INSTITUT DE FRANCE
Académie des sciences

Comptes Rendus

Mécanique

Alexandre Caboussat, Dimitrios Gourzoulidis and Marco Picasso

An adaptive least-squares algorithm for the elliptic Monge–Ampère equation

Volume 351, Special Issue S1 (2023), p. 277-292

Online since: 16 October 2023

Part of Special Issue: The scientific legacy of Roland Glowinski
Guest editors: Grégoire Allaire (CMAP, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, France),
Jean-Michel Coron (Laboratoire Jacques-Louis Lions, Sorbonne Université)
and Vivette Girault (Laboratoire Jacques-Louis Lions, Sorbonne Université)

https://doi.org/10.5802/crmeca.222

Comptes Rendus
Mécanique
2023, Vol. 351, Special Issue S1, p. 277-292
https://doi.org/10.5802/crmeca.222

The scientific legacy of Roland Glowinski / *L'héritage scientifique de Roland Glowinski*

An adaptive least-squares algorithm for the elliptic Monge–Ampère equation

Alexandre Caboussat [*, a], **Dimitrios Gourzoulidis** [b, c] **and Marco Picasso** [c]

[a] Geneva School of Business Administration (HEG-GE), University of Applied Sciences and Arts Western Switzerland (HES-SO), 1227 Carouge, Geneva, Switzerland

[b] Institute of Mathematics, Ecole Polytechnique Fédérale de Lausanne (EPFL) 1018 Lausanne, Switzerland

E-mails: alexandre.caboussat@hesge.ch (A. Caboussat), dimigourz@gmail.com (D. Gourzoulidis), marco.picasso@epfl.ch (M. Picasso)

In memory of Professor Roland Glowinski

Abstract. We address the numerical solution of the Dirichlet problem for the two-dimensional elliptic Monge–Ampère equation using a least-squares/relaxation approach. The relaxation algorithm allows the decoupling of the differential operators from the nonlinearities of the equation, within a splitting approach. The approximation relies on mixed low order finite element methods with regularization techniques. In order to account for data singularities in non-smooth cases, we introduce an adaptive mesh refinement technique. The error indicator is based an independent formulation of the Monge–Ampère equation under divergence form, which allows to explicit a residual term. We show that the error is bounded from above by an a posteriori error indicator plus an extra term that remains to be estimated. This indicator is then used within the existing least-squares framework. The results of numerical experiments support the convergence of our relaxation method to a convex classical solution, if such a solution exists. Otherwise they support convergence to a generalized solution in a least-squares sense. Adaptive mesh refinement proves to be efficient, robust, and accurate to tackle test cases with singularities.

Keywords. Fully nonlinear PDE, Monge–Ampère equation, Least-squares algorithm, Mixed finite elements, Adaptive mesh refinement, Non-smooth problems.

Funding. This work has been partially supported by Swiss National Science Foundation through grant #165785 .

Manuscript received 14 December 2022, accepted 6 September 2023.

1. Introduction

The Monge–Ampère equation is the prototypical example of fully nonlinear elliptic equations [1]. Various approaches have been identified to solve it numerically, relying on finite differences [2,3] or finite element [4,5] approximations.

The Monge–Ampère equation may not have smooth classical solutions, even for smooth data [1]. Therefore, several notions of generalized solutions have been introduced, such as

[*] Corresponding author.

Aleksandrov solutions [6], viscosity solutions [7, 8], or generalized solutions in a least-squares sense [4, 9].

In this article, we consider a two-dimensional bounded domain Ω with a Lipschitz continuous boundary $\partial\Omega$. We look for $u:\Omega\to\mathbb{R}$ solution of the following problem:

$$\det \mathbf{D}^2 u = f \text{ in } \Omega,$$

$$u = g \text{ on } \partial\Omega.$$

The methods discussed in [9] are revisited here for non-smooth problems, typically with non-smooth data. The introduction of such a non-smooth character leads us to the design of adaptive mesh refinement algorithms.

A posteriori error estimates have been discussed, for linear elliptic equations, e.g., in [10–13], and for nonlinear equations, e.g., in [14, 15]. Adaptive mesh refinement algorithms have been described in [15–17]. In particular, adaptive methods for the Monge–Ampère equation have been proposed in [18], based on a Zienkiewicz–Zhu gradient recovery for the a posteriori indicator, and in [19], based on an adaptive finite difference method. Adaptive techniques are also very efficient in the presence of boundary layers or singularities [20]. Since the underlying mathematical problem is fully nonlinear, a posteriori estimates are difficult to derive.

More precisely, we rely on the least-squares approach for the approximation of a solution to the Monge–Ampère equation that is based on existing works [4, 9]. The underlying relaxation algorithm allows to decouple the variational aspects and the local nonlinear operators. The objective of this work is to introduce a novel additional mesh adaptation algorithm to automatically track singularities of the solution.

Following [21–23], the Monge–Ampère operator can be expressed under a divergence form. In two dimensions of space, it reads:

$$\det \mathbf{D}^2 u = \frac{1}{2}\nabla\cdot\left(\operatorname{cof}\left(\mathbf{D}^2 u\right)\nabla u\right),$$

where the cofactor matrix of $\mathbf{D}^2 u$ is defined as

$$\operatorname{cof}\left(\mathbf{D}^2 u\right) = \begin{pmatrix} \dfrac{\partial^2 u}{\partial y^2} & -\dfrac{\partial^2 u}{\partial x \partial y} \\ -\dfrac{\partial^2 u}{\partial x \partial y} & \dfrac{\partial^2 u}{\partial x^2} \end{pmatrix}.$$

Based on the divergence formulation of the Monge–Ampère operator, and on the Galerkin approximation of the variational weak formulation of the corresponding Monge–Ampère equation, we derive a novel a posteriori error indicator. Since this error indicator is not necessarily consistent with the least-squares formulation used to compute the approximate solution, it is expected to be approximate. However, numerical experiments show appropriate convergence properties of the error. Furthermore, even though the effectivity index may depend on the solution, numerical experiments show that it seems to be independent of the mesh discretization, for sufficiently small mesh sizes. An isotropic mesh refinement algorithm based on this posteriori error indicator is then appropriately inserted at some iterations of the relaxation algorithm to track the singularities of the solution.

The structure of this article is as follows. In Section 2, we describe the proposed methodology, and the underlying relaxation algorithm. The mixed finite element discretization is discussed in Section 3, and the adaptive mesh refinement strategy is described in Section 4. The method is applied in Section 5 to the solution of several numerical examples, including examples without a classical exact solution or with non-smooth data.

2. A least-squares algorithm for the elliptic Monge–Ampère equation

We design an adaptive algorithm for the steady state Monge–Ampère equation. We expand the two-dimensional least-squares approach initially derived in [4], but we describe it briefly here first for the sake of clarity.

Let Ω be a bounded convex domain of $\mathbb{R}^2$; we denote by Γ the boundary of Ω. The Dirichlet problem for the elliptic Monge–Ampère equation reads as follows: Find $u : \Omega \to \mathbb{R}$ satisfying

$$\left\{ \begin{array}{rl} \det \mathbf{D}^2 u & = f\,(>0) \text{ in } \Omega, \\ u & = g \text{ on } \Gamma, \end{array} \right. \tag{1}$$

where $\mathbf{D}^2 u = (\frac{\partial^2 u}{\partial x_i \partial x_j})_{1 \le i,j \le 2}$ is the *Hessian* of the unknown function u, and where $f : \Omega \to \mathbb{R}$ and $g : \Gamma \to \mathbb{R}$ are given, sufficiently smooth, functions. We advocate a nonlinear least-squares method for the solution of (1), which relies on the introduction of an additional auxiliary variable $\mathbf{p}$. Namely, we look for:

$$\left(u, \mathbf{p}\right) \in V_g \times \mathbf{Q}_f \;\text{ such that }\; J\left(u, \mathbf{p}\right) \le J(v, \mathbf{q}), \quad \forall\, (v, \mathbf{q}) \in V_g \times \mathbf{Q}_f \tag{2}$$

where:

$$J\left(v, \mathbf{q}\right) = \frac{1}{2} \int_\Omega \left|\mathbf{D}^2 v - \mathbf{q}\right|^2 d\mathbf{x},$$

using the Frobenius norm and inner product defined by $|\mathbf{T}| = \sqrt{\mathbf{T} : \mathbf{T}}$, $\mathbf{S} \,:\, \mathbf{T} = \sum_{i,j=1}^2 s_{ij} t_{ij}$, for all $\mathbf{S} = (s_{ij})$, $\mathbf{T} = (t_{ij}) \in \mathbb{R}^{2\times 2}$. The functional spaces in (2) are respectively defined by:

$$\begin{aligned} V_g &= \left\{ u \in H^2(\Omega),\, u = g \;\text{ on } \Gamma \right\}, \\ \mathbf{Q} &= \left\{ \mathbf{q} \in L^2(\Omega)^{2\times 2},\, \mathbf{q} = \mathbf{q}^t \right\}, \\ \mathbf{Q}_f &= \left\{ \mathbf{q} \in \mathbf{Q}, \det \mathbf{q} = f,\, q_{ii} > 0,\, i = 1,2 \right\}. \end{aligned}$$

For simplicity's sake, we initially describe the problem in $H^2(\Omega)$ where classical solutions exist. Thus, if f and g are sufficiently smooth, the spaces V_g and $\mathbf{Q}_f$ are both non-empty. However, generally speaking, the elliptic Dirichlet Monge–Ampère problem is well-posed in the sense of Aleksandrov [6]. In order to solve (2), and to force the convexity of the solution, we advocate a relaxation algorithm of the Gauss–Seidel-type. The initialization of the algorithm is performed first by solving the linear elliptic problem:

$$\left\{ \begin{array}{ll} \Delta u^0 = 2\sqrt{f} & \text{in } \Omega, \\ u^0 = g & \text{on } \Gamma. \end{array} \right. \tag{3}$$

Then, for $n \ge 0$ and assuming that u^n is known, one successively computes $\mathbf{p}^n, u^{n+1/2}$ and u^{n+1} as follows:

$$\mathbf{p}^n = \arg\min_{\mathbf{q} \in \mathbf{Q}_f} J\left(u^n, \mathbf{q}\right); \tag{4}$$

$$u^{n+1/2} = \arg\min_{v \in V_g} J\left(v, \mathbf{p}^n\right); \tag{5}$$

$$u^{n+1} = u^n + \omega\left(u^{n+1/2} - u^n\right), \tag{6}$$

with $1 \le \omega \le \omega_{\max} < 2$. For the numerical experiments presented in Section 5, we use $\omega \equiv 1$ (unless otherwise specified).

2.1. *Numerical solution of local nonlinear problems*

The minimization problem (4) can be solved point-wise, as it does not involve any derivatives. It leads, a.e. in Ω, to the solution of the following finite dimensional minimization problem

$$\mathbf{p}^n(\mathbf{x}) = \arg \min_{\mathbf{q}\in\mathbf{E}_f(\mathbf{x})} \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{D}^2 u^n(\mathbf{x}) : \mathbf{q} \right], \tag{7}$$

where

$$\mathbf{E}_f(\mathbf{x}) = \left\{ \mathbf{q} \in \mathbb{R}^{2\times 2}, \mathbf{q} = \mathbf{q}^t, \det \mathbf{q} = f(\mathbf{x}), q_{11} > 0, q_{22} > 0 \right\}.$$

2.2. *Numerical solution of linear variational problems*

The minimization problem (5) is a linear variational problem and results in a fourth-order partial differential equation: find $u^{n+1/2} \in V_g$ satisfying

$$\int_\Omega \mathbf{D}^2 u^{n+1/2} : \mathbf{D}^2 v d\mathbf{x} = \int_\Omega \mathbf{p}^n : \mathbf{D}^2 v d\mathbf{x}, \quad \forall\, v \in V_0, \tag{8}$$

where $V_0 = H^2(\Omega) \cap H_0^1(\Omega)$. Following, e.g., [24, Chapter 3], [25, Appendix 1] and [4], we solve (8) using a conjugate gradient algorithm operating in the spaces V_0 and V_g. Even though we first formulate the problem as a conjugate gradient algorithm in infinite dimensions, we discretize it in Section 3 and express (and solve) it only in a finite dimensional framework. Both spaces V_0 and V_g are equipped with the scalar product defined by $(v, w) \to \int_\Omega \Delta v \Delta w d\mathbf{x}$, and the corresponding norm. This algorithm reads as follows:

Step 1.

$$u^0 \in V_g \text{ given.} \tag{9}$$

Step 2.

$$\text{Find } g^0 \in V_0 \text{ such that} \int_\Omega \Delta g^0 \Delta v d\mathbf{x} = \int_\Omega \mathbf{D}^2 u^0 : \mathbf{D}^2 v d\mathbf{x} - \int_\Omega \mathbf{p}^n : \mathbf{D}^2 v d\mathbf{x}, \quad \forall\, v \in V_0, \tag{10}$$

and set the first descent direction:

$$w^0 = g^0. \tag{11}$$

Then, for $k \geq 0$, u^k, g^k, and w^k being known, with g^k and w^k different from zero, we compute u^{k+1}, g^{k+1} and, if necessary, w^{k+1} as follows.

Step 3.

$$\text{Find } \bar{g}^k \in V_0 \text{ such that} \int_\Omega \Delta \bar{g}^k \Delta v d\mathbf{x} = \int_\Omega \mathbf{D}^2 w^k : \mathbf{D}^2 v d\mathbf{x}, \quad \forall\, v \in V_0, \tag{12}$$

and compute the new iterates as follows:

$$\rho_k = \frac{\displaystyle\int_\Omega \left|\Delta g^k\right|^2 d\mathbf{x}}{\displaystyle\int_\Omega \Delta \bar{g}^k \Delta w^k d\mathbf{x}}, \tag{13}$$

$$u^{k+1} = u^k - \rho_k w^k, \tag{14}$$

$$g^{k+1} = g^k - \rho_k \bar{g}^k. \tag{15}$$

Step 4.

$$\text{Compute } \delta_k = \frac{\displaystyle\int_\Omega \left|\Delta g^{k+1}\right|^2 d\mathbf{x}}{\displaystyle\int_\Omega \left|\Delta g^{0}\right|^2 d\mathbf{x}}. \tag{16}$$

If $\delta_k < \varepsilon$ (meaning that the residual is small enough), take $u = u^{k+1}$; otherwise, compute:

$$\gamma_k = \frac{\displaystyle\int_\Omega \left|\Delta g^{k+1}\right|^2 d\mathbf{x}}{\displaystyle\int_\Omega \left|\Delta g^{k}\right|^2 d\mathbf{x}}, \tag{17}$$

and update the descent direction via

$$w^{k+1} = g^{k+1} + \gamma_k w^k. \tag{18}$$

Step 5. Do $k+1 \to k$ and return to Step 3.

3. Mixed Finite Element Approximation

We use a mixed finite element approximation with low order (piecewise linear and globally continuous) finite elements on a partition of Ω made of triangles.

For simplicity, let us assume that Ω is a bounded polyhedral domain of $\mathbb{R}^2$, and define $\mathcal{T}_h$ as a finite element partition of Ω made out of triangles (see, e.g., [25, Appendix 1]). Let Σ_h be the set of the vertices of $\mathcal{T}_h$, $\Sigma_{0h} = \{P \in \Sigma_h, P \notin \Gamma\}$, $N_h = \text{Card}(\Sigma_h)$, and $N_{0h} = \text{Card}(\Sigma_{0h})$, such that $\dim V_h = N_h$ and $\dim V_{0h} = N_{0h}$. We suppose that $\Sigma_{0h} = \{P_j\}_{j=1}^{N_{0h}}$ and $\Sigma_h = \Sigma_{0h} \cup \{P_j\}_{j=N_{0h}+1}^{N_h}$. From $\mathcal{T}_h$, we approximate the spaces $L^2(\Omega)$, $H^1(\Omega)$ and $H^2(\Omega)$ by the finite dimensional space V_h defined by:

$$V_h = \left\{v \in C^0(\bar{\Omega}),\ v|_T \in \mathbb{P}_1,\ \forall\ T \in \mathcal{T}_h\right\},$$

with $\mathbb{P}_1$ the space of the two-variable polynomials of degree ≤ 1. We define also V_{0h} as

$$V_{0h} = \{v \in V_h,\ v = 0 \text{ on } \Gamma\}.$$

In the sequel, V_{0h} will be used to approximate both $H_0^1(\Omega)$ and $H^2(\Omega) \cap H_0^1(\Omega)$. Mixed finite element methods have been widely used for elliptic problems, see, e.g., [26, 27], and in particular for the Monge–Ampère equation [28, 29]. It allows to use a continuous piecewise linear approximation of the function, and also a continuous piecewise linear approximation of its second derivatives. Furthermore, we define the discrete analogues of spaces $\mathbf{Q}$ and $\mathbf{Q}_f$ as follows:

$$\mathbf{Q}_h = \left\{\mathbf{q}_h \in (V_h)^{2\times 2},\ \mathbf{q}_h(P_k) = \mathbf{q}_h^t(P_k),\ k = 1, \ldots, N_h\right\},$$
$$\mathbf{Q}_{fh} = \left\{\mathbf{q}_h \in \mathbf{Q}_h,\ \det \mathbf{q}_h(P_k) = f_h(P_k),\ \mathbf{q}_{h,ii}(P_k) > 0,\ k = 1, \ldots, N_{0h}\right\}.$$

We associate with V_h (or V_{0h} and V_{gh}) and $\mathbf{Q}_h$, the discrete inner products: $(v, w)_{0h} = \frac{1}{3}\sum_{k=1}^{N_h} A_k v(P_k) w(P_k)$ (with corresponding norm $\|v\|_{0h} = \sqrt{(v, v)_h}$), for all $v, w \in V_{0h}$, and $((\mathbf{S}, \mathbf{T}))_{0h} = \frac{1}{3}\sum_{k=1}^{N_h} A_k \mathbf{S}(P_k) : \mathbf{T}(P_k)$ (with corresponding norm $|||\mathbf{S}|||_{0h} = \sqrt{((\mathbf{S}, \mathbf{S}))_{0h}}$) for all $\mathbf{S}, \mathbf{T} \in \mathbf{Q}_h$, where A_k is the volume of the polygonal domain which is the union of those triangles of $\mathcal{T}_h$ which have P_k as a common vertex.

3.1. *Finite Element Approximation of the Monge–Ampère Equation*

When solving (8) by the conjugate gradient algorithm (9)-(18), one has to i) compute the discrete analogues of the second order derivatives, e.g., $\mathbf{D}^2 w^k$ and $\mathbf{D}^2 u^0$, and ii) solve biharmonic problems such as (10) (12).

In order to compute the discrete analogues of the second order derivatives, we rely on the same smoothing approach, a *Tychonoff-like regularization* [30], as the one presented in [4, 9]. More precisely, for a function φ being given in $H^2(\Omega)$, we denote $\partial^2\varphi/\partial x_i \partial x_j$ by $D^2_{ij}(\varphi)$. Then

$$\int_\Omega D^2_{ij}(\varphi) v d\mathbf{x} = -\frac{1}{2}\int_\Omega \left[\frac{\partial \varphi}{\partial x_i}\frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j}\frac{\partial v}{\partial x_i} \right] d\mathbf{x},\ \forall\ v \in H^1_0(\Omega),\ \forall\ i,j = 1,2. \tag{19}$$

Consider now $\varphi \in V_h$. Using (19), we define the discrete analogues of the differential operators D^2_{ij} by

$$D^2_{hij}(\varphi) \in V_{0h},\quad \int_\Omega D^2_{hij}(\varphi) v d\mathbf{x} = -\frac{1}{2}\int_\Omega \left[\frac{\partial \varphi}{\partial x_i}\frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j}\frac{\partial v}{\partial x_i} \right] d\mathbf{x},\ \forall\ v \in V_{0h},\ \forall\ i,j = 1,2. \tag{20}$$

For the sake of simplification, it is also possible to use the trapezoidal rule to evaluate the integrals in the left hand sides of (20). When using piecewise linear mixed finite elements, the a priori estimates for the error on the second derivatives of the solution φ are, in general, $\mathcal{O}(1)$ in the L^2-norm [31]. Therefore the convergence (or non-convergence) properties of the method depend strongly on the type of triangulations. Using a Tychonoff-like regularization [30] allows to keep a mixed finite element approach based on continuous piecewise linear approximations, while obtaining better convergence properties for the discrete second order derivatives. Practically, with $\alpha > 0$ (small) and $|K| = \text{meas}(K)$, we compute the discrete second derivatives $D^2_{hij}(\varphi)$ as follows:

Find $D^2_{hij}(\varphi) \in V_{0h}$ such that, $\forall\ v \in V_{0h},\ i,j = 1,2,$

$$\int_\Omega D^2_{hij}(\varphi) v d\mathbf{x} + \alpha \sum_{K \in \mathcal{T}_h} |K| \int_K \nabla D^2_{hij}(\varphi) \cdot \nabla v d\mathbf{x} = -\frac{1}{2}\int_\Omega \left[\frac{\partial \varphi}{\partial x_i}\frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j}\frac{\partial v}{\partial x_i} \right] d\mathbf{x}, \tag{21}$$

Numerical experiments have shown that this regularization procedure provides approximations of optimal or nearly optimal orders.

In order to solve the biharmonic problems, let us denote the *discrete Laplacian operator* by the operator $\Delta_h \in \mathcal{L}(V_{0h}, V_{0h})$ defined by

$$\Delta_h u \in V_{0h},\ -\int_\Omega \Delta_h u\, v d\mathbf{x} = \int_\Omega \nabla u \cdot \nabla v d\mathbf{x} \quad \forall\ (u,v) \in V_{0h} \times V_{0h}.$$

We observe that the solution of the discrete versions of the bi-harmonic problems (10) and (12), after space discretization, are all particular cases of the following:

$$\text{Find } r \in V_{0h} \text{ such that } \int_\Omega \Delta_h r \Delta_h v d\mathbf{x} = \Lambda_h(v), \quad \forall\ v \in V_{0h}, \tag{22}$$

where $\Lambda_h \in \mathcal{L}(V_{0h}, \mathbb{R})$ defined by

$$\Lambda_h(v) = \begin{cases} \displaystyle\int_\Omega \mathbf{D}^2_h u^0 : \mathbf{D}^2_h v d\mathbf{x} - \int_\Omega \mathbf{p}^n : \mathbf{D}^2_h v d\mathbf{x} & \text{when solving (10)}, \\ \displaystyle\int_\Omega \mathbf{D}^2 w^k : \mathbf{D}^2 v d\mathbf{x} & \text{when solving (12)}. \end{cases}$$

It follows from this definition, and by formally setting $-\Delta_h r = \omega$, that (22) is equivalent to the following system of discrete Poisson–Dirichlet problems:

$$\begin{aligned}
&\text{Find } \omega \in V_{0h}, \quad \int_\Omega \nabla\omega \cdot \nabla v \, d\mathbf{x} = \Lambda_h(v), \quad \forall\, v \in V_{0h},\\
&\text{Find } r \in V_{0h}, \quad \int_\Omega \nabla r \cdot \nabla v \, d\mathbf{x} = \int_\Omega \omega\, v \, d\mathbf{x}, \quad \forall\, v \in V_{0h}.
\end{aligned}$$

4. Adaptive mesh refinement algorithm

We want to derive an error indicator η^I and build an adaptive mesh such that the estimated relative error is close to a preset tolerance TOL, namely

$$0.5\,\mathrm{TOL} \le \frac{\eta^I}{\|\nabla u_h\|_{L^2(\Omega)}} \le 1.5\,\mathrm{TOL},$$

where u_h is the mixed finite element approximation of the least-squares solution described in the previous section. There are many options in the literature for estimating η^I. On the one hand, a posteriori error estimates for fourth-order elliptic problems such as (8) exist but need high-order finite element approximations [32]. On the other hand, the biharmonic problems (10) (12) are solved several times within the conjugate gradient algorithm used to solve (8). A posteriori and a priori error estimates for (10) (12) can be obtained, but high-order finite element approximations are again needed [33].

The strategy we adopt here for adaptive mesh refinement is the following. The least-squares framework introduced in [4,9], and briefly summarized above, allows to compute a solution to the Monge–Ampère equation in a weak sense. However, it does not provide a variational framework to derive a posteriori error estimates. Therefore, to derive such estimates, we use the Monge–Ampère operator under a divergence form

$$\det \mathbf{D}^2 u = \frac{1}{2}\nabla\cdot\left(\mathrm{cof}\left(\mathbf{D}^2 u\right)\nabla u\right). \tag{23}$$

The derivation of an algorithm based on a variational version of this re-formulation would lead to an approximation of the solution that is different from the solution obtained with the least-squares approach. However, this re-formulation allows to derive a posteriori error estimates. The strategy is thus to derive below an error indicator η^I from the equation (23) under divergence form, and insert it into the least-squares framework. By doing so knowingly, we understand that the error indicator is simplified, not necessarily optimal, nor necessarily consistent. However, numerical results will show that it has appropriate convergence properties.

Let us re-write the Monge–Ampère equation using the expression (23): find $u : \Omega \to \mathbb{R}$ such that

$$\begin{cases}
\frac{1}{2}\nabla\cdot\left(\mathrm{cof}\left(\mathbf{D}^2 u\right)\nabla u\right) = f & \text{in } \Omega,\\
u = g & \text{on } \partial\Omega.
\end{cases}$$

The weak formulation of this equation reads as: find $u \in V_g$ satisfying

$$\frac{1}{2}\int_\Omega \mathrm{cof}\left(\mathbf{D}^2 u\right)\nabla u \cdot \nabla v \, d\mathbf{x} = \int_\Omega -f v \, d\mathbf{x}, \quad \forall\, v \in H_0^1(\Omega),$$

while a finite element approximation reads as: find $u_h \in V_{gh}$ satisfying

$$\frac{1}{2}\int_\Omega \mathrm{cof}\left(\mathbf{D}^2 u_h\right)\nabla u_h \cdot \nabla v_h \, d\mathbf{x} = \int_\Omega -f v_h \, d\mathbf{x}, \quad \forall\, v_h \in V_{0h}.$$

Note that the solution u_h to this equation is not the same as the least-squares solution. To derive an a posteriori error estimate, we re-express the residual based on this formulation:

$$\frac{1}{2}\int_\Omega \left(\operatorname{cof}\left(\mathbf{D}^2 u\right)\nabla u - \operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\nabla u_h\right)\cdot\nabla(u-u_h)\,d\mathbf{x}$$
$$= -\int_\Omega f\,(u-u_h)\,d\mathbf{x} - \frac{1}{2}\int_\Omega \operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\nabla u_h\cdot\nabla(u-u_h)\,d\mathbf{x}. \quad (24)$$

The right-hand side of (24) is denoted by $R(u-u_h) := -\int_\Omega f(u-u_h)d\mathbf{x} - \frac{1}{2}\int_\Omega \operatorname{cof}(\mathbf{D}_h^2 u_h)\nabla u_h\cdot\nabla(u-u_h)d\mathbf{x}$. Following [13], an upper bound for $R(u-u_h)$ is given by:

$$R(u-u_h) \le c\left(\sum_{K\in\mathcal{T}_h}\left(h_K^2\left\|-f+\frac{1}{2}\nabla\cdot\left(\operatorname{cof}(\mathbf{D}_h^2 u_h)\nabla u_h\right)\right\|_{L^2(K)}^2 + \frac{1}{16}h_K\left\|\left[\operatorname{cof}(\mathbf{D}_h^2 u_h)\nabla u_h\cdot\mathbf{n}\right]\right\|_{L^2(\partial K)}^2\right)\right)^{\frac{1}{2}}\|\nabla(u-u_h)\|_{L^2(\Omega)},$$

where c is a constant that depends the mesh aspect ratio. Note that the discrete second order derivatives $\mathbf{D}_h^2 u_h$ are defined as in (21), see [4]. We denote by η^I the error indicator given by

$$\eta^I = \left(\sum_{K\in\mathcal{T}_h}\left(\eta_K^I\right)^2\right)^{\frac{1}{2}}, \quad (25)$$

where

$$\left(\eta_K^I\right)^2 = h_K^2\left\|-f+\frac{1}{2}\nabla\cdot\left(\operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\nabla u_h\right)\right\|_{L^2(K)}^2 + \frac{1}{16}h_K\left\|\left[\operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\nabla u_h\cdot\mathbf{n}\right]\right\|_{L^2(\partial K)}^2. \quad (26)$$

This error indicator, derived from the divergence form of the Monge–Ampère equation, is thus used within the least-squares framework presented earlier. If we plug (23) in (26), we get

$$\left(\eta_K^I\right)^2 = h_K^2\left\|-f+\det\mathbf{D}_h^2 u_h\right\|_{L^2(K)}^2 + \frac{1}{16}h_K\left\|\left[\operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\nabla u_h\cdot\mathbf{n}\right]\right\|_{L^2(\partial K)}^2.$$

This provides us with an upper bound for the residual in terms of η^I:

$$R(u-u_h) \le c\left(\sum_{K\in\mathcal{T}_h}\left(\eta_K^I\right)^2\right)^{\frac{1}{2}}\|\nabla(u-u_h)\|_{L^2(\Omega)}. \quad (27)$$

Next, we find a lower bound for the residual. Another expression of the latter can be obtained by adding and subtracting $\operatorname{cof}(\mathbf{D}^2 u)\nabla u_h$ in (24) and proceeding as follows:

$$\frac{1}{2}\int_\Omega \left(\operatorname{cof}\left(\mathbf{D}^2 u\right)\nabla u - \operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\nabla u_h\right)\cdot\nabla(u-u_h)\,d\mathbf{x}$$
$$= \frac{1}{2}\int_\Omega \operatorname{cof}\left(\mathbf{D}^2 u\right)\nabla(u-u_h)\cdot\nabla(u-u_h) + \frac{1}{2}\int_\Omega\left(\operatorname{cof}\left(\mathbf{D}^2 u\right)-\operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\right)\nabla u_h\cdot\nabla(u-u_h)\,d\mathbf{x}.$$

A lower bound for this term is the sum of the two lower bounds of the terms on the right-hand side. Assuming that u is strictly convex, which implies that $\operatorname{cof}(\mathbf{D}^2 u)$ is symmetric positive definite, we obtain

$$\frac{1}{2}\int_\Omega \left(\operatorname{cof}\left(\mathbf{D}^2 u\right)\nabla u - \operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\nabla u_h\right)\cdot\nabla(u-u_h)\,d\mathbf{x}$$
$$\ge \frac{\lambda_1}{2}\|\nabla(u-u_h)\|_{L^2(\Omega)}^2 - \frac{1}{2}\left\|\left(\operatorname{cof}\left(\mathbf{D}^2 u\right)-\operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\right)\nabla u_h\right\|_{L^2(\Omega)}\|\nabla(u-u_h)\|_{L^2(\Omega)},$$

where λ_1 is the smallest eigenvalue of $\operatorname{cof}(\mathbf{D}^2 u)$. Finally, (24) and (27) are inserted into the above inequality, to get

$$\frac{\lambda_1}{2}\|\nabla(u-u_h)\|_{L^2(\Omega)} \le c\left(\sum_{K\in\mathcal{T}_h}(\eta_K^I)^2\right)^{\frac{1}{2}} + \frac{1}{2}\left\|\left(\operatorname{cof}(\mathbf{D}^2 u)-\operatorname{cof}(\mathbf{D}_h^2 u_h)\right)\nabla u_h\right\|_{L^2(\Omega)}. \tag{28}$$

We denote the second term of (28) by

$$\gamma := \frac{1}{2}\left\|\left(\operatorname{cof}\left(\mathbf{D}^2 u\right)-\operatorname{cof}\left(\mathbf{D}_h^2 u_h\right)\right)\nabla u_h\right\|_{L^2(\Omega)},$$

which is used in the next section to illustrate the numerical experiements. The decision of mesh refinement in the adaptive algorithm is based on the first term of the right-hand side of (28), i.e. $(\sum_{K\in\mathcal{T}_h}(\eta_K^I)^2)^{\frac{1}{2}}$. In [31], a $\mathbb{P}_1$ finite element method is considered and the second derivatives are computed in a similar way. An order of convergence $\mathcal{O}(h)$ for the H^2 error semi-norm can be obtained for various mesh types. The second term γ in (28) is therefore expected to have a similar rate of convergence of $\mathcal{O}(h)$, but, in special cases, it can even be $\mathcal{O}(h^2)$ (see [31, Remark 2.1]).

In conclusion, the discrete relaxation algorithm now reads as follows. Let us define

$$J_h\left(v_h,\mathbf{q}_h\right) = \frac{1}{2}\left|\left|\left|\mathbf{D}_h^2 v_h - \mathbf{q}_h\right|\right|\right|_{0h}^2.$$

First find $u_h^0 \in V_{gh}$ such that

$$\int_{\Omega_h} \nabla u_h^0 \cdot \nabla v_h d\mathbf{x} = -\left(2\sqrt{f_h}, v_h\right)_{0h}, \quad \forall\, v_h \in V_{0h}.$$

Then, for $n \ge 0$, and assuming that u_h^n is known:

(1) Compute $\mathbf{p}_h^n$:

$$\mathbf{p}_h^n = \arg\min_{\mathbf{q}_h\in\mathbf{Q}_{fh}} J_h\left(u_h^n,\mathbf{q}_h\right);$$

(2) Compute $u_h^{n+1/2}$:

$$u_h^{n+1/2} = \arg\min_{v_h\in V_{gh}} J_h\left(v_h,\mathbf{p}_h^n\right);$$

(3) Compute u_h^{n+1} (with $1 \le \omega \le \omega_{\max} < 2$):

$$u_h^{n+1} = u_h^n + \omega\left(u_h^{n+1/2} - u_h^n\right);$$

(4) At some iterations, adapt/refine the mesh based on the indicator η^I; the frequency of adaptation is discussed in the numerical experiments section.

5. Numerical Experiments

We present several numerical experiments to study the efficiency of the indicator η^I. For all test cases, the computational domain is the unit square $\Omega = (0,1)^2$. The parameter ω is initially chosen to be one and gradually increases to two, and $\alpha = 1$. The BL2D mesh generator [34] is used to reconstruct an adapted mesh at each of the given iterations. We consider a structured asymmetric mesh of size $h_K = 1/20$. We perform the first mesh refinement when the condition $\|u_h^{n+1} - u_h^n\|_{L_2(\Omega)} \le \nu$ is satisfied. Unless specified otherwise, $\nu = 5\cdot 10^{-5}$. This condition is usually satisfied in less than 100 iterations. Then we adapt the mesh every 50 iterations or if the condition is satisfied. The algorithm stop $h_{\min} = \min_{K\in\mathcal{T}_h} h_K$ and $h_{\max} = \max_{K\in\mathcal{T}_h} h_K$. In order to prevent the algorithm from adapting the mesh infinitely, we enforce $(h_{\max}/h_{\min}) \le 40$.

5.1. *Smooth exponential example*

First we consider a smooth example: find $u : \Omega \to \mathbb{R}$ satisfying

$$\begin{cases} \det \mathbf{D}^2 u = \left(x^2 + y^2 + 1\right) e^{(x^2+y^2)} & \text{in } \Omega, \\ u = e^{\frac{1}{2}(x^2+y^2)} & \text{on } \partial\Omega. \end{cases}$$

The convex solution is defined by $u(x, y) = e^{\frac{1}{2}(x^2+y^2)}$, for all $(x, y) \in \Omega$, and the eigenvalues of its Hessian are given by

$$\lambda_1 = e^{\frac{1}{2}(x^2+y^2)}, \qquad \lambda_2 = e^{\frac{1}{2}(x^2+y^2)}\left(1 + x^2 + y^2\right).$$

We use a triangular unstructured mesh and we set $\nu = 10^{-7}$. For this first test case, we do not consider a mesh adaptation, but we vary h_K uniformly, in order to study the behavior of the error indicator η^I for smooth solutions and unstructured meshes. Table 1 shows that the H^2 error norm, the estimation of γ, and the error indicator η^I converge with approximate order $\mathcal{O}(h)$. The effectivity index settles at a value around 14.

Table 1. Smooth exponential example with uniformly refined unstructured mesh and without mesh adaptation. Estimated errors of $u - u_h$ in H^1 error norm, values of the indicator η^I and the corresponding convergence orders for various h_K.

h_K	η^I	rate	$\dfrac{\eta^I}{\lvert u-u_h\rvert_{H^1(\Omega)}}$	$\lvert u-u_h\rvert_{H^2(\Omega)}$	rate	γ	rate	iterations
0.03125	$4.90 \cdot 10^{-1}$	-	9.73	$5.96 \cdot 10^{-1}$	-	$3.21 \cdot 10^{-1}$		15
0.01561	$2.77 \cdot 10^{-1}$	0.82	12.55	$3.44 \cdot 10^{-1}$	0.79	$2.18 \cdot 10^{-1}$	0.56	16
0.01035	$1.84 \cdot 10^{-1}$	1.00	13.43	$1.91 \cdot 10^{-1}$	1.45	$1.39 \cdot 10^{-1}$	1.10	16
0.00776	$1.38 \cdot 10^{-1}$	1.00	14.07	$1.47 \cdot 10^{-1}$	0.91	$1.05 \cdot 10^{-1}$	0.97	15

5.2. *Non-smooth example with an exact solution*

Let us consider problems involving various types of non-smoothness in the data or the solution. The first problem reads as: find $u : \Omega \to \mathbb{R}$ satisfying

$$\begin{cases} \det \mathbf{D}^2 u = \dfrac{R^2}{\left(R^2 - (x-0.5)^2 - (y-0.5)^2\right)^2} & \text{in } \Omega, \\ u = -\sqrt{R^2 - (x-0.5)^2 - (y-0.5)^2} & \text{on } \partial\Omega, \end{cases} \tag{29}$$

where $R = 1/\sqrt{2}$. The exact solution u of problem (29) is given by

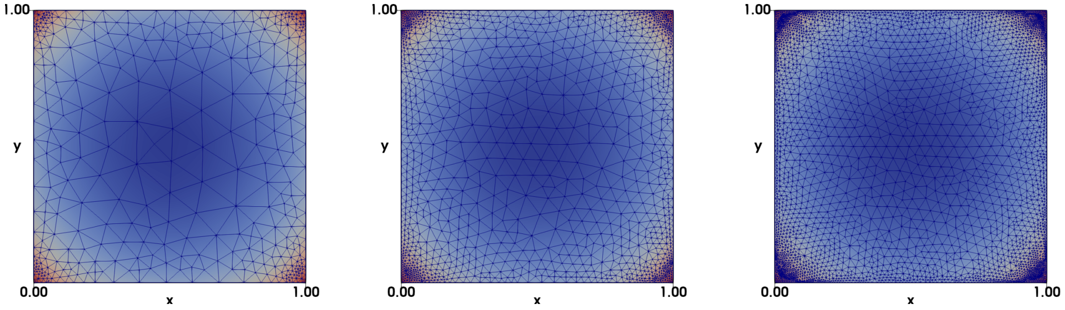
$$u(x, y) = -\sqrt{R^2 - (x-0.5)^2 - (y-0.5)^2}, \quad \forall\, (x, y) \in \overline{\Omega}.$$

The eigenvalues of its Hessian are given by

$$\lambda_1 = \frac{R^2}{\left(R^2 - (x-0.5)^2 - (y-0.5)^2\right)^{3/2}}, \qquad \lambda_2 = \frac{\left(2R^2 - 2x^2 - 2y^2 + 2x + 2y - 1\right)}{2\left(R^2 - (x-0.5)^2 - (y-0.5)^2\right)^{3/2}}.$$

The solution u to (29) is smooth in $\overline{\Omega}$ but ∇u is not defined in the four corners of $\overline{\Omega}$. Due to the lack of regularity of the solution, the adaptive mesh refinement algorithm will track those discontinuities and refine the mesh in the corners. Figure 1 illustrates the refined meshes for various tolerances.

Numerical results are displayed in Table 2. When the tolerance decreases, $h_{\min}$, $h_{\max}$ and the L^2 error norm decrease while the number of elements and vertices increase as expected. The effectivity index $\eta^I / |u - u_h|_{H^1}$ stabilizes at a value close to 10 for all TOL.



(a) TOL = 1.0 **(b)** TOL = 0.5 **(c)** TOL = 0.25

Figure 1. Non-smooth example with data $f(x,y) = \frac{R^2}{(R^2-(x-0.5)^2-(y-0.5)^2)^2}$, $g(x,y) = -\sqrt{R^2-(x-0.5)^2-(y-0.5)^2}$ and $R = \frac{1}{\sqrt{2}}$. Graphs of the final adapted mesh for various values of TOL and after 500 iterations.

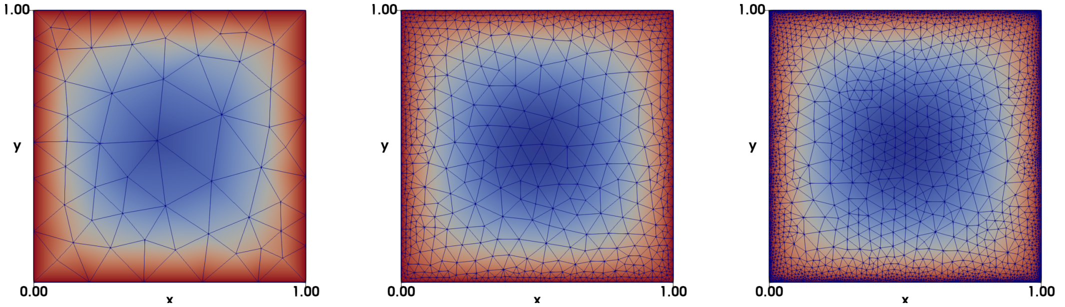
Table 2. Non-smooth example with data $f(x,y) = \frac{R^2}{(R^2-(x-0.5)^2-(y-0.5)^2)^2}$, $g(x,y) = -\sqrt{R^2-(x-0.5)^2-(y-0.5)^2}$ and $R = \frac{1}{\sqrt{2}}$. Convergence behavior of the algorithm for various values of parameter TOL, after 500 iterations. The columns contain the final minimal and maximal mesh size, the final numbers of elements and vertices, the value of the indicator, the H^1 and L^2 error norms, and the effectivity index.

TOL	$h_{\min}$	$h_{\max}$	# elem	# vertices	η^I	$\|u-u_h\|_{H^1}$	$\frac{\eta^I}{\|u-u_h\|_{H^1}}$	$\|u-u_h\|_{L^2}$
1.0	$4.81\cdot10^{-3}$	$1.89\cdot10^{-1}$	825	471	$1.31\cdot10^{0}$	$1.23\cdot10^{-1}$	10.65	$2.21\cdot10^{-2}$
0.5	$3.41\cdot10^{-3}$	$9.33\cdot10^{-2}$	2565	1399	$0.92\cdot10^{0}$	$9.32\cdot10^{-2}$	9.87	$6.49\cdot10^{-3}$
0.25	$1.74\cdot10^{-3}$	$6.41\cdot10^{-2}$	7281	3862	$0.81\cdot10^{0}$	$8.84\cdot10^{-2}$	9.16	$3.48\cdot10^{-3}$

5.3. *Non-smooth example without an exact solution*

The second non-smooth problem that we consider reads as follows: find $u : \Omega \to \mathbb{R}$ satisfying

$$\begin{cases} \det \mathbf{D}^2 u = 1 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{30}$$



(a) TOL = 1.0 **(b)** TOL = 0.5 **(c)** TOL = 0.25

Figure 2. Non-smooth example with data $f(x,y) = 1$ and $g(x,y) = 0$. Graphs of the final adapted mesh for various values of TOL, after 500 iterations.

Despite the smooth data, this problem does not admit a classical solution. As mentioned in [35], the main difficulties, e.g., the lack of convexity, occur near the boundary of the domain. Figure 2 shows the refined mesh for several values of the tolerance. For a tolerance level smaller than 0.5, the algorithm successfully refines the mesh around the boundary with a coarser mesh in the center of the domain. Table 3 shows similar numerical results as the previous test case. Observations made about $h_{\min}$, $h_{\max}$, the number of elements and vertices still holds. However, the indicator η^I does not decrease for decreasing TOL. A possible explanation could come from the bound we set for refinement $\frac{h_{\max}}{h_{\min}} \le 40$, which prevents η^I from decreasing as expected.

Table 3. Non-smooth example with data $f(x,y)=1$ and $g(x,y)=0$. Convergence behavior of the algorithm for various values of parameter TOL, after 500 iterations. The columns contain the final minimal and maximal mesh size, the final numbers of elements and vertices, the value of the indicator, and norm of the minimal solution u_h.

TOL	$h_{\min}$	$h_{\max}$	# elem	# vertices	η^I	$\|u_h\|_{L^\infty(\Omega)}$
1.0	$7.13\cdot10^{-2}$	$2.55\cdot10^{-1}$	110	69	$4.02\cdot10^{-1}$	$1.66722\cdot10^{-1}$
0.5	$1.03\cdot10^{-2}$	$5.90\cdot10^{-1}$	1189	663	$1.96\cdot10^{-1}$	$1.67842\cdot10^{-1}$
0.25	$1.33\cdot10^{-3}$	$7.41\cdot10^{-2}$	5516	3059	$2.72\cdot10^{-1}$	$1.74659\cdot10^{-1}$

5.4. *Numerical experiment with data involving a Dirac function*

The next test problem reads as follows: find $u:\Omega\to\mathbb{R}$ satisfying

$$\begin{cases} \det \mathbf{D}^2 u(x,y) = \pi\delta_{(1/2,1/2)}(x,y) & \text{in } \Omega, \\ u(x,y) = \sqrt{(x-0.5)^2+\left(y-0.5\right)^2} & \text{on } \partial\Omega. \end{cases} \tag{31}$$

The exact solution u of problem (31) is defined by

$$u(x,y) = \sqrt{(x-0.5)^2+\left(y-0.5\right)^2} \;\; \forall\,(x,y)\in\overline{\Omega}.$$

We regularize the problem and approximate the Dirac function $\delta_{(\alpha,\beta)}$ by

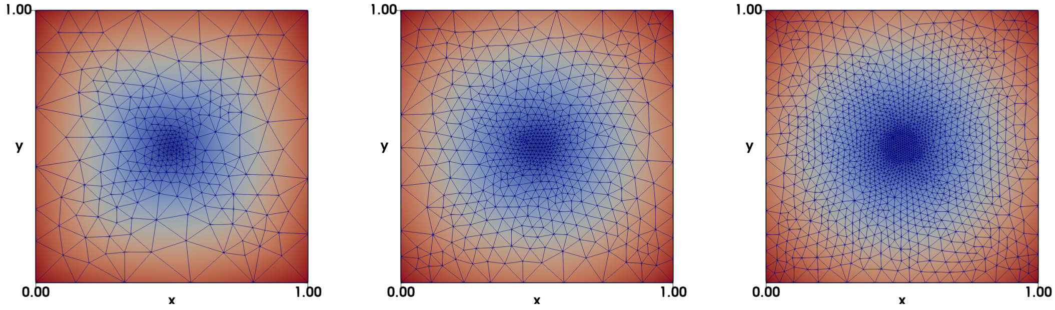
$$\delta_{(\alpha,\beta)}(x,y) \sim f_\varepsilon^{(\alpha,\beta)}\left(x,y\right) = \frac{\varepsilon^2}{\pi\left(\varepsilon^2+(x-\alpha)^2+\left(y-\beta\right)^2\right)^2},$$

where $\varepsilon>0$ is a small parameter. Therefore, we approximate $\pi\delta_{(1/2,1/2)}(x,y)$ by $\pi f_\varepsilon^{(1/2,1/2)}(x,y)$, and consider the approximate problem: find $u_\varepsilon:\Omega\to\mathbb{R}$ satisfying

$$\begin{cases} \det \mathbf{D}^2 u_\varepsilon(x,y) = \pi f_\varepsilon^{(1/2,1/2)}(x,y) & \text{in } \Omega, \\ u_\varepsilon(x,y) = \sqrt{(x-0.5)^2+\left(y-0.5\right)^2} & \text{on } \partial\Omega. \end{cases} \tag{32}$$

To estimate the error norms, we assume that $\lim_{h\to0}\lim_{\varepsilon\to0} u_{\varepsilon,h}=u$, where $u_{\varepsilon,h}$ is the discrete approximation of u_ε obtained with our approach. Unless specified otherwise, we set here $\varepsilon=10^{-3/2}$. This approximation of the right-hand side of (31) allows to calculate the L^2-norm of the residual that appears within the error indicator (26). In this experiment, we expect the adaptive mesh refinement algorithm to adapt the mesh around the point $(0.5,0.5)$.

Figure 3 visualizes the final refined meshes for different tolerances. Table 4 shows the quantified numerical results. When the tolerance decreases, $h_{\min}$, $h_{\max}$, η_K^I, and the L^2 error norm decrease, and the number of elements and vertices increase as expected. Figure 4 visualizes the refined meshes with fixed tolerance TOL = 0.25 and different values of ε. We observe that the smaller the parameter ε is, the larger the number of elements. Table 5 shows that as ε decreases, $h_{\min}$, $h_{\max}$, and the L^2 error norm decreases and the number of elements and vertices increases.

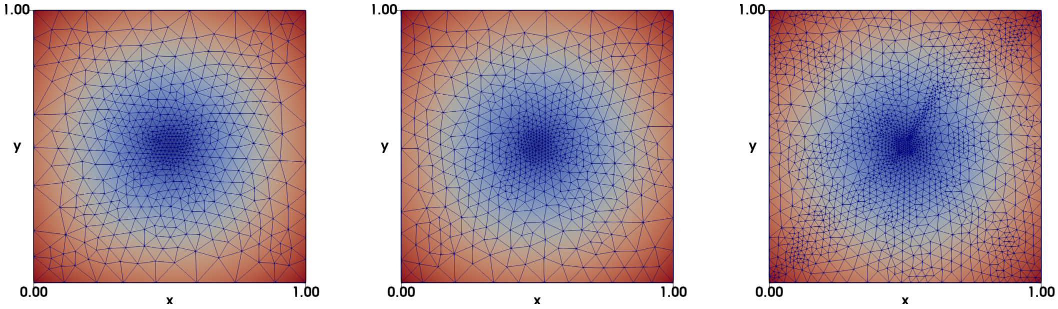


(a) TOL = 0.5 **(b)** TOL = 0.25 **(c)** TOL = 0.125

Figure 3. Numerical experiments with a Dirac function, for data $f(x, y) = \pi\delta_{(1/2,1/2)}$ and $g(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. Graphs of the final adapted mesh for various values of TOL, fixed $\varepsilon = 10^{-3/2}$, and after 500 iterations.

Table 4. Numerical experiments with a Dirac function, for data $f(x, y) = \pi\delta_{(1/2,1/2)}$ and $g(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. Convergence behavior of the algorithm for various values of parameter TOL, fixed $\varepsilon = 10^{-3/2}$, and after 500 iterations. The columns contain the final minimal and maximal mesh size, the final numbers of elements and vertices, the value of the indicator, and L^2 error norms.

TOL	$h_{\min}$	$h_{\max}$	# elem	# vertices	η^I	$\|u-u_h\|_{L^2}$
0.5	$1.36\cdot10^{-2}$	$1.77\cdot10^{-1}$	509	269	$8.71\cdot10^{-1}$	$7.73\cdot10^{-2}$
0.25	$8.41\cdot10^{-3}$	$1.24\cdot10^{-1}$	1128	582	$5.50\cdot10^{-1}$	$3.84\cdot10^{-2}$
0.125	$4.89\cdot10^{-3}$	$7.78\cdot10^{-2}$	2920	1489	$3.70\cdot10^{-1}$	$1.92\cdot10^{-2}$



(a) $\varepsilon = 10^{-3/2}$ **(b)** $\varepsilon = 10^{-2}$ **(c)** $\varepsilon = 10^{-5/2}$

Figure 4. Numerical experiments with a Dirac function, for data $f(x, y) = \pi\delta_{(1/2,1/2)}$ and $g(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. Graphs of the final adapted mesh for various values of ε, fixed TOL= 0.25, and after 500 iterations.

5.5. *Numerical experiment with data involving two Dirac functions*

Finally, let us consider a numerical experiment with two Dirac functions on the right hand side of the Monge–Ampère equation: find $u : \Omega \to \mathbb{R}$ satisfying

$$\begin{cases} \det \mathbf{D}^2 u = \frac{\pi}{2}\delta_{(1/4,1/2)} + \frac{\pi}{2}\delta_{(3/4,1/2)} & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{33}$$

Table 5. Numerical experiments with a Dirac function, for data $f(x,y) = \pi\delta_{(1/2,1/2)}$ and $g(x,y) = \sqrt{(x-0.5)^2+(y-0.5)^2}$. Convergence behavior of the algorithm for various values of parameter $\varepsilon = 10^{-3/2}$, fixed TOL= 0.25, and after 500 iterations. The columns contain the final minimal and maximal mesh size, the final numbers of elements and vertices, the value of the indicator, and the L^2 error norm.

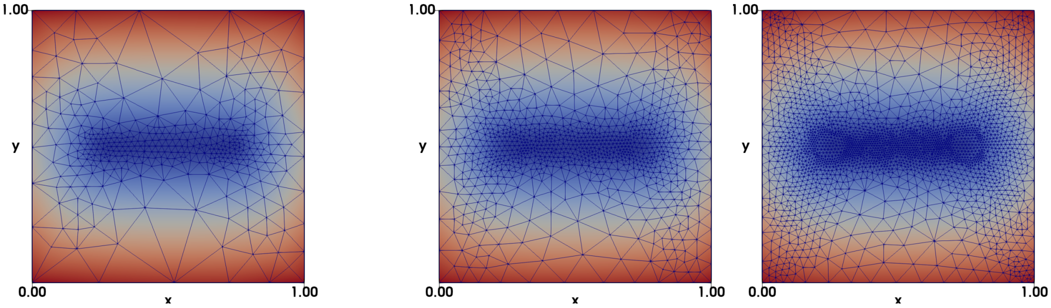
ε^2	$h_{\min}$	$h_{\max}$	# elem	# vertices	η^I	$\|u-u_h\|_{L^2}$
$1\cdot10^{-3}$	$8.41\cdot10^{-3}$	$1.24\cdot10^{-1}$	1128	582	$5.50\cdot10^{-1}$	$3.84\cdot10^{-2}$
$1\cdot10^{-4}$	$7.93\cdot10^{-3}$	$1.40\cdot10^{-1}$	1224	636	$0.12\cdot10^{-1}$	$2.29\cdot10^{-2}$
$1\cdot10^{-5}$	$4.19\cdot10^{-3}$	$7.02\cdot10^{-2}$	2986	1544	$0.15\cdot10^{-1}$	$1.29\cdot10^{-2}$

where

$$g(x,y) = \begin{cases} |y-0.5| & \text{if } 1/4 < x < 3/4, \\ \min\left(\sqrt{(x-1/4)^2+(y-0.5)^2}, \sqrt{(x-3/4)^2+(y-0.5)^2}\right) & \text{otherwise}. \end{cases} \tag{34}$$

The exact solution is given by $u = g$ inside the domain Ω.

We approximate $\frac{\pi}{2}\delta_{(1/4,1/2)} + \frac{\pi}{2}\delta_{(3/4,1/2)}$ by $\frac{\pi}{2} f_\varepsilon^{(1/4,1/2)}(x,y) + \frac{\pi}{2} f_\varepsilon^{(3/4,1/2)}(x,y)$, and we set $\varepsilon = 10^{-3/2}$. To estimate the error norms, we assume that $\lim_{h\to0}\lim_{\varepsilon\to0} u_\varepsilon^\eta = u$. In this case, we expect the adaptive algorithm to refine the mesh in the neighborhood of the two points $(1/4,1/2)$ and $(3/4,1/2)$, and in the neighborhood of the segment between these two points. Figure 3 visualizes the final refined meshes for different tolerances, and shows that the mesh is refined at the appropriate locations. Table 4 shows the numerical results; the L^2 error norm is of order 10^{-2} and decreases monotonically with the tolerance.



(**a**) TOL = 0.5 (**b**) TOL = 0.25 (**c**) TOL = 0.125

Figure 5. Examples with Dirac function for data $f(x,y) = \frac{\pi}{2}\delta_{(1/4,1/2)} + \frac{\pi}{2}\delta_{3/4,1/2)}$ and $g(x,y)$ given by (34). Graphs of the final adapted mesh for various values of TOL, fixed $\varepsilon = 10^{-3/2}$ and after 500 iterations.

6. Conclusion

A least-squares approach has been advocated for the solution to the two-dimensional Monge-Ampère equation. A relaxation algorithm decouples this least-squares problem into a sequence of local nonlinear problems and variational linear problems.

An adaptive mesh refinement algorithm has been added to the relaxation algorithm. An incomplete a posteriori error indicator has been derived based on a different, conservative, formulation of the Monge–Ampère equation, and then inserted into the least-squares framework.

Table 6. Examples with Dirac function for data $f(x,y) = \frac{\pi}{2}\delta_{(1/4,1/2)} + \frac{\pi}{2}\delta_{(3/4,1/2)}$ and $g(x,y)$ given by (34). Convergence behavior of the algorithm for various values of parameter TOL, fixed $\varepsilon = 10^{-3/2}$ and after 500 iterations. The columns contain the final minimal and maximal mesh size, the final numbers of elements and vertices, the value of the indicator, and L^2 error norms.

TOL	$h_{\min}$	$h_{\max}$	# elem	# vertices	η^I	$\|u-u_h\|_{L^2}$
0.5	$1.23\cdot10^{-2}$	$2.17\cdot10^{-1}$	687	356	$5.98\cdot10^{-1}$	$3.84\cdot10^{-2}$
0.25	$8.14\cdot10^{-3}$	$2.25\cdot10^{-1}$	1617	828	$4.82\cdot10^{-1}$	$2.08\cdot10^{-2}$
0.125	$4.89\cdot10^{-3}$	$7.78\cdot10^{-2}$	2920	1489	$3.70\cdot10^{-1}$	$1.92\cdot10^{-2}$

Numerical experiments validate the accuracy and the efficiency of the method. In particular, numerical experiments in non-smooth cases have shown optimal error convergence orders and robustness, and an appropriate mesh refinement near the singularities of the solution. The adaptive strategy must be improved with a derivation of more accurate error estimates, to obtain an optimal performance of the error indicator.

Future work may include the extension of the algorithm to three dimensions of space. The least-squares framework used in this work has been initially introduced in [4] in two dimensions of space, and then extended in [9] in three dimensions of space. Three-dimensional mesh adaptation is not an issue, since efficient mesh generators are available, see, e.g., the `Mmg platform` [36].

Declaration of interests

The authors do not work for, advise, own shares in, or receive funds from any organization that could benefit from this article, and have declared no affiliations other than their research organizations.

Dedication

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

References

[1] L. A. Caffarelli, X. Cabré, *Fully Nonlinear Elliptic Equations*, Colloquium Publications, vol. 43, American Mathematical Society, 1995.

[2] J.-D. Benamou, B. D. Froese, A. M. Oberman, "Numerical solution of the optimal transportation problem using the Monge–Ampère equation", *J. Comput. Phys.* **49** (2014), no. 4, p. 107-126.

[3] B. Engquist, B. D. Froese, Y. Yang, "Optimal transport for seismic full waveform inversion", *Commun. Math. Sci.* **14** (2016), no. 8, p. 2309-2330.

[4] 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, Control Optim. Calc. Var.* **19** (2013), no. 3, p. 780-810.

[5] X. Feng, M. Neilan, R. Glowinski, "Recent Developments in Numerical Methods for Fully Nonlinear 2nd Order PDEs", *SIAM Rev.* **55** (2013), no. 2, p. 205-267.

[6] A. D. Aleksandrov, "Uniqueness conditions and estimates for the solution of the Dirichlet problem", *Transl., Ser. 2, Am. Math. Soc.* **68** (1968), p. 89-119.

[7] H. Ishii, P.-L. Lions, "Viscosity solutions of fully nonlinear second-order elliptic partial differential equations", *J. Differ. Equations* **83** (1990), no. 1, p. 26-78.

[8] A. M. Oberman, "Wide stencil finite difference schemes for the elliptic Monge–Ampère equations and functions of the eigenvalues of the Hessian", *Discrete Contin. Dyn. Syst., Ser. B* **10** (2008), no. 1, p. 221-238.

[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), no. 1, p. 53-78.

[10] I. Babuska, W. C. Rheinboldt, "A posteriori error estimates for the finite element method", *Int. J. Numer. Methods Eng.* **12** (1978), no. 10, p. 1597-1615.

[11] R. Verfürth, *A review of a posteriori error estimation and adaptive mesh-refinement techniques*, Advances in numerical mathematics, John Wiley & Sons, Chichester, 1996.

[12] N. R. Iyer, "A posteriori error estimation and adaptive mesh refinement for reliable finite element solutions", *Current Science* **77** (1999), no. 10, p. 1319-1324.

[13] R. Verfürth, *A Posteriori Error Estimation Techniques for Finite Element Methods*, Numerical Mathematics and Scientific Computation, Oxford University Press, 2013.

[14] G. Akrivis, C. Makridakis, R. H. Nochetto, "Optimal order a posteriori error estimates for a class of Runge–Kutta and Galerkin methods", *Numer. Math.* **114** (2009), no. 1, p. 133-160.

[15] R. Verfürth, "A posteriori error estimates for nonlinear problems. Finite element discretizations of elliptic equations", *Math. Comput.* **62** (1994), no. 206, p. 445-475.

[16] W. Dorfler, "A Convergent Adaptive Algorithm for Poisson's Equation", *SIAM J. Numer. Anal.* **33** (1996), no. 3, p. 1106-1124.

[17] W. Bangerth, R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Lectures in Mathematics, ETH Zürich, Springer, 2003.

[18] O. Lakkis, T. Pryer, "A finite element method for nonlinear elliptic problems", *SIAM J. Sci. Comput.* **35** (2013), no. 4, p. A2025-A2045.

[19] B. H. Froese, T. Salvador, "Higher-Order Adaptive Finite Difference Methods for Fully Nonlinear Elliptic Equations", *J. Sci. Comput.* **75** (2018), no. 3, p. 1282-1306.

[20] A. Laadhari, P. Saramito, C. Misbah, "An adaptive finite element method for the modeling of the equilibrium of red blood cells", *Int. J. Numer. Methods Fluids* **80** (2015), p. 397-428.

[21] S. C. Brenner, M. Neilan, "Finite element approximations of the three dimensional Monge–Ampère equation", *M2AN, Math. Model. Numer. Anal.* **46** (2012), no. 5, p. 979-1001.

[22] S. C. Brenner, T. Gudi, M. Neilan, L.-Y. Sung, "C^0 penalty methods for the fully nonlinear Monge–Ampère equation", *Math. Comput.* **80** (2011), no. 276, p. 1979-1995.

[23] 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. Comput.* **81** (2019), no. 3, p. 2271-2302.

[24] R. Glowinski, "Finite Element Methods For Incompressible Viscous Flow", in *Numerical methods for fluids (Part 3)*, Handbook of Numerical Analysis, vol. 9, Elsevier, 2003, p. 3-1176.

[25] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, 2nd ed., Scientific Computation, Springer, 2008.

[26] D. N. Arnold, "Mixed finite element methods for elliptic problems", *Comput. Methods Appl. Mech. Eng.* **82** (1990), no. 1-3, p. 281-300.

[27] D. Boffi, F. Brezzi, M. Fortin, *Mixed Finite Element Methods and Applications*, Springer Series in Computational Mathematics, vol. 44, Springer, 2013.

[28] G. Awanou, H. Li, "Error Analysis of a Mixed Finite Element Method for the Monge–Ampère Equation", *Int. J. Numer. Anal. Model.* **11** (2014), no. 4, p. 745-761.

[29] 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), no. 2, p. 1226-1250.

[30] A. N. Tychonoff, "The regularization of incorrectly posed problems", *Dokl. Akad. Nauk SSSR* **153** (1963), p. 49-52.

[31] M. Picasso, F. Alauzet, H. Borouchaki, P.-L. George, "A Numerical Study of Some Hessian Recovery Techniques on Isotropic and Anisotropic Meshes", *SIAM J. Sci. Comput.* **33** (2011), no. 3, p. 1058-1076.

[32] E. H. Georgoulis, P. Houston, J. Virtanen, "An a posteriori error indicator for discontinuous Galerkin approximations of fourth-order elliptic problems", *IMA J. Numer. Anal.* **31** (2011), no. 1, p. 281-298.

[33] A. Charbonneau, K. Dossou, R. Pierre, "A residual-based a posteriori error estimator for the Ciarlet–Raviart formulation of the first biharmonic problem", *Numer. Methods Partial Differ. Equations* **13** (1997), no. 1, p. 93-111.

[34] P. Laug, H. Borouchaki, "The BL2D Mesh Generator: Beginner's Guide, User's and Programmer's Manual", Tech. Report T-0194, INRIA, 1996, https://hal.inria.fr/inria-00069977.

[35] E. J. Dean, R. Glowinski, "Numerical methods for fully nonlinear elliptic equations of the Monge–Ampère type", *Comput. Methods Appl. Mech. Eng.* **195** (2006), no. 13-16, p. 1344-1386.

[36] C. Dapogny, C. Dobrzynski, P. Frey, "Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems", *J. Comput. Phys.* **262** (2014), p. 358-378.