SIAM J. SCI. COMPUT.
Vol. 46, No. 5, pp. A3230–A3257

OPERATOR-SPLITTING/FINITE ELEMENT METHODS FOR THE MINKOWSKI PROBLEM*

HAO LIU†, SHINGYU LEUNG‡, AND JIANLIANG QIAN§

Abstract. The classical Minkowski problem for convex bodies has deeply influenced the development of differential geometry. During the past several decades, abundant mathematical theories have been developed for studying the solutions of the Minkowski problem; however, the numerical solution of this problem has been largely left behind, with only a few methods available to achieve that goal. In this article, focusing on the two-dimensional Minkowski problem with Dirichlet boundary conditions, we introduce two solution methods, both based on operator-splitting. One of these two methods deals directly with the Dirichlet condition, while the other one uses an approximation à la Robin of this Dirichlet condition. The relaxation of the Dirichlet condition makes the second method better suited than the first one to treat those situations where the Minkowski equation (of Monge–Ampère type) and the Dirichlet condition are not compatible. Both methods are generalizations of the solution method for the canonical Monge–Ampère equation discussed by Glowinski et al. [*J. Sci. Comput.,* 81 (2019), pp. 2271–2302]; as such they take advantage of a divergence formulation of the Minkowski problem, which makes it well suited to both a mixed finite-element approximation and the time-discretization via an operator-splitting scheme of an associated initial value problem. Our methodology can be easily implemented on convex domains of rather general shape (with curved boundaries, possibly). The numerical experiments validate both methods, showing that if one uses continuous piecewise affine finite-element approximations of the solution of the Minkowski problem and of its three second order derivatives, these two methods provide nearly second-order accuracy for the L^2 and L^∞ norms of the approximation error, where the Minkowski–Dirichlet problem is assumed to have a smooth solution. One can easily extend the methods discussed in this article to address the solution of three-dimensional Minkowski problems.

Key words. operator-splitting methods, Minkowski problem, Monge–Ampère equation, mixed finite element methods

MSC codes. 65N30, 65M60

DOI. 10.1137/23M1590779

1. Introduction.

The Minkowski problem (named after Hermann Minkowski (1864–1909)) is an important problem in differential geometry. It asks for the construction of a compact surface S as boundary of a convex bounded domain, knowing its Gaussian curvature. Given a compact strictly convex hypersurface S in the d–dimensional real space $\mathbb{R}^d$, the Gauss map $\mathbf{G}$ is a diffeomorphism from S to the unit sphere $\mathbf{S}^{d-1}$ of $\mathbb{R}^d$. Map $\mathbf{G}$ is defined by $\mathbf{G}(\mathbf{x}) = \mathbf{n}(\mathbf{x})\, \forall \mathbf{x} \in S$, where $\mathbf{n}(\mathbf{x})$ denotes the unit outward normal of S at $\mathbf{x}$. Accordingly, the Gauss–Kronecker curvature K is the Jacobian of the Gauss map. Minkowski stated that one has

*Submitted to the journal's Numerical Algorithms for Scientific Computing section July 31, 2023; accepted for publication (in revised form) May 28, 2024; published electronically October 8, 2024.
https://doi.org/10.1137/23M1590779
Funding: The first author was partially supported by National Natural Science Foundation of China grants 12201530, HKRGC ECS 22302123, and HKBU 179356. The second author was partially supported by the Hong Kong RGC under grant 16302223. The third author was partially funded by NSF 2012046, 2152011, and 2309534.
†Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (haoliu@hkbu.edu.hk).
‡Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (masyleung@ust.hk).
§Department of Mathematics and Department of CMSE, Michigan State University, East Lansing, MI 48824 USA (jqian@msu.edu).

$$\int_{\mathbf{S}^{n-1}} \mathbf{x}(K(\mathbf{G}^{-1}(\mathbf{x})))^{-1} d\sigma(\mathbf{x}) = \mathbf{0}, \tag{1.1}$$

where σ is the Lebesgue measure on $\mathbf{S}^{d-1}$. Conversely, Minkowski posed the following (inverse) problem: Suppose that f is a strictly positive function defined over $\mathbf{S}^{d-1}$ verifying $\int_{\mathbf{S}^{n-1}} \mathbf{x} f(\mathbf{x}) d\sigma(\mathbf{x}) = \mathbf{0}$; can one find a hypersurface S having $1/f$ as Gaussian curvature? In [41, 42], Minkowski discussed the existence and uniqueness of solutions to the above inverse problem. For $d = 2$, the solution regularity was proved by Lewy [33, 34], Nirenberg [44], and Pogorelov [45], while, for $d > 2$, the solution regularity was analyzed by Cheng and Yau [11] and Pogorelov [46].

Despite being around for more than a century and being one of the most important problems in differential geometry, not much was done concerning the numerical solution of the Minkowski problem. The earliest attempt we could find was discussed in [36, 37], two publications dedicated to the solution of a related problem—namely, reconstructing a shape from extended Gaussian images. In [31], after generalizing Minkowski's proof, Lamberg converted the Minkowski problem into an optimization one, the resulting algorithm solving a polyhedral version of the Minkowski problem. In [32], Lamberg introduced an algorithm based on Minkowski's isoperimetric inequality, leading to an approximate Minkowski problem taking place in a finite-dimensional function space spanned by truncated spherical harmonic series. In a more recent publication [10], Cheng designed a level-set based finite-difference PDE method to drive an implicitly defined surface towards shapes arising from the Minkowski problem.

In all the above-cited works the hypersurface is supposed to be closed. Actually, another type of Minkowski problem is the Minkowski–Dirichlet problem. For the Minkowski–Dirichlet problem, one supposes that the hypersurface is open and bounded, with a Dirichlet condition imposed on its boundary. The well-posedness of this problem has been addressed by many authors: For example, Bakelman [3], Lions [35], and Urbas [50, 51, 52] have proved the existence and uniqueness of a solution. Trudinger and Urbas [48] proved a necessary and sufficient condition for the classical solvability of the Minkowski–Dirichlet problem. Recently, in [30] Hamfelt designed a monotone finite-difference method to solve the Minkowski–Dirichlet problem; since the method relies on wide stencils, it is advantageous for those situations where, due to the lack of classical solutions, one looks for viscosity solutions.

Here, we propose two new methods for the numerical solution of the Minkowski–Dirichlet problem in dimension $d = 2$. The first method, well suited to problems with classical solutions, imposes the Dirichlet condition in a strong sense. On the other hand, the second method imposes the Dirichlet condition in a least-squares sense (via a quadratic penalty technique), making it appropriate for those situations where, due to data incompatibility, the Minkowski–Dirichlet problem has no solution. Of course, the second method has also the ability to capture classical solutions, if such solutions do exist. The Minkowski problem we will look at can be described as follows: Let Ω be a bounded domain of $\mathbb{R}^d$ and K be a positive function defined over Ω, and let g be a function defined on the boundary $\partial\Omega$; can one find a function u defined over Ω and verifying $u|_{\partial\Omega} = g$, such that K is the Gauss curvature of the graph of u (a surface in $\mathbb{R}^{d+1}$)? In partial differential equation form, the above Minkowski–Dirichlet problem reads as follows:

$$\begin{cases} \dfrac{\det(\mathbf{D}^2 u)}{(1 + |\nabla u|^2)^{1+d/2}} = K & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega. \end{cases} \tag{1.2}$$

The partial differential equation in (1.2) belongs to a family of Monge–Ampère equations. The simplest element of this family is clearly the following canonical Monge–Ampère equation:

$$\det(\mathbf{D}^2 u) = f \text{ in } \Omega. \tag{1.3}$$

Equation (1.3) is elliptic if $f > 0$. The above Monge–Amperère equation (1.3) is a fully nonlinear second order partial differential equation; it has been drawing a lot of attention lately, mostly because of its relations with optimal transport problems (other applications are described in, e.g., [19]; see also the references therein). During the past three decades, a variety of methods have been designed to solve numerically equation (1.3), completed by boundary conditions (mostly Dirichlet's) (some of these methods are described in the review article [19]). As expected, most of these methods focus on the two–dimensional Monge–Ampère equation and cover a large variety of approaches. Combinations of (mixed) finite-element approximations and augmented Lagrangian or least-squares formulations have been applied to the solution of (1.3) and related fully nonlinear elliptic equations such as Pucci's (see [4, 8, 13, 15, 14, 17, 18, 16, 26, 29, 25, 43, 9, 19] for details). Alternative finite-difference and finite-element methods have been developed for these fully nonlinear elliptic equations as well; see [1, 5, 6, 7, 21, 20, 40, 47, 28, 38, 19]; this list is far from complete.

The main goal of this article is to extend to problem (1.2) (assuming $d = 2$), the operator-splitting based methods developed in [28, 38] for the solution of (1.3) (completed by Dirichlet conditions) in dimensions 2 and 3 and in [27, 39] for the eigenvalue problems of (1.3). Following [28, 38], the first step in that direction is to take advantage of a divergence formulation of problem (1.2), which is better suited to finite-element approximations. The second step is to decouple (in some sense) differential operators and nonlinearities by introducing as additional unknown functions $\mathbf{p} = \mathbf{D}^2 u$ (as done in [27, 38]) and $\mathbf{s} = \nabla u$ (which was not necessary in [27, 38]). At the end of the second step, one has replaced the highly nonlinear scalar Minkowski equation by an equivalent system of linear and nonlinear equations for $u, \mathbf{p}$, and $\mathbf{s}$, whose formalism is simpler. In the third step, we associate an initial value problem with the above system and use operator-splitting to time-discretize the above initial value problem, in order to capture its steady state solution(s). We use simple finite-element approximations of mixed type to implement the above methodology: indeed, we use finite-element spaces of continuous piecewise affine functions to approximate u and its three second-order derivatives, making our methods well suited to solve problem (1.2) on domains Ω with curved boundaries.

As mentioned above we will develop two new methods for the solution of problem (1.2). These two methods are very close to each other; the first one deals directly with the boundary condition $u = g$ on $\partial\Omega$, while the second one imposes the boundary condition in a least-squares sense.

This article is organized as follows: In section 2, we state some theoretical results on the existence and uniqueness of solutions to the Minkowski–Dirichlet problem (1.2). In section 3, we provide the divergence formulation of problem (1.2) and associate with it two initial value problems, which differ in the way the Dirichlet boundary condition is treated. The time discretization of these two initial value problems by operator-splitting is discussed in section 4, followed by their finite-element space discretization in section 5. We address in section 6 the initialization of the two above algorithms. In section 7, we report the results of numerical experiments validating our methodology. Section 8 concludes the article.

2. Problem formulation, existence, uniqueness, and regularity results. We defined the Minkowski problem in section 1. In this article, we will focus on the numerical solution of the Minkowski–Dirichlet problem (1.2), assuming that $d = 2$ (2-D). A first step to that goal is to rewrite (1.2) as

$$\begin{cases} \det(\mathbf{D}^2 u) = K(1+|\nabla u|^2)^{1+d/2} & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

a Monge–Ampère-type formulation better suited for numerical solution. In (2.1), K (>0) is the prescribed curvature and $\mathbf{D}^2 u = (\frac{\partial^2 u}{\partial x_i \partial x_j})_{1\le i,j\le d}$ is the Hessian matrix of function u.

To put our computational investigations into perspective, we recall some classical results concerning the existence, uniqueness, and regularity of classical solutions to problem (2.1). In [48], one proves the following results about existence and uniqueness.

THEOREM 2.1. *Suppose that in* (2.1), Ω *is a uniformly bounded convex domain of* $\mathbb{R}^d$, *its boundary* $\partial\Omega$ *having* $C^{1,1}$*-regularity. Then, problem* (2.1) *has, for any* $g \in C^{1,1}(\bar{\Omega})$, *a unique solution in* $C^2(\Omega) \cap C^{0,1}(\bar{\Omega})$ *if and only if*

$$\int_\Omega K dx < \omega_d, \tag{2.2}$$

and

$$K = 0 \text{ on } \partial\Omega. \tag{2.3}$$

The constant ω_d *in* (2.2) *is given by* $\omega_d = \int_{\mathbb{R}^d} \frac{d\xi}{(1+|\xi|^2)^{1+d/2}}$ *(implying* $\omega_2 = \pi$ *and* $\omega_3 = 4\pi/3$*); actually,* ω_d *is the volume of the unit ball of* $\mathbb{R}^d$.

Condition (2.3) is required to make sure that a solution exists for arbitrary g. It is proved in [48] that if K does not vanish on the boundary, one can find a smooth function g such that problem (2.1) has no solution.

In [50, 51, 52], one discusses regularity of the solution in the critical case defined by

$$\int_\Omega K dx = \omega_d, \tag{2.4}$$

where the following results are proved.

THEOREM 2.2. *Let* Ω *be a uniformly convex domain of* $\mathbb{R}^d$ *with a* $C^{2,1}$ *smooth boundary and* K *be a positive* C^2 *smooth function verifying* (2.4). *If* u *is a solution of the Minkowski–Dirichlet problem* (2.1), *then*

(i) $u \in C^{0,1/2}(\Omega)$;
(ii) *the graph of* u *is* $C^{2,\alpha}$*-smooth for some* $\alpha \in (0,1)$;
(iii) $u|_{\partial\Omega}$ *is* $C^{1,\alpha}$*-smooth;*
(iv) *if* $\partial\Omega$ *is* $C^{k+1,\alpha}$ *and* $K \in C^{k-1,\alpha}$ *with* $k \ge 2$, *then the graph of* u *is* $C^{k+1,\alpha}$*-smooth and* $u|_{\partial\Omega}$ *is* $C^{k+1,\alpha}$*-smooth.*

See [49] for more details on the solution of the Minkowski problem.

Some of the conditions in the above two theorems are rather restrictive and/or not easy to verify. Nevertheless, the results they are reporting are very useful from two perspectives: on one hand, they suggest test problems, where we know in advance that solutions exist; on the other hand, they also suggest some other examples, where the answer to existence will be indicated by the results of our computations. Finally, we will also consider test problems with known solutions so as to check how fast and how accurately our methods recover them.

3. Divergence formulations of the 2-D Minkowski problem and relaxation by penalty of the Dirichlet condition.

3.1. Synopsis. There are cases where the data K and g do not allow the existence of classical smooth solutions to problem (2.1). In [30], one introduces a notion of viscosity solution to problem (2.1), with the solution satisfying the generalized Monge–Ampère equation in [2], but not necessarily the Dirichlet condition. In the following subsections, we will consider two divergence formulations of problem (2.1) in dimension two to enforce the Dirichlet condition. The first formulation keeps the Dirichlet condition as it is and is well suited to those situations where problem (2.1) has classical solutions. On the other hand, the second formulation makes use of a penalty to relax the Dirichlet condition; for large values of the penalty parameter, one recovers accurately classical solutions if such solutions do exist, or generalized solutions in the absence of classical solutions.

3.2. A first divergence formulation of the 2-D Minkowski–Dirichlet problem. If $d = 2$, problem (2.1) enjoys the following equivalent formulation (in the sense of distributions):

$$\begin{cases} -\nabla \cdot \big(\mathrm{cof}(\mathbf{D}^2 u)\nabla u\big) + 2K(1+|\nabla u|^2)^2 = 0 & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{3.1}$$

where matrix $\mathrm{cof}(\mathbf{D}^2 u)$ is the cofactor matrix of Hessian $\mathbf{D}^2 u$, that is,

$$\mathrm{cof}(\mathbf{D}^2 u) = \begin{pmatrix} \frac{\partial^2 u}{\partial x_2^2} & -\frac{\partial^2 u}{\partial x_1 \partial x_2} \\ -\frac{\partial^2 u}{\partial x_1 \partial x_2} & \frac{\partial^2 u}{\partial x_1^2} \end{pmatrix}.$$

Problem (3.1) is equivalent to

$$\begin{cases} \begin{cases} -\nabla \cdot (\mathrm{cof}(\mathbf{p})\nabla u) + 2K(1+|\mathbf{s}|^2)^2 = 0 & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \\ \mathbf{p} - \mathbf{D}^2 u = \mathbf{0} \quad \text{in } \Omega, \\ \mathbf{s} = \nabla u \quad \text{in } \Omega. \end{cases} \tag{3.2}$$

In order to avoid possible difficulties at those points of $\bar{\Omega}$ where K may vanish, we approximate system (3.2) by

$$\begin{cases} \begin{cases} -\nabla \cdot ((\varepsilon \mathbf{I} + \mathrm{cof}(\mathbf{p}))\nabla u) + 2K(1+|\mathbf{s}|^2)^2 = 0 & \text{in } \Omega, \\ u|_{\partial\Omega} = g & \text{on } \Omega, \end{cases} \\ \mathbf{p} - \mathbf{D}^2 u = \mathbf{0}, \\ \mathbf{s} - \nabla u = \mathbf{0}, \end{cases} \tag{3.3}$$

with ε a small positive parameter. We successfully used this type of regularization in [28] for the solution of the canonical Monge–Ampère equation (1.3) completed by Dirichlet boundary conditions. In practice, we will use a piecewise linear finite-element basis and take ε of the order of h^2, h being a space discretization step. Such a choice makes the scheme stable while providing optimal second-order accuracy.

To solve system (3.3) we are going to associate with it the following initial value problem:

$$
\begin{cases}
\begin{cases}
\dfrac{\partial u}{\partial t} - \nabla \cdot \left(\left(\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}) \right) \nabla u \right) + 2K(1 + |\mathbf{s}|^2)^2 = 0 & \text{in } \Omega \times (0, +\infty), \\
u|_{\partial\Omega} = g & \text{on } \partial\Omega \times (0, +\infty),
\end{cases} \\
\dfrac{\partial \mathbf{p}}{\partial t} + \gamma_1 \left(\mathbf{p} - \mathbf{D}^2 u \right) = \mathbf{0} \text{ in } \Omega \times (0, +\infty), \\
\dfrac{\partial \mathbf{s}}{\partial t} + \gamma_2 \left(\mathbf{s} - \nabla u \right) = \mathbf{0} \text{ in } \Omega \times (0, +\infty), \\
(u(0), \mathbf{p}(0), \mathbf{s}(0)) = (u_0, \mathbf{p}_0, \mathbf{s}_0),
\end{cases} \tag{3.4}
$$

which will be time-discretized by operator-splitting (in section 4.1). In (3.4), γ_1 and γ_2 are two positive coefficients chosen so that the smooth modes of $\mathbf{p}$ and $\mathbf{s}$ evolve in time roughly at the same speed as that of u. Roughly speaking, the evolution speed of u is controlled by the eigenvalue of $-\nabla^2 u$ and the eigenvalue of $\mathbf{p} \approx \mathbf{D}^2 u$. According to (2.1), if the eigenvalues of $\mathbf{D}^2 u$ are close to each other, then they are in the order of $\sqrt{K}$. Following [28], we advocate defining γ_1 and γ_2 by

$$
\begin{aligned}
\gamma_1 &= \beta_1 \lambda_0 \left(\varepsilon + \sqrt{\alpha} \right), \\
\gamma_2 &= \beta_2 \lambda_0 \left(\varepsilon + \sqrt{\alpha} \right),
\end{aligned}
$$

where λ_0 is the smallest eigenvalue of operator $-\nabla^2$ in $H_0^1(\Omega)$, α is the lower bound of K, and β_1 and β_2 are two constants of order one.

We comment in passing that we have used and will continue to use the notation $\phi(t)$ for the function $x \to \phi(x, t)$. In section 6, we will discuss the initialization of system (3.4).

3.3. A divergence formulation of the 2-D Minkowski–Dirichlet problem with relaxation of the boundary condition. Theorem 2.1 implies that problem (2.1) may have no solution, unless function K belongs to a very specific class of functions. In order to deal with such no-solution scenarios as well as we can, we are going to relax the boundary condition $u = g$ using a penalty technique of the least-squares type. If problem (2.1) has a classical solution, we expect to recover it when the penalty parameter converges to $+\infty$.

The simplest way to proceed is to start from the following variational formulation verified (formally) by any solution u of problem (2.1):

$$
\begin{cases}
u \in H^1(\Omega), \\
\displaystyle\int_\Omega (\operatorname{cof}(\mathbf{D}^2 u) \nabla u) \cdot \nabla v dx + 2 \int_\Omega K(1 + |\nabla u|^2)^2 \, v \, dx = 0 \ \ \forall v \in H_0^1(\Omega), \\
u = g \text{ on } \partial\Omega.
\end{cases} \tag{3.5}
$$

In order to relax the Dirichlet boundary condition, we are going to apply to problem (3.5) the well-known penalty method discussed in [23, 24] to approximate Dirichlet's problems for linear second-order elliptic operators by Robin's ones.

Let κ be a positive constant. We (formally) approximate the variational problem (3.5) by

$$
\begin{cases}
u \in H^1(\Omega), \\
\displaystyle\int_\Omega (\operatorname{cof}(\mathbf{D}^2 u) \nabla u) \cdot \nabla v dx + 2 \int_\Omega K(1 + |\nabla u|^2)^2 v dx \\
\quad + \kappa \textstyle\int_{\partial\Omega} (u - g) v d\Gamma = 0 \ \ \forall v \in H^1(\Omega),
\end{cases} \tag{3.6}
$$

where coefficient κ acts as a weight, controlling the level of penalization. Some remarks are in order.

Remark 3.1. Let us consider the functional $j_2 : H^1(\Omega) \to \mathbb{R}$ defined by

$$j_2(v) = \frac{\kappa}{2} \int_{\partial\Omega} |v - g|^2 d\Gamma \quad \forall v \in H^1(\Omega).$$

Functional j_2 is convex and C^∞ over $H^1(\Omega)$, its differential $Dj_2(v)$ at v being given by

$$\langle Dj_2(v), w\rangle = \kappa \int_{\partial\Omega} (v - g) w d\Gamma \quad \forall v, w \in H^1(\Omega), \tag{3.7}$$

where $\langle \cdot, \cdot \rangle$ denotes a duality pairing between $(H^1(\Omega))'$ (the dual space of $H^1(\Omega)$) and $H^1(\Omega)$. Consequently, we can identify $Dj_2(u)$ with $\kappa(u|_{\partial\Omega} - g)$ and replace $\kappa \int_{\partial\Omega}(u - g)v d\Gamma$ in (3.6) by $\langle Dj_2(u), v\rangle$.

Remark 3.2. If a function u is a solution of the nonlinear variational problem (3.6), it is also a solution (in the sense of distributions) of the following (fully nonlinear) boundary value problem:

$$\begin{cases} -\nabla \cdot (\mathrm{cof}(\mathbf{D}^2 u)\nabla u) + 2K(1 + |\nabla u|^2)^2 = 0 & \text{in } \Omega, \\ \dfrac{1}{\kappa}(\mathrm{cof}(\mathbf{D}^2 u)\nabla u) \cdot \mathbf{n} + u = g & \text{on } \partial\Omega, \end{cases} \tag{3.8}$$

where, in (3.8), $\mathbf{n}$ denotes the unit outward normal vector at $\partial\Omega$. The boundary condition in (3.8) is a (nonlinear) *Robin* boundary condition. When $\kappa \to +\infty$, problem (3.8) "converges" (formally) to problem (2.1), justifying our second divergence formulation of problem (2.1).

Remark 3.3. A natural alternative to problem (3.6) is the one described by

$$\begin{cases} u \in H^1(\Omega), \\ \displaystyle\int_\Omega (\mathrm{cof}(\mathbf{D}^2 u)\nabla u) \cdot \nabla v dx + 2\int_\Omega K(1 + |\nabla u|^2)^2 v dx \\ \quad + \langle \partial j_1(u), v\rangle = 0 \ \ \forall v \in H^1(\Omega), \end{cases} \tag{3.9}$$

where, in (3.9), $\partial j_1(u)$ is the subdifferential at u of the convex Lipschitz continuous functional $j_1 : H^1(\Omega) \to \mathbb{R}$, defined by

$$j_1(v) = \kappa \int_{\partial\Omega} |v - g| d\Gamma \quad \forall v \in H^1(\Omega).$$

This type of L^1 functional is very common in nonsmooth mechanics and increasingly popular in data science as shown by various chapters of [29].

Proceeding as in section 3.2, we associate with (3.6) the semivariational system

$$\begin{cases} \begin{cases} u \in H^1(\Omega), \\ \displaystyle\int_\Omega \left((\frac{1}{2}\varepsilon\mathbf{I} + \mathrm{cof}(\mathbf{p}))\nabla u \right) \cdot \nabla v dx \ + \ \int_\Omega \frac{1}{2}\varepsilon \nabla u \cdot \nabla v dx \\ \qquad + 2\displaystyle\int_\Omega K(1 + |\mathbf{s}|^2)^2 v dx \ + \ \kappa \int_{\partial\Omega} (u - g) v d\Gamma = 0 \quad \forall v \in H^1(\Omega), \end{cases} \\ \mathbf{p} - \mathbf{D}^2 u = \mathbf{0}, \\ \mathbf{s} - \nabla u = \mathbf{0}, \end{cases} \tag{3.10}$$

where in the second row, $\int_\Omega (\frac{1}{2}\varepsilon\mathbf{I})\nabla u \cdot \nabla v dx + \int_\Omega \frac{1}{2}\varepsilon \nabla u \cdot \nabla v dx (= \int_\Omega \varepsilon \nabla u \cdot \nabla v dx)$ is the regularization term with a role similar to $\nabla \cdot ((\varepsilon\mathbf{I})\nabla u)$ in (3.3). The next step

is to associate with (3.10) an initial value problem, as we have done with (3.3) in section 3.2. The initial value problem reads as

$$
\begin{cases}
\begin{cases}
u(t) \in H^1(\Omega),\ \forall t > 0, \\
\displaystyle\int_\Omega \frac{\partial u}{\partial t} v dx + \int_\Omega \left[\frac{1}{2}\varepsilon I + \operatorname{cof}(\mathbf{p})\right] \nabla u \cdot \nabla v dx + \int_\Omega \frac{1}{2}\varepsilon \nabla u \cdot \nabla v dx \\
\quad + 2\displaystyle\int_\Omega K(1+|\mathbf{s}|^2)^2 v dx + \kappa \int_{\partial\Omega} (u-g) v d\Gamma = 0 \quad \forall v \in H^1(\Omega),
\end{cases} \\
\dfrac{\partial \mathbf{p}}{\partial t} + \gamma_1(\mathbf{p} - \mathbf{D}^2 u) = \mathbf{0} \text{ in } \Omega \times (0, +\infty), \\
\dfrac{\partial \mathbf{s}}{\partial t} + \gamma_2(\mathbf{s} - \nabla u) = \mathbf{0} \text{ in } \Omega \times (0, +\infty), \\
(u(0), \mathbf{p}(0), \mathbf{s}(0)) = (u_0, \mathbf{p}_0, \mathbf{s}_0).
\end{cases} \tag{3.11}
$$

As in section 3.2, we advocate taking

$$
\begin{aligned}
\gamma_1 &= \beta_1 \lambda_0 \left(\varepsilon + \sqrt{\alpha}\right), \\
\gamma_2 &= \beta_2 \lambda_0 \left(\varepsilon + \sqrt{\alpha}\right).
\end{aligned}
$$

In section 6, we will discuss the initialization of system (3.11).

The main difference between (3.4) and (3.11) is how the boundary condition is implemented. Problem (3.4) enforces the Dirichlet boundary condition in a pointwise manner, while (3.11) enforces the Dirichlet boundary condition in a weak sense so that pointwise mismatch is allowed.

4. Discretization of the initial value problems (3.4) and (3.11) by operator-splitting. In this section, we are going to apply the Lie scheme to the time-discretization of the initial value problems (3.4) and (3.11); see [29] for details on the Lie scheme. In our splitting strategy, each evolution step is split into several fractional steps so that at each fractional step, we only focus on a few operators and update each variable implicitly and independently instead of solving a large system including all variables simultaneously. Another benefit is that with this strategy, $\mathbf{p}$ and $\mathbf{s}$ are updated using the already updated u, which, in general, will improve the convergence behavior of the algorithm.

In the following, let Δt (> 0) denote a time-discretization step, $t^n = n\Delta t$, and let $(u^n, \mathbf{p}^n, \mathbf{s}^n)$ denote an approximation of $(u, \mathbf{p}, \mathbf{s})$ at $t = t^n$.

4.1. Time discretization of the initial value problem (3.4). The Lie-scheme we employ here is a variant of the one we used in [28] to solve the Monge–Ampère equation (1.2) completed by a Dirichlet boundary condition. It reads as

$$
(u^0, \mathbf{p}^0, \mathbf{s}^0) = (u_0, \mathbf{p}_0, \mathbf{s}_0). \tag{4.1}
$$

For $n \geq 0$, $(u^n, \mathbf{p}^n, \mathbf{s}^n) \to (u^{n+1/2}, \mathbf{p}^{n+1/2}, \mathbf{s}^{n+1/2}) \to (u^{n+1}, \mathbf{p}^{n+1}, \mathbf{s}^{n+1})$ as follows.

The first fractional step. Solve

$$
\begin{cases}
\begin{cases}
\dfrac{\partial u}{\partial t} - \nabla \cdot \left[\left(\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}^n)\right) \nabla u\right] + 2K(1 + |\mathbf{s}^n|^2)^2 = 0 & \text{in } \Omega \times (t^n, t^{n+1}), \\
u = g & \text{on } \partial\Omega \times (t^n, t^{n+1}),
\end{cases} \\
\dfrac{\partial \mathbf{p}}{\partial t} = \mathbf{0} \text{ in } \Omega \times (t^n, t^{n+1}), \\
\dfrac{\partial \mathbf{s}}{\partial t} = \mathbf{0} \text{ in } \Omega \times (t^n, t^{n+1}), \\
(u, \mathbf{p}, \mathbf{s})(t^n) = (u^n, \mathbf{p}^n, \mathbf{s}^n),
\end{cases} \tag{4.2}
$$

and set

$$u^{n+1/2} = u(t^{n+1}), \quad \mathbf{p}^{n+1/2} = \mathbf{p}(t^{n+1})(= \mathbf{p}^n), \quad \mathbf{s}^{n+1/2} = \mathbf{s}(t^{n+1})(= \mathbf{s}^n). \tag{4.3}$$

The second fractional step. Solve

$$\begin{cases} \dfrac{\partial u}{\partial t} = 0 & \text{in } \Omega \times (t^n, t^{n+1}), \\ \dfrac{\partial \mathbf{p}}{\partial t} + \gamma_1(\mathbf{p} - \mathbf{D}^2 u^{n+1/2}) = 0 & \text{in } \Omega \times (t^n, t^{n+1}), \\ \dfrac{\partial \mathbf{s}}{\partial t} + \gamma_2(\mathbf{s} - \nabla u^{n+1/2}) = 0 & \text{in } \Omega \times (t^n, t^{n+1}), \\ (u, \mathbf{p}, \mathbf{s})(t^n) = (u^{n+1/2}, \mathbf{p}^{n+1/2}, \mathbf{s}^{n+1/2}), \end{cases} \tag{4.4}$$

and set

$$u^{n+1} = u(t^{n+1})(= u^{n+1/2}), \quad \mathbf{p}^{n+1} = P_+\left[\mathbf{p}(t^{n+1})\right], \quad \mathbf{s}^{n+1} = \mathbf{s}(t^{n+1}). \tag{4.5}$$

In (4.5), $P_+(\cdot)$ is a (kind of) projection operator which maps the space of the 2×2 symmetric matrices onto the closed cone of the 2×2 symmetric positive semidefinite matrices; we will return to operator P_+ in section 5.6.

We still need to solve the initial value problems that one encounters in (4.2) and (4.4). There is no difficulty with (4.4) since the three initial value problems it contains have closed-form solutions, leading to

$$\begin{cases} u(t^{n+1}) = u^{n+1/2}, \\ \mathbf{p}(t^{n+1}) = e^{-\gamma_1 \Delta t}\mathbf{p}^n + \left(1 - e^{-\gamma_1 \Delta t}\right)\mathbf{D}^2 u^{n+1}, \\ \mathbf{s}(t^{n+1}) = e^{-\gamma_2 \Delta t}\mathbf{s}^n + \left(1 - e^{-\gamma_2 \Delta t}\right)\nabla u^{n+1}. \end{cases}$$

It remains to solve the parabolic problem (4.2); for its solution, we advocate performing just one step of the backward Euler scheme, which enables us to use a relatively large time step while keeping the algorithm stable. We obtain then

$$\begin{cases} \dfrac{u^{n+1} - u^n}{\Delta t} - \nabla \cdot \left[(\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}^n))\nabla u^{n+1}\right] + 2K(1 + |\mathbf{s}^n|^2)^2 = 0 & \text{in } \Omega, \\ u^{n+1} = g & \text{on } \partial\Omega, \end{cases}$$

a (relatively) simple Dirichlet problem for a linear self-adjoint second-order strongly elliptic operator with variable coefficients, well-suited to finite-element approximations as we shall see in section 5.

Collecting the above results, we will employ the following time-discretization scheme to solve the initial value problem (3.4):

$$(u^0, \mathbf{p}^0, \mathbf{s}^0) = (u_0, \mathbf{p}_0, \mathbf{s}_0). \tag{4.6}$$

For $n \geq 0$, $(u^n, \mathbf{p}^n, \mathbf{s}^n) \to (u^{n+1}, \mathbf{p}^{n+1}, \mathbf{s}^{n+1})$ as follows.

Solve

$$\begin{cases} \dfrac{u^{n+1} - u^n}{\Delta t} - \nabla \cdot \left[(\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}^n))\nabla u^{n+1}\right] + 2K(1 + |\mathbf{s}^n|^2)^2 = 0 & \text{in } \Omega, \\ u^{n+1} = g & \text{on } \partial\Omega, \end{cases} \tag{4.7}$$

and compute

$$\begin{cases} \mathbf{p}^{n+1} = P_+\left[e^{-\gamma_1 \Delta t}\mathbf{p}^n + \left(1 - e^{-\gamma_1 \Delta t}\right)\mathbf{D}^2 u^{n+1}\right], \\ \mathbf{s}^{n+1} = e^{-\gamma_2 \Delta t}\mathbf{s}^n + \left(1 - e^{-\gamma_2 \Delta t}\right)\nabla u^{n+1}. \end{cases} \tag{4.8}$$

4.2. Time discretization of the initial value problem (3.11). As expected, there are many commonalities between the ways we discretize systems (3.4) and (3.11); we will take advantage of them. The major difference is that it is much easier to operate directly on the variational formulation of the Monge–Ampère part of the problem so as to avoid dealing explicitly with the complicated Robin condition we visualized in (3.8). Denote the updated boundary condition at t^n by g^n. The Lie scheme we are going to use reads as

$$(u^0,\mathbf{p}^0,\mathbf{s}^0,g^0)=(u_0,\mathbf{p}_0,\mathbf{s}_0,g). \tag{4.9}$$

$$\text{For } n\geq 0,\quad (u^n,\mathbf{p}^n,\mathbf{s}^n,g^n)\to(u^{n+1/3},\mathbf{p}^{n+1/3},\mathbf{s}^{n+1/3},g^{n+1/3})\to$$
$$\to(u^{n+2/3},\mathbf{p}^{n+2/3},\mathbf{s}^{n+2/3},g^{n+2/3})\to(u^{n+1},\mathbf{p}^{n+1},\mathbf{s}^{n+1},g^{n+1}),$$

where we outline the three fractional steps as the following.

The first fractional step. Solve

$$\begin{cases}\begin{cases}u(t)\in H^1(\Omega)\ \ \forall t\in(t^n,t^{n+1}),\\ \displaystyle\int_\Omega\frac{\partial u}{\partial t}(t)v dx+\int_\Omega\left[\left(\frac{\varepsilon}{2}\mathbf{I}+\operatorname{cof}(\mathbf{p}(t))\right)\nabla u(t)\right]\cdot\nabla v dx\\ \quad+2\displaystyle\int_\Omega K(1+|\mathbf{s}(t)|^2)^2dx=0 \text{ in } \Omega\times(t^n,t^{n+1})\ \forall v\in H_0^1(\Omega),\\ u=g^n \text{ on } \partial\Omega\times(t^n,t^{n+1}),\end{cases}\\ \dfrac{\partial\mathbf{p}}{\partial t}=\mathbf{0} \text{ in } \Omega\times(t^n,t^{n+1}),\\ \dfrac{\partial\mathbf{s}}{\partial t}=\mathbf{0} \text{ in } \Omega\times(t^n,t^{n+1}),\\ (u,\mathbf{p},\mathbf{s})(t^n)=(u^n,\mathbf{p}^n,\mathbf{s}^n),\end{cases} \tag{4.10}$$

and set

$$u^{n+1/3}=u(t^{n+1}),\ \mathbf{p}^{n+1/3}=\mathbf{p}(t^{n+1}),\ \mathbf{s}^{n+1/3}=\mathbf{s}(t^{n+1}),\ g^{n+1/3}=g^n. \tag{4.11}$$

The second fractional step. Solve

$$\begin{cases}\dfrac{\partial u}{\partial t}=0 \text{ in } \Omega\times(t^n,t^{n+1}) & \text{in } \Omega\times(t^n,t^{n+1}),\\ \dfrac{\partial\mathbf{p}}{\partial t}+\gamma_1(\mathbf{p}-\mathbf{D}^2u^{n+1/3})=0 & \text{in } \Omega\times(t^n,t^{n+1}),\\ \dfrac{\partial\mathbf{s}}{\partial t}+\gamma_2(\mathbf{s}-\nabla u^{n+1/3})=0 & \text{in } \Omega\times(t^n,t^{n+1}),\\ (u,\mathbf{p},\mathbf{s})(t^n)=(u^{n+1/3},\mathbf{p}^{n+1/3},\mathbf{s}^{n+1/3}),\end{cases} \tag{4.12}$$

and set

$$u^{n+2/3}=u(t^{n+1}),\ \mathbf{p}^{n+2/3}=P_+\left[\mathbf{p}(t^{n+1})\right],\ \mathbf{s}^{n+2/3}=\mathbf{s}(t^{n+1}),\ g^{n+2/3}=g^{n+1/3}. \tag{4.13}$$

The third fractional step. Solve

$$\begin{cases}\begin{cases}u\in H^1(\Omega),\\ \displaystyle\int_\Omega\frac{\partial u}{\partial t}(t)v dx+\frac{\varepsilon}{2}\int_\Omega\nabla u(t)\cdot\nabla v dx+\kappa\int_{\partial\Omega}(u(t)-g)v d\Gamma=0\\ \forall v\in H^1(\Omega),\end{cases}\\ \dfrac{\partial\mathbf{p}}{\partial t}=\mathbf{0} \text{ in } \Omega\times(t^n,t^{n+1}),\\ \dfrac{\partial\mathbf{s}}{\partial t}=\mathbf{0} \text{ in } \Omega\times(t^n,t^{n+1}),\\ (u,\mathbf{p},\mathbf{s})(t^n)=(u^{n+2/3},\mathbf{p}^{n+2/3},\mathbf{s}^{n+2/3}),\end{cases} \tag{4.14}$$

and set

$$u^{n+1} = u(t^{n+1}), \mathbf{p}^{n+1} = P_+\left[\mathbf{p}(t^{n+1})\right], \mathbf{s}^{n+1} = \mathbf{s}(t^{n+1}), g^{n+1} = u^{n+1}|_{\partial\Omega}. \tag{4.15}$$

Assuming that one uses just one step of the backward Euler scheme to solve the parabolic problem in (4.10) and (4.14), the Lie scheme (4.9)–(4.15) reduces to the following variant of scheme (4.6)–(4.8):

$$(u^0, \mathbf{p}^0, \mathbf{s}^0, g^0) = (u_0, \mathbf{p}_0, \mathbf{s}_0, g). \tag{4.16}$$

For $n \geq 0$, $(u^n, \mathbf{p}^n, \mathbf{s}^n, g^n) \to u^{n+1/2} \to (u^{n+1}, \mathbf{p}^{n+1}, \mathbf{s}^{n+1}, g^{n+1})$ as follows.

Solve

$$\begin{cases} u^{n+1/2} \in H^1(\Omega), \\ \displaystyle\int_\Omega \frac{u^{n+1/2} - u^n}{\Delta t} + \int_\Omega \left[\left(\frac{\varepsilon}{2}\mathbf{I} + \operatorname{cof}(\mathbf{p}^n)\right)\nabla u^{n+1/2}\right]\cdot\nabla v dx \\ \quad + 2\displaystyle\int_\Omega K(1+|\mathbf{s}^n|^2)^2 v dx = 0 \quad \forall v \in H_0^1(\Omega), \\ u^{n+1/2} = g^n \text{ on } \partial\Omega, \end{cases} \tag{4.17}$$

and compute

$$\begin{cases} \mathbf{p}^{n+1} = P_+\left[e^{-\gamma_1\Delta t}\mathbf{p}^n + \left(1 - e^{-\gamma_1\Delta t}\right)\mathbf{D}^2 u^{n+1/2}\right], \\ \mathbf{s}^{n+1} = e^{-\gamma_2\Delta t}\mathbf{s}^n + \left(1 - e^{-\gamma_2\Delta t}\right)\nabla u^{n+1/2}, \\ \begin{cases} u \in H^1(\Omega), \\ \displaystyle\int_\Omega \frac{u^{n+1} - u^{n+1/2}}{\Delta t} v dx + \frac{\varepsilon}{2}\int_\Omega \nabla u^{n+1}\cdot\nabla v dx \\ \quad + \kappa\displaystyle\int_{\partial\Omega}(u^{n+1} - g^n) v d\Gamma = 0 \quad \forall v \in H^1(\Omega), \end{cases} \\ g^{n+1} = u^{n+1}|_{\partial\Omega}. \end{cases} \tag{4.18}$$

5. Finite elements for the new operator-splitting scheme. The divergence form strongly suggests that we apply a finite-element method to implement (4.7)–(4.8) and (4.17)–(4.18). Here we choose a mixed finite-element method: we use the same function space to approximate u, ∇u, $\mathbf{D}^2 u$, $\mathbf{s}$, and $\mathbf{p}$. Since we will choose basis functions to be piecewise affine functions, the resulting approximations are continuous piecewise affine on Ω.

5.1. Finite-element spaces. Let $\mathcal{T}_h$ be the triangulation of the domain Ω, and let h denote the maximum edge length of the triangles in $\mathcal{T}_h$. Let $\Sigma_h = \{Q_j\}_{j=1}^{N_h}$ be the collection of vertices in $\mathcal{T}_h$, where Q_i denotes a typical vertex. We define the first finite-element space as

$$V_h = \left\{v | v \in C^0\left(\bar{\Omega}\right), v|_T \in P_1 \,\forall T \in \mathcal{T}_h\right\}, \tag{5.1}$$

where P_1 denotes the space of polynomials with degree no larger than 1.

Accordingly, we associate each vertex Q_j with a shape function w_j such that

$$w_j \in V_h, w_j(Q_j) = 1, w_j(Q_k) = 0 \ \forall k = 1, \ldots, N_h, \, k \neq j,$$

where the support of w_j, denoted θ_j, is the union of triangles that have the same common vertex Q_j, and we denote the area of θ_j by $|\theta_j|$. The set $\mathcal{B} = \{w_j\}_{j=1}^{N_h}$ forms a collection of basis functions of V_h. In other words, we have

$$v = \sum_{j=1}^{N_h} v(Q_j) w_j \quad \forall v \in V_h.$$

In addition, we define

$$V_{gh} = \{v | v \in V_h, v(Q_j) = g(Q_j)\ \forall Q_j \in \Sigma_h \cap \partial\Omega\}, \tag{5.2}$$

where g can be any function which is C^0 on $\partial\Omega$. When $g = 0$, we have that

$$V_{0h} = V_h \cap H_0^1.$$

Meanwhile, we define the following vector-valued spaces:

$$\mathbf{R}_h = \left\{\mathbf{r} |\ \mathbf{r} \in V_h^{2\times 1}\right\},$$
$$\mathbf{Q}_h = \left\{\mathbf{q} |\ \mathbf{q} \in V_h^{2\times 2},\ \mathbf{q} = \mathbf{q}^T\right\},$$

so that we can use functions in $\mathbf{R}_h$ to approximate ∇u and $\mathbf{s}$ and use functions in $\mathbf{Q}_h$ to approximate $\mathbf{D}^2 u$ and $\mathbf{p}$.

5.2. Approximations of the two first-order derivatives of u. For any $v \in V_h$, we denote the first-order derivative approximation $\frac{\partial v}{\partial x_i}$ of v by $D_{ih}(v)$ for $i = 1, 2$, and this approximate derivative operator is defined in the following weak sense:

$$\int_\Omega D_{ih}(v) w dx = \int_\Omega \frac{\partial v}{\partial x_i} w dx,\ \ i = 1, 2,\ \forall w \in H^1(\Omega). \tag{5.3}$$

Since Ω is partitioned by the triangulation $\mathcal{T}_h$, we restrict the test functions w to be in V_h so that we only need to test the above integral against those basis functions w_k for $k = 1, 2, \ldots, N_h$. Since w_k is only supported on θ_k, we have

$$\begin{cases} D_{ih}(v) \in V_h\ \ \forall i = 1, 2, \\ D_{ih}(v)(Q_k) = \frac{3}{|\theta_k|} \int_{\theta_k} \frac{\partial v}{\partial x_i} w_k dx\ \ \forall k = 1, 2, \ldots, N_h. \end{cases} \tag{5.4}$$

We remark in passing that on a regular mesh such as the one shown in Figure 1(a), (5.4) recovers the central-difference approximation at an interior node and one-sided approximation at a boundary node in a finite-difference method based on this mesh.

In some problems, ∇u has singularities on Ω. One challenging situation is when the singularities appear on the boundary. The approximation at nodes near the boundary can blow up, especially when the gradient of the exact solution blows up at the boundary of a computational domain, such as a semisphere. To resolve this problem, we need to regularize the approximation of ∇u. One possible way is to adopt the idea from [28, 8], which is used to approximate the second-order derivative:

$$\begin{cases} D_{ih}(v) \in H_0^1, \\ \varepsilon_1 \displaystyle\int_\Omega \nabla D_{ih} \cdot \nabla w dx + \int_\Omega D_{ih}(v) w dx = \int_\Omega \frac{\partial v}{\partial x_i} w dx,\ \ i = 1, 2,\ \forall w \in H_0^1(\Omega). \end{cases} \tag{5.5}$$

The error of the regularized approximation can be larger than that of the direct approximation, but it is more robust. Moreover, we have

$$\lim_{\epsilon_1, h \to 0} D_{ih}(v) = \frac{\partial v}{\partial x_i} \quad \text{in } L^2(\Omega).$$

5.3. Approximations of second-order derivatives of u. The general idea to approximate the second-order derivatives is similar to the one used in [22, 12, 28]. For completeness, we mention the details here.

For any $v \in V_h$, we denote the approximations of $\frac{\partial^2 v}{\partial x_i \partial x_j}$ by $D^2_{ijh}(v)$ for $i, j = 1, 2$, so that the approximate operator $D^2_{ijh}(v)$ of second-order derivatives is defined in the following weak sense:

$$\int_\Omega D^2_{ijh}(v) w_k dx = \int_\Omega \frac{\partial^2 v}{\partial x_i \partial x_j} w_k dx. \tag{5.6}$$

To resolve the right-hand side of (5.6), we apply the divergence theorem,

$$\int_\Omega \frac{\partial^2 v}{\partial x_i \partial x_j} w dx = \frac{1}{2} \int_{\partial\Omega} \left(\frac{\partial v}{\partial x_i} n_j + \frac{\partial v}{\partial x_j} n_i \right) w d(\partial\Omega) - \frac{1}{2} \int_\Omega \left(\frac{\partial v}{\partial x_i} \frac{\partial w}{\partial x_j} + \frac{\partial v}{\partial x_j} \frac{\partial w}{\partial x_i} \right) dx, \tag{5.7}$$

where $\mathbf{n} = (n_1, n_2)$ is the outward normal direction along $\partial\Omega$. The above approximation is accurate at interior nodes, but the approximation error is large at nodes on the boundary. For example, consider the approximate derivative operator D^2_{11h} on a regular mesh of the unit square; after some derivation, we can show that there is always one node at one of the corners of the unit square such that $D^2_{11h}(v) = 0$ at that node, no matter what form v takes.

To deal with this issue, we treat interior nodes and boundary nodes separately. Let $\Sigma_{0h} = \{Q_k\}_{k=1}^{N_0}$ denote the set of interior nodes in Ω, where we assume that the first N_0 nodes of Σ_h are in the interior of Ω. It follows that we have $\Sigma_h \cap \partial\Omega = \{Q_k\}_{k=N_0+1}^{N_h}$. For $k = 1, 2, \ldots, N_0$, the approximation of (5.6)–(5.7) reduces to

$$\int_\Omega D^2_{ijh}(v) w_k dx = -\frac{1}{2} \int_\Omega \left(\frac{\partial v}{\partial x_i} \frac{\partial w_k}{\partial x_j} + \frac{\partial v}{\partial x_j} \frac{\partial w_k}{\partial x_i} \right) dx. \tag{5.8}$$

To treat nodes on the boundary, the work in [8] used the zero Dirichlet boundary condition for the operator $D^2_{ijh}, i, j = 1, 2$, though the boundary value is not needed in the resulting algorithm. In comparison with the numerical method in [8], ours are different in that the boundary value of D^2_{ijh} is crucial for our splitting algorithm. Specifically, in (4.7)–(4.8) we need boundary values to update $\mathbf{p}$, which is in turn used to compute the divergence operator and to update u. Therefore, we need a better treatment of the boundary nodes.

Here we adopt a strategy from [28, 38] to treat boundary nodes by committing a "variational crime." First, we impose the zero Neumann boundary condition

$$\frac{\partial D^2_{ijh}(v)}{\partial \mathbf{n}} = 0. \tag{5.9}$$

Multiplying (5.9) by w_k for $k = N_0 + 1, \ldots, N_h$ and integrating along $\partial\Omega$, we get

$$\begin{aligned} 0 &= \int_{\partial\Omega} \frac{\partial D^2_{ijh}(v)}{\partial \mathbf{n}} w_k d(\partial\Omega) = \int_\Omega \nabla \cdot \left(\nabla D^2_{ijh}(v) w_k \right) dx \\ &= \int_\Omega \nabla^2 D^2_{ijh}(v) w_k dx + \int_\Omega \nabla D^2_{ijh}(v) \cdot \nabla w_k dx. \end{aligned} \tag{5.10}$$

If $D^2_{ijh}(v)$ is harmonic, implying that $\nabla^2 D^2_{ijh}(v) = 0$, then we have

$$\int_\Omega \nabla D^2_{ijh}(v) \cdot \nabla w_k dx = 0. \tag{5.11}$$

In our algorithm, although D^2_{ijh} is only piecewise harmonic, we still use (5.11) to update boundary values, which is the so-called variational crime. In either approximation (5.6)–(5.7) or approximation (5.8) and (5.11), since w_k is only supported on θ_k, the integration domain can be replaced by θ_k if the test function is w_k. Under certain conditions, a rough derivation shows that the variational crime introduces an error to (5.10) of $O(h)$. Since $D^2_{ijh}(v) \in V_h$, $\nabla D^2_{ijh}(v)$ is piecewise constant over Ω. For any $T \in \mathcal{T}_h$, let ν be one of its edges. Along ν, $\nabla^2 D^2_{ijh}(v)$ is a Dirac-δ function multiplied by a factor (the difference of the values of $\nabla D^2_{ijh}(v)$ over the two triangles having ν as the common boundary). In the interior of T, $\nabla^2 D^2_{ijh}(v)$ is 0. Thus

$$\int_\Omega \nabla^2 D^2_{ijh}(v) w_k dx = \sum_{T\in\theta_k} \int_{\nu\in\partial T} \nabla^2 D^2_{ijh}(v) w_k dx = O(h)$$

if $\nabla D^2_{ijh}(v)$ is bounded by a constant.

In our numerical experiments, with the regularization mechanism introduced below, the accuracy by (5.8) and (5.11) is similar to that by (5.6)–(5.7), but (5.8) and (5.11) make the algorithm more robust. It is worth mentioning that as implemented in [28] both approximations work for two-dimensional Monge–Ampère equations; however, as shown in [38], only the approximation based on the variational crime works for three-dimensional Monge–Ampère equations.

As reported in [8, 28, 38], if we directly use the above approximations, the performance of our algorithm depends on triangulations; in the worst case, on a symmetric mesh as shown in Figure 1(b), our algorithm does not converge. To obtain an algorithm which is robust for all kinds of meshes, we need to regularize the problem by adding some viscosity to our formulation of second-order derivatives.

As a first approach of regularization, we incorporate a local regularization term into the weak definition of second-order derivatives at interior nodes:

$$\begin{cases} \forall i,j=1,2,\ \forall v\in V_h,\ D^2_{ijh}(v)\in V_h \text{ and} \\ C\displaystyle\sum_{T\in\mathcal{T}_h^k} |T| \int_T \nabla D^2_{ijh}(v)\cdot\nabla w_k dx + \int_{\theta_k} D^2_{ijh}(v) w_k dx \\ \quad = -\dfrac{1}{2}\displaystyle\int_{\theta_k} \left[\frac{\partial v}{\partial x_i}\frac{\partial w_k}{\partial x_j} + \frac{\partial v}{\partial x_j}\frac{\partial w_k}{\partial x_i}\right] dx\ \forall k = 1,\dots,N_{0h}, \\ \displaystyle\int_{\theta_k} \nabla D^2_{ijh}(v)\cdot\nabla w_k dx = 0\ \forall k = N_{0h}+1,\dots,N_h, \end{cases} \tag{5.12}$$

where C is a positive constant of order 1, and $\mathcal{T}_h^k$ is the set of all triangles with the common vertex Q_k.

If all triangles in $\mathcal{T}_h$ are of a similar size, (5.12) can be slightly simplified to be

$$\begin{cases} \forall i,j=1,2,\ \forall v\in V_h, D^2_{ijh}(v)\in V_h \text{ and} \\ \varepsilon_1 \displaystyle\int_{\theta_k} \nabla D^2_{ijh}(v)\cdot\nabla w_k dx + \int_{\theta_k} D^2_{ijh}(v) w_k dx \\ \quad = -\dfrac{1}{2}\displaystyle\int_{\theta_k} \left[\frac{\partial v}{\partial x_i}\frac{\partial w_k}{\partial x_j} + \frac{\partial v}{\partial x_j}\frac{\partial w_k}{\partial x_i}\right] dx\ \forall k = 1,\dots,N_{0h}, \\ \displaystyle\int_{\theta_k} \nabla D^2_{ijh}(v)\cdot\nabla w_k dx = 0\ \forall k = N_{0h}+1,\dots,N_h, \end{cases} \tag{5.13}$$

where ε_1 is of order $O(h^2)$.

As a second approach of regularization, we incorporate a double-regularization mechanism into our weak formulation of second-order derivatives. Assuming that $\psi \in H^2$, we consider the following linear elliptic variational problem:

(5.14)
$$\begin{cases} p_{ij}^{\varepsilon} \in H_0^1(\Omega), \\ \varepsilon_1 \displaystyle\int_\Omega \nabla p_{ij}^{\varepsilon} \cdot \nabla \phi dx + \int_\Omega p_{ij}^{\varepsilon} \phi dx = -\frac{1}{2} \int_\Omega \left[\frac{\partial \psi}{\partial x_i} \frac{\partial \phi}{\partial x_j} + \frac{\partial \psi}{\partial x_j} \frac{\partial \phi}{\partial x_i} \right] dx \ \forall \phi \in H_0^1(\Omega), \end{cases}$$

which yields the following relations in the weak sense:

$$\lim_{\varepsilon_1 \to 0} p_{ij}^{\varepsilon} = \frac{\partial^2 \psi}{\partial x_i \partial x_j} \quad \text{in } L^2(\Omega), \tag{5.15}$$

and

$$\begin{cases} -\varepsilon_1 \nabla^2 p_{ij}^{\varepsilon} + p_{ij}^{\varepsilon} = \dfrac{\partial^2 \psi}{\partial x_i \partial x_j} \quad \text{in } \Omega, \\ p_{ij}^{\varepsilon} = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{5.16}$$

Since, as reported in [28], this approximation is not effective in treating the zero-Dirichlet boundary condition, we apply the following correction step:

$$\begin{cases} -\varepsilon_1 \nabla^2 \tilde{p}_{ij}^{\varepsilon} + \tilde{p}_{ij}^{\varepsilon} = p_{ij}^{\varepsilon} \quad \text{in } \Omega, \\ \dfrac{\partial \tilde{p}_{ij}^{\varepsilon}}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{5.17}$$

whose variational formulation reads as

$$\begin{cases} \tilde{p}_{ij}^{\varepsilon} \in H^1(\Omega), \\ \varepsilon_1 \displaystyle\int_\Omega \nabla \tilde{p}_{ij}^{\varepsilon} \cdot \nabla \phi dx + \int_\Omega \tilde{p}_{ij}^{\varepsilon} \phi dx = \int_\Omega p_{ij}^{\varepsilon} \phi dx \ \ \forall \phi \in H^1(\Omega). \end{cases} \tag{5.18}$$

It follows that $\tilde{p}_{ij}^{\varepsilon}$ verifies $\lim_{\varepsilon \to 0} \tilde{p}_{ij}^{\varepsilon} = \frac{\partial^2 \psi}{\partial x_i \partial x_j}$ in $L^2(\Omega)$, and $\tilde{p}_{ij}^{\varepsilon} \in H^4(\Omega)$.

Consequently, the discrete analogue $D_{ijh}^2(v)$ of $\frac{\partial^2 v}{\partial x_i \partial x_j} (1 \le i, j \le 2)$ can be computed in the following way.

Solve

(5.19)
$$\begin{cases} p_{ij} \in V_{0h}, \\ C \displaystyle\sum_{T \in \mathcal{T}_h^k} |T| \int_T \nabla p_{ij} \cdot \nabla w_k dx + \frac{|\theta_k|}{3} p_{ij}(Q_k) = -\frac{1}{2} \int_{\theta_k} \left[\frac{\partial v}{\partial x_i} \frac{\partial w_k}{\partial x_j} + \frac{\partial v}{\partial x_j} \frac{\partial w_k}{\partial x_i} \right] dx \\ \forall k = 1, \ldots, N_{0h}, \end{cases}$$

and then

$$\begin{cases} D_{ijh}^2(v) \in V_h, \\ C \displaystyle\sum_{T \in \mathcal{T}_h^k} |T| \int_T \nabla D_{ijh}^2(v) \cdot \nabla w_k dx + \frac{|\theta_k|}{3} D_{ijh}^2(v)(Q_k) = \frac{|\theta_k|}{3} p_{ij}(Q_k) \\ \forall k = 1, \ldots, N_h, \end{cases} \tag{5.20}$$

where C is a constant of order 1. Similar to the first approach of regularization, if all triangles in $\mathcal{T}_h$ are of a similar size, we can replace $C \sum_{T \in \mathcal{T}_h^k} |T|$ in (5.19) and (5.20) by ϵ_1 that is of order $O(h^2)$.

5.4. Implementation of scheme (4.6)–(4.8). We give a fully discretized analogue of scheme (4.6)–(4.8) as follows.

Initialize

$$u^0 = u_0 \in V_h, \mathbf{p}^0 = \mathbf{p}_0 \in \mathbf{Q}_h, \mathbf{s}^0 = \mathbf{s}_0 \in \mathbf{R}_h. \tag{5.21}$$

For $n \geq 0$, proceed with $\{u^n, \mathbf{p}^n, \mathbf{s}^n\} \to \{u^{n+1}, \mathbf{p}^{n+1}, \mathbf{s}^{n+1}\}$ as follows.

Solve

$$\begin{cases} u^{n+1} \in V_{gh}, \\ \displaystyle\int_\Omega u^{n+1} v dx + \Delta t \int_\Omega (\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}^n)) \nabla u^{n+1} \cdot \nabla v dx \\ \quad \displaystyle = \int_\Omega u^n v dx - 2\Delta t K \int_\Omega (1 + |\mathbf{s}^n|^2)^2 dx \ \forall v \in V_{0h}, \end{cases} \tag{5.22}$$

and compute $\mathbf{p}^{n+1}$ and $\mathbf{s}^{n+1}$ via

$$\begin{cases} \forall k = 1, \ldots, N_h, \\ \alpha = e^{-\gamma_1 \Delta t}, \\ \mathbf{p}^{n+\frac{1}{2}}(Q_k) = \alpha\, \mathbf{p}^n(Q_k) + (1-\alpha) \begin{pmatrix} D^2_{11h}(u^{n+1})(Q_k) & D^2_{12h}(u^{n+1})(Q_k) \\ D^2_{12h}(u^{n+1})(Q_k) & D^2_{22h}(u^{n+1})(Q_k) \end{pmatrix}, \\ \mathbf{p}^{n+1}(Q_k) = P_+ \left[\mathbf{p}^{n+1/2}(Q_k)\right] \end{cases} \tag{5.23}$$

and

$$\begin{cases} \forall k = 1, \ldots, N_h, \\ \mathbf{s}^{n+1}(Q_k) = e^{-\gamma_2 \Delta t} \mathbf{s}^n(Q_k) + \left(1 - e^{-\gamma_2 \Delta t}\right) \begin{pmatrix} D_{1h}(u^{n+1}(Q_k) \\ D_{2h}(u^{n+1}(Q_k) \end{pmatrix}. \end{cases} \tag{5.24}$$

Here, all integrations in (5.22) are computed by the trapezoidal rule. In (5.23) and (5.24), D^1_{ih} for $i = 1, 2$ are computed using (5.5) or (5.3); $D^2_{ijh}(u^{n+1})$ for $i, j = 1, 2$ are computed by approximation (5.12) or (5.19)–(5.20).

5.5. Implementation of scheme (4.16)–(4.18). The discretized analogue of scheme (4.6)–(4.8) can be written as follows.

Initialize

$$u^0 = u_0 \in V_h,\ \mathbf{p}^0 = \mathbf{p}_0 \in \mathbf{Q}_h,\ \mathbf{s}^0 = \mathbf{s}_0 \in \mathbf{R}_h,\ g^0 = g. \tag{5.25}$$

For $n \geq 0$, proceed with $\{u^n, \mathbf{p}^n, \mathbf{s}^n\} \to \{u^{n+1}, \mathbf{p}^{n+1}, \mathbf{s}^{n+1}\}$ as follows.

Solve

$$\begin{cases} u^{n+1/2} \in V_{g^n h}, \\ \displaystyle\int_\Omega u^{n+1/2} v dx + \Delta t \int_\Omega (\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}^n)) \nabla u^{n+1/2} \cdot \nabla v dx \\ \quad \displaystyle = \int_\Omega u^n v dx - 2\Delta t K \int_\Omega (1 + |\mathbf{s}^n|^2)^2 dx \ \forall v \in V_{g^n h}. \end{cases} \tag{5.26}$$

Compute $\mathbf{p}^{n+1}$ and $\mathbf{s}^{n+1}$ via

(5.27)
$$\begin{cases} \forall k = 1, \dots, N_h, \\ \alpha = e^{-\gamma_1 \Delta t}, \\ \mathbf{p}^{n+\frac{1}{2}}(Q_k) = \alpha\, \mathbf{p}^n(Q_k) + (1-\alpha) \begin{pmatrix} D^2_{11h}\left(u^{n+1/2}\right)(Q_k) & D^2_{12h}\left(u^{n+1/2}\right)(Q_k) \\ D^2_{12h}\left(u^{n+1/2}\right)(Q_k) & D^2_{22h}\left(u^{n+1/2}\right)(Q_k) \end{pmatrix}, \\ \mathbf{p}^{n+1}(Q_k) = P_+\left[\mathbf{p}^{n+1/2}(Q_k)\right] \end{cases}$$

and

$$\begin{cases} \forall k = 1, \dots, N_h, \\ \mathbf{s}^{n+1}(Q_k) = e^{-\gamma_2 \Delta t} \mathbf{s}^n(Q_k) + \left(1 - e^{-\gamma_2 \Delta t}\right) \begin{pmatrix} D_{1h}(u^{n+1/2}(Q_k)) \\ D_{2h}(u^{n+1/2}(Q_k)) \end{pmatrix}. \end{cases} \tag{5.28}$$

Compute

$$\begin{cases} u^{n+1} \in V_h, \\ \displaystyle\int_\Omega u^{n+1} v dx + \Delta t \varepsilon \int_\Omega \nabla u^{n+1} \cdot \nabla v dx + \Delta t \int_{\partial\Omega} u^{n+1} v dx \\ \quad = \displaystyle\int_\Omega u^{n+1/2} v dx + \Delta t \varepsilon \int_{\partial\Omega} g v dx \ \forall v \in V_h \end{cases} \tag{5.29}$$

and update

$$g^{n+1} = u^{n+1}|_{\partial\Omega}. \tag{5.30}$$

All integrations in (5.26) and (5.29) are computed by the trapezoidal rule. In (5.27) and (5.28), D^1_{ih} for $i = 1, 2$ are computed using (5.5) or (5.3); $D^2_{ijh}(u^{n+1})$ for $i, j = 1, 2$ are computed by approximation (5.12) or (5.19)–(5.20).

5.6. The projection operator $P_+(\cdot)$. Since we want to find a convex solution u, we need to have some mechanism to enforce convexity in our algorithm. There are many possible approaches to handling the issue.

One particular approach that we discuss here is to modify one of the finite-element components, $\mathbf{p}$, after each iteration so that the modified $\mathbf{p}$ satisfies some convexity-related properties. Since the Hessian matrix of a convex function is semipositive definite and we expect $\mathbf{p}$ to converge to the Hessian matrix of the exact solution u^*, which is convex, it is reasonable to force $\mathbf{p}$ to be semipositive definite; therefore, we introduce a spectral projection operator to achieve this, and $P_+(\cdot)$ is such a projector in our algorithm.

Let $\mathbf{A}$ be a symmetric 2×2 matrix. Assume that $\mathbf{A}$ has a spectral decomposition, $\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$, where the columns of $\mathbf{S}$ are the eigenvectors of $\mathbf{A}$ and $\mathbf{\Lambda} = \left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$. We define the spectral projector operator $P_+(\cdot)$ as

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

where $\lambda_i^+ = \max\{\lambda_i, 0\}$ for $i = 1, 2$. The effect of $P_+(A)$ is to project A onto the cone consisting of semipositive definite matrices. This projection during each iteration makes (4.7) an elliptic PDE of u.

Another possible approach is to choose a convex initial condition, which will be discussed in the next section.

6. Initialization.

6.1. Initial condition for scheme (4.6)–(4.8). To initialize u_0 and $\mathbf{p}_0$ for scheme (4.6)–(4.8), we solve the standard Monge–Ampère equation

$$\begin{cases} \det(\mathbf{D}^2 u_0) = K, \\ u_0 = g \ \text{ on } \partial\Omega. \end{cases} \tag{6.1}$$

We will deal with (6.1) by adopting the method in [28], which solves the following initial value problem to steady state:

$$\begin{cases} \begin{cases} \dfrac{\partial u}{\partial t} - \nabla \cdot ((\varepsilon \mathbf{I} + \operatorname{cof}(\mathbf{p}))\nabla u) + 2K = 0, \\ u = g \text{ on } \partial\Omega, \end{cases} \\ \dfrac{\partial \mathbf{p}}{\partial t} + \gamma(\mathbf{p} - \mathbf{D}^2 u) = \mathbf{0}. \end{cases} \tag{6.2}$$

Let $\{u_*, \mathbf{p}_*\}$ be the steady state of (6.2). Accordingly, we set $u_0 = u_*$, $\mathbf{p}_0 = \mathbf{D}^2 u_*$, and $\mathbf{s} = \mathbf{D}u_*$ as the initial condition for our scheme (4.6)–(4.8). Therefore, our algorithm can be summarized as a two-stage method.

Stage 1. In the algorithm in [28], set $\varepsilon = \varepsilon_1 = h^2$ and $dt = 2h^2$. Solve (6.2) until $\|u^{n+1} - u^n\|_2 < tol$ to get u_0. Compute $\mathbf{p}_0 = \mathbf{D}^2 u_0$ and $\mathbf{s}_0 = \mathbf{D}u_0$.

Stage 2. With the initial condition u_0, $\mathbf{p}_0$, and $\mathbf{s}_0$, solve (4.6)–(4.8) to steady state.

6.2. Initial condition for scheme (4.16)–(4.18). When we use scheme (4.16)–(4.18), the boundary value of the computed solution does not satisfy the given boundary condition, so the initial condition used for scheme (4.6)–(4.8) may not help. To initialize scheme (4.16)–(4.18), we use the initial condition used to solve (6.2) in [28]:

$$\begin{cases} \nabla^2 u_0 = 2\lambda\sqrt{K}, \\ u_0|_{\partial\Omega} = g, \end{cases} \tag{6.3}$$

where λ (> 0) is of order $O(1)$.

7. Numerical experiments.

In this section, we carry out a variety of numerical experiments in different settings to demonstrate the performance of scheme (4.6)–(4.8) and scheme (4.16)–(4.18). Four different meshes, as shown in Figure 1, will be used in our experiments: (a) regular meshes on a unit square, (b) symmetric meshes on a unit square, (c) unstructured meshes on a unit square, and (d) unstructured meshes

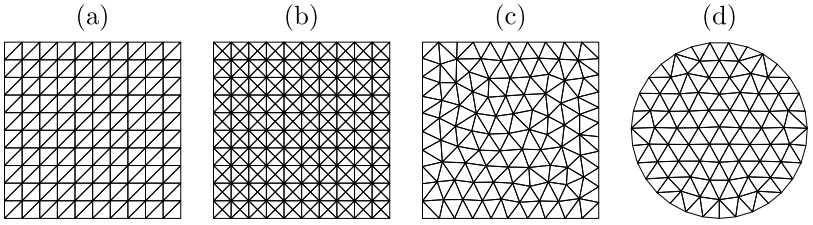

FIG. 1. *Four meshes for two different domains used in our numerical experiments.* (a) *A regular mesh on a square.* (b) *A (highly) symmetric mesh on a square.* (c) *An anisotropic unstructured mesh on a square.* (d) *An anisotropic unstructured mesh on a half-unit disk.*

on a half-unit disk. In all of our experiments, in Stage 1 of our algorithm, we use the method in [28] to initialize the iteration of our algorithm, where we use $tol = h^2$.

There are several parameters in our algorithm: γ_1 and γ_2 (defined in section 3.2), ε (regularization parameter in the PDE (3.3)), ε_1 (regularization parameter in the first-order and second-order derivative approximations in sections 5.2 and 5.3), and time step Δt. In general, when a smooth solution exists, our algorithm is not sensitive to the choice of parameters. Setting ε and ε_1 in the order of h^2 and β_1 and β_2 (in the formulas of γ_1 and γ_2) of $O(1)$ makes the algorithm stable. The time step Δt determines how fast our algorithm converges. Our algorithm converges as long as Δt is small enough and other parameters are set as mentioned above. A large Δt will make our algorithm converge faster, but it may destroy the stability. In our experiments, setting Δt in the order of h^2 makes our algorithm yield reasonable results. For some problems with singular solutions or derivatives blowing up along boundaries, we will take both ε and ε_1 to be a larger value and take Δt to be a smaller value in order to stabilize the algorithm.

Without specification, we choose $\Delta t = 2h^2$ and $\varepsilon = \varepsilon_1 = h^2$ in both Stages 1 and 2 of our algorithm. For examples with compatible boundary condition, scheme (4.6)–(4.8) is used. For examples with incompatible boundary condition, scheme (4.16)–(4.18) is used. We also compare the numerical solutions by both schemes on some examples. Without specification, stopping criterion $\|u^{n+1} - u^n\|_2 < 10^{-6}$ and scheme (4.6)–(4.8) are used. This stopping criterion is selected so that our algorithm converges on all meshes. This criterion may be demanding for some coarse meshes, as many iterations are not necessary. Nevertheless, our current setting does not affect the demonstration of the performance of our proposed algorithms. A more practical way is to set the stopping criterion depending on h. Additional numerical results are presented in the supplementary materials (M159077_SM.pdf [local/web 2.41MB]).

7.1. Example 1. For the first example, we choose the exact solution u^* as a quadratic function,

$$u^* = \alpha\,(x_1 - 0.5)^2 + (x_2 - 0.5)^2/\alpha, \tag{7.1}$$

so that the Gauss curvature $K = \frac{4}{1+4\alpha\,(x_1-0.5)^2+\frac{4}{\alpha}(x_2-0.5)^2}$ and the boundary condition $g = u^*|_{\partial\Omega}$, where α is a positive constant.

Since the solution of this example is smooth, we use (5.4) to approximate the first-order derivatives. In the first test, we choose $\alpha = 1$ so that u^* represents a family of concentric circles which vary isotropically.

With the second-order derivatives approximated by (5.13) and scheme (4.6)–(4.8), the graphs and convergence histories of numerical solutions on different meshes are shown in Figure 2. The numbers of iterations and accuracy orders are shown in Table 1, where the accuracy orders in the L_2 and L_∞ norms are in general larger than 1.5. Stopping criterion $\|u^{n+1} - u^n\| < 10^{-9}$ is used. Since the time step is in the order of h^2, we expect that the rate of convergence is close to 2. In Table 1, the rate is around 1.8, which is slightly better than our expectation.

With the second-order derivatives approximated by (5.19) and (5.20) and scheme (4.6)–(4.8), we can use a less demanding stopping criterion. Here we use $\|u^{n+1} - u^n\| < 10^{-7}$. The numbers of iterations, the errors of approximation, and the rates of convergence on regular and symmetric meshes of the unit square are shown in Table 2, which demonstrates that, in general, our algorithm with approximation (5.19) and (5.20) is first-order accurate, and in comparison with the results based on the

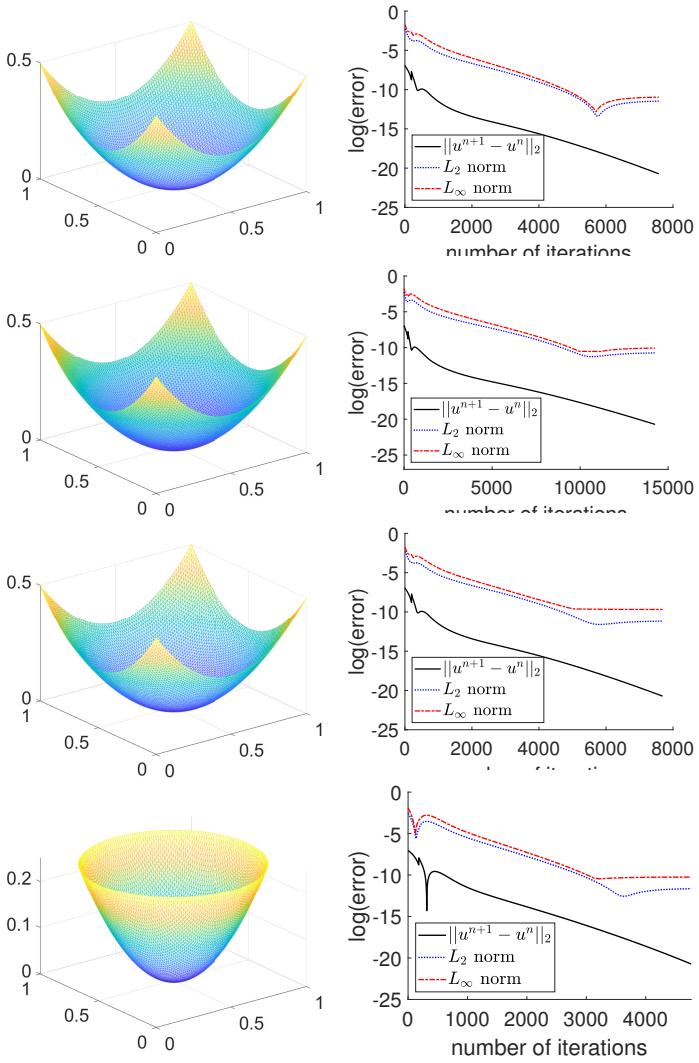

FIG. 2. *(Test problem* (7.1) *with* $\alpha = 1$. *Scheme* (4.6)–(4.8).*) Graphs of the computed solutions and the related convergence history. Row* 1*: Regular triangulation of the unit square. Row* 2*: Symmetric triangulation of the unit square. Row* 3*: Unstructured anisotropic triangulation of the unit square. Row* 4*: Unstructured anisotropic triangulation of a half-unit disk. The second-order derivatives are approximated by* (5.13).

approximation (5.13), the errors based on the approximation (5.19) and (5.20) are larger and the convergence rates are smaller.

For comparison, we also show the results by scheme (4.16)–(4.18) with the second-order derivatives approximated by (5.13). Since the boundary condition is compatible, we use a large $\kappa = 500$. The number of iterations and accuracy orders are shown in Table 3. Its efficiency and accuracy are similar to that of scheme (4.16)–(4.18). If κ goes to infinity, scheme (4.16)–(4.18) has an additional stabilization (diffusion) term which provides larger error but extra robustness—the same as what is observed by comparing Table 1(a) and Table 3.

Since the exact solution is a quadratic function, its second-order derivatives are constants so that the zero Neumann boundary condition on these derivatives is exact.

TABLE 1

(Test problem (7.1) *with* $\alpha = 1$. *Scheme* (4.6)–(4.8).*) Numbers of iterations necessary for convergence, approximation errors, and accuracy orders.* (a) *Regular triangulation of the unit square.* (b) *Symmetric triangulation of the unit square.* (c) *Unstructured anisotropic triangulation of the unit square.* (d) *Unstructured anisotropic triangulation of the half-unit disk. The second-order derivatives are approximated by* (5.13).

	h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
	1/10	193	9.88×10^{-10}	6.06×10^{-4}		9.86×10^{-4}	
(a)	1/20	606	9.81×10^{-10}	1.66×10^{-4}	1.87	2.73×10^{-4}	1.85
	1/40	2064	9.96×10^{-10}	4.34×10^{-5}	1.94	7.11×10^{-5}	1.94
	1/80	7577	9.99×10^{-10}	1.05×10^{-5}	2.05	1.73×10^{-5}	2.04
	h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
	1/10	305	9.99×10^{-10}	1.37×10^{-3}		2.64×10^{-3}	
(b)	1/20	1021	9.99×10^{-10}	3.53×10^{-4}	1.96	6.88×10^{-4}	1.94
	1/40	3961	9.99×10^{-10}	8.98×10^{-5}	1.97	1.75×10^{-4}	1.98
	1/80	14259	9.99×10^{-10}	2.16×10^{-5}	2.06	4.24×10^{-5}	2.05
	h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
	1/10	180	9.38×10^{-10}	5.70×10^{-4}		2.04×10^{-3}	
(c)	1/20	591	9.80×10^{-10}	1.90×10^{-4}	1.59	5.99×10^{-4}	1.77
	1/40	2080	9.97×10^{-10}	5.27×10^{-5}	1.85	1.57×10^{-4}	1.93
	1/80	7690	9.99×10^{-10}	1.42×10^{-5}	1.89	6.22×10^{-5}	1.34
	h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
	1/10	111	8.49×10^{-10}	6.10×10^{-4}		1.20×10^{-3}	
(d)	1/20	374	9.97×10^{-10}	1.65×10^{-4}	1.89	4.55×10^{-4}	1.40
	1/40	1221	9.93×10^{-10}	3.61×10^{-5}	2.20	1.03×10^{-4}	2.14
	1/80	4765	9.97×10^{-10}	8.73×10^{-6}	2.05	3.54×10^{-5}	1.54

TABLE 2

(Test problem (7.1) *with* $\alpha = 1$. *Scheme* (4.6)–(4.8).*) Numbers of iterations necessary for convergence, approximation errors, and accuracy orders.* (a) *Regular triangulation of the unit square.* (b) *Symmetric triangulation of the unit square. The second-order derivatives are approximated by* (5.19)–(5.20).

	h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
	1/10	266	9.50×10^{-8}	1.01×10^{-1}		1.22×10^{-1}	
(a)	1/20	512	9.87×10^{-8}	4.02×10^{-2}	1.33	4.63×10^{-2}	1.40
	1/40	1432	9.99×10^{-8}	1.82×10^{-2}	1.14	2.13×10^{-2}	1.12
	1/80	4529	9.99×10^{-8}	8.73×10^{-3}	1.06	1.03×10^{-2}	1.05
	h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
	1/10	471	9.57×10^{-8}	8.24×10^{-2}		9.74×10^{-2}	
(b)	1/20	782	9.95×10^{-8}	3.46×10^{-2}	1.25	3.97×10^{-2}	1.29
	1/40	2581	9.99×10^{-8}	1.60×10^{-2}	1.11	1.80×10^{-2}	1.14
	1/80	7690	9.99×10^{-8}	7.78×10^{-3}	1.04	8.56×10^{-3}	1.07

TABLE 3

(Test problem (7.1) *with* $\alpha = 1$. *Scheme* (4.16)–(4.18).*) Numbers of iterations necessary for convergence, approximation errors, and accuracy orders. The second-order derivatives are approximated by* (5.13).

h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
1/10	198	9.78×10^{-10}	1.88×10^{-3}		2.75×10^{-3}	
1/20	604	9.80×10^{-10}	3.72×10^{-4}	2.34	5.91×10^{-4}	2.22
1/40	2057	9.92×10^{-10}	8.86×10^{-5}	2.07	1.44×10^{-4}	2.04
1/80	7566	9.99×10^{-10}	2.14×10^{-5}	2.05	3.53×10^{-5}	2.03

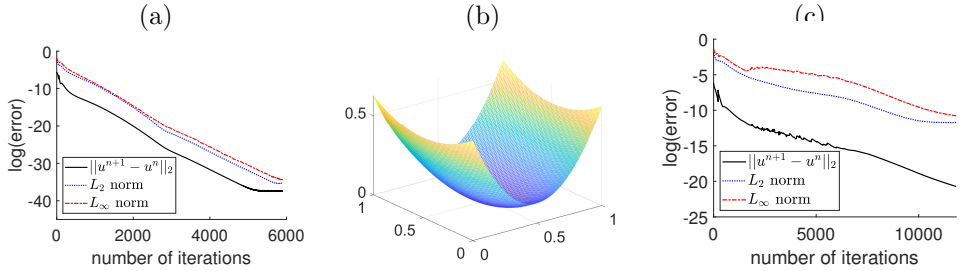

FIG. 3. *(Test problem* (7.1)*. Scheme* (4.6)–(4.8)*.)* (a) *With* $\alpha = 1$, $\varepsilon_1 = \varepsilon_2 = 0$ *in both stages, the convergence history on the regular triangulation of the unit square. The second-order derivatives are approximated by* (5.13). (b)–(c) *The graph of the computed solution and the convergence history on the regular triangulation of the unit square. The second-order derivatives are approximated by* (5.13).

TABLE 4

(Test problem (7.1) *with* $\alpha = 2$*. Scheme* (4.6)–(4.8)*.) Numbers of iterations necessary for convergence, approximation errors, and accuracy orders on the regular triangulation of the unit square. The second-order derivatives are approximated by* (5.13).

h	Iterations	$\|u^{n+1}-u^n\|$	L_2 norm	rate	L_∞ norm	rate
1/10	309	9.67×10^{-10}	5.01×10^{-4}		8.07×10^{-4}	
1/20	938	9.93×10^{-10}	1.32×10^{-4}	1.92	2.12×10^{-4}	1.93
1/40	2982	9.97×10^{-10}	3.38×10^{-5}	2.01	5.39×10^{-5}	1.98
1/80	14565	9.99×10^{-11}	8.51×10^{-6}	1.99	1.36×10^{-5}	1.99

With $\varepsilon = \varepsilon_1 = 0$ and $h = 1/40$, the convergence history of scheme (4.6)–(4.8) is shown in Figure 3(a). We can see that although approximation (5.13) is a kind of variational crime, the error decreases to machine precision.

In the second test, we choose $\alpha = 2$ so that u^* represents a family of concentric ellipses which vary anisotropically. We apply our algorithm to this problem on the unit square with regular meshes. The number of iterations necessary to satisfy the stopping criterion and the corresponding approximation error accuracy is shown in Table 4. The graph of the computed solution and the related convergence history are shown in Figure 3(b)–(c).

7.2. Example 2. In the second example, we consider a problem with the exact solution

$$u = -\sqrt{1 - x_1^2 - x_2^2}, \tag{7.2}$$

which is a part of the unit sphere, and the corresponding Gauss curvature is constant: $K = 1$. The computational domain is chosen to be half of the unit disk, $\Omega = \{(x_1, x_2) \,|\, x_1 \geq 0,\, x_1^2 + x_2^2 \leq 1\}$. Accordingly, the boundary condition is given as

$$g = \begin{cases} 0, & x_1 > 0, \\ -\sqrt{1 - x_2^2}, & x_1 = 0. \end{cases} \tag{7.3}$$

This problem is interesting since the gradient of the exact solution along the boundary where $x_1 > 0$ is infinite (a more challenging problem is solved in the supplementary materials section SM1.3). This problem is also solved in [30]. Since the first-order derivatives are infinite along a part of the boundary, we have to use

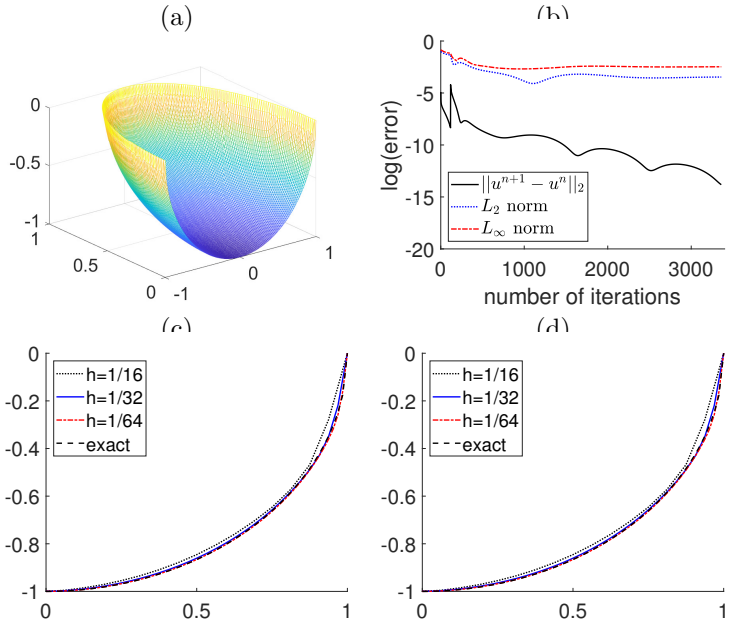

FIG. 4. *(Test problem* (7.2)*. Scheme* (4.6)–(4.8)*)* (a) *The graph of the solution with* $h = 1/64$*, where the second-order derivatives are approximated by* (5.13)*.* (b) *The convergence history of* (a)*.* (c)–(d) *Graphs of the restrictions of the numerical solutions to the line* $x_1 = 0$ *with different* h*'s.* (c) *The second-order derivatives approximated by* (5.13) *and* (d) *the second-order derivatives approximated by* (5.19)–(5.20)*.*

the regularized approximation (5.5) for the first-order derivatives; otherwise our solution will blow up. We use $\varepsilon = h$, $\varepsilon_1 = h^2$, $\Delta t = h^2$, and the stopping criterion $\|u^{n+1} - u^n\|_2 < 10^{-6}$. Figure 4(a)–(b) shows the graph of the numerical solution for $h = 1/64$ and the related convergence history with second-order derivatives approximated by (5.13). The cross sections of the numerical solutions along the boundary $x_1 = 0$ with second-order derivatives approximated by (5.13) or (5.19)–(5.20) are shown in Figure 4(c)–(d), and the convergence of numerical solutions using both approximations is clearly observed.

To further quantify both approximations of second-order derivatives, we show the numbers of iterations, the L_2- and L_∞-errors, and their corresponding rates of convergence in Table 5(a) and (b). In Table 5, we can see that both approximations of the second-order derivatives behave reasonably well. Although the algorithm equipped with approximation (5.19)–(5.20) produces smaller errors than the one equipped with (5.13), the algorithm with approximation (5.13) is more stable as its convergence rate is uniformly about 0.5. As a comparison, we also list the L_∞-errors and related convergence rates from [30] in Table 5(c). When the mesh is fine enough, our algorithm equipped with either approximation produces smaller L_∞-errors than that of [30].

7.3. Example 3. We end this section by considering a problem with no classical solution. The curvature is a constant in Ω:

$$K = 1/2 \text{ in } \Omega, \tag{7.4}$$

where $\Omega = [0,1]^2$. We use the boundary condition $g = 0$ on $\partial\Omega$. This problem has no classical solution since $\det(\mathbf{D}^2 u)$ vanishes on $\partial\Omega$. In other words, this problem is

TABLE 5

(Test problem (7.2)*. Scheme* (4.6)–(4.8)*.) Numbers of iterations, approximation errors, and accuracy orders with the second-order derivatives approximated by* (a) (5.13) *and* (b) (5.19)–(5.20)*.* (c) *shows the* L_∞ *errors and accuracy orders from* [30].

	h	Iteration	$\|u^{n+1}-u^n\|_2$	L_2 error	rate	L_∞ error	rate
(a)	1/16	177	9.57×10^{-7}	9.79×10^{-2}		1.69×10^{-1}	
	1/32	791	9.98×10^{-7}	5.61×10^{-2}	0.80	1.19×10^{-1}	0.51
	1/64	3360	9.97×10^{-7}	3.12×10^{-2}	0.85	8.39×10^{-2}	0.50
	1/128	17273	9.99×10^{-7}	1.55×10^{-2}	1.01	5.86×10^{-2}	0.52
	h	Iteration	$\|u^{n+1}-u^n\|_2$	L_2 error	rate	L_∞ error	rate
(b)	1/16	236	9.72×10^{-7}	2.86×10^{-2}		7.40×10^{-2}	
	1/32	1179	9.98×10^{-7}	1.13×10^{-2}	1.34	4.53×10^{-2}	0.71
	1/64	5261	9.95×10^{-7}	7.33×10^{-3}	0.62	4.12×10^{-2}	0.14

	h	L_∞ error	rate
(c)	1/16	1.61×10^{-1}	
	1/32	1.28×10^{-1}	0.33
	1/64	1.09×10^{-1}	0.23
	1/128	8.80×10^{-2}	0.31

TABLE 6

(Test problem (7.4)*. Scheme* (4.6)–(4.8)*.) Numbers of iterations, iteration errors, and minimum values. The second-order derivatives are approximated by* (a) (5.13) *and* (b) (5.19)–(5.20)*.*

	h	Iter.	$\|u^{n+1}-u^n\|_2$	$\|\mathbf{p}^n-\mathbf{D}^2u^n\|_2$	$\frac{\|\mathbf{p}^n-\mathbf{D}^2u^n\|_2}{\|\mathbf{p}^n\|_2}$	min	$\|\mathbf{p}^n-\mathbf{D}^2u^n\|_2$ in.
(a)	1/20	177	9.62×10^{-7}	4.44×10^{-2}	2.35×10^{-2}	-0.1192	3.28×10^{-3}
	1/40	672	9.72×10^{-7}	1.80×10^{-1}	7.26×10^{-2}	-0.1263	5.98×10^{-3}
	1/80	2149	9.98×10^{-7}	4.94×10^{-1}	1.67×10^{-1}	-0.1305	9.41×10^{-3}
	h	Iter.	$\|u^{n+1}-u^n\|_2$	$\|\mathbf{p}^n-\mathbf{D}^2u^n\|_2$	$\frac{\|\mathbf{p}^n-\mathbf{D}^2u^n\|_2}{\|\mathbf{p}^n\|_2}$	min	$\|\mathbf{p}^n-\mathbf{D}^2u^n\|_2$ in.
(b)	1/20	246	9.88×10^{-7}	4.60×10^{-3}	2.43×10^{-3}	-0.1345	7.47×10^{-4}
	1/40	695	9.88×10^{-7}	7.53×10^{-2}	2.43×10^{-2}	-0.1359	4.77×10^{-3}
	1/80	2468	9.99×10^{-7}	3.52×10^{-1}	6.82×10^{-2}	-0.1376	6.54×10^{-3}

incompatible. In our experiment, we first use scheme (4.6)–(4.8) with $\varepsilon=\varepsilon_1=\varepsilon_2=h^2$ and $\Delta t=2h^2$. The second-order derivatives are approximated by (5.13). The number of iterations, convergence errors, and minimum values are shown in Table 6. The graphs and contour of the numerical solution with $h=1/80$ are shown in Figure 5, row 1. The comparisons of the restriction of the numerical solution with different h along $x_1=1/2$ and $x_1=x_2$ are shown in Figure 5, row 2. Our solution is smooth and almost convex, except for the region near the corners of the domain.

Then we use scheme (4.16)–(4.18) with $\varepsilon=\varepsilon_1=\varepsilon_2=h^2$ and $\Delta t=8h^2$ to solve it. With $h=1/80$, the graph and contour of the computed solution are shown in Figure 5, row 3. We can see that the boundary value of the computed solution is no longer constant. At the middle segment on each edge, its value is away from 0 to be compatible with its interior value. The comparisons of the restriction of the numerical solution with different h along $x_1=1/2$ and $x_1=x_2$ are shown in the fourth row of Figure 5. Compared to the graph in the second row of Figure 5, we observe the deviation of the boundary value from 0. The same problem is solved on an ellipse domain in the supplementary materials section SM1.4.

8. Conclusion. In this work, we have proposed two operator-splitting/mixed finite-element methods to solve the Dirichlet Minkowski problem in dimension two.

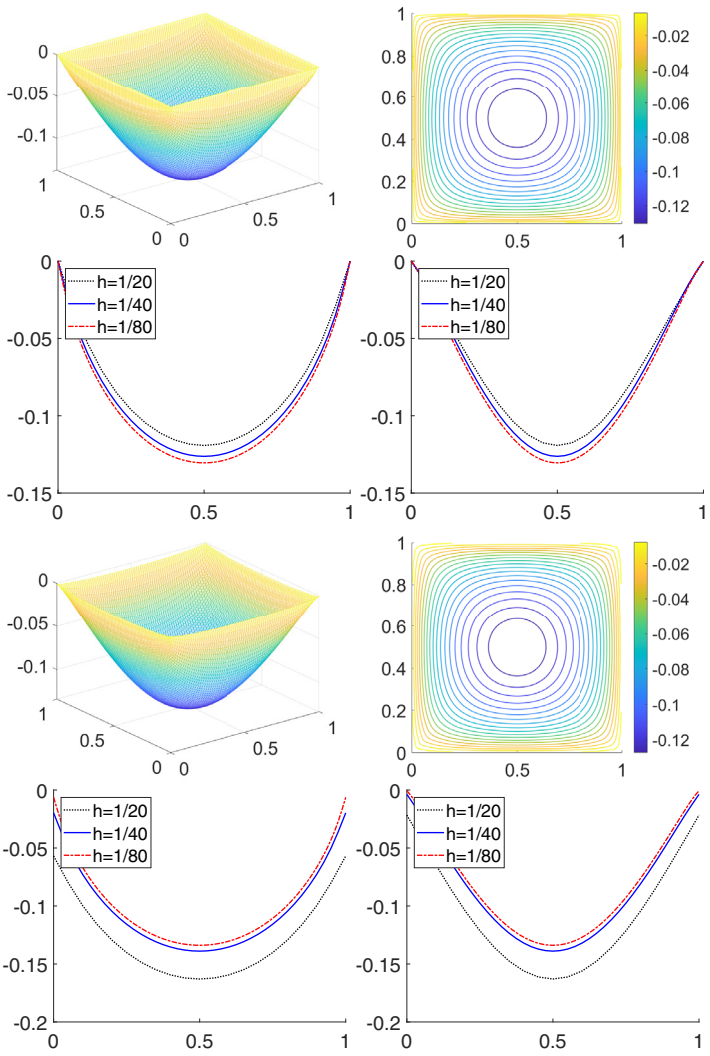

FIG. 5. *(Test problem* (7.4)*.) Rows* 1–2: *Scheme* (4.6)–(4.8). *Row* 1*: Graphs and contour of the numerical solution with* $h = 1/80$. *Row* 2*: Graphs of the restrictions of numerical solutions along* $x_1 = 1/2$ *(left) and* $x_1 = x_2$ *(right) with* $h = 1/20$, $1/40$, *and* $1/80$. *Rows* 3–4: *Scheme* (4.16)–(4.18).) *Row* 3*: Graphs and contour of the numerical solution with* $h = 1/80$. *Row* 4*: Graphs of the restrictions of numerical solutions along* $x_1 = 1/2$ *(left) and* $x_1 = x_2$ *(right) with* $h = 1/20$, $1/40$, *and* $1/80$. *The second-order derivatives are approximated by* (5.13).

Our algorithms are easy to implement since only a system of PDEs is to be solved and the basis functions are chosen to be piecewise linear. When the problem has a classical solution, scheme (4.6)–(4.8) using approximation (5.19)–(5.20) for second-order derivatives is first-order accurate, while using approximation (5.13) it is almost second-order accurate. For an incompatible problem, scheme (4.16)–(4.18) can adjust the boundary value of the computed solution to make it compatible with its interior values. Our algorithm can solve the Minkowski problem on arbitrarily shaped domains and can also solve problems with singularities in the solution gradient. Our algorithm can be easily extended to high dimensions, which constitutes an ongoing work.

Acknowledgment. The preparation of this manuscript has been overshadowed by Prof. Roland Glowinski's passing away. 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.

REFERENCES

[1] G. Awanou, *Standard finite elements for the numerical resolution of the elliptic Monge-Ampère equation: Classical solutions*, IMA J. Numer. Anal., 35 (2014), pp. 1150–1166.

[2] I. Bakelman, *Generalized elliptic solutions of the Dirichlet problem for the Monge-Ampère equations*, in Proc. Symp. Pure Math., Vol. 44, AMS, 1986, pp. 1–30.

[3] I. J. Bakel'Man, *The Dirichlet problem for the elliptic n-dimensional Monge-Ampère equations and related problems in the theory of quasilinear equations*, in Proceedings of Seminar on Monge-Ampère Equations and Related Topics, Firenze, 1980, pp. 1–78.

[4] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numer. Math., 84 (2000), pp. 375–393.

[5] J. D. Benamou, B. D. Froese, and A. M. Oberman, *Two numerical methods for the elliptic Monge–Ampère equation*, ESAIM Math. Model. Numer. Anal., 44 (2010), pp. 737–758.

[6] S. Brenner, T. Gudi, M. Neilan, and L. Sung, *C^0 penalty methods for the fully nonlinear Monge-Ampère equation*, Math. Comp., 80 (2011), pp. 1979–1995.

[7] S. C. Brenner and M. Neilan, *Finite element approximations of the three dimensional Monge-Ampère equation*, ESAIM Math. Model. Numer. Anal., 46 (2012), pp. 979–1001.

[8] A. Caboussat, R. Glowinski, and 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 Control Optim. Calc. Var., 19 (2013), pp. 780–810.

[9] L. Caffarelli and R. Glowinski, *Numerical solution of the Dirichlet problem for a Pucci equation in dimension two. Application to homogenization*, J. Numer. Math., 16 (2008), pp. 185–216.

[10] L. T. Cheng, *Construction of shapes arising from the Minkowski problem using a level set approach*, J. Sci. Comput., 19 (2003), pp. 123–138.

[11] S. Y. Cheng and S. T. Yau, *On the regularity of the solution of the n-dimensional Minkowski problem*, Comm. Pure Appl. Math., 29 (1976), pp. 495–516.

[12] P. G. Ciarlet and P.-A. Raviart, *A mixed finite element method for the biharmonic equation*, in Mathematical Aspects of Finite Elements in Partial Differential Equations, Elsevier, 1974, pp. 125–145.

[13] E. J. Dean and R. Glowinski, *Numerical solution of the two-dimensional elliptic Monge–Ampère equation with Dirichlet boundary conditions: An augmented Lagrangian approach*, C. R. Math., 336 (2003), pp. 779–784.

[14] E. J. Dean and R. Glowinski, *Numerical solution of the two-dimensional elliptic Monge–Ampère equation with Dirichlet boundary conditions: A least-squares approach*, C. R. Math., 339 (2004), pp. 887–892.

[15] E. J. Dean and R. Glowinski, *An augmented Lagrangian approach to the numerical solution of the Dirichlet problem for the elliptic Monge–Ampère equation in two dimensions*, Electron. Trans. Numer. Anal., 22 (2006), pp. 71–96.

[16] E. J. Dean and R. Glowinski, *Numerical methods for fully nonlinear elliptic equations of the Monge–Ampère type*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 1344–1386.

[17] E. J. Dean and R. Glowinski, *On the numerical solution of the elliptic Monge–Ampère equation in dimension two: A least-squares approach*, in Partial Differential Equations, Springer, 2008, pp. 43–63.

[18] E. J. Dean, R. Glowinski, and T. W. Pan, *Operator-splitting methods and applications to the direct numerical simulation of particulate flow and to the solution of the elliptic Monge–Ampère equation*, in Control and Boundary Analysis, CRC Press, 2005, pp. 1–27.

[19] X. Feng, R. Glowinski, and M. Neilan, *Recent developments in numerical methods for fully nonlinear second order partial differential equations*, SIAM Rev., 55 (2013), pp. 205–267, https://doi.org/10.1137/110825960.

[20] B. D. Froese and A. M. Oberman, *Convergent finite difference solvers for viscosity solutions of the elliptic Monge–Ampère equation in dimensions two and higher*, SIAM J. Numer. Anal., 49 (2011), pp. 1692–1714, https://doi.org/10.1137/100803092.

[21] B. D. Froese and A. M. Oberman, *Fast finite difference solvers for singular solutions of the elliptic Monge–Ampère equation*, J. Comput. Phys., 230 (2011), pp. 818–834.

[22] R. Glowinski, *Approximations externes par éléments finis d'ordre un et deux du problème de Dirichlet pour* Δ^2, in Topics in Numerical Analysis, J. J. H. Miller, ed., Academic Press, 1973, pp. 123–171.
[23] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Sci. Comput., Springer, 1984.
[24] R. Glowinski, *Lectures on Numerical Methods for Non-linear Variational Problems*, Springer, 2008.
[25] R. Glowinski, *Numerical methods for fully nonlinear elliptic equations*, in 6th International Congress on Industrial and Applied Mathermatics, ICIAM, Vol. 7, 2009, pp. 155–192.
[26] R. Glowinski, *Variational Methods for the Numerical Solution of Nonlinear Elliptic Problems*, CBMS-NSF Regional Conf. Ser. Appl. Math. 86, SIAM, Philadelphia, 2015.
[27] R. Glowinski, S. Leung, H. Liu, and J. Qian, *On the numerical solution of nonlinear eigenvalue problems for the Monge-Ampère operator*, ESAIM Control Optim. Calc. Var., 26 (2020), 118.
[28] R. Glowinski, H. Liu, S. Leung, and J. Qian, *A finite element/operator-splitting method for the numerical solution of the two dimensional elliptic Monge–Ampère equation*, J. Sci. Comput., 81 (2019), pp. 2271–2302.
[29] R. Glowinski, S. Osher, and W. Yin, *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer, 2016.
[30] B. Hamfeldt, *Convergent approximation of non-continuous surfaces of prescribed Gaussian curvature*, Comm. Pure Appl. Anal., 17 (2018), pp. 671–707.
[31] L. Lamberg, *On the Minkowski Problem and the Lightcurve Operator*, Dissertation, Academia Scientiarum Fennica, Series A. I, 1993.
[32] L. Lamberg and M. Kaasalainen, *Numerical solution of the Minkowski problem*, J. Comput. Appl. Math., 137 (2001), pp. 213–227.
[33] H. Lewy, *A priori limitations for solutions of Monge–Ampère equations*, Trans. Amer. Math. Soc., 37 (1935), pp. 417–434.
[34] H. Lewy, *On differential geometry in the large.* I. *Minkowski's problem*, Trans. Amer. Math. Soc., 43 (1938), pp. 258–270.
[35] P. L. Lions, *Sur les equations de Monge-Ampère*, Arch. Ration. Mech. Anal., 89 (1985), pp. 93–122.
[36] J. J. Little, *An iterative method for reconstructing convex polyhedra from external Guassian images*, in AAAI, 1983, pp. 247–250.
[37] J. J. Little, *Recovering Shape and Determining Attitude from Extended Gaussian Images*, Ph.D. thesis, University of British Columbia, 1985.
[38] H. Liu, R. Glowinski, S. Leung, and J. Qian, *A finite element/operator-splitting method for the numerical solution of the three dimensional Monge–Ampère equation*, J. Sci. Comput., 81 (2019), pp. 2271–2302.
[39] H. Liu, S. Leung, and J. Qian, *An efficient operator-splitting method for the eigenvalue problem of the Monge-Ampère equation*, Commun. Optim. Theory, 2022 (2022), pp. 1–22.
[40] G. Loeper and F. Rapetti, *Numerical solution of the Monge-Ampère equation by a Newton algorithm*, C. R. Acad. Sci. Paris Ser. I, 340 (2005), pp. 319–324.
[41] H. Minkowski, *Allgemeine Lehrsätze über die konvexen Polyeder*, in Ausgewählte Arbeiten zur Zahlentheorie und zur Geometrie, Springer, 1989, pp. 121–139.
[42] H. Minkowski, *Volumen und oberfläche*, in Ausgewählte Arbeiten zur Zahlentheorie und zur Geometrie, Springer, 1989, pp. 146–192.
[43] B. Mohammadi, *Optimal transport, shape optimization and global minimization*, C. R. Math., 344 (2007), pp. 591–596.
[44] L. Nirenberg, *The Weyl and Minkowski problems in differential geometry in the large*, Comm. Pure Appl. Math., 6 (1953), pp. 337–394.
[45] A. V. Pogorelov, *Regularity of a convex surface with given Gaussian curvature*, Mat. Sb., 73 (1952), pp. 88–103.
[46] A. V. Pogorelov, *The Minkowski Multidimensional Problem*, Scripta Ser. Math., John Wiley & Sons, 1978.
[47] H. Schaeffer and T. Y. Hou, *An accelerated method for nonlinear elliptic PDE*, J. Sci. Comput., 69 (2016), pp. 556–580.
[48] N. S. Trudinger and J. I. E. Urbas, *The Dirichlet problem for the equation of prescribed Gauss curvature*, Bull. Austral. Math. Soc., 28 (1983), pp. 217–231.
[49] N. S. Trudinger and X. J. Wang, *The Monge–Ampère equation and its geometric applications*, in Handbook of Geometric Analysis 1, Adv. Lect. Math. 7, International Press, 2008, pp. 467–524.

[50] J. I. E. Urbas, *The equation of prescribed Gauss curvature without boundary conditions*, J. Differential Geom., 20 (1984), pp. 311–327.
[51] J. I. E. Urbas, *Global Hölder estimates for equations of Monge–Ampère type*, Invent. Math., 91 (1988), pp. 1–29.
[52] J. I. E. Urbas, *Boundary regularity for solutions of the equation of prescribed Gauss curvature*, Ann. Inst. H. Poincaré C Anal. Non Linéaire, 8 (1991), pp. 499–522.