

Fast finite difference solvers for singular solutions of the elliptic Monge–Ampère equation

B.D. Froese, A.M. Oberman *

Department of Mathematics, Simon Fraser University Burnaby, British Columbia, Canada V5A 1S6

ARTICLE INFO

Article history:

Received 30 June 2010

Received in revised form 15 October 2010

Accepted 15 October 2010

Available online 23 October 2010

Keywords:

Fully nonlinear elliptic Partial Differential Equations

Monge–Ampère equations

Nonlinear finite difference methods

Viscosity solutions

Monotone schemes

Convexity constraints

ABSTRACT

The elliptic Monge–Ampère equation is a fully nonlinear Partial Differential Equation which originated in geometric surface theory, and has been applied in dynamic meteorology, elasticity, geometric optics, image processing and image registration. Solutions can be singular, in which case standard numerical approaches fail.

In this article we build a finite difference solver for the Monge–Ampère equation, which converges even for singular solutions. Regularity results are used to select *a priori* between a stable, provably convergent monotone discretization and an accurate finite difference discretization in different regions of the computational domain. This allows singular solutions to be computed using a stable method, and regular solutions to be computed more accurately. The resulting nonlinear equations are then solved by Newton's method.

Computational results in two and three-dimensions validate the claims of accuracy and solution speed. A computational example is presented which demonstrates the necessity of the use of the monotone scheme near singularities.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

In this article we build a finite difference solver for the Monge–Ampère equation, which converges even for singular solutions. Regularity results are used to select *a priori* between two discretizations in different regions of the computational domain. Near possible singularities, a stable, provably convergent monotone discretization is used. Elsewhere a more accurate discretization is used. This allows singular solutions to be computed using a stable method, and regular solutions to be computed more accurately. The resulting nonlinear equations are then solved by Newton's method, which is fast, $\mathcal{O}(M^{1.3})$, where M is the number of data points, independent of the regularity of the solution.

1.1. The setting for equation

The Monge–Ampère equation is a fully nonlinear Partial Differential Equation (PDE).

$$\det(D^2u(x)) = f(x), \quad \text{for } x \text{ in } \Omega. \quad (\text{MA})$$

The Monge–Ampère operator, $\det(D^2u)$, is the determinant of the Hessian of the function u . The equation is augmented by the convexity constraint

* Corresponding author. Tel./fax: +1 604 268 6990.

E-mail addresses: bdf1@sfu.ca (B.D. Froese), aoberman@sfu.ca, aoberman@math.sfu.ca (A.M. Oberman).

URLs: <http://www.divbyzero.ca/froese/> (B.D. Froese), <http://math.sfu.ca/~aoberman> (A.M. Oberman).

u is convex, (C)

which is necessary for the equation to be elliptic. The convexity constraint is made explicit for emphasis: it is necessary for uniqueness of solutions and it is essential for numerical stability.

While other boundary conditions appear naturally in applications, we consider the simplest boundary conditions: the Dirichlet problem in a convex bounded subset $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$,

$$u(x) = g(x), \quad \text{for } x \text{ on } \partial\Omega. \quad (\text{D})$$

Under suitable assumptions on the domain and the functions $f(x), g(x)$, recalled in Section 2.1, there exist unique classical (C^2) solutions to (MA), (C). However, when these conditions fail, solutions can be singular. For singular solutions, the correct notion of weak solutions must be used. In this case, novel discretizations and solution methods must be used to approximate the solution.

1.2. Applications

The PDE (MA) is a geometric equation, which goes back to Monge and Ampère (see [1]). The equation naturally arises in geometric problems of existence and uniqueness of surfaces with prescribed metrics or curvatures [2,3]. Early applications identified in [4] include dynamic meteorology, elasticity, and geometric optics [5–8]. For an application of Monge–Ampère equations in mathematical finance, see [9].

The Monge–Ampère equation arises as the optimality conditions for the problem of optimal mass transport with quadratic cost [1,10,11]. This application of the Monge–Ampère equation has been used in many areas: image registration [12–14], mesh generation [15–17], reflector design [18], and astrophysics (estimating the shape of the early universe) [19].

The problem here is to find a mapping $g(x)$ that moves the measure $\mu_1(x)$ to $\mu_2(y)$ and minimizes the transportation cost functional

$$\int_{\mathbb{R}^d} |x - g(x)|^2 d\mu_1.$$

The optimal mapping is given by $g = \nabla u$, where u satisfies the Monge–Ampère equation

$$\det(D^2 u(x)) = \mu_1(x) / \mu_2(\nabla u(x)).$$

In this situation, the Dirichlet boundary condition (D) is replaced by the implicit boundary condition:

$$g(\cdot) : \Omega_1 \rightarrow \Omega_2, \quad (1)$$

where the sets Ω_1 and Ω_2 are the support of the measures μ_1, μ_2 . These boundary conditions are difficult to implement numerically; we are not aware of an implementation using PDE methods. For many applications, both domains are squares, and a simplifying assumption that edges are mapped to edges allows Neumann boundary conditions to be used. In other applications, periodic boundary conditions are used.

In other problems, the Monge–Ampère operator appears in an *inequality constraint* in a variational problem for optimal mappings where the cost is no longer the transportation cost. Here the operator has the effect of restricting the local area change on the set of admissible mappings, see [20,21].

1.3. Related numerical works

Despite the number of applications, until recently there have been few numerical publications devoted to solving the Monge–Ampère equation. We make a distinction between numerical approaches with optimal transportation type boundary conditions (1) and the standard Dirichlet boundary conditions (D). In the latter case, a number of numerical methods have been recently proposed for the solution of the Monge–Ampère equation.

An early work is [4], which presents a discretization which converges to the Aleksandrov solution in two-dimensions. Another early work by Benamou and Brenier [22] used a fluid mechanical approach to compute the solution to the optimal transportation problem.

For the problem with Dirichlet boundary conditions which is treated here, a series of papers have recently appeared by two groups of authors, Dean and Glowinski [23–25], and Feng and Neilan, [26,27]. The methods introduced by these authors perform best in the regular case and can break down in the singular case. The case of periodic boundary conditions was treated in [28]. See [29] for a more complete discussion.

Work which has appeared since then include the following. The work of [30] studies the case of periodic boundary conditions in odd-dimensional space. Several different formulations of the equation appear, including a Fourier integral form. The work of [15] solved the optimal transportation problem for the purpose of applications to mappings. A finite difference discretization was used with the boundary conditions assumed to be known on a rectangular domain.

1.4. Numerical challenges

When the conditions for regularity are satisfied, classical solutions can be approximated successfully using a range of standard techniques (see, for example works such as [23–27]). However, for singular solutions, standard numerical methods break down: either by becoming unstable, poorly conditioned, or by selecting the wrong (non-convex) solution.

Weak solutions

For singular solutions, the appropriate notion of weak (viscosity or Aleksandrov) solutions must be used. Numerical methods have been developed which capture weak solutions: Olier and Prussner, in [4], presented a method which converges to the Aleksandrov solution. One of the authors introduced a finite difference method which converges to the viscosity solution in [31]. Both of these methods were restricted to two-dimensions. In [32] we introduced a monotone discretization which is valid in arbitrary-dimensions. A proof of convergence to viscosity solutions is provided, as well as a proof of convergence of Newton's method.

Convexity

The convexity constraint is necessary for both uniqueness and stability. In particular, the Eq. (MA) fails to be elliptic if u is non-convex (see Section 2.5), so instabilities can arise if the convexity condition (C) is violated, as demonstrated in Section 8.1. Any approximation of (MA) requires some selection principle to choose the convex solution. This selection principle can be built into the discretization, as in [31], or built into the solution method, as in [29].

Accuracy

The convergent monotone schemes of [31,32] use a wide stencil, and the accuracy of the scheme depends on the *directional resolution*, which depends on the width of the stencil. As we demonstrate below, for highly singular solutions, such as (17), the directional resolution error can dominate. On the other hand, more accurate discretizations, such as standard finite differences, can be unstable for singular solutions.

Fast solvers

Previous work by the authors and a coauthor [29] investigated fast solvers for (MA). An explicit method was presented which was moderately fast, independent of the solution time. For regular solutions, a faster (by an order of magnitude) semi-implicit solution method was introduced (see Section 6.2) but this method was slower (by an order of magnitude) on singular solutions.

2. Analysis and weak solutions

In this section we present relevant regularity results and background analysis. In particular, the regularity results of Section 2.1 are used to determine the discretization used in Section 5.

The definition of viscosity solutions and Aleksandrov solutions presented in Sections 2.2, 2.3 are used to make sense of the weak solutions (15) and (17), respectively.

2.1. Regularity

Under the following conditions, the Monge–Ampère equation is guaranteed to have a unique $C^{2,\alpha}$ solution. Regularity results for the Monge–Ampère equation have been established in [33–35]. We refer to the book [36] for the following result. See also [37].

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

Remark 1. In the extreme case, with $f(x) = 0$ for all $x \in \Omega$, the Eq. (MA), (C) reduces to the computation of the convex envelope of the boundary conditions [38,39]. In this case, solutions may not even be continuous up to the boundary and can also be non-differentiable in the interior.

Remark 2. The fact that regularity can break down on convex polygons [11,40] is important, since often computations are performed on a rectangle.

2.2. Viscosity solutions

We recall the definition of viscosity solutions [41], which are defined for the Monge–Ampère equation in [36].

Definition 1. Let $u \in C(\Omega)$ be convex and $f \geq 0$ be continuous. The function u is a *viscosity subsolution (supersolution)* of the Monge–Ampère equation in Ω if whenever convex $\phi \in C^2(\Omega)$ and $x_0 \in \Omega$ 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 a supersolution.

In the following, we will use the notation

$$x^+ = \max(x, 0), \quad \text{for } x \in \mathbb{R}.$$

Example 1 (*Viscosity solution of Monge–Ampère*). We consider an example which will later be solved numerically in two and three-dimensions (Sections 8, 9). Consider (MA) with solution and f given by:

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

This function, although it is not a classical C^2 solution of the Monge–Ampère equation, is a viscosity solution.

2.3. Aleksandrov solutions

Next we turn our attention to the Aleksandrov solution, which is a more general weak solution than the viscosity solutions. Here f is generally a measure. In this section we give a few essential definitions for the convenience of the reader. More details can be found in the reference [36].

We begin by recalling the definition of the normal mapping or subdifferential of a function.

Definition 2. The *normal mapping (subdifferential)* of a function u is the set-valued function ∂u defined by:

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

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

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

Definition 3. Given a function $u \in C(\Omega)$, the *Monge–Ampère measure* associated with u is defined as:

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

for any set $V \subset \Omega$, where $|E|$ is the Lebesgue measure of the set E .

This measure naturally leads to the notion of the generalized or Aleksandrov solution of the Monge–Ampère equation.

Definition 4. Let μ be a Borel measure defined in a convex set $\Omega \in \mathbb{R}^d$. Then the convex function u is an *Aleksandrov solution* of the Monge–Ampère equation

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

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

Example 2 (*Aleksandrov solution*). As an example, we consider the cone and the scaled Dirac measure

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

2.4. A PDE for convexity

The convexity constraint (C) is necessary for uniqueness, since without it, $-u$ is also a solution of (MA).

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

$$\lambda_1[D^2u] \geq 0,$$

understood in the viscosity sense [38,39], 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:

$$\det^+(M) = \prod_{j=1}^d \lambda_j^+ \tag{3}$$

where M is a symmetric matrix, with eigenvalues, $\lambda_1 \leq \dots \leq \lambda_n$. Using this notation, (MA), (C) becomes:

$$\det^+(D^2u(x)) = f(x), \quad \text{for } x \text{ in } \Omega. \quad (\text{MA}^+)$$

Remark 3. Notice that there is a trade off in defining (3): the constraint (C) is eliminated but the operator becomes non-differentiable near singular matrices.

2.5. Linearization and ellipticity

The linearization of the determinant is given by:

$$\nabla \det(M) \cdot N = \text{trace}(M_{\text{adj}}N).$$

Where M_{adj} is the adjugate [42], 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

$$M_{\text{adj}} = \det(M)M^{-1}. \quad (4)$$

We now apply these considerations to the linearization of the Monge–Ampère operator. When $u \in C^2$ we can linearize this operator as:

$$\nabla \det(D^2u) \cdot v = \text{trace}\left((D^2u)_{\text{adj}}D^2v\right). \quad (5)$$

Example 3. In two-dimensions we obtain

$$\nabla \det(D^2u)v = u_{xx}v_{yy} + u_{yy}v_{xx} - 2u_{xy}v_{xy},$$

which is homogenous of order one in D^2u . In dimension $d \geq 2$, the linearization is homogeneous order $d - 1$ in D^2u .

The linear operator

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

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

Lemma 1. Let $u \in C^2$. The linearization of the Monge–Ampère operator, (5) is elliptic if D^2u is positive definite or, equivalently, if u is (strictly) convex.

Remark 4. When the function u fails to be strictly convex, the linearization can be degenerate elliptic, which affects the conditioning of the linear system (5). When the function u is non-convex, the linear system can be unstable.

The definition of a nonlinear elliptic PDE operator generalizes the definition of linear elliptic operator. It also allows for the operators to be non-differentiable. The operator $F(M)$ is defined on symmetric matrices. In the equation, M will be replaced by the Hessian, D^2u .

Definition 5. Let the function $F(M)$ be a continuous function defined on symmetric matrices. Then $F(M)$ is *elliptic* if it satisfies the monotonicity condition

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

where for symmetric matrices $M \leq N$ means $x^TMx \leq x^TNx$ for all x .

Example 4. The function $\det^+(M)$ is a non-decreasing function of the eigenvalues, so it is elliptic. The corresponding PDE is the Monge–Ampère operator $\det^+(D^2u)$.

3. The standard finite difference discretization

We begin by considering the standard finite difference discretization of the Monge–Ampère equation. For brevity, we describe the discretization in two-dimensions, but this is easily generalized to higher-dimensions.

This discretization does not enforce the convexity condition (C), so it can lead to instabilities. In particular, we show in Section 8.1 that Newton's method can become unstable if this discretization is used.

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

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

In two-dimensions, the natural discretization of the operator is given by:

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

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

$$[\mathcal{D}_{xx}u]_{ij} = \frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} - 2u_{ij})$$

$$[\mathcal{D}_{yy}u]_{ij} = \frac{1}{h^2}(u_{i,j+1} + u_{i,j-1} - 2u_{ij})$$

$$[\mathcal{D}_{xy}u]_{ij} = \frac{1}{4h^2}(u_{i+1,j+1} + u_{i-1,j-1} - u_{i-1,j+1} - u_{i+1,j-1}).$$

Remark 5. There is no reason to assume that the standard discretization converges. In fact, the two-dimensional scheme has multiple solutions. In [29] this discretization was used, but the solvers were designed to select the convex solution.

4. Convergent monotone discretization

The method of [31] describes a discretization of the two-dimensional Monge–Ampère equation that converges to the viscosity solution. In [32] we introduced another discretization, which generalized to higher-dimensions, and also converged to the viscosity solution. Both methods require the use of a wide stencil scheme, which has an additional discretization parameter, the *directional resolution*, explained below.

In addition to being monotonic, which implies it is provably convergent, the latter method discretizes the convexified version of the equation, (MA^+) , which is enough to ensure convergence of Newton’s method. The proof of this result can be found in [32].

In this section we present the convergent discretization, which will be used to build the hybrid solver.

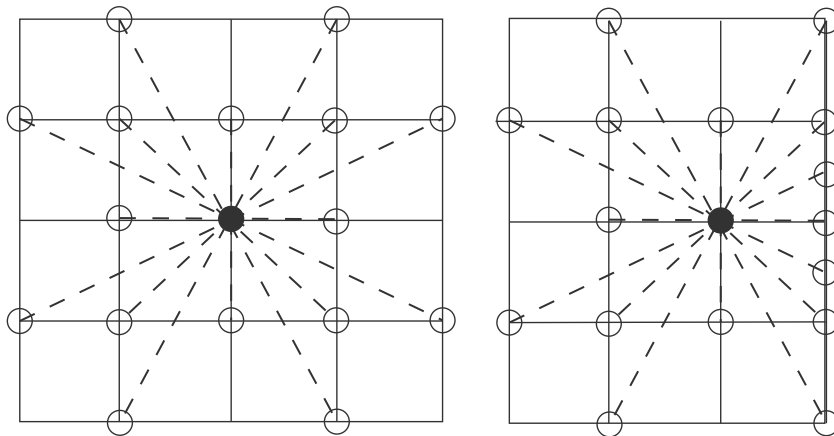
4.1. Wide stencils

When we discretize the operator on a finite difference grid, we approximate the second derivatives by centered finite differences (spatial discretization). In addition to the approximation of second directional derivatives by finite differences, we need to discretize the directions of these derivatives, restricting to directions v that lie on the grid. We call this the *directional discretization*.

We consider the finite difference operator for the second directional derivative in the direction v , which lies on the finite difference grid. These directional derivatives are discretized by simply using finite differences on the grid

$$\mathcal{D}_{vv}u_i = \frac{1}{|v|h^2}(u(x_i + vh) + u(x_i - vh) - 2u(x_i)).$$

Depending on the direction of the vector v , this may involve a wide stencil. At points near the boundary of the domain, some values required by the wide stencil will not be available; see Fig. 1. In these cases, we use interpolation at the boundary to construct a (lower accuracy) stencil for the second directional derivative; see [31] for more details.



(a) In the interior.

(b) Near the boundary.

Fig. 1. Wide stencils on a two-dimensional grid.

Since the discretization considers only a finite number of directions v , 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.

4.2. Discretization of the convexified Monge–Ampère operator

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

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

This formula was used in [31] to build a monotone scheme for the Monge–Ampère operator, which is the product of the eigenvalues of the Hessian, by replacing the min, max over all directions, by a finite number of grid directions.

In higher-dimensions, the formula above does not generalize naturally. Instead, in [32], we used another characterization, which applied to positive definite matrices.

Lemma 2 (Variational characterization 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 of \mathbb{R}^d :*

$$V = \{(v_1, \dots, v_d) \mid v_j \in \mathbb{R}^d, v_i \perp v_j \text{ if } i \neq j, \|v_j\|_2 = 1\}.$$

Then the determinant of A is equivalent to

$$\prod_{j=1}^d \lambda_j = \min_{(v_1, \dots, v_d) \in V} \prod_{j=1}^d v_j^T A v_j.$$

We use Lemma 2 to characterize the determinant of the Hessian of a convex C^2 function ϕ in terms of second directional derivatives of ϕ .

$$\det(D^2 \phi) = \min_{(v_1, \dots, v_d) \in V} \prod_{j=1}^d v_j^T D^2 \phi v_j = \min_{(v_1, \dots, v_d) \in V} \prod_{j=1}^d \frac{\partial^2 \phi}{\partial v_j^2}.$$

The convexified Monge–Ampère operator (MA^+) can then be represented by simply enforcing positivity of the eigenvalues, which leads to the following,

$$\det^+(D^2 \phi) = \min_{\{v_1, \dots, v_d\} \in V} \prod_{j=1}^d \left(\frac{\partial^2 \phi}{\partial v_j^2} \right)^+.$$

To discretize the operator on a finite difference grid, restrict to the set of orthogonal vectors, \mathcal{G} , available on the given stencil. Then the convexified Monge–Ampère operator (MA^+) is approximated by

$$MA^M[u] \equiv \min_{\{v_1, \dots, v_d\} \in \mathcal{G}} \prod_{j=1}^d (D_{v_j v_j} u)^+. \quad (MA^M)$$

Theorem 3 (Convergence to Viscosity Solution). *Let the PDE (MA) have a unique viscosity solution. Then the solutions of the scheme (MA^M) , converge to the viscosity solution of (MA) as $h, d\theta, \delta \rightarrow 0$.*

The proof of convergence follows from verifying consistency and degenerate ellipticity and can be found in [32].

5. A hybrid discretization

In this section we propose a hybrid discretization of the Monge–Ampère equation which takes advantage of the best features of each of the previous discretizations. We want to make use of the natural discretization (MA^N) wherever possible in order to take advantage of its simplicity and higher accuracy. However, we wish to use the monotone discretization (MA^M) in regions where the solution may be singular in order to properly capture the behaviour of the viscosity solution. In this way we hope to achieve the second-order accuracy of the simple discretization in smooth regions and the monotonicity necessary to capture the behaviour of the viscosity solution in non-smooth regions.

We propose the following hybrid scheme.

Discretize (MA) using a weighted average of the two discretizations:

$$MA^H = w(x)MA^N + (1 - w(x))MA^M, \quad (MA^H)$$

where $w: \Omega \rightarrow [0, 1]$ is a weight function defined *a priori* from the data as follows.

We first identify Ω^ε which is a neighborhood of the possible singular set of u on Ω , using conditions (2).

$$\Omega^\varepsilon = \{x \in \Omega \mid \varepsilon < f(x) < 1/\varepsilon\} \cup \{x \in \partial\Omega \mid \partial\Omega \text{ flat or } g(x) \notin C^{2,\alpha}\}, \quad (6)$$

where ε is a small parameter, which we can take equal to h , the spatial resolution.

Then define $w(x)$ to be a differentiable function which is zero in an h -neighborhood of Ω^ε , and which goes to 1 elsewhere.

Remark 6. The hybrid scheme will sometimes be less accurate than the standard finite differences when the solution is C^2 , because it will lose some accuracy at the flat boundary. While this might seem conservative, there are examples, (see [29]), where the flat boundary causes blow up in the Hessian, so the monotone scheme is needed.

6. Explicit and semi-implicit solution methods

Any discretization of (MA) leads to a system of nonlinear equations which must be solved in order to obtain the approximate solution.

6.1. Explicit solution methods for monotone schemes

Using a monotone discretization $F[u]$ 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

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

Explicit iterative methods have the advantage that they are simple to implement, but the number of iterations required suffers from the well known CFL condition (which applies in a nonlinear form to monotone discretizations, as explained in [43]). This approach is slow because for stability it requires a small time step dt , which depends on the spatial resolution h . The time step, which can be computed explicitly, is $\mathcal{O}(h^2)$. This was the approach used in [31].

6.2. A semi-implicit solution method

The next method we discuss is a semi-implicit method, which involves solving the Laplace equation at each iteration. In [29] we used the identity (8) to build a semi-implicit solution method. We showed that the method is a contraction, but the strictness of the contraction requires strict positivity of f . In practice, this meant that the iteration was fast for regular solutions, but degenerated to become slower than the explicit method when f was zero in large parts of the domain.

The conditioning of the linearized Eq. (5), which affects solution time, depends on the strict convexity of the solution, see Lemma 1. The convexity, in turn depends on strict positivity of f , see Section 2.1. This explains why solution time of the semi-implicit solver depends on regularity.

Next, we describe a generalization of the semi-implicit method to higher-dimensions. We won't be using the method to solve (MA). Instead, we will use one iteration to set up the initial value for Newton's method.

Begin with the following identity for the Laplacian in two-dimensions,

$$|\Delta u| = \sqrt{(\Delta u)^2} = \sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xx}u_{yy}}. \quad (7)$$

So if u solves the Monge–Ampère equation, then

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

This leads to a semi-implicit scheme for solving the Monge–Ampère equation, used in [29].

$$\Delta u^{n+1} = \sqrt{2f + |D^2 u^n|^2}. \quad (8)$$

To generalize this 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.

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

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

The result is the semi-implicit scheme

$$\Delta u^{n+1} = \sqrt{d!f + P(\lambda_1[D^2 u^n], \dots, \lambda_d[D^2 u^n])}. \quad (9)$$

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

$$\Delta u^0 = \sqrt{d!f}. \quad (10)$$

7. Implementation of Newton's method

To solve the discretized equation

$$MA^H[u] = f,$$

we use a damped Newton iteration

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

for some $0 < \alpha < 1$. The damping parameter α is chosen at each step to ensure that the residual $\|MA^H(u^n) - f\|$ is decreasing. (In practice we can often take $\alpha = 1$, but damping is sometimes needed.)

The corrector v^n solves the linear system

$$(\nabla_u MA^H[u^n]) v^n = MA^H[u^n] - f. \quad (11)$$

To set up the Eq. (11), the Jacobian of the scheme is needed. Since the hybrid discretization is a weighted average of the monotone and standard discretization, and the weight function, $w(x)$, is determined *a priori*, the Jacobian of the hybrid scheme will simply be a weighted average of the corresponding Jacobians.

The Jacobian of the Monge–Ampère operator, discretized using standard finite differences, is given by

$$\nabla_u MA^N[u] = (\mathcal{D}_{xx}u)\mathcal{D}_{yy} + (\mathcal{D}_{yy}u)\mathcal{D}_{xx} - 2(\mathcal{D}_{xy}u)\mathcal{D}_{xy}, \quad (12)$$

which is a discrete version of the linearization of the Monge–Ampère Eq. (5).

The Jacobian for the monotone discretization is obtained by using Danskin's Theorem [44] and the product rule.

$$\nabla_u MA^M[u] = \sum_{j=1}^d \text{diag} \left(\prod_{k \neq j} \mathcal{D}_{v_k^* v_k^*} u \right) \mathcal{D}_{v_j^* v_j^*}$$

where the v_j^* are the directions active in the minimum in (MA^M) . It is certainly possible that the active directions change from iterate to iterate - this does not affect the convergence properties of the algorithm.

Thus the corrector is obtained by solving the weighted average of the two linearizations

$$(w(x)\nabla_u MA^N[u^n] + (1 - w(x))\nabla_u MA^M[u^n]) v^n = w(x)MA^N[u^n] + (1 - w(x))MA^M[u^n]. \quad (13)$$

In order for the linear Eq. (11) to be well-posed, we require the coefficient matrix to be positive definite. As observed in Remark 4, this condition can fail if the iterate u^n is not strictly convex.

7.1. Initialization of Newton's method

Newton's method requires a good initialization for the iteration. Since we need the resulting linear system to be well posed it is essential that the initial iterate: (i) be convex, (ii) respect the boundary conditions, (iii) be close to the solution.

In order to do this, we first use one step of the semi-implicit scheme (9), to obtain a close initial value. This amounts to solving (10) along with consistent Dirichlet boundary conditions (D). Then convexify the result, using the method of [38]. Since both the steps can be performed on a very coarse grid, and interpolated onto the finer grid, the cost of the initialization is low.

7.2. Regularization

In degenerate examples, the PDE for v^n (13) may be degenerate, which can lead to an ill-conditioned or singular Jacobian. To get around this problem, we regularize the Jacobian to make sure the linear operator is strictly negative definite; this will not change the fixed points of Newton's method. We accomplish this by replacing the second directional derivatives u_{vv} with

$$\tilde{u}_{vv} = \max\{u_{vv}, \epsilon\}.$$

Here ϵ is a small parameter. In the computations of Section 8, we take $\epsilon = \frac{1}{2dx^2} \times 10^{-8}$.

7.3. Solution of the linear system

The linear equations given by Newton's method were solved using LU decomposition. Since the problem is sparse and highly structured, the LU decomposition was effective for the problem sizes we considered. A possibly more efficient alternative would be a preconditioned inexact Newton–Krylov solver, as in [15].

8. Computational results in two-dimensions

In this section, we summarize the results of a number of two-dimensional examples solved using the hybrid scheme described in Section 5. In particular, we are interested in comparing the computation time for Newton's method with the time required by the methods proposed in [29]. We also visualize the map generated by the gradient of the solution.

These computations are performed on an $N \times N$ grid on the square $[0, 1]^2$. The monotone scheme used a 17 point stencil.

When needed as part of the initialization, the convex envelope is computed on a coarse grid using the discretization described in [38]. Since the solution can be computed on a coarse grid, and interpolated, the added computational time is negligible.

8.1. Failure of Newton's method for natural finite differences

In this section, we give an example where Newton's method breaks down when standard finite differences are used.

We chose an example which is only singular at one point, on the boundary. Nevertheless, this mild singularity is enough for Newton's method to break down.

Consider the solution of (MA) in $[0, 1]^2$, given by:

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

The gradient of the solution is unbounded on $|\mathbf{x}|^2 = 2$, in particular at the point $(1, 1)$. The singularity arises from the fact that f is unbounded there.

Due to the singularity, there is an instability in Newton's method if the natural finite difference method is used. The iteration is initialized with the exact solution. The result after performing two iterations of Newton's method along with the gradient map, is illustrated in Fig. 2. The correct computed solution is presented in Fig. 3(g) and (h).

8.2. Four representative examples

We have tested the hybrid method on a number of examples of varying regularity; the results are summarized in Sections 8.4, 8.3. To illustrate these results, we present more detailed results for four representative examples.

Write $\mathbf{x}_0 = (.5, .5)$ for the center of the domain.

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

$$u(\mathbf{x}) = \exp\left(\frac{|\mathbf{x}|^2}{2}\right), \quad f(\mathbf{x}) = (1 + |\mathbf{x}|^2) \exp(|\mathbf{x}|^2). \quad (14)$$

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

$$u(\mathbf{x}) = \frac{1}{2}((|\mathbf{x} - \mathbf{x}_0| - 0.2)^+)^2, \quad f(\mathbf{x}) = \left(1 - \frac{0.2}{|\mathbf{x} - \mathbf{x}_0|}\right)^+. \quad (15)$$

The third example is the one which was used in Section 8.1 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:

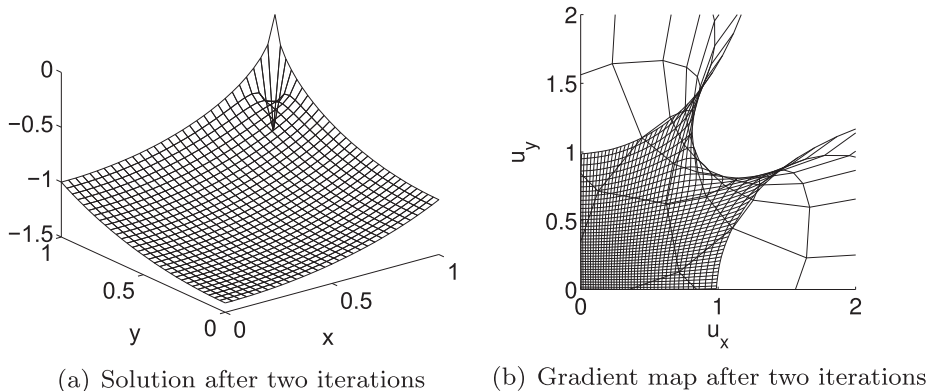


Fig. 2. Failure of Newton's method using standard finite differences: the solution oscillates and becomes non-convex.

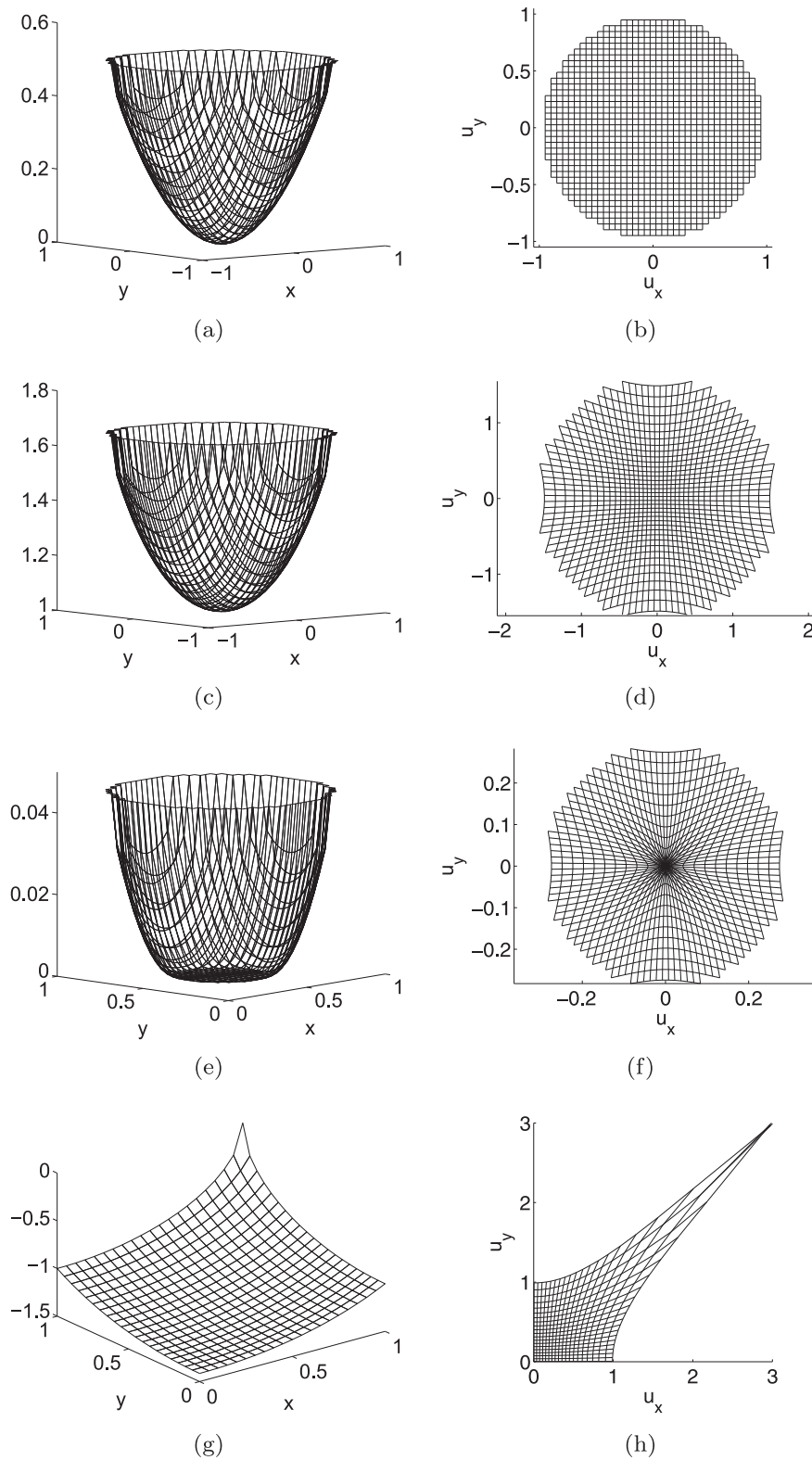


Fig. 3. Solutions and mappings for the (a), (b) identity map, (c), (d) C^2 example, (e), (f) C^1 example, and (g), (h) example with blow-up.

$$u(\mathbf{x}) = -\sqrt{2 - |\mathbf{x}|^2}, \quad f(\mathbf{x}) = 2(2 - |\mathbf{x}|^2)^{-2}. \quad (16)$$

This final example solution is the cone, which was discussed in Section 2.3. It is Lipschitz continuous.

$$u(\mathbf{x}) = \sqrt{|\mathbf{x} - \mathbf{x}_0|}, \quad f = \mu = \pi \delta_{\mathbf{x}_0}. \quad (17)$$

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

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

8.3. Visualization of solutions and gradient maps

In Fig. 3 the solutions and the gradient maps for the three representative examples are presented. For example (17) the gradient map is too singular to illustrate. To visualize the maps, the image of a Cartesian mesh under the mapping

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} \mathcal{D}_{x_1} u \\ \mathcal{D}_{x_2} u \end{pmatrix},$$

is shown, where $(\mathcal{D}_{x_1} u, \mathcal{D}_{x_2} u)$ is the numerical gradient of the solution of the Monge–Ampère equation. The image of a circle is plotted for visualization purposes; the equation is solved on a square. For reference, the identity mapping is also displayed.

In each case, the maps agree with the maps obtained using the gradient of the exact solution.

Table 1

Computation times for the Newton method (current work), and the Poisson and Gauss–Seidel methods (from [29]) for four representative examples.

| N | Newton Iterations | CPU Time (seconds) | | |
|------------------------------------|-------------------|--------------------|---------|--------------|
| | | Newton | Poisson | Gauss–Seidel |
| C^2 Example (14) | | | | |
| 31 | 3 | 0.2 | 0.7 | 2.2 |
| 45 | 4 | 0.2 | 1.1 | 4.1 |
| 63 | 4 | 0.4 | 1.9 | 15.0 |
| 89 | 4 | 1.0 | 4.8 | 57.6 |
| 127 | 5 | 2.9 | 9.6 | 236.7 |
| 181 | 5 | 9.0 | 23.2 | 1004.0 |
| 255 | 5 | 30.5 | 52.6 | – |
| 361 | 6 | 131.4 | 162.6 | – |
| C^1 Example (15) | | | | |
| 31 | 4 | 0.4 | 1.1 | 0.8 |
| 45 | 6 | 0.4 | 6.1 | 2.8 |
| 63 | 7 | 0.8 | 20.5 | 9.5 |
| 89 | 9 | 2.0 | 80.0 | 35.9 |
| 127 | 11 | 5.7 | 256.8 | 145.5 |
| 181 | 13 | 17.7 | – | 558.0 |
| 255 | 16 | 55.3 | – | – |
| 361 | 20 | 200.0 | – | – |
| Example with blow-up (16) | | | | |
| 31 | 4 | 0.2 | 0.5 | 0.8 |
| 45 | 4 | 0.4 | 1.4 | 5.3 |
| 63 | 4 | 0.7 | 2.9 | 19.4 |
| 89 | 5 | 1.8 | 8.1 | 74.1 |
| 127 | 7 | 5.1 | 17.7 | 293.3 |
| 181 | 7 | 12.9 | 51.4 | 1637.1 |
| 255 | 7 | 36.1 | 128.2 | – |
| 361 | 8 | 152.9 | 374.5 | – |
| $C^{0,1}$ (Lipschitz) Example (17) | | | | |
| 31 | 9 | 0.5 | 5.3 | 0.8 |
| 45 | 11 | 0.6 | 27.8 | 5.9 |
| 63 | 15 | 1.4 | 91.9 | 21.5 |
| 89 | 22 | 4.3 | 451.0 | 90.5 |
| 127 | 32 | 14.1 | 1758.2 | 373.9 |
| 181 | 30 | 34.6 | – | 1576.1 |
| 255 | 34 | 101.7 | – | – |
| 361 | 29 | 280.2 | – | – |

8.4. Computation time

Computations were performed on a laptop with a 2 GHz Intel processor, using MATLAB running in the Windows Vista operating system. The implementation could be made much more efficient in various ways: for example using a compiled programming language. However the results we obtained were sufficient to demonstrate the efficiency of the method.

The computation times for the four representative examples are presented in Table 1. The computations time are compared to those for the Gauss–Seidel and Poisson iterations described in [29]. The Newton solver is faster in terms of absolute solution time in each case. Table 2 presents order of magnitude solutions times. The solution time for Newton’s method is independent of the regularity of the solutions and faster than both of the other methods.

Table 2

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

| Method | Regularity of Solution | | |
|--------------|---|--|---|
| | $C^{2,\alpha}$ (14) | $C^{1,\alpha}$ (15) and (16) | $C^{0,1}$ (17) |
| Gauss–Seidel | Moderate ($\sim \mathcal{O}(M^{1.8})$) | Moderate ($\sim \mathcal{O}(M^{1.9})$) | Moderate ($\sim \mathcal{O}(M^2)$) |
| Poisson | Fast ($\sim \mathcal{O}(M^{1.4})$) | Fast–Slow ($\sim \mathcal{O}(M^{1.4})$ –blow-up) | Slow ($\sim \mathcal{O}(M^2)$ –blow-up) |
| Newton | Fast ($\sim \mathcal{O}(M^{1.3})$) | Fast ($\sim \mathcal{O}(M^{1.3})$) | Fast ($\sim \mathcal{O}(M^{1.3})$) |

Table 3

Accuracy for the standard, monotone, and hybrid discretizations for four representative examples.

| N | Maximum Error | | |
|--|------------------------|------------------------|------------------------|
| | Standard | Monotone | Hybrid |
| <i>C^2 Example (14)</i> | | | |
| 31 | 7.14×10^{-5} | 89.09×10^{-5} | 24.45×10^{-5} |
| 45 | 3.39×10^{-5} | 60.50×10^{-5} | 15.29×10^{-5} |
| 63 | 1.73×10^{-5} | 50.88×10^{-5} | 9.06×10^{-5} |
| 89 | 0.87×10^{-5} | 47.51×10^{-5} | 5.32×10^{-5} |
| 127 | 0.43×10^{-5} | 45.53×10^{-5} | 3.02×10^{-5} |
| 181 | 0.21×10^{-5} | 44.65×10^{-5} | 1.61×10^{-5} |
| 255 | 0.11×10^{-5} | 44.22×10^{-5} | 0.87×10^{-5} |
| 361 | 0.05×10^{-5} | 44.00×10^{-5} | 0.46×10^{-5} |
| <i>C^1 Example (15)</i> | | | |
| 31 | 2.6×10^{-4} | 17.5×10^{-4} | 12.2×10^{-4} |
| 45 | 1.8×10^{-4} | 11.6×10^{-4} | 5.9×10^{-4} |
| 63 | 1.5×10^{-4} | 9.8×10^{-4} | 4.2×10^{-4} |
| 89 | 0.9×10^{-4} | 8.4×10^{-4} | 2.6×10^{-4} |
| 127 | 0.6×10^{-4} | 7.9×10^{-4} | 2.0×10^{-4} |
| 181 | 0.4×10^{-4} | 7.4×10^{-4} | 1.2×10^{-4} |
| 255 | – | 7.2×10^{-4} | 1.0×10^{-4} |
| 361 | – | 7.0×10^{-4} | 0.7×10^{-4} |
| <i>Example with blow-up (16)</i> | | | |
| 31 | 17.15×10^{-3} | 1.74×10^{-3} | 1.74×10^{-3} |
| 45 | 14.59×10^{-3} | 0.98×10^{-3} | 0.98×10^{-3} |
| 63 | 12.53×10^{-3} | 0.59×10^{-3} | 0.59×10^{-3} |
| 89 | 10.67×10^{-3} | 0.37×10^{-3} | 0.35×10^{-3} |
| 127 | 9.00×10^{-3} | 0.35×10^{-3} | 0.20×10^{-3} |
| 181 | 7.59×10^{-3} | 0.34×10^{-3} | 0.12×10^{-3} |
| 255 | 6.42×10^{-3} | 0.33×10^{-3} | 0.07×10^{-3} |
| 361 | 5.41×10^{-3} | 0.33×10^{-3} | 0.04×10^{-3} |
| <i>$C^{0,1}$ (Lipschitz) Example (17)</i> | | | |
| 31 | 10×10^{-3} | 3×10^{-3} | 3×10^{-3} |
| 45 | 8×10^{-3} | 3×10^{-3} | 3×10^{-3} |
| 63 | 6×10^{-3} | 3×10^{-3} | 3×10^{-3} |
| 89 | 4×10^{-3} | 4×10^{-3} | 4×10^{-3} |
| 127 | 3×10^{-3} | 4×10^{-3} | 4×10^{-3} |
| 181 | 2×10^{-3} | 4×10^{-3} | 4×10^{-3} |
| 255 | – | 4×10^{-3} | 4×10^{-3} |
| 361 | – | 4×10^{-3} | 4×10^{-3} |

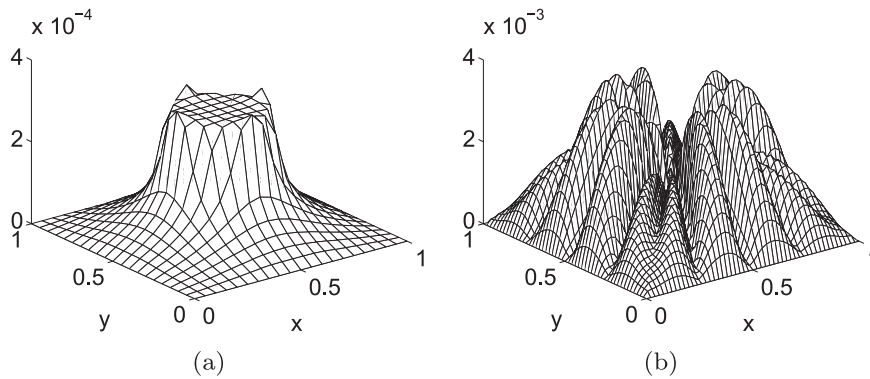


Fig. 4. Surface plots of error using the hybrid scheme for the (a) C^1 example and (b) cone example.

8.5. Accuracy

Numerical errors are presented in Table 3. We compare the accuracy of the hybrid scheme to the standard finite difference discretization, (using the results of [29]) and to the monotone scheme which was also solved using Newton's method. We discuss each example in turn.

The C^2 solution (14)

The standard finite difference schemes gives $\mathcal{O}(h^2)$ accuracy (see [29]). In this case, the hybrid scheme is slightly *less* accurate, because the monotone scheme is used near the boundary. On a strictly convex domain the hybrid scheme would reduce to the standard discretization and achieve the same accuracy.

The effect diminishes as the number of grid points increases so that the number of interior points using the higher order scheme dominates. Accuracy approaches $\mathcal{O}(h^2)$ as the number of grid points increases. This is a definite improvement over the monotone scheme, which has its accuracy limited by the stencil width.

The C^1 solution (15)

The accuracy is $\mathcal{O}(h)$, which is similar to the standard discretization and better than the limited accuracy permitted by the monotone discretization with a fixed stencil width. We also look at the error at each point (see Fig. 4); it is evident that the singularity around the circle is the factor that most affects the accuracy. Because of this non-smoothness, there is no reason to expect our scheme to produce the $\mathcal{O}(h^2)$ accuracy that is possible on C^2 solutions.

The blow-up solution (16)

In this case, the hybrid scheme accuracy is $\mathcal{O}(h^{1.5})$. This is better than the accuracy of both the standard discretization, which was $\mathcal{O}(h^{0.5})$ [29], and the monotone scheme, which is limited by the stencil width.

Table 4

Maximum error and computation time for the hybrid Newton's method on three representative examples in the three-dimensional case.

| N | Max Error | Iterations | CPU Time (s) |
|--------------------------------------|----------------------|------------|--------------|
| <i>C^2 Example (18)</i> | | | |
| 7 | 0.0151 | 2 | 0.04 |
| 11 | 0.0140 | 3 | 0.10 |
| 15 | 0.0129 | 5 | 0.71 |
| 21 | 0.0121 | 6 | 6.72 |
| 31 | 0.0111 | 5 | 86.63 |
| <i>C^1 Example (19)</i> | | | |
| 7 | 0.0034 | 1 | 0.02 |
| 11 | 0.0022 | 1 | 0.09 |
| 15 | 0.0016 | 1 | 0.22 |
| 21 | 0.0009 | 1 | 1.03 |
| 31 | 0.0005 | 1 | 17.12 |
| <i>Example with blow-up (20)</i> | | | |
| 7 | 9.6×10^{-3} | 1 | 0.03 |
| 11 | 5.2×10^{-3} | 3 | 0.11 |
| 15 | 4.6×10^{-3} | 3 | 0.48 |
| 21 | 4.0×10^{-3} | 6 | 7.42 |
| 31 | 2.9×10^{-3} | 8 | 138.74 |

The cone solution (17)

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. We observed that the 17 point stencil reduced the error by an order of magnitude compared to the 9 point stencil. This dependence on the stencil width is also evident in the surface plot of error (Fig. 4), which demonstrates that error is largest along directions that are not captured by the stencil. Since this solution is so singular, the reduced accuracy is to be expected.

9. Computational results in 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. The size of the computation was restricted by the available memory, not by solution time.

The solution methods of [29] were restricted to the two-dimensional Monge–Ampère equation, so we are no longer able to compare solution times to Newton's method for these examples.

As before, we provide specific results for three representative examples of varying regularity. In this section we use the notation

$$\mathbf{x} = (x, y, z)$$

and let $\mathbf{x}_0 = (.5, .5, .5)$ be the centre of the domain.

The first example is the C^2 solution given by:

$$u(\mathbf{x}) = \exp\left(\frac{|\mathbf{x}|^2}{2}\right), \quad f(\mathbf{x}) = (1 + |\mathbf{x}|^2) \exp\left(\frac{3}{2}|\mathbf{x}|^2\right). \quad (18)$$

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

$$u(\mathbf{x}) = \frac{1}{2}((|\mathbf{x} - \mathbf{x}_0| - 0.2)^+)^2, \quad (19)$$

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

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

$$u(\mathbf{x}) = -\sqrt{3 - |\mathbf{x}|^2}, \quad f(\mathbf{x}) = 3(3 - |\mathbf{x}|^2)^{-5/2}. \quad (20)$$

Solution times for the hybrid Newton's method in three-dimensions are presented in Table 4. (The fact that the solver required only one iteration for Example (19) was simply an artifact – for larger problems sizes more iterations were required.)

10. Conclusions

The purpose of this work was to build a fast, accurate finite difference solver for the elliptic Monge–Ampère equation.

A hybrid finite difference discretization was used which selects between an accurate standard finite difference discretization and a stable (provably convergent) monotone discretization. The choice of discretization was based on known regularity results which depended on the boundary data, g , the right hand side function f , and strict convexity of the domain. Wherever the requirements on the data are not met, the hybrid discretization chooses the monotone discretization.

We have resolved the problem of ill-conditioning near singular solutions faced by existing solution methods. Given a right hand side function f which is zero or very large, the linearized system becomes degenerate. Existing methods exhibit an increase in solution times by a factor of 30 going from the smooth to even the moderately singular case.

By carefully choosing a convergent discretization, we produced linear problems for Newton's method which are better conditioned. In fact, the monotone (or hybrid) discretization was shown to be necessary for stability of Newton's method: an example with a mildly singular solution showed that the standard discretization leads to instabilities. In our case, the discretized equations were solved by Newton's method, which is fast, $\mathcal{O}(M^{1.3})$, where M is the number of data points, (or half a minute on a 256^2 grid), independent of the regularity of the solution. We also computed the completely degenerate example (where f is an approximation of a delta function), which has not been computed by other methods. In this case, our solution times increased by a factor of less than three.

Compared to our two previous methods, the implementation of Newton's method was significantly (orders of magnitude) faster.

The hybrid discretization was introduced to improve the accuracy of the monotone discretization on regular solutions. This expected improvement was achieved. On regular solutions the hybrid solver was (asymptotically) as accurate as the

standard finite difference discretization. For one moderately singular example the hybrid solver was more accurate than standard finite differences by $\mathcal{O}(h)$.

The discretization and solution method used was not restricted to two-dimensions. This allowed for the solution of three-dimensional problems on moderate (32^3) sized grids.

In summary, the solver presented used a novel discretization in general-dimensions, accompanied by a fast solution method. The resulting solver is a significant improvement over existing methods for the solution of possibly singular solutions of the elliptic Monge–Ampère equation, in terms of solution time, stability, and accuracy.

References

- [1] L.C. Evans, Partial differential equations and Monge–Kantorovich mass transfer, Current Developments in Mathematics, Int. Press, Boston, MA, Cambridge, MA, 1997.
- [2] I. Bakelman, Convex Analysis and Nonlinear Geometric Elliptic Equations, Springer-Verlag, 1994.
- [3] A.V. Pogorelov, Monge–Ampère equations of elliptic type, Translated from the first Russian edition by F. Leo Boron with the assistance of Albert L. Rabenstein, C. Richard, and P. Bollinger, Noordhoff Ltd., Groningen, 1964.
- [4] V.I. Oliker, 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 (1988) 271–293.
- [5] G.J. Haltiner, Numerical Weather Prediction, Wiley, New York, 1971.
- [6] A. Kasahara, Significance of non-elliptic regions in balanced flows of the tropical atmosphere, Mon. Weather Rev. 110 (1982).
- [7] J.J. Stoker, Nonlinear Elasticity, Gordon and Breach Science Publishers, New York, 1968.
- [8] B.S. Westcott, Shaped Reflector Antenna Design, Research Studies Press, New York, 1983.
- [9] S.D. Stojanovic, Risk premium and fair option prices under stochastic volatility: the hara solution, C. R. Math. 340 (2005) 551–556.
- [10] L. Ambrosio, Lecture notes on optimal transport problems, in: Mathematical aspects of evolving interfaces (Funchal, 2000), Lecture Notes in Math, Springer, Berlin, 2003, pp. 1–52. vol. 1812.
- [11] C. Villani, Topics in optimal transportation, Graduate Studies in Mathematics, 58, American Mathematical Society, Providence, RI, 2003.
- [12] S. Haker, A. Tannenbaum, R. Kikinis, Mass preserving mappings and image registration, MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer-Verlag, London, UK, 2001.
- [13] S. Haker, L. Zhu, A. Tannenbaum, S. Angenent, Optimal mass transport for registration and warping, Int. J. Comput. Vision 60 (2004) 225–240.
- [14] T. ur Rehman, E. Haber, G. Pryor, J. Melonakos, A. Tannenbaum, 3D nonrigid registration via optimal mass transport on the GPU, Med. Image. Anal. 13 (2009) 931–940.
- [15] G.L. Delzanno, L. Chacón, J.M. Finn, Y. Chung, G. Lapenta, An optimal robust equidistribution method for two-dimensional grid adaptation based on Monge–Kantorovich optimization, J. Comput. Phys. 227 (2008) 9841–9864.
- [16] J.M. Finn, G.L. Delzanno, L. Chacón, Grid generation and adaptation by Monge–Kantorovich optimization in two and three-dimensions, in: Proceedings of the 17th International Meshing Roundtable, pp. 551–568.
- [17] C.J. Budd, J.F. Williams, Moving mesh generation using the parabolic Monge–Ampère equation, SIAM J. Sci. Comput. 31 (2009) 3438–3465.
- [18] T. Glimm, V. Oliker, Optical design of single reflector systems and the Monge–Kantorovich mass transfer problem, J. Math. Sci. (N.Y.) 117 (2003) 4096–4108. Nonlinear problems and function theory.
- [19] U. Frisch, S. Matarrese, R. Mohayaee, A. Sobolevski, A reconstruction of the initial conditions of the universe by optimal mass transportation, Nature 417 (2002).
- [20] E. Haber, R. Horeish, J. Modersitski, Numerical optimization for constrained image registration, 2010, preprint.
- [21] D. Cohen-Or, Space deformations, surface deformations and the opportunities in-between, J. Comput. Sci. Technol. 24 (2009) 2–5.
- [22] J.-D. Benamou, Y. Brenier, A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem, Numer. Math. 84 (2000) 375–393.
- [23] E.J. Dean, R. Glowinski, On the numerical solution of the elliptic Monge–Ampère equation in dimension two: a least-squares approach, in: Partial differential equations, Computational Methods in Applied Science, vol. 16, Springer, Dordrecht, 2008, pp. 43–63.
- [24] E.J. Dean, 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 (2006) 71–96. electronic.
- [25] R. Glowinski, Numerical methods for fully nonlinear elliptic equations, in: R. Jeltsch, G. Wanner (Eds.), 6th International Congress on Industrial and Applied Mathematics, ICIAM 07, Invited Lectures, pp. 155–192.
- [26] X. Feng, M. Neilan, Mixed finite element methods for the fully nonlinear Monge–Ampère equation based on the vanishing moment method, SIAM J. Numer. Anal. 47 (2009) 1226–1250.
- [27] X. Feng, M. Neilan, Vanishing moment method and moment solutions for fully nonlinear second order partial differential equations, J. Sci. Comput. 38 (2009) 74–98.
- [28] G. Loeper, F. Rapetti, Numerical solution of the Monge–Ampère equation by a Newton's algorithm, C.R. Math. Acad. Sci. Paris 340 (2005) 319–324.
- [29] J.-D. Benamou, B.D. Froese, A.M. Oberman, Two numerical methods for the elliptic Monge–Ampère equation, ESAIM: Math. Model. Numer. Anal. 44 (2010).
- [30] V. Zheligovsky, O. Podvigina, U. Frisch, The Monge–Ampère equation: Various forms and numerical solution, J. Comput. Phys. 229 (2010) 5043–5061.
- [31] A.M. Oberman, Wide stencil finite difference schemes for the elliptic Monge–Ampère equation and functions of the eigenvalues of the Hessian, Discrete Cont. Dyn. Syst. Ser. B 10 (2008) 221–238.
- [32] B.D. Froese, A.M. Oberman, Convergent finite difference solvers for viscosity solutions of the Monge–Ampère equation in dimensions two and higher (2010). <<http://arxiv.org/abs/1007.0765>>.
- [33] L. Caffarelli, L. Nirenberg, J. Spruck, The Dirichlet problem for nonlinear second-order elliptic equations. I. Monge–Ampère equation, Commun. Pure Appl. Math. 37 (1984) 369–402.
- [34] J.I.E. Urbas, The generalized Dirichlet problem for equations of Monge–Ampère type, Ann. Inst. H. Poincaré Anal. Non Linéaire 3 (1986) 209–228.
- [35] L.A. Caffarelli, Interior $W^{2,p}$ estimates for solutions of the Monge–Ampère equation, Ann. Math. 131 (2) (1990) 135–150.
- [36] C.E. Gutiérrez, The Monge–Ampère equation, Progress in Nonlinear Differential Equations and their Applications, vol. 44, Birkhäuser Boston Inc., Boston, MA, 2001.
- [37] A.V. Pogorelov, The Minkowski multidimensional problem, in: V.H. Winston & Sons, Washington, D.C., Translated from the Russian by Vladimir Oliker, Introduction by Louis Nirenberg, Scripta Series in Mathematics, 1978.
- [38] A.M. Oberman, Computing the convex envelope using a nonlinear partial differential equation, Math. Models Methods Appl. Sci. 18 (2008) 759–780.
- [39] A.M. Oberman, L. Silvestre, The Dirichlet problem for the convex envelope, Trans. Amer. Math. Soc., in press <<http://arxiv.org/abs/1007.0773>>.
- [40] A.V. Pogorelov, The Dirichlet problem for the multidimensional analogue of the Monge–Ampère equation, Dokl. Akad. Nauk SSSR 201 (1971) 790–793.
- [41] M.G. Crandall, H. Ishii, P.-L. Lions, User's guide to viscosity solutions of second order partial differential equations, Bull. Amer. Math. Soc. (N.S.) 27 (1992) 1–67.

- [42] G. Strang, *Linear Algebra and its Applications*, second ed., Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1980.
- [43] A.M. Oberman, Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–Jacobi equations and free boundary problems, *SIAM J. Numer. Anal.* 44 (2006) 879–895. electronic.
- [44] D.P. Bertsekas, *Convex analysis and optimization*, Athena Scientific, Belmont, MA, 2003. With Angelia Nedić and Asuman E. Ozdaglar.