Commun. Optim. Theory 2022 (2022) 7 https://doi.org/10.23952/cot.2022.7

MATHRES Communications in Optimization Theory
Available online at http://cot.mathres.org

AN EFFICIENT OPERATOR–SPLITTING METHOD FOR THE EIGENVALUE PROBLEM OF THE MONGE-AMPÈRE EQUATION

HAO LIU[1], SHINGYU LEUNG[2,*], JIANLIANG QIAN[3]

[1]Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong
[2]Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[3]Department of Mathematics and Department of CMSE, Michigan State University, East Lansing, MI 48824, USA

Dedicated to the memory of Professor Roland Glowinski

Abstract. We develop an efficient operator–splitting method for the eigenvalue problem of the Monge–Ampère operator in the Aleksandrov sense. The backbone of our method relies on a convergent Rayleigh inverse iterative formulation proposed by Abedin and Kitagawa (Inverse iteration for the Monge–Ampère eigenvalue problem, Proceedings of the American Mathematical Society, 148 (2020) 4975-4886). Modifying the theoretical formulation, we develop an efficient algorithm for computing the eigenvalue and eigenfunction of the Monge–Ampère operator by solving a constrained Monge–Ampère equation during each iteration. Our method consists of four essential steps: (i) Formulate the Monge–Ampère eigenvalue problem as an optimization problem with a constraint; (ii) Adopt an indicator function to treat the constraint; (iii) Introduce an auxiliary variable to decouple the original constrained optimization problem into simpler optimization subproblems and associate the resulting new optimization problem with an initial value problem; and (iv) Discretize the resulting initial-value problem by an operator–splitting method in time and a mixed finite element method in space. The performance of our method is demonstrated by several experiments. Compared to existing methods, the new method is more efficient in terms of computational cost and has a comparable rate of convergence in terms of accuracy.

Keywords. Eigenvalue problem; Monge–Ampère equation; Inverse iteration; Operator–splitting method.

1. INTRODUCTION

The Monge-Ampère equation is a second-order fully nonlinear PDE in the form of

$$\det \mathbf{D}^2 u = f, \tag{1.1}$$

*Corresponding author.
E-mail address: haoliu@hkbu.edu.hk (H. Liu), masyleung@ust.hk (S. Leung), jqian@msu.edu (J. Qian).
Received October 15, 2021; Accepted June 23, 2022.

where $\mathbf{D}^2u$ denotes the Hessian of u. The Monge-Ampère equation originates from differential geometry in which it describes a surface with prescribed Gaussian curvature [3, 35]. The existence, uniqueness and regularity of the solution has been extensively studied [3, 26, 44], and related applications can be found in optimal transport [4, 23], seismology [17], image processing [33], finance [45], and geostrophic flows [21].

Due to its broad applications, in the past decade, a lot of efforts have been devoted to developing numerical methods for the Monge-Ampère equation. One line of research is to develop wide-stencil based finite-difference schemes [24, 25] for equation (1.1) with Dirichlet boundary conditions. Such a class of methods utilizes the fact that $\det\mathbf{D}^2u$ equals the product of the eigenvalues of $\mathbf{D}^2u$, so that these methods use wide-stencils to estimate the eigenvalues. Later on, such methods were extended to accommodate transport boundary conditions in [23]. Another line of research is to design finite-element based methods. In [20, 22], the authors proposed the vanishing moment method, which approximates a fully nonlinear second-order PDE by a fourth-order PDE. In [9, 10, 14, 15], the authors formulate equation (1.1) as an optimization problem. Fast augmented Lagrangian algorithms are then designed to solve the new problems. Recently, operator–splitting methods have been proposed in [30, 39]. Taking advantage of the divergence form of $\det\mathbf{D}^2u$, the authors of [30, 39] decouple the nonlinearity of equation (1.1) by introducing an auxiliary variable so that solving equation (1.1) is reduced to finding the steady-state solution of an initial value problem, which is time-discretized by an operator–splitting method and space-discretized by a mixed finite-element method. Other numerical methods for equation (1.1) include [2, 5, 6, 11, 12, 19]; see the survey [18] for more related works.

Existing works discussed above target equation (1.1) with various boundary conditions. Another interesting problem of the Monge–Ampère type is the eigenvalue problem, reading as

$$\begin{cases} \det(\mathbf{D}^2u) = \lambda\,|u|^d & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{1.2}$$

where $\Omega \subset \mathbb{R}^d$ $(d \geq 2)$ is an open bounded convex domain, and $\lambda = \lambda[\Omega]$ is the unknown eigenvalue of the Monge–Ampère operator on Ω. Problem (1.2) was first studied by Lions in [38] and later by Tso in [46]. They proved the existence, uniqueness and regularity of the solution on an open, bounded, smooth, uniformly convex domain. The result was then extended by Le in [36] to general bounded convex domains. Theoretically, to find the solution of equation (1.2), a variational formulation was proposed in [46], and a convergent Rayleigh quotient inverse iterative formulation was proposed in [1] which was further improved in [37]. Since, during each Rayleigh quotient iteration, the algorithm in [1] requires solving a Monge–Ampère type equation, how to efficiently implement this formulation numerically has not been studied. The only work on the numerical solution of equation (1.2) we are aware of is [28], in which the authors proposed operator–splitting methods for a class of Monge-Ampère eigenvalue problem. In [28], taking advantage of the divergence form, the authors takes equation (1.2) as the optimality condition of a constrained optimization problem, in which λ is considered as the Lagrange multiplier, and an operator–splitting method was proposed to solve the new problem.

Similar to equation (1.1), the eigenvalue problem (1.2) is a fully nonlinear second-order PDE. One effective way to solve such PDEs is the operator–splitting method, which decomposes

complicated problems into several easy–to–solve subproblems by introducing auxiliary variables. Then the new problem will be formulated as solving an initial value problem, which is then time discretized using operator–splittings. All variables will be updated in an alternative fashion, where each subproblem either has an explicit solution or can be solved efficiently. The operator–splitting method has been applied to numerically solving PDEs [30, 39], image processing [16, 40, 41, 42], surface reconstruction [34], inverse problems [29], obstacle problems [43], and computational fluid dynamics [7, 8]. We refer readers to monographs [31, 32] for detailed discussions on operator–splitting methods.

In this work, we propose an efficient numerical implementation of the formulation proposed in [1] to compute the eigenvalue and eigenfunction of the Monge–Ampère operator on an open, bounded, convex domain Ω. Since each Rayleigh quotient inverse iteration of the formulation in [1] requires solving a Monge–Ampère equation, we first use the divergence form of the Monge–Ampère operator to rewrite the problem as an optimization problem. To stabilize our formulation, we consider a constrained version of the optimization problem by forcing the eigenfunction u to have unit L_2-norm: $\|u\|_2 = 1$. The constrained problem is converted to an unconstrained problem by utilizing an indicator function of the constraint set. Then we decouple the nonlinearity of the functional by introducing an auxiliary variable, and we associate it with an initial value problem in the flavor of gradient flow. The initial value problem is time discretized by an operator-splitting method and space discretized by a mixed finite-element method in the space of piecewise-linear continuous functions. The efficiency of the proposed method is demonstrated by several numerical experiments.

We organize the rest of this article as follows: We introduce the background and summarize the convergent formulation of [1] for equation (1.2) in Section 2. Our new operator-splitting approach for implementing this convergent formulation is presented in Section 3. Our operator-splitting scheme is time discretized in Section 4 and space discretized in Section 5. We demonstrate the efficiency of the proposed method by several numerical experiments in Section 6 and conclude this article in Section 7.

2. A Convergent Inverse Iteration for the Eigenvalue Problem

Let $\Omega \subset \mathbb{R}^d$ be an open bounded convex domain. In equation (1.2), if u is a convex function, one has $u \leq 0$ and $|u| = -u$. The existence and uniqueness of the eigen-pair was studied in [38]:

Theorem 2.1. *Assume that $\Omega \subset \mathbb{R}^d$ is a smooth, bounded, uniformly convex domain. There exist a unique positive constant λ_{MA} and a unique (up to positive multiplicative constants) nonzero convex function $u \in C^{1,1}(\bar{\Omega}) \cap C^{\infty}(\Omega)$ solving the eigenvalue problem (1.2). The constant λ_{MA} is called the Monge-Ampère eigenvalue of Ω and u is called a Monge-Ampère eigenfunction of Ω.*

Define the Rayleigh quotient of a function u for the Monge-Ampère operator as

$$R(u) = \frac{\int_\Omega -u \det(\mathbf{D}^2 u) d\mathbf{x}}{\int_\Omega (-u)^{d+1} d\mathbf{x}}, \tag{2.1}$$

and the function space $\mathscr{K}$ as

$$\mathscr{K} = \left\{ u \in C^{0,1}(\bar{\Omega}) \cap C^{\infty}(\Omega) : u \text{ is convex and nonzero in } \Omega,\ u = 0 \text{ on } \partial\Omega \right\}.$$

Tso [46] showed that λ_{MA} can be written as the infimum of Rayleigh quotients:

Theorem 2.2. *Assume that $\Omega \subset \mathbb{R}^d$ is a smooth, bounded and uniformly convex domain. Then*

$$\lambda_{\mathrm{MA}} = \inf_{u \in \mathscr{K}} R(u). \tag{2.2}$$

Based on the property (2.2), the following inverse iterative scheme for the eigenvalue problem (1.2) was proposed by Abedin and Kitagawa in [1]:

$$\begin{cases} u^0 = u_0, \\ \det(\mathbf{D}^2 u^{k+1}) = R(u^k)|u^k|^d & \text{in } \Omega, \\ u^{k+1} = 0 & \text{on } \partial\Omega, \end{cases} \tag{2.3}$$

where u_0 is a given initial condition, and they further proved the convergence of the inverse iteration:

Theorem 2.3. *Assume that $\Omega \subset \mathbb{R}^d$ is an open bounded convex domain. Let $u_0 \in C(\bar{\Omega})$ satisfy the following:*

(i) u_0 is convex and $u_0 \leq 0$ on $\partial\Omega$;
(ii) $R(u_0) < \infty$;
(iii) $\det(\mathbf{D}^2 u_0) \geq c_0$ in Ω, where c_0 is some positive constant.

Then, for $k > 0$, u^k in equation (2.3) converges uniformly on $\bar{\Omega}$ to a nonzero Monge-Ampère eigenfunction, and $R(u_k)$ converges to λ_{MA}.

Theorem 2.3 was improved in [37] so that conditions (i) and (iii) are removed; consequently, the inverse iteration converges for all convex initial data having finite and nonzero Rayleigh quotient to a nonzero Monge-Ampère eigenfunction of Ω.

3. A Modified Formulation of the Inverse Iteration

Given an initial convex function u_0 with bounded nonzero Rayleigh quotient, the inverse iteration (2.3) generates the sequence $\{(R(u^k), u^k)\}$ which is guaranteed to converge to the solution of the eigenvalue problem (1.2). When updating u^{k+1} from u^k, one needs to solve a Monge-Ampère equation with the Dirichlet boundary condition, which is a nonlinear problem. It has not been studied yet how to implement the inverse iteration efficiently to produce numerical approximations to the eigenvalue problem of the Monge-Ampère operator. Therefore, we are motivated to develop an efficient algorithm to implement this inverse iterative method.

To achieve this purpose, we adopt a recently developed operator-splitting method (see [28, 30, 39]) to solve equation (2.3) numerically. We focus on the case $d = 2$. Our method can be easily extended to higher dimensional problems.

We first reformulate equation (2.3) using the following identity:

$$\det(\mathbf{D}^2 u) = \frac{1}{2} \nabla \cdot (\mathrm{cof}(\mathbf{D}^2 u) \nabla u), \tag{3.1}$$

where $\mathrm{cof}(\mathbf{D}^2 u) = \begin{bmatrix} \frac{\partial^2 u}{\partial x_1^2} & -\frac{\partial^2 u}{\partial x_1 \partial x_2} \\ -\frac{\partial^2 u}{\partial x_1 \partial x_2} & \frac{\partial^2 u}{\partial x_2^2} \end{bmatrix}$ is the cofactor matrix of $\mathbf{D}^2 u$.

Incorporating equation (3.1) into equations (2.3) and (2.1) gives rise to

$$\begin{cases} u^0 = u_0, \\ \nabla \cdot (\mathrm{cof}(\mathbf{D}^2 u^{k+1}) \nabla u^{k+1}) = 2\, R(u^k) |u^k|^d & \text{in } \Omega, \\ u^{k+1} = 0 & \text{on } \partial\Omega, \end{cases} \tag{3.2}$$

with

$$R(u) = \frac{\displaystyle\int_\Omega (\mathrm{cof}(\mathbf{D}^2 u)\nabla u) \cdot \nabla u d\mathbf{x}}{\displaystyle 2\int_\Omega (-u)^3 d\mathbf{x}}, \tag{3.3}$$

where we used integration by parts when deriving equation (3.3).

From equation (3.2), updating u^{k+1} from u^k is equivalent to solving the optimization problem

$$\begin{cases} \min\limits_w \left[\displaystyle\int_\Omega (\mathrm{cof}(\mathbf{D}^2 w)\nabla w) \cdot \nabla w d\mathbf{x} + 6 \int_\Omega f^k w d\mathbf{x} \right], \\ w = 0 \text{ on } \partial\Omega, \end{cases} \tag{3.4}$$

with $f = R(u^k)|u^k|^2$, which can be derived from the first-order variational principle; see [30, 39]. Note that if $(\lambda_{\mathrm{MA}}, u^*)$ is a solution to equation (1.2), $(\lambda_{\mathrm{MA}}, \alpha u^*)$ is also a solution for any $\alpha > 0$ (assuming that we are looking for convex eigenfunctions). To make the solution of equation (1.2) unique, we restrict our attention to looking for the eigenfunction u^* satisfying $\|u^*\|_2 = 1$. Therefore it is natural to add the constraint $\|w\|_2 = 1$ to equation (3.4). However, usually a constrained optimization problem is more challenging to solve than an unconstrained one. Therefore, to remove the constraint while enforcing $\|w\|_2 = 1$, we utilize an indicator function.

Define the set

$$S = \{w : w \text{ is smooth}, \|w\|_2 = 1\}$$

and its indicator function

$$I_S(w) = \begin{cases} 0 & \text{if } w \in S, \\ +\infty & \text{otherwise.} \end{cases}$$

Equation (3.4) with constraint $\|w\|_2 = 1$ can be rewritten as

$$\begin{cases} \min\limits_w \left[\displaystyle\int_\Omega (\mathrm{cof}(\mathbf{D}^2 w)\nabla w) \cdot \nabla w d\mathbf{x} + 6 \int_\Omega f^k w d\mathbf{x} + I_S(w) \right], \\ w = 0 \text{ on } \partial\Omega. \end{cases} \tag{3.5}$$

We follow [30] to introduce a matrix-valued auxiliary variable $\mathbf{p}$ to decouple the nonlinearity in equation (3.5). Then solving equation (3.5) is equivalent to solving

$$\begin{cases} \min\limits_{w,\mathbf{p}} & \left[\displaystyle\int_\Omega (\mathrm{cof}(\mathbf{p})\nabla w) \cdot \nabla w d\mathbf{x} + 6 \int_\Omega f^k w d\mathbf{x} + I_S(w) \right], \\ w = 0 & \text{on } \partial\Omega, \\ \mathbf{p} = \mathbf{D}^2 w & \text{in } \Omega. \end{cases} \tag{3.6}$$

After computing the Euler-Lagrange equation, if $(v,\mathbf{p})$ is a solution to equation (3.6), we have

$$\begin{cases} \nabla\cdot(\mathrm{cof}(\mathbf{p})\nabla v) - 2f^k + \partial I_S(v) \ni 0 & \text{in } \Omega, \\ v = 0 & \text{on } \partial\Omega, \\ \mathbf{p} = \mathbf{D}^2 v, & \text{in } \Omega, \end{cases} \tag{3.7}$$

where ∂I_S denotes the sub-differential of I_S.

We associate equation (3.7) with the following initial value problem (in the flavor of gradient flow)

$$\begin{cases} \begin{cases} \frac{\partial v}{\partial t} + \nabla\cdot((\varepsilon\mathbf{I} + \mathrm{cof}(\mathbf{p}))\nabla v) - 2f^k + \partial I_S(v) \ni 0 & \text{in } \Omega\times(0,+\infty), \\ v = 0 & \text{on } \partial\Omega\times(0,+\infty), \end{cases} \\ \frac{\partial \mathbf{p}}{\partial t} + \gamma(\mathbf{p} - D^2 v) = \mathbf{0} \quad \text{in } \Omega\times(0,+\infty), \\ v(0) = v_0,\ \mathbf{p}(0) = \mathbf{p}_0, \end{cases} \tag{3.8}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{0}$ is the zero matrix, and $\varepsilon > 0$ is a small constant. The term $\varepsilon\mathbf{I}$ is a regularization term in order to handle the case that $\inf_{\mathbf{x}\in\Omega} f^k(\mathbf{x}) = 0$. Then u^{k+1} is the steady state of v.

In equation (3.8), γ controls the evolution speed of $\mathbf{p}$. A natural choice is to let $\mathbf{p}$ evolve with a similar speed as that of v, leading to

$$\gamma = \beta\lambda_0$$

with λ_0 being the smallest eigenvalue of $-\nabla^2$ and $\beta > 0$ being some constant.

4. An Operator Splitting Method to Solve Equation (3.8)

4.1. **The operator splitting strategy.** The structure of equation (3.8) is well–suited to be time-discretized by the operator splitting method. Among many possible discretization schemes, we choose the simplest Lie scheme.

Let $\tau > 0$ denote the time step and denote $t^n = n\tau$. We time-discretize equation (3.8) as follows:

Initialization:

$$v^0 = v_0,\ \mathbf{p}^0 = \mathbf{p}_0. \tag{4.1}$$

For $n > 0$, update $(v^n,\mathbf{p}^n) \to (v^{n+1/3},\mathbf{p}^{n+1/3}) \to (v^{n+2/3},\mathbf{p}^{n+2/3}) \to (v^{n+1},\mathbf{p}^{n+1})$ as:

Step 1: Solve

$$\begin{cases} \begin{cases} \frac{\partial v}{\partial t} + \nabla\cdot((\varepsilon\mathbf{I} + \mathrm{cof}(\mathbf{p}))\nabla v) - 2f^k = 0 & \text{in } \Omega\times(t^n,t^{n+1}), \\ v = 0 & \text{on } \partial\Omega\times(t^n,t^{n+1}), \end{cases} \\ \frac{\partial \mathbf{p}}{\partial t} = \mathbf{0} \quad \text{in } \Omega\times(t^n,t^{n+1}), \\ v(t^n) = v^n,\ \mathbf{p}(t^n) = \mathbf{p}^n, \end{cases} \tag{4.2}$$

and set $v^{n+1/3} = v(t^{n+1}),\ \mathbf{p}^{n+1/3} = \mathbf{p}(t^{n+1})$.

Step 2: Solve

$$\begin{cases} \begin{cases} \frac{\partial v}{\partial t} = 0 & \text{in } \Omega \times (t^n, t^{n+1}), \\ v = 0 & \text{on } \partial\Omega \times (t^n, t^{n+1}), \end{cases} \\ \frac{\partial \mathbf{p}}{\partial t} + \gamma(\mathbf{p} - D^2 v) = \mathbf{0} \quad \text{in } \Omega \times (t^n, t^{n+1}), \\ v(t^n) = v^{n+1/3},\ \mathbf{p}(t^n) = \mathbf{p}^{n+1/3}, \end{cases} \tag{4.3}$$

and set $v^{n+2/3} = v(t^{n+1}),\ \mathbf{p}^{n+2/3} = \mathbf{p}(t^{n+1})$.

Step 3: Solve

$$\begin{cases} \begin{cases} \frac{\partial v}{\partial t} + \partial I_S(v) \ni 0 & \text{in } \Omega \times (t^n, t^{n+1}), \\ v = 0 & \text{on } \partial\Omega \times (t^n, t^{n+1}), \end{cases} \\ \frac{\partial \mathbf{p}}{\partial t} = \mathbf{0} \quad \text{in } \Omega \times (t^n, t^{n+1}), \\ v(t^n) = v^{n+2/3},\ \mathbf{p}(t^n) = \mathbf{p}^{n+2/3}, \end{cases} \tag{4.4}$$

and set $v^{n+1} = v(t^{n+1}),\ \mathbf{p}^{n+1} = \mathbf{p}(t^{n+1})$.

The scheme (4.1)–(4.4) is only semi-constructive since one still needs to solve the subproblems in equations (4.2)–(4.4). For equation (4.3), we have the explicit solution for $\mathbf{p}^{n+2/3}$:

$$\mathbf{p}^{n+2/3} = e^{-\gamma\tau}\mathbf{p}^n + (1 - e^{-\gamma\tau})\mathbf{D}^2 v^{n+1/3}.$$

Since the solution of equation (1.2) is a convex function, the Hessian $\mathbf{D}^2 u$ is a semi-positive definite matrix. Since $\mathbf{p}$ is an auxiliary variable estimating $\mathbf{D}^2 v$, we project it onto the space of semi-positive definite symmetric matrices once $\mathbf{p}^{n+2/3}$ is computed. We denote the projection operator by P_+; see more details in Section 5.4.

For other subproblems, we adopt the one-step backward Euler scheme (the Markchuk-Yanenko type). Our updating formulas are summarized as follows:

$$\begin{cases} \frac{v^{n+1/3} - v^n}{\tau} + \nabla \cdot \left((\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}^n)) \nabla v^{n+1/3} \right) - 2f^k = 0 & \text{in } \Omega, \\ v^{n+1/3} = 0 & \text{on } \partial\Omega, \end{cases} \tag{4.5}$$

$$\mathbf{p}^{n+1} = P_+ \left(e^{-\gamma\tau}\mathbf{p}^n + (1 - e^{-\gamma\tau})\mathbf{D}^2 v^{n+1/3} \right), \tag{4.6}$$

$$\begin{cases} \frac{v^{n+1} - v^{n+1/3}}{\tau} + \partial I_S(v^{n+1}) \ni 0 & \text{in } \Omega, \\ v^{n+1} = 0 & \text{on } \partial\Omega. \end{cases} \tag{4.7}$$

Remark 4.1. Equation (3.8) is very similar to problem (36) in [28], except that in our current scheme the constraint is $\|u\|_2 = 1$ and that in [28] it is $\|u\|_3 = 1$. Despite similar formulations, the numerical treatments are very different. In equations (4.5)-(4.7), f^k and the indicator function ∂I_S are separately distributed into two sub-steps. Equation (4.7) simply results in a projection to the unit sphere; see Section 4.2 for details.

In [28], $\lambda\, d\, u|u|$ with d being the spatial dimension plays the role of f^k and the constraint plays the role of ∂I_S, and both terms are arranged in the same sub-step (problem (50b) in [28]):

$$\begin{cases} u^{n+2/3} - u^{n+1/3} = 3\tau\lambda^{n+1} u^{n+2/3} |u^{n+2/3}|, \\ \int_\Omega |u^{n+2/3}|^3 d\mathbf{x} = 1 \end{cases} \tag{4.8}$$

The constraint $\|u\|_3 = 1$ cannot be replaced by $\|u\|_2 = 1$ since equation (4.8) was considered as an optimality condition of a Lagrangian functional and $\tau\lambda^{n+1}$ is the Lagrange multiplier. As a result, $u^{n+2/3}$ solves

$$u^{n+2/3} \in \underset{v:\int_\Omega |v|^3 d\mathbf{x}=1}{\arg\min} \left[\frac{1}{2}\int_\Omega |v|^2 d\mathbf{x} - \int_\Omega u^{n+1/3} v d\mathbf{x} \right]. \tag{4.9}$$

Unlike (4.7), the solution to problem (4.9) does not have an explicit expression, so that an iterative method (such as sequential quadratic programming) was used in [28] to solve problem (4.9).

Remark 4.2. Compared to the algorithm (2.3) proposed in [1], our scheme has an additional term related to the constraint $\|u\|_2 = 1$, and such a constraint leads to the projection step (4.7) which helps stabilize our numerical algorithm.

Remark 4.3. Scheme (4.2)–(4.4) is an approximation of the gradient flow of the functional in (3.6). The convergence of scheme (4.2)–(4.4) is closely related to that of the gradient flow together with an approximation error. It has been shown that when there is only one variable and the operator in each step has sufficient regularity, the approximation error is of $O(\tau)$ (see [13] and [27, Chapter 6]). However, the terms in (3.6) are non–trivial and non–smooth, traditional analysis techniques are not applicable in this scenario. As the splitting error is closely related to the time step τ, we expect the approximation error reduces (and thus the convergence of scheme (4.2)–(4.4) follows that of the gradient flow) as τ goes to 0.

4.2. **On the solution to equation (4.7).** In the scheme above, problems (4.5) and (4.6) are easy to solve. In equation (4.7), v^{n+1} solves

$$\begin{cases} \min\limits_w \left[\frac{1}{2\tau}\int_\Omega \|w - v^{n+1/3}\|_2^2 d\mathbf{x} + I_S(w) \right], \\ w = 0 \text{ on } \partial\Omega. \end{cases} \tag{4.10}$$

Since $I_S(w)$ is the indicator function of S in which $\|w\|_2 = 1$, the exact solution of equation (4.10) reads as

$$v^{n+1} = \frac{v^{n+1/3}}{\|v^{n+1/3}\|_2}. \tag{4.11}$$

4.3. **On the initial condition.** We next discuss the initial condition u_0 in the outer iteration and $(v_0, \mathbf{p}_0)$ in the inner iteration. The convergence theorem for the scheme (2.3), Theorem 2.3, requires the initial condition to be convex and smooth. A simple choice is to set u_0 as the solution to

$$\begin{cases} \det \mathbf{D}^2 u_0 = 1 & \text{in } \Omega, \\ u_0 = 0 & \text{on } \partial\Omega. \end{cases} \tag{4.12}$$

However, solving equation (4.12) is not trivial. Since u_0 is only the initial condition and the iterates generated by the inverse iteration are eventually smooth as shown in [37], we do not need to solve equation (4.12) exactly. An operator splitting method is proposed in [30] to solve equation (4.12). To make the initialization simpler, we will choose u_0 as the initial condition

according to a strategy used in [30]. Specifically, u_0 is the solution to the Poisson problem

$$\begin{cases} \nabla^2 u_0 = 2\eta & \text{in } \Omega, \\ u_0 = 0 & \text{on } \partial\Omega, \end{cases} \tag{4.13}$$

where $\eta > 0$ is of $O(1)$.

For the initial condition $(v_0, \mathbf{p}_0)$ in the $k+1$-th outer iteration, we simply set

$$v_0 = u^k, \ \mathbf{p}_0 = \mathbf{D}^2 v_0. \tag{4.14}$$

Our algorithm is summarized in Algorithm 1.

Algorithm 1: An operator-splitting method for solving problem (1.2)

Input: Parameters $\gamma, \tau, \varepsilon, N$.
Initialization: Set $k = 0$. Initialize u^0 according to equation (4.13).
while not converge **do**
  Step 1. Compute $f^k = R(u^k)|u^k|^2$ according to equation (3.3).
  Step 2. Set $n = 0$. Initialize $(v^0, \mathbf{p}^0)$ according to equation (4.14).
  while not converge **do**
    Step 3.1. Solve equation (4.5) for $v^{n+1/3}$.
    Step 3.2. Solve equation (4.6) for $\mathbf{p}^{n+1}$.
    Step 3.3. Solve equation (4.7) for v^{n+1}.
    Step 3.4. Set $n = n+1$.
  end while
  Step 4. Set u^{k+1} as the converged v^*.
  Step 5. Set $k = k+1$.
end while
Output: The converged eigenfunction u^* and eigenvalue λ_{MA}.

5. A Finite Element Implementation of Scheme (4.5)-(4.7)

5.1. **Generalities.** Let $\Omega \subset \mathbb{R}^2$ be an open bounded convex polygonal domain (or it has been approximated by such a domain). Let $\mathscr{T}_h$ be a triangulation of Ω, where h denotes the length of the longest edge of triangles in $\mathscr{T}_h$. Define the following two piecewise linear function spaces

$$V_h = \{\phi \in C^0(\bar{\Omega}) : \phi_T \in \mathbf{P}_1 \text{ for } \forall T \in \mathscr{T}_h\},$$
$$V_{0h} = \{\phi \in V_h : \phi|_{\partial\Omega} = 0\},$$

where $\mathbf{P}_1$ is the space of polynomials of two variables with degree no larger than 1. Let $H^1(\Omega)$ be the Sobolev space of order 1 and $H_0^1(\Omega)$ be the collection of functions in $H^1(\Omega)$ with vanishing trace on $\partial\Omega$. Then V_h and V_{0h} are approximations of $H^1(\Omega)$ and $H_0^1(\Omega)$, respectively.

Denote the set of vertices of $\mathscr{T}_h$ by Σ_h. We further denote the interior vertices of $\mathscr{T}_h$ by $\Sigma_{0h} = \Sigma_h \backslash (\Sigma_h \cap \partial\Omega)$. We use N_h and N_{0h} to denote the cardinality of Σ_h and Σ_{0h}, respectively. We have

$$\dim V_h = N_h \quad \text{and} \quad \dim V_{0h} = N_{0h}.$$

We order the vertices of $\mathscr{T}_h$ so that $\Sigma_{0h} = \{Q_l\}_{l=1}^{N_{0h}}$, where Q_l's denote the vertices. For any $1 \le l \le N_h$, we use ω_l to denote the union of triangles in $\mathscr{T}_h$ that have Q_l as a common vertex.

Denote the area of ω_l by $|\omega_l|$. For each vertex Q_l, we define the hat function ϕ_l so that

$$\phi_l \in V_h,\ \phi_l(Q_l) = 1 \text{ and } \phi_l(Q_m) = 0 \text{ for } m \neq l.$$

We have that ϕ_l is supported on ω_l. For any function $f \in H^1(\Omega)$, its finite element approximation $f_h \in V_h$ can be written as

$$f_h = \sum_{l=1}^{N_h} f(Q_l)\phi_l.$$

We further equip V_h with the inner product $(f_h, g_h)_h : V_h \times V_h \to \mathbb{R}$ defined by

$$(f_h, g_h)_h = \frac{1}{3}\sum_{l=1}^{N_h} |\omega_l| f_h(Q_l) g_h(Q_l), \forall f_h, g_h \in V_h.$$

The induced norm is defined as

$$\|f_h\|_h = \sqrt{(f_h, f_h)}.$$

Because of the eventual smoothness of solutions to the inverse iteration (2.3) as shown in [37], our mixed finite-element method uses the space V_h to approximate both the solution u and its second-order partial derivatives $\partial^2 u/\partial x_i \partial x_j$ for $i, j = 1, 2$. In the rest of this section, we denote the finite-element approximation of v and $\mathbf{p}$ by $v_h \in V_{0h}$ and $\mathbf{p}_h \in (V_h)^{2\times 2}$, respectively.

5.2. **Finite element approximation of the three second-order partial derivatives.** In equation (4.6), one needs to compute $\mathbf{D}^2 v^{n+1/3}$, the Hessian of $v^{n+1/3}$, which will be numerically computed, and we adopt to our current setting the *double regularization* method introduced in [30].

The double regularization method is a two-step process to get a smooth approximation of $\mathbf{D}^2 u$. In the first step, one solves

$$\begin{cases} -\varepsilon_1 \nabla^2 \pi_{ij} + \pi_{ij} = \frac{\partial^2 u}{\partial x_i \partial x_j} & \text{in } \Omega, \\ \pi_{ij} = 0 & \text{on } \partial\Omega, \end{cases} \tag{5.1}$$

in which $\varepsilon_1 = O(h^2)$ is a constant, π_{ij} is a regularized approximation of $\partial^2 u/\partial x_i \partial x_j$ with zero boundary condition. Although π_{ij} is a smooth approximation, the zero boundary condition will have a disastrous influence to the solution u of our scheme, as mentioned in [30]. To mitigate the influence, the second step is a correction step which solves

$$\begin{cases} -\varepsilon_1 \nabla^2 D^2_{ij} u + D^2_{ij} u = \pi_{ij} & \text{in } \Omega, \\ \frac{\partial D^2_{ij} u}{\partial \mathbf{n}} = 0 & \text{on } \partial\Omega, \end{cases} \tag{5.2}$$

where $\mathbf{n}$ denotes the outward normal direction of $\partial\Omega$. The resulting $D^2_{ij} u$ is the doubly regularized approximation of $\partial^2 u/\partial x_i \partial x_j$.

From the divergence theorem, one has

$$\begin{cases} \forall i, j = 1, 2,\ \forall v \in H^2(\Omega), \\ \displaystyle\int_\Omega \frac{\partial^2 v}{\partial x_i \partial x_j} w d\mathbf{x} = -\frac{1}{2}\int_\Omega \left[\frac{\partial v}{\partial x_i}\frac{\partial w}{\partial x_j} + \frac{\partial v}{\partial x_j}\frac{\partial w}{\partial x_i}\right] d\mathbf{x}, \\ \forall w \in H^1_0(\Omega). \end{cases} \tag{5.3}$$

Based on equation (5.3), the discrete analogues of equations (5.1)-(5.2) read as:

$$\begin{cases} \pi_{ijh} \in V_{0h}, \\ c \sum\limits_{T \in \omega_l} |T| \int_T \nabla \pi_{ijh} \cdot \nabla \phi_l d\mathbf{x} + \dfrac{1}{3} |\omega_l| \pi_{ijh}(Q_l) = -\dfrac{1}{2} \int_{\omega_l} \left[\dfrac{\partial u_h}{\partial x_i} \dfrac{\partial \phi_l}{\partial x_j} + \dfrac{\partial u_h}{\partial x_j} \dfrac{\partial \phi_l}{\partial x_i} \right] d\mathbf{x}, \\ \forall l = 1, ..., N_{0h} \end{cases} \tag{5.4}$$

and

$$\begin{cases} D^2_{ijh} u_h \in V_h, \\ c \sum\limits_{T \in \omega_l} |T| \int_T \nabla D^2_{ijh} u_h \cdot \nabla \phi_l d\mathbf{x} + \dfrac{1}{3} |\omega_l| D^2_{ijh} u_h(Q_l) = \dfrac{1}{3} |\omega_l| \pi_{ijh}(Q_l), \\ \forall l = 1, ..., N_h, \end{cases} \tag{5.5}$$

where $c = O(1)$ is a constant.

5.3. **On the finite-element approximation of problem (4.5).** We first rewrite equation (4.5) in the variational form

$$\begin{cases} v^{n+1/3} \in V_{0h}, \\ \int_\Omega v^{n+1/3} \psi d\mathbf{x} + \tau \int_\Omega (\varepsilon \mathbf{I} + \mathrm{cof}(\mathbf{p}^n)) \nabla v^{n+1/3} \cdot \nabla \psi d\mathbf{x} = 2 \int_\Omega f^k \psi d\mathbf{x}, \\ \forall \psi \in V_{0h}. \end{cases} \tag{5.6}$$

If $\mathbf{p}^n$ is semi–positive definite, then problem (5.6) admits a unique solution. Denote $\mathbf{M} = \varepsilon \mathbf{I} + \mathrm{cof}(\mathbf{p}^n_h)$. The discrete analogue of equation (5.6) reads as

$$\begin{cases} v_h^{n+1/3} \in V_{0h}, \\ \frac{1}{3} |\omega_l| v_h^{n+1/3}(Q_l) + \tau \sum\limits_{m=1}^{N_{0h}} \left(v_h^{n+1/3}(Q_m) \int_{\omega_l \cap \omega_m} \mathbf{M} \nabla \phi_m \cdot \nabla \phi_l dx \right) = \frac{2}{3} |\omega_l| f^k(Q_l), \\ \forall l = 1, ..., N_{0h}. \end{cases} \tag{5.7}$$

Solving problem (5.7) is equivalent to solving a sparse linear system, for which many efficient solvers, such as the Cholesky decomposition, can be used.

5.4. **On the finite element approximation of problem (4.6).** We first define the projection operator P_+ that projects 2×2 real symmetric matrices to the set of real symmetric semi-positive definite matrices. Let $\mathbf{A}$ be a 2×2 real symmetric matrix. By spectral decomposition, there exists a 2×2 orthogonal matrix $\mathbf{S}$ so that $\mathbf{A} = \mathbf{S} \mathbf{\Lambda} \mathbf{S}^{-1}$, where

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

with λ_1, λ_2 being eigenvalues of $\mathbf{A}$. If $\mathbf{A}$ is semi–positive definite, one has $\lambda_1, \lambda_2 \geq 0$. Therefore we define P_+ as

$$P_+(\mathbf{A}) = \mathbf{S} \begin{bmatrix} \max(\lambda_1, 0) & 0 \\ 0 & \max(\lambda_2, 0) \end{bmatrix} \mathbf{S}^{-1}.$$

In equation (4.6), we compute

$$\mathbf{p}_h^{n+1} = P_+\left(e^{-\gamma\tau}\mathbf{p}_h^n + (1-e^{-\gamma\tau})\begin{bmatrix} D_{11h}^2 v_h^{n+1/3} & D_{12h}^2 v_h^{n+1/3} \\ D_{21h}^2 v_h^{n+1/3} & D_{22h}^2 v_h^{n+1/3}\end{bmatrix}\right),$$

where the entries $D_{ijh}^2 v_h^{n+1/3}$ are computed using equations (5.4)-(5.5).

5.5. **On the finite element approximation of problem (4.7).** According to equation (4.11), we compute v_h^{n+1} as

$$v_h^{n+1} = \frac{v_h^{n+1/3}}{\left(\sum\limits_{l=1}^{N_{0h}} \frac{1}{3}|\omega_l|\left(v_h^{n+1/3}(Q_l)\right)^2\right)^{1/2}}.$$

5.6. **On the finite element approximation of equation (3.3).** For any $u_h \in H_0^1(\Omega)$, the discrete analogue of equation (3.3) reads as

$$R(u_h) = -\frac{\sum\limits_{m,l=1}^{N_{0h}} u_h(Q_m)u_h(Q_l)\int_{\omega_l\cap\omega_m}(\mathrm{cof}(\mathbf{D}_h^2 u_h(Q_m))\nabla\phi_m)\cdot\nabla\phi_l d\mathbf{x}}{\sum\limits_{m=1}^{N_{0h}}\frac{2}{3}|\omega_m|(-u_h(Q_m))^3},$$

where $\mathbf{D}_h^2 u_h$ is the finite-element approximation of $\mathbf{D}^2 u$ computed using equations (5.4) and (5.5).

Note that if u is an eigenfunction of the Monge–Ampère equation (1.2), by Theorem 2.2, one can compute the eigenvalue as $\lambda_{\mathrm{MA}} = \inf_{u\in\mathscr{K}} R(u)$. Therefore, for every time step, we can compute the approximate 'eigenvalue' corresponding to u_h^k as

$$\lambda_h^k = R(u_h^k)$$

and monitor the evolution of λ_h^k, which will monotonically converge to λ_{MA} as shown in [37].

5.7. **On the finite element approximation of the initial condition.** Denote the finite element of u_0 and $(v_0, \mathbf{p}_0)$ by u_{0h} and $(v_{0h}, \mathbf{p}_{0h})$, respectively. The discrete analogue of the initial condition (4.13) reads as

$$\begin{cases} u_{0h} \in V_{0h}, \\ \sum\limits_{m=1}^{N_{0h}} u_{0h}(Q_m)\int_{\omega_l\cap\omega_m}\nabla\phi_m\cdot\nabla\phi_l d\mathbf{x} = -\dfrac{2}{3}\eta|\omega_l|, \\ \forall l = 1,\ldots,N_{0h}. \end{cases}$$

For $(v_{0h}, \mathbf{p}_{0h})$, we set

$$v_{0h} = u_h^k, \quad \mathbf{p}_{0h} = \mathbf{D}_h^2 v_{0h},$$

where $\mathbf{D}_h^2$ is the double regularization approximation using equations (5.4)-(5.5).

6. Numerical Experiments

We demonstrate the efficiency of scheme (4.5)-(4.7) by several numerical experiments. We set the stopping criterion as $\|u_h^{k+1} - u_h^k\|_h < \xi$ for some small $\xi > 0$. Without specification, in

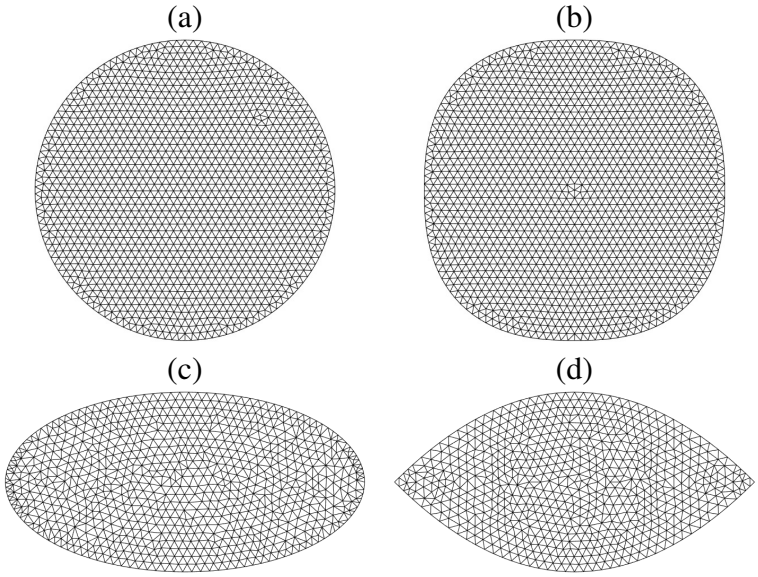

FIGURE 1. The triangulation of domains used in the examples. (a) The unit disk domain (6.1) with $h = 1/20$. (b) The smoothed square domain (6.3) with $h = 1/20$. (c) The ellipse domain (6.4) with $h = 1/20$. (d) The eye-shape domain (6.5) with $h = 1/40$.

all of our experiments, we set $\xi = 10^{-6}$, $\varepsilon = 2h^2$, and $c = 2$, where ε and c are regularization parameters in equation (3.8) and scheme (5.4)-(5.5), respectively.

When the exact solution, denoted by u_h^*, is given, we define the L^2 error and L^∞ error of u_h as

$$\|u_h - u_h^*\|_h \quad \text{and} \quad \max_m |u_h(Q_m) - u_h^*(Q_m)|,$$

respectively.

Algorithm 1 consists of two iterations: the outer iteration for u and the inner iteration for v and $\mathbf{p}$. Since both u and v are estimates of the solution of equation (1.2), it is not necessary to solve every inner iteration until steady state. Instead, one can just solve the inner iteration for a few steps. In our experiments, we observe that just 1 iteration step for the inner iteration is sufficient for our algorithm to converge. Thus in all of our experiments, we solve the inner iteration for only 1 step in each outer iteration.

6.1. **Example 1.** In the first example, we test our algorithm on the unit disk

$$\Omega = \{(x_1, x_2) : x_1^2 + x_2^2 < 1\}. \tag{6.1}$$

The triangulation of the domain with $h = 1/20$ is visualized in Figure 1(a).

In this case, equation (1.2) has a radial solution. Let $r = \sqrt{x_1^2 + x_2^2}$. For a radial function $g(r)$, one has $\det \mathbf{D}^2 g = \frac{g'g''}{r}$. Therefore, we write the solution to equation (1.2) as $u(r)$, which

satisfies

$$\begin{cases} u \le 0,\ \lambda > 0, \\ u'u'' = -\lambda\, r u^2 \ \text{ in } (0,1), \\ u'(0) = 0,\ u(1) = 0, \\ 2\pi \displaystyle\int_0^1 |u|^2 r\, dr = 1. \end{cases} \tag{6.2}$$

Using a shooting method, we can solve the ODE problem (6.2) very accurately. The *exact* solution verifies $u(\mathbf{0}) \approx -1.0628$ and $\lambda \approx 7.4897$. On the domain (6.1), we test our algorithm with $h = 1/20, 1/40, 1/80$ and $1/160$. In Figure 2(a)–(d), we show results with $h = 1/80$. Our numerical result is visualized in Figure 2(a). The contour of Figure 2(a) is shown in Figure 2(b). Our result is a smooth radial function, whose contour consists of several circles with the same center. The convergence histories of the error $\|u_h^{k+1} - u_h^k\|_h$ and the computed eigenvalue are shown in Figure 2(c) and Figure 2(d), respectively. Linear convergence is observed for the error $\|u_h^{k+1} - u_h^k\|_h$, and the convergence rate is approximately 0.47. The computed eigenvalue converges with just 5 iterations. In Figure 2(e), we show the cross sections of the results with various h along $x_2 = 0$. As h goes to 0, our computed solution converges to the exact solution. For better visualization, the zoomed bottom region of Figure 2(e) is shown in Figure 2(f).

To quantify the convergence of the proposed algorithm, we present in Table 1 the number of iterations needed for convergence, L^2- and L^∞-errors, computed eigenvalues and the minimal value of the computed solution with various h. For all resolutions of mesh, 13 iterations are sufficient for the algorithm to converge. As h goes to zero, the convergence rate of the L^2- and L^∞-error goes to 1, and the computed eigenvalue and the minimal value converge to the exact solutions. The eigenvalue λ_h converges linearly to the exact eigenvalue with an error of $O(h)$.

We next compare Algorithm 1 with the method proposed in [28]. For the method from [28], we have to use small time steps to make sure that the method does converge. In the numerical experiment, we set the time step as $h/2$ and stopping criterion as 10^{-6}. Note that the method from [28] finds the solution of equation (1.2) with $\|u_h\|_3 = 1$. When computing the L^2- and L^∞-errors, we first normalize the solution so that $\|u_h\|_2 = 1$ and we then compute the errors. The comparisons are shown in Table 2. For both L^2- and L^∞- errors, both algorithms have errors with similar magnitudes. We compare the computational efficiency between the two algorithms in Table 3. The number of iterations used by Algorithm 1 is independent of the mesh resolution, while the number of iterations used by [28] grows approximately linearly with $1/h$. For the CPU time, Algorithm (1) is also much faster than the method in [28]. Note that in Algorithm (1), the constraint $\|u_h\|_2 = 1$ is enforced by the projection step (4.7). In [28], the constraint is $\|u_h\|_3 = 1$, which was enforced by a sequential quadratic programming algorithm, which in turn uses around 15 iterations in each outer iteration.

6.2. **Example 2.** In the second example, we consider the convex smoothed square domain

$$\Omega = \left\{ (x_1, x_2) : |x_1|^{2.5} + |x_2|^{2.5} < 1 \right\}. \tag{6.3}$$

The triangulation of the domain with $h = 1/20$ is visualized in Figure 1(b), which has a shape between the unit disk and a square. We test our algorithm with h varying from $h = 1/20$ to $h = 1/160$. Similar to our settings in the previous example, we set the stopping criterion $\xi = 10^{-6}$. The time step is set as $\tau = 1/2$. Our results with $h = 1/80$ are visualized in Figure

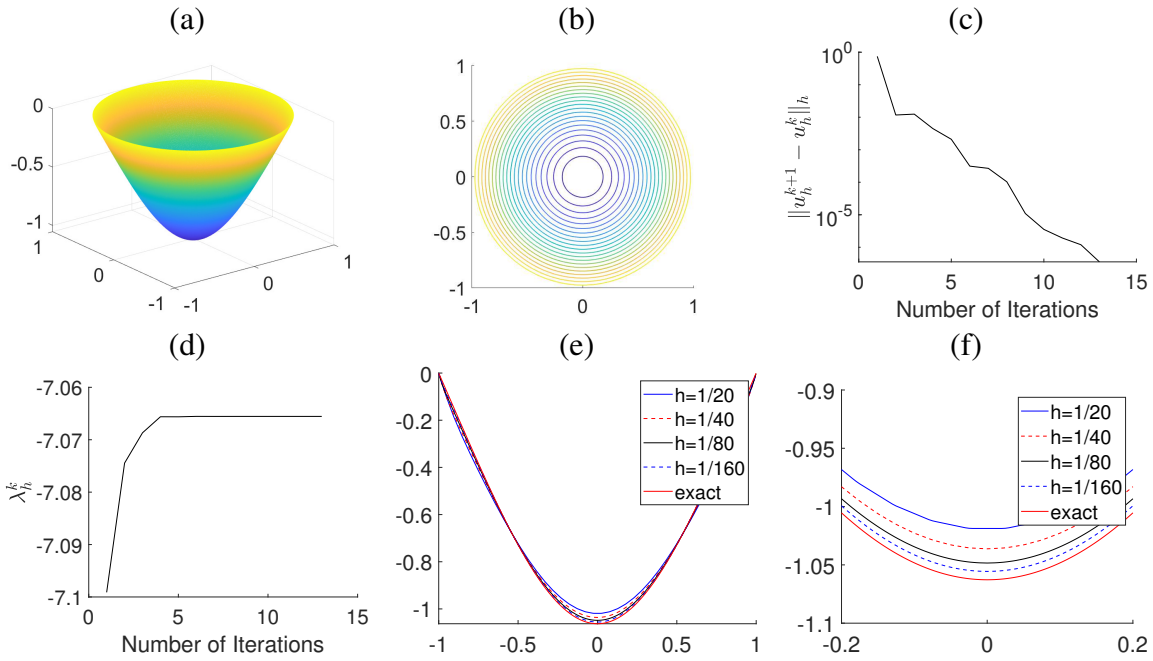

FIGURE 2. The unit disk domain (6.1). (a) The computed result with $h = 1/80$. (b) The contour of (a). (c) The history of the error $\|u_h^{k+1} - u_h^k\|_h$ with $h = 1/80$. (d) The history of the computed eigenvalue λ_h^k with $h = 1/80$. (e) Comparison of the cross sections along $x_2 = 0$ of the computed solution with various h. (f) Zoomed plot of the bottom region of (e).

TABLE 1. The unit disk domain (6.1). Variations with h of the number of iterations necessary to achieve convergence (2nd column), of the L^2 and L^∞ approximation errors and of the associated convergence rates (columns 4, 5, 6 and 7), of the computed eigenvalue (8th column) and of the minimal value of u_h over Ω (that is $u_h(\mathbf{0})$) (9th column). The exact eigenvalue is around 7.4897. The minimal value of the exact solution is around -1.0628.

h	# Iter.	$\|u_h^{k+1} - u_h^k\|_h$	L^2-error	rate	L^∞-error	rate	λ_h	$\min u_h$
1/20	13	2.13×10^{-7}	4.91×10^{-2}		4.29×10^{-2}		5.9716	-1.0189
1/40	13	2.91×10^{-7}	3.36×10^{-2}	0.54	3.04×10^{-2}	0.50	6.6656	-1.0362
1/80	13	3.56×10^{-7}	1.94×10^{-2}	0.79	1.86×10^{-2}	0.71	7.0655	-1.0484
1/160	13	4.04×10^{-7}	1.01×10^{-2}	0.94	1.03×10^{-2}	0.85	7.2816	-1.0556

3(a)–(d). Our computed solution is shown in Figure 3(a), whose contour is shown in Figure 3(b). Again, our solution is very smooth. The convergence histories of the error $\|u_h^{k+1} - u_h^k\|_h$ and the computed eigenvalues λ_h^k are shown in Figure 3(c) and Figure 3(d), respectively. The error $\|u_h^{k+1} - u_h^k\|_h$ converges linearly with a rate of 0.38. In this numerical experiment, the stopping criterion is satisfied after 16 iterations. The computed eigenvalue achieves its steady state with about 5 iterations. With various h, the comparison of cross sections of our results along $x_2 = 0$ is shown in Figure 3(e)–(f). As h goes to 0, the convergence of the solution along cross sections is observed.

TABLE 2. The unit disk domain (6.1). Variations with h of the number of iterations necessary to achieve convergence (2nd column), of the L^2 and L^∞ approximation errors and of the associated convergence rates (columns 4, 5, 6 and 7), of the computed eigenvalue (8th column) and of the minimal value of u_h over Ω (that is $u_h(\mathbf{0})$) (9th column). The exact eigenvalue is around 7.4897. The minimal value of the exact solution is around -1.0628.

	Algorithm (1)				Method from [28]			
h	L^2-error	rate	L^∞-error	rate	L^2-error	rate	L^∞-error	rate
1/20	4.91×10^{-2}		4.29×10^{-2}		4.01×10^{-2}		8.40×10^{-2}	
1/40	3.36×10^{-2}	0.54	3.04×10^{-2}	0.50	2.33×10^{-2}	0.78	4.00×10^{-2}	1.07
1/80	1.94×10^{-2}	0.79	1.86×10^{-2}	0.71	1.37×10^{-2}	0.76	2.05×10^{-2}	0.96
1/160	1.01×10^{-2}	0.94	1.03×10^{-2}	0.85	7.55×10^{-3}	0.86	1.08×10^{-2}	0.92

TABLE 3. The unit disk domain (6.1). Comparison of the number of iterations and the CPU time needed by Algorithm 1 and the method in [28] for convergence.

	Algorithm (1)		Method from [28]	
h	# Iter.	CPU time	# Iter.	CPU time
1/20	13	1.44	62	3.55
1/40	13	4.58	101	22.39
1/80	13	18.35	151	138.47
1/160	13	83.95	263	1206.96

TABLE 4. The smoothed square domain (6.3). Variations with h of the number of iterations necessary to achieve convergence (2nd column), of the computed eigenvalue (4th column) and of the minimal value of u_h over Ω (that is $u_h(\mathbf{0})$) (5th column).

h	# Iter.	$\|u_h^{k+1}-u_h^k\|_h$	λ_h	$\min u_h$
1/20	14	6.05×10^{-7}	5.17	-0.9833
1/40	14	8.00×10^{-7}	5.72	-0.9982
1/80	16	2.08×10^{-7}	6.05	-1.0094
1/160	18	7.77×10^{-7}	6.22	-1.0159

We then report the computational cost and convergence behavior of the computed eigenvalue and minimal value with various h in Table 4. The convergence of the eigenvalue is similar to that in [28]: the eigenvalue λ_h converges to λ uniformly in the rate $\lambda_h \approx \lambda - ch$ with $\lambda \approx 6.4, c \approx 26$. In terms of the computational cost, Algorithm 1 is very efficient since all experiments used less than 20 iterations to satisfy the stopping criterion.

6.3. **Example 3.** In the third example, we consider an ellipse domain defined by

$$\Omega = \left\{(x_1,x_2) : x_1^2 + 2x_2^2 < 1\right\}. \tag{6.4}$$

A triangulation of the domain with $h = 1/20$ is visualized in Figure 1(c). In this set of experiments, we set stopping criterion $\xi = 10^{-6}$ and time step $\tau = 1/2$. The results with $h = 1/80$ are

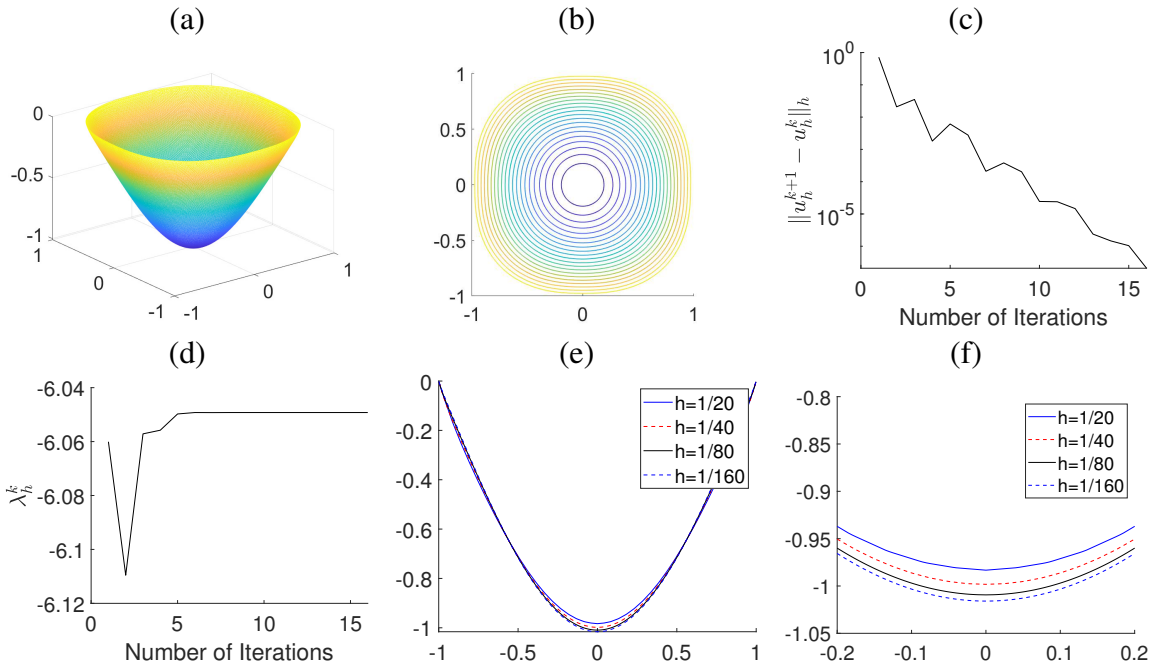

FIGURE 3. The smoothed square domain (6.3). (a) The computed result with $h = 1/80$. (b) The contour of (a). (c) The convergence history of the errors $\|u_h^{k+1} - u_h^k\|_h$. (d) The history of the computed eigenvalue λ_h^k. (e) Comparison of the cross sections along $x_2 = 0$ of the computed solution with various h. (f) Zoomed plot of the bottom region of (e).

TABLE 5. The ellipse domain (6.4). Variations with h of the number of iterations necessary to achieve convergence (2nd column), of the computed eigenvalue (4th column) and of the minimal value of u_h over Ω (that is $u_h(\mathbf{0})$) (5th column).

h	# Iter.	$\|u_h^{k+1} - u_h^k\|_h$	λ_h	$\min u_h$
1/20	16	6.80×10^{-7}	21.55	-1.4277
1/40	16	9.68×10^{-7}	25.18	-1.4525
1/80	17	6.44×10^{-7}	27.41	-1.4734
1/160	17	7.00×10^{-7}	28.67	-1.4875

shown in Figure 4(a)–(d). Similar to the results in the previous examples, the computed solution is smooth, and its contour consists of several ellipses with the same center, as shown in Figure 4(a) and Figure 4(b), respectively. In Figure 4(c), linear convergence is observed for the error $\|u_h^{k+1} - u_h^k\|_h$, and the convergence rate is about 0.34. The computed eigenvalue λ_h^k attains its steady state with 6 iterations. With various h, we compare in Figure 4(e)–(f) the cross sections of the computed results along $x_2 = 0$. Convergence is observed as h goes to 0.

With various h, the computational cost, the computed eigenvalue and minimal value of the computed solution are presented in Table 5. The eigenvalue λ_h converges to λ uniformly in the rate $\lambda_h \approx \lambda - ch$ with $\lambda \approx 29.5, c \approx 161$. In terms of the computational cost, all experiments used less than 20 iterations to satisfy the stopping criterion.

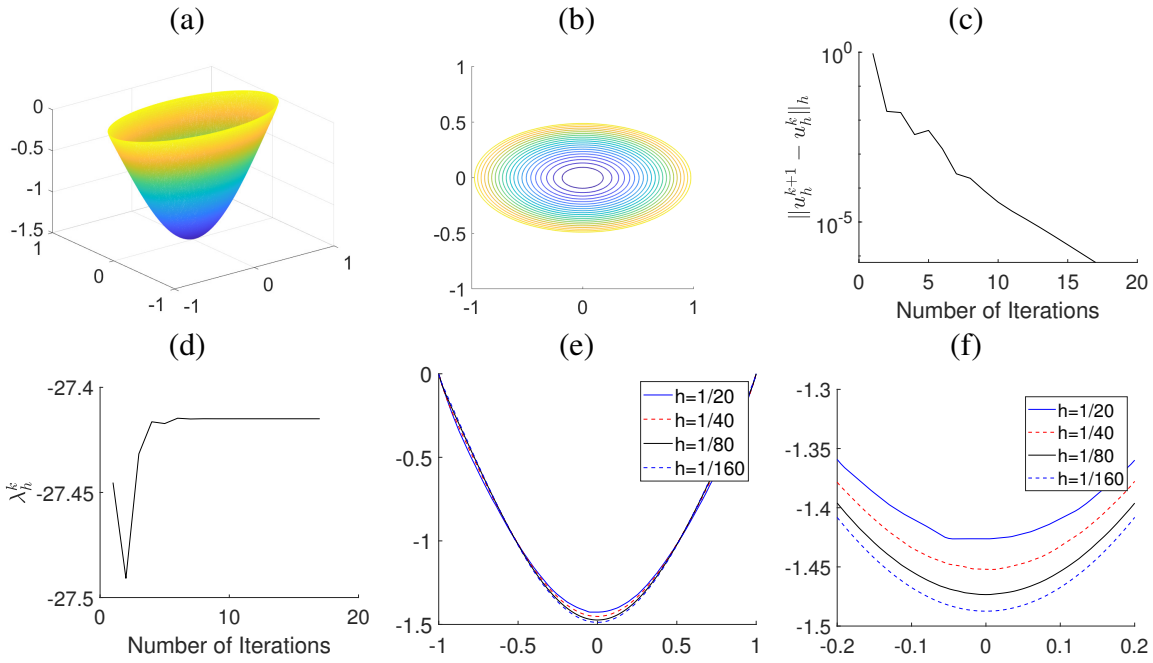

FIGURE 4. The ellipse domain (6.4). (a) The computed result with $h = 1/80$. (b) The contour of (a). (c) The history of the error $\|u_h^{k+1} - u_h^k\|_h$. (d) The history of the computed eigenvalue λ_h^k. (e) Comparison of the cross sections along $x_2 = 0$ of the computed solution with various h. (f) Zoomed plot of the bottom region of (e).

6.4. **Example 4.** We conclude this section by considering an open convex domain with a non-smooth boundary:

$$\Omega = \{(x_1, x_2) : -x_1(1-x_1) < x_2 < x_1(1-x_1),\ 0 < x_1 < 1\}. \tag{6.5}$$

The domain described in the set (6.5) has an eye shape, and its triangulation with $h = 1/40$ is visualized in Figure 1(d). Since the domain is not smooth, in our experiments we use a smaller time step $\tau = 1/8$ and larger regularization parameters $\varepsilon = 4h^2$ and $c = 4$. We set stopping criterion $\xi = 10^{-6}$. The results with $h = 1/160$ are shown in Figure 5(a)–(d). The computed solution is smooth, and its level curves have the same center, as shown in Figure 5(a) and Figure 5(b), respectively. In Figure 5(c), linear convergence is observed for the error $\|u_h^{k+1} - u_h^k\|_h$. The computed eigenvalue λ_h^k attains its steady state with 7 iterations. With various h, we compare in Figure 5(e)–(f) the cross sections of the computed results along $x_2 = 0$. Convergence is observed as h goes to 0.

With various h, the computational cost, the computed eigenvalue, and the minimal value of the computed solution are presented in Table 6. The eigenvalue λ_h converges to λ uniformly in the rate $\lambda_h \approx \lambda - ch$ with $\lambda \approx 618, c \approx 7792.3$. In terms of the computational cost, all experiments used no more than 30 iterations to satisfy the stopping criterion.

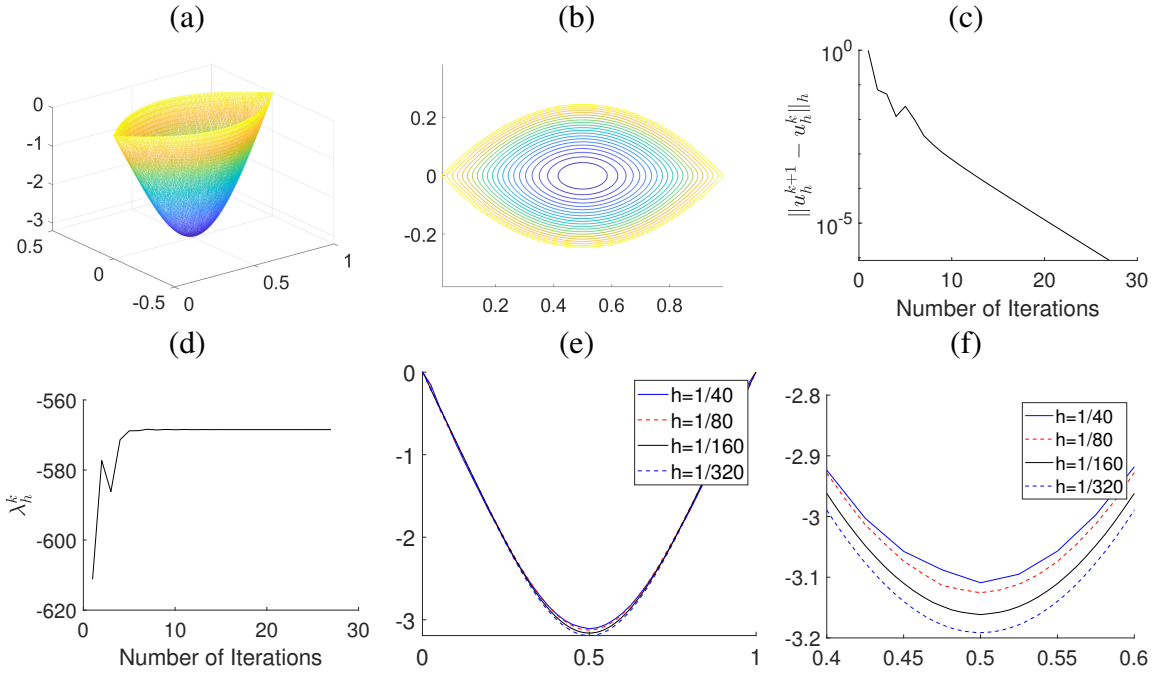

FIGURE 5. The eye–shape domain (6.5). (a) The computed result with $h = 1/160$. (b) The contour of (a). (c)The history of the error $\|u_h^{k+1} - u_h^k\|_h$. (d) The history of the computed eigenvalue λ_h^k. (e) Comparison of the cross sections along $x_2 = 0$ of the computed solution with various h. (f) Zoomed plot of the bottom region of (e).

TABLE 6. The eye–shape domain (6.5). Variations with h of the number of iterations necessary to achieve convergence (2nd column), of the computed eigenvalue (4th column) and of the minimal value of u_h over Ω (that is $u_h(\mathbf{0})$) (5th column).

h	# Iter.	$\|u_h^{k+1} - u_h^k\|_h$	λ_h	$\min u_h$
1/40	15	7.80×10^{-7}	425.51	-3.1091
1/80	20	7.53×10^{-7}	516.57	-3.1256
1/160	27	8.35×10^{-7}	568.47	-3.1617
1/320	30	7.87×10^{-7}	597.39	-3.1913

7. CONCLUSION

We proposed an efficient operator–splitting method to solve the eigenvalue problem of the Monge–Ampère equation. The backbone of our method relies on a convergent algorithm proposed in [1]. In each iteration, we solve a constrained optimization problem whose optimality condition is of the Monge–Ampère type. We remove the constraint by including an indicator function and decouple the nonlinearity by introducing an auxiliary variable. The resulting problem is then converted to finding the steady state solution of an initial value problem which is time discretized by an operator–splitting method. The efficiency and effectiveness of the proposed method is demonstrated with several numerical experiments. In our experiments, we can

choose a large constant time step. On smooth convex domains, our algorithm converges with a few iterations and is much faster than existing methods.

Acknowledgements
Dr. Jun Kitagawa is acknowledged for bringing the Abedin-Kitagawa paper to the third author's attention in January 2021. Subsequently, Prof. Roland Glowinski and the authors initiated the project to develop an efficient algorithm to implement the Abedin-Kitagawa formulation.

The preparation of this manuscript has been overshadowed by Roland's passing away in January 2022. Roland and the authors had intended to write jointly: most of the main ideas were worked out together and the authors have done their best to complete them. In sorrow, the authors dedicate this work to his memory. Roland's creativity, generosity, and friendship will be remembered.

H. Liu was partially supported by HKBU under grants 179356 and 162784. S. Leung was supported by the Hong Kong RGC under grant 16302819. J. Qian is partially supported by NSF grants.

REFERENCES

[1] F. Abedin, J. Kitagawa, Inverse iteration for the Monge–Ampère eigenvalue problem, Proc. Amer. Math. Soc. 14 8(2020) 4875–4886.
[2] G. Awanou, Standard finite elements for the numerical resolution of the elliptic Monge–Ampère equation: classical solutions, IMA J. Numer. Anal. 35 (2015) 1150–1166.
[3] I. J. Bakelman, Convex Analysis and Nonlinear Geometric Elliptic Equations, Springer Science & Business Media, 2012.
[4] J.-D. Benamou, Y. Brenier, A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem, Numer. Math. 84 (2000) 375–393.
[5] 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) 737–758.
[6] J.-D. Benamou, B. D. Froese, A. M. Oberman, Numerical solution of the optimal transportation problem using the Monge–Ampère equation, J. Comput. Phys. 260 (2014) 107–126.
[7] A. Bonito, A. Caboussat, M. Picasso, Operator splitting algorithms for free surface flows: Application to extrusion processes, In Splitting Methods in Communication, Imaging, Science, and Engineering, pages 677–729, Springer, 2016.
[8] M. Bukač, S. Čanić, R. Glowinski, J. Tambača, A. Quaini, Fluid–structure interaction in blood flow capturing non-zero longitudinal structure displacement, J. Comput. Phys. 235 (2013) 515–541.
[9] A. Caboussat, R. Glowinski, D. Gourzoulidis, A least-squares/relaxation method for the numerical solution of the three-dimensional elliptic Monge–Ampère equation, J. Sci. Comput. 77 (2018) 53–78.
[10] A. Caboussat, R. Glowinski, D. C. Sorensen. A least-squares method for the numerical solution of the Dirichlet problem for the elliptic Monge–Ampère equation in dimension two, ESAIM: Contr, Optim. Cal. Var. 19 (2013) 780–810.
[11] A. Caboussat, D. Gourzoulidis, A second order time integration method for the approximation of a parabolic 2d Monge-Ampère equation. In Numerical Mathematics and Advanced Applications ENUMATH 2019, pages 225–234, Springer, 2021.
[12] A. Caboussat, D. Gourzoulidis, M. Picasso, An adaptive method for the numerical solution of a 2d Monge-Ampère equation. In Proceedings of the 10th International Conference on Adaptive Modeling and Simulation (ADMOS 2021), 2021.
[13] A. J. Chorin, T. J. Hughes, M. F. McCracken, J. E. Marsden, Product formulas and numerical algorithms, Commun. Pure Appl. Math. 31 (1978) 205–256.

[14] E. J. Dean, R. Glowinski, Numerical solution of the two-dimensional elliptic Monge–Ampère equation with dirichlet boundary conditions: an augmented Lagrangian approach, Comptes Rendus Math. 336 (2003) 779–784.
[15] E. J. Dean, R. Glowinski, Numerical solution of the two-dimensional elliptic Monge–Ampère equation with dirichlet boundary conditions: a least-squares approach, Comptes Rendus Math. 339 (2004) 887–892.
[16] L.-J. Deng, R. Glowinski, X.-C. Tai, A new operator splitting method for the Euler elastica model for image smoothing, SIAM J. Imaging Sci. 12 (2019) 1190–1230.
[17] B. Engquist, B. D. Froese, Y. Yang, Optimal transport for seismic full waveform inversion, arXiv preprint arXiv:1602.01540, 2016.
[18] X. Feng, R. Glowinski, M. Neilan, Recent developments in numerical methods for fully nonlinear second order partial differential equations, SIAM Rev. 55 (2013) 205–267.
[19] X. Feng, T. Lewis, K. Ward, A narrow-stencil framework for convergent numerical approximations of fully nonlinear second order PDEs, arXiv preprint arXiv:2202.12782, 2022.
[20] 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.
[21] X. Feng, M. Neilan, A modified characteristic finite element method for a fully nonlinear formulation of the semigeostrophic flow equations, SIAM J. Numer. Anal. 47 (2009) 2952–2981.
[22] 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.
[23] B. D. Froese, A numerical method for the elliptic Monge–Ampère equation with transport boundary conditions, SIAM J. Sci. Comput. 34 (2012) A1432–A1459.
[24] B. D. Froese, A. M. Oberman, Convergent finite difference solvers for viscosity solutions of the elliptic Monge–Ampère equation in dimensions two and higher, SIAM J. Numer. Anal. 49 (2011) 1692–1714.
[25] B. D. Froese, A. M. Oberman, Fast finite difference solvers for singular solutions of the elliptic Monge-Ampère equation, J. Comput. Phys. 230 (2011) 818–834.
[26] D. Gilbarg, N. S. Trudinger, Elliptic Partial Differential Equations of Second Order, volume 224, Springer, 1977.
[27] R. Glowinski, Finite Element Methods for Incompressible Viscous Flow, Handbook of Numerical Analysis, 9 (2003), 3–1176.
[28] R. Glowinski, S. Leung, H. Liu, J. Qian, On the numerical solution of nonlinear eigenvalue problems for the Monge–Ampère operator, ESAIM: Contr. Optim. Cal. Var. 26 (2020) 118.
[29] R. Glowinski, S. Leung, J. Qian, A penalization-regularization-operator splitting method for eikonal based traveltime tomography, SIAM J. Imaging Sci. 8 (2015) 1263–1292.
[30] R. Glowinski, H. Liu, S. Leung, J. Qian, A finite element/operator-splitting method for the numerical solution of the two dimensional elliptic Monge–Ampère equation, J. Sci. Comput. 79 (2019) 1–47.
[31] R. Glowinski, S. J. Osher, W. Yin, Splitting Methods in Communication, Imaging, Science, and Engineering, Springer, 2017.
[32] R. Glowinski, T.-W. Pan, X.-C. Tai, Some facts about operator-splitting and alternating direction methods, In Splitting Methods in Communication, Imaging, Science, and Engineering, pages 19–94, Springer, 2016.
[33] S. Haker, L. Zhu, A. Tannenbaum, S. Angenent, Optimal mass transport for registration and warping, Int. J. Comput. Vision 60 (2004) 225–240.
[34] Y. He, S. H. Kang, H. Liu, Curvature regularized surface reconstruction from point clouds, SIAM J. Imaging Sci. 13 (2020) 1834–1859.
[35] J. L. Kazdan, Prescribing the Curvature of a Riemannian Manifold, volume 57. American Mathematical Soc. 1985.
[36] N. Q. Le, The eigenvalue problem for the Monge–Ampère operator on general bounded convex domains, arXiv preprint arXiv:1701.05165, 2017.
[37] N. Q. Le, Convergence of an iterative scheme for the Monge–Ampère eigenvalue problem with general initial data, arXiv preprint arXiv:2006.06564, 2020.
[38] P.-L. Lions, Two remarks on Monge–Ampère equations, Annali di Matematica Pura ed Applicata, 142 (1985) 263–275.

[39] H. Liu, R. Glowinski, S. Leung, J. Qian, A finite element/operator-splitting method for the numerical solution of the three dimensional Monge–Ampère equation, J. Sci. Computing 81 (2019) 2271–2302.
[40] H. Liu, X.-C. Tai, R. Glowinski, An operator-splitting method for the Gaussian curvature regularization model with applications in surface smoothing and imaging, arXiv preprint arXiv:2108.01914, 2021.
[41] H. Liu, X.-C. Tai, R. Kimmel, R. Glowinski, A color elastica model for vector-valued image regularization, SIAM J. Imaging Sci. 14 (2021) 717–748.
[42] H. Liu, X.-C. Tai, R. Kimmel, R. Glowinski, Elastica models for color image regularization, arXiv preprint arXiv:2203.09995, 2022.
[43] H. Liu, D. Wang, Fast operator splitting methods for obstacle problems, arXiv preprint arXiv:2203.08380, 2022.
[44] L. A. Roberts, L. A. Caffarelli, X. Cabré, Fully Nonlinear Elliptic Equations, volume 43, American Mathematical Soc. 1995.
[45] S. Stojanovic, Optimal momentum hedging via Monge–Ampère PDEs and a new paradigm for pricing options, SIAM J. Contr. Optim. 43 (2004) 1151–1173.
[46] K. Tso, On a real Monge–Ampère functional, Inventiones Math. 101 (1990) 425–448.