(x) = \phi(x), & \text{on } \partial X \\ u \text{ is convex.} \end{cases} \tag{2.6}$$

Even in this simple setting, solutions need not be smooth. For a simple example where regularity breaks down, consider the elliptic Monge-Ampère equation in a square domain with constant Dirichlet boundary data and a strictly positive right-hand side. If we suppose that a C^2 solution exists, we can also conclude from the boundary condition that the second derivative

$$u_{x_1x_1} = 0$$

along the boundary $x_2 = 0$. Consequently, the equation reduces to

$$u_{x_1x_1}u_{x_2x_2} - u_{x_1x_2}^2 = -u_{x_1x_2}^2 = f > 0,$$

which is not possible. We conclude that even for this very simple problem with smooth data (aside from the square domain), the Monge-Ampère equation does not have a classical solution.

Remark. We can obtain a similar result even if we replace the square domain with a smooth domain that is convex but not strictly convex.

Using regularity results in [14, 15, 48, 87], we know that the Monge-Ampère equation with Dirichlet boundary conditions is guaranteed to have a unique $C^{2,\alpha}$ solution under the following conditions:

$$\begin{cases} \text{The domain } X \text{ is strictly convex with boundary } \partial X \in C^{2,\alpha}. \\ \text{The boundary values } \phi \in C^{2,\alpha}(\partial X). \\ \text{The function } f \in C^{\alpha}(X) \text{ is strictly positive.} \end{cases} \tag{2.7}$$

Remark. While is it usual to perform numerical solutions on a rectangle, regularity can break down in convex polygons [77, 88], as in the example presented earlier in this section.

2.2.2 Divergence Form of the Equation

Because the Monge-Ampère equation may not have a C^2 solution, the equation must be interpreted using some notion of weak solution.

One common approach to constructing a weak formulation of an equation is to multiply the equation by a smooth test function and integrate by parts. In order to do this, the Monge-Ampère operator needs to be written in divergence structure. In two dimensions, this can be done as follows:

$$\det(D^2 u) = \frac{1}{2}\mathrm{div}\left(\begin{pmatrix} u_{yy} & -u_{xy} \\ -u_{xy} & u_{xx} \end{pmatrix}\begin{pmatrix} u_x \\ u_u \end{pmatrix}\right).$$

However, we note that this expression still involves second derivatives of u. As a result, this approach will still require solutions of the Monge-Ampère equation to have sufficient regularity. This tends to limit the use of finite element methods to solutions that have more regularity than we can expect in general.

2.2.3 Viscosity Solutions

A more useful notion of weak solution for the Monge-Ampère equation, which will guide much of the work in this thesis, is the viscosity solution. We first recall the definition of a viscosity solution [26, 60], which is defined for the Monge-Ampère equation in [48].

Definition 2.1. Let $u \in C(X)$ be convex and $f \geq 0$ be continuous. The function u is a *viscosity subsolution (supersolution)* of the Monge-Ampère equation in X if whenever convex $\phi \in C^2(X)$ and $x_0 \in X$ are such that $(u-\phi)(x) \leq (\geq)(u-\phi)(x_0)$ for all x in a neighbourhood of x_0, then we must have

$$\det(D^2\phi(x_0)) \geq (\leq) f(x_0).$$

The function u is a *viscosity solution* if it is both a viscosity subsolution and supersolution.

Example (Viscosity solution of Monge-Ampère). We consider an example which will later be solved numerically in two and three dimensions (§4.9-4.10 and §5.3-5.4). Consider (2.6) with solution u and right-hand side f given by

$$u(\mathbf{x}) = \frac{1}{2}((|\mathbf{x}|-1)^+)^2, \qquad f(\mathbf{x}) = (1-1/|\mathbf{x}|)^+.$$

(The function f changes in three dimensions; see §2.5.2). This function u is a viscosity solution—but not a classical C^2 solution—of the Monge-Ampère equation.

We verify that this function is a viscosity solution. This only needs to be done at points where $|\mathbf{x}_0| = 1$ (since u is locally C^2 away from this circle). We note that f is equal to zero on this circle.

We begin by checking convex C^2 functions $\phi \leq u$ with $\phi(\mathbf{x}_0) = u(\mathbf{x}_0) = 0$ (that is, $u - \phi$ has a local minimum here). Since $\nabla u(\mathbf{x}_0) = 0$, we require $\nabla\phi(\mathbf{x}_0) = 0$ as well. Since u is constant in part of any neighbourhood of $\mathbf{x}_0$, any convex ϕ must also be constant in this part of the neighbourhood in order to ensure that $u - \phi$ has a local minimum. This means that ϕ has zero curvature in some directions, so that $\det D^2\phi(\mathbf{x}_0) = 0$, as required by the definition of the viscosity solution. We conclude that u is a supersolution of the Monge-Ampère equation.

We also need to check functions $\phi \geq u$ with $\phi(\mathbf{x}_0) = u(\mathbf{x}_0) = 0$ (so that $u - \phi$ has a local maximum). Since ϕ is convex, it will automatically satisfy the condition $\det D^2\phi(\mathbf{x}_0) \geq 0$. We conclude that u is also a subsolution, and is therefore a viscosity solution.

Viscosity solutions of the Monge-Ampère equation satisfy a very important property known as the *comparison principle*.

Theorem 2.4 (Comparison Principle)**.** *Let u be a sub-solution and v a super-solution of equation* (1.2) *in X with $u \leq v$ on ∂X. Then $u \leq v$ in X.*

This property guarantees uniqueness of solutions and plays an important role in the development of convergent approximation schemes.

The viscosity solution is equipped with a rich L^∞ theory that includes maximum and comparison principles. This is a very natural setting for finite difference schemes, which makes finite difference methods a natural choice for approximating viscosity solutions of the Monge-Ampère equation.

2.2.4 Aleksandrov Solutions

Next we turn our attention to the Aleksandrov solution, which is a more general weak solution than the viscosity solution. Here f is generally a measure [48]. We begin by recalling the definition of the normal mapping or subdifferential of a function.

Definition 2.2. The *normal mapping* (*subdifferential*) of a function u is the set-valued function ∂u defined by

$$\partial u(x_0) = \{p : u(x) \geq u(x_0) + p \cdot (x - x_0)\}, \quad \text{for all } x \in X.$$

For a set $V \subset X$, we define $\partial u(V) = \bigcup_{x \in V} \partial u(x)$.

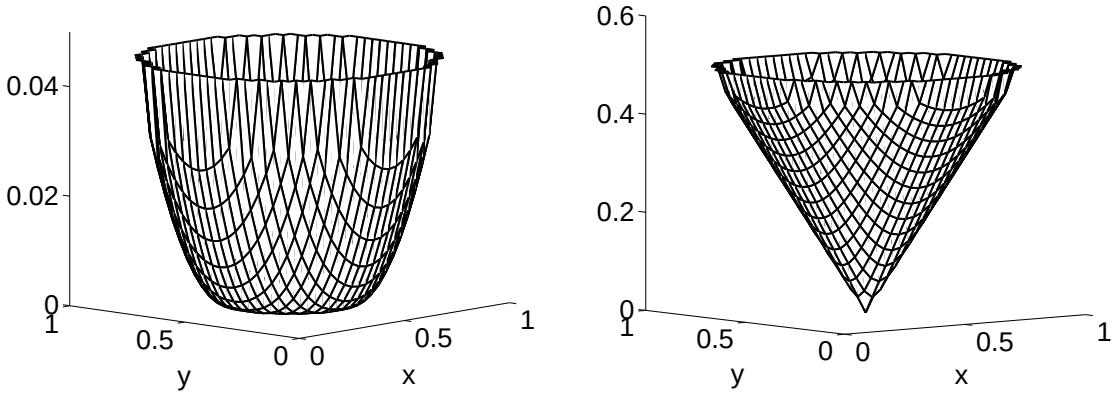


Figure 2.4: Examples that can be interpreted as (a) viscosity or (b) Aleksandrov solutions.

Now we want to look at a measure generated by the Monge-Ampère operator.

Definition 2.3. Given a function $u \in C(X)$, the *Monge-Ampère measure* associated with u is defined as

$$\mu(V) = \big|\partial u(V)\big|$$

for any set $V \subset X$.

This measure naturally leads to the notion of the generalised or Aleksandrov solution of the Monge-Ampère equation.

Definition 2.4. Let μ be a Borel measure defined in a convex set $X \in \mathbb{R}^d$. Then the convex function u is an *Aleksandrov solution* of the Monge-Ampère equation

$$\det(D^2 u) = \mu$$

if the Monge-Ampère measure associated with u is equal to the given measure μ.

Example (Aleksandrov solution). As an example, we consider the cone and the the scaled Dirac measure

$$u(\mathbf{x}) = \|\mathbf{x}\|\,, \qquad \mu(V) = \pi \int_V \delta(\mathbf{x})\, d\mathbf{x}.$$

We verify from the definition that u, μ is an Aleksandrov solution of the Monge-Ampère equation. (Since μ is a measure, we cannot interpret u as a viscosity solution of the equation.)

It is straightforward to check that the subdifferential ∂u is given by

$$\partial u(\mathbf{x}) = \begin{cases} \mathbf{x}/\|\mathbf{x}\|\,, & \|\mathbf{x}\| > 0 \\ B_1, & \mathbf{x} = \mathbf{0}, \end{cases}$$

where $B_1 = \{\mathbf{x} \,|\, \|\mathbf{x}\| \leq 1\}$. Then the associated Monge-Ampère measure will be

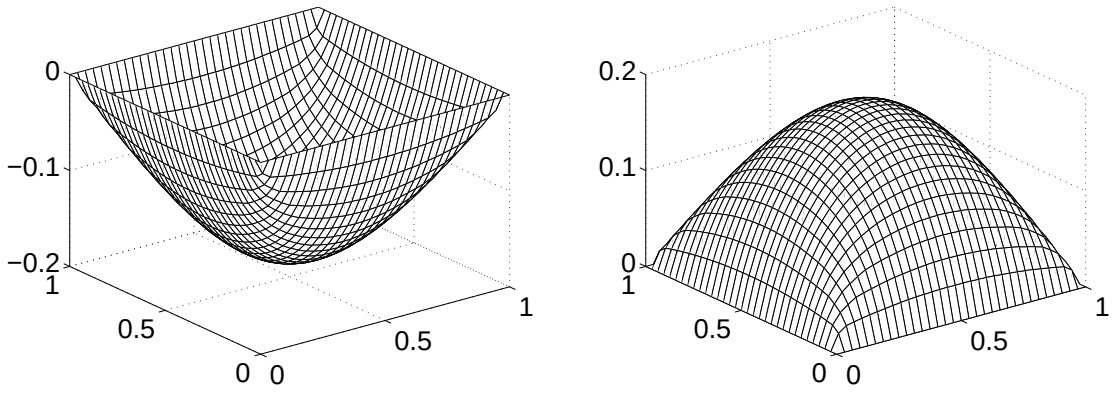
$$\big|\partial u(V)\big| = \begin{cases} \pi & \mathbf{0} \in V \\ 0 & \mathbf{0} \notin V \end{cases} = \pi \int_V \delta(x)\, dx = \mu(V).$$

2.2.5 Convexity

The convexity constraint (1.1) is necessary for uniqueness. As a simple illustration of the convexity requirement, consider the two-dimensional Monge-Ampère equation (2.6) with homogeneous Dirichlet boundary data

$$\phi(x) = 0, \quad x \in \partial X.$$

Then if u is a convex solution of the Monge-Ampère equation, -u will be a concave solution of the equation. See Figure 2.5.



(a) (b)

Figure 2.5: Without the convexity constraint, the two-dimensional Monge-Ampère equation has (a) a convex and (b) a concave solution.

For a twice continuously differentiable function u, the convexity restriction (1.1) can be written as D^2u is positive definite. Since we wish to work with less regular solutions, this

restriction can be enforced through the inequality

$$\lambda_1(D^2u) \geq 0,$$

understood in the viscosity sense [73, 75], where $\lambda_1(D^2u)$ is the smallest eigenvalue of the Hessian of u.

The convexity constraint can be absorbed into the operator by defining

$$\det{}^+(M) = \prod_{j=1}^{d} \lambda_j^+ \tag{2.8}$$

where M is a symmetric matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ and

$$x^+ = \max(x, 0).$$

Using this notation, (1.1) and (1.2) become

$$\det{}^+(D^2u(x)) = f(x), \quad x \in X. \tag{2.9}$$

2.2.6 Ellipticity

The Monge-Ampère equation is a member of the class of PDEs known as *elliptic equations*. In order to build correct numerical methods for this PDE, it is important to make use of the theory for this class of equations.

We say that the linear second-order operator

$$L[u] \equiv -\text{trace}\Big(A(x)D^2u\Big)$$

is *elliptic* if the coefficient matrix $A(x)$ is positive definite.

The definition of a nonlinear elliptic PDE operator generalises this simple definition. It also allows for operators that are non-differentiable.

Definition 2.5. Let the PDE operator $F(M)$ be a continuous function defined on symmetric $d \times d$ matrices. Then the equation

$$F(D^2u(x)) = 0$$

is *elliptic* if it satisfies the monotonicity condition

$$F(M) \geq F(N) \text{ whenever } M \leq N.$$

For symmetric matrices, the inequality $M \leq N$ means that $x^TMx \leq x^TNx$ for all $x \in \mathbb{R}^d$.

The Monge-Ampère operator

$$F(M) = -\det{}^{+}(M)$$

is a non-increasing function of the eigenvalues, so it is elliptic. We stress, however, that in the absence of the convexity constraint, the Monge-Ampère equation fails to be elliptic.

2.2.7 Linearisation

The linearisation of the Monge-Ampère equation will also play an important role in our numerical methods, particularly when we construct fast solvers.

The linearisation of the determinant is given by

$$\nabla \det(M) \cdot N = \operatorname{trace}\left(M_{adj} N\right)$$

where M_{adj} is the adjugate [81], which is the transpose of the cofactor matrix. The adjugate matrix is positive definite if and only if M is positive definite. When the matrix M is invertible, the adjugate, M_{adj}, satisfies

$$M_{adj} = \det(M) M^{-1}. \tag{2.10}$$

We now apply these considerations to the linearisation of the Monge-Ampère operator [20]. When $u \in C^2$ we can linearise this operator as

$$-\nabla_M \det(D^2 u) \cdot v = \operatorname{trace}\left(-(D^2 u)_{adj} D^2 v\right). \tag{2.11}$$

Example. In two dimensions we obtain

$$\nabla_M \det(D^2 u) v = -\left(u_{xx} v_{yy} + u_{yy} v_{xx} - 2 u_{xy} v_{xy}\right).$$

Lemma 2.1. Let $u \in C^2$. The linearisation of the Monge-Ampère operator (2.11) is elliptic if $D^2 u$ is positive definite or, equivalently, if u is (strictly) convex.

Remark. When the function u fails to be strictly convex, the linearisation can be degenerate elliptic, which affects the conditioning of the linear system (2.11). When the function u is nonconvex, the linear system can be unstable.

2.3 Newton's Method

Since we ultimately want to develop fast solvers for the Monge-Ampère equation, it is natural to turn our attention to the use of Newton's method. In this chapter, we restrict our analysis to the continuous setting. In this situation, Newton's method can be written as the iteration

$$u^{n+1} = u^n - v^n \tag{2.12}$$

where the corrector v^n solves a PDE involving the linearisation of the Monge-Ampère operator (2.11):

$$\begin{cases} \text{trace}((D^2u^n)_{adj} D^2v^n) = \det(D^2u^n) - f & \text{in } X \\ v^n = 0 & \text{on } \partial X. \end{cases}$$

This equation depends on the determinant of the Hessian of the current iterate, which we denote by

$$f^n \equiv \det(D^2u^n). \tag{2.13}$$

If the Hessian of the current iterate D^2u^n is invertible, then using (2.10) and (2.13), the equation for the corrector (2.12) can be re-expressed as

$$\begin{cases} f^n \text{trace}((D^2u^n)^{-1} D^2v^n) = f^n - f & \text{in } X \\ v^n = 0 & \text{on } \partial X. \end{cases} \tag{2.14}$$

In order for this linear PDE to be well posed, we require it to be elliptic. From Lemma 2.1, it is elliptic provided the current iterate u^n is convex.

In general, an arbitrary Newton step will not produce a convex iterate u^n. The problem is that while u^{n-1} is convex, the corrector v^{n-1} may not be. The solution to this problem is to incorporate sufficient damping into the iteration to ensure convexity of the new iterate.

Thus, we replace the Newton iteration (2.12) with the damped iteration

$$u^{n+1} = u^n - \tau v^n \tag{2.15}$$

for some $0 < \alpha \leq 1$. With a suitable damping parameter τ, which will depend on the given data, we can prove convergence of the Newton iteration to sufficiently regular solutions of the Monge-Ampère equation.

2.3.1 Convergence of Newton's method

In this section we restrict our attention to cases where the conditions (2.7) are met, which ensure $C^{2,\alpha}$ regularity of solutions.

To ensure convergence of Newton's method, we will also require an initial iterate with the properties

$$\begin{cases} u^0 \in C^{2,\alpha}(X). \\ u^0 \text{ is strictly convex.} \\ u^0 \text{ satisfies the Dirichlet boundary condition (1.5).} \\ u^0 \text{ is sufficiently close to the exact solution of (1.1),(1.2) in } C^{2,\alpha}. \end{cases} \tag{2.16}$$

Theorem 2.5 (Newton's Method for the Monge-Ampère Equation)**.** *Suppose the conditions* (2.7)*,* (2.16) *hold. Then for sufficiently small* $0 < \tau \leq 1$ *the damped Newton iteration* (2.15) *converges to the exact solution of the Monge-Ampère equation* (2.6)*.*

We prove the convergence of Newton's method using an approach similar to the proof for the periodic case in [65]. We begin with the following result about the sequence produced by the Newton iteration.

Lemma 2.2. Suppose the conditions (2.7), (2.16) are satisfied. Then we can choose $\tau_n \in (0, 1]$ so that the damped Newton iteration (2.15) produces sequences $(u^n) \in C^{2,\alpha}$, $(f^n) \in C^\alpha$ with the properties

1. Each u^n satisfies the Dirichlet condition in (1.2).
2. Each u^n is strictly convex.
3. Each $f^n > C_1 f$ for some constant C_1.
4. Each $\|f^n - f\|_{C^\alpha} \leq \|f^0 - f\|_{C^\alpha}$.

Proof. Part (1) of the lemma is trivial. We prove the remainder of this result by induction.

The base case holds trivially from (2.16) and suggests a choice of

$$0 < C_1 < \inf_X (f_0/f).$$

We proceed with the inductive step by assuming parts (2)-(4) of the lemma for $u^n \in C^{2,\alpha}$.

We denote the eigenvalues of the Hessian of u^n by

$$\lambda_1^n \leq \ldots \leq \lambda_d^n.$$

Since u^n is strictly convex, the PDE for the corrector v^n is elliptic. From Schauder elliptic theory [43] and property (4), the corrector satisfies the bound

$$\|v^n\|_{C^{2,\alpha}} \leq C_2(\lambda_1^n)\|f^n - f\|_{C^\alpha} \leq \tilde{C}_2(\lambda_1^n).$$

An immediate consequence is $u^{n+1} \in C^{2,\alpha}$ and $f^{n+1} \in C^\alpha$.

We can separate the term f^{n+1} into terms linear in the corrector plus a remainder:

$$\begin{aligned} f^{n+1} &= \det(D^2(u^n - \tau_n v^n)) \\ &= \det(D^2 u^n) - \tau_n \det(D^2 u^n)\mathrm{trace}((D^2 u^n)^{-1} D^2 v^n) + \tau_n^2 r_n \\ &= f^n - \tau_n(f^n - f) + \tau_n^2 r_n. \end{aligned}$$

Here the remainder r_n consists of products of at least two second derivatives of v^n and at most $d-2$ second derivatives of u^n. Thus we can bound the remainder by

$$\|r_n\|_{C^\alpha} \leq C_3(\lambda_1^n, \lambda_d^n)\|f^n - f\|_{C^\alpha}^2.$$

We now choose the damping parameter to satisfy

$$\tau_n < \frac{1}{C_3(\lambda_1^n, \lambda_d^n)} \min\left\{1, \frac{f(1-C_1)}{\|f^n - f\|_{C^\alpha}^2}\right\}.$$

We recall that by assumption, $f - f^n < f(1 - C_1)$. Thus we have

$$\begin{aligned} f - f^{n+1} &= (f - f^n)(1 - \tau_n) - \tau_n^2 r_n \\ &< f(1-C_1)(1-\tau_n) + \tau_n \frac{f(1-C_1)}{C_3(\lambda_1^n, \lambda_d^n)\|f^n - f\|_{C^\alpha}^2} C_3(\lambda_1^n, \lambda_d^n)\|f^n - f\|_{C^\alpha}^2 \\ &= f(1 - C_1), \end{aligned}$$

which gives us $f^{n+1} > C_1 f > 0$.

To show that u^{n+1} is strictly convex, we recall that the eigenvalues of the Hessian of u^{n+1} depend continuously on the damping parameter τ. In addition, if we set the damping parameter to 0, we simply have $u^{n+1} = u^n$, which is strictly convex. Thus for $\tau = 0$, all the λ_j^{n+1} are strictly positive. We have just shown that f^{n+1}, the product of the eigenvalues,

remains strictly positive for any choice of damping parameter between 0 and τ_n. Thus all the λ_j^{n+1} must also remain positive for any damping parameter in this range. We conclude that u^{n+1} will also be strictly convex.

Finally, we observe that

$$\begin{aligned}\|f^{n+1} - f\|_{C^\alpha} &\leq (1-\tau_n)\|f^n - f\|_{C^\alpha} + \tau_n^2\|r_n\|_{C^\alpha} \\ &< (1-\tau_n)\|f^n - f\|_{C^\alpha} + \tau_n\|f^n - f\|_{C^\alpha}^2 \\ &\leq \|f^0 - f\|\end{aligned}$$

where the last step requires $\|f^0 - f\| < 1$, which follows from the conditions (2.16). □

We also show that the sequence $f^n = \det(D^2u^n)$, which is produced by Newton's Method and defined in (2.13), will converge.

Lemma 2.3. Suppose the conditions (2.7), (2.16) are satisfied. Then we can choose $\tau \in (0, 1]$ (independent of n) so that the sequence (f^n) produced by the damped Newton's method (2.15) converges in C^α. Moreover, the sequence (u^n) is bounded in $C^{2,\alpha}$.

Proof. From Lemma 2.2, the sequence (f^n) satisfies

$$\|f^n - f\|_{C^\alpha} \leq \|f^0 - f\|_{C^\alpha}.$$

This inequality gives an upper bound on f^n. Lemma 2.2 also gives a lower bound $C_1 \inf f > 0$ for the f^n. We conclude that the sequence (u^n) is bounded uniformly in $C^{2,\alpha}$ [15]. The bounds on $\|u^n\|_{C^{2,\alpha}}$ and f^n imply that that eigenvalues of the Hessian of the u^n $(\lambda_1^n, \ldots, \lambda_d^n)$ are bounded uniformly away from 0 and infinity.

We recall now the requirement on the damping parameter:

$$\tau_n < \frac{1}{C_3(\lambda_1^n, \lambda_d^n)} \min\left\{1, \frac{f(1-C_1)}{\|f^n - f\|_{C^\alpha}^2}\right\}.$$

Since λ_1^n, λ_d^n are bounded away from 0 and infinity, the constant $C_3(\lambda_1^n, \lambda_d^n)$ is bounded and we can choose a suitable τ independent of n.

We are left with the inequality

$$\|f^{n+1} - f\|_{C^\alpha} < (1-\tau)\|f^n - f\|_{C^\alpha} + \tau\|f^n - f\|_{C^\alpha}^2,$$

which implies that f^n converges to f. □

With these lemmas, we can complete the proof of convergence of Newton's method (Theorem 2.5).

Proof of Theorem 2.5. Consider any subsequence u^{n_j} of the sequence produced by Newton's method. This subsequence is bounded in $C^{2,\alpha}$ by Lemma 2.3 and is therefore pre-compact by the Arzela-Ascoli compactness criterion. Thus there is a subsequence $u^{n_{j_k}}$ that converges in $C^{2,\alpha}$. Moreover, the corresponding subsequence $f^{n_{j_k}}$ converges to f. Since the solution of Monge-Ampère is unique, the subsequence $u^{n_{j_k}}$ must converge to the unique solution of the Monge-Ampère equation (1.1), (1.2), (1.5) and the original sequence u^n must also converge to this solution in $C^{2,\alpha}$. □

2.4 Numerical Challenges

There are a number of issues that make the Monge-Ampère equation such an interesting and challenging problem to solve numerically. We summarise several of these, which we intend to address in this thesis.

As we have already noted, the Monge-Ampère equation will not always have a classical C^2 solution. From the point of view of mappings, we want to allow for the possibility of singular or nearly singular maps. In this context, a singular solution can simply mean that one point goes to several locations (as when the solution has a corner) or an interval goes to a point (as when the solution is flat). It is desirable to allow for these situations in order to encompass a larger class of maps. When the conditions for regularity are satisfied, classical solutions can be approximated successfully using a range of standard techniques (as discussed in §1.4). However, for singular solutions, standard numerical methods can break down by becoming unstable, poorly conditioned, or by converging to the wrong solution. The challenge in this setting is to develop discretisations and solution methods that capture the weak solutions of the equation.

Another challenge is the convexity constraint, which is necessary for uniqueness. In addition, the equation (1.2) fails to be elliptic if u is non-convex (see §2.2.6), so instabilities can arise if the convexity condition (1.1) is violated. Any approximation of (1.2) requires some selection principle to choose the convex solution. This selection principle can be built into either the discretisation or the solution method.

A further goal of this thesis is to improve the accuracy of numerical methods for weak solutions of Monge-Ampère. This is important since provably convergent methods are often

less accurate than methods that work on more regular solutions. For example, the convergent monotone scheme of [74] uses a wide stencil, and the accuracy of the scheme depends on the *directional resolution*, which depends on the width of the stencil.

We also want to develop fast solvers for this equation. Although we have proved that Newton's method converges for sufficiently smooth solutions of Monge-Ampère, this does not guarantee that Newton's method will converge once the equation is discretised. Moreover, the convergence proof requires more regularity than we can generally assume for solutions of the Monge-Ampère equation. In fact, Newton's method applied to a simple discretisation of the Monge-Ampère equation can become unstable on singular examples (see §3.2.3).

Since we are interested in using the Monge-Ampère equation to generate invertible maps, it is also important that we can obtain not only the solutions of the Monge-Ampère equation, but also the gradients of the solutions. This is not automatic since a method may converge with oscillations, leading to an accurate solution with an inaccurate gradient; this results in a poor map.

2.5 Four Representative Solutions

As we build numerical methods, we want to test these on examples of varying regularity. Here we describe four representative solutions of the Monge-Ampère equation, for which we will provide detailed computational results throughout this thesis.

2.5.1 Two Dimensions

Throughout, we write $\mathbf{x} = (x, y)$ for a general point in $\mathbb{R}^2$ and $\mathbf{x}_0 = (.5, .5)$ for the center of the domain.

The first example solution, which is smooth and radial, is given by

$$u(\mathbf{x}) = \exp\left(\frac{\|\mathbf{x} - \mathbf{x}_0\|^2}{2}\right), \qquad f(\mathbf{x}) = (1 + \|\mathbf{x}\|^2)\exp(\|\mathbf{x} - \mathbf{x}_0\|^2). \tag{2.17}$$

The second example, which is C^1, is given by

$$u(\mathbf{x}) = \frac{1}{2}\left((\|\mathbf{x} - \mathbf{x}_0\| - 0.2)^+\right)^2, \quad f(\mathbf{x}) = \left(1 - \frac{0.2}{\|\mathbf{x} - \mathbf{x}_0\|}\right)^+. \tag{2.18}$$

The third example is used in §3.2.3 to demonstrate that Newton's method for standard finite differences is unstable. The solution is twice differentiable in the interior of the domain,

but has an unbounded gradient near the boundary point $(1,1)$. The solution is given by

$$u(\mathbf{x}) = -\sqrt{2 - \|\mathbf{x}\|^2}, \qquad f(\mathbf{x}) = 2\Big(2 - \|\mathbf{x}\|^2\Big)^{-2}. \tag{2.19}$$

The final is example is the cone, which was discussed in §2.2.4:

$$u(\mathbf{x}) = \sqrt{\|\mathbf{x} - \mathbf{x}_0\|}, \qquad f = \mu = \pi\, \delta_{\mathbf{x}_0} \tag{2.20}$$

This solution is only Lipschitz continuous and, in fact, is not actually a viscosity solution of the Monge-Ampère equation. Although the methods we construct in this thesis are designed to solve for viscosity solutions, we would like to see if they can also be used to obtain more general weak solutions of the Monge-Ampère equation.

In order to approximate the solution on a grid with spatial resolution h, we approximate the measure μ by its average over the ball of radius $h/2$, which gives

$$f^h = \begin{cases} 4/h^2 & \text{for } \|\mathbf{x} - \mathbf{x}_0\| \le h/2, \\ 0 & \text{otherwise.} \end{cases}$$

2.5.2 Three Dimensions

We can also generalise these examples to three dimensions. We now use $\mathbf{x} = (x, y, z)$ for a general point in $\mathbb{R}^3$ and let $\mathbf{x}_0 = (.5, .5, .5)$ be the centre of the domain. In this case, the smooth example becomes

$$u(\mathbf{x}) = \exp\left(\frac{\|\mathbf{x} - \mathbf{x}_0\|^2}{2}\right), \qquad f(\mathbf{x}) = (1 + \|\mathbf{x} - \mathbf{x}_0\|^2)\exp\left(\frac{3}{2}|\mathbf{x} - \mathbf{x}_0|^2\right). \tag{2.21}$$

The second example, the C^1 solution, is given by

$$u(\mathbf{x}) = \frac{1}{2}\Big((\|\mathbf{x} - \mathbf{x}_0\| - 0.2)^+\Big)^2, \tag{2.22}$$

$$f(\mathbf{x}) = \begin{cases} 1 - \frac{0.4}{\|\mathbf{x}-\mathbf{x}_0\|} + \frac{0.04}{\|\mathbf{x}-\mathbf{x}_0\|^2}, & |\mathbf{x} - \mathbf{x}_0| > 0.2 \\ 0 & \text{otherwise.} \end{cases}$$

The third example is the surface of a ball, which as in two dimensions is differentiable in the interior of the domain, but has an unbounded gradient at the boundary.

$$u(\mathbf{x}) = -\sqrt{3 - \|\mathbf{x}\|^2}, \qquad f(\mathbf{x}) = 3(3 - \|\mathbf{x}\|^2)^{-5/2}. \tag{2.23}$$

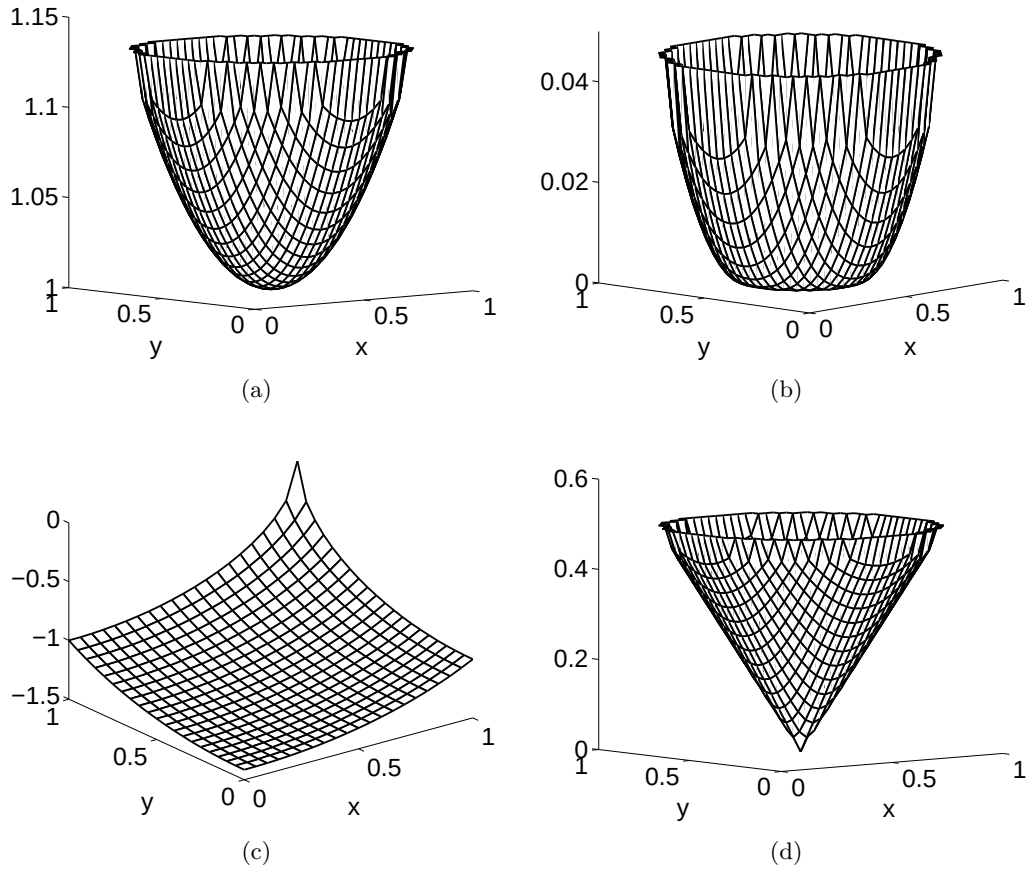


Figure 2.6: Representative solutions of Monge-Ampère: (a) C^2 example, (b) C^1 example, (c) example with blow-up, and (d) Lipschitz example.

Chapter 3

Standard Finite Difference Methods

In this thesis, we want to develop finite difference methods for solving the Monge-Ampère equation numerically. This is a two step process:

- We must *discretise* the equation to produce a system of nonlinear equations.
- We much build a *solution method* for the discrete system of equations.

In this chapter, we describe a natural, centred difference discretisation of the Monge-Ampère equation. We build several different solvers for the resulting systems and discuss the advantages and limitations of this discretisation.

3.1 Discretisation

The Monge-Ampère operator has a particularly simple form in two dimensions:

$$\det(D^2u) = \frac{\partial^2 u}{\partial x^2}\frac{\partial^2 u}{\partial y^2} - \left(\frac{\partial^2 u}{\partial x \partial y}\right)^2, \quad \text{in } X \subset \mathbb{R}^2. \tag{3.1}$$

In this case, a standard discretisation of the operator is given by

$$MA^S[u] \equiv (\mathcal{D}_{xx}u)(\mathcal{D}_{yy}u) - (\mathcal{D}_{xy}u)^2 \tag{$MA)^S$}$$

where, writing h for the spatial resolution of the grid,

$$\begin{aligned}
[\mathcal{D}_{xx}u]_{ij} &= \frac{1}{h^2}\left(u_{i+1,j} + u_{i-1,j} - 2u_{i,j}\right) \\
[\mathcal{D}_{yy}u]_{ij} &= \frac{1}{h^2}\left(u_{i,j+1} + u_{i,j-1} - 2u_{i,j}\right) \\
[\mathcal{D}_{xy}u]_{ij} &= \frac{1}{4h^2}\left(u_{i+1,j+1} + u_{i-1,j-1} - u_{i-1,j+1} - u_{i+1,j-1}\right).
\end{aligned}$$

In three dimensions, the Monge-Ampère operator has the form

$$\begin{aligned}
\det(D^2u) = &\frac{\partial^2 u}{\partial x^2}\frac{\partial^2 u}{\partial y^2}\frac{\partial^2 u}{\partial z^2} \\
&+ 2\frac{\partial^2 u}{\partial x \partial y}\frac{\partial^2 u}{\partial x \partial z}\frac{\partial^2 u}{\partial y \partial z} - \frac{\partial^2 u}{\partial x^2}\left(\frac{\partial^2 u}{\partial y \partial z}\right)^2 - \frac{\partial^2 u}{\partial y^2}\left(\frac{\partial^2 u}{\partial x \partial z}\right)^2 - \frac{\partial^2 u}{\partial z^2}\left(\frac{\partial^2 u}{\partial x \partial y}\right)^2. \quad (3.2)
\end{aligned}$$

We can discretise this using centred differences, just as we did in the two-dimensional case. Of course, the same thing can also be accomplished in higher dimensions.

It is important to recognise that there is no reason to expect that this discretisation will converge to the correct weak solution of the Monge-Ampère equation. Convergence of schemes for nonlinear equations is not simply a matter of verifying consistency and stability. In fact, this discretisation is not amenable to the proof techniques that will be discussed in Chapter 4. However, this discretisation does appear to behave correctly for a suprising range of challenging examples. Moreover, it will play an important role in the construction of convergent, higher-order schemes in Chapter 5.

We also note that the solution of the discretised system need not be unique, which can introduce instabilities into solvers. This is because the discretisation does not enforce the convexity constraint. Consequently, it is necessary to build the convexity constraint into the solution method.

3.2 Newton's Method

The natural finite difference discretisation results in a system of nonlinear equations that must be solved. One way to attempt this is using Newton's method.

To solve the discretised equation

$$MA[u] = f$$

we use want to use a Newton iteration

$$u^{n+1} = u^n - v^n.$$

The corrector v^n solves the linear system

$$\left(\nabla_u MA[u^n]\right) v^n = MA[u^n] - f. \tag{3.3}$$

To set up the equation (3.3), the Jacobian of the scheme is needed. For the natural finite differences, the Jacobian of the two-dimensional Monge-Ampère operator is given by

$$\nabla_u MA^N[u] = (\mathcal{D}_{xx}u)\mathcal{D}_{yy} + (\mathcal{D}_{yy}u)\mathcal{D}_{xx} - 2(\mathcal{D}_{xy}u)\mathcal{D}_{xy}, \tag{3.4}$$

which is a discrete version of the linearisation of Monge-Ampère (2.11). These ideas are easily extended to higher dimensions, though the expressions become much more complicated. In three-dimensions, for example, the Jacobian is given by

$$\begin{aligned}\nabla_u MA^N[u] = {} & (\mathcal{D}_{yy}u\mathcal{D}_{zz}u - (\mathcal{D}_{yz}u)^2)\mathcal{D}_{xx} + (\mathcal{D}_{xx}u\mathcal{D}_{zz}u - (\mathcal{D}_{xz}u)^2))\mathcal{D}_{yy} \\ & + (\mathcal{D}_{xx}u\mathcal{D}_{yy}u - (\mathcal{D}_{xy}u)^2)\mathcal{D}_{zz} + 2(\mathcal{D}_{xz}u\mathcal{D}_{yz}u - \mathcal{D}_{zz}u\mathcal{D}_{xy}u)\mathcal{D}_{xy} \\ & + 2(\mathcal{D}_{xy}u\mathcal{D}_{yz}u - \mathcal{D}_{yy}u\mathcal{D}_{xz}u)\mathcal{D}_{xz} + 2(\mathcal{D}_{xy}u\mathcal{D}_{xz}u - \mathcal{D}_{xx}u\mathcal{D}_{yz}u)\mathcal{D}_{yz}.\end{aligned} \tag{3.5}$$

3.2.1 Regularisation of the Jacobian

One obvious danger with using Newton's method is that the Jacobian matrix may not be invertible, which would prevent us from obtaining the corrector. For example, we might initialise Newton's method with the exact solution (pictured in Figure 2.6(b))

$$u(\mathbf{x}) = \frac{1}{2}\left((\|\mathbf{x} - \mathbf{x}_0\| - 0.2)^+\right)^2.$$

This function is constant inside the circle $\|\mathbf{x} - \mathbf{x}_0\| \leq 2$. Consequently, the second derivatives (as well as their discrete approximations) will vanish in this region, which will cause the Jacobian matrix (3.4) to be singular.

To ensure that we can actually solve the linear systems that appear in the implementation of Newton's method, we regularise the Jacobian matrices. This will not change the fixed points of Newton's method; it simply ensures that we can solve for the corrector at each step. We describe the regularisation process in the two-dimensional case; the generalisation to higher dimensions is straightforward.

We choose a parameter $\epsilon > 0$, replace the discrete second derivatives appearing in the Jacobian by

$$\mathcal{D}^{\epsilon}_{xx}u = \max\{\mathcal{D}_{xx}u, \epsilon\}, \quad \mathcal{D}^{\epsilon}_{yy}u = \max\{\mathcal{D}_{yy}u, \epsilon\},$$

and ensure that the mixed derivatives satisfy

$$\left|\mathcal{D}^{\epsilon}_{xy}u\right| < \sqrt{\mathcal{D}^{\epsilon}_{xx}u \cdot \mathcal{D}^{\epsilon}_{yy}u}.$$

3.2.2 Damping

We also incorporate damping into the Newton iteration. That is, we replace the Newton's method with

$$u^{n+1} = u^n - \tau v^n$$

where the damping parameter $\tau \in (0, 1]$ is chosen to ensure that the residual is decreasing.

In many cases, we may simply choose $\tau = 1$. However, allowing additional damping can be helpful if a poor initial guess is chosen or if solutions are non-smooth.

3.2.3 Failure of Newton's Method

We have already noted in 2.3 that the convergence proof for the continuous Newton iteration does not guarantee that Newton's method will converge for a discretised system, particularly when solutions are non-smooth. One issue with the natural discretisation is that there is no guarantee that a Newton step will preserve convexity, which can lead to instabilities in the iteration. As an example of this, we look at the exact solution

$$u(\mathbf{x}) = -\sqrt{2 - \|\mathbf{x}\|^2}, \qquad f(\mathbf{x}) = 2\Big(2 - \|\mathbf{x}\|^2\Big)^{-2}$$

on the domain $[0, 1] \times [0, 1]$. The gradient of the solution is unbounded at the point $(1, 1)$. The singularity arises from the fact that f is unbounded there, which leads to an instability in Newton's method. The result after performing two iterations of an undamped Newton's method, along with the gradient map, is illustrated in Figure 3.1. We also remark that if damping is incorporated, the iteration will simply stagnate (that is, the damping parameter is forced to zero). The correct computed solution is presented in Figure 2.6(c)-5.5(h).

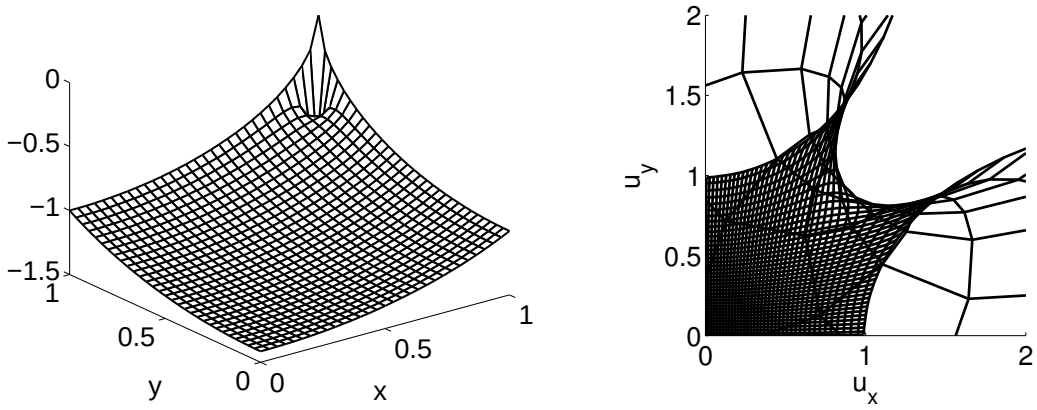


Figure 3.1: Failure of Newton's method using standard finite differences: the solution oscillates and becomes non-convex. (a) Solution and (b) gradient map after two iterations.

3.3 Two-Dimensional Solution Methods

One of the chief limitations of Newton's method for the natural finite difference discretisation is that it is not required to respect the convexity constraint (3.8). Since convexity is not built in to the discretisation, it must instead be enforced by the solution method. In two dimensions, this can be accomplished by exploiting the quadratic structure of the equation. We now develop two methods (which are also discussed in the M.Sc. thesis [38])–an explicit Gauss-Seidel iteration and a semi-implicit Poisson solver–for solving the two-dimensional Monge-Ampère equation.

3.4 Explicit Gauss-Seidel Iteration

One of the limitations of the natural finite difference discretisation is that it does not enforce the convexity constraint. In two dimensions, however, we can exploit the quadratic structure of the Monge-Ampère equation in order to select the convex solution. This leads to a robust Gauss-Seidel iteration for the two-dimensional Monge-Ampère equation.

We recall the natural finite difference discretisation of the two-dimensional equation, as

in $(MA)^S$:

$$\left(\frac{u_{i+1,j}+u_{i-1,j}-2u_{ij}}{h^2}\right)\left(\frac{u_{i,j+1}+u_{i,j-1}-2u_{ij}}{h^2}\right) - \left(\frac{u_{i+1,j+1}+u_{i-1,j-1}-u_{i-1,j+1}-u_{i+1,j-1}}{4h^2}\right)^2 = f_{ij}.$$

Solving this quadratic equation for u_{ij} and choosing the smaller root in order to select the convex solution, we obtain:

$$u_{ij} = \frac{d_1+d_2}{2} - \sqrt{\left(\frac{d_1-d_2}{2}\right)^2 + \left(\frac{d_3-d_4}{4}\right)^2 + \frac{1}{4}f_{ij}h^4} \tag{3.6}$$

where we introduce the notation

$$\begin{aligned} d_1 &= \tfrac{u_{i+1,j}+u_{i-1,j}}{2} \qquad d_2 = \tfrac{u_{i,j+1}+u_{i,j-1}}{2} \\ d_3 &= \tfrac{u_{i+1,j+1}+u_{i-1,j-1}}{2} \quad d_4 = \tfrac{u_{i-1,j+1}+u_{i+1,j-1}}{2}. \end{aligned} \tag{3.7}$$

We can now use Gauss-Seidel iteration to find the fixed point of (3.6). Dirichlet boundary conditions are enforced at boundary grid points.

Remark. In the computations of §3.6, we perform the Gauss-Seidel iteration using a lexicographical ordering. Other orderings are possible and may improve convergence and allow for parallelisation of the method.

3.4.1 Improving Convexity

As explained in the convergence proof of the wide stencil schemes [74], the main obstacle to monotonicity of the discrete scheme is the lack of convexity along directions other than grid lines. Because we are looking for the convex solution of the Monge-Ampère equation, the solution should satisfy

$$u(x) \le \frac{u(x+h)+u(x-h)}{2} \tag{3.8}$$

for all grid directions h. We check that this holds in some of the grid directions. This convexity is partially built in to (3.6).

Lemma 3.1. The fixed point of (3.6) satisfies the inequalities (3.8) for the grid directions

$$h = (1,0), \quad (0,1).$$

Proof. We assume without loss of generality that

$$\frac{u_{i,j+1} + u_{i,j-1}}{2} \leq \frac{u_{i+1,j} + u_{i-1,j}}{2}.$$

In the notation of Equation (3.7) this reads

$$d_2 \leq d_1.$$

Since f is non-negative,

$$\begin{aligned} u_{ij} &\leq \frac{d_1 + d_2}{2} - \frac{d_1 - d_2}{2} \\ &= d_2 \\ &= \frac{u_{i,j+1} + u_{i,j-1}}{2} \\ &\leq \frac{u_{i+1,j} + u_{i-1,j}}{2}. \end{aligned}$$

□

From this lemma, we observe that solutions of (3.6) are necessarily "convex" in the x and y directions. Along the lines of [74] (where convexity along several directions ensures convergence), we can also build additional convexity requirements into our method. This is accomplished by modifying (3.6) slightly:

$$u_{ij} = \min \left\{ \frac{d_1 + d_2}{2} - \sqrt{\left(\frac{d_1 - d_2}{2}\right)^2 + \left(\frac{d_3 - d_4}{4}\right)^2 + \frac{1}{4} f_{ij} h^4}, \quad d_3, \quad d_4 \right\}. \tag{3.9}$$

Lemma 3.2. The fixed point of (3.10) satisfies the inequality (3.8) for the grid directions

$$h = (1, 0), \quad (0, 1), \quad (-1, 1), \quad (1, 1).$$

Proof. The proof of the first part of this lemma is the same as the proof of the first part of Lemma 3.1. The second half of this lemma is built directly into (3.9). □

3.4.2 Higher Dimensions

One big limitation of this Gauss-Seidel iteration is that it does not generalise naturally to higher dimensions. In two-dimensions, the Monge-Ampère equation is quadratic in the second derivatives. Consequently, the discretised equations are quadratic in the solution values u_{ij} and it is straightforward to solve these quadratic equations for the correct (convex) solution.

In higher dimensions, the quadratic structure of the Monge-Ampère equation is lost. In three dimensions, for example, the equation is cubic in the second derivatives. As a result, solving for the convex solution becomes a much more complicated task.

3.5 Semi-Implicit Poisson Iteration

The next method is based on a reformulation of the two-dimensional Monge-Ampère equation. As in the previous section, we use the convexity requirement to select the correct square root.

Definition 3.1. We define the operator

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

This operator can be used to reformulate (1.2) due to the following lemma.

Lemma 3.3. The convex solution of (1.2) satisfies

$$u = T[u]. \tag{3.10}$$

Proof. Let v be the convex solution of (1.2), which satisfies

$$f - \det(D^2 v) = 0.$$

Inserting this into Definition 3.1 we obtain

$$\begin{aligned} T[v] &= \Delta^{-1}\left(\sqrt{(\Delta v)^2}\right) \\ &= \Delta^{-1}(|\Delta v|). \end{aligned}$$

Since v is convex,

$$\Delta v > 0.$$

As a result,

$$\begin{aligned} T[v] &= \Delta^{-1}(|\Delta v|) \\ &= \Delta^{-1}(\Delta v) \\ &= v. \end{aligned}$$

Therefore, v is a fixed point of (3.10). □

In this section, we focus on the two-dimensional equation. With this in mind we rewrite the operator $T[u]$ in two dimensions.

Lemma 3.4. In $\mathbb{R}^2$, the operator $T[u]$ defined in Definition 3.1 is equivalent to

$$\begin{aligned} T[u] &= \Delta^{-1}\left(\sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2 + 2f}\right) \\ &= \Delta^{-1}\left(\sqrt{\left|D^2u\right|^2 + 2f}\right). \end{aligned} \tag{3.11}$$

Proof. In $\mathbb{R}^2$, $T[u]$ takes the form

$$\begin{aligned} T[u] &= \Delta^{-1}\left(\sqrt{(\Delta u)^2 + 2(f - \det(D^2u))}\right) \\ &= \Delta^{-1}\left(\sqrt{(u_{xx} + u_{yy})^2 + 2f - 2(u_{xx}u_{yy} - u_{xy}^2)}\right) \\ &= \Delta^{-1}\left(\sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xx}u_{yy} + 2f - 2u_{xx}u_{yy} + 2u_{xy}^2)}\right) \\ &= \Delta^{-1}\left(\sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2 + 2f}\right) \\ &= \Delta^{-1}\left(\sqrt{\left|D^2u\right|^2 + 2f}\right). \end{aligned}$$

□

Obtaining the fixed point of (3.10) consists in iterating $u_{n+1} = T[u_n]$ by solving

$$\Delta u_{n+1} = \sqrt{u_{n,xx}^2 + u_{n,yy}^2 + 2u_{n,xy}^2 + 2f}.$$

with the prescribed Dirichlet boundary conditions.

We implement (3.10) using a simple finite difference method. This involves discretising (3.11) using central differences (as with the first method) and iterating to find the fixed point. In the computations of §3.6 we solved the resulting Poisson equation using the MATLAB backslash operator.

3.5.1 Contractivity

In the numerical experiments of §3.6 we observe that the Poisson iteration converges very quickly when the solutions are smooth and the function f is strictly positive, but is fairly slow when solutions are not smooth or f is very close to 0. In this section we consider a one-dimensional version of (3.10) and prove that this mapping is a contraction with a rate of convergence depending on how far f is from zero. We provide a similar result for the two-dimensional case on a rectangle, although we do not have a complete proof that (3.10) is a contraction mapping on a general domain. We begin with an observation about the contractivity of the real valued function $h(x) = \sqrt{a^2 + x^2}$.

Lemma 3.5. The function

$$h(x) = \sqrt{a^2 + x^2}$$

is a strict contraction on the domain

$$\{|x| \leq ka\}.$$

In other words, there exists a constant $\mu_k < 1$ such that

$$\left|h(x_1) - h(x_2)\right| \leq \mu_k |x_1 - x_2|$$

for any x_1, x_2 in $\{|x| \leq ka\}$.

Proof.

$$\begin{aligned} \left|h'(x)\right| &= \frac{|x|}{\sqrt{a^2 + x^2}} \\ &\leq \frac{ka}{\sqrt{a^2 + k^2 a^2}} \\ &= \frac{k}{\sqrt{1 + k^2}} \\ &= \mu_k < 1. \end{aligned}$$

It follows that

$$\left|h(x_1) - h(x_2)\right| \leq \mu_k |x_1 - x_2|. \qquad \square$$

Lemma 3.6. Let v be an exact, smooth solution of (3.1) and u a smooth function. Further suppose that

$$f \geq \alpha > 0$$

is a strictly positive function. Then at every point in the domain

$$\begin{aligned} \left|\Delta(T[u] - T[v])\right| &= \left|\sqrt{2f + \left|D^2 u\right|^2} - \sqrt{2f + \left|D^2 v\right|^2}\right| \\ &\leq \mu \left|D^2(u - v)\right| \end{aligned}$$

for some constant $\mu < 1$.

Proof. Since u, v are smooth and f is strictly positive, there exists a constant k so that

$$\left|D^2 u\right|, \left|D^2 v\right| \leq k\sqrt{2f}.$$

It follows from Lemma 3.5 that

$$\begin{aligned}\left|\sqrt{2f+\left|D^2u\right|^2}-\sqrt{2f+\left|D^2v\right|^2}\right| &\leq \mu_k\left|\left|D^2u\right|-\left|D^2v\right|\right| \\ &\leq \mu_k\left|D^2(u-v)\right|,\end{aligned}$$

which completes the proof. □

Remark. It is worth noting that as $f \to 0$ or u becomes more and more non-smooth, the constant k will increase so that μ_k increases and approaches 1.

Now we define the semi-norm

$$\|u\|_L = \int_\Omega (\Delta u)^2 \, dx \, dy. \tag{3.12}$$

Lemma 3.7. Let $u(x, y)$ be a C^2 function that vanishes on the boundary of a rectangle Ω. Then

$$\int_\Omega (\Delta u)^2 \, dx \, dy = \int_\Omega |D^2u|^2 \, dx \, dy.$$

Proof. Using repeated integration by parts we find that

$$\begin{aligned}\int_\Omega (\Delta u)^2 \, dx \, dy &= \int_\Omega \left(u_{xx}^2 + u_{yy}^2 + 2u_{xx}u_{yy}\right) \, dx \, dy \\ &= \int_\Omega \left(u_{xx}^2 + u_{yy}^2 - 2u_x u_{xyy}\right) \, dx \, dy \\ &= \int_\Omega \left(u_{xx}^2 + u_{yy}^2 + 2u_{xy}u_{xy}\right) \, dx \, dy \\ &= \int_\Omega |D^2u|^2 \, dx \, dy.\end{aligned}$$

Throughout this computation the boundary terms vanish since u is constant along the sides of the rectangle (and thus at any point on the boundary either u_x, u_{xx} or u_y, u_{yy} vanish). □

Theorem 3.1 (Contractivity on a Rectangle)**.** *The mapping T on a rectangular domain Ω is a contraction in the semi-norm $\|u\|_L$.*

Proof. Let u, v be any C^2 functions that satisfy the Dirichlet boundary conditions associated with (3.1). Compute

$$\begin{aligned}\|T(u) - T(v)\|_L &= \int_\Omega [\Delta(T(u) - T(v))]^2 \, dx \, dy \\ &\leq \int_\Omega \mu^2 \left|D^2(u-v)\right|^2 \, dx \, dy,\end{aligned}$$

where the last step follows from Lemma 3.6. Since u and v are identical on $\partial\Omega$, we can apply Lemma 3.7 to obtain

$$\begin{aligned}\|T(u) - T(v)\|_L &\leq \mu^2 \int_\Omega [\Delta(u-v)]^2 \, dx \, dy \\ &= \mu^2 \|u - v\|_L.\end{aligned}$$

Since $\mu^2 < 1$, this completes the proof. □

We have already noted that for f close to zero or u with large second derivatives, the constant μ will be close to 1. This suggests that the mapping $T[u]$ will converge more slowly in these situations, which is exactly what we observe in the computations of §3.6.

Remark. We should note that the proof of convergence for this Poisson iteration is only valid in the continuous setting. That is, it assumes that we are exactly solving the Poisson equation at each step. Thus this proof does not guarantee that a particular numerical implementation of the Poisson iteration will converge.

3.5.2 Higher Dimensions

To generalize this Poisson iteration to $\mathbb{R}^d$, we can write the Laplacian in terms of the eigenvalues of the Hessian: $\Delta u = \sum_{i=1}^d \lambda_i[D^2 u]$. Taking the d-th power and expanding, gives the sum of all possible products of d eigenvalues.

$$(\Delta u)^d = d! \prod_{i=1}^d \lambda_i + P(\lambda_1, \ldots, \lambda_d),$$

where $P(\lambda)$ is a d-homogeneous polynomial, which we won't need explicitly.

The result is the semi-implicit scheme

$$\Delta u^{n+1} = (d! f + P(\lambda_1[D^2 u^n], \ldots, \lambda_d[D^2 u^n]))^{1/d}. \tag{3.13}$$

A natural initial value for the iteration is given by the solution of

$$\Delta u^0 = (d! f)^{1/d}. \tag{3.14}$$

Unfortunately, because the equation is no longer quadratic in dimensions greater than two, there is no reason to expect that this iteration will preserve convexity. We also recall that even in two dimensions, the Poisson iteration could become very slow when solutions were singular.

3.6 Computational Results

We are now ready to provide numerical results for the Gauss-Seidel and Poisson iterations. We have tested these methods on a number of examples of varying regularity. For concreteness, we provided detailed results for the four representative examples described in §2.5.

All of the computations are performed on an $N \times N$ grid with spatial resolution h.

To initialise the iterations, we first solve the problem on a coarser grid and interpolate the results onto the refined grid. To obtain the coarse solution, we initialise the iterations with the solution of the Poisson equation

$$\Delta u = \sqrt{2f}$$

with the correct Dirichlet boundary conditions for the problem. However, we note that both methods appear to converge regardless of the initial data. In particular, they converge even when we initialise with random data that does not respect the Dirichlet boundary conditions.

Results are summarised in Table 3.1.

3.6.1 Accuracy

We provide log-log plots of error in Figure 3.2. For the C^2 example, the natural finite difference discretisation gives $\mathcal{O}(h^2)$ accuracy, as anticipated by the formal error estimate coming from the Taylor series. Not surprisingly, the accuracy becomes worse on examples with less regularity. For the C^1 example, accuracy is $\mathcal{O}(h)$. For the example with blow-up at the boundary, accuracy is only $\mathcal{O}(h^{0.5})$ and for the Lipschitz example, the solution accuracy is $\mathcal{O}(h)$. Although the natural finite difference discretisation becomes less accurate on examples with less regularity, the discretisation does appear to converge to the weak solution in all the examples we have computed.

3.6.2 Computation Time

We also look at the computation time required by the Gauss-Seidel and Poisson iterations. These results are plotted in Figure 3.3.

C^2 Example (2.17)

N	Max Error	Iterations		CPU Time (seconds)	
		Poisson	Gauss-Seidel	Poisson	Gauss-Seidel
31	4.54×10^{-5}	42	2204	0.3	0.8
45	2.11×10^{-5}	44	4597	0.9	3.0
63	1.06×10^{-5}	44	8872	1.6	10.7
89	0.53×10^{-5}	45	17339	4.1	41.0
127	0.26×10^{-5}	44	34419	8.3	163.1
181	0.13×10^{-5}	44	67968	19.3	666.0
255	0.06×10^{-5}	45	—	44.0	—
361	0.03×10^{-5}	55	—	124.5	—

C^1 Example (2.18)

N	Max Error	Iterations		CPU Time (seconds)	
		Poisson	Gauss-Seidel	Poisson	Gauss-Seidel
31	3.78×10^{-4}	164	1848	1.0	0.6
45	1.82×10^{-4}	367	3854	6.0	2.6
63	1.34×10^{-4}	839	7430	24.7	8.8
89	0.85×10^{-4}	1497	14520	114.0	33.8
127	0.59×10^{-4}	2890	28816	447.1	139.9
181	0.37×10^{-4}	—	56885	—	541.8

Example with Blow-up (2.19)

N	Max Error	Iterations		CPU Time (seconds)	
		Poisson	Gauss-Seidel	Poisson	Gauss-Seidel
31	1.74×10^{-2}	74	2205	0.4	0.7
45	1.47×10^{-2}	81	4601	1.2	2.8
63	1.26×10^{-2}	90	8885	2.5	9.6
89	1.07×10^{-2}	102	17378	7.4	36.5
127	0.90×10^{-2}	115	34515	16.6	144.8
181	0.76×10^{-2}	130	68177	45.0	577.6
255	0.64×10^{-2}	148	—	113.7	—
361	0.54×10^{-2}	177	—	331.9	—

$C^{0,1}$ (Lipschitz) Example (2.20)

N	Max Error	Iterations		CPU Time (seconds)	
		Poisson	Gauss-Seidel	Poisson	Gauss-Seidel
31	5.19×10^{-3}	844	2453	4.9	0.8
45	3.82×10^{-3}	1673	5137	25.0	3.1
63	2.86×10^{-3}	3100	9943	86.9	10.6
89	2.12×10^{-3}	5815	19502	417.2	40.1
127	1.54×10^{-3}	11033	38857	1576.7	160.4
181	1.12×10^{-3}	—	77016	—	642.9

Table 3.1: Computation times and maximum error for the Poisson and Gauss-Seidel methods on four representative examples.

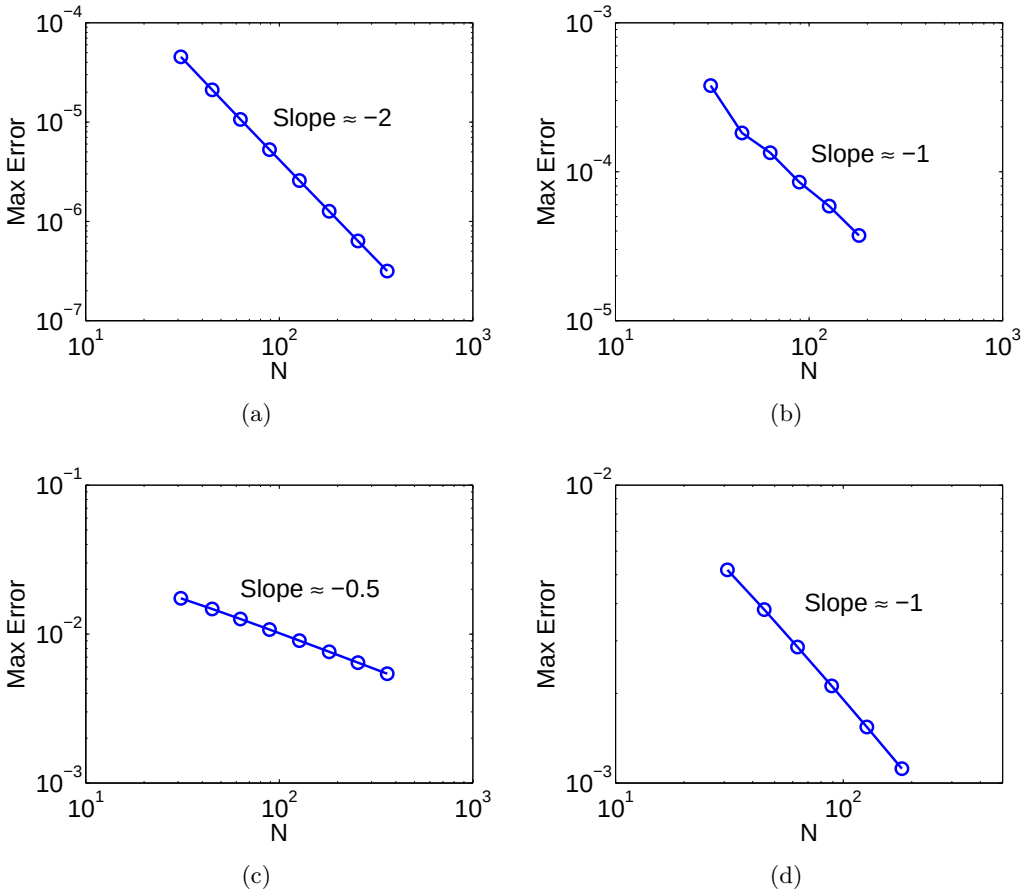


Figure 3.2: Error of standard discretisation on the (a) C^2 example, (b) C^1 example, (c) example with blow-up, and (d) Lipschitz example.

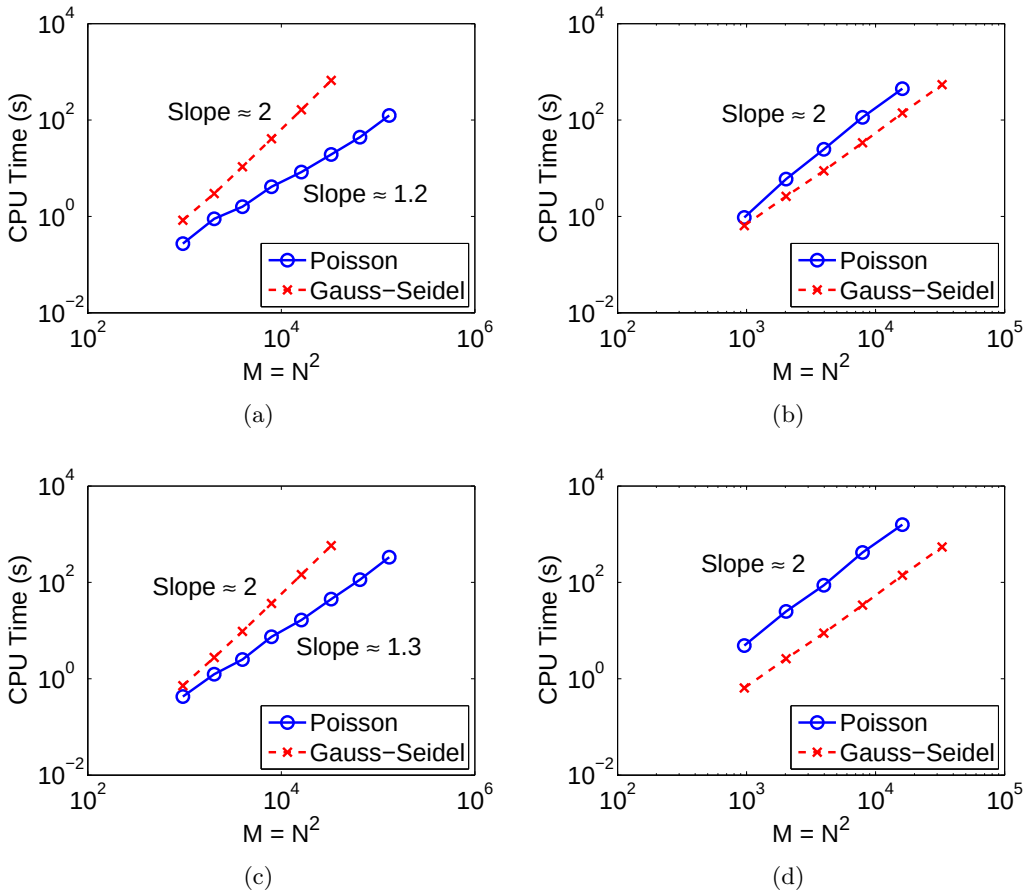


Figure 3.3: Computation time for the Poisson and Gauss-Seidel methods on the (a) C^2 example, (b) C^1 example, (c) example with blow-up, and (d) Lipschitz example.

First, we observe that the Gauss-Seidel method requires a moderate amount of time to converge. However, it is interesting to note that the computation time appears to be essentially independent of the regularity of the solutions.

The Poisson iteration, on the other hand, appears to be very fast for smooth solutions. However, on solutions with less regularity, this iteration can be extremely slow. This is consistent with the analysis of §3.5.1, where we showed that the iteration is a contraction method with a rate depending on the size of the second derivatives and the strict convexity of the solution.

3.7 Conclusions

In this chapter, we have investigated the use of standard finite difference discretisations for the Monge-Ampère equation. In all the examples we computed, these natural finite differences appear to converge to the viscosity solution of Monge-Ampère. Unfortunately, although we observe numerical convergence for a number of examples of varying regularity, we cannot prove that this discretisation will always converge to the correct weak solution.

We find that fast methods such as Newton's method can fail for this simple discretisation. Consequently, it is necessary to develop new solutions methods for solving the system of discrete equations coming from the natural finite difference discretisation of the Monge-Ampère equation. In two dimensions, we have developed two solution methods to accomplish this. The first is an explicit Gauss-Seidel iteration that is only moderately fast, but has a solution time that appears independent of the solution regularity. The second method involves iteratively solving a Poisson equation. This method is quite fast for smooth examples, but can be very slow on examples with less regularity.

We also recall that these methods, while quite robust in two dimensions, do not generalise naturally to higher dimensions. We conclude that this natural finite difference discretisation, though it appears more powerful than we might suppose given the lack of convergence theory, is not the right approach for constructing general solvers for the Monge-Ampère equation.

Chapter 4

Monotone Finite Difference Methods

In the last chapter, we found that standard finite difference techniques applied to the Monge-Ampère equation face several limitations. These include the lack of convergence proof, the difficulty of generalising to higher dimensions, and the challenge of building fast solvers for singular solutions. In light of these difficulties, we now turn our attention to more sophisticated discretisation techniques. This enables us to construct finite difference methods that provably converge to the viscosity solution of the Monge-Ampère equation in any spatial dimension. We also use Newton's method to build fast, convergent solvers for the discretised system. Finally, we provide computational results in both two and three dimensions.

To keep the key ideas of these methods clear, we begin by limiting our discussion to the case where the right-hand side of the equation does not depend on gradients of the solution. That is, we focus on the theory of convergent finite difference methods for the problem

$$\det(D^2 u) = F(x, \nabla u) \equiv f(x). \tag{4.1}$$

Towards the end of this chapter, we will also show how our techniques can be extended to more general Monge-Ampère equations.

Because we want to focus on the problem of correctly approximating the Monge-Ampère operator in the interior of the domain, we will simply implement Dirichlet boundary conditions. As Dirichlet (or, sometimes, periodic) boundary conditions are found most often in the literature, this will also enable us to more easily compare our methods with results obtained using other methods for solving the Monge-Ampère equation.

4.1 Convergence of Finite Difference Schemes

While it is not too hard to construct consistent, stable approximation schemes for the Monge-Ampère equation, this is not enough to guarantee convergence to the weak (viscosity) solution of this nonlinear equation. To motivate and lay the theoretical foundation for a convergent discretisation of the Monge-Ampère equation, we review a framework for convergence of finite difference schemes to the viscosity solutions of elliptic PDEs. This theory, developed by Barles and Souganidis [4] and extended by Oberman [72], gives more easily verified conditions for when approximation schemes converge to the unique viscosity solution of a PDE. It relies on the fact that viscosity solutions are stable under perturbations of the operator as long as the perturbed operator is also elliptic. In this setting, the consistent finite difference scheme can be regarded as a perturbed operator.

Theorem 4.1 (Convergence of Approximation Schemes [4])**.** *The solution of a consistent, stable, monotone approximation scheme converges uniformly on compact subsets to the unique viscosity solution of the limiting equation, provided this equation satisfies a comparison principle.*

One of the advantages of this convergence result is that it only requires consistency to be verified on smooth solutions, which is much simpler than checking consistency with the viscosity solution for singular functions. In [72], this theorem was used to establish a framework for building and verifying the monotonicity of finite difference schemes. This was accomplished using the notion of a degenerate elliptic approximation scheme. We recall that a finite difference equation at the discrete location x_i, $i = 1, \ldots, M$ has the form

$$F^i[u] = F^i(u_i, u_i - u_j|_{j \neq i}).$$

Then a degenerate elliptic finite difference scheme is characterised as follows.

Definition 4.1. The scheme F is *degenerate elliptic* if F^i is non-decreasing in each variable.

By taking advantage of this special structure, we can verify both stability and monotonicity of our finite difference scheme, as in [72].

Theorem 4.2 (Verifying Monotonicity and Stability)**.** *A scheme is monotone and nonexpansive in the L^∞ norm if and only if it is degenerate elliptic.*

Another property of certain finite difference schemes that is useful for constructing a convergence theory is the notion of a proper scheme.

Definition 4.2. The scheme F is *proper* if there exists $\kappa > 0$ such that for any $i = 1, \ldots, M$, $u_i - u_j|_{j \neq i}$ and $x_0, y_0 \in \mathbb{R}$ the inequality

$$x_0 \leq y_0$$

implies that the scheme satisfies

$$F^i(x_0, u_i - u_j|_{j \neq i}) - F^i(y_0, u_i - u_j|_{j \neq i}) \leq \kappa(x_0 - y_0).$$

These properties of approximation schemes are sufficient to guarantee the existence of a unique solution to a scheme, as proved in Theorem 8 of [72].

Theorem 4.3 (Uniqueness of Solutions)**.** *A proper, locally Lipschitz continuous, degenerate elliptic scheme has a unique solution.*

It is helpful to observe that any scheme $F^i(u_i, u_i - u_j|_{j \neq i})$ that is non-decreasing in its first argument can be made proper by replacing it with

$$\tilde{F}^i(u_i, u_i - u_j|_{j \neq i}) = F^i(u_i, u_i - u_j|_{j \neq i}) + \kappa u_i$$

where $\kappa > 0$ can be chosen to be smaller than the discretisation error of the scheme. This modification does not affect consistency, degenerate ellipticity, or Lipschitz continuity. Since the Monge-Ampère equations we consider in this thesis do not depend on the solution u (only on its Hessian and gradient), our schemes for the PDE will fall into this category. Consequently, we will not be concerned with building proper schemes in this thesis, since our schemes can easily be made proper without affecting the formal accuracy of the discretisation.

Given these general results, the work in proving that a (locally) Lipschitz continuous discretisation of (1.2) converges is reduced to verifying two conditions: consistency and degenerate ellipticity. This is accomplished in Lemmas 4.5-4.6

Remark (Convergence rates). While the formal accuracy of the scheme can be determined by Taylor series applied to smooth test functions, the theorem only guarantees uniform convergence to the viscosity solution. In general, the rate of convergence (accuracy) of the scheme may not agree with the convergence rates suggested by the formal analysis. This is to be expected since, in general, viscosity solutions can be singular, which means that Taylor series are not valid. The power of the convergence result is that it applies even in the singular case. In general, numerically observed convergence rates depend on both the regularity of the solution and the discretisation.

4.1.1 Wide Stencil Schemes

Even in the linear case, it is not always possible to build a monotone discretisation of a second-order elliptic equation using a narrow (9-point) finite difference stencil [69]. Instead, wide stencils are typically needed to build monotone discretisations of degenerate elliptic second order PDEs. This type of discretisation was introduced by Oberman to build convergent schemes for the equation for level set motion by mean curvature [70] and for the infinity Laplace equation [71]. In [74] wide stencils were used for the two dimensional Monge-Ampère equation. A wide stencil discretisation of the convex envelope was given in [73]. A study of consistent discretisations of Hamilton-Jacobi-Belman equations using wide stencil schemes has been performed in [9].

When we discretise an operator on a finite difference grid, we approximate second derivatives by centred finite differences (spatial discretisation). In addition, we can consider a finite number of possible directions ν that lie on the grid (directional discretisation). This allows us to discretise the second directional derivative in the direction ν by

$$\mathcal{D}_{\nu\nu}u_i = \frac{1}{(|\nu|\, h)^2}\left(u(x_i + \nu h) + u(x_i - \nu h) - 2u(x_i)\right). \tag{4.2}$$

Depending on the direction of the vector ν, this may involve a wide stencil.

As a concrete example, we can consider the direction $\nu = (1, 2)$ in $\mathbb{R}^2$. The second directional derivative in this direction is discretised as

$$\mathcal{D}_{\nu\nu}u_{i,j} = \frac{1}{5h^2}(u_{i+1,j+2} + u_{i-1,j-2} - 2u_{i,j}).$$

At points near the boundary of the domain, some values required by the wide stencil will not be available (Figure 4.1). In these cases, we can use intermediate boundary values, which may not lie on grid points, to construct a lower accuracy ($\mathcal{O}(h)$) stencil for the second directional derivative. For example, at the point $(i, N-1)$ on an $N \times N$ grid, the discretisation (4.2) of the second derivative in the direction $(1, 2)$ requires the point $(i+1, N+1)$, which lies outside the grid. In this case, we will discretise this directional derivative by

$$\mathcal{D}_{\nu\nu}u_{i,j} = \frac{4}{15h^2}(2u_{i+1/2,N} + u_{i-1,N-3} - 3u_{i,N-1}).$$

Since the value of $u_{i+1/2,N}$ is on the boundary, it can be regarded as data, which is either given or obtained by interpolation.

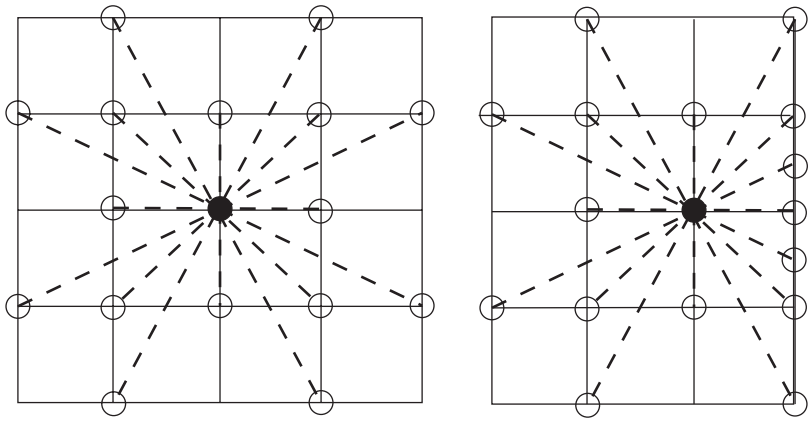


Figure 4.1: Wide stencils on a two dimensional grid (a) in the interior and (b) near the boundary.

4.1.2 Monotone Discretisation in Two Dimensions

In two dimensions, the largest and smallest eigenvalues of a symmetric matrix can be represented by the variational formula

$$\lambda_1[A] = \min_{\|\nu\|=1} \nu^T A \nu, \qquad \lambda_2[A] = \max_{\|\nu\|=1} \nu^T A \nu.$$

This formula was used in [74] to build monotone schemes for functions of the eigenvalues of the Hessian. This work includes the Monge-Ampère operator, which is the product of the eigenvalues of the Hessian. However, the above formulae do not generalise naturally to higher dimensions.

4.2 A Variational Characterisation of the Equation

In this chapter, we want to use the theory of [4, 72] to construct a convergent discretisation of the Monge-Ampère equation. To do this, we require a monotone discretisation of the equation. We recall that the two-dimensional Monge-Ampère operator can be written as

$$\det(D^2 u) = \frac{\partial^2 u}{\partial x^2}\frac{\partial^2 u}{\partial y^2} - \left(\frac{\partial^2 u}{\partial x \partial y}\right)^2.$$

Unfortunately, there is no obvious way to produce a monotone discretisation of the mixed derivatives. Of course, this situation does not get any easier in higher dimensions.

In order to proceed, we want to rewrite the Monge-Ampère operator in a way that better lends itself to a monotone discretisation. Given the difficulty in building a monotone discretisation of mixed derivatives, as well as our knowledge of how to construct monotone discretisations of second directional derivatives, we wish to find an alternative characterisation of the Monge-Ampère operator that does not involve mixed second derivatives.

4.2.1 A Variational Characterisation for Strictly Convex Solutions

We begin by establishing a matrix analysis result that will provide a characterisation of the determinant of the Hessian (that is, the Monge-Ampère operator) that leads to a natural discretisation in any spatial dimension.

Consider an arbitrary symmetric positive definite matrix, A. Then we can characterise the determinant of A as follows.

Theorem 4.4 (Variational Characterisation of the Determinant)**.** *Let A be a $d\times d$ symmetric positive definite matrix with eigenvalues λ_j and let V be the set of all orthonormal bases for $\mathbb{R}^d$:*

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

Then the determinant of A is equivalent to

$$\prod_{j=1}^{d} \lambda_j = \min_{(\nu_1,\dots,\nu_d)\in V} \prod_{j=1}^{d} \max\left\{\nu_j^T A \nu_j, 0\right\}.$$

Proof. Since A is symmetric and positive definite, we can find a set of d orthonormal eigenvectors v_j.

Any $(\nu_1, \dots, \nu_d) \in V$, can be expressed as a linear combination of the eigenvectors:

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

Since the ν_j and v_j are both orthonormal, we can make some claims about the coefficients c_{jk}.

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

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

We can use these results to compute

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

Using Jensen's inequality, we conclude that

$$\begin{aligned} \log \prod_{j=1}^{d} \nu_j^T A \nu_j &\geq \sum_{j=1}^{d} \sum_{k=1}^{d} c_{jk}^2 \log(\lambda_k) \\ &= \sum_{k=1}^{d} \left(\sum_{j=1}^{d} c_{jk}^2 \right) \log(\lambda_k) = \log \prod_{j=1}^{d} \lambda_j. \end{aligned}$$

Since the logarithmic function is increasing, we conclude that

$$\prod_{j=1}^{d} \nu_j^T A \nu_j \geq \prod_{j=1}^{d} \lambda_j$$

with equality if the ν_j are identical to the eigenvectors v_j. This implies that

$$\prod_{j=1}^{d} \lambda_j = \min_{(\nu_1, \ldots, \nu_d) \in V} \prod_{j=1}^{d} \nu_j^T A \nu_j.$$

Moreover, since A is positive definite, all of the $\nu_j^T A \nu_j$ are positive and we conclude that

$$\prod_{j=1}^{d} \lambda_j = \min_{(\nu_1, \ldots, \nu_d) \in V} \prod_{j=1}^{d} \max \left\{ \nu_j^T A \nu_j, 0 \right\}. \qquad \square$$

Theorem 4.4 allows us to characterise the determinant of the Hessian of a strictly convex C^2 function ϕ in terms of its second directional derivatives:

$$\det{}^+(D^2\phi) = \min_{(\nu_1, \ldots, \nu_d) \in V} \prod_{j=1}^{d} \left(\nu_j^T D^2 \phi \, \nu_j \right)^+ = \min_{(\nu_1, \ldots, \nu_d) \in V} \prod_{j=1}^{d} \max \left\{ \frac{\partial^2 \phi}{\partial {\nu_j}^2}, 0 \right\}.$$

Theorem 4.5 (Variational Form of the Monge-Ampère Equation)**.** *Let $f : X \to \mathbb{R}$ be a strictly positive function. A function $u \in C^2(X)$ is a strictly convex solution of the Monge-Ampère equation*

$$\det(D^2 u) = f$$

if and only if it satisfies the variational expression

$$\min_{\{\nu_1 \ldots \nu_d\} \in V} \prod_{j=1}^{d} \max \left\{ \frac{\partial^2 u}{\partial \nu_j^2}, 0 \right\} = f. \tag{4.3}$$

Proof. Suppose $u \in C^2(X)$ is a strictly convex solution of the Monge-Ampère equation. Then the Hessian D^2u is symmetric and positive definite. By Theorem 4.4, the Hessian satisfies

$$\min_{\{\nu_1 \ldots \nu_d\} \in V} \prod_{j=1}^{d} \max \left\{ \frac{\partial^2 u}{\partial \nu_j^2}, 0 \right\} = f,$$

as required.

Now we suppose that u satisfies this variational expression. If u is not strictly convex, then at least one of the second directional derivatives is negative or zero so that

$$\max \left\{ \frac{\partial^2 u}{\partial \nu_j^2}, 0 \right\} = 0$$

for some ν_j. Consequently, the variational expression will have the value 0, which cannot be equal to a positive right-hand side. We conclude that u must be strictly convex. Then the Hessian of u is a symmetric, positive definite matrix and by Theorem 4.4, u satisfies the Monge-Ampère equation. □

We want to stress that this variational formulation of the Monge-Ampère equation accomplishes two important tasks:

1. The mixed derivatives have been eliminated.

2. The convexity constraint has been absorbed into the equation.

4.2.2 A Variational Characterisation of Degenerate Equations

We have shown that for strictly convex solutions, the variational expression (4.3) is equivalent to the Monge-Ampère equation with the convexity constraint. However, in the degenerate case where the right-hand side f can vanish and solutions are no longer strictly convex, the variational expression may not uniquely determine the solution of the Monge-Ampère equation. This is because the variational equation with a vanishing right-hand side can also permit non-convex solutions. To remedy this problem, we modify our variational equation by adding a term that will penalise non-convex functions.

To begin, we need to verify that our variational characterisation of the determinant remains valid for matrices that are only positive semi-definite.

Lemma 4.1 (Determinant of a Positive Semi-Definite Matrix). Let A be a $d \times d$ symmetric positive semi-definite matrix and let V be the set of all orthonormal bases of $\mathbb{R}^d$:

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

Then the determinant of A is equivalent to

$$\det(A) = \min_{(\nu_1,\ldots,\nu_d)\in V} \prod_{j=1}^{d} \max\left\{\nu_j^T A \nu_j, 0\right\}.$$

Proof. If A is positive definite, this result follows immediately from Theorem 4.4.

Now we suppose that A has at least one eigenvalue that vanishes, so that the determinant of A also vanishes. Then the variational formula satisfies

$$\begin{aligned} 0 &\leq \min_{(\nu_1,\ldots,\nu_d)\in V} \prod_{j=1}^{d} \max\left\{\nu_j^T A \nu_j, 0\right\} \\ &\leq \prod_{j=1}^{d} \max\left\{v_j^T A v_j, 0\right\} \\ &= 0. \end{aligned}$$

In the above, the v_j are the eigenvectors of A.

We conclude that the variational expression will have the value zero, and it continues to be identical to the determinant. □

Next we propose incorporating an additional term into this expression, which will involve the negative part of the eigenvalues.

Lemma 4.2 (Determinant of a Positive Semi-Definite Matrix). Let A be a $d \times d$ symmetric positive semi-definite matrix, γ any positive real number, and V the set of all orthonormal bases of $\mathbb{R}^d$:

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

Then the determinant of A is equivalent to

$$\det(A) = \min_{(\nu_1,\ldots,\nu_d)\in V} \left\{ \prod_{j=1}^{d} \max\left\{\nu_j^T A \nu_j, 0\right\} + \gamma \sum_{j=1}^{d} \min\left\{\nu_j^T A \nu_j, 0\right\} \right\}.$$

Proof. Since A is positive semi-definite, $\nu_j^T A \nu_j$ will be non-negative for any choice of ν_j. This means that

$$\min\left\{\nu_j^T A \nu_j, 0\right\} = 0$$

for any choice of ν_j. Then the result follows immediately from Lemma 4.1. □

This result immediately gives us another formulation of the Monge-Ampère equation.

Lemma 4.3 (Monge-Ampère Operator for Convex Functions). If $u \in C^2(X)$ is convex and γ is any positive real number, the Monge-Ampère operator will be equal to

$$\det(D^2 u) = \min_{\{\nu_1 \dots \nu_d\} \in V} \left\{ \prod_{j=1}^{d} \max\{u_{\nu_j \nu_j}, 0\} + \gamma \sum_{j=1}^{d} \min\{u_{\nu_j \nu_j}, 0\} \right\}.$$

Proof. Since u is convex, its Hessian $D^2 u$ is positive semi-definite and the result follows immediately from Lemma 4.2. □

The important thing about this adjusted variational formulation is that the new term will serve to penalise non-convex functions, which will allow us to absorb the convexity constraint into the equation even when the right-hand side vanishes. This is made clear in the following lemma.

Lemma 4.4 (Convexity of Solutions). Let $u \in C^2(X)$ be a solution of the equation

$$\min_{\{\nu_1 \dots \nu_d\} \in V} \left\{ \prod_{j=1}^{d} \max\{u_{\nu_j \nu_j}, 0\} + \gamma \sum_{j=1}^{d} \min\{u_{\nu_j \nu_j}, 0\} \right\} = f \tag{4.4}$$

where γ is a positive real number and f is a non-negative function. Then u is convex.

Proof. Let $u \in C^2(X)$ be a non-convex function. Then at some point $x \in X$, and for some $v \in \mathbb{R}^d$, the second directional derivative in that direction is negative:

$$u_{vv} < 0.$$

This means that

$$\max\{u_{vv}, 0\} = 0, \quad \min\{u_{vv}, 0\} < 0.$$

Thus the variational equation will have a negative value at this point:

$$\begin{aligned}\min_{\{\nu_1 \dots \nu_d\} \in V} \left\{ \prod_{j=1}^{d} \max\{u_{\nu_j \nu_j}, 0\} + \gamma \sum_{j=1}^{d} \min\{u_{\nu_j \nu_j}, 0\} \right\} \\ \leq \gamma \min\{u_{vv}, 0\} \\ < 0.\end{aligned}$$

On the other hand, the right-hand side is non-negative ($f \geq 0$), so this non-convex function cannot satisfy our new variational equation (4.4). □

When we put these lemmas together, we find that the adjusted variational equation is equivalent to the original Monge-Ampère equation with the convexity constraint.

Theorem 4.6 (Variational Characterisation of the Monge-Ampère Equation)**.** *A function $u \in C^2(X)$ satisfies the Monge-Ampère equation* (1.2) *together with the convexity constraint* (1.1) *if and only if it satisfies the variational equation* (4.4)*.*

Proof. Let u be a convex solution of the Monge-Ampère equation (1.2). By Lemma 4.3, it also satisfies the variational equation (4.4).

Now we suppose that u is a solution of the variational equation (4.4). By Lemma 4.4, u must be a convex function. Then from Lemma 4.3, it is a convex solution of the Monge-Ampère equation (1.2). □

4.3 Monotone Discretisation

We now turn to the problem of constructing a monotone discretisation of the Monge-Ampère equation using our new formulation of the equation. One big advantage of this formulation is that the convexity constraint is built into the PDE. This means that we no longer have to concern ourselves with consistency with the convexity constraint; it is enough that our discretisation be consistent with the PDE (4.3).

4.3.1 Wide Stencil Discretisation

This formulation of the Monge-Ampère operator lends itself to the wide stencil discretisation described in §4.1.1 To implement this, we consider a finite number of possible directions ν

that lie on the grid. We denote this set of orthogonal vectors by $\mathcal{G}$. Then we can discretise the convexified Monge-Ampère operator as

$$-\min_{\{\nu_1 \dots \nu_d\} \in \mathcal{G}} \left\{ \prod_{j=1}^{d} \max\{\mathcal{D}_{\nu_j \nu_j} u, 0\} + \gamma \sum_{j=1}^{d} \min\{\mathcal{D}_{\nu_j \nu_j} u, 0\} \right\}.$$

where $\mathcal{D}_{\nu\nu}$ is the finite difference operator for the second directional derivative in the direction ν, which lies on the finite difference grid; see (4.2).

We note that this expression may not be uniformly elliptic if the (discrete) second directional derivatives vanish. Thus we choose to relax this expression slightly be introducing a small parameter $\delta \geq 0$ and instead defining our monotone discretisation as

$$MA_M[u] \equiv -\min_{\{\nu_1 \dots \nu_d\} \in \mathcal{G}} \left\{ \prod_{j=1}^{d} \max\{\mathcal{D}_{\nu_j \nu_j} u, \delta\} + \gamma \sum_{j=1}^{d} \min\{\mathcal{D}_{\nu_j \nu_j} u, \delta\} \right\}. \qquad (MA)^M$$

Since the discretisation considers only a finite number of directions ν, there will be an additional term in the consistency error coming from the angular resolution $d\theta$ of the stencil. This angular resolution will decrease and approach zero as the stencil width is increased. In practice, we use relatively narrow stencils for most computations, but for singular solutions, the directional resolution error can dominate.

An interesting question is whether this discretisation—in two dimensions—is equivalent to the wide stencil discretisation of the two-dimensional Monge-Ampère equation described in [74]. A simple example demonstrates that these two discretisations are genuinely distinct. For example, we can consider the function $u(x, y) = x^2 + y^2 + x^2 y^2$ and discretise the Monge-Ampère operator using a 9-point stencil. This allows us to choose from the set of directions

$$\{(1, 0),\ (0, 1),\ (1, 1),\ (1, -1)\}.$$

Using the two-dimensional characterisation of the Monge-Ampère equation (recalled in §4.1.2), the monotone discretisation produces

$$-\left(\min_{\nu_1} \mathcal{D}_{\nu_1 \nu_1} u\right)\left(\max_{\nu_2} \mathcal{D}_{\nu_2 \nu_2} u\right) = -2(2 + h^2).$$

On the other hand, our new discretisation has the value

$$-\min_{\nu_1 \perp \nu_2} \left\{ \mathcal{D}_{\nu_1 \nu_1} u\, \mathcal{D}_{\nu_2 \nu_2} u \right\} = -4.$$

4.3.2 Regularisation

The monotone discretisation we have described in $(MA)^M$ may not be differentiable at points where the minimum is attained along more than one direction ν, or at points where the value is zero. Since we wish to differentiate the operator when we build fast solvers using Newton's method, we need to regularise this discretisation. For convergence to viscosity solutions, we need to make the regularisation monotone.

One way to do this is to notice that the non-differentiability of $(MA)^M$ arises only from the operations of max and min. This means that if we regularise each of these operations in a monotone way, we can reconstruct a regularised version of $(MA)^M$ by substitution.

With that in mind, we define

$$\max^\delta(a,b) = \frac{1}{2}\left(a+b+\sqrt{(a-b)^2+\delta^2}\right)$$
$$\min^\delta(a,b) = \frac{1}{2}\left(a+b-\sqrt{(a-b)^2+\delta^2}\right).$$

Clearly $\max^\delta \to \max$ and $\min^\delta \to \min$ as $\delta \to 0$. Moreover, these functions are differentiable and non-decreasing in each variable. We also note that

$$\max^\delta(a,0) = \frac{1}{2}(a+\sqrt{a^2+\delta^2}) > \frac{1}{2}(a+|a|) \geq 0,$$

$$\min^\delta(a,0) = \frac{1}{2}(a-\sqrt{a^2+\delta^2}) < \frac{1}{2}(a-|a|) \leq 0.$$

Now we can build up the regularised operator as follows. We begin by replacing the approximations to the positive and negative parts of the second directional derivatives with regularised versions:

$$\max\{\mathcal{D}_{\nu\nu}u,\delta\} \to \max^\delta(\mathcal{D}_{\nu\nu},0), \quad \min\{\mathcal{D}_{\nu\nu}u,\delta\} \to \min^\delta(\mathcal{D}_{\nu\nu},0)$$

Next, the minimum over orthogonal vectors is regarded as a succession of minimums, each of which is replaced by its regularised version.

The resulting discretisation is denoted by

$$MA_R[u]. \tag{$(MA)^R$}$$

It is a smooth function of u_i, strictly increasing in each of the $\mathcal{D}_{\nu_j^k\nu_j^k}u_i$, and converges to the original discretisation $(MA)^M$ as $\delta \to 0$.

4.4 Convergence to the Viscosity Solution

Theorem 4.7 (Convergence to Viscosity Solution)**.** *Let the PDE* (4.1) *have a unique viscosity solution. Then the solutions of the schemes* $(MA)^M$, $(MA)^R$ *converge to the viscosity solution of* (1.2) *as* $h, d\theta, \delta \to 0$.

Proof. The convergence follows from verifying consistency and degenerate ellipticity, as discussed in §4.1. This is accomplished in Lemmas 4.5-4.6. □

4.4.1 Degenerate Ellipticity

We recall that according to Definition 4.1, a finite difference equation of the form

$$F^i[u] = F^i(u_i, u_i - u_j|_{j \neq i}).$$

is degenerate elliptic if $F^i$is non-decreasing in each variable.

Lemma 4.5 (Degenerate Ellipticity)*.* The finite difference schemes given by $(MA)^M$ and $(MA)^R$ are degenerate elliptic.

Proof. From their definitions, the discrete second directional derivatives $\mathcal{D}_{\nu\nu}u$ are non-decreasing functions of each $u_j - u_i$ for each grid direction ν. Ignoring the minus sign in front of it, the scheme $(MA)^M$ is a non-decreasing combination of the operators min and max applied to the non-decreasing terms $\mathcal{D}_{\nu\nu}u$, so it is also non-decreasing in each of the $u_j - u_i$.

Replacing the minus sign in front of the scheme, we find that $(MA)^M$ is non-decreasing in each of the $u_i - u_j$ and is thus degenerate elliptic.

We recall from the construction of the scheme in §4.3.2 that the regularised scheme $(MA)^R$ comes from replacing the operations of min and max in $(MA)^M$ by a non-decreasing regularisation of these operations. So the regularised scheme is also degenerate elliptic. □

4.4.2 Consistency

We also require the schemes $(MA)^R$ and $(MA)^M$ to be consistent with the Monge-Ampère equation.

Definition 4.3. The scheme $MA^{h,d\theta,\delta}$ is consistent with the equation (1.2) at x_0 if for every twice continuously differentiable function $\phi(x)$ defined in a neighbourhood of x_0,

$$MA^{h,d\theta,\delta}[\phi](x_0) \to MA[\phi](x_0) \text{ as } h, d\theta, \delta \to 0.$$

The global scheme defined on X is consistent if this limit holds uniformly for all $x \in X$.

Before we prove the consistency of our scheme, we recall that we have used h to denote the spatial resolution of our grid. However, because we are discretising the second directional derivatives using a wide stencil, the effective spatial resolution will be larger. For example, the discrete version of the second derivative in the direction ν_j will be

$$\mathcal{D}_{\nu_h \nu_j} u = u_{\nu_j \nu_j} + \mathcal{O}(\left|\nu_j\right|^2 h^2) = u_{\nu_j \nu_j} + \mathcal{O}(h_j^2).$$

We will denote the effective spatial resolution of our stencil by

$$h_{eff} \equiv \max_{\nu_j \in \mathcal{G}} h_j.$$

As we refine the grid, it is important not only that h approaches zero, but also that h_{eff} approaches zero. Since h_{eff} is related to the stencil width, which in turn determines the angular resolution $d\theta$ of the stencil, this means that h should be converging to zero faster than $d\theta$ converges to zero.

Now we prove consistency of $(MA)^M$ and $(MA)^R$. The consistency proofs are identical since

$$\max^\delta\{a, 0\} = \max\{a, 0\} + \mathcal{O}(\delta) = \max\{a, \delta\} + \mathcal{O}(\delta).$$

Lemma 4.6. Let $x_0 \in X$ be a reference point on the grid and $\phi(x)$ be a twice continuously differentiable function that is defined in a neighbourhood of the grid. Then the schemes $MA_M[\phi]$ and $MA_R[\phi]$ defined in $(MA)^M$ and $(MA)^R$ approximate the PDE $MA[\phi]$ with accuracy

$$MA_{M,R}[\phi](x_0) = MA[\phi](x_0) + \mathcal{O}(h_{eff}^2 + d\theta + \delta).$$

Proof. From a simple Taylor series computation we have

$$\mathcal{D}_{\nu\nu}\phi(x_0) = \phi_{\nu\nu}(x_0) + \mathcal{O}(h_{eff}^2).$$

We also recall that in subsection 4.3.2 we regularised the second directional derivatives to obtain

$$\max^\delta\{\mathcal{D}_{\nu\nu}\phi(x_0), 0\} = \max\{\mathcal{D}_{\nu\nu}\phi(x_0), 0\} + \mathcal{O}(\delta),$$

$$\min^\delta\{\mathcal{D}_{\nu\nu}\phi(x_0), 0\} = \min\{\mathcal{D}_{\nu\nu}\phi(x_0), 0\} + \mathcal{O}(\delta).$$

We know that the negation of the Monge-Ampère operator can be expressed as

$$\min_{\nu\in V}\left\{\prod_{j=1}^{d}\max\{u_{\nu_j\nu_j},0\}+\gamma\sum_{j=1}^{d}\min\{u_{\nu_j\nu_j},0\}\right\}=\prod_{j=1}^{d}u_{v_jv_j}+\gamma\sum_{j=1}^{d}\min\{u_{v_jv_j},0\}$$

where the v_j are orthogonal unit vectors, which may not be in the set of grid vectors $\mathcal{G}$. We can then choose a set of vectors

$$\frac{v+dv}{|v+dv|}\in\mathcal{G}$$

so that each remainder $|dv_j|=\mathcal{O}(d\theta)$.

Now we consider the discretised problem

$$\begin{aligned}
-MA^{h,d\theta,\delta}_{M,R}[\phi](x_0)&=\min_{\nu\in\mathcal{G}}{}^{\delta}\left\{\prod_{j=1}^{d}\max{}^{\delta}\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}+\gamma\sum_{j=1}^{d}\min{}^{\delta}\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}\right\}\\
&=\min_{\nu\in\mathcal{G}}\left\{\prod_{j=1}^{d}\max\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}+\gamma\sum_{j=1}^{d}\min\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}\right\}+\mathcal{O}(\delta)\\
&\le\prod_{j=1}^{d}\max\{\mathcal{D}_{(v_j+dv_j)(v_j+dv_j)}\phi(x_0),0\}\\
&\quad+\gamma\sum_{j=1}^{d}\min\{\mathcal{D}_{(v_j+dv_j)(v_j+dv_j)}\phi(x_0),0\}+\mathcal{O}(\delta)\\
&=\prod_{j=1}^{d}\max\left\{\frac{(v_j+dv_j)^TD^2\phi(x_0)(v_j+dv_j)}{|v_j+dv_j|^2},0\right\}\\
&\quad+\gamma\sum_{j=1}^{d}\min\left\{\frac{(v_j+dv_j)^TD^2\phi(x_0)(v_j+dv_j)}{|v_j+dv_j|^2},0\right\}+\mathcal{O}(h_{eff}^2+\delta)\\
&=\prod_{j=1}^{d}\max\{v_j^TD^2\phi(x_0)v_j,0\}+\gamma\sum_{j=1}^{d}\min\{v_j^TD^2\phi(x_0)v_j,0\}\\
&\quad+\mathcal{O}(h_{eff}^2+d\theta+\delta)\\
&=\min_{\nu\in V}\left\{\prod_{j=1}^{d}\max\{\phi_{\nu_j\nu_j}(x_0),0\}+\gamma\sum_{j=1}^{d}\min\{\phi_{\nu_j\nu_j}(x_0),0\}\right\}\\
&\quad+\mathcal{O}(h_{eff}^2+d\theta+\delta)\\
&=-MA[\phi](x_0)+\mathcal{O}(h_{eff}^2+d\theta+\delta).
\end{aligned}$$

In addition, since the set of grid vectors $\mathcal{G}$ is a subset of the set of all orthogonal vectors V, we find that

$$\begin{aligned}
-MA^{h,d\theta,\delta}_{M,R}[\phi](x_0) &= \min_{\nu\in\mathcal{G}}{}^{\delta}\left\{\prod_{j=1}^{d}\max{}^{\delta}\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}+\gamma\sum_{j=1}^{d}\min{}^{\delta}\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}\right\}\\
&= \min_{\nu\in\mathcal{G}}\left\{\prod_{j=1}^{d}\max\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}+\gamma\sum_{j=1}^{d}\min\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}\right\}+\mathcal{O}(\delta)\\
&\geq \min_{\nu\in V}\left\{\prod_{j=1}^{d}\max\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}+\gamma\sum_{j=1}^{d}\min\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0),0\}\right\}+\mathcal{O}(\delta)\\
&= \min_{\nu\in V}\left\{\prod_{j=1}^{d}\max\{\phi_{\nu_j\nu_j}(x_0),0\}+\gamma\sum_{j=1}^{d}\min\{\phi_{\nu_j\nu_j}(x_0),0\}\right\}+\mathcal{O}(h^2_{eff}+\delta)\\
&= -MA[\phi](x_0)+\mathcal{O}(h^2_{eff}+\delta).
\end{aligned}$$

We conclude that

$$MA^{h,d\theta,\delta}_{M,R}[\phi](x_0) = MA[\phi](x_0)+\mathcal{O}(h^2_{eff}+d\theta+\delta).$$

Thus the schemes are consistent. □

4.5 Forward Euler for the Parabolic Equation

Having described a convergent discretisation of the Monge-Ampère equation, we now need to provide a method for solving the discrete system.

Using a monotone discretisation $-F[u]$ of the Monge-Ampère operator, the simplest way to solve the Monge-Ampère equation is by solving the parabolic version of the equation using forward Euler. That is, we perform the iteration

$$u^{n+1} = u^n + dt(F[u^n]-f)$$

until the solution reaches a steady state.

Explicit iterative methods have the advantage of being simple to implement. However, stability requires the stepsize dt to satisfy a CFL condition (which applies in a nonlinear form to monotone discretisations, as explained in [72]). Because of the small size of dt, which depends on the spatial resolution h, approximating the steady state solution requires

a large number of iterations. In particular, the required time step is given by the inverse of the Lipschitz constant for the scheme

$$dt = K(F[u^n])^{-1}.$$

For example, we consider the (unregularised) scheme $(MA)^M$

$$MA^M[u] \equiv -\min_{\{\nu_1 \dots \nu_d\} \in \mathcal{G}} \left\{ \prod_{j=1}^{d} \max\{\mathcal{D}_{\nu_j \nu_j} u, \delta\} + \gamma \sum_{j=1}^{d} \min\{\mathcal{D}_{\nu_j \nu_j} u, \delta\} \right\}.$$

We recall that the Lipschitz constant of the maximum or minimum of two functions f_1, f_2 is bounded by the maximum of the Lipschitz constants K_1, K_2

$$K(\max(f_1, f_2)),\ K(\min(f_1, f_2)) \leq \max(K_1, K_2)$$

and the Lipschitz constant of the sum of two functions is bounded by the sum of the Lipschitz constants

$$K(f_1 + f_2) \leq K_1 + K_2.$$

Using these properties and the chain rule, we can bound the Lipschitz constant of the monotone scheme by

$$K(MA^M[u]) \leq \frac{2}{h^2} \max_{\{\nu_1 \dots \nu_d\} \in \mathcal{G}} \left\{ \sum_{i=1}^{d} \prod_{j \neq i} \max\{\mathcal{D}_{\nu_j \nu_j} u, \delta\} + d\gamma \right\},$$

which implies that the optimal time step is $\mathcal{O}(h^2)$ and may become very small if the eigenvalues of the Hessian are large.

Although this time step is an improvement over the one obtained by dimensional scaling (which is $\mathcal{O}(h^{2d})$), it still places a severe restriction on the solution speed that is possible using an explicit forward Euler iteration. Consequently, we now turn our attention to the construction of an implicit solution method that will allow for much faster solution times.

4.6 Newton's Method

We now consider the use of Newton's method for solving the system of equations we obtain when we discretise the Monge-Ampère equation. Although Newton's method can fail if the Monge-Ampère equation is discretised naively, the use of a monotone discretisation ensures that the Newton step will remain well-defined and that the iteration will converge.

Again, we use use the Newton iteration

$$u^{n+1} = u^n - \tau v^n$$

where the corrector v^n must solve the linear system

$$\left(\nabla_u MA[u^n]\right) v^n = MA[u^n] + f.$$

4.6.1 Monotone Discretisation

The Jacobian for the monotone discretisation is obtained by using Danskin's Theorem [7] and the product rule.

$$\nabla_u MA^M[u] = -\sum_{j=1}^{d} \operatorname{diag}\left(\mathbb{1}_{\mathcal{D}_{\nu_j^*\nu_j^*}u > \delta} \prod_{k \neq j} \max\{\mathcal{D}_{\nu_k^*\nu_k^*}u, \delta\} + \gamma \mathbb{1}_{\mathcal{D}_{\nu_j^*\nu_j^*}u \leq \delta} \right) \mathcal{D}_{\nu_j^*\nu_j^*}$$

where the ν_j^* are the directions active in the minimum in $(MA)^M$.

In order to ensure that the linear equation (3.3) is well-posed, we want the coefficients of each $\mathcal{D}_{\nu_j^*\nu_j^*}$ in the Jacobian to be negative. This requirement shows an additional advantage we obtain from the addition of the linear terms that penalise convexity (see §4.2.2). It is evident that without this correction to the PDE (the case $\gamma = 0$), the Jacobian can be singular if the (discrete) second directional derivatives of u vanish. However, the addition of the extra penalty term ensures that this cannot happen. In fact, this correction to the equation ensures that the linear system is well-posed even if u at the current iterate is non-convex.

4.6.2 Regularised Discretisation

The monotone discretisation described above still faces a subtle limitation in that the formulation of Newton's method $(MA)^M$ may not be differentiable at points where the minimum is attained along more than one direction ν. This was the motivation for the regularised discretisation given by $(MA)^R$. As this discretisation is differentiable, we can easily compute the Jacobian and apply Newton's method. We can also use the analysis we have done for this discretisation in §4.4 to prove convergence of Newton's method.

Theorem 4.8 (Newton's Method for the discretised Monge-Ampère Equation)**.** *Suppose the PDE* (1.2) *has a unique viscosity solution. Then Newton's method for the discretised system given by* $(MA)^R$ *converges quadratically.*

In order to prove this result, we recall a theorem on the convergence of Newton's method for a system of equations [57].

Theorem 4.9 (Newton's Method for a System of Equations)**.** *Consider a system of equations $F[u] = 0$ where the operator $F : \mathbb{R}^d \to \mathbb{R}^d$ and let $U \subset \mathbb{R}^d$ be open. Suppose the following conditions hold:*

1. *A solution $u^* \in U$ exists.*
2. *$\nabla F : U \to \mathbb{R}^{N \times N}$ is Lipschitz continuous.*
3. *$\nabla F(u^*)$ is non-singular.*

Then the Newton iteration

$$u^{n+1} = u^n - \nabla F(u^n)^{-1} F(u^n)$$

converges quadratically to u^* if $u^0 \in U$ is sufficiently close to u^*.

Remark. In order to apply this result to our non-linear system, we rely on the fact that our discretisation is degenerate elliptic. This is necessary to ensure both that a solution to the system exists and that the Jacobian ∇F in the Newton iteration is non-singular. This general theorem about Newton's method will not necessarily apply to other discretisations such as the one described in Chapter 3.

Proof of Theorem 4.8. For any fixed grid, the discretised system of equations has a solution, as established in Theorem 4.7.

The scheme $(MA)^R$ is smooth in u and is thus locally Lipschitz continuous.

By construction, the discrete Monge-Ampère operator is strictly decreasing in each of the discrete second directional derivatives (§4.3.2). Thus the Jacobian will have the form

$$\nabla_u MA^\delta[u] = -\sum_{\nu^k \in \mathcal{G}} \sum_{j=1}^{d} A_{jk}(u) \mathcal{D}_{\nu_j^k \nu_j^k}$$

where each of the $A_{jk}(u)$ is a positive definite diagonal matrix. The Jacobian is positive definite and thus invertible.

By Theorem 4.9, Newton's method converges for the discretised system $(MA)^R$. □

4.7 Numerical Implementation

Before we provide computational results, we discuss several additional details of the computational results.

4.7.1 Damping

We will use the form of Newton's method in §4.6 with the addition of damping,

$$u^{n+1} = u^n - \tau v^n,$$

to solve the discretised equation coming from the monotone discretisation of §4.3.1. Here the damping parameter τ, $0 < \tau \leq 1$, is chosen at each step to ensure that the residual $\|MA^M(u^n) + f\|$ is decreasing. In most cases, we can simply choose $\tau = 1$. However, damping can improve convergence if we choose a poor initial value.

4.7.2 Initialisation

Newton's method requires a good initialisation in order for convergence to be guaranteed. In particular, we desire an initial value that:

- is close to the exact solution
- respects the boundary conditions.

A function that is convex may also be desirable, but this is no longer required for the monotone scheme we have described. This is because we have built convexity directly into the PDE and ensured that Newton's method remains well-posed even if the initial guess is non-convex. Thus we do not have to be overly concerned about forcing our initial guess to be convex.

In order to find a suitable initial value u^0, we suggest using one step of the semi-implicit scheme (3.13). This amounts to solving the Poisson equation (3.14)

$$\Delta u^0 = (d!f)^{1/d}$$

along with the specified Dirichlet boundary conditions.

If the solution of the Monge-Ampère equation is sufficiently regular, we may also accelerate the convergence of Newton's method by first solving the equation on a coarse grid and then interpolating onto the finer grid. This can result in a very accurate initial guess, leading to rapid convergence of Newton's method.

4.8 Extensions to Other Monge-Ampère Equations

Up to this point, all the theory developed in this chapter was applicable only to Monge-Ampère equations of the form

$$\det(D^2u(x)) = f(x),$$

where the right-hand side has no dependence on the solution u. However, in many applications the right-hand side can depend on the gradient of the solution:

$$\det(D^2u(x)) = F(x, \nabla u(x)).$$

We now describe how our monotone finite difference methods can be extended to allow for the numerical solution of this more general equation.

4.8.1 Discretisation of Functions of the Gradient

The main point we need to address here is the discretisation of functions of the gradient. The simplest approach would be to simply used standard centred differences for the first derivatives:

$$\mathcal{D}_{x_j}u(\mathbf{x}) = \frac{1}{2h}(u(\mathbf{x} + h\mathbf{e}_j) - u(\mathbf{x} - h\mathbf{e}_j))$$

where $\mathbf{e}_j$ is the vector whose i^{th} component is equal to the Kronecker delta δ_{ij}. While this discretisation is consistent with C^2 solutions of the Monge-Ampère equation, it is not monotone and there is no guarantee that it will converge to the viscosity solution.

Oberman [72] provided some examples illustrating the construction of monotone discretisations for functions of the gradient. For example, that work describes a monotone discretisation of the absolute value of a first derivative:

$$\left|u_x(x_j)\right| = \frac{1}{h}\max\{u(x_j) - u(x_{j-1}), u(x_{j+1}) - u(x_j), 0\} + \mathcal{O}(h).$$

For more general functions of the gradient, one approach to producing a monotone discretisation is to simply using centred differences and add on a small multiple of the laplacian:

$$g(u_x) = g(\mathcal{D}_x u) + hK_g\mathcal{D}_{xx}u + \mathcal{O}(h).$$

Here K_g is the Lipschitz constant of the function g.

However, instead of adding an additional term to the discretised equation, we could instead make use of the second derivatives that are already present in the Monge-Ampère

equation. In the case where solutions are smooth and strictly convex, this will also allow an improvement in the formal accuracy of the finite difference scheme. This is the subject of the following sections.

4.8.2 Discretisation of the Monge-Ampère equation

So far we have attempted to produce a monotone discretisation for each individual term in the Monge-Ampère equation. As an alternative to this, we suggest using a wide stencil to produce a discretisation of the Monge-Ampère equation which, though it may not be monotone for each of the individual terms, is monotone when considered as a whole.

To accomplish this, we make use of the second directional derivatives $u_{\nu_j\nu_j}$ that are already present in the Monge-Ampère equation, as noted in §4.8.1. By making a change of coordinates, we can write the gradient

$$\nabla u = \left(u_{x_1}, \ldots, u_{x_d}\right)$$

in terms of first derivatives in the directions ν_j:

$$\tilde{\nabla} u = \left(u_{\nu_1}, \ldots, u_{\nu_d}\right).$$

To accomplish all this, we first need to rewrite the gradient in terms of the new coordinate system. We consider any set of d orthogonal vectors in $\mathbb{R}^d$: $(v_1, \ldots, v_d)$. Now we can rewrite the gradient of a function u in terms of directional derivatives along these axes:

$$\nabla u = \left(u_{x_1}, \ldots, u_{x_d}\right) = \left(\sum_{j=1}^{d} \frac{v_j \cdot \mathbf{e}_1}{|v_j|} u_{v_j}, \ldots, \sum_{j=1}^{d} \frac{v_j \cdot \mathbf{e}_d}{|v_j|} u_{v_j}\right).$$

This enables us to discretise the gradient using a wide stencil by discretising the directional derivative in the direction v_j as

$$\mathcal{D}_{v_j} u_i = \frac{1}{2|v_j|\, h} \left(u(x_i + v_j h) - u(x_i - v_j h)\right), \tag{4.5}$$

which has an accuracy of $\mathcal{O}(h^2)$. Near the boundary, where some of the required values may not be available, we can simply use a first-order accurate forward or backward difference. We stress again that this discretisation of the gradient is valid for *any* set of orthogonal vectors $v_1, \ldots, v_d$.

Using this characterisation of the gradient, we can rewrite the Monge-Ampère equation as

$$\begin{aligned}
MA[u] &= -\min_{(\nu_1,\dots,\nu_d)\in V}\left\{\prod_{j=1}^{d}\max\{u_{\nu_j\nu_j},0\}+\gamma\sum_{j=1}^{d}\min\left\{u_{\nu_j\nu_j},0\right\}\right\}+F(x,\nabla u)\\
&= -\min_{(\nu_1,\dots,\nu_d)\in V}\left\{\prod_{j=1}^{d}\max\{u_{\nu_j\nu_j},0\}+\gamma\sum_{j=1}^{d}\min\left\{u_{\nu_j\nu_j},0\right\}-F(x,\nabla u)\right\}\\
&= -\min_{(\nu_1,\dots,\nu_d)\in V}\left\{\prod_{j=1}^{d}\max\{u_{\nu_j\nu_j},0\}+\gamma\sum_{j=1}^{d}\min\left\{u_{\nu_j\nu_j},0\right\}\right.\\
&\qquad\left.-F\left(x,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_1}{|\nu_j|}u_{\nu_j},\dots,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_d}{|\nu_j|}u_{\nu_j}\right)\right\}\\
&= -\min_{(\nu_1,\dots,\nu_d)\in V}G_{(\nu_1,\dots,\nu_d)}.
\end{aligned}$$

As we have already described in (4.2),(4.5), the directional first and second derivatives can be discretised using a wide stencil by limiting the set of possible directions in the set V to a finite set $\mathcal{G}$ of orthogonal vectors that lie on the grid. As before, we introduce a small parameter $\delta>0$ in order to bound the maximum and minimum functions away from zero:

$$\max\{\cdot,0\},\min\{\cdot,0\}\to\max\{\cdot,\delta\},\min\{\cdot,\delta\}.$$

We can now define the discretisation of the Monge-Ampère equation as

$$MA_M^{h,d\theta,\delta}[u]=-\min_{(\nu_1,\dots,\nu_d)\in\mathcal{G}}G^{h,d\theta,\delta}_{(\nu_1,\dots,\nu_d)}[u] \tag{4.6}$$

where each of the $G^{h,d\theta,\delta}_{(\nu_1,\dots,\nu_d)}[u]$ is defined as

$$\begin{aligned}
G^{h,d\theta,\delta}_{(\nu_1,\dots,\nu_d)}[u] =&\prod_{j=1}^{d}\max\{\mathcal{D}_{\nu_j\nu_j}u,\delta\}+\gamma\sum_{j=1}^{d}\min\{\mathcal{D}_{\nu_j\nu_j}u,\delta\}-\\
&F\left(x,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_1}{|\nu_j|}\mathcal{D}_{\nu_j}u,\dots,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_d}{|\nu_j|}\mathcal{D}_{\nu_j}u\right).
\end{aligned} \tag{4.7}$$

4.8.3 Convergence

Theorem 4.10 (Convergence to Viscosity Solution)**.** *Let the PDE* (1.3) *have a unique viscosity solution and let the right-hand side $F(x,\nabla u)$ be Lipschitz continuous on $\overline{\Omega}\times\mathbb{R}^d$*

with Lipschitz constant K_F. Then the solution of the scheme (4.6) *converges to the viscosity solution of* (1.2) *as $h, d\theta, \delta \to 0$ with $\gamma \geq \delta^{d-1} \geq K_F|\nu_j|\, h/2$ and $h_{eff} \geq h|\nu_j| \to 0$ for every $\nu_j \in \mathcal{G}$.*

Proof. The convergence follows from verifying consistency and degenerate ellipticity. This is accomplished in Lemmas 4.8-4.9. □

Lemma 4.7. Under the hypotheses of Theorem 4.10, the scheme for $G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}[u]$ in (4.7) is non-decreasing in each $u_j - u_i$.

Proof. We introduce the notation

$$p_j^+(x_i) = u(x_i + h\nu_j) - u(x_i), \quad p_j^-(x_i) = u(x_i - h\nu_j) - u(x_i).$$

This allows us to write $G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}[u]$ in the form of Definition 4.1 as follows:

$$\begin{aligned} G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}(p_1^+, p_1^-, \ldots, p_d^+, p_d^-) = &\prod_{j=1}^{d} \max\left\{ \frac{p_j^+ + p_j^-}{|\nu_j|^2 h^2}, \delta \right\} \\ &+ \gamma \sum_{j=1}^{d} \min\left\{ \frac{p_j^+ + p_j^-}{|\nu_j|^2 h^2}, \delta \right\} - F\left(\frac{p_1^+ - p_1^-}{2|\nu_1|\, h}, \ldots, \frac{p_d^+ - p_d^-}{|2\nu_d|\, h} \right). \end{aligned} \tag{4.8}$$

Now we need only check that this is non-decreasing in each of its arguments. We verify this for the term p_1^+; the reasoning is identical for the remaining terms.

Choose any $\epsilon > 0$ and consider:

$$\begin{aligned} &G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}(p_1^+ + \epsilon) - G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}(p_1^+) \\ &\quad \geq \delta^{d-1} \left(\max\left\{ \frac{p_1^+ + \epsilon + p_1^-}{|\nu_1|^2 h^2}, \delta \right\} - \max\left\{ \frac{p_1^+ + p_1^-}{|\nu_1|^2 h^2}, \delta \right\} \right) \\ &\qquad + \delta^{d-1} \left(\min\left\{ \frac{p_1^+ + \epsilon + p_1^-}{|\nu_1|^2 h^2}, \delta \right\} - \min\left\{ \frac{p_1^+ + p_1^-}{|\nu_1|^2 h^2}, \delta \right\} \right) \\ &\qquad - K_F \left(\frac{p_1^+ + \epsilon - p_1^-}{2|\nu_1|\, h} - \frac{p_1^+ - p_1^-}{2|\nu_1|\, h} \right). \end{aligned}$$

In the above, we have used the facts that

$$\min\left\{ \frac{p_1^+ + \epsilon + p_1^-}{|\nu_1|^2 h^2}, \delta \right\} - \min\left\{ \frac{p_1^+ + p_1^-}{|\nu_1|^2 h^2}, \delta \right\} \geq 0$$

and that $\gamma \geq \delta^{d-1}$.

We continue with this expression to conclude that

$$\begin{aligned} G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}(p_1^+ + \epsilon) - G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}(p_1^+) \\ &\geq \delta^{d-1}\left(\frac{p_1^+ + \epsilon + p_1^-}{|\nu_1|^2 h^2} + \delta - \frac{p_1^+ + p_1^-}{|\nu_1|^2 h^2} - \delta\right) - K_F \frac{\epsilon}{2|\nu_1| h} \\ &= \frac{\epsilon}{|\nu_1|^2 h^2}(\delta^{d-1} - K_F|\nu_1| h/2). \end{aligned}$$

This expression is non-negative as long as $\delta^{d-1} \geq K_F|\nu_1| h/2$.

We conclude that each of the $G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}$ is non-decreasing in each $u_j - u_i$. □

Lemma 4.8. Under the hypotheses of Theorem 4.10, the scheme for $MA^{h,d\theta,\delta}_M[u]$ in (4.6) is degenerate elliptic.

Proof. The negation of this scheme is the minimum of schemes that are non-increasing in each of the $u_i - u_j$. Consequently, the scheme for $MA^{h,d\theta,\delta}_M[u]$ is non-decreasing in each argument and is therefore degenerate elliptic. □

Lemma 4.9. Let $x_0 \in X$ be a reference point on the grid and $\phi(x)$ be a twice continuously differentiable function that is defined and convex in a neighbourhood of the grid. Then the scheme $MA_M[\phi]$ defined in 4.6 approximates the Monge-Ampère equation (1.3) with accuracy

$$MA_M[\phi](x_0) = MA[\phi](x_0) + \mathcal{O}\left(\min_j\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0)\}\delta + \gamma\delta + h_{eff}^2 + d\theta\right).$$

Proof. We recall that the discretisation is of the form

$$MA^{h,d\theta,\delta}_M[\phi](x_0) = \min_{(\nu_1,\ldots,\nu_d)\in\mathcal{G}} G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}[\phi](x_0).$$

We begin by considering the term inside the minimum.

$$\begin{aligned} G^{h,d\theta,\delta}_{(\nu_1,\ldots,\nu_d)}[\phi](x_0) = \prod_{j=1}^d \max\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0), \delta\} + \gamma\sum_{j=1}^d \min\{\mathcal{D}_{\nu_j\nu_j}\phi(x_0), \delta\} \\ - F\left(x, \sum_{j=1}^d \frac{\nu_j\cdot\mathbf{e}_1}{|\nu_j|}\mathcal{D}_{\nu_j}\phi(x_0), \ldots, \sum_{j=1}^d \frac{\nu_j\cdot\mathbf{e}_d}{|\nu_j|}\mathcal{D}_{\nu_j}\phi(x_0)\right) \end{aligned}$$

$$
\begin{aligned}
&= \prod_{j=1}^{d} \max\{\phi_{\nu_j\nu_j}(x_0), \delta\} + \gamma \sum_{j=1}^{d} \min\{\phi_{\nu_j\nu_j}(x_0), \delta\} \\
&\quad - F\left(x, \sum_{j=1}^{d} \frac{\nu_j \cdot \mathbf{e}_1}{|\nu_j|}\phi_{\nu_j}(x_0), \ldots, \sum_{j=1}^{d} \frac{\nu_j \cdot \mathbf{e}_d}{|\nu_j|}\phi_{\nu_j}(x_0)\right) + \mathcal{O}(h_{eff}^2) \\
&= \prod_{j=1}^{d} \max\{\phi_{\nu_j\nu_j}(x_0), 0\} + \mathcal{O}\left(\delta \sum_{j=1}^{d} \mathbb{1}_{\phi_{\nu_j\nu_j}(x_0)<\delta}\right) \\
&\quad + \gamma \sum_{j=1}^{d} \min\{\phi_{\nu_j\nu_j}(x_0), 0\} + \mathcal{O}(\gamma\delta) \\
&\quad - F\left(x, \sum_{j=1}^{d} \frac{\nu_j \cdot \mathbf{e}_1}{|\nu_j|}\phi_{\nu_j}(x_0), \ldots, \sum_{j=1}^{d} \frac{\nu_j \cdot \mathbf{e}_d}{|\nu_j|}\phi_{\nu_j}(x_0)\right) + \mathcal{O}(h_{eff}^2).
\end{aligned}
$$

Here we have made use of the fact that the centred difference discretisations of the first and second derivatives have a formal accuracy of $\mathcal{O}(h_{eff}^2)$ and that the function F is Lipschitz continuous.

Using the reasoning of Lemma 4.6, we conclude that

$$
MA_M^{h,d\theta,\delta}[\phi](x_0) = MA[\phi](x_0) + \mathcal{O}\left(\delta \sum_{j=1}^{d} \mathbb{1}_{\phi_{\nu_j\nu_j}(x_0)<\delta} + \gamma\delta + h_{eff}^2 + d\theta\right).
$$

Thus

$$
M_M^{h,d\theta,\delta}[\phi](x_0) \to MA[\phi](x_0)
$$

as $h, d\theta, \delta \to 0$ such that $h_{eff} \to 0$ as well.

Therefore the scheme is consistent. □

Remark. As before, we could also replace the max and min functions with the smooth functions

$$
\begin{aligned}
\max^{\delta}(a, b) &= \frac{1}{2}\left(a + b + \sqrt{(a-b)^2 + \delta^2}\right) \\
\min^{\delta}(a, b) &= \frac{1}{2}\left(a + b - \sqrt{(a-b)^2 + \delta^2}\right).
\end{aligned}
$$

Since these regularised functions preserve the property that

$$
\max^{\delta}(a, b) + \min^{\delta}(a, b) = a + b,
$$

the convergence proof (Theorem 4.10) is unchanged.

4.8.4 Formal Accuracy

Although the convergence proof does not provide an error estimate, it is interesting to look at the formal accuracy of the finite difference scheme, which suggests the accuracy we might expect to observe on smooth enough examples. The formal error of this scheme is on the order of

$$\delta \sum_{j=1}^{d} \mathbb{1}_{u_{\nu_j \nu_j} < \delta} + \gamma\delta + h_{eff}^2 + d\theta$$

when u is a smooth solution.

We recall also that for convergence, the parameters in the Monge-Ampère equation and the difference scheme must satisfy

$$\gamma \geq \delta^{d-1} \geq \frac{1}{2} K_F h_{eff}.$$

Although γ occurs in the PDE itself, not merely the approximation scheme, the PDE (4.3) is equivalent to the Monge-Ampère equation with the convexity constraint for any (arbitrarily small) positive value of γ. Thus we expect that the best possible consistency error we can observe would occur if we set $\gamma = \mathcal{O}(\delta^{d-1}) = \mathcal{O}(h_{eff})$. In this case, the formal consistency error has the form

$$h_{eff}^{\frac{1}{d-1}} \sum_{j=1}^{d} \mathbb{1}_{u_{\nu_j \nu_j} < h_{eff}^{1/(d-1)}} + h_{eff}^{\frac{d}{d-1}} + h_{eff}^2 + d\theta.$$

In particular, we want to consider the case of non-degenerate examples, where solutions are strictly convex. Then all the second directional derivatives will be strictly positive and, for small enough h, the solution will satisfy

$$u_{\nu_j \nu_j} > h_{eff}^{1/(d-1)}.$$

This means that the consistency error will simplify to

$$h_{eff}^{\frac{d}{d-1}} + h_{eff}^2 + d\theta.$$

In this case, the formal spatial accuracy will be better than first order. In particular, in the two-dimensional setting we actually obtain second order accuracy in the spatial resolution h_{eff}.

4.8.5 Newton's Method

The monotone discretisation of the more general Monge-Ampère equation results in a system of equations that, as before, can be solved efficiently using Newton's method.

This again involves performing the iteration

$$u^{k+1} = u^k - v^k$$

where the corrector v^k is obtained by solving the equation

$$\nabla MA[u^k]v^k = MA[u^k].$$

We recall that this discretisation has the form

$$MA_M[u] = -\min_{(\nu_1,\ldots,\nu_d)\in\mathcal{G}} G_{(\nu_1,\ldots,\nu_d)}[u].$$

As before, we can write the Jacobian as

$$\nabla MA_M[u] = -\nabla G_{(\nu_1,\ldots,\nu_d)}[u],$$

where the $(\nu_1,\ldots,\nu_d)$ are the directions active in the minimum. The components of this Jacobian are now given by:

$$\begin{aligned}\nabla_{u_i} G_{(\nu_1,\ldots,\nu_d)}[u] = -\sum_{m=1}^{d}\left[\left(\prod_{j\neq m}\max\{\mathcal{D}_{\nu_j\nu_j}u_i,\delta\}\right)\mathbb{1}_{\mathcal{D}_{\nu_j\nu_j}u_i\geq\delta} + \gamma\mathbb{1}_{\mathcal{D}_{\nu_j\nu_j}u_i<\delta}\right]\mathcal{D}_{\nu_m\nu_m}\\ + \sum_{m=1}^{d}\frac{\partial F}{\partial p_m}\left(x,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_1}{|\nu_j|}\mathcal{D}_{\nu_j}u_i,\ldots,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_d}{|\nu_j|}\mathcal{D}_{\nu_j}u_i\right)\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_m}{|\nu_j|}\mathcal{D}_{\nu_j}.\end{aligned}$$

4.9 Computational Results: Two Dimensions

We now provide computational results to validate the theory developed in this chapter. We begin by providing specific details for the four two-dimensional examples described in §2.5. In all these examples, the right-hand side of the Monge-Ampère equation will not depend on gradients of the solution. Results for a more general Monge-Ampère equation will be delayed until Chapter 6, when we will be better equipped to evaluate these results in the context of optimal transport.

We perform computations using 9, 17, and 33 point stencils. These stencils look along the directions in:

$$\begin{aligned}
\mathcal{G}_9 &= \big\{\{(1,0),(0,1)\},\{(1,1),(1,-1)\}\big\} \\
\mathcal{G}_{17} &= \mathcal{G}_9 \cup \big\{\{(1,2),(2,-1)\},\{(2,1),(1,-2)\}\big\} \\
\mathcal{G}_{33} &= \mathcal{G}_{17} \cup \big\{\{(1,3),(3,-1)\},\{(3,1),(1,-3)\},\{(2,3),(3,-2)\},\{(3,2),(2,-3)\}\big\}.
\end{aligned}$$

We also perform a comparison with the standard methods described in Chapter 3.

4.9.1 Accuracy

In this section, we present accuracy results for the four representative examples described in §2.5; see Table 4.1 and Figure 4.2. We perform the computations using the monotone scheme on 9, 17, and 33 point stencils. The accuracy of the scheme is determined by a combination of the directional resolution, $d\theta$, error and the spatial discretisation error. Widening the stencil, which has the effect of decreasing $d\theta$, improves the accuracy, as does increasing the number of grid points.

We also compared the accuracy to standard finite differences. Standard finite differences are formally more accurate since there is no $d\theta$ error, and we certainly observe this in the computations. This is particularly evident for the C^2 and C^1 examples, where the error in the standard discretisation is much lower than the error in the monotone discretisations (with reasonably narrow stencils).

4.9.2 Computation Time

One of the big advantages of the monotone scheme is that it allows us to use Newton's method, which could become unstable or converge to the wrong solution when combined with the standard discretisation. Consequently, the monotone scheme allows for a big improvement in solution time.

To support this claim, we compare computation times required by the monotone Newton's method (on a 17 point stencil) with the times required by the Poisson and Gauss-Seidel iterations described in Chapter 3. These are presented in Table 4.2 and Figure 4.3. In terms of absolution solution time, the Newton solver is faster for each of the four representative examples of §2.5.

Max Error, C^2 Example (2.17)

N	9 Point	17 Point	33 Point	Standard
31	9.45×10^{-5}	9.12×10^{-5}	9.38×10^{-5}	4.54×10^{-5}
45	6.31×10^{-5}	5.36×10^{-5}	5.40×10^{-5}	2.11×10^{-5}
63	4.91×10^{-5}	3.42×10^{-5}	3.40×10^{-5}	1.06×10^{-5}
89	4.17×10^{-5}	2.30×10^{-5}	2.17×10^{-5}	0.53×10^{-5}
127	3.79×10^{-5}	1.67×10^{-5}	1.39×10^{-5}	0.26×10^{-5}
181	3.60×10^{-5}	1.34×10^{-5}	0.92×10^{-5}	0.13×10^{-5}
255	3.51×10^{-5}	1.17×10^{-5}	0.66×10^{-5}	0.06×10^{-6}
361	3.48×10^{-5}	1.08×10^{-5}	0.51×10^{-5}	0.03×10^{-6}

Max Error, C^1 Example (2.18)

N	9 Point	17 Point	33 Point	Standard
31	21.54×10^{-4}	8.66×10^{-4}	6.39×10^{-4}	3.78×10^{-4}
45	20.89×10^{-4}	6.84×10^{-4}	4.07×10^{-4}	1.82×10^{-4}
63	21.33×10^{-4}	6.82×10^{-4}	3.18×10^{-4}	1.34×10^{-4}
89	21.40×10^{-4}	6.51×10^{-4}	2.70×10^{-4}	0.85×10^{-4}
127	21.55×10^{-4}	6.63×10^{-4}	2.49×10^{-4}	0.59×10^{-4}
181	21.54×10^{-4}	6.62×10^{-4}	2.40×10^{-4}	0.37×10^{-4}
255	21.51×10^{-4}	6.58×10^{-4}	2.36×10^{-4}	—
361	21.53×10^{-4}	6.62×10^{-4}	2.37×10^{-4}	—

Max Error, Example with blow-up (2.19)

N	9 Point	17 Point	33 Point	Standard
31	1.74×10^{-3}	1.74×10^{-3}	1.74×10^{-3}	17.38×10^{-3}
45	0.98×10^{-3}	0.98×10^{-3}	0.98×10^{-3}	14.74×10^{-3}
63	0.86×10^{-3}	0.59×10^{-3}	0.59×10^{-3}	12.62×10^{-3}
89	0.84×10^{-3}	0.37×10^{-3}	0.35×10^{-3}	10.72×10^{-3}
127	0.83×10^{-3}	0.35×10^{-3}	0.20×10^{-3}	9.04×10^{-3}
181	0.83×10^{-3}	0.34×10^{-3}	0.17×10^{-3}	7.61×10^{-3}
255	0.83×10^{-3}	0.33×10^{-3}	0.16×10^{-3}	6.43×10^{-3}
361	0.83×10^{-3}	0.33×10^{-3}	0.15×10^{-3}	5.42×10^{-3}

Max Error, $C^{0,1}$ (Lipschitz) Example (2.20)

N	9 Point	17 Point	33 Point	Standard
31	11.83×10^{-3}	3.57×10^{-3}	1.61×10^{-3}	5.19×10^{-3}
45	10.35×10^{-3}	3.42×10^{-3}	1.68×10^{-3}	3.82×10^{-3}
63	11.10×10^{-3}	3.49×10^{-3}	1.65×10^{-3}	2.86×10^{-3}
89	10.12×10^{-3}	3.44×10^{-3}	1.69×10^{-3}	2.12×10^{-3}
127	11.80×10^{-3}	3.45×10^{-3}	1.64×10^{-3}	1.54×10^{-3}
181	10.38×10^{-3}	3.70×10^{-3}	1.64×10^{-3}	1.12×10^{-3}
255	10.47×10^{-3}	3.46×10^{-3}	1.64×10^{-3}	—
361	10.40×10^{-3}	3.45×10^{-3}	1.64×10^{-3}	—

Table 4.1: Accuracy of the monotone and standard discretisations on four representative examples.

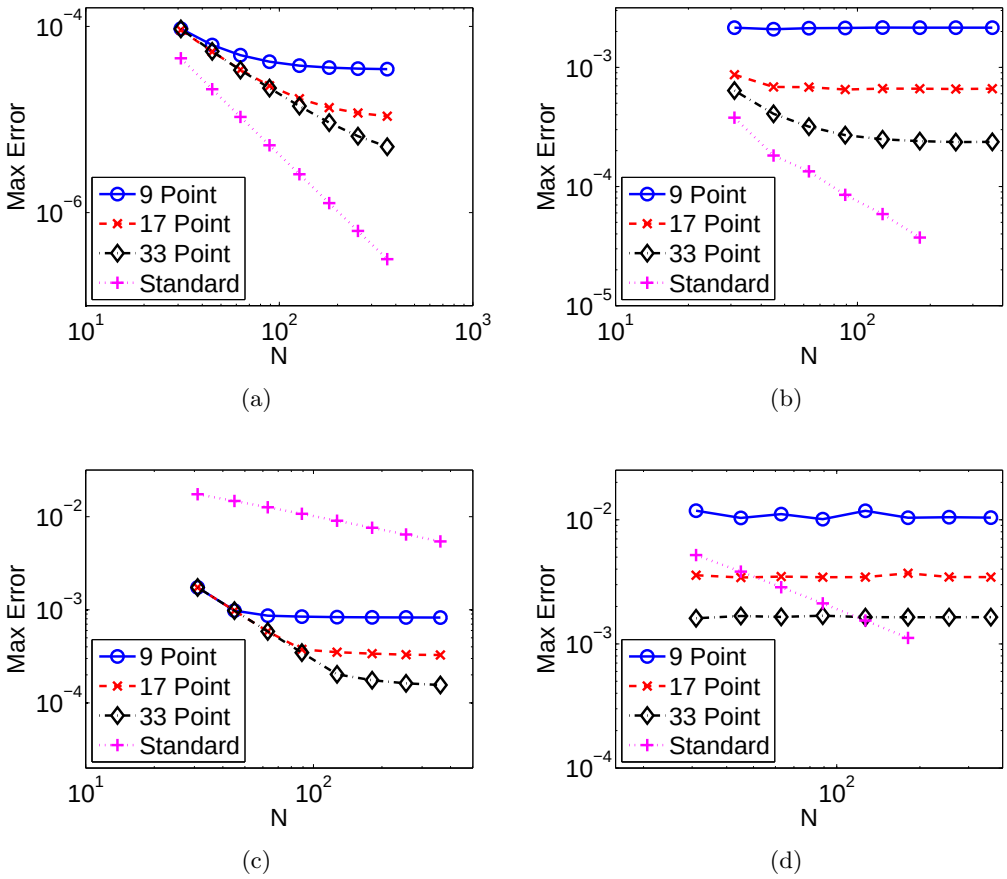


Figure 4.2: Accuracy of the monotone and standard discretisations on the (a) C^2 example, (b) C^1 example, (c) example with blow-up, and (d) Lipschitz example.

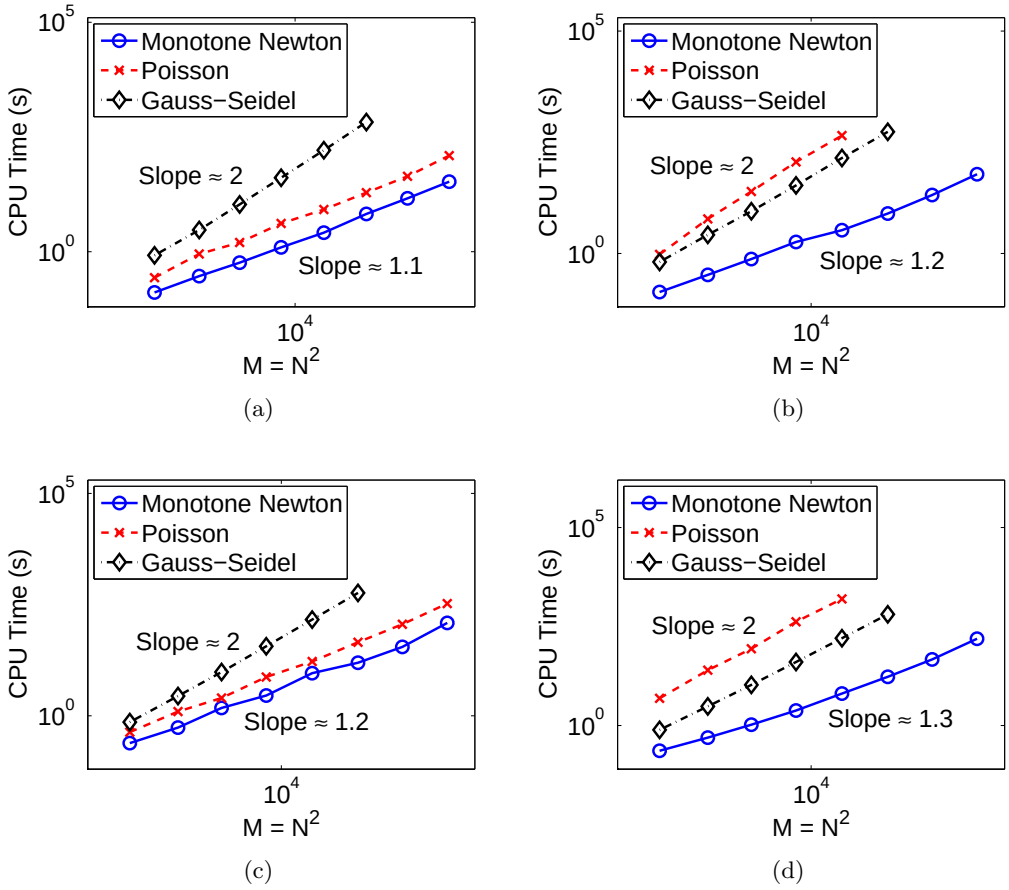


Figure 4.3: Computation times for the 17 point monotone and standard discretisations on the (a) C^2 example, (b) C^1 example, (c) example with blow-up, and (d) Lipschitz example.

We are also interested in how well the computation times scale as the number of grid points ($M \equiv N^2$) is increased. Order of magnitude solution times are presented in Table 4.3. We find that the order of magnitude solution time for Newton's method is similar to or, in the more singular examples, faster than the solution times for the other two-dimensional methods.

C^2 Example (2.17)

N	Newton Iterations	CPU Time (seconds) Newton	Poisson	Gauss-Seidel
31	3	0.1	0.3	0.8
45	3	0.3	0.9	3.0
63	3	0.6	1.6	10.7
89	3	1.2	4.1	41.0
127	3	2.6	8.3	163.1
181	3	6.6	19.3	666.0
255	3	14.6	44.0	—
361	3	33.6	124.5	—

C^1 Example (2.18)

N	Newton Iterations	CPU Time (seconds) Newton	Poisson	Gauss-Seidel
31	3	0.1	1.0	0.6
45	3	0.3	6.0	2.6
63	4	0.7	24.7	8.8
89	5	1.8	114.0	33.8
127	4	3.3	447.1	139.9
181	4	7.9	—	541.8
255	5	20.6	—	—
361	6	60.4	—	—

Example with blow-up (2.19)

N	Newton Iterations	CPU Time (seconds) Newton	Poisson	Gauss-Seidel
31	6	0.2	0.4	0.7
45	6	0.5	1.2	2.8
63	9	1.5	2.5	9.6
89	7	2.8	7.4	36.5
127	11	9.1	16.6	144.9
181	7	15.5	45.0	577.6
255	7	35.2	113.7	—
361	11	122.2	331.9	—

$C^{0,1}$ (Lipschitz) Example (2.20)

N	Newton Iterations	CPU Time (seconds) Newton	Poisson	Gauss-Seidel
31	6	0.2	4.9	0.8
45	6	0.5	25.0	3.1
63	6	1.1	86.9	10.6
89	7	2.4	417.2	40.1
127	8	6.4	1576.7	160.4
181	8	17.0	—	642.9
255	9	46.6	—	—
361	10	155.6	—	—

Table 4.2: Computation times for the 17 point monotone Newton, Poisson, and Gauss-Seidel methods for four representative examples.

Method	Regularity of Solution $C^{2,\alpha}$ (2.17)	$C^{1,\alpha}$ (2.18)	Blow-up (2.19)	$C^{0,1}$ (2.20)
Gauss-Seidel	$\sim \mathcal{O}(M^2)$	$\sim \mathcal{O}(M^2)$	$\sim \mathcal{O}(M^2)$	$\sim \mathcal{O}(M^2)$
Poisson	$\sim \mathcal{O}(M^{1.2})$	$\sim \mathcal{O}(M^2)$	$\sim \mathcal{O}(M^{1.3})$	$\sim \mathcal{O}(M^2)$
Monotone Newton	$\sim \mathcal{O}(M^{1.1})$	$\sim \mathcal{O}(M^{1.2})$	$\sim \mathcal{O}(M^{1.2})$	$\sim \mathcal{O}(M^{1.3})$

Table 4.3: Order of magnitude computation time for the different solvers in terms of solution regularity. Here $M = N^2$ is the total number of grid points.

4.10 Computational Results: Three Dimensions

In this section, we perform computations to test the speed and accuracy of the monotone Newton's method for three dimensional problems. These computations are performed on an $N \times N \times N$ grid on the square $[0,1]^3$ using a 19 point stencil, which leads to the allowed directions:

$$\mathcal{G} = \big\{\{(1,0,0),(0,1,0),(0,0,1)\},\{(1,0,0),(0,1,1),(0,1,-1)\},\\ \{(0,1,0),(1,0,1),(1,0,-1)\},\{(0,0,1),(1,1,0),(1,-1,0)\}\big\}.$$

We present results for the three dimensional versions of the examples in §2.5. Since the methods of Chapter 3 are restricted to the two-dimensional Monge-Ampère equation, we cannot compare these methods in three dimensions.

The size of the computation was restricted by the available memory, not by solution time. The linear systems that arise in Newton's method involve sparse $N^3 \times N^3$ matrices. In our implementation of Newton's methods, we construct these matrices and solve the resulting linear systems using the Matlab backslash operator. Although these matrices are formed using a sparse data structure, their construction and solution still require a great deal of memory. However, this situation could certainly be improved by using an iterative method that does not require the construction of the large Jacobian matrices.

Computation times and accuracy results for the three dimensional examples are presented in Table 4.4.

C^2 Example (2.21)

N	Max Error	Iterations	CPU Time (s)
7	1.46×10^{-3}	3	0.1
11	0.67×10^{-3}	2	0.1
15	0.42×10^{-3}	3	0.4
21	0.27×10^{-3}	3	1.8
31	0.22×10^{-3}	4	20.2
45	0.20×10^{-3}	4	242.0

C^1 Example (2.22)

N	Max Error	Iterations	CPU Time (s)
7	5.29×10^{-3}	5	0.1
11	4.04×10^{-3}	8	0.3
15	3.15×10^{-3}	8	0.9
21	2.78×10^{-3}	8	4.2
31	2.52×10^{-3}	6	34.6

Example with Blow-up (2.23)

N	Max Error	Iterations	CPU Time (s)
7	7.11×10^{-3}	4	0.04
11	5.29×10^{-3}	8	0.22
15	4.62×10^{-3}	6	0.77
21	4.22×10^{-3}	10	5.67
31	4.03×10^{-3}	14	79.02

Table 4.4: Maximum error and computation time on three representative examples.

4.11 Conclusions

In this chapter, we have succeeded in constructing finite difference methods for the elliptic Monge-Ampère equation that will provably converge to the convex viscosity solution in any spatial dimension. The resulting system of nonlinear equations can be solved with Newton's method. Computational examples indicate that this monotone scheme is competitive with—and in many non-smooth cases, much faster than—finite difference methods based on standard discretisations.

One of the main limitations of this monotone method is the accuracy of solutions, which is limited by the stencil width. Since we want to use relatively narrow finite difference stencils in practice, this can severely limit the accuracy we can achieve. Techniques for improving the accuracy of the methods will be addressed in the next chapter.

Chapter 5

Hybrid Finite Difference Methods

In Chapter 4, we developed a finite difference discretisation that converges to the viscosity solution of the elliptic Monge-Ampère equation. The main downside to this scheme is that it has limited accuracy, with a consistency error that depends not only on the spatial resolution h, but also on the angular resolution $d\theta$. This means that impractically wide stencils may be required to achieve high accuracy. A formally more accurate ($\mathcal{O}(h^2)$) discretisation was studied in Chapter 3. Despite the better formal accuracy, this scheme may not converge to the correct weak solution when solutions are singular.

In this chapter, we combine the best features of these two schemes in order to build a hybrid discretisation that achieves higher accuracy in smooth regions of the solution, while still successfully capturing the behaviour of the viscosity solution near singularities. This is done by using the monotone scheme $(MA)^M$ near points where the solution is (or may be) singular and the standard scheme $(MA)^S$ elsewhere. To do this, we require a systematic way of characterising a solution (or its discrete approximation) as either singular or non-singular. In this chapter, we explore two possible characterisations. For one of these, we can prove that our hybrid scheme converges to the viscosity solution of the Monge-Ampère equation. In fact, in the course of obtaining this result, we also prove a very general theorem about the convergence of certain formally higher-order approximation schemes for a large class of degenerate elliptic PDEs.

5.1 *A Priori* Hybrid Discretisation

A natural option for distinguishing between singular and non-singular discrete approximations is to look at the size of certain derivatives (or their discrete approximations). Since the Monge-Ampère equation is second-order, it is natural to characterise a discrete solution as singular if its second derivatives are large. One advantage of this approach is that we can make use of regularity results for the Monge-Ampère equation to define an *a priori* hybrid discretisation. That is, the particular choice of monotone or standard scheme at each point is pre-determined and does not depend on the computed solution.

5.1.1 Discretisation

The Monge-Ampère equation has a rich regularity theory, which we have briefly discussed in §2.2.1. Using this theory and the given data, we can characterise the possible regions where the solution of the Monge-Ampère equation can become singular.

We begin by identifying the set X^s, which is a neighborhood of the possible singular set of u on X that is defined using the regularity conditions (2.7). Letting ϵ be a small parameter, which we can take equal to the spatial resolution h, we define the singular set as

$$\begin{aligned} X^s = &\{x \in X \mid f(x) < \epsilon \text{ or } f(x) > 1/\epsilon \text{ or } f(x) \notin C^\alpha \text{ in an } \epsilon\text{-neighbourhood of } x\} \cup \\ &\{x \in \partial X \mid \partial X \text{ is not strictly convex at } x \text{ or } \phi(x) \notin C^{2,\alpha} \text{ in an } \epsilon\text{-neighbourhood of } x\}. \end{aligned}$$

Next we choose a weight function $w : X \to [0, 1]$ that is zero in an h-neighbourhood of X^s, and that goes to 1 elsewhere.

This allows us to construct the following *a priori* hybrid discretisation, which is simply an average of the monotone and standard schemes:

$$MA^H = w(x)MA^S + (1 - w(x))MA^M. \qquad (MA)^H$$

We remark that for C^2 solutions, this hybrid scheme will sometimes be less accurate than the standard finite differences. This is because it will lose some accuracy near any flat (non-strictly convex) boundary. While this might seem conservative, we have seen in §2.2.1 that the flat boundary can lead to a loss of regularity.

5.1.2 Newton's Method

Next we consider Newton's method for this hybrid scheme. To set up the equation (3.3) for the Newton step, the Jacobian of the scheme is again needed. Since the hybrid discretisation is a weighted average of the monotone and standard discretisations, and since the weight function $w(x)$ is determined *a priori*, the Jacobian of the hybrid scheme will simply be a weighted average of the Jacobians of the component schemes.

Thus Newton's method is simply

$$u^{n+1} = u^n - v^n$$

where the corrector is obtained by solving the weighted average of the two linearisations

$$(w(x)\nabla_u MA^S[u^n] + (1-w(x))\nabla_u MA^M[u^n])v^n \\ = w(x)MA^S[u^n] + (1-w(x))MA^M[u^n]. \quad (5.1)$$

We incorporate damping and regularisation into this scheme as described in §3.2.

5.2 Filtered Discretisation

The hybrid discretisation we have just described is formally more accurate than the monotone discretisation since, providing the data is sufficiently well-behaved, it does not require a wide stencil. However, by sacrificing monotonicity we also sacrifice the convergence proof of Chapter 4.

Now we consider an alternative approach for classifying a discrete solution as singular, which will depend on the particular scheme we are considering. Instead of looking at the derivatives of the solution, we now look at the value of the standard scheme when we input the solution. Intuitively, the idea is that for smooth functions, any two consistent discretisations (for example, our monotone and standard schemes) should have values that are close to each other. We will say that a function is singular with respect to a particular scheme if the value of that scheme is far from the value of the monotone scheme (which we know correctly approximates the viscosity solution). This idea leads us to consider a filtered scheme of the form

$$MA_M[u] + \mathcal{F}\left(MA_S[u] - MA_M[u]\right). \quad (5.2)$$

Here the function $\mathcal{F}$ should be equal to the identity if its argument is small in magnitude and should vanish otherwise. Thus as long as the standard scheme is approximating the PDE well in some sense, the filtered scheme will simply reduce to the higher-order scheme. If the standard scheme is not approximating the PDE correctly, the monotone scheme is used to ensure correctness.

As with our *a priori* hybrid scheme, this filtered scheme is not monotone. However, this formulation ensures that it is at least close to a monotone scheme. This property will enable us to prove that this non-monotone scheme converges to the viscosity solution of the Monge-Ampère equation.

5.2.1 Viscosity Solutions of Elliptic Equations

Because the filtered discretisation we are considering is not monotone, it will no longer fit into the convergence framework of [72], which we relied on in Chapter 4. The convergence of certain higher-order, non-monotone schemes has been studied for Hamilton-Jacobi equations [1, 61]. However, we are not aware of similar results for second order equations. This means that we must establish new convergence results that will apply to our filtered scheme.

We begin our discussion in the very general setting of second-order degenerate elliptic equations of the form

$$F(x, u(x), \nabla u(x), D^2 u(x)) = 0, \quad x \in X \subset \mathbb{R}^d, \tag{5.3}$$

together with appropriate boundary conditions.

Remark. Throughout the remainder of this section, we will assume that boundary conditions have been incorporated into the operator F so that equation (5.3) can be posed in the closed domain $\bar{X}$.

In Chapter 2, we remarked that the Monge-Ampère equation belongs to the class of elliptic equations because of its monotone dependence on the eigenvalues of the Hessian. The equations we are now considering call for a slightly more general definition of degenerate ellipticity.

Definition 5.1 (Degenerate Elliptic Equations). The equation (5.3) is *degenerate elliptic* if

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

for all $x \in \bar{X}$, $r, s \in \mathbb{R}$, $p \in \mathbb{R}^n$, $Z, Y \in S^n$ with $Z \geq Y$ and $r \leq s$.

As we have already seen for the Monge-Ampère equation, elliptic equations need not have smooth solutions. Viscosity solutions, which we defined for the Monge-Ampère equation in §2.2.3, can also be defined in this more general setting.

Before we give this more general definition, we need to introduce the notion of semi-continuity.

Definition 5.2 (Semi-Continuous). A function $u : X \to \mathbb{R}$ is *upper (lower) semi-continuous* if for every point $x_0 \in X$,

$$u(x_0) \geq \limsup_{x \to x_0} u(x)$$

$$\left(u(x_0) \leq \liminf_{x \to x_0} u(x) \right).$$

For brevity of notion, we will use $USC(X)$ and $LSC(X)$ to denote the sets of real-valued upper and lower semi-continuous functions defined on the domain X.

We can also define the upper and lower-semi continuous envelopes of a function.

Definition 5.3 (Upper and Lower Semi-Continuous Envelope). The *upper and lower semi-continuous envelopes* of a function $u(x)$ are defined, respectively, by

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

$$u_*(x) = \liminf_{y \to x} u(y).$$

We are now prepared to define viscosity solutions of elliptic equations.

Definition 5.4 (Viscosity Solution). An upper (lower) semi-continuous function u is a *viscosity sub(super)-solution* of (5.3) if for every $\phi \in C^2(\bar{X})$, if $u - \phi$ has a local maximum (minimum) at $x \in \bar{X}$, then

$$F_*(x, u(x), \nabla\phi(x), D^2\phi(x)) \leq 0$$

$$(F^*(x, u(x), \nabla\phi(x), D^2\phi(x)) \geq 0).$$

A function u is a *viscosity solution* if it is both a sub- and a super-solution.

A very useful property of viscosity solutions is their stability under perturbation not only of the solution, but also of the operator. This is important in developing approximation schemes. Another important property of viscosity solutions of degenerate elliptic equations is the comparison property, which guarantees uniqueness [60].

Theorem 5.1 (Comparison Property)**.** *Under mild structure conditions on a degenerate elliptic operator, the following result holds. If $u \in USC(\bar{X})$ is a sub-solution and $v \in LSC(\bar{X})$ is a super-solution of* (5.3) *then $u \leq v$ on $\bar{X}$.*

Remark. As we have already noted in Theorem 2.4, the Monge-Ampère equation does satisfy a comparison principle.

5.2.2 Convergence of Approximation Schemes

We now want to consider a scheme for approximating the degenerate elliptic equation (5.3). We will be using an approximation scheme of the form

$$F^\epsilon(x, u^\epsilon(x), u^\epsilon(\cdot)) = 0 \tag{5.4}$$

where ϵ is a discretisation parameter. In practice, this could be the spatial and/or directional resolution of a finite difference stencil.

Remark. The solution of the approximation scheme will normally be given on the grid, but we assume that we have a continuous extension of this into the domain $\bar{X}$.

The work of Barles and Souganidis [4], which was foundational to the schemes constructed in Chapter 4, demonstrates that approximation schemes will converge if they are consistent, stable, and monotone. To facilitate the development of a higher-order filtered scheme, we now want to relax this requirement and allow for schemes that may not be monotone. In particular, our theory will closely follow the work of [4] except that we now require schemes to be consistent, stable, and almost monotone.

Definition 5.5 (Consistent)*.* The scheme (5.4) is *consistent* with the equation (5.3) if for any smooth function ϕ and $x \in \bar{X}$,

$$\limsup_{\epsilon \to 0, y \to x, \xi \to 0} F^\epsilon(y, \phi(y) + \xi, \phi(\cdot) + \xi) \leq F^*(x, \phi(x), \nabla\phi(x), D^2\phi(x)),$$

$$\liminf_{\epsilon \to 0, y \to x, \xi \to 0} F^\epsilon(y, \phi(y) + \xi, \phi(\cdot) + \xi) \geq F_*(x, \phi(x), \nabla\phi(x), D^2\phi(x)).$$

Definition 5.6 (Stable)*.* The scheme (5.4) is *stable* if any solution u^ϵ of (5.4) is bounded independently of ϵ.

Definition 5.7 (Almost Monotone)*.* The scheme (5.4) is *almost monotone* if for every $\epsilon > 0$, $x \in \bar{X}$, $t \in \mathbb{R}$ and bounded $u \geq v$

$$F^\epsilon(x, t, u(\cdot)) \leq F^\epsilon(x, t, v(\cdot)) + r(\epsilon)$$

where

$$\lim_{\epsilon \to 0} r(\epsilon) = 0.$$

With these definitions, we can now give our convergence result.

Theorem 5.2 (Convergence of Approximation Schemes)**.** *For each $\epsilon > 0$ let u^ϵ be a solution of* (5.4)*. Then as $\epsilon \to 0$, u^ϵ converges locally uniformly to the viscosity solution of* (5.3)*.*

We begin with two lemmas.

Lemma 5.1 (Viscosity Solutions)*.* In the definition of viscosity solutions (Definition 5.4), it is sufficient to consider unique, strict, global maxima (minima) with $u(x) - \phi(x) = 0$ at the extremum.

The relaxations allowed by this lemma are fairly standard; see, for example, [60, Prop. 2.2]. We include a proof here for completeness and clarity.

Proof. Suppose that a bounded, upper semi-continuous function u satisfies the criteria of Definition 5.4 where "local max (min)" is replaced with "unique, strict, global max (min) with a value of zero". We verify that u is a viscosity subsolution. We can similarly show that it is a supersolution.

Choose any smooth function ϕ such that $u - \phi$ has a local max at a point $x_0 \in \bar{X}$. Then there exists $r > 0$ such that

$$u(x_0) - \phi(x_0) \geq u(x) - \phi(x), \quad \text{for } x \in B(x_0, r).$$

Now we choose a number

$$M > \frac{1}{r^4} \left(\max_{\bar{X}} \left| \phi(x) + u(x_0) - \phi(x_0) \right| + \max_{\bar{X}} u(x) \right)$$

and define

$$\tilde{\phi}(x) = \phi(x) + (u(x_0) - \phi(x_0)) + M|x - x_0|^4.$$

Then by hypothesis,

$$\begin{aligned} 0 &\geq F(x_0, u(x_0), \nabla\tilde{\phi}(x_0), D^2\tilde{\phi}(x_0)) \\ &= F(x_0, u(x_0), \nabla\phi(x_0), D^2\phi(x_0)). \end{aligned}$$

Thus u is a subsolution and the definitions agree. □

Lemma 5.2 (Stability of Maxima). Define

$$\bar{u}(x) = \limsup_{\epsilon \to 0, y \to x} u^{\epsilon}(y) \in USC(\bar{X}),$$

which is bounded by the stability property. For a smooth function ϕ, let x_0 be the unique strict global maximizer of $\bar{u} - \phi$ with $\bar{u}(x_0) = \phi(x_0)$. Then there exist sequences:

$$\begin{cases} \epsilon_n \to 0 \\ y_n \to x_0 \\ u^{\epsilon_n}(y_n) \to \bar{u}(x_0) \end{cases}$$

where y_n is a global maximiser of $u^{\epsilon_n} - \phi$.

Proof. From the definition of the limit superior, we can find sequences

$$\epsilon_n \to 0, \quad z_n \to x_0$$

such that

$$u^{\epsilon_n}(z_n) \to \bar{u}(x_0).$$

Now we define $y_n \in \bar{X}$ to be maximisers of $u^{\epsilon_n}(x) - \phi(x)$.

We have

$$u^{\epsilon_n}(y_n) - \phi(y_n) \geq u^{\epsilon_n}(z_n) - \phi(z_n) \to \bar{u}(x_0) - \phi(x_0) = 0.$$

Also, for any $\delta > 0$ and large enough n,

$$u^{\epsilon_n}(y_n) - \phi(y_n) \leq \bar{u}(y_n) - \phi(y_n) + \delta \leq \bar{u}(x_0) - \phi(x_0) + \delta = \delta.$$

Thus we have

$$u^{\epsilon_n}(y_n) - \phi(y_n) \to 0.$$

Now suppose we do not have $y_n \to x_0$. Then (possibly through a subsequence) there is an $R > 0$ such that

$$|y_n - x_0| > R.$$

Also, since the max is strict, global, and unique, there is a $K > 0$ such that

$$\bar{u}(y) - \phi(y) < -K < 0$$

whenever $|y - x_0| > R$.

Thus for any $\delta > 0$ and large enough n,

$$u^{\epsilon_n}(y_n) - \phi(y_n) \leq \bar{u}(y_n) - \phi(y_n) + \delta < -K + \delta \to -K < 0,$$

which contradicts the fact that $u^{\epsilon_n}(y_n) - \phi(y_n) \to 0$. We conclude that

$$y_n \to x_0.$$

Finally, it is clear that

$$\begin{aligned} \left|u^{\epsilon_n}(y_n) - \bar{u}(x_0)\right| &= \left|u^{\epsilon_n}(y_n) - \phi(x_0)\right| \\ &\leq \left|u^{\epsilon_n}(y_n) - \phi(y_n)\right| + \left|\phi(y_n) - \phi(x_0)\right| \\ &\to 0. \end{aligned}$$

Therefore,

$$u^{\epsilon_n}(y_n) \to \bar{u}(x_0). \qquad \square$$

Proof of Theorem 5.2. Define

$$\bar{u}(x) = \limsup_{\epsilon \to 0, y \to x} u^{\epsilon}(y) \in USC(\bar{X}),$$

$$\underline{u}(x) = \liminf_{\epsilon \to 0, y \to x} u^{\epsilon}(y) \in LSC(\bar{X}).$$

These are bounded by the stability property.

Now we show that $\bar{u}$ is a sub-solution. For a smooth function ϕ, let x_0 be a strict global maximum of $\bar{u} - \phi$ with $\phi(x_0) = \bar{u}(x_0)$ (Lemma 5.1). By Lemma 5.2, we can find sequences with

$$\begin{cases} \epsilon_n \to 0 \\ y_n \to x_0 \\ u^{\epsilon_n}(y_n) \to \bar{u}(x_0) \end{cases}$$

where y_n is a global maximiser of $u^{\epsilon_n} - \phi$.

We define

$$\xi_n = u^{\epsilon_n}(y_n) - \phi(y_n) \to \bar{u}(x_0) - \phi(x_0) = 0.$$

We also recall that

$$u^{\epsilon_n}(x) - \phi(x) \leq u^{\epsilon_n}(y_n) - \phi(y_n) = \xi^n \quad \text{for any } x \in \bar{X}.$$

Using these definitions and the almost monotonicity of the scheme, we find that

$$\begin{aligned}
0 &= F^{\epsilon_n}(y_n, u^{\epsilon_n}(y_n), u^{\epsilon_n}(\cdot)) \\
&= F^{\epsilon_n}(y_n, \phi(y_n) + \xi_n, \phi(\cdot) + (u^{\epsilon_n}(\cdot) - \phi(\cdot))) \\
&\geq F^{\epsilon_n}(y_n, \phi(y_n) + \xi_n, \phi(\cdot) + \xi_n) - r(\epsilon_n).
\end{aligned}$$

By consistency, we have

$$\begin{aligned}
0 &\geq \liminf_{n\to\infty} \left\{ F^{\epsilon_n}(y_n, \phi(y_n) + \xi_n, \phi(\cdot) + \xi_n) - r(\epsilon_n) \right\} \\
&\geq \liminf_{\epsilon\to 0, y\to x, \xi\to 0} F^{\epsilon_n}(y, \phi(y) + \xi, \phi(\cdot) + \xi) \\
&\geq F_*(x_0, \phi(x_0), \nabla\phi(x_0), D^2\phi(x_0)) \\
&= F_*(x_0, \bar{u}(x_0), \nabla\phi(x_0), D^2\phi(x_0)),
\end{aligned}$$

which shows that $\bar{u}$ is a subsolution. Similarly, we can show that $\underline{u}$ is a super-solution. By the comparison principle we have

$$\bar{u} \leq \underline{u}.$$

However, from their definitions, we know that

$$\underline{u} \leq \bar{u}.$$

Thus we conclude that $\bar{u} = \underline{u}$ is both a sub-solution and a super-solution, and is therefore the viscosity solution of (5.3). □

5.2.3 Convergence of Almost Monotone Finite Difference Methods

We now want to use the framework of Theorem 5.2 to construct a convergent, formally higher-order approximation scheme for the Monge-Ampère equation. We continue our discussion in the general setting and consider an almost monotone discretisation of the form

$$F^\epsilon[u] \equiv F^\epsilon_M[u] + \epsilon^\alpha S[x, u, \epsilon] = 0. \tag{5.5}$$

Here F^ϵ_M is a convergent monotone scheme. The function S should be bounded and continuous. We note that with a suitable choice of the function S, this scheme resembles the filtered scheme suggested in (5.2).

In this thesis, we have already constructed a convergent monotone scheme for the Monge-Ampère equation. We now want to use the properties of the monotone scheme to establish convergence of the perturbed scheme (5.5).

Theorem 5.3 (Convergence of Perturbed Schemes)**.** *Suppose that the scheme $F^\epsilon_M[u]$ is degenerate elliptic, proper, and locally Lipschitz continuous. Suppose also that S is a continuous, bounded function. Then solutions of the perturbed scheme*

$$F^\epsilon[u] \equiv F^\epsilon_M[u] + \epsilon S\left[x, u, \epsilon\right] = 0$$

exist and converge locally uniformly to the viscosity solution of the PDE (5.3)*.*

Before we prove this result, we state several lemmas, which will enable us to use Theorem 5.2.

Lemma 5.3 (Consistency and Almost Monotonicity). The perturbed scheme (5.5) is consistent with Equation (5.3) and is almost monotone.

Proof. This result follows immediately from the consistency and monotonicity of F^ϵ_M. □

Lemma 5.4 (Existence). Suppose that the scheme $F^\epsilon_M[u]$ is degenerate elliptic, proper, and locally Lipschitz continuous. Suppose also that S is a continuous, bounded function. Then the perturbed scheme

$$F^\epsilon[u] \equiv F^\epsilon_M[u] + \epsilon S\left[x, u, \epsilon\right] = 0$$

has a solution.

Proof. For a fixed $\epsilon > 0$, consider the function $y(u)$, defined as the solution vector of the scheme

$$F^\epsilon_M[y(u)] + \epsilon S\left[x, u, \epsilon\right] = 0.$$

From the theory in [72] and the continuity of S, the function $y(u)$ is uniquely defined and continuous. In addition, since the function S is bounded, the function y will also be bounded. In particular, there exists an R so that for any input u,

$$y(u) \in B_R$$

where B_R is the ball of radius R.

Now we restrict the domain of y to this ball and note that $y : B_R \to B_R$. By Brouwer's fixed point theorem, the function y has a fixed point in this same ball.

We conclude that the perturbed scheme has a solution. □

Lemma 5.5 (Stability). Suppose that the scheme $F^\epsilon_M[u]$ is degenerate elliptic, proper, and locally Lipschitz continuous. Suppose also that S is a continuous, bounded function. Then any solution u^ϵ of the perturbed scheme

$$F^\epsilon[u] \equiv F^\epsilon_M[u] + \epsilon S\left[x, u, \epsilon\right] = 0$$

can be bounded uniformly as $\epsilon \to 0$.

Proof. Let u be any solution of the perturbed scheme. Then u is also a solution of the monotone scheme

$$F^\epsilon_M[v] + \epsilon S\left[x, u, \epsilon\right] = 0.$$

Since the function S is bounded independently of u and ϵ, we can use the theory of [72] to bound the solution uniformly as $\epsilon \to 0$. □

Proof of Theorem 5.3. The hypotheses of Theorem 5.2 are established in Lemmas 5.3–5.5, which proves convergence to the viscosity solution. □

5.2.4 Construction of Filtered Schemes

Now we want to use this theory to construct more accurate approximation schemes. This can be done by appropriate choice of the function S, which we will refer to as a filter.

To do this, let us denote by $F^\epsilon_A[u]$ a more accurate approximation scheme. For example, we can consider the standard finite difference discretisation of the Monge-Ampère equation that was described in Chapter 3. Other higher-order schemes for this and other PDEs can also be constructed by looking at Taylor series expansions. In order for our filtered scheme to make use of this more accurate scheme, we need to choose the function S so that

$$S[x, u, \epsilon] = \frac{F^\epsilon_A[u] - F^\epsilon_M[u]}{\epsilon}$$

for sufficiently regular functions u. We also want the filtered scheme to reduce back to the monotone scheme if the accurate and monotone schemes give very different values, which might happen on a singular solution. This means that the function S should vanish if the difference

$$F^\epsilon_A[u] - F^\epsilon_M[u]$$

is large in magnitude.

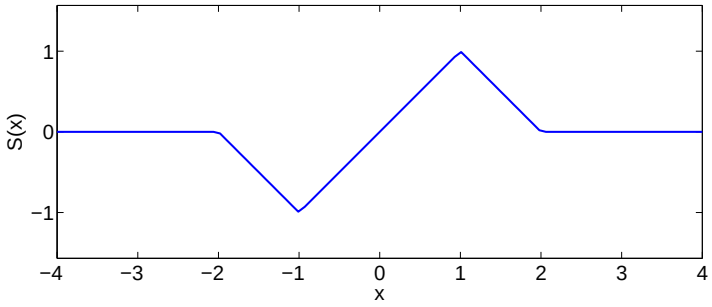


Figure 5.1: The filter used to construct a formally higher-order discretisation.

This motivates us to choose a filter S of the form

$$S\left[x, u, \epsilon\right] = S\left[\frac{F_A^\epsilon[u] - F_M^\epsilon[u]}{\epsilon}\right]$$

where F_A is an accurate scheme with a formal discretisation error that is less than $\mathcal{O}(\epsilon^\alpha)$. We define the function S by

$$S(x) = \begin{cases} x & |x| \leq 1 \\ \max\{2 - x, 0\} & x > 1 \\ \min\{-2 - x, 0\} & x < -1. \end{cases} \tag{5.6}$$

This filter is plotted in Figure 5.1.

Remark. Any continouous, bounded function S that is equal to the identity in a neighbourhood of the origin is equally valid.

The discussion of this chapter is all valid for the Monge-Ampère equation. Thus we now propose the following filtered discretisation.

$$MA_F^{h,d\theta,\delta}[u] \equiv MA_M^{h,d\theta,\delta} + \epsilon(h, d\theta, \delta) S\left[\frac{MA_S^h[u] - MA_M^{h,d\theta,\delta}[u]}{\epsilon(h, d\theta, \delta)}\right] \tag{$MA)^F$}$$

where $\epsilon(h, d\theta, \delta)$ converges to zero as h, $d\theta$, and δ go to zero.

Theorem 5.4 (Convergence of Filtered Scheme for Monge-Ampère)**.** *Let the PDE* (1.3) *have a unique viscosity solution and let the right-hand side $F(x, \nabla u)$ be Lipschitz continuous on $\overline{X} \times \mathbb{R}^d$ with Lipschitz constant K_F. Then the solutions of the scheme $(MA)^F$ exist and converge to the viscosity solution of* (1.3) *as $h, d\theta, \delta \to 0$ with $\gamma \geq \delta^{d-1} \geq K_F \left|\nu_j\right| h/2$ and $h_{eff} \geq h\left|\nu_j\right| \to 0$ for every $\nu_j \in \mathcal{G}$.*

Proof. This follows immediately from Theorems 4.10 and 5.3. □

5.2.5 Formal Accuracy

Now we want to verify that the filtered scheme does in fact lead to an improvement in the formal accuracy.

Now let us consider a smooth solution ϕ of the Monge-Ampère equation. By construction, the standard scheme $(MA)^S$ has a formal accuracy of $\mathcal{O}(h^2)$. With an appropriate choice of parameters, the formal accuracy of the monotone scheme is at worst $\mathcal{O}(h + d\theta)$ (§4.8.4). (It is $\mathcal{O}(h^2 + d\theta)$ for strictly convex solutions). Let us also choose the perturbation size $\epsilon(h, d\theta, \delta)$ to be $\mathcal{O}(h^\alpha + d\theta^\beta)$ where α and β are less than or equal to one.

We first observe that the argument of the filter S will be on the order of

$$\begin{aligned}\frac{MA_S^h[\phi] - MA_M^{h,d\theta}[\phi]}{\epsilon(h, d\theta, \delta)} &= \frac{\mathcal{O}(h^2) + \mathcal{O}(h + d\theta)}{h^\alpha + d\theta^\beta} \\ &= \frac{\mathcal{O}(h + d\theta)}{\mathcal{O}\left(\max\{h^\alpha, d\theta^\beta\}\right)} \\ &= \mathcal{O}\left(\min\{h^{1-\alpha}, h/d\theta^\beta\} + \min\{d\theta^{1-\beta}, d\theta/h^\alpha\}\right) \\ &\leq \mathcal{O}(1).\end{aligned}$$

This means that the filter will act as the identity operator.

Thus the filtered scheme will be given by

$$\begin{aligned}MA_F^{h,d\theta,\delta}[\phi] = MA_M^{h,d\theta,\delta}[\phi] + \epsilon(h, d\theta, \delta)\frac{MA_S^h[\phi] - MA_M^{h,d\theta,\delta}[\phi]}{\epsilon(h, d\theta, \delta)} \\ MA_S^h[\phi],\end{aligned}$$

which is just the standard scheme.

We conclude that the formal discretisation error in the filtered scheme will be $\mathcal{O}(h^2)$, just like the original standard scheme.

5.2.6 Newton's method

As before, we solve the discrete system using Newton's method:

$$u^{n+1} = u^n - (\nabla MA_F[u^n])^{-1} MA_F[u^n]$$

where the Jacobian is given by

$$\nabla MA_F[u] = \big(1 - S'[u]\big)\, \nabla MA_M[u] + S'[u] \nabla MA_S[u].$$

The derivative of the filter (5.6) is given by

$$S'(x) = \begin{cases} 1 & |x| < 1 \\ -1 & 1 < |x| < 2 \\ 0 & |x| > 2. \end{cases}$$

However, allowing this derivative to take on negative values can lead to poorly conditioned or ill-posed linear systems. Instead, we approximate the Jacobian by

$$\tilde{\nabla} MA_F[u] = \big(1 - S'[u]\big)\, \nabla MA_M[u] + \max\{S'[u], 0\} \nabla MA_S[u].$$

5.3 Computational Results–Two Dimensions

In this section, we present computational results for the hybrid and filtered schemes. In the implementation of the filtered scheme, we have fixed the parameter $\epsilon = \sqrt{h} + d\theta/10$. For brevity, we only present results on a 17 point stencil. Computations were also performed on the 9 and 33 point stencils, but these results do not affect our qualitative observations. We compare these results to the results obtained using the monotone method (also on a 17 point stencil) and the standard finite differences. As in the previous chapters, we present detailed results for the four representative examples of §2.5.

5.3.1 Accuracy

We begin by looking at the numerical accuracy of the hybrid methods. Numerical errors are presented in Table 5.1 and Figure 5.2. To assist in the interpretation of our results, we are also interested in knowing which scheme (monotone or standard) is active in the hybrid or filtered discretisations. This information is presented in Figure 5.3. In these pictures, yellow indicates that the value of the scheme is equal to the value of the standard scheme. Green indicates that the value is given by the value of the monotone scheme. Intermediate colours indicate that the value of the filtered scheme is between the values of the two component schemes.

C^2 Example (2.17)

N	Maximum Error			
	Monotone	Hybrid	Filtered	Standard
31	9.12×10^{-5}	6.76×10^{-5}	4.54×10^{-5}	4.54×10^{-5}
45	5.36×10^{-5}	3.00×10^{-5}	2.11×10^{-5}	2.11×10^{-5}
63	3.42×10^{-5}	1.46×10^{-5}	1.06×10^{-5}	1.06×10^{-5}
89	2.30×10^{-5}	0.71×10^{-5}	0.53×10^{-5}	0.53×10^{-5}
127	1.67×10^{-5}	0.35×10^{-5}	0.26×10^{-5}	0.26×10^{-5}
181	1.34×10^{-5}	0.17×10^{-5}	0.13×10^{-5}	0.13×10^{-5}
255	1.17×10^{-5}	0.09×10^{-5}	0.06×10^{-5}	0.06×10^{-5}
361	1.08×10^{-5}	0.04×10^{-5}	0.03×10^{-5}	0.03×10^{-5}

C^1 Example (2.18)

N	Maximum Error			
	Monotone	Hybrid	Filtered	Standard
31	8.66×10^{-4}	6.62×10^{-4}	3.99×10^{-4}	3.78×10^{-4}
45	6.84×10^{-4}	3.70×10^{-4}	2.03×10^{-4}	1.82×10^{-4}
63	6.82×10^{-4}	2.75×10^{-4}	1.40×10^{-4}	1.34×10^{-4}
89	6.51×10^{-4}	1.98×10^{-4}	1.03×10^{-4}	0.85×10^{-4}
127	6.63×10^{-4}	1.68×10^{-4}	0.76×10^{-4}	0.59×10^{-4}
181	6.62×10^{-4}	1.19×10^{-4}	0.56×10^{-4}	0.37×10^{-4}
255	6.58×10^{-4}	0.85×10^{-4}	0.46×10^{-4}	—
361	6.62×10^{-4}	0.60×10^{-4}	0.31×10^{-4}	—

Example with blow-up (2.19)

N	Maximum Error			
	Monotone	Hybrid	Filtered	Standard
31	1.74×10^{-3}	1.74×10^{-3}	1.74×10^{-3}	17.38×10^{-3}
45	0.98×10^{-3}	0.98×10^{-3}	0.98×10^{-3}	14.74×10^{-3}
63	0.59×10^{-3}	0.59×10^{-3}	0.59×10^{-3}	12.62×10^{-3}
89	0.37×10^{-3}	0.35×10^{-3}	0.35×10^{-3}	10.72×10^{-3}
127	0.35×10^{-3}	0.20×10^{-3}	0.20×10^{-3}	9.04×10^{-3}
181	0.34×10^{-3}	0.12×10^{-3}	0.12×10^{-3}	7.61×10^{-3}
255	0.33×10^{-3}	0.07×10^{-3}	0.13×10^{-3}	6.43×10^{-3}
361	0.33×10^{-3}	0.04×10^{-3}	0.13×10^{-3}	5.42×10^{-3}

$C^{0,1}$ (Lipschitz) Example (2.20)

N	Maximum Error			
	Monotone	Hybrid	Filtered	Standard
31	3.57×10^{-3}	3.57×10^{-3}	4.16×10^{-3}	5.19×10^{-3}
45	3.42×10^{-3}	3.42×10^{-3}	2.60×10^{-3}	3.82×10^{-3}
63	3.49×10^{-3}	3.49×10^{-3}	2.82×10^{-3}	2.86×10^{-3}
89	3.44×10^{-3}	3.44×10^{-3}	2.90×10^{-3}	2.12×10^{-3}
127	3.45×10^{-3}	3.45×10^{-3}	2.83×10^{-3}	1.54×10^{-3}
181	3.70×10^{-3}	3.70×10^{-3}	3.02×10^{-3}	1.12×10^{-3}
255	3.46×10^{-3}	3.46×10^{-3}	3.06×10^{-3}	—
361	3.45×10^{-3}	3.45×10^{-3}	3.21×10^{-3}	—

Table 5.1: Accuracy for the 17 point monotone, hybrid, filtered, and standard discretisations for four representative examples.

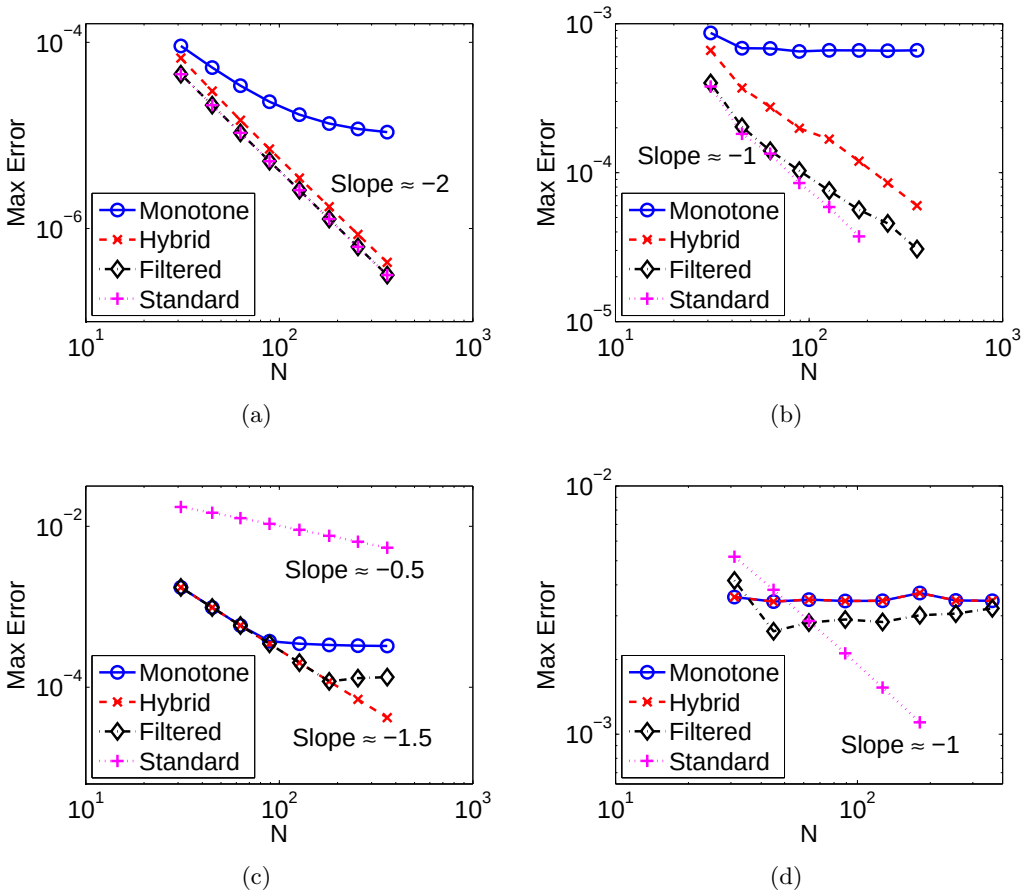


Figure 5.2: Error of the 17 point monotone, hybrid, filtered, and standard discretisations on the (a) C^2 example, (b) C^1 example, (c) example with blow-up, and (d) Lipschitz example.

Our qualitative observations differ somewhat depending on the regularity of the particular problem so we discuss each example in turn.

The C^2 solution (2.17)

The standard finite difference schemes gives $\mathcal{O}(h^2)$ accuracy. In this case, the hybrid scheme is slightly less accurate (though it still exhibits approximately $\mathcal{O}(h^2)$ accuracy). This happens because the monotone scheme is used near the non-strictly convex boundary as a precaution. Because the filtered scheme is allowed to use the more accurate discretisation right up to the boundary, it achieves the same accuracy as the standard scheme. Both the

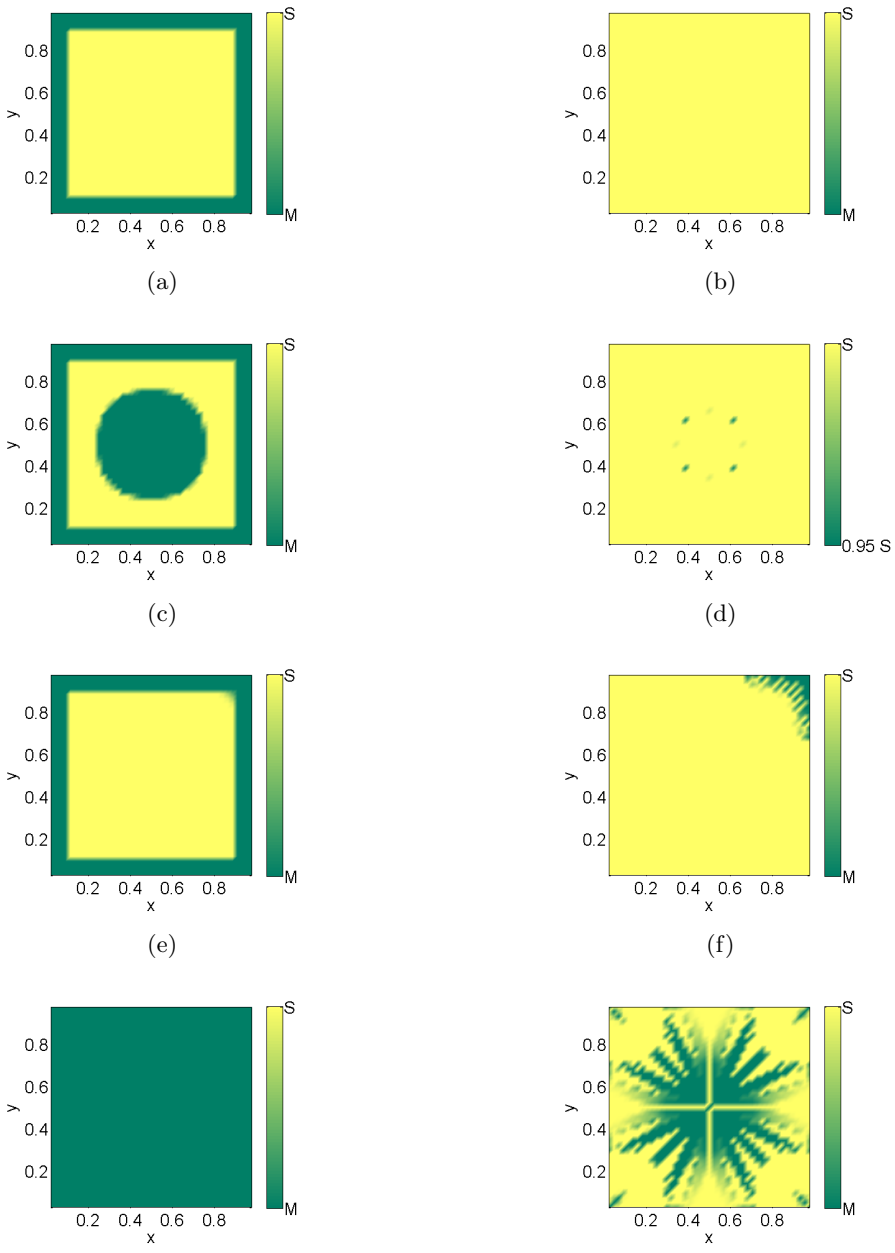


Figure 5.3: The discretisation that is active in the hybrid and filtered schemes for the (a),(b) C^2, (c),(d) C^1, (e),(f) blow-up, and (g),(h) Lipschitz examples.

hybrid and filtered schemes represent a clear improvement over the monotone scheme, which had its accuracy limited by the width of the stencil.

The C^1 solution (2.18)

This solution is non-smooth around a circle, so there is no reason to expect the second-order accuracy that was possible on the smooth solution. In fact, we find that the accuracy for the hybrid scheme is about $\mathcal{O}(h)$, which is similar to the standard discretisation. However, the absolute error is somewhat larger than the accurate scheme due to the fact that the monotone scheme is used in the interior of the circle and at points near the boundary, where the solution is in fact smooth. From Figure 5.3(d), we see that the filtered scheme applies the standard scheme at most of these points, with a small weight assigned to the monotone scheme at some points around the circle. (Note that the scale on this image is different than the scale on the other images. Without this change in scale, it is difficult to see the small weights assigned to the monotone discretisation.) This results in a lower absolute error than the hybrid scheme could achieve.

It is worth noting that the precise accuracy of the filtered scheme will depend on our choice of the parameter ϵ, which determines the allowable deviation from the value of the monotone scheme. We have chosen to set this value to

$$\epsilon = \sqrt{h} + \frac{1}{10} d\theta.$$

By changing this scaling (for example, by allowing ϵ to scale with $\sqrt{d\theta}$), we can allow for greater deviations from the monotone scheme, which could improve solution accuracy by permitting the use of the accurate scheme in a larger reason. For this particular example, where the standard discretisation appears to converge to the correct solution, this approach would probably improve the accuracy of the filtered scheme. In general, however, increasing the value of ϵ too much could also make it possible to use the standard scheme near a singularity, where it could instead cause a decrease in accuracy.

Both the hybrid and the filtered schemes again allow for a big improvement over the limited accuracy of the monotone scheme.

The blow-up solution (2.19)

In this case, the accuracy of the hybrid scheme is $\mathcal{O}(h^{1.5})$, which is much better than the accuracy of both the standard discretisation, which was only $\mathcal{O}(h^{0.5})$, and the monotone

scheme, which is limited by the stencil width.

The accuracy of the filtered scheme is better than the accuracy of the monotone scheme, but still appears to be limited by the width of the stencil. This is caused by our choice of the parameter ϵ, which scales like $\sqrt{h}$ in these computations. Given our observation that the accuracy of the standard scheme is only $\sqrt{h}$, it is unreasonable to expect the values of the standard and monotone schemes to differ by less than $\sqrt{h}$. As a result, the filtered scheme may reduce to the monotone scheme even in regions where the solution is smooth. By increasing the value of ϵ, we can improve the accuracy of the filtered scheme.

The cone solution (2.20)

For this singular example, the hybrid scheme is identical to the monotone scheme (since the right-hand side is either 0 or very large everywhere in the domain). Consequently, the angular resolution (stencil width) limits the accuracy of solutions. The singularity also limits the accuracy we can achieve with the filtered scheme. Since this solution is so singular (in fact, it is not even a viscosity solution), the reduced accuracy is to be expected.

5.3.2 Computation Time

Next we look at the computation times for the hybrid and filtered schemes. The incorporation of the monotone discretisation into these more accurate schemes appears to be enough to ensure the stability of Newton's method. In Chapter 4, we saw that the monotone Newton's method performed much more quickly than our other two-dimensional methods. We now want to verify that the computation time required by Newton's method is not adversely affected by the use of a hybrid or filtered scheme.

Computation times for the 17 point schemes are presented in Table 5.2 and Figure 5.4. As we had hoped, the computation times appear to be essentially the same for all three of the monotone, hybrid, and filtered schemes.

5.3.3 Gradient Maps

At this point, we recall that one of the motivations for solving the Monge-Ampère equation was to solve various mapping problems. With this goal in mind, it is important that not

C^2 Example (2.17)

N	Newton Iterations			CPU Time (seconds)		
	Monotone	Hybrid	Filtered	Monotone	Hybrid	Filtered
31	3	3	2	0.1	0.1	0.1
45	3	3	2	0.3	0.3	0.2
63	3	3	2	0.6	0.6	0.5
89	3	3	2	1.2	1.2	1.1
127	3	3	2	2.6	2.4	2.0
181	3	3	2	6.6	5.9	4.6
255	3	3	2	14.6	12.5	9.9
361	3	3	2	33.6	28.2	22.5

C^1 Example (2.18)

N	Newton Iterations			CPU Time (seconds)		
	Monotone	Hybrid	Filtered	Monotone	Hybrid	Filtered
31	3	2	2	0.1	0.1	0.1
45	3	3	3	0.3	0.3	0.3
63	4	3	2	0.7	0.6	0.5
89	5	4	3	1.8	1.5	1.2
127	4	5	3	3.3	3.8	2.6
181	4	4	4	7.9	7.6	7.0
255	5	5	3	20.6	19.1	12.9
361	6	5	6	60.4	48.3	50.6

Example with blow-up (2.19)

N	Newton Iterations			CPU Time (seconds)		
	Monotone	Hybrid	Filtered	Monotone	Hybrid	Filtered
31	6	6	7	0.2	0.3	0.3
45	6	6	6	0.5	0.6	0.6
63	9	9	9	1.5	1.4	1.5
89	7	7	7	2.8	2.7	2.6
127	11	11	11	9.1	8.6	8.4
181	7	7	8	15.5	14.2	15.0
255	7	7	8	35.2	30.5	32.4
361	11	11	12	122.2	101.5	108.7

$C^{0,1}$ (Lipschitz) Example (2.20)

N	Newton Iterations			CPU Time (seconds)		
	Monotone	Hybrid	Filtered	Monotone	Hybrid	Filtered
31	6	6	7	0.2	0.2	0.2
45	6	6	7	0.5	0.5	0.6
63	6	6	9	1.1	1.0	1.4
89	7	7	9	2.4	2.4	2.9
127	8	8	8	6.4	6.6	6.4
181	8	8	9	17.0	17.3	15.4
255	9	9	10	46.6	47.1	38.2
361	10	10	9	155.6	155.8	81.7

Table 5.2: Computation times for the 17 point monotone, hybrid, and filtered Newton's methods.

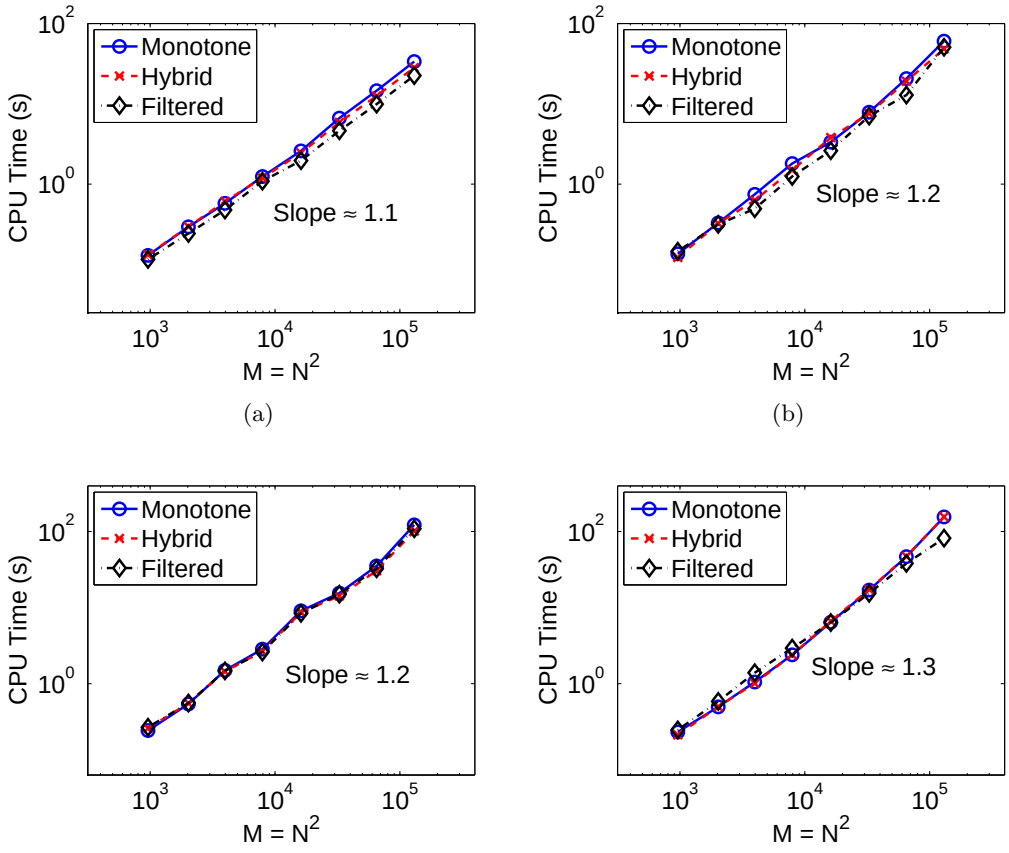


Figure 5.4: Computation times for the 17 point monotone, hybrid, and filtered Newton's methods for the (a) C^2 example, (b) C^1 example, (c) example with blow-up, and (d) Lipschitz example.

only the solutions of the equation, but also their gradients, are obtained accurately. In particular, it is critical that the gradient map be monotone.

In Figure 5.5 the solutions and corresponding gradient maps for the first three representative examples are presented. For example (2.20), the gradient map is too singular to illustrate. To visualise the maps,we show the image of a Cartesian mesh under the mapping

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} \mathcal{D}_x u \\ \mathcal{D}_y u \end{pmatrix},$$

where $(\mathcal{D}_x u, \mathcal{D}_y u)$ is the numerical gradient of the solution of the Monge-Ampère equation. In some cases, the image of a circle is plotted for visualisation purposes; the equation was actually solved on a square. For reference, the identity mapping is also displayed.

In each case, the computed map agrees with the gradient map coming from the exact solution.

5.4 Computational Results–Three Dimensions

In this section, we demonstrate the speed and accuracy of the hybrid Newton's method for three dimensional problems. These computations are performed on an $N \times N \times N$ grid on the square $[0,1]^3$. The monotone scheme used a 19 point stencil.

As before, we provide specific results for three representative examples of varying regularity, which are described in §2.5. Although the results are obtained on fairly coarse grids (up to $45 \times 45 \times 45$), Figure 5.6 suggests trends similar to what we saw in the two-dimensional case. In particular, the filtered and hybrid schemes lead to an improvement over the limited accuracy that is possible with the narrow-stencil monotone scheme. We also find that, as in the two-dimensional case, the computation time is essentially the same for all three schemes.

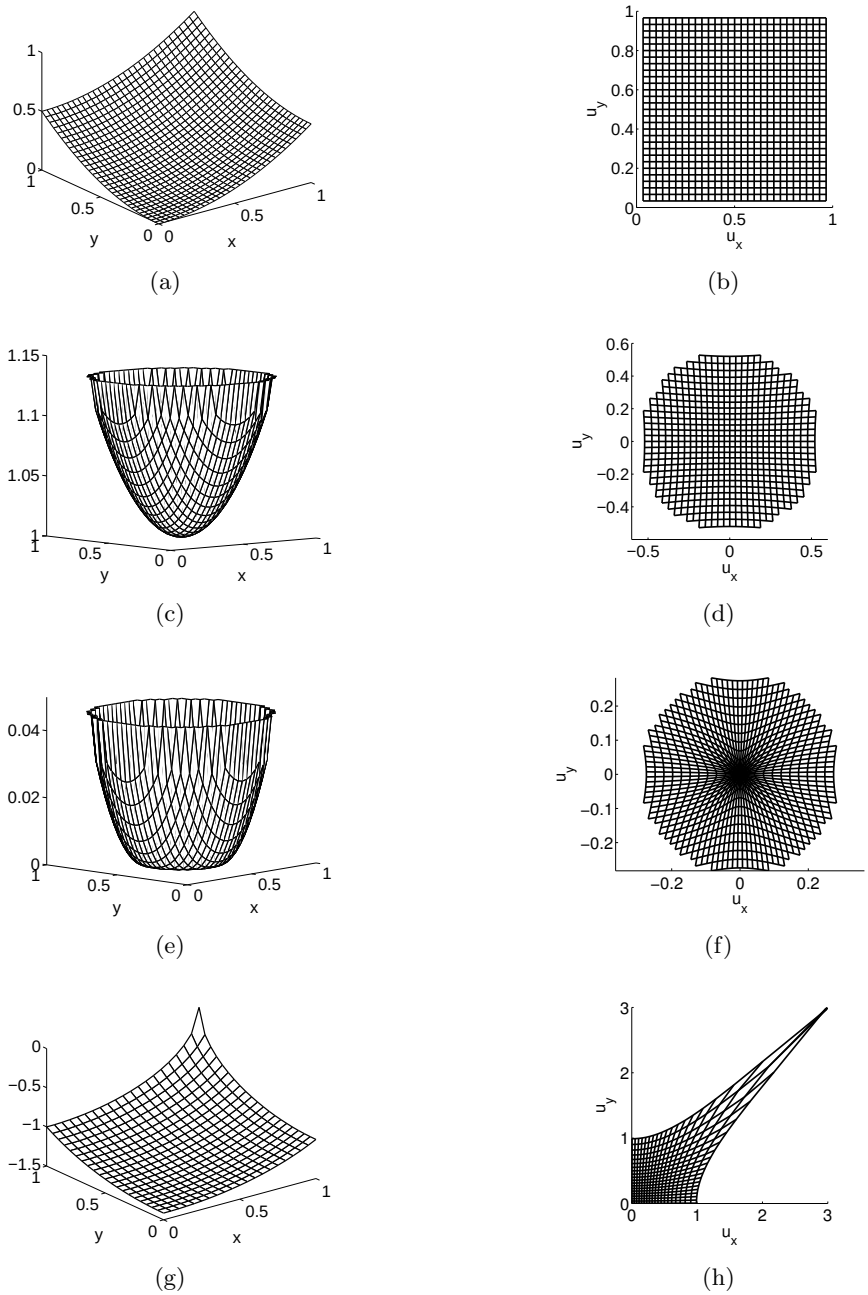


Figure 5.5: Solutions and mappings for the (a),(b) identity map, (c),(d) C^2 example, (e),(f) C^1 example, and (g),(h) example with blow-up.

C^2 Example (2.21)

N	Max Error		
	Monotone	Hybrid	Filtered
7	1.46×10^{-3}	1.43×10^{-3}	1.24×10^{-3}
11	0.67×10^{-3}	0.58×10^{-3}	0.46×10^{-3}
15	0.42×10^{-3}	0.29×10^{-3}	0.24×10^{-3}
21	0.27×10^{-3}	0.14×10^{-3}	0.12×10^{-3}
31	0.22×10^{-3}	0.06×10^{-3}	0.05×10^{-3}
45	0.20×10^{-3}	0.03×10^{-3}	0.02×10^{-3}

C^1 Example (2.22)

N	Max Error		
	Monotone	Hybrid	Filtered
7	5.29×10^{-3}	5.01×10^{-3}	3.82×10^{-3}
11	4.04×10^{-3}	3.82×10^{-3}	2.69×10^{-3}
15	3.15×10^{-3}	2.61×10^{-3}	1.03×10^{-3}
21	2.78×10^{-3}	1.78×10^{-3}	0.72×10^{-3}
31	2.52×10^{-3}	1.35×10^{-3}	0.41×10^{-3}

Example with Blow-up (2.23)

N	Max Error		
	Monotone	Hybrid	Filtered
7	7.11×10^{-3}	7.09×10^{-3}	6.38×10^{-3}
11	5.29×10^{-3}	5.38×10^{-3}	5.32×10^{-3}
15	4.62×10^{-3}	4.12×10^{-3}	4.36×10^{-3}
21	4.22×10^{-3}	3.43×10^{-3}	3.90×10^{-3}
31	4.03×10^{-3}	2.84×10^{-3}	3.86×10^{-3}

Table 5.3: Accuracy for the monotone, hybrid, and filtered discretisations for three representative three-dimensional examples.

C^2 Example (2.21)

N	Iterations			CPU Time (seconds)		
	Monotone	Hybrid	Filtered	Monotone	Hybrid	Filtered
7	3	3	3	0.1	0.1	0.1
11	2	2	2	0.1	0.1	0.1
15	3	3	2	0.4	0.4	0.3
21	3	3	3	1.8	1.5	1.4
31	4	4	2	20.2	17.6	8.7
45	4	5	5	242.0	204.9	192.6

C^1 Example (2.22)

N	Iterations			CPU Time (seconds)		
	Monotone	Hybrid	Filtered	Monotone	Hybrid	Filtered
7	5	5	4	0.1	0.2	0.1
11	8	10	4	0.3	0.3	0.1
15	8	10	6	0.9	1.0	0.6
21	8	7	6	4.2	3.4	2.6
31	6	8	7	34.6	37.9	29.5

Example with Blow-up (2.23)

N	Iterations			CPU Time (seconds)		
	Monotone	Hybrid	Filtered	Monotone	Hybrid	Filtered
7	4	4	4	0.03	0.04	0.08
11	8	10	10	0.22	0.29	0.29
15	6	6	6	0.77	0.61	0.66
21	10	10	10	5.67	4.73	4.58
31	14	11	14	79.02	48.66	56.83

Table 5.4: Computation times for the monotone, hybrid, and filtered Newton's methods in three-dimensions.

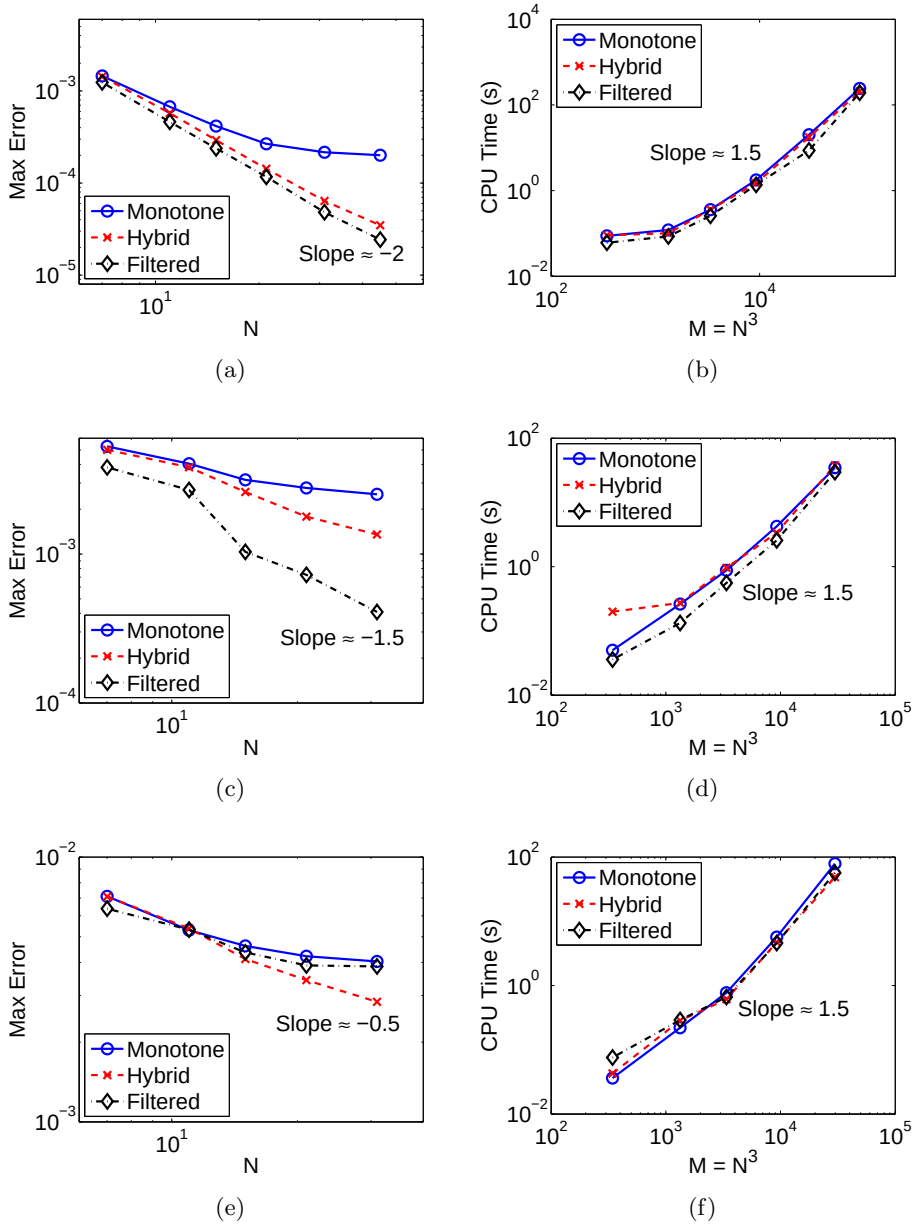


Figure 5.6: Maximum error and computation times for the monotone, hybrid, and filtered schemes on the three-dimensional (a),(b) C^2 example, (c),(d) C^1 example, and (e),(f) example with blow-up.

Chapter 6

Optimal Transport

In this chapter, we turn our attention to an important application of the elliptic Monge-Ampère equation: the L^2 optimal mass transport problem. After reviewing the special boundary conditions that arise in this setting, we propose a method for solving the transport problem by solving a sequence of Monge-Ampère equations with Neumann boundary conditions. We conclude this chapter by providing several challenging and representative computational examples from optimal transport.

6.1 Transport Boundary Conditions

In this section, we discuss the transport boundary conditions in more detail. We describe a method for solving this challenging problem by solving a sequence of more tractable subproblems; these are Monge-Ampère equations subject to Neumann boundary conditions.

6.1.1 Nonlinear Boundary Conditions

In the problem of L^2 optimal transport between convex sets $X, Y \in \mathbb{R}^d$, the transport condition (1.7)

$$\nabla u : X \to Y,$$

also known as the second boundary value problem, can be enforced by simply requiring the boundary points to map to boundary points [77, 84, 86]:

$$\nabla u : \partial X \to \partial Y.$$

In particular, if the boundary of the region Y is defined by the function

$$\Phi(y) = 0,$$

we can write the transport boundary condition as

$$\Phi(\nabla u(x)) = 0, \quad x \in \partial X. \tag{6.1}$$

While we might try simply enforcing this nonlinear equation at boundary points, the function ϕ can be highly nonlinear and non-smooth. As a result, it will be difficult to construct a discretisation that is consistent with the boundary condition even when solutions are singular. Additionally, we want to ensure that the discretisation we use permits fast solvers to remain stable. As was the case for standard schemes for the Monge-Ampère equation (§3.2.3), we expect that a naive discretisation of the boundary condition could affect the stability of Newton's Method.

6.1.2 Mapping Between Rectangles

The situation simplifies significantly if we are simply mapping a rectangle to a rectangle. In this case, since the optimal L^2 mapping does not permit twisting or rotation, we expect the four sides of the rectangle X to map to the corresponding sides of the rectangle Y.

As a concrete example (see Figure 6.1), suppose that the sets $X, Y \in \mathbb{R}^2$ are defined as

$$X = [0,1] \times [0,1], \quad Y = [0,1] \times [0,1].$$

Then, for example, we expect the function $\nabla u(x)$ to map the segment $x_1 = 0$, $x_2 \in [0,1]$ to the segment $y_1 = 0$, $y_2 \in [0,1]$. That is,

$$u_{x_1}(0, x_2) = 0.$$

Similarly, we will have

$$u_{x_1}(1, x_2) = 1, \quad u_{x_2}(x_1, 0) = 0, \quad u_{x_2}(x_2, 1) = 1.$$

This is simply a (linear) Neumann boundary condition, which is straightforward to implement [4, 72].

Given the ease with which we can explicitly express the optimal transport boundary condition for maps between rectangles, a natural solution for more general geometries would

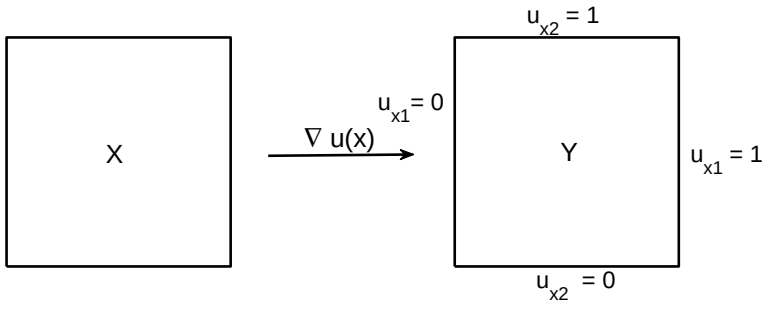


Figure 6.1: Mapping between rectangles.

be to simply embed the sets X and Y into squares. The optimal map will not change as long as we do not allow the addition of any mass. This is easily accomplished by extending the density functions as follows:

$$\tilde{f}(x) = \begin{cases} f(x), & x \in X \\ 0, & x \notin X \end{cases} \qquad \tilde{g}(y) = \begin{cases} g(y), & y \in Y \\ 0, & y \notin Y. \end{cases}$$

However, a problem is immediately evident when we recall that we are solving the PDE

$$\det(D^2 u(x)) = f(x)/g(\nabla u(x)),$$

which involves division by the density function g. Clearly, we must ensure that $g(y)$ remains positive. In addition, we recall that the convergence of our monotone discretisation is dependent on $g(y)$ being a positive Lipschitz function (Theorem 4.10).

A simple solution would be to simply regularise the density functions slightly to ensure that they are strictly positive and Lipschitz continuous. However, an approximation to a discontinuous density function will still have a large Lipschitz constant. We also recall that the formal consistency error of the monotone scheme is affected by the Lipschitz constant K_F of the right-hand side since convergence requires the regularisation parameter δ to satisfy

$$\delta^{d-1} \geq K_F \left|\nu_j\right| h/2.$$

While in theory we can still establish the convergence of the finite difference scheme, in practice the grid will have to be extremely well-refined before we are able to achieve meaningful results.

We conclude that while the idea of extending the density functions into a square is simple, it is not practical from a computational standpoint. Thus a more sophisticated method for implementing the transport boundary conditions is desirable.

6.1.3 A Sequence of Neumann Boundary Conditions

Given the appearance of the gradient in the transport boundary condition (6.1) and the simplicity of implementing a Neumann boundary condition, we would like to find the Neumann boundary condition

$$\frac{\partial u}{\partial \mathbf{n}} = \phi(x), \quad x \in \partial X$$

for the Monge-Ampère equation that is equivalent to solving the more challenging problem (1.3), (1.7), (1.1). Here the vector $\mathbf{n}$ refers to the unit outward normal vector at each point $x \in \partial X$.

It is not at all apparent from (1.7) what the equivalent Neumann boundary condition should be. However, we suggest a sequence of Neumann boundary conditions that can be used to numerically determine the correct function ϕ.

We first recall that the gradient of the exact solution u maps the boundary of the set X to the boundary of Y

$$\nabla u : \partial X \to \partial Y$$

and that the correct Neumann condition is given by

$$\phi(x) = \nabla u(x) \cdot \mathbf{n}(x), \quad x \in \partial X.$$

To find this function, we suppose that we have a convex approximation u^k to the solution of the Monge-Ampère transport problem. Then the (sub-)gradient of this function will map the domain X onto some set $Y^k \in \mathbb{R}^d$ and, since u^k is convex,

$$\nabla u^k : \partial X \to \partial Y^k.$$

In reality, we would like the image of the gradient to be ∂Y, the boundary of the target set. This motivates us to consider the projection of $\partial Y^k = \nabla u^k(\partial X)$ onto the correct set of boundary points ∂Y:

$$\text{Proj}_{\partial Y}(\nabla u^k(x)) = \underset{y \in \partial Y}{\text{argmin}} \, \|y - \nabla u^k(x)\|_2^2, \quad x \in \partial X.$$

From this we extract a new Neumann boundary condition

$$\phi^k(x) = \text{Proj}_{\partial Y}(\nabla u^k(x)) \cdot \mathbf{n}$$

and solve the Monge-Ampère equation once again with this updated boundary condition to obtain a new approximation u^{k+1}.

To summarize, we iterate to produce a sequence of functions $(u^1, u^2, \ldots)$ obtained by solving the Monge-Ampère equation

$$\begin{cases} \det(D^2 u^{k+1}(x)) = f(x)/g(\nabla u^{k+1}(x)), & x \in X \\ \nabla u^{k+1}(x) \cdot \mathbf{n}(x) = \text{Proj}_{\partial Y}(\nabla u^k(x)) \cdot \mathbf{n} \equiv \phi^k(x), & x \in \partial X \\ u^{k+1} \text{ is convex.} \end{cases} \tag{6.2}$$

We make the important observation that these boundary conditions do *not* pin down the values of ∇u^{k+1} on the boundary. This would be a mistake since we know only that $\nabla u : \partial X \to \partial Y$ and not the exact values of $\nabla u(x)$ on the boundary. Instead, each Neumann condition fixes only one component of the gradient (the normal component) and allows the remaining component(s) to slide as needed to ensure that the Monge-Ampère equation is satisfied.

6.1.4 Solvability of Sub-problems

We note that the iteration (6.2) may not be well-posed. The problem here is that, while the Monge-Ampère equation with the correct Neumann values $\phi(x)$ has a solution, the sub-problems we have described may not be solvable.

One important point to note is that for the Monge-Ampère equation with Neumann boundary conditions, a solution (unique up to an additive constant) does not exist for general data. This is analogous to the Neumann problem for the linear Poisson equation:

$$\begin{cases} \Delta u(x) = f(x), & x \in X \\ \nabla u(x) \cdot \mathbf{n}(x) = \psi(x), & x \in \partial X. \end{cases}$$

If we integrate the forcing f over the domain, we find via integration by parts that

$$\begin{aligned}\int_X f(x) &= \int_X \Delta u(x) \\ &= \int_{\partial X} \nabla u(x) \cdot \mathbf{n}(x) \\ &= \int_{\partial X} \psi(x).\end{aligned}$$

Thus the Neumann problem will not have a solution unless that data satisfies the solvability condition

$$\int_X f(x) = \int_{\partial X} \psi(x).$$

For the Monge-Ampère equation with Neumann boundary conditions, we are not aware of an explicit representation of the corresponding solvability condition. However, it is true that the problem:

$$\begin{cases} \det(D^2 u) = f(x)/g(\nabla u(x)), & x \in X \\ \nabla u(x) \cdot \mathbf{n}(x) = \psi(x), & x \in \partial X \\ u \text{ is convex}, \end{cases}$$

has a solution (unique up to an additive constant) only if an implicit solvability condition is satisfied [62].

Even if the problem we are given is well-posed, the system of discretised equations may not be well-posed: numerical error can mean that the solvability conditions for the continuous and discrete problems are slightly different. To get around this problem, we will instead solve an equation of the form

$$\begin{cases} \det(D^2 u) = cF(x, \nabla u(x)), & x \in X \\ \nabla u(x) \cdot \mathbf{n}(x) = \psi(x), & x \in \partial X \\ u \text{ is convex}, \\ \int_X u \, dx = 0 \end{cases}$$

for the unknowns $c > 0$ and $u(x)$, where the constant c is chosen to ensure the equation has a solution and the mean-zero condition forces the solution to be unique (instead of unique up to an additive constant).

Of course, if we are given the correct Neumann values $\phi(x)$ for the solution to the transport problem, the constant c will simply be equal to one. However, by relaxing this condition we make it possible to solve the sub-problems when the solvability condition requires c to be slightly different than one.

To summarize, we solve the transport problem by performing the iteration

$$\begin{cases} \det(D^2 u^{k+1}(x)) = c^{k+1} f(x)/g(\nabla u^{k+1}(x)), & x \in X \\ \nabla u^{k+1}(x) \cdot \mathbf{n}(x) = \mathrm{Proj}_{\partial Y}(\nabla u^k(x)) \cdot \mathbf{n} \equiv \phi^k(x), & x \in \partial X \\ u^{k+1} \text{ is convex,} \\ \int_X u^{k+1}\, dx = 0. \end{cases} \tag{6.3}$$

Although we do not present detailed computational results until §6.4, we do want to provide an idea of the sequence of maps that is produced using this method. We illustrate this by mapping a square with uniform density onto a circle with uniform density. The sequence of maps produced by this method is presented in Figure 6.2. We can see that this iteration successfully transforms a square mesh into a circular mesh in just a few iterations.

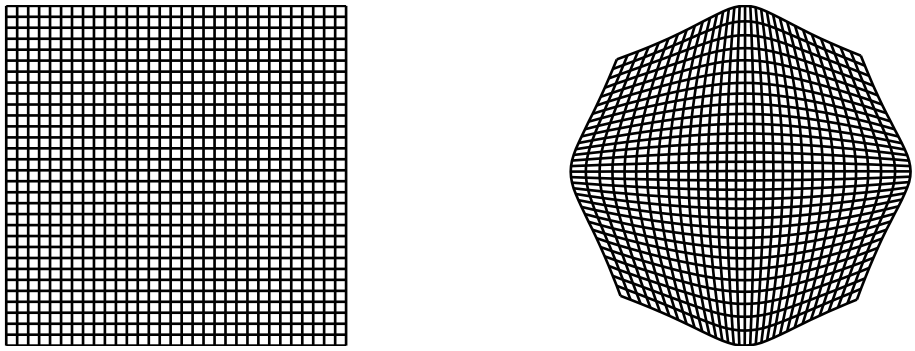
6.1.5 Extension of Target Density

Another point that needs to be addressed is the definition of the target density function $g(y)$ at points outside the target set Y. Of course, if we substitute the exact transport potential u into the Monge-Ampère equation

$$\det(D^2 u(x)) = f(x)/g(\nabla u(x)),$$

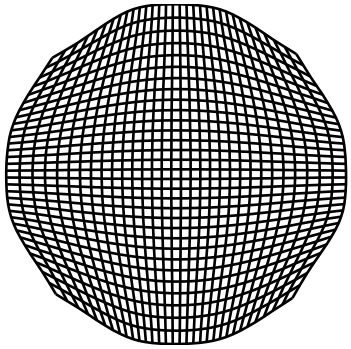
the gradient $\nabla u(x)$ will only give values in the set Y and g will only need to be defined in this set. However, in the course of computing the solution to the mass transport problem, we will have approximations that can map points in X to points outside of Y. Thus it is important that $g(y)$ is actually defined at these points.

From the viewpoint of optimal transport, the most natural option is to let the density $g(y)$ vanish outside the target Y since all mass is inside the target set. However, this is not practical from a computational standpoint since convergence of the Monge-Ampère solver requires the density function $g(y)$ to be strictly positive and Lipschitz continuous (Theorem 4.10 and §6.1.2).

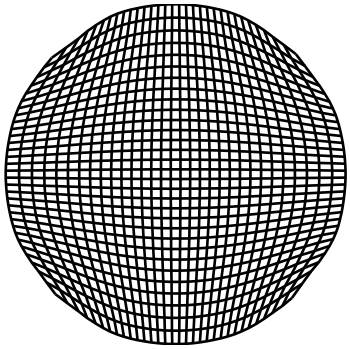


(a)

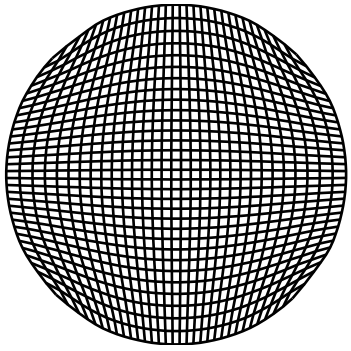
(b)



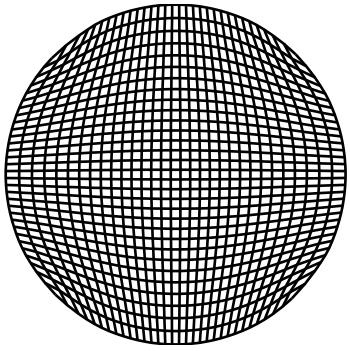
(c)



(d)



(e)



(f)

Figure 6.2: Mapping a square onto a circle using the iteration (6.3).

Instead, we allow for any positive, Lipschitz continuous extension of $g(y)$ into all space. In some cases, when g is a given function, there is an obvious way of extending it into all space. Lipschitz extensions can always be obtained by using, for example, the method of [71]. The resulting function $g^*(y)$ can also be bounded away from zero by considering $\max\{g^*(y), \epsilon\}$ for some $0 < \epsilon \leq \min_{y \in Y} g(y)$.

Of course, a positive extension of $g(y)$ means that the initial mass $\int_{\mathbb{R}^d} f(x)\,dx = \int_X f(x)\,dx$ will not be equal to the total final mass in all space $\int_{\mathbb{R}^d} g^*(y)\,dy$. However, the maps we compute will only take the set X into a bounded region that need not include the whole support of the extended density $g^*(y)$. This property, together with the scaling parameter c introduced in §6.1.4, ensures that mass is conserved in each step of our projection scheme.

6.2 Numerical Implementation

The biggest challenge in implementing this projection scheme is the solution of the Monge-Ampère equation at each iteration. This challenge is easily and efficiently handled using the finite difference schemes developed in the previous chapters. We now turn our attention to a few remaining computational details.

One important issue is the implementation of the Neumann boundary conditions since, in the previous chapters, we limited our attention to the Dirichlet problem.

The iterations we have described in this paper also need to be initialised. There are really two aspects to this: we need to initialise u and c each time we solve the Monge-Ampère equation and we also need to initialise our estimation of the boundary condition $\phi(x)$.

We also describe a simple method for computing in complicated domains without having to resort to more complicated finite difference stencils.

6.2.1 Implementation of Neumann Boundary Conditions

We begin by describing our numerical implementation of the Neumann boundary condition (1.6):

$$u_{\mathbf{n}}(x) = \phi(x), \quad x \in \partial X.$$

Here $\mathbf{n}$ denotes the unit outward normal to the boundary ∂X.

Our computational domain is the square, which means we must impose values for u_{x_1} on the left and right sides of the domain and for u_{x_2} on the top and bottom edges of the

domain.

We accomplish this by adding a layer of ghost points around the outside of our computational domain. The value of the normal derivatives on the boundary can then be discretised using simple centred differences. For example, at a point on the left edge ($x_1 = x_{\min}$), we can discretise the normal derivative as

$$u_{\mathbf{n}}(x) = \frac{1}{2h}(u(x_{\min} + h, x_2) - u(x_{\min} - h, x_2)).$$

The use of ghost points ensures that all values needed in this discretisation are available.

We also need to provide four more equations at the corner points in our grid. We specify the value of the derivative in the "diagonal" direction ($(1,1)$, $(1,-1)$, $(-1,1)$, or $(-1,-1)$) that points outward from the grid at each of these four points. This is enforced using centred differences. So, for example, at the points $(x_{1,\min}, x_{2,\min})$ we require that

$$\frac{1}{2\sqrt{2}h}(u(x_{1,\min} - h, x_{2,\min} - h) - (x_{1,\min} + h, x_{2,\min} + h)) = \\ - \frac{1}{\sqrt{2}}(u_{x_1}(x_{1,\min}, x_{2,\min}) + u_{x_2}(x_{1,\min}, x_{2,\min})).$$

As before, the ghost points ensure that all of these values are available.

6.2.2 Newton's Method

Because we have included the scaling factor c (which comes from the solvability condition) as an unknown, we must slightly adjust Newton's method to obtain this. We will now perform the iteration

$$u^{k+1} = u^k - v^k, \quad c^{k+1} = c^k - d^k$$

where the correctors v^k, d^k are obtained by solving the equation

$$\nabla MA[u^k, c^k](v^k, d^k)^T = MA[u^k, c^k].$$

As long as the initial iterate u^0 satisfies the given Neumann boundary condition, we can simply enforce a homogeneous Neumann condition on the corrector v^k at each step.

As usual, we obtain the Jacobian of the hybrid system via

$$\nabla MA[u, c] = w(x)\nabla MA_M[u, c] + (1 - w(x))\nabla MA_S[u, c].$$

We begin by computing the Jacobian of the monotone discretisation. We recall that this discretisation has the form

$$MA_M[u,c] = \min_{(\nu_1,\ldots,\nu_d)\in\mathcal{G}} G_{(\nu_1,\ldots,\nu_d)}[u,c].$$

By Danskin's Theorem [7], we can write the Jacobian of this as

$$\nabla MA_M[u,c] = \nabla G_{(\nu_1,\ldots,\nu_d)}[u,c],$$

where the $(\nu_1,\ldots,\nu)$ are the directions active in the minimum.

This Jacobian can be broken down into two basic components: the gradient with respect to the solution vector u and the gradient with respect to the scaling factor c. The first component is identical to what we computed in §4.8.5 except for the addition of the scaling factor c:

$$\begin{aligned}\nabla_{u_i} G_{(\nu_1,\ldots,\nu_d)}[u,c] = \sum_{m=1}^{d}\left[\left(\prod_{j\neq m}\max\{\mathcal{D}_{\nu_j\nu_j}u_i,\delta\}\right)\mathbb{1}_{\mathcal{D}_{\nu_j\nu_j}u_i\geq\delta} + \mathbb{1}_{\mathcal{D}_{\nu_j\nu_j}u_i<\delta}\right]\mathcal{D}_{\nu_m\nu_m}\\ - c\sum_{m=1}^{d}\frac{\partial F}{\partial p_m}\left(x,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_1}{|\nu_j|}\mathcal{D}_{\nu_j}u_i,\ldots,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_d}{|\nu_j|}\mathcal{D}_{\nu_j}u_i\right)\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_m}{|\nu_j|}\mathcal{D}_{\nu_j}.\end{aligned}$$

The final component is given by

$$\nabla_c G_{(\nu_1,\ldots,\nu_d)}[u,c] = -F\left(x,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_1}{|\nu_j|}\mathcal{D}_{\nu_j}u_i,\ldots,\sum_{j=1}^{d}\frac{\nu_j\cdot\mathbf{e}_d}{|\nu_j|}\mathcal{D}_{\nu_j}u_i\right).$$

For the standard discretisation, the first component of the Jacobian (in two dimensions) is simply

$$\begin{aligned}\nabla_{u_i} MA_S[u,c] = (\mathcal{D}_{x_2x_2}u_i)\mathcal{D}_{x_1x_1} + (\mathcal{D}_{x_1x_1}u_i)\mathcal{D}_{x_2x_2} + 2(\mathcal{D}_{x_1x_2}u_i)\mathcal{D}_{x_1x_2}\\ - c\frac{\partial F}{\partial p_1}(x,\mathcal{D}_{x_1}u_i,\mathcal{D}_{x_2}u_i)\mathcal{D}_{x_1} - c\frac{\partial F}{\partial p_2}(x,\mathcal{D}_{x_1}u_i,\mathcal{D}_{x_2}u_i)\mathcal{D}_{x_2}\end{aligned}$$

and the second component is

$$\nabla_c MA_S[u,c] = -F(x,\mathcal{D}_{x_1}u_i,\mathcal{D}_{x_2}u_i).$$

6.2.3 Initialisation of Boundary Data

Next we discuss the initialisation of the boundary data ϕ^0 in the iteration (6.3). The simplest approach would be to extract boundary conditions from the identity map $s(x) = x$. However, if this mapping does not overlap with the target set Y, the iteration is likely to fail.

We can remedy this problem by instead extracting boundary data from the scaled identity map $s(x) = Mx$ where the constant M is chosen large enough that the set $s(X)$ encompasses the target set Y.

Once this constant is chosen, we simply choose the initial boundary condition

$$\phi^0(x) = Mx \cdot \mathbf{n}(x), \quad x \in \partial X.$$

We can accelerate the convergence of this method by first solving the transport problem on a coarser grid, then interpolating the resulting boundary data onto the refined mesh.

6.2.4 Initialisation of Newton's Method

We also need to initialise Newton's method each time we solve the Monge-Ampère equation. We can use the approach we have employed in previous chapters, which involves obtaining the initial guess by solving the equation

$$\Delta u(x) = (cd!f(x)/g(x - x_0))^{1/d}$$

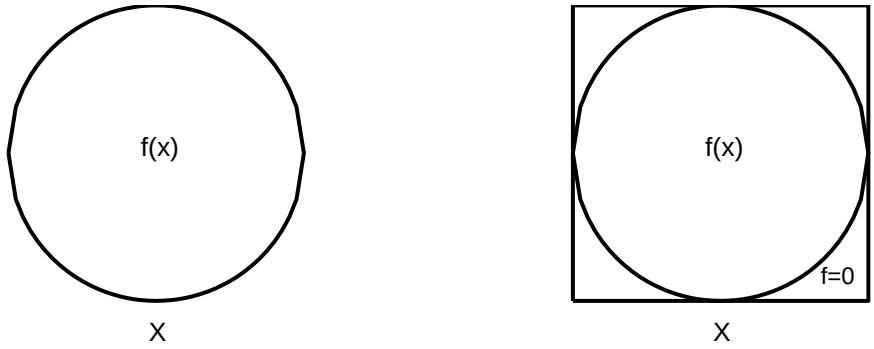
where x_0 is a point in the interior of the target set Y.

However, since we will be solving the Monge-Ampère equation multiple times with different boundary conditions, we can also accelerate the convergence of the $(k+1)^{st}$ iteration by initialising with the solution found during the previous solve (u^k). One important point here is that the boundary data changes from step to step. Thus it is important to change the values of u^k at the boundary points so as to ensure that correct boundary conditions are satisfied.

6.2.5 Computing in General Domains

When computing with finite difference methods, it is most convenient to work in rectangular domains. However, it is often desirable to solve the mass transport problem in more general domains. This motivates us to return to the idea of extending the density functions into a square, which was discussed in §6.1.2. We observed earlier that extending the densities

was not a practical option because of the large Lipschitz constant of the regularised version of the extended target density $\tilde{g}$. However, there is nothing to prevent us from using a vanishing or discontinuous initial density $\tilde{f}$. Because, in the context of optimal transport, the functions f, g appearing on the right-hand side are really density functions, computing in square domains is actually sufficient. More general domains can simply be embedded in a square, with the density function f set equal to zero outside the region of interest; see Figure 6.3. Because this approach leads to very degenerate Monge-Ampère equations, many of the currently available solvers for Monge-Ampère equations would not allow this option. However, we stress again that our finite difference solvers are equipped to enforce the non-strict convexity and correctly approximate the possibly singular solutions that can result in this degenerate setting.



(a) (b)

Figure 6.3: (a) A non-rectangular domain can simply be (b) embedded in a square.

6.3 Computational Results: Mapping Between Rectangles

We now provide computational results for several different examples. We begin by considering mappings between rectangles. In this case, our projection scheme reduces to a simple Neumann boundary condition (§6.1.2). This allows us to focus on the correctness of our discretisation, which must now deal with a right-hand side that depends on gradients.

In each example, our domain is a square, which is discretised on an $N \times N$ grid using the 17 point hybrid scheme $(MA)^H$. As in earlier chapters, we let $h = 1/(N-1)$ denote

the spatial resolution of the grid and let $M = N^2$ denote the total number of grid points.

When an exact solution u^{exact} is available, we also provide the maximum error in the gradient map:

$$\text{Error} = \max\{\|u_{x_1}^{exact} - u_{x_1}\|_\infty, \|u_{x_2}^{exact} - u_{x_2}\|_\infty\}.$$

We also provide the total number of Newton iterations and computation time required for each example.

The examples we consider include:

- A (linear) map between Gaussian densities.
- A comparison between a map obtained by solving the direct problem and a map obtained by inverting the solution to the inverse problem.
- A map from a uniform density onto a density that blows up at a point.
- A map between two brain MRI images.

6.3.1 Gaussian Densities

We begin by showing that we can recover a linear mapping between two rectangles with Gaussian densities. We consider the problem of mapping the square $[-0.5, 0.5] \times [-0.5, 0.5]$ onto the rectangle $[0.5, 1.5] \times [-0.25, 0.25]$ with the density functions:

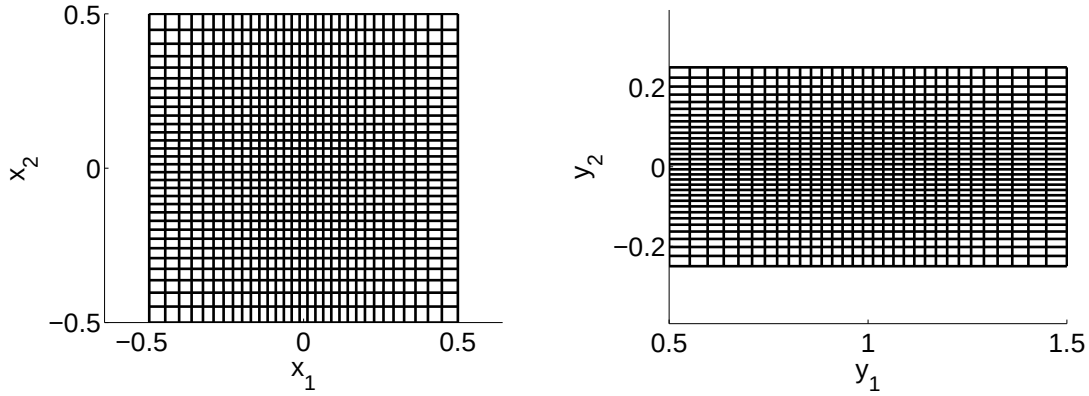
$$f(x_1, x_2) = \frac{1}{0.16} \exp\left(-\frac{1}{2}\frac{x_1^2}{0.4^2} - \frac{1}{2}\frac{x_2^2}{0.4^2}\right),$$

$$g(y_1, y_2) = \frac{1}{0.08} \exp\left(-\frac{1}{2}\frac{(y_1 - 1)^2}{0.4^2} - \frac{1}{2}\frac{y_2^2}{0.2^2}\right).$$

In this case, we have an explicit expression for the optimal map:

$$u_{x_1} = x_1 + 1, \quad u_{x_2} = \frac{1}{2}x_2.$$

We present the results in Table 6.1 and Figure 6.4. In this example, can actually achieve machine accuracy (if we take enough Newton steps). This is because the exact solution is simply a linear map, which will exactly solve the discretised system of equations. In addition to this, we find that the Newton solver for the Monge-Ampère equation converges in $\mathcal{O}(M)$ time.



(a) (b)

Figure 6.4: (a) A mesh with Gaussian density f and (b) its image under the gradient map ∇u (§6.3.1).

N	h	Newton Iterations	CPU Time (s)	Maximum Error
32	0.0323	1	0.1	5.71×10^{-8}
46	0.0222	1	0.2	3.34×10^{-8}
64	0.0159	1	0.3	0.26×10^{-8}
90	0.0112	1	0.6	0.18×10^{-8}
128	0.0079	1	1.1	0.13×10^{-8}
182	0.0055	1	2.4	0.09×10^{-8}
256	0.0039	1	5.3	0.07×10^{-8}
362	0.0028	1	12.4	0.05×10^{-8}

Table 6.1: Computation time and maximum error for the map between two Gaussian densities (§6.3.1).

6.3.2 Recovering an Inverse Map

For our next example, we consider another problem with an exact solution, which will be used to verify that we can correctly recover inverse maps. To set up this example, we define the function

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

Now we map the density

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

in the square $[-0.5, 0.5] \times [-0.5, 0.5]$ onto a uniform density in the same square. This transport problem has the exact solution

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

We will solve this problem in two ways:

- Directly, as in the previous example.
- By solving the inverse problem (mapping g to f) and inverting the resulting map.

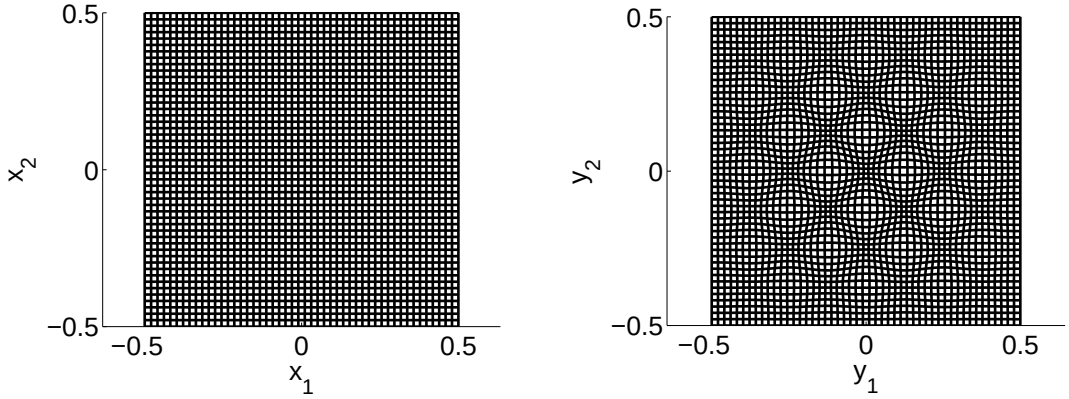
Results are presented in Figure 6.5 and Table 6.2. We find that the maps obtained from both the forward and inverse formulations have about $\mathcal{O}(h^2)$ accuracy. Both problems are solved in about $\mathcal{O}(M)$ time.

6.3.3 An Example with Blow-up

Next we consider the problem of mapping a uniform density onto a density that blows up at a point:

$$g(y_1, y_2) = \frac{\exp\left(-2\sqrt{(y_1 - 0.5)^2 + (y_2 - 0.5)^2}\right)}{\sqrt{(y_1 - 0.7)^2 + (y_2 - 0.7)^2}}.$$

In this case, both X and Y are the square $[0, 1] \times [0, 1]$. This example is taken from [32], which allows us to compare results. In this example, we slightly regularise the density g (bounding it by a $\mathcal{O}(1/h^2)$ function) to prevent infinities from appearing.



(a) (b)

Figure 6.5: (a) A uniform Cartesian mesh and (b) its image under the gradient map ∇u (§6.3.2).

	Forward Problem			Inverse Problem		
N	Iterations	Time (s)	Max Error	Iterations	Time (s)	Max Error
32	3	0.2	2.476×10^{-3}	4	0.4	2.450×10^{3}
46	2	0.2	0.631×10^{-3}	2	0.5	0.575×10^{3}
64	2	0.5	0.241×10^{-3}	2	1.1	0.244×10^{3}
90	1	0.6	0.106×10^{-3}	1	1.3	0.101×10^{3}
128	1	1.3	0.049×10^{-3}	1	2.9	0.048×10^{3}
182	1	2.9	0.024×10^{-3}	1	5.1	0.023×10^{3}
256	1	6.3	0.012×10^{-3}	1	10.9	0.011×10^{3}
362	1	14.0	0.006×10^{-3}	1	22.6	0.006×10^{3}

Table 6.2: Newton iterations, computation time and maximum error for a map obtained by a direct solve and by inverting the inverse map (§6.3.2).

We present the timing results in Table 6.3. We provide not only the number of Newton iterations and computation time, but also the ratio

$$R = \max\left\{g(y_1, y_2)/f(x_1, x_2)\right\},$$

since many currently available Monge-Ampère solvers can become slow or unstable when this ratio is large. For comparison, we provide the same information for the method of [32] (which is essentially our "standard" discretisation solved with an optimised Newton-Krylov method). The method of [32] runs in $\mathcal{O}(M)$ time. Our method, though it runs in about $\mathcal{O}(M^{1.1})$ time, has lower computation times and deals with larger density ratios. Naturally, we cannot conclude too much from the comparison of computation times since the computations were performed on different computers. However, it is evident that, in terms of computation time, our method is very competitive with other fast solvers.

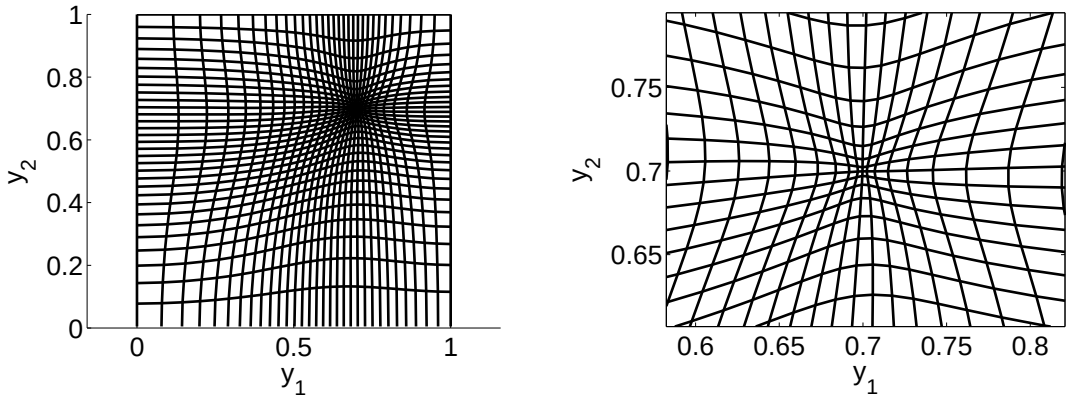
We also present the deformed mesh; see Figure 6.6. In addition, we zoom into the region of high density to verify that our method has produced an untangled mesh.

	Hybrid Method			Method of [32]		
N	R	Iterations	CPU Time (s)	R	Iterations	CPU Time (s)
32	546	4	0.2	356	6	1
46	1,151	4	0.3	—	—	—
64	2,254	5	0.8	1,127	7	4
90	4,066	5	1.6	—	—	—
128	9,162	5	3.5	2,829	7	17.4
182	18,608	5	8.3	—	—	—
256	36,933	5	19.4	8,886	7	70
362	74,018	4	36.3	—	—	—

Table 6.3: Ratio of density functions, Newton iterations, and total computation time for our hybrid method and the method of [32].

6.3.4 Mapping Between Brain MRI Images

We conclude this section with an example from image processing. In this example, we obtain our density functions from the pixel intensities in two synthetic brain MRI images [22, 25, 23]. The images are shown in Figures 6.7(a)-6.7(b). In this case, the regions X and Y are identical



(a) (b)

Figure 6.6: (a) The image of a Cartesian mesh under the gradient map ∇u (§6.3.3) and (b) a zoomed in view of the same mesh in the region of large density.

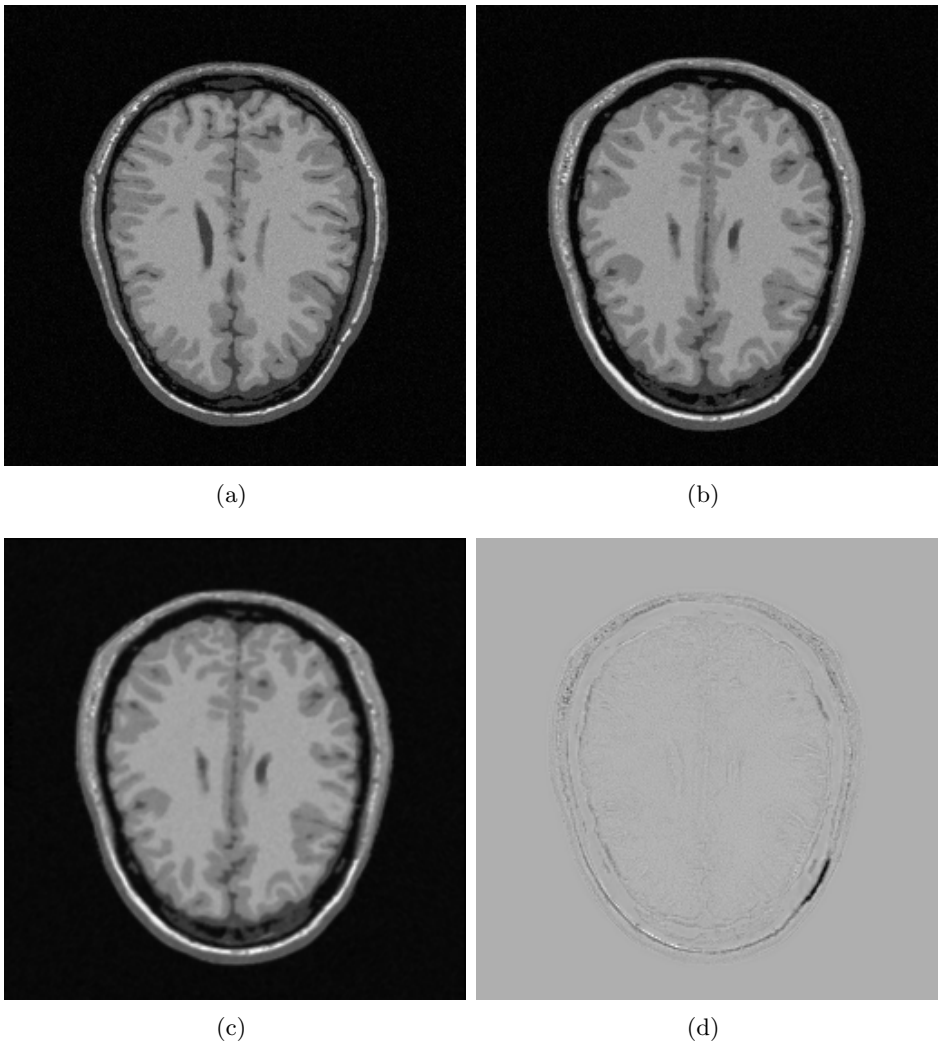
and are equal to the unit square. The fully resolved images contain 256×256 pixels. For the computations presented here, we have also interpolated both images onto coarser grids so that in each case we are mapping an $N \times N$ grid onto the density function obtained from an $N \times N$ image.

In this example, the density functions have large gradients, which effectively increase as we map onto more refined images. The solver now runs in about $\mathcal{O}(M^{1.1})$ time; see Table 6.4.

Figures 6.7(c)-6.7(d) show the image we obtain by solving the Monge-Ampère equation and interpolating and the error in this image. The mapped image we obtain agrees well with the given image. Not surprisingly, the largest error occurs around the edges of the brain where the density function is essentially discontinuous; consequently, small errors in the map can lead to large errors in estimated pixel intensity.

6.4 Computational Results: Optimal Transport

Next, we turn our attention to computational results for the mass transport problem. In each example, we embed our domain in the square $[-0.5, 0.5] \times [-0.5, 0.5]$ (setting the density $f = 0$ outside our domain X). While this can lead to singularities in the solutions, our methods are robust enough to handle this non-smoothness.



(a) (b)

(c) (d)

Figure 6.7: (a) The initial density function f, (b) the final density function g, (c) the image obtained by solving the Monge-Ampère equation and interpolating, and (d) the error in the resulting image.

N	Newton Iterations	CPU Time (s)
32	7	1.1
46	7	1.2
64	9	3.0
90	10	7.0
128	12	13.7
182	12	34.9
256	13	81.6

Table 6.4: Computation time for a map between two brain MRI images (§6.3.4).

In each case, we present the total number of Monge-Ampère solves required on the $N \times N$ grid (this does not include solves performed on coarser grids during the initialization process), as well as the total computation time required. When an exact solution is available for comparison, we provide the maximum error in the map:

$$\text{Error} = \max\{\|u_{x_1}^{exact} - u_{x_1}\|_\infty, \|u_{x_2}^{exact} - u_{x_2}\|_\infty\}.$$

The examples considered in this section include:

- A map between two ellipses, for which an exact solution is available for comparison.
- A map from two disconnected semi-circles onto a circle, for which an exact solution is available for comparison.
- A map from a square onto a convex polygon, which is neither smooth nor strictly convex, together with recovery of the inverse map.
- A map from a square onto a non-convex region.

6.4.1 Mapping an Ellipse to an Ellipse

First we consider the problem of mapping an ellipse onto an ellipse. To describe the ellipses, we let M_x, M_y be symmetric positive definite matrices and let B_1 be the unit ball in $\mathbb{R}^d$. Now we take $X = M_x B_1$, $Y = M_y B_2$ to be ellipses with constant densities f, g in each ellipse.

In R^2, the optimal map can be obtained explicitly [68] from

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

where R is the rotation matrix

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

the angle θ is given by

$$\tan(\theta) = \text{trace}(M_x^{-1} M_y^{-1} J)/\text{trace}(M_x^{-1} M_y^{-1}),$$

and the matrix J is equal to

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

We use the particular example

$$M_x = \begin{pmatrix} 0.4 & 0 \\ 0 & 0.2 \end{pmatrix}, \quad M_y = \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.4 \end{pmatrix},$$

which is pictured in Figure 6.8.

Projections onto the ellipse at each step are accomplished efficiently using the method described in [58].

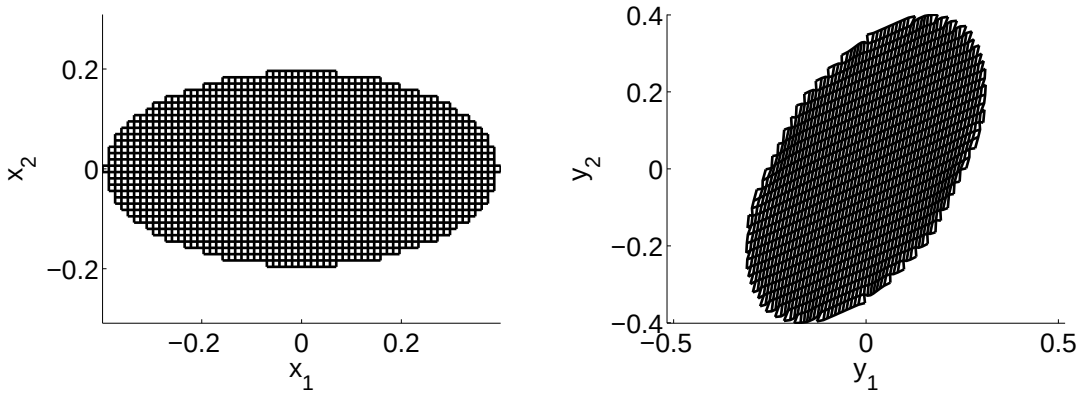
Computational results are presented in Table 6.5 and Figure 6.8. The error is decreasing uniformly (about $\mathcal{O}(h^{0.8})$). We cannot expect high accuracy for this example due to the degeneracy of this example: the density f vanishes in part of the domain. This means that the lower accuracy monotone stencil is needed in this region, which will in turn affect the error in the map.

Despite the degeneracy of this example and the multiple Monge-Ampère solves required to initialize and solve this problem, the computation requires only $\mathcal{O}(M^{1.1})$ time.

6.4.2 Mapping from a Disconnected Region

We now return to the degenerate example considered in §2.2.1. This is the problem of mapping the two half-circles

$$X = \{(x_1, x_2) \mid x_1 \leq -0.1, (x_1 + 0.1)^2 + x_2^2 \leq 0.3^2\} \cup \{(x_1, x_2) \mid x_1 \geq 0.1, (x_1 - 0.1)^2 + x_2^2 \leq 0.3^2\}$$



(a) (b)

Figure 6.8: (a) A Cartesian mesh in the ellipse X and (b) its image under the gradient map ∇u (§6.4.1).

N	h	(1.2) Solves	CPU Time (s)	Maximum Error
32	0.0323	4	0.7	0.0264
46	0.0222	13	1.7	0.0180
64	0.0159	3	1.8	0.0152
90	0.0112	6	5.5	0.0117
128	0.0079	3	9.9	0.0083
182	0.0055	3	25.3	0.0060
256	0.0039	2	61.9	0.0048

Table 6.5: Computation time and maximum error for the map between two ellipses (§6.4.1).

onto the circle

$$Y = \{(y_1, y_2) \mid y_1^2 + y_2^2 \leq 0.3^2\}.$$

Results are presented in Table 6.6 and Figure 6.9. In this case, the error appears to approach a constant value of around 0.004. This is not surprising since in this case, the monotone stencil is needed in the region where f vanishes or is discontinuous. The width of the stencil then limits the accuracy of solutions, as we explained in Chapter 4. The computation time for this very degenerate example is about $\mathcal{O}(M^{1.3})$.

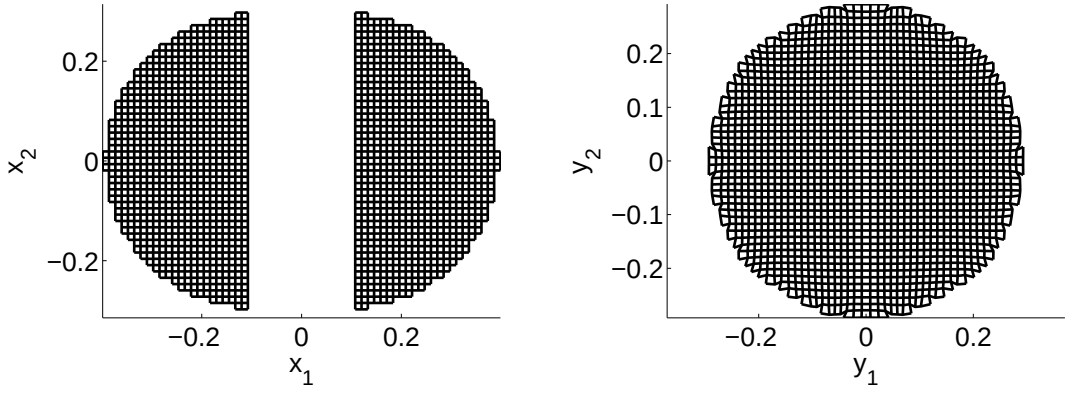


Figure 6.9: (a) A Cartesian mesh in two half-circles X and (b) its image under the gradient map ∇u (§6.4.2).

N	h	(1.2) Solves	CPU Time (s)	Maximum Error
32	0.0323	5	0.5	0.0171
46	0.0222	2	0.5	0.0160
64	0.0159	5	1.6	0.0129
90	0.0112	9	6.0	0.0082
128	0.0079	5	11.8	0.0052
182	0.0055	4	30.3	0.0040
256	0.0039	3	66.7	0.0038

Table 6.6: Computation time and maximum error for the map from two half-circles to a circle (§6.4.2).

6.4.3 Mapping to a Convex Polygon

Next we consider a map onto a convex polygon Y, which has a very non-smooth boundary. We use the polygon Y with vertices:

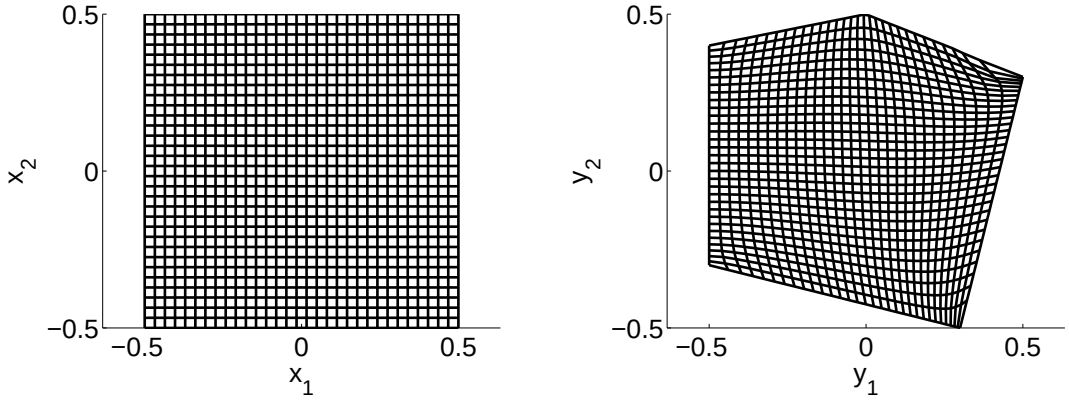
$$(-0.5, -0.3),\ (-0.5, 0.4),\ (0, 0.5),\ (0.5, 0.3),\ (0.3, -0.5).$$

Despite the non-smoothness of ∂Y, our method successfully maps the square $[-0.5, 0.5] \times [-0.5, 0.5]$ into the prescribed polygon, though we do not have an exact solution to compare with.

We also solve the problem by solving the inverse problem (mapping the polygon to the square) and inverting this map as in §6.3.2. While no exact solution is available for comparison, we can check the maximum difference between components of the two maps:

$$\max\{\|u_{x_1} - u^{inv}_{x_1}\|_\infty, \|u_{x_2} - u^{inv}_{x_2}\|_\infty\}.$$

Results are presented in Table 6.7 and Figure 6.10. The computation is reasonably efficient, requiring about $\mathcal{O}(M^{1.2})$ time for both the forward and inverse problem. We also observe that the agreement between the maps obtained from the forward and inverse approaches improves as we refine the grid.



(a) (b)

Figure 6.10: (a) A Cartesian mesh and (b) its image under the gradient map ∇u (§6.4.3).

	Forward Problem		Inverse Problem		
N	Iterations	Time (s)	Iterations	Time (s)	Max Difference
32	3	0.4	1	0.3	0.0397
46	3	0.8	1	0.7	0.0227
64	3	1.5	1	1.1	0.0153
90	4	3.2	1	2.3	0.0119
128	4	8.5	1	6.2	0.0087
182	4	21.0	1	13.5	0.0063
256	4	61.8	1	33.9	0.0050
362	4	154.3	1	92.6	0.0044

Table 6.7: Monge-Ampère solves, computation time and maximum difference for a map from square to polygon obtained by a direct solve and by inverting the inverse map (§6.4.3).

6.4.4 Mapping to a Non-convex Region

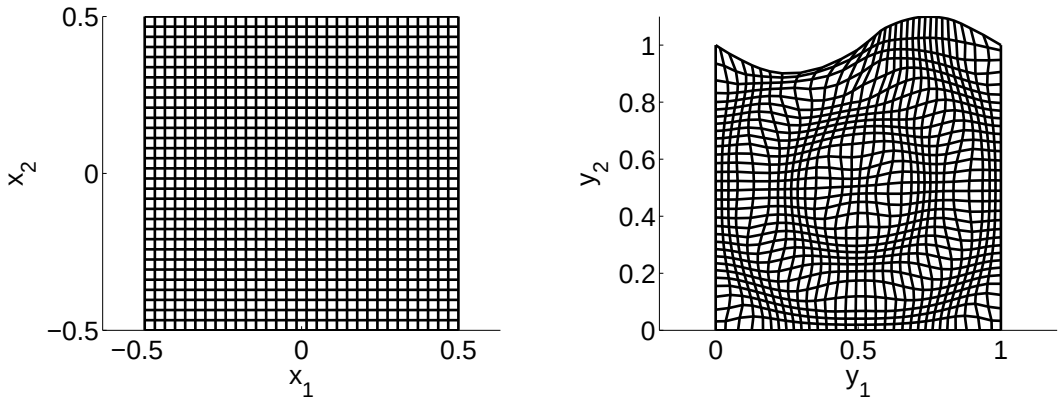
Next, we compute the mapping of the square with constant density f onto a non-convex region given by

$$Y = \left\{(y_1, y_2) \mid 0 \leq y_1 \leq 1,\, 0 \leq y_2 \leq 1 - 0.1\sin(2\pi y_1)\right\}.$$

We impose the following periodic density in the region Y:

$$g(y_1, y_2) = 2 + \cos\left(8\pi\sqrt{(y_1 - 0.5)^2 + (y_2 - 0.5)^2}\right).$$

The results are displayed in Table 6.8 and Figure 6.11. Despite the non-convexity of Y, the method successfully maps the region X into the non-convex region Y. The non-convexity does not appear to affect the computation time at all: the solution time is roughly $\mathcal{O}(M)$.



(a) (b)

Figure 6.11: (a) A Cartesian mesh and (b) its image under the gradient map ∇u (§6.4.4).

N	(1.2) Solves	CPU Time (s)
32	5	1.7
46	4	2.2
64	4	5.4
90	5	8.4
128	5	21.4
182	5	41.9
256	3	68.1
362	3	197.4

Table 6.8: Computation time for the map onto a non-convex region (§6.4.4).

Chapter 7

Conclusions

7.1 Summary

In this thesis, we have focused on the problem of numerically computing solutions to the elliptic Monge-Ampère equation. Because of the nonlinearity of the equation, a classical solution may not exist and standard techniques can fail. In addition, fast solution methods such as Newton's method can become unstable, making it necessary to use slower solvers.

The numerical solution of the Monge-Ampère equation requires a suitable approximation to the determinant of the Hessian of a convex function. Instead of using a standard expansion of the determinant, we have rewritten the equation in a variational form that includes the constraint that solutions must be convex. Using this form of the equation, together with the definition of a viscosity solution, we successfully produced a monotone finite difference discretisation that provably converges to the weak (viscosity) solution. Moreover, the special structure of the discretisation allows us to use Newton's method to efficiently solve the resulting system of nonlinear equations.

We have also looked at the problem of building a formally more accurate scheme for the Monge-Ampère equation that nevertheless handles singularities correctly. We accomplished this by constructing two hybrid discretisations that carefully combined the monotone scheme with a formally more accurate discretisation. For one of these schemes, we succeeded in proving convergence to the viscosity solution of the equation.

Finally, we looked at the related problem of optimal mass transport with quadratic cost. This problem can be expressed as a Monge-Ampère equation coupled to an implicit transport boundary condition. Previously, this boundary condition had been implemented

only in the simplest geometries. In this thesis we proposed a new method for enforcing this boundary condition by solving a sequence of Monge-Ampère equations with a simpler Neumann boundary condition. By combining this with our fast solver for the Monge-Ampère equation, we were able to successfully and efficiently compute solutions to the optimal transport problem in a number of challenging cases that included mapping onto unbounded densities, recovery of inverse maps, maps from disconnected domains, and maps into non-convex regions.

7.2 Future Work

The work of this thesis answers a few interesting questions, but it also raises many new questions and suggests several possible directions for future research.

One very important research problem is the construction of formally high-order numerical methods that converge to the correct weak solution of the underlying equation. In this thesis, we proved a very general result about the convergence of certain numerical methods for the class of second-order degenerate elliptic PDEs. Further study is needed to flesh out the full implications of this theorem. For instance, this result could lead to construction of or convergence proofs for high-order methods for Hamilton-Jacobi equations.

The problem of enforcing optimal transport boundary conditions certainly deserves additional study. The method we proposed in this thesis appeared to perform well for mappings into convex—and some non-convex—target sets. However, at this time we do not have a proof that this method converges. It would also be desirable to extend this method to even more general non-convex targets, where the projection operator may not be well-defined.

We are also interested in the issue of computing solutions to more general optimal transport problems. In this thesis, we limited our attention to a quadratic cost function. The particular structure of this special case allowed the problem to be re-expressed in terms of the elliptic Monge-Ampère equation. The situation becomes much more complicated when we consider other cost functions. However, the more general case can still be brought into the realm of partial differential equations [66]. It would be very interesting to try to extend the analytical and computational results of this thesis to the more challenging problem of optimal transport with a non-quadratic cost function.

The application of techniques developed in this thesis to various applications of optimal transport is another possible direction for future research. One interesting application is

adaptive mesh generation. This is useful in the study of other equations, whose solutions may change rapidly in a small region of the domain. In order to increase the accuracy and decrease the cost of computations, it is desirable to compute on a mesh that clusters more grid points in areas where the solution is changing rapidly. The Monge-Ampère equation can be used to generate these equidistributing meshes by mapping a constant density function (uniform mesh) onto a density function (or monitor function) that contains information about the solution of the underlying equation [13]. Our ability to map into different types of geometries also suggests a method of generating equidistributing meshes in different types of domains. In order to realise the benefits of these equidistributing meshes in solving other problems, it will be necessary to couple the Monge-Ampère equation to other problems of interest.

Another interesting application is the problem of image registration. For images such as brain MRIs, which provide information about proton density, it is natural to use techniques related to optimal transport to establish correspondences between different images [51]. Depending on the particular approach used, the optimal transport problem may be coupled to other equations or constraints. Again, techniques for doing optimal transport can be used as a starting point for this important application.

Bibliography

[1] R. Abgrall. Construction of simple, stable, and convergent high order schemes for steady first order Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 31(4):2419–2446, 2009.

[2] Luigi Ambrosio. Lecture notes on optimal transport problems. In *Mathematical aspects of evolving interfaces (Funchal, 2000)*, volume 1812 of *Lecture Notes in Math.*, pages 1–52. Springer, Berlin, 2003.

[3] I. Bakelman. *Convex analysis and nonlinear geometric elliptic equations*. Springer-Verlag, 1994.

[4] Guy Barles and Panagiotis E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Anal.*, 4(3):271–283, 1991.

[5] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numer. Math.*, 84(3):375–393, 2000.

[6] Jean-David Benamou, Brittany D. Froese, and Adam M. Oberman. Two numerical methods for the elliptic Monge-Ampère equation. *ESAIM: Math. Model. Numer. Anal.*, 44(4), 2010.

[7] Dimitri P. Bertsekas. *Convex analysis and optimization*. Athena Scientific, Belmont, MA, 2003. With Angelia Nedić and Asuman E. Ozdaglar.

[8] Klaus Böhmer. On finite element methods for fully nonlinear elliptic equations of second order. *SIAM J. Numer. Anal.*, 46(3):1212–1249, 2008.

[9] J. Frédéric Bonnans and Housnaa Zidani. Consistency of generalized finite difference schemes for the stochastic HJB equation. *SIAM J. Numer. Anal.*, 41(3):1008–1021 (electronic), 2003.

[10] Susanne C. Brenner, Thirupathi Gudi, Michael Neilan, and Li-Yen Sung. C^0 penalty methods for the fully nonlinear Monge-Ampère equation. *Math. Comp.*, 80(276):1979–1995, 2011.

[11] Susanne C. Brenner and Michael Neilan. Finite element approximations of the three-dimensional Monge-Ampère equation. 2011. Submitted.

[12] C. J. Budd and J. F. Williams. Moving mesh generation using the parabolic Monge-Ampère equation. *SIAM J. Sci. Comput.*, 31(5):3438–3465, 2009.

[13] Chris J. Budd, Weizhang Huang, and Robert D. Russell. Adaptivity with moving grids. *Acta Numer.*, 18:111–241, 2009.

[14] L. Caffarelli, L. Nirenberg, and J. Spruck. The Dirichlet problem for nonlinear second-order elliptic equations. I. Monge-Ampère equation. *Comm. Pure Appl. Math.*, 37(3):369–402, 1984.

[15] Luis A. Caffarelli. Interior $W^{2,p}$ estimates for solutions of the Monge-Ampère equation. *Ann. of Math. (2)*, 131(1):135–150, 1990.

[16] Luis A. Caffarelli. Some regularity properties of solutions of Monge Ampère equation. *Comm. Pure Appl. Math.*, 44(8-9):965–969, 1991.

[17] Luis A. Caffarelli. Boundary regularity of maps with convex potentials. *Comm. Pure Appl. Math.*, 45(9):1141–1151, 1992.

[18] Luis A. Caffarelli. The regularity of mappings with a convex potential. *J. Amer. Math. Soc.*, 5(1):99–104, 1992.

[19] Luis A. Caffarelli. Boundary regularity of maps with convex potentials. II. *Ann. of Math. (2)*, 144(3):453–496, 1996.

[20] Luis A. Caffarelli and Cristian E. Gutiérrez. Properties of the solutions of the linearized Monge-Ampère equation. *Amer. J. Math.*, 119(2):423–465, 1997.

[21] Luis A. Caffarelli and Mario Milman, editors. *Monge Ampère equation: applications to geometry and optimization*, volume 226 of *Contemporary Mathematics*, Providence, RI, 1999. American Mathematical Society.

[22] McConnell Brain Imaging Center. Brainweb: Simulated brain database, November 2010. http://www.bic.mni.mcgill.ca/brainweb.

[23] C. A. Cocosco, V. Kollokian, Kwan R. K.-S., and A. C. Evans. Brainweb: Online interface to a 3d mri simulated brain database. In *NeuroImage*, volume 5, 1997.

[24] Daniel Cohen-Or. Space deformations, surface deformations and the opportunities in-between. *J. Comput. Sci. Technol*, 24(1):2–5, 2009.

[25] D. L. Collins, A. P. Zijenbos, N. J. Kollokian, J.and Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans. Design and construction of a realistic digital brain phantom. *IEEE Transactions on Medical Imaging*, 17(3):463–468, 1998.

[26] Michael G. Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc. (N.S.)*, 27(1):1–67, 1992.

[27] M. J. P. Cullen and R. J. Douglas. Applications of the Monge-Ampère equation and Monge transport problem to meteorology and oceanography. In *Monge Ampère equation: applications to geometry and optimization (Deerfield Beach, FL, 1997)*, volume 226 of *Contemp. Math.*, pages 33–53. Amer. Math. Soc., Providence, RI, 1999.

[28] E. J. Dean and R. Glowinski. An augmented Lagrangian approach to the numerical solution of the Dirichlet problem for the elliptic Monge-Ampère equation in two dimensions. *Electron. Trans. Numer. Anal.*, 22:71–96 (electronic), 2006.

[29] E. J. Dean and R. Glowinski. Numerical methods for fully nonlinear elliptic equations of the Monge-Ampère type. *Comput. Methods Appl. Mech. Engrg.*, 195(13-16):1344–1386, 2006.

[30] Edward J. Dean and Roland Glowinski. On the numerical solution of the elliptic Monge-Ampère equation in dimension two: a least-squares approach. In *Partial differential equations*, volume 16 of *Comput. Methods Appl. Sci.*, pages 43–63. Springer, Dordrecht, 2008.

[31] Edward J. Dean, Roland Glowinski, and Tsorng-Whay Pan. Operator-splitting methods and applications to the direct numerical simulation of particulate flow and to the solution of the elliptic Monge-Ampère equation. In *Control and boundary analysis*, volume 240 of *Lect. Notes Pure Appl. Math.*, pages 1–27. Chapman & Hall/CRC, Boca Raton, FL, 2005.

[32] G. L. Delzanno, L. Chacón, J. M. Finn, Y. Chung, and G. Lapenta. An optimal robust equidistribution method for two-dimensional grid adaptation based on Monge-Kantorovich optimization. *J. Comput. Phys.*, 227(23):9841–9864, 2008.

[33] Lawrence C. Evans. Partial differential equations and Monge-Kantorovich mass transfer. In *Current developments in mathematics, 1997 (Cambridge, MA)*, pages 65–126. Int. Press, Boston, MA, 1999.

[34] Xiaobing Feng and Michael Neilan. Mixed finite element methods for the fully nonlinear Monge-Ampère equation based on the vanishing moment method. *SIAM J. Numer. Anal.*, 47(2):1226–1250, 2009.

[35] Xiaobing Feng and Michael Neilan. Vanishing moment method and moment solutions for fully nonlinear second order partial differential equations. *J. Sci. Comput.*, 38(1):74–98, 2009.

[36] J. M. Finn, G. L. Delzanno, and L. Chacón. Grid generation and adaptation by Monge-Kantorovich optimization in two and three dimensions. In *Proceedings of the 17th International Meshing Roundtable*, pages 551–568, 2008.

[37] Uriel Frisch, Sabino Matarrese, Roya Mohayaee, and Andrei Sobolevski. A reconstruction of the initial conditions of the universe by optimal mass transportation. *Nature*, 417, 2002.

[38] Brittany D. Froese. Numerical methods for two second order elliptic equations. Master's thesis, Simon Fraser University, 2009.

[39] Brittany D. Froese. A numerical method for the elliptic Monge-Ampère equation with transport boundary conditions. *SIAM J. Sci. Comput.*, 34(3):A1432–A1459, 2012.

[40] Brittany D. Froese and Adam 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.

[41] Brittany D. Froese and Adam M. Oberman. Fast finite difference solvers for singular solutions of the elliptic Monge-Ampère equation. *J. Comput. Phys.*, 230(3):818–834, 2011.

[42] Brittany D. Froese and Adam M. Oberman. Accurate convergent finite difference approximations for viscosity solutions of the elliptic Monge-Ampère partial differential equation. 2012. Submitted.

[43] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order*, volume 224 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, second edition, 1983.

[44] T. Glimm and V. Oliker. Optical design of single reflector systems and the Monge-Kantorovich mass transfer problem. *J. Math. Sci. (N. Y.)*, 117(3):4096–4108, 2003. Nonlinear problems and function theory.

[45] Tilmann Glimm and Vladimir Oliker. Optical design of two-reflector systems, the Monge-Kantorovich mass transfer problem and Fermat's principle. *Indiana Univ. Math. J.*, 53(5):1255–1277, 2004.

[46] R. Glowinski, E. J. Dean, G. Guidoboni, L. H. Juárez, and T.-W. Pan. Applications of operator-splitting methods to the direct numerical simulation of particulate and free-surface flows and to the numerical solution of the two-dimensional elliptic Monge-Ampère equation. *Japan J. Indust. Appl. Math.*, 25(1):1–63, 2008.

[47] Roland Glowinski. Numerical methods for fully nonlinear elliptic equations. In Rolf Jeltsch and Gerhard Wanner, editors, *6th International Congress on Industrial and Applied Mathermatics, ICIAM 07, Invited Lectures*, pages 155–192, 2009.

[48] Cristian E. Gutiérrez. *The Monge-Ampère equation*. Progress in Nonlinear Differential Equations and their Applications, 44. Birkhäuser Boston Inc., Boston, MA, 2001.

[49] Eldad Haber, Tauseef Rehman, and Allen Tannenbaum. An efficient numerical method for the solution of the L_2 optimal mass transfer problem. *SIAM J. Sci. Comput.*, 32(1):197–211, 2010.

[50] Steven Haker, Allen Tannenbaum, and Ron Kikinis. Mass preserving mappings and image registration. In *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 120–127, London, UK, 2001. Springer-Verlag.

[51] Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. Optimal mass transport for registration and warping. *Int. J. Comput. Vision*, 60(3):225–240, 2004.

[52] George J. Haltiner. *Numerical weather prediction*. Wiley, New York, 1971.

[53] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph*, 26(3):71, 2007.

[54] L. V. Kantorovich. On the transfer of masses. *Dokl. Akad. Nauk. SSSR*, 37(7–8):227–229, 1942.

[55] L. V. Kantorovich. On a problem of Monge. *Uspekhi Mat. Nauk.*, 3(2):225–226, 1948.

[56] Akira Kasahara. Significance of non-elliptic regions in balanced flows of the tropical atmosphere. *Monthly Weather Review*, 110(12), 1982.

[57] C. T. Kelley. *Iterative methods for linear and nonlinear equations*, volume 16 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. With separately available software.

[58] Yu. N. Kiselev. Algorithms for the projection of a point onto an ellipsoid. *Liet. Mat. Rink.*, 34(2):174–196, 1994.

[59] M. Knott and C. S. Smith. On the optimal mapping of distributions. *J. Optim. Theory Appl.*, 43(1):39–49, 1984.

[60] Shigeaki Koike. *A beginner's guide to the theory of viscosity solutions*, volume 13 of *MSJ Memoirs*. Mathematical Society of Japan, Tokyo, 2004.

[61] P.-L. Lions and P. E. Souganidis. Convergence of MUSCL and filtered schemes for scalar conservation laws and Hamilton-Jacobi equations. *Numer. Math.*, 69(4):441–470, 1995.

[62] P.-L. Lions, N. S. Trudinger, and J. I. E. Urbas. The Neumann problem for equations of Monge-Ampère type. *Comm. Pure Appl. Math.*, 39(4):539–563, 1986.

[63] Yaron Lipman, Johannes Kopf, Daniel Cohen-Or, and David Levin. GPU-assisted positive mean value coordinates for mesh deformations. In Alexander G. Belyaev and Michael Garland, editors, *Symposium on Geometry Processing*, volume 257 of *ACM International Conference Proceeding Series*, pages 117–123. Eurographics Association, 2007.

[64] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. *ACM Trans. Graph*, 27(3), 2008.

[65] Grégoire Loeper and Francesca Rapetti. Numerical solution of the Monge-Ampére equation by a Newton's algorithm. *C. R. Math. Acad. Sci. Paris*, 340(4):319–324, 2005.

[66] Xi-Nan Ma, Neil S. Trudinger, and Xu-Jia Wang. Regularity of potential functions of the optimal transportation problem. *Arch. Ration. Mech. Anal.*, 177(2):151–183, 2005.

[67] Robert J. McCann. Existence and uniqueness of monotone measure-preserving maps. *Duke Math. J.*, 80(2):309–323, 1995.

[68] Robert J. McCann and Adam M. Oberman. Exact semi-geostrophic flows in an elliptical ocean basin. *Nonlinearity*, 17(5):1891–1922, 2004.

[69] T. S. Motzkin and W. Wasow. On the approximation of linear elliptic differential equations by difference equations with positive coefficients. *J. Math. Physics*, 31:253–259, 1953.

[70] Adam M. Oberman. A convergent monotone difference scheme for motion of level sets by mean curvature. *Numer. Math.*, 99(2):365–379, 2004.

[71] Adam M. Oberman. A convergent difference scheme for the infinity Laplacian: construction of absolutely minimizing Lipschitz extensions. *Math. Comp.*, 74(251):1217–1230 (electronic), 2005.

[72] Adam 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 (electronic), 2006.

[73] Adam M. Oberman. Computing the convex envelope using a nonlinear partial differential equation. *Math. Models Methods Appl. Sci.*, 18(5):759–780, 2008.

[74] Adam M. Oberman. Wide stencil finite difference schemes for the elliptic Monge-Ampère equation and functions of the eigenvalues of the Hessian. *Discrete Contin. Dyn. Syst. Ser. B*, 10(1):221–238, 2008.

[75] Adam M. Oberman and Luis Silvestre. The Dirichlet problem for the convex envelope. *Trans. Amer. Math. Soc.*, to appear.

[76] V. I. Oliker and L. D. Prussner. On the numerical solution of the equation $(\partial^2 z/\partial x^2)(\partial^2 z/\partial y^2) - (\partial^2 z/\partial x \partial y)^2 = f$ and its discretizations, I. *Numer. Math.*, 54(3):271–293, 1988.

[77] A. V. Pogorelov. The Dirichlet problem for the multidimensional analogue of the Monge-Ampère equation. *Dokl. Akad. Nauk SSSR*, 201:790–793, 1971.

[78] R. T. Rockafellar. Characterization of the subdifferentials of convex functions. *Pacific J. Math.*, 17:497–510, 1966.

[79] Filippo Santambrogio. Models and applications of optimal transport in economics, traffic and urban planning. 2010. http://arxiv.org/PS_cache/arxiv/pdf/1009/1009.3857v1.pdf.

[80] J. J. Stoker. *Nonlinear elasticity*. Gordon and Breach Science Publishers, New York, 1968.

[81] Gilbert Strang. *Linear algebra and its applications*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, second edition, 1980.

[82] Mohamed Sulman, J. F. Williams, and R. D. Russell. Optimal mass transport for higher dimensional adaptive grid generation. *J. Comput. Phys.*, 230(9):3302–3330, 2011.

[83] Mohamed M. Sulman, J. F. Williams, and Robert D. Russell. An efficient approach for the numerical solution of the Monge-Ampère equation. *Appl. Numer. Math.*, 61(3):298–307, 2011.

[84] Neil S. Trudinger and Xu-Jia Wang. On the second boundary value problem for Monge-Ampère type equations and optimal transportation. *Ann. Sc. Norm. Super. Pisa Cl. Sci. (5)*, 8(1):143–174, 2009.

[85] T. ur Rehman, E. Haber, G. Pryor, J. Melonakos, and A. Tannenbaum. 3D nonrigid registration via optimal mass transport on the GPU. *Med Image Anal*, 13(6):931–40, 12 2009.

[86] John Urbas. On the second boundary value problem for equations of Monge-Ampère type. *J. Reine Angew. Math.*, 487:115–124, 1997.

[87] John I. E. Urbas. The generalized Dirichlet problem for equations of Monge-Ampère type. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 3(3):209–228, 1986.

[88] Cédric Villani. *Topics in optimal transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2003.

[89] Xu-Jia Wang. On the design of a reflector antenna. *Inverse Problems*, 12(3):351–375, 1996.

[90] Xu-Jia Wang. On the design of a reflector antenna. II. *Calc. Var. Partial Differential Equations*, 20(3):329–341, 2004.

[91] B. S. Westcott. *Shaped reflector antenna design*. Research Studies Press, New York, 1983.

[92] V. Zheligovsky, O. Podvigina, and U. Frisch. The Monge-Ampère equation: Various forms and numerical solution. *J. Comput. Phys.*, 229(13):5043–5061, 2010.