SIAM J. Sci. Comput.
Vol. 41, No. 6, pp. B1341–B1367

NUMERICAL APPROXIMATION OF ORTHOGONAL MAPS[*]

ALEXANDRE CABOUSSAT[†], ROLAND GLOWINSKI[‡], DIMITRIOS GOURZOULIDIS[†][§], AND MARCO PICASSO[§]

Abstract. Orthogonal maps are the solutions of the mathematical model of paper-folding, also called the origami problem. They consist of a system of first-order fully nonlinear equations involving the gradient of the solution. The Dirichlet problem for orthogonal maps is considered here. A variational approach is advocated for the numerical approximation of the maps. The introduction of a suitable objective function allows us to enforce the uniqueness of the solution. A strategy based on a splitting algorithm for the corresponding flow problem is presented and leads to decoupling the time-dependent problem into a sequence of local nonlinear problems and a global linear variational problem at each time step. Numerical experiments validate the accuracy and the efficiency of the method for various domains and meshes.

Key words. orthogonal maps, Eikonal equation, origami, operator splitting, finite element methods, dynamical flow

AMS subject classifications. 65N30, 65K10, 65D18, 65N20, 49M20, 35F30

DOI. 10.1137/19M1243683

1. Introduction.

The *Eikonal equation* $|\nabla u| = 1/c$, with $c > 0$, is a prototypical equation in analysis. It has various applications in science and engineering, such as optics, wave propagation, material science, differential geometry, geophysics, and image processing [14, 15, 32, 35, 36, 41, 43]. It is a classical example of a so-called *first-order implicitly, or fully, nonlinear equation* [26, 27]. The analysis of such strongly nonlinear models can be found, e.g., in [10, 11, 16, 25] (see also the references therein).

Inspired by [20], our interest in this article lies in the numerical approximation of *origami problems*. The mathematical analysis of paper-folding problems has been addressed, e.g., in [4, 16, 17, 18, 19], a related challenge being finding a so-called *orthogonal map*.

Orthogonal maps, or isogeometric maps, are related to Eikonal equations in the sense that they can be considered as a multivariate version of the classical, scalar, Eikonal equation. More precisely, we are interested in the computation of the approximate solutions of the Dirichlet problem for an orthogonal maps equation. Namely, we want to find $\mathbf{u} : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^2$ satisfying

$$\begin{cases} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega, \end{cases}$$

[*] Submitted to the journal's Computational Methods in Science and Engineering section February 12, 2019; accepted for publication (in revised form) August 6, 2019; published electronically December 10, 2019.

https://doi.org/10.1137/19M1243683

Funding: This work was partially supported by the Swiss National Science Foundation (SNF grant 165785) and by the Hong Kong based Kennedy Wong Foundation.

[†] Geneva School of Business Administration, University of Applied Sciences and Arts Western Switzerland (HES-SO), 1227 Carouge, Switzerland (alexandre.caboussat@hesge.ch, dimitrios.gourzoulidis@hesge.ch).

[‡] Department of Mathematics, University of Houston, Houston, TX 77204-3008, and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (roland@math.uh.edu).

[§] Institute of Mathematics, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (dimitrios.gourzoulidis@epfl.ch, marco.picasso@epfl.ch).

where $\mathbf{g}$ is a given, sufficiently smooth, function, and $\mathcal{O}(2)$ is the space of orthonormal 2×2 matrix-valued functions. Note that $\nabla \mathbf{u} \in \mathcal{O}(2)$ can be written as $\nabla \mathbf{u}(\nabla \mathbf{u})^T = \mathbf{I}$.

Although our initial application of orthogonal maps appears in the theory of paper-folding, there exist similar related formulations in other fields, such as computational geometry [1, 23], rigid maps [24, 42], or rigid displacements and bending [2, 3]. The solutions to the orthogonal maps problem have a low regularity, and the uniqueness of the solution is not even guaranteed. A possible definition of a *generalized* solution is provided by the notion of *viscosity solutions*; see, e.g., [13]. On the other hand, we introduce here a variational framework to enforce uniqueness of the solution.

We address the solution of implicitly nonlinear equations from a variational viewpoint. Such numerical methods have been devised, e.g., in [5, 8, 14, 15, 31] when dealing with Eikonal-type equations, in [7, 9] when addressing the approximation of the related second-order Monge–Ampère equation, or in [12, 22, 29, 34] for the Pucci's equation. Tentative approaches for the orthogonal maps problem, combining these different methods, have been described in [33] or in [30, Chapter 8].

In this article, a calculus of variations approach is introduced to solve the orthogonal maps problem. The main ingredients of the methodology are

- a *penalization* method to relax the orthogonality condition $\nabla \mathbf{u}(\nabla \mathbf{u})^T = \mathbf{I}$;
- the derivation of the Euler–Lagrange equation of the penalized and regularized problem;
- the introduction of a related initial value *dynamical flow* problem;
- the use of an *operator-splitting* scheme (à la Marchuk–Yanenko) to time-discretize the above flow problem;
- a low-order C^0-conforming finite element approximation.

The operator-splitting approach allows the decoupling of the differential operators from the nonlinearities of the problem. It is worth noting that low-order finite element approximations are well-suited to the Lipschitz continuous regularity of the solutions, and to domains with curved boundaries.

The article is organized as follows: In section 2 we formulate the initial orthogonal maps problem. Our computational approach is described and detailed in section 3. The finite element implementation of our methodology is discussed in section 4. The numerical experiments are presented in section 5. The conclusion includes some perspectives for future work.

2. Mathematical model.

Let Ω be an open bounded domain of $\mathbb{R}^2$, and consider $\mathbf{g} : \partial\Omega \to \mathbb{R}^2$ a sufficiently smooth given function on its boundary. The unit square $\Omega = (0,1)^2$ is a typical domain when considering origami applications. The problem of interest reads as follows: Find $\mathbf{u} = [u_1, u_2]^T : \Omega \to \mathbb{R}^2$ satisfying

$$\left\{ \begin{array}{ll} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega. \end{array} \right. \tag{2.1}$$

Here u_1 (resp., u_2) denotes the first (resp., second) component of the mapping $\mathbf{u}$. Actually $\mathbf{u}$ is a mapping that maps Ω (the original sheet) into another domain of $\mathbb{R}^2$. The image $\mathbf{u}(\Omega)$ of Ω through this mapping corresponds to the resulting paper sheet, described in the two-dimensional space, after the folding. Problem (2.1) can be equivalently written as

$$\begin{cases} |\nabla u_1| = 1 & \text{a.e in } \Omega, \\ |\nabla u_2| = 1 & \text{a.e in } \Omega, \\ \nabla u_1 \cdot \nabla u_2 = 0 & \text{a.e in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega. \end{cases} \tag{2.2}$$

Remark 2.1 (orthogonal maps and rigid maps). Actually, problem (2.1) is derived from the so-called *rigid maps* problem that consists in finding the map $\tilde{\mathbf{u}} : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ such that

$$\begin{cases} \nabla\tilde{\mathbf{u}} \in \mathcal{O}(2,3) & \text{in } \Omega, \\ \tilde{\mathbf{u}} = \tilde{\mathbf{g}} & \text{on } \partial\Omega, \end{cases} \tag{2.3}$$

where $\mathcal{O}(2,3) = \left\{\mathbf{A} : \Omega \to \mathbf{R}^{2\times 3} : \mathbf{A}^T\mathbf{A} = \mathbf{I} \text{ in } \Omega\right\}$. The solution to this equivalent problem consists in finding the mapping $\tilde{\mathbf{u}}$ that embeds the folded paper $\tilde{\mathbf{u}}(\Omega)$ into $\mathbb{R}^3$. A graphical representation of problems (2.1) and (2.3) can be found in Figure 1.

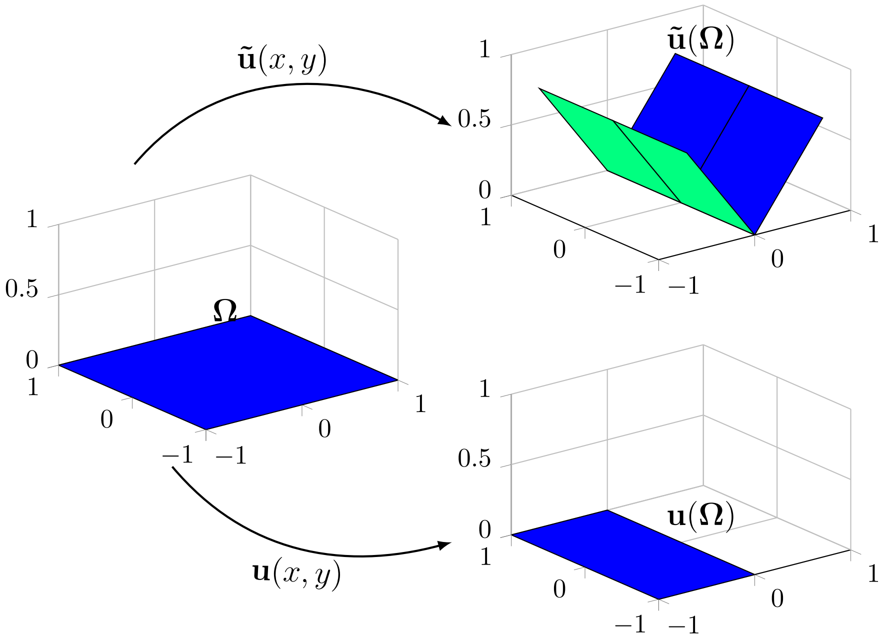

FIG. 1. *Visualization (bottom) of the orthogonal map* $\mathbf{u} : \mathbb{R}^2 \to \mathbb{R}^2$*, where* $\mathbf{u}(x,y) = (-|x|, y)$ *is the solution to* (2.1) *when folding a paper sheet along a centered axis oriented with the Ox axis, and visualization (top) of the rigid map* $\tilde{\mathbf{u}} : \mathbb{R}^2 \to \mathbb{R}^3$*, where* $\tilde{\mathbf{u}}(x,y) = (x/\sqrt{2}, y, |x|/\sqrt{2})$ *is the solution to* (2.3) *when folding a paper sheet in* $\mathbb{R}^3$*.*

Remark 2.2 (determinant property and angle condition). A consequence of (2.1) is that the solution $\mathbf{u}$ satisfies

$$\det\nabla\mathbf{u}(\mathbf{x}) = \pm 1 \quad \text{a.e.}\, \mathbf{x} \in \Omega. \tag{2.4}$$

Moreover, for each vertex that lies at the intersection of folding lines, conditions apply on the number of singularity lines crossing each other [16]. This property is sometimes called the *angle condition* and typically states that the number of edges coming from one vertex is even, and the determinant $\det \nabla \mathbf{u}$ alternates between 1 and -1 for adjacent regions touching the vertex.

The problem of interest (2.2) is reminiscent of the scalar Eikonal equation but introduces a coupling between the two components; it can be seen therefore as a *vectorial Eikonal problem*. Theoretical considerations of (2.1) or (2.2) have been addressed, e.g., in [16, 18, 20]. For instance, an explicit solution of the homogeneous Dirichlet problem has been exhibited in [21]. Existence conditions of a solution have been shown, but the uniqueness of that solution is generally not guaranteed (we can easily construct problems with multiple solutions).

3. Numerical algorithms.

The proposed solution method for (2.2) relies on a mix of classical variational techniques. The underlying principles are the introduction of the variational principle (energy to minimize), and of the corresponding flow problem, which is solved by operator-splitting techniques. Ultimately, the splitting strategy allows the decoupling of the local nonlinearities and of the differential operators (linear here).

3.1. Regularization and penalization.

To solve (2.2) and to enforce the uniqueness of the solution in some sense, we consider the following variational problem: Find $\mathbf{u} \in \mathbf{E_g}$ satisfying

$$J(\mathbf{u}) \leq J(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{E_g}, \tag{3.1}$$

where

$$J(\mathbf{v}) = \frac{C}{2}\int_\Omega |\mathbf{v}-\mathbf{f}|^2 \, \mathrm{d}\mathbf{x} + \frac{1}{2}\int_\Omega |\nabla \mathbf{v}|^2 \mathrm{d}\mathbf{x}, \tag{3.2}$$

and

$$\mathbf{E_g} = \{\mathbf{v} \in (H^1(\Omega))^2,\ \mathbf{v}|_{\partial\Omega} = \mathbf{g},\ \nabla\mathbf{v} \in \mathcal{O}(2) \text{ a.e. in } \Omega\}. \tag{3.3}$$

In (3.2), $C > 0$ is a positive constant. In the first term of (3.2), the function $\mathbf{f}$ is a given vector-valued function, which corresponds to some *target* solution, for instance, given by prior information and/or existing data. In other words, the first term of the objective function is a distance term to minimize (fidelity term). In most numerical experiments, no a priori information is known about the solution, and we use $\mathbf{f} = \mathbf{0}$. The second term is a classical smoothing term.

Note that (2.2) itself has been transferred into a constraint in (3.3). To handle such constraints, we use an approach which has been successful with the scalar Eikonal equation in [5], namely, we penalize the constraint $\nabla\mathbf{v} \in \mathcal{O}(2)$ (or, equivalently, the first three equations of (2.2)). Let $\varepsilon_1 > 0$ be a regularization parameter, and let $\varepsilon_2 > 0$ be a given (penalization) parameter. The modified objective function is defined as follows:

$$\begin{aligned} J_\varepsilon(\mathbf{v}) := {} & J(\mathbf{v}) + \frac{\varepsilon_1}{2}\int_\Omega |\nabla^2\mathbf{v}|^2 \mathrm{d}\mathbf{x} \\ & + \frac{1}{4\varepsilon_2}\int_\Omega \left[(|\nabla v_1|^2-1)^2 + (|\nabla v_2|^2-1)^2 + |\nabla v_1 \cdot \nabla v_2|^2\right] \mathrm{d}\mathbf{x}. \end{aligned} \tag{3.4}$$

The second term is a biharmonic regularization we introduce to improve the convergence of the numerical algorithm. The introduction of such a term has already been experimented with for similar problems (see [6]) and will be discussed numerically via experiments. We denote $(\varepsilon_1, \varepsilon_2)$ by ε. The variational problem (3.1) becomes the following: Find $\mathbf{u}^\varepsilon \in \mathbf{V_g}$ satisfying

$$J_\varepsilon(\mathbf{u}^\varepsilon) \leq J_\varepsilon(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V_g}, \tag{3.5}$$

where

$$\mathbf{V_g} = \{\mathbf{v} \in (H^2(\Omega))^2,\, \mathbf{v}|_{\partial\Omega} = \bar{\mathbf{g}}\}. \tag{3.6}$$

The numerical approach to solving (3.5) relies on an appropriate reformulation of the problem when considering the first-order optimality conditions, together with the introduction of a flow problem.

Remark 3.1. In (3.6), we used $\bar{\mathbf{g}}$ (an approximation of $\mathbf{g}$ belonging to $(H^{3/2}(\partial\Omega))^2$) to make the boundary data compatible with the regularizing term $\frac{\varepsilon_1}{2}\int_\Omega |\nabla^2\mathbf{v}|^2 \mathrm{d}\mathbf{x}$ we introduced in (3.2), this term forcing the $(H^2(\Omega))^2$-regularity of the solutions of problem (3.1). Actually, the results of the numerical experiments reported in section 3 show that there is no need to regularize $\mathbf{g}$ in practice. Our explanation of this good news is that the original function $\mathbf{g}$, being necessarily piecewise affine over $\partial\Omega$ (if Ω is a rectangle), has the $(W^{1,\infty}(\partial\Omega) \cap H^s(\partial\Omega))^2$-regularity for all $s < 3/2$. From this observation, we feel justified using (for simplicity) $\mathbf{g}$ instead of $\bar{\mathbf{g}}$ in the remainder of this article (a very small abuse of language, indeed).

Remark 3.2. Indeed, if Ω is a rectangle, a piecewise affine function f over $\partial\Omega$ has the $(W^{1,\infty}(\partial\Omega) \cap H^s(\partial\Omega))^2$-regularity for all $s < 3/2$. This property comes from the fact that a function $f \in H^s(\mathbb{R})$ if and only if $(1 + |\xi|^2)^{s/2}\hat{f} \in L^2(\mathbb{R})$, where $\hat{f}$ denotes the Fourier transform of function f. In the case of a piecewise affine function with a compact support, one can show via direct calculations that its Fourier transform verifies the above inclusion for all $s < 3/2$.

3.2. First-order optimality conditions. We define the tensor-valued function $\mathbf{p}^\varepsilon := \nabla\mathbf{u}^\varepsilon$. Problem (3.5) is equivalent to the following: Find $\mathbf{p}^\varepsilon \in \mathbf{Q}_4$ satisfying

$$j_\varepsilon(\mathbf{p}^\varepsilon) \leq j_\varepsilon(\mathbf{q}) \quad \forall \mathbf{q} \in \mathbf{Q}_4, \tag{3.7}$$

where

$$\begin{aligned} j_\varepsilon(\mathbf{q}) = {} & \frac{1}{2}\int_\Omega |\mathbf{q}|^2 + \frac{C}{2}\int_\Omega |\mathbf{Bq} + \mathbf{u_g} - \mathbf{f}|^2 \,\mathrm{d}\mathbf{x} + I_\nabla(\mathbf{q}) \\ & + \frac{1}{4\varepsilon_2}\int_\Omega \left[(|q_1|^2 - 1)^2 + (|q_2|^2 - 1)^2 + |q_1 \cdot q_2|^2\right] \mathrm{d}\mathbf{x}, \end{aligned} \tag{3.8}$$

where $\mathbf{Q}_4 = (L^4(\Omega))^{2\times 2}$ and q_1, q_2 are the two column vectors of $\mathbf{q}$, and where we define

(i) the function $\mathbf{u_g} \in (H^1(\Omega))^2$ as the unique solution (harmonic extension) of

$$\begin{cases} \nabla^2\mathbf{u_g} = \mathbf{0} & \text{in } \Omega, \\ \mathbf{u_g} = \mathbf{g} & \text{on } \partial\Omega; \end{cases}$$

(ii) the function $\mathbf{Bq}$ as the unique solution in $(H_0^1(\Omega))^2$ of

$$\nabla^2\mathbf{Bq} = \nabla \cdot \mathbf{q} \quad \text{in } \Omega;$$

(iii) the functional $I_\nabla(\mathbf{q})$ as

$$I_\nabla(\mathbf{q}) = \begin{cases} \dfrac{\varepsilon_1}{2}\displaystyle\int_\Omega |\nabla\cdot\mathbf{q}|^2 d\mathbf{x} & \text{if } \mathbf{q}\in\nabla\mathbf{V_g},\\ +\infty & \text{otherwise.}\end{cases}$$

Based on this *change of variables*, the first-order optimality conditions (Euler–Lagrange equations) relative to (3.8) read as follows: Find $\mathbf{p}^\varepsilon\in\mathbf{Q}_4$ such that

$$\begin{aligned} &\int_\Omega \mathbf{p}^\varepsilon : \mathbf{q}\mathrm{d}\mathbf{x} + C\int_\Omega(\mathbf{B}\mathbf{p}^\varepsilon+\mathbf{u_g}-\mathbf{f})\cdot\mathbf{B}\mathbf{q}\mathrm{d}\mathbf{x} + \langle\partial I_\nabla(\mathbf{p}^\varepsilon),\mathbf{q}\rangle\\ &+\frac{1}{\varepsilon_2}\int_\Omega\Big[(|p_1^\varepsilon|^2-1)p_1^\varepsilon\cdot q_1 + (|p_2^\varepsilon|^2-1)p_2^\varepsilon\cdot q_2\\ &+\frac{1}{2}p_1^\varepsilon\cdot p_2^\varepsilon(p_2^\varepsilon\cdot q_1+p_1^\varepsilon\cdot q_2)\Big]\mathrm{d}\mathbf{x} = 0 \qquad \forall\mathbf{q}\in\mathbf{Q}_4. \end{aligned} \tag{3.9}$$

Here, $\partial I_\nabla(\cdot)$ denotes the subdifferential of the nonsmooth proper lower semicontinuous (l.s.c.) convex functional I_∇. In what follows, the superscript ε will be dropped for simplicity.

3.3. Flow problem and operator-splitting algorithm.

The solution method for (3.9) relies on an associated initial value problem (*flow* in the dynamical systems terminology), to be integrated from $t=0$ to $t=+\infty$. This initial value problem is defined as follows: Find $\mathbf{p}(t)\in\mathbf{Q}_4$ for a.e. $t\in(0,+\infty)$ satisfying

$$\begin{aligned} &\int_\Omega \frac{\partial\mathbf{p}(t)}{\partial t} : \mathbf{q}\mathrm{d}\mathbf{x} + \int_\Omega \mathbf{p}(t):\mathbf{q}\mathrm{d}\mathbf{x} + C\int_\Omega(\mathbf{B}\mathbf{p}(t)+\mathbf{u_g}-\mathbf{f})\cdot\mathbf{B}\mathbf{q}\mathrm{d}\mathbf{x} + \langle\partial I_\nabla(\mathbf{p}(t)),\mathbf{q}\rangle\\ &+\frac{1}{\varepsilon_2}\int_\Omega\Big[(|p_1(t)|^2-1)p_1(t)\cdot q_1 + (|p_2(t)|^2-1)p_2(t)\cdot q_2\\ &+\frac{1}{2}p_1(t)\cdot p_2(t)(p_2(t)\cdot q_1+p_1(t)\cdot q_2)\Big]\mathrm{d}\mathbf{x} = 0 \qquad \forall\mathbf{q}\in\mathbf{Q}_4, \end{aligned} \tag{3.10}$$

together with the given initial condition $\mathbf{p}(0)=\mathbf{p}_0$. The initial condition $\mathbf{p}_0\in\mathbf{Q}_4$ is defined as $\nabla\mathbf{u}_0$, where $\mathbf{u}_0\in\mathbf{V_g}$ satisfies

$$-\nabla^2\mathbf{u}_0 = \boldsymbol{\delta} \quad \text{ in } \Omega,$$

where $\boldsymbol{\delta}=(0.0005, 0.0005)^T$ is a small right-hand side (quasi-harmonic extension).

We apply an operator-splitting strategy to solve (3.10) (namely, a first-order Marchuk–Yanenko scheme). Let $\Delta t>0$ be a constant given time step, $t^n=n\Delta t$, $n=1,2,\ldots$, to define the approximations $\mathbf{p}^n\simeq\mathbf{p}(t^n)$. Starting from the initial condition $\mathbf{p}^0=\mathbf{p}_0$, the Marchuk–Yanenko scheme allows us, using $\mathbf{p}^n$ for all $n\geq 0$, to compute successively $\mathbf{p}^{n+1/2}$ and $\mathbf{p}^{n+1}$ using the two following intermediate steps:

(A) Prediction step: Find $\mathbf{p}^{n+1/2}\in\mathbf{Q}_4$ satisfying

$$\begin{aligned} &\int_\Omega \frac{\mathbf{p}^{n+1/2}-\mathbf{p}^n}{\Delta t} : \mathbf{q}\mathrm{d}\mathbf{x} + \int_\Omega \mathbf{p}^{n+1/2}:\mathbf{q}\mathrm{d}\mathbf{x}\\ &+\frac{1}{\varepsilon_2}\int_\Omega\Big[(|p_1^{n+1/2}|^2-1)p_1^{n+1/2}\cdot q_1 + (|p_2^{n+1/2}|^2-1)p_2^{n+1/2}\cdot q_2\\ &+\frac{1}{2}p_1^{n+1/2}\cdot p_2^{n+1/2}(p_2^{n+1/2}\cdot q_1+p_1^{n+1/2}\cdot q_2)\Big]\mathrm{d}\mathbf{x} = 0 \end{aligned} \tag{3.11}$$

for all $\mathbf{q} \in \mathbf{Q}_4$. It is worth noting that this problem does not involve any derivatives of the variable $\mathbf{p}$, a significant simplification, indeed (see section 3.4).

(B) Correction step: Find $\mathbf{p}^{n+1} \in \mathbf{Q}_4$ satisfying

$$\begin{aligned}(3.12)\qquad &\int_\Omega \frac{\mathbf{p}^{n+1} - \mathbf{p}^{n+1/2}}{\Delta t} : \mathbf{q} d\mathbf{x} + C \int_\Omega (\mathbf{B}\mathbf{p}^{n+1} + \mathbf{u_g} - \mathbf{f}) \cdot \mathbf{B}\mathbf{q} \, d\mathbf{x} \\ &+ \langle \partial I_\nabla(\mathbf{p}^{n+1}), \mathbf{q} \rangle = 0\end{aligned}$$

for all $\mathbf{q} \in \mathbf{Q}_4$. This problem is actually a (hidden) elliptic linear variational problem whose solution will be addressed in section 3.5.

3.4. Local optimization problems. The subproblem (3.11) that arises in the splitting algorithm does not involve any derivatives of the variable $\mathbf{q}$. Therefore, it can be solved pointwise a.e. in Ω (see also, e.g., [6, 7, 9] for other instances of similar simplifications). Suppose that

$$\mathbf{q} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}.$$

Using the above notation, a more explicit formulation of (3.11) reads as

$$(3.13)\qquad \begin{aligned} &(1+\Delta t)\, p_{11}^{n+1/2} + \frac{\Delta t \mu^{n+1/2} p_{11}^{n+1/2}}{\varepsilon_2} - p_{11}^n + \frac{\Delta t \kappa^{n+1/2} p_{21}^{n+1/2}}{2\ \varepsilon_2} = 0,\\ &(1+\Delta t)\, p_{12}^{n+1/2} + \frac{\Delta t \mu^{n+1/2} p_{12}^{n+1/2}}{\varepsilon_2} - p_{12}^n + \frac{\Delta t \kappa^{n+1/2} p_{22}^{n+1/2}}{2\ \varepsilon_2} = 0,\\ &(1+\Delta t)\, p_{21}^{n+1/2} + \frac{\Delta t \lambda^{n+1/2} p_{21}^{n+1/2}}{\varepsilon_2} - p_{21}^n + \frac{\Delta t \kappa^{n+1/2} p_{11}^{n+1/2}}{2\ \varepsilon_2} = 0,\\ &(1+\Delta t)\, p_{22}^{n+1/2} + \frac{\Delta t \lambda^{n+1/2} p_{22}^{n+1/2}}{\varepsilon_2} - p_{22}^n + \frac{\Delta t \kappa^{n+1/2} p_{12}^{n+1/2}}{2\ \varepsilon_2} = 0,\end{aligned}$$

where

$$\begin{aligned} \mu^{n+1/2} &:= \left(\left(p_{11}^{n+1/2}\right)^2 + \left(p_{12}^{n+1/2}\right)^2 - 1 \right),\\ \lambda^{n+1/2} &:= \left(\left(p_{21}^{n+1/2}\right)^2 + \left(p_{22}^{n+1/2}\right)^2 - 1 \right),\\ \kappa^{n+1/2} &:= \left(p_{11}^{n+1/2}\, p_{21}^{n+1/2} + p_{12}^{n+1/2}\, p_{22}^{n+1/2} \right).\end{aligned}$$

Actually, system (3.13) can be reformulated in a more condensed form. Let us denote $[p_{11}, p_{12}]^T$ by α and $[p_{21}, p_{22}]^T$ by β; then (3.13) becomes

(3.14)

$$\begin{aligned} (1+\Delta t)\alpha^{n+1/2} + \frac{\Delta t(|\alpha^{n+1/2}|^2 - 1)\alpha^{n+1/2}}{\varepsilon_2} + \frac{\Delta t(\alpha^{n+1/2} \cdot \beta^{n+1/2})\beta^{n+1/2}}{2\varepsilon_2} &= \alpha^n,\\ (1+\Delta t)\beta^{n+1/2} + \frac{\Delta t(|\beta^{n+1/2}|^2 - 1)\beta^{n+1/2}}{\varepsilon_2} + \frac{\Delta t(\alpha^{n+1/2} \cdot \beta^{n+1/2})\alpha^{n+1/2}}{2\varepsilon_2} &= \beta^n.\end{aligned}$$

The above nonlinear system consists of four cubic equations. It is solved using a Newton–Raphson method. Algorithm (3.11), (3.12) requires the condition $\Delta t \le \varepsilon_2$

for system (3.14) to have a unique solution and guarantee the convergence as $n \to +\infty$. One does not need safeguarding in practice.

In practice (see section 4), once a finite element discretization of Ω is constructed, (3.14) is solved pointwise on each element of the discretization. The number of systems to solve thus depends on the number of elements of the triangulation.

3.5. Variational problems. The subproblem (3.12) that arises in the splitting algorithm is a well-posed, classical, linear elliptic variational problem. In order to highlight this statement, let us consider the reverse change of variable and take $\nabla \mathbf{u}^{n+1} := \mathbf{p}^{n+1}$. Problem (3.12) can be rewritten as follows: Find $\mathbf{u}^{n+1} \in \mathbf{V_g}$ such that

$$\begin{aligned}&\varepsilon_1 \Delta t \int_\Omega (\nabla^2 \mathbf{u}^{n+1}) \cdot (\nabla^2 \mathbf{v}) d\mathbf{x} + \int_\Omega \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} d\mathbf{x} + C\Delta t \int_\Omega \mathbf{u}^{n+1} \cdot \mathbf{v} d\mathbf{x} \\ &= C\Delta t \int_\Omega \mathbf{f} \cdot \mathbf{v} d\mathbf{x} + \int_\Omega \mathbf{p}^{n+1/2} : \nabla \mathbf{v} d\mathbf{x}, \quad \mathbf{v} \in (H^2(\Omega) \cap H_0^1(\Omega))^2.\end{aligned} \tag{3.15}$$

Problem (3.15) is a fourth-order linear elliptic variational problem of the biharmonic type. We introduce a coupled problem with an auxiliary variable. The additional equation reads as follows: Find $\mathbf{w}^{n+1} \in \left(H_0^1(\Omega)\right)^2$ such that

$$\mathbf{w}^{n+1} = -\nabla^2 \mathbf{u}^{n+1} \quad \text{in } \Omega. \tag{3.16}$$

Aggregating (3.15) and (3.16) allows us to obtain a coupled second-order linear system that reads as follows: Find $(\mathbf{u}^{n+1}, \mathbf{w}^{n+1}) \in \mathbf{V_g} \times \left(H_0^1(\Omega)\right)^2$ such that

$$\left\{\begin{aligned}&\varepsilon_1 \Delta t \int_\Omega \nabla \mathbf{w}^{n+1} : \nabla \mathbf{v} d\mathbf{x} + \int_\Omega \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} d\mathbf{x} + C\Delta t \int_\Omega \mathbf{u}^{n+1} \cdot \mathbf{v} d\mathbf{x} \\ &= C\Delta t \int_\Omega \mathbf{f} \cdot \mathbf{v} d\mathbf{x} + \int_\Omega \mathbf{p}^{n+1/2} : \nabla \mathbf{v} d\mathbf{x}, \\ &\int_\Omega \nabla \mathbf{u}^{n+1} : \nabla \mathbf{q} d\mathbf{x} - \int_\Omega \mathbf{w}^{n+1} \cdot \mathbf{q} d\mathbf{x} = 0\end{aligned}\right. \tag{3.17}$$

for all $(\mathbf{v}, \mathbf{q}) \in \left(H_0^1(\Omega)\right)^2 \times \left(H_0^1(\Omega)\right)^2$.

Remark 3.3. It follows from (3.17) that the pair $(\mathbf{u}^{n+1}, \mathbf{w}^{n+1})$ verifies $\mathbf{w}^{n+1} = -\nabla^2 \mathbf{u}^{n+1}$ and

$$\left\{\begin{aligned}&-\varepsilon_1 \Delta t \nabla^2 \mathbf{w}^{n+1} + \mathbf{w}^{n+1} + C\Delta t \mathbf{u}^{n+1} = C\Delta t \mathbf{f} - \nabla \cdot \mathbf{p}^{n+1/2} && \text{in } \Omega, \\ &\mathbf{w}^{n+1} = \mathbf{0} && \text{on } \Gamma.\end{aligned}\right. \tag{3.18}$$

Anticipating on the space discretization (to be addressed in section 4), *boundary layer thickness* considerations suggest taking $\varepsilon_1 \Delta t$ of the order of h^2 in (3.18), that is, taking ε_1 of the order of $h^2/\Delta t$. The same conclusion can be reached by balancing, after space discretization, the main diagonals (or the largest eigenvalues) of the symmetric matrices approximating the operators $-\varepsilon_1 \Delta t \nabla^2$ and $\mathbf{I}$.

4. Finite element discretization.

4.1. Generalities. The space approximation of the time-stepping algorithm (3.11)–(3.12) is addressed with piecewise linear continuous finite elements. As mentioned earlier, the use of low-order finite elements is appropriate for problems such as

(2.1), due to the low regularity of the solution. Since the solutions of the orthogonal maps equation are piecewise affine, the set of folding lines describes the discontinuity set of the solution gradient, and higher-order methods will not bring additional accuracy in the neighborhood of these lines.

Let us denote by $h > 0$ a space discretization step, together with an associated *triangulation* $\mathcal{T}_h$ that satisfies the usual compatibility conditions (see, e.g., [28]). Let us denote by Σ_h the (finite) set of the vertices of $\mathcal{T}_h$, by N_h the number of elements in Σ_h, and by Σ_{0h} the subset of those elements in Σ_h not located on Γ (with $N_{0h} := \text{card}(\Sigma_{0h})$).

From the triangulation $\mathcal{T}_h$ we define the following finite element spaces:

$$\begin{aligned}\mathbf{V}_h &= \{\mathbf{v} \in (C^0(\overline{\Omega}))^2, \mathbf{v} \in (\mathbb{P}_1)^2 \quad \forall K \in \mathcal{T}_h\},\\ \mathbf{V}_{\mathbf{g},h} &= \{\mathbf{v} \in \mathbf{V}_h, \mathbf{v}(Q) = \mathbf{g}(Q) \quad \forall Q \text{ vertices of } \mathcal{T}_h \text{ belonging to } \Gamma\},\\ \mathbf{Q}_h &= \{\mathbf{q} \in (L^\infty(\Omega))^{2\times 2}, \mathbf{q}|_K \in \mathbb{R}^{2\times 2} \quad \forall K \in \mathcal{T}_h\},\end{aligned}$$

where $\mathbb{P}_1$ is the space of two-variable polynomials of degree ≤ 1. Note that the gradient of functions in $\mathbf{V}_h$ belongs to $\mathbf{Q}_h$. Next, we equip $\mathbf{V}_h$, and its subspaces $\mathbf{V}_{\mathbf{g},h}$, with the following discrete inner product (based on classical quadrature formulas):

$$(\mathbf{v}, \mathbf{w})_{0h} = \sum_{k=1}^{N_h} \sum_{i=1}^{m_k} W_i \mathbf{v}(\zeta_{\mathbf{i}}) \cdot \mathbf{w}(\zeta_i) \quad \forall \mathbf{v}, \mathbf{w} \in \mathbf{V}_h,$$

where W_i (resp., ζ_i) are the weights (resp., evaluation points) of a Gauss quadrature rule of order ≥ 2, and m_k is the number of quadrature points in the element k (supposed constant). The quadrature formulas we used are implemented in the library `libmesh` [37]. The corresponding norm is $||\mathbf{v}||_{0h} := \sqrt{(\mathbf{v}, \mathbf{v})_{0h}}$ for all $\mathbf{v} \in \mathbf{V}_h$. In a similar fashion, we equip the space $\mathbf{Q}_h$ with the inner product and norm respectively defined by

$$((\mathbf{p}, \mathbf{q}))_{0h} = \sum_{K \in \mathcal{T}_h} |K| \; \mathbf{p}|_K : \mathbf{q}|_K$$

and $|||\mathbf{q}|||_{0h} = \sqrt{((\mathbf{q}, \mathbf{q}))_{0h}}$ (with $|K|$ = area of K).

The discrete version of the numerical algorithm uses the same steps as the continuous version presented in section 3. However, let us sketch the main discrete milestones in what follows.

4.2. Discretization of the flow problem.

The variational formulation of the initial value problem (3.10) reads as follows: Find $\mathbf{p}_h(t) \in \mathbf{Q}_h$ for a.e. $t \in (0, +\infty)$ satisfying

$$\begin{aligned}&\int_\Omega \frac{\partial \mathbf{p}_h(t)}{\partial t} : \mathbf{q}_h d\mathbf{x} + \int_\Omega \mathbf{p}_h(t) : \mathbf{q}_h d\mathbf{x} + C \int_\Omega (\mathbf{B}\mathbf{p}_h(t) + \mathbf{u}_{\mathbf{g},h} - \mathbf{f}) \cdot \mathbf{B}\mathbf{q}_h d\mathbf{x}\\ &+ \langle \partial I_{\nabla_h}(\mathbf{p}_h(t)), \mathbf{q}_h \rangle\\ &+ \frac{1}{\varepsilon_2} \int_\Omega \Big[(|p_{1,h}(t)|^2 - 1) p_{1,h}(t) \cdot q_{1,h} + (|p_{2,h}(t)|^2 - 1) p_{2,h}(t) \cdot q_{2,h}\\ &+ \frac{1}{2} p_{1,h}(t) \cdot p_{2,h}(t) (p_{2,h}(t) \cdot q_{1,h} + p_{1,h}(t) \cdot q_{2,h}) \Big] d\mathbf{x} = 0 \qquad \forall \mathbf{q}_h \in \mathbf{Q}_h,\end{aligned} \tag{4.1}$$

together with the given initial condition $\mathbf{p}_h(0) = \mathbf{p}_{0,h}$. Here, we define

(i) the function $\mathbf{u}_{\mathbf{g},h} \in \mathbf{V}_{\mathbf{g},h}$ as the unique element of $\mathbf{V}_{\mathbf{g},h}$ verifying

$$\int_\Omega \nabla \mathbf{u}_{\mathbf{g},h} : \nabla \mathbf{v}_h \mathrm{d}\mathbf{x} = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_{\mathbf{0},h};$$

(ii) the function $\mathbf{B}\mathbf{q}_h$ as the unique element of $\mathbf{V}_{\mathbf{0},h}$ verifying

$$\int_\Omega \nabla \mathbf{B}\mathbf{q}_h : \nabla \mathbf{v}_h \mathrm{d}\mathbf{x} = \int_\Omega (\nabla \cdot \mathbf{q}_h) \cdot \mathbf{v}_h \mathrm{d}\mathbf{x} \quad \forall \mathbf{v}_h \in \mathbf{V}_{\mathbf{0},h};$$

(iii) the functional I_{∇_h} by

$$I_{\nabla_h}(\mathbf{q}_h) = \begin{cases} \dfrac{\varepsilon_1}{2}(\boldsymbol{\theta}_h(\mathbf{q}_h), \boldsymbol{\theta}_h(\mathbf{q}_h))_{0h} & \text{if } \mathbf{q}_h \in \nabla \mathbf{V}_{\mathbf{g},h}, \\ +\infty & \text{if } \mathbf{q}_h \in \mathbf{Q}_h \backslash \nabla \mathbf{V}_{\mathbf{g},h}, \end{cases}$$

where $\boldsymbol{\theta}_h(\mathbf{q}_h)$ is uniquely defined from $\mathbf{q}_h$ by

$$\begin{cases} \boldsymbol{\theta}_h(\mathbf{q}_h) \in \mathbf{V}_{\mathbf{0},h}, \\ (\boldsymbol{\theta}_h(\mathbf{q}_h), \varphi)_{0h} = ((\mathbf{q}_h, \nabla \varphi))_{0h} \quad \forall \varphi \in \mathbf{V}_{\mathbf{0},h}. \end{cases}$$

(iv) The initial condition $\mathbf{p}_{0,h} \in \mathbf{Q}_h$ is obtained as follows: First we calculate $\mathbf{u}_{0,h} \in \mathbf{V}_{\mathbf{g},h}$ verifying

$$\int_\Omega \nabla \mathbf{u}_{0,h} : \nabla \mathbf{v}_h d\mathbf{x} = \boldsymbol{\delta} \quad \forall \in \mathbf{v}_h \in \mathbf{V}_{\mathbf{0},h},$$

where $\boldsymbol{\delta} = (0.0005, 0.0005)^T$; then, $\mathbf{p}_{0,h}$ is calculated as the piecewise constant gradient $\nabla \mathbf{u}_{0,h}$ on each element $K \in \mathcal{T}_h$.

We apply the operator-splitting strategy (3.11), (3.12) to solve (4.1), and we denote by $\mathbf{p}_h^n$ the related approximations of $\mathbf{p}_h(t^n)$. Starting from the initial condition $\mathbf{p}_h^0 = \mathbf{p}_{0,h}$, we compute successively $\mathbf{p}_h^{n+1/2}$ and $\mathbf{p}_h^{n+1}$ via the following two intermediate steps:

(A) Prediction step: Find $\mathbf{p}_h^{n+1/2} \in \mathbf{Q}_h$ satisfying

$$\begin{aligned} &\int_\Omega \frac{\mathbf{p}_h^{n+1/2} - \mathbf{p}_h^n}{\Delta t} : \mathbf{q}_h \mathrm{d}\mathbf{x} + \int_\Omega \mathbf{p}_h^{n+1/2} : \mathbf{q}_h \mathrm{d}\mathbf{x} \\ &+ \frac{1}{\varepsilon_2} \int_\Omega \Big[(|p_{1,h}^{n+1/2}|^2 - 1) p_{1,h}^{n+1/2} \cdot q_{1,h} + (|p_{2,h}^{n+1/2}|^2 - 1) p_{2,h}^{n+1/2} \cdot q_{2,h} \\ &+ \frac{1}{2} p_{1,h}^{n+1/2} \cdot p_{2,h}^{n+1/2} (p_{2,h}^{n+1/2} \cdot q_{1,h} + p_{1,h}^{n+1/2} \cdot q_{2,h}) \Big] \mathrm{d}\mathbf{x} = 0 \end{aligned} \tag{4.2}$$

for all $\mathbf{q}_h \in \mathbf{Q}_h$.

(B) Correction step: Find $\mathbf{p}_h^{n+1} \in \mathbf{Q}_h$ satisfying

$$\begin{aligned} &\int_\Omega \frac{\mathbf{p}_h^{n+1} - \mathbf{p}_h^{n+1/2}}{\Delta t} : \mathbf{q}_h \mathrm{d}\mathbf{x} + C \int_\Omega (\mathbf{B}\mathbf{p}_h^{n+1} + \mathbf{u}_{\mathbf{g},h} - \mathbf{f}) \cdot \mathbf{B}\mathbf{q}_h \, \mathrm{d}\mathbf{x} \\ &+ \langle \partial I_{\nabla_h}(\mathbf{p}_h^{n+1}), \mathbf{q}_h \rangle = 0 \end{aligned} \tag{4.3}$$

for all $\mathbf{q}_h \in \mathbf{Q}_h$.

4.3. Solution of the discrete local optimization problems. The finite-dimensional nonlinear problem (4.2) can be solved *trianglewise*. Indeed, if $\mathbf{p}_h^{n+1/2} := \{\mathbf{p}_{K,h}^{n+1/2}\}_{K\in\mathcal{T}_h}$, one can rewrite (4.2) as follows: For each triangle $K \in \mathcal{T}_h$, solve

$$\begin{aligned}
&(1+\Delta t)\, p_{K,h,11}^{n+1/2} + \frac{\Delta t \mu_h^{n+1/2} p_{K,h,11}^{n+1/2}}{\varepsilon_2} - p_{K,h,11}^{n} + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,21}^{n+1/2}}{2\ \varepsilon_2} = 0,\\
&(1+\Delta t)\, p_{K,h,12}^{n+1/2} + \frac{\Delta t \mu_h^{n+1/2} p_{K,h,12}^{n+1/2}}{\varepsilon_2} - p_{K,h,12}^{n} + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,22}^{n+1/2}}{2\ \varepsilon_2} = 0,\\
&(1+\Delta t)\, p_{K,h,21}^{n+1/2} + \frac{\Delta t \lambda_h^{n+1/2} p_{K,h,21}^{n+1/2}}{\varepsilon_2} - p_{K,h,21}^{n} + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,11}^{n+1/2}}{2\ \varepsilon_2} = 0,\\
&(1+\Delta t)\, p_{K,h,22}^{n+1/2} + \frac{\Delta t \lambda_h^{n+1/2} p_{K,h,22}^{n+1/2}}{\varepsilon_2} - p_{K,h,22}^{n} + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,12}^{n+1/2}}{2\ \varepsilon_2} = 0,
\end{aligned} \tag{4.4}$$

where

$$\begin{aligned}
\mu_h^{n+1/2} &:= \left(\left(p_{K,h,11}^{n+1/2}\right)^2 + \left(p_{K,h,12}^{n+1/2}\right)^2 - 1\right),\\
\lambda_h^{n+1/2} &:= \left(\left(p_{K,h,21}^{n+1/2}\right)^2 + \left(p_{K,h,22}^{n+1/2}\right)^2 - 1\right),\\
\kappa_h^{n+1/2} &:= \left(p_{K,h,11}^{n+1/2}\, p_{K,h,21}^{n+1/2} + p_{K,h,12}^{n+1/2}\, p_{K,h,22}^{n+1/2}\right).
\end{aligned}$$

System (4.4) is similar to (3.13) and can be solved by Newton's techniques, taking $\mathbf{p}_{K,h}^n$ as an initial guess. When applied to the solution of problem (4.4), the Newton method always converged and never required more than 10 iterations for the test problems considered in section 5.

4.4. Solution of the discrete linear variational problems. Problem (4.3) is equivalent to the following: Find $(\mathbf{u}_h^{n+1}, \mathbf{w}_h^{n+1}) \in \mathbf{V}_{\mathbf{g},h} \times \mathbf{V}_{\mathbf{0},h}$ such that

$$\left\{\begin{aligned}
&\varepsilon_1 \Delta t((\nabla \mathbf{w}_h^{n+1}, \nabla \mathbf{v}_h))_{0h} + ((\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h))_{0h} + C\Delta t(\mathbf{u}_h^{n+1}, \mathbf{v}_h)_{0h}\\
&= C\Delta t(\mathbf{f}, \mathbf{v}_h)_{0h} + ((\mathbf{p}_h^{n+1/2}, \nabla \mathbf{v}_h))_{0h},\\
&((\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{q}_h))_{0h} - (\mathbf{w}_h^{n+1}, \mathbf{q}_h)_{0h} = 0
\end{aligned}\right. \tag{4.5}$$

for all $(\mathbf{v}_h, \mathbf{q}_h) \in \mathbf{V}_{\mathbf{0},h} \times \mathbf{V}_{\mathbf{0},h}$.

Remark 4.1. Let φ_i, $i = 1, \dots, N_h$, be the piecewise linear basis functions of $\mathbf{V}_h$. Note that, by defining the classical stiffness and mass matrices A and M by $A_{i,j} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j d\mathbf{x}$, $M_{i,j} = \int_\Omega \varphi_i\varphi_j d\mathbf{x}$, and suitable right-hand side $\mathbf{F}$, the linear system can be compactly written as

$$\begin{pmatrix} A + C\Delta t M & \varepsilon_1 \Delta t A \\ A & -M \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0} \end{pmatrix}.$$

Here we used a monolithic approach to solve this block-structured linear system.

Remark 4.2. Note here that we could also have rewritten (4.5) as

$$(4.6)\qquad \begin{cases} \varepsilon_1 \Delta t((\nabla \mathbf{w}_h^{n+1}, \nabla \mathbf{v}_h))_{0h} + (\mathbf{w}_h^{n+1}, \mathbf{q}_h)_{0h} + C\Delta t(\mathbf{u}_h^{n+1}, \mathbf{v}_h)_{0h} \\ = C\Delta t(\mathbf{f}, \mathbf{v}_h)_{0h} + ((\mathbf{p}_h^{n+1/2}, \nabla \mathbf{v}_h))_{0h}, \\ ((\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{q}_h))_{0h} - (\mathbf{w}_h^{n+1}, \mathbf{q}_h)_{0h} = 0 \end{cases}$$

for every $(\mathbf{v}_h, \mathbf{q}_h)$ belonging to $\mathbf{V}_{\mathbf{0},h} \times \mathbf{V}_{\mathbf{0},h}$. Then, using the previous matrix-vector notation, the linear system would have become

$$\begin{pmatrix} C\Delta t M & \varepsilon_1 \Delta t A + M \\ A & -M \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} F \\ \mathbf{0} \end{pmatrix}.$$

5. Numerical experiments.

In this section we are going to report on the results of numerical experiments aimed at assessing the robustness, the accuracy, and the efficiency of our methodology. The computational domain (i.e., the paper sheet to be folded) is chosen either as the unit square $\Omega = (0,1)^2$ or as the unit disk $\Omega = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 < 1\}$. All the experiments have been performed on a desktop computer with Intel Xeon E5-1650 (3.50 GHz × 12) and 64 GB memory.

We consider several discretizations of the computational domain, using various finite element meshes as shown in Figure 2. The influence of the chosen mesh on the solution obtained by the algorithm is also discussed. For all the numerical experiments we consider $\mathbf{f} = (0,0)^T$, $\varepsilon_2 = 5 \times 10^{-10}$, $\Delta t = \varepsilon_2/2$, $C = 10$, and $\varepsilon_1 = \frac{h^2}{5\Delta t}$ (unless stated otherwise). The stopping criterion we use to decide on the flow stationarity is either $n \leq 1000$ or $||\mathbf{p}^{n+1} - \mathbf{p}^n||_{L^2(\Omega)} \leq 5 \times 10^{-4}$ (unless stated otherwise). We observe numerically that for this stopping criterion, $||\mathbf{u}^{n+1} - \mathbf{u}^n||_{L^2(\Omega)}$ is of the order of 10^{-7}.

The choice of $\mathbf{f}$ allows us to give an a priori estimate of the solution. With another choice of $\mathbf{f}$, another solution may be obtained, and numerical experiments have shown some sensitivity with respect to the parameters $\mathbf{f}$ and C. The penalization constant ε_2 is chosen in order to guarantee that the orthogonality conditions are satisfied accurately. Values of ε_2 ranging from 10^{-5} to 10^{-10} are suitable. The number of iterations of the Newton method in section 3.4 may decrease when ε_2 is larger. The choice of ε_1 allows us to have a regularization term in (3.15) of the order of h^2. In light of these comments, a thorough sensitivity analysis has not been performed, but the influence of some parameters will be discussed in the numerical experiments.

5.1. Smooth validation example.

In the first experiment, the algorithm is validated with a boundary condition that corresponds to a smooth mapping, without any singularities. We consider

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} \quad \forall \mathbf{x} \in \partial\Omega.$$

An exact solution corresponding to this given boundary is the identity mapping $\mathbf{u}(\mathbf{x}) = \mathbf{x}$. This corresponds to an origami without any actual folding.

For such an example, when choosing $\varepsilon_1 = 0.0$ as a smoothing parameter, the problem is solved up to machine precision for all meshes. Figure 3 visualizes the graph of the two components of the computed approximate solution. Since the two components of the solution are piecewise affine, the continuous piecewise affine finite element approximation we advocated in section 4 is ideally suited to the solution of the problem under consideration.

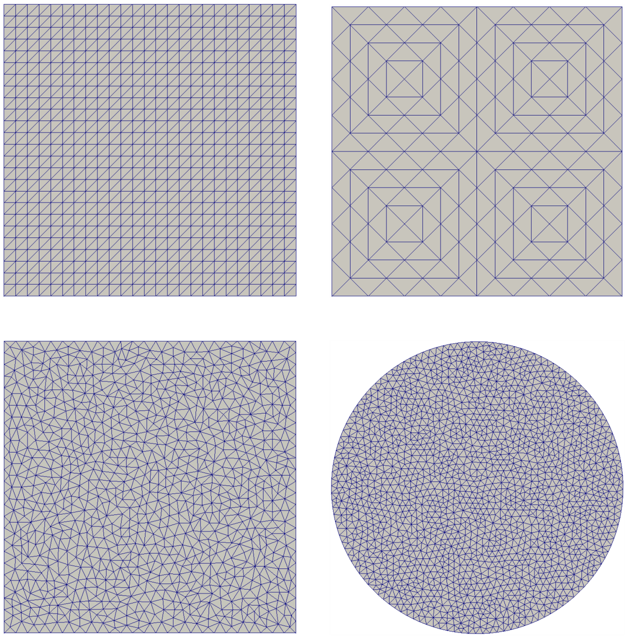

Fig. 2. *Finite element triangulations used for the numerical experiments. Top left: Structured asymmetric mesh ($\Omega = (0,1)^2$, $h = 0.04$). Top right: Structured symmetric mesh ($\Omega = (0,1)^2$, $h = 0.125$). Bottom left: Isotropic unstructured mesh ($\Omega = (0,1)^2$, $h \simeq 0.02$). Bottom right: Isotropic unstructured mesh ($\Omega = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 < 1\}$, $h \simeq 0.08$).*

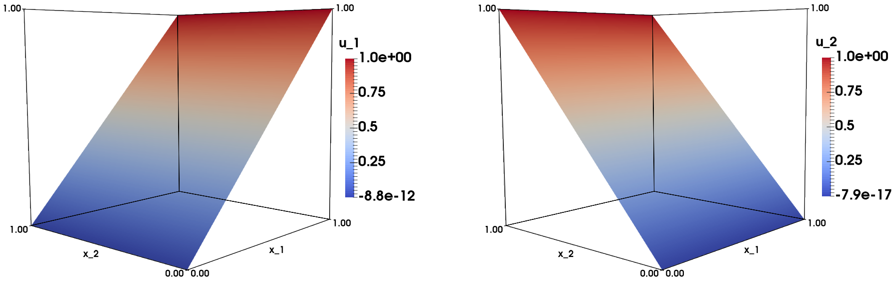

Fig. 3. *Smooth validation example. Graphs of the two components of the computed approximate solution of* (2.1) *with* $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ *on* $\partial\Omega$*. Left: Approximation of the first component* u_1*. Right: Approximation of the second component* u_2*. These results have been obtained using a structured asymmetric triangulation with* $h = 0.01$*.*

Note that a very accurate (up to machine precision) approximation of the solution can be obtained with ε_1 of the order of h^2, together with $\varepsilon_2 \simeq 10^{-7}$, and a stopping criterion of $||\mathbf{p}^{n+1} - \mathbf{p}^n||_{L^2(\Omega)} \leq 5 \times 10^{-10}$. Table 1 illustrates the results obtained with the same setup as described previously. It shows that orthogonality conditions are not jeopardized by the introduction of the well-chosen smoothing parameter.

TABLE 1

Smooth validation example ($\varepsilon_1 \simeq h^2$). (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2, 3, and 4).* (ii) *Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6).* (iii) *Variations with respect to h of the number of time steps necessary to achieve convergence (column 7). ($\Omega = (0,1)^2$, structured asymmetric meshes.)*

h	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h} \cdot \nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert \mathbf{u} - \mathbf{u}_h \rvert\rvert_{L^2}$	Rate	Iter
0.02	1.0	1.0	1.89e-6	2.35e-07	-	31
0.01	1.0	1.0	8.94e-7	1.04e-07	1.17	36
0.005	1.0	1.0	4.38e-7	4.93e-08	1.07	40
0.0025	1.0	1.0	2.46e-7	2.72e-08	0.85	43

5.2. Simple folding and mesh dependency. For the second experiment, we consider the unit square and the boundary data $\mathbf{g}$ given by

$$
\begin{aligned}
g_1(x_1,x_2) &= \begin{cases} x_1 & \text{if } x_1 < 0.5 \text{ and } x_2 = 0 \text{ or } x_2 = 1, \\ 1 - x_1 & \text{if } x_1 \geq 0.5 \text{ and } x_2 = 0 \text{ or } x_2 = 1, \\ 0 & \text{otherwise,} \end{cases} \\
g_2(x_1,x_2) &= x_2 \text{ on } \partial\Omega.
\end{aligned} \tag{5.1}
$$

In this case, the exact solution to (2.1) is given by $\mathbf{u} = (u_1, u_2)^T$, with

$$
\begin{aligned}
u_1(x_1,x_2) &= \begin{cases} x_1 & \text{if } x_1 < 0.5, \\ 1 - x_1 & \text{if } x_1 \geq 0.5, \end{cases} \\
u_2(x_1,x_2) &= x_2 \text{ in } \Omega.
\end{aligned} \tag{5.2}
$$

The function $\mathbf{u}$ defined by (5.2) corresponds to a single folding of the domain Ω along the middle line. This implies that the singularity is a line singularity along $x_1 = 0.5$. Thus, when choosing $\varepsilon_1 = 0$, the algorithm obtains a discretized solution that is accurate up to machine precision. In the case of the structured meshes of Figure 2, the mesh edges can be aligned with this line singularity or not, depending on whether the vertical mesh edges are along $x_1 = 0.5$. In the case of unstructured meshes, the edges are never aligned with the line $x_1 = 0.5$. Tables 2, 3, and 4 show results using $\varepsilon_1 = \frac{h^2}{5\Delta t}$. One can observe that the results are more accurate when using a structured mesh with edges aligned with the line singularity; when the line singularity is not aligned with the (structured) mesh edges, the behavior of the algorithm is actually similar to when the mesh is unstructured. Moreover, the number of iterations (time steps) needed to obtain a stationary solution is larger for the two latter cases. Finally, the convergence to the exact solution is superlinear when $h \to 0$ but more stable for structured meshes in Table 2.

We have visualized in Figure 4 (i) the graphs of the two components of the computed approximate solution (first row); (ii) the values of $\det \nabla \mathbf{u}$ (second row (left)); (iii) the image $\mathbf{u}(\Omega) \subset \mathbb{R}^2$ (second row (right)), which shows that the initial domain Ω is folded in half (note that this illustration is postprocessed from the computed approximations of u_1 and u_2).

5.3. Double diagonal folding. Let us consider now the unit square $\Omega = (0,1)^2$ and the boundary data defined by

$$
\mathbf{g}(x_1,x_2) = (0, |x_1 - x_2|), \qquad (x_1,x_2) \in \partial\Omega. \tag{5.3}
$$

TABLE 2

Simple folding. (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns* 2, 3, *and* 4*).* (ii) *Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns* 5 *and* 6*).* (iii) *Variations with respect to h of the number of time steps necessary to achieve convergence (column* 7*). ($\Omega = (0,1)^2$, structured asymmetric meshes, line singularity aligned with mesh edges.)*

h	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h} \cdot \nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert\mathbf{u} - \mathbf{u}_h\rvert\rvert_{L^2}$	Rate	Iter
0.02	0.9732	1.0	0.0028	1.87e-03	-	57
0.01	0.9866	1.0	0.0008	6.43e-04	1.54	65
0.005	0.9933	1.0	0.0002	2.22e-04	1.53	72
0.0025	0.9966	1.0	5.92e-05	7.76e-05	1.51	79

TABLE 3

Simple folding. (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns* 2, 3, *and* 4*).* (ii) *Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns* 5 *and* 6*).* (iii) *Variations with respect to h of the number of time steps necessary to achieve convergence (column* 7*). ($\Omega = (0,1)^2$, isotropic unstructured meshes.)*

h	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h} \cdot \nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert\mathbf{u} - \mathbf{u}_h\rvert\rvert_{L^2}$	Rate	Iter
0.05	0.9413	1.0	0.0167	6.20e-03	-	48
0.026	0.9700	1.0	0.0024	2.45e-03	1.33	96
0.013	0.9849	1.0	0.0011	1.10e-03	1.15	184
0.006	0.9925	1.0	0.0003	3.13e-04	1.82	117

TABLE 4

Simple folding. (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns* 2, 3, *and* 4*).* (ii) *Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns* 5 *and* 6*).* (iii) *Variations with respect to h of the number of time steps necessary to achieve convergence (column* 7*). ($\Omega = (0,1)^2$, structured asymmetric meshes, line singularity not aligned with mesh edges.)*

h	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h} \cdot \nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert\mathbf{u} - \mathbf{u}_h\rvert\rvert_{L^2}$	Rate	Iter
0.0196	0.9724	1.0	0.0094	4.72e-03	-	58
0.0099	0.9864	1.0	0.0048	2.44e-03	0.95	66
0.0049	0.9932	1.0	0.0024	1.25e-03	0.96	157
0.0024	0.9966	1.0	0.0027	4.39e-04	1.50	553

A corresponding exact solution is

$$\begin{aligned} u_1(x_1, x_2) &= d(\mathbf{x}, \partial\Omega) \quad \forall \mathbf{x} \text{ in } \Omega, \\ u_2(x_1, x_2) &= \begin{cases} \min(x_2, 1 - x_1) & \text{if } x_1 < x_2, \\ \min(x_1, 1 - x_2) & \text{otherwise.} \end{cases} \end{aligned} \tag{5.4}$$

For this test example, the line singularities are aligned with the two diagonals of Ω ($x_1 = x_2$ and $1 - x_1 = x_2$). When using the structured symmetric mesh (see Figure 2, top right) and $\varepsilon_1 = 0.0$, the algorithm obtains an approximated solution that is highly accurate, since the singularities are aligned with mesh edges. Indeed, after 170 iterations (time steps of the splitting algorithm), the numerical solution satisfies $||\mathbf{u} - \mathbf{u}_h||_{L_2(\Omega)} = 1.45 \times 10^{-8}$, $\int_\Omega |\nabla u_1| d\mathbf{x} = 1.0$, $\int_\Omega |\nabla u_2| d\mathbf{x} = 1.0$, and $\int_\Omega |\nabla u_1 \cdot \nabla u_2| d\mathbf{x} = 3.88 \times 10^{-8}$.

Numerical results are reported in Tables 5 and 6 for the structured asymmetric and unstructured triangulations, respectively. Similar convergence orders are observed

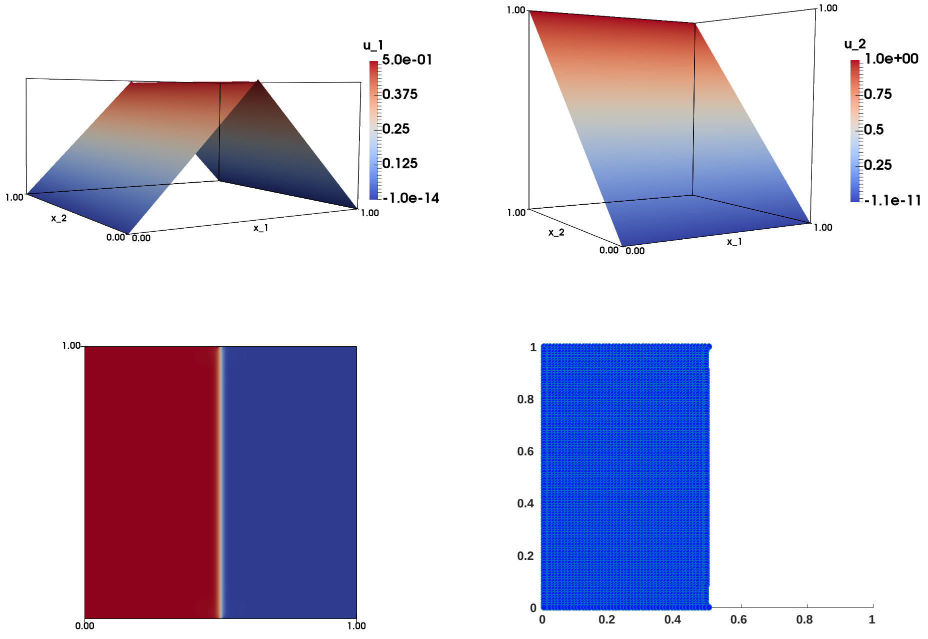

FIG. 4. *Simple folding. Visualization of the graph of the first and second components of the computed approximate solution* $\mathbf{u}_h$ *(*$= (u_{1,h}, u_{2,h})$*) of* (2.1) *with* $\Omega = (0,1)^2$ *and* $\mathbf{g}$ *given by* (5.1). *Top left: First component of the computed approximate solution* $u_{1,h}$*. Top right: Second component of the computed approximate solution* $u_{2,h}$*. Bottom left: Visualization of* $\det \nabla \mathbf{u}_h$*. Bottom right: Visualization of the domain* $\mathbf{u}_h(\Omega)$*.*

for both types of triangulations, since none of them match exactly the singularity of the gradient solution. Figure 5 illustrates the approximate solution $\mathbf{u}_h$. One can see that the two components actually satisfy the orthogonality conditions. Figure 5 (bottom left) visualizes $\det \nabla \mathbf{u}_h$ and shows that the angle condition (see Remark 2.2) is satisfied. Figure 5 (bottom right) visualizes the image $\mathbf{u}(\Omega)$ of Ω through the mapping $\mathbf{u}$. Table 7 illustrates the dependency of the solution with respect to the smoothing parameter ε_1. One observes that, as $\varepsilon_1 \to 0$, the $L^2(\Omega)$-approximation error decreases and the orthogonality properties verified by ∇u_1 and ∇u_2 are better satisfied by $\nabla u_{1,h}$ and $\nabla u_{2,h}$. On the other hand, the number of iterations necessary to achieve convergence increases as $\varepsilon_1 \to 0$. This numerical example shows that the introduction of the regularization term, such that $\varepsilon_1 \neq 0$, leads to, overall, better convergence properties of the time-stepping algorithm. Numerical results have consistently shown that the introduction of this term not only helps the convergence of the time-stepping algorithm towards a stationary solution, but also allows us to reduce drastically the number of time iterations in some cases.

The sensitivity of the results with respect to the target function $\mathbf{f}$ is now investigated. When modifying the target function $\mathbf{f}$, the algorithm is able to track different solutions of the orthogonal maps problem. Let us consider $\mathbf{f}_1(x_1, x_2) = (-2000, 0)^T$. Figure 6 (left column) illustrates snapshots of the two components of the numerical solution $\mathbf{u}_h$, for $h = 0.01$, and of the determinant $\det \nabla \mathbf{u}_h$, and shows that the first component is the concave up version of the one illustrated in Figure 6 when $\mathbf{f} = \mathbf{0}$,

TABLE 5

Double diagonal folding. (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns* 2, 3, *and* 4*).* (ii) *Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u}-\mathbf{u}_h$ and related convergence orders (columns* 5 *and* 6*).* (iii) *Variations with respect to h of the number of time steps necessary to achieve convergence (column* 7*). ($\Omega=(0,1)^2$, structured asymmetric meshes.)*

h	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h}\cdot\nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert\mathbf{u}-\mathbf{u}_h\rvert\rvert_{L^2}$	Rate	Iter
0.02	0.9672	0.9615	0.0710	3.86e-03	-	64
0.01	0.9837	0.9804	0.0370	1.55e-03	1.31	71
0.005	0.9918	0.9901	0.0189	6.58e-04	1.23	75
0.0025	0.9959	0.9950	0.0095	2.95e-04	1.15	84

TABLE 6

Double diagonal folding. (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns* 2, 3, *and* 4*).* (ii) *Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u}-\mathbf{u}_h$ and related convergence orders (columns* 5 *and* 6*).* (iii) *Variations with respect to h of the number of time steps necessary to achieve convergence (column* 7*). ($\Omega=(0,1)^2$, isotropic unstructured meshes.)*

h	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h}\cdot\nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert\mathbf{u}-\mathbf{u}_h\rvert\rvert_{L^2}$	Rate	Iter
0.05	0.9346	0.9279	0.1305	9.93e-03	-	61
0.026	0.9677	0.9618	0.0724	3.93e-03	1.33	83
0.013	0.9840	0.9805	0.0393	1.73e-03	1.18	313
0.006	0.9920	0.9901	0.0208	7.63e-04	1.18	292

while the second component remains unchanged. The numerical solution satisfies

$$\int_\Omega |\nabla u_1| d\mathbf{x} = 1.01, \quad \int_\Omega |\nabla u_2| d\mathbf{x} = 0.98, \quad \int_\Omega |\nabla u_1\cdot\nabla u_2| d\mathbf{x} = 0.047$$

after 120 iterations. A second variation in the fidelity term is illustrated by a perturbation of the target function $\mathbf{f}$ around the null function $\mathbf{0}$, through a local perturbation around the midpoint $(0.5,0.5)$. Figure 6 (right column) illustrates the snapshots of the two components of the numerical solution $\mathbf{u}_h$, and of the determinant $\det\nabla\mathbf{u}_h$, obtained when

$$\mathbf{f}_2(x_1,x_2) = \begin{cases} -2000 & \text{if } (x_1-0.5)^2+(x_2-0.5)^2 \le 2h, \\ 0 & \text{otherwise,} \end{cases}$$

where $h=0.01$ is the mesh size. The numerical solution satisfies

$$\int_\Omega |\nabla u_1| d\mathbf{x} = 0.96, \quad \int_\Omega |\nabla u_2| d\mathbf{x} = 0.97, \quad \int_\Omega |\nabla u_1\cdot\nabla u_2| d\mathbf{x} = 0.040$$

after 615 iterations. We can observe that the local perturbation of the target function induces a local perturbation of the approximation of the first component of the solution, which corresponds to an additional folding in the mapping. The sensitivity of the solution with respect to the target function is thus limited to the neighborhood of the perturbation. This is a consequence of the low regularity of the piecewise affine solutions.

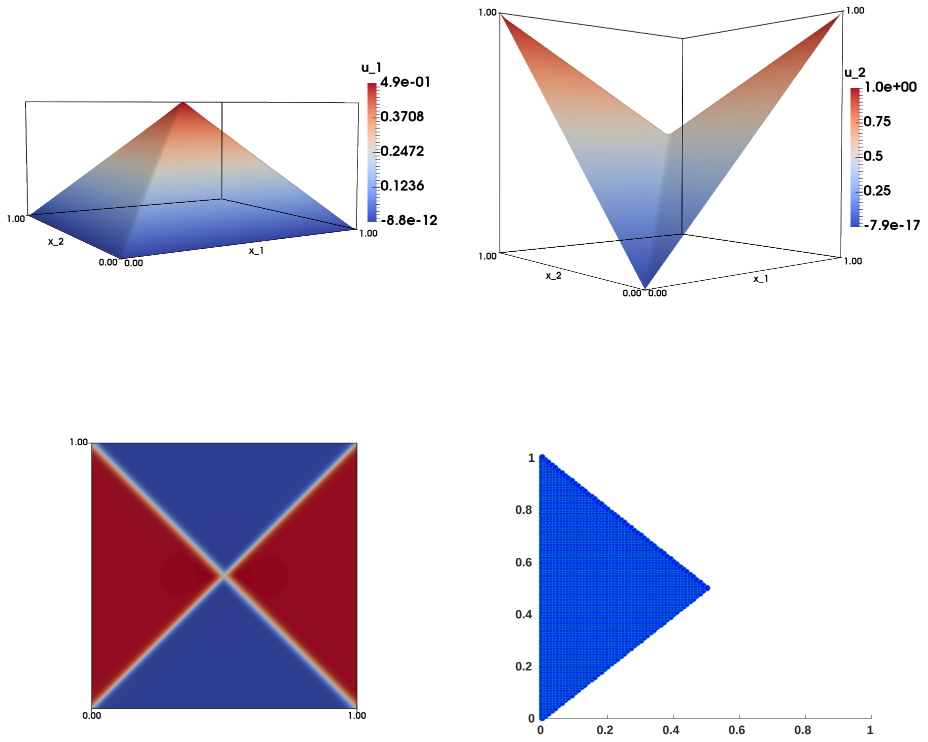

Fig. 5. *Double diagonal folding. Snapshots of the numerical stationary solution of* (2.1) *with* $\mathbf{g}$ *given by* (5.3). *Top left: First component* u_1. *Top right: Second component* u_2. *Bottom left: Visualization of* $\det \nabla \mathbf{u} = \pm 1$. *Bottom right: Visualization of the domain* $\mathbf{u}(\Omega)$.

Table 7

Double diagonal folding. (i) *Variations with respect to* ε_1 *of the approximate orthogonality conditions verified by* $\nabla u_{1,h}$ *and* $\nabla u_{2,h}$ *(columns* 2, 3, *and* 4*).* (ii) *Variations with respect to* ε_1 *of the* $L^2(\Omega)$ *norm of the computed approximation error* $\mathbf{u}-\mathbf{u}_h$ *and related convergence orders (column* 5*).* (iii) *Variations with respect to* ε_1 *of the number of time steps necessary to achieve convergence (column* 6*).* *(*$\Omega = (0,1)^2$*, structured asymmetric mesh,* $h = 0.01$.*)*

ε_1	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h} \cdot \nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert\mathbf{u} - \mathbf{u}_h\rvert\rvert_{L^2}$	Iter
2×10^{-3}	0.8337	0.8554	0.2242	2.99e-02	37
10^{-3}	0.8873	0.8880	0.1878	1.87e-02	41
2×10^{-4}	0.9517	0.9444	0.1023	6.32e-03	52
10^{-4}	0.9658	0.9595	0.0758	4.03e-03	55
0.0	0.9960	0.9965	0.0142	1.84e-03	413

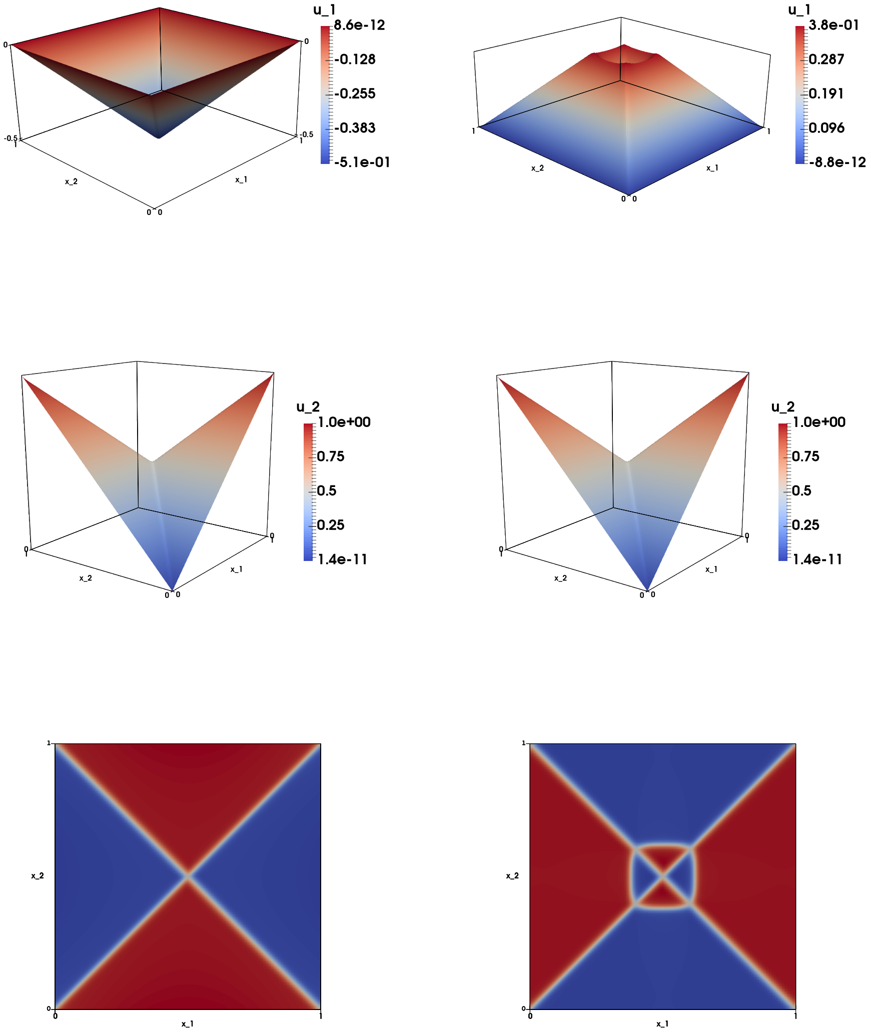

Fig. 6. *Double diagonal folding. Snapshots of the numerical stationary solution of* (2.1) *with* $\mathbf{g}$ *given by* (5.3) *and nonzero target functions* $\mathbf{f}$ *(first row: first component,* $u_{1,h}$*; second row: second component,* $u_{2,h}$*; third row:* $\det\nabla\mathbf{u}_h$*). Left column: Numerical results when* $\mathbf{f}=\mathbf{f}_1$*. Right column: Numerical results when* $\mathbf{f}=\mathbf{f}_2$*.*

5.4. A nonsmooth example with a point singularity. For this example, we have $\Omega=(0,1)^2$ and the boundary data $\mathbf{g}$ defined by

$$
\begin{aligned}
g_1(x_1,x_2) &= \begin{cases} \min(x_1, 1-x_1) & \text{if } x_2 = 1,\\ 0 & \text{otherwise,}\end{cases}\\
g_2(x_1,x_2) &= \begin{cases} x_2 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5,\\ x_2 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1+1,\\ x_1 & \text{if } x_2 \leq x_1 \text{ and } x_1 \leq 0.5,\\ 1-x_1 & \text{if } x_2 > x_1 \text{ and } x_2 \leq -x_1+1.\end{cases}
\end{aligned}
\tag{5.5}
$$

An exact solution to the related problem (2.1) is

$$
\begin{aligned}
u_1(x_1,x_2) &= \begin{cases} x_1 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5,\\ 1-x_1 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1+1,\\ x_2 & \text{if } x_2 \leq x_1 \text{ and } x_2 \leq -x_1+1,\end{cases}\\
u_2(x_1,x_2) &= \begin{cases} x_2 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5,\\ x_2 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1+1,\\ x_1 & \text{if } x_2 \leq x_1 \text{ and } x_1 \leq 0.5,\\ 1-x_1 & \text{if } x_2 > x_1 \text{ and } x_2 \leq -x_1+1.\end{cases}
\end{aligned} \tag{5.6}
$$

The main numerical difficulty here is the point singularity, located at $(0.5, 0.5)$. In the neighborhood of this point, the numerical approximation of $\det \nabla \mathbf{u}$ oscillates between the values $+1$ and -1, which increases significantly the number of iterations required to achieve convergence.

However, using the structured symmetric mesh, the line singularities are again aligned with the mesh edges, and a very accurate solution is obtained. Indeed, with $\varepsilon_1 = 0$, and after 130 time steps, we have

$$
\begin{aligned}
&||\mathbf{u}-\mathbf{u}_h||_{L^2(\Omega)} = 2.77 \cdot 10^{-10},\\
&\int_\Omega |\nabla u_{1,h}| d\mathbf{x} = 1.0, \quad \int_\Omega |\nabla u_{2,h}| d\mathbf{x} = 1.0, \quad \int_\Omega |\nabla u_{1,h} \cdot \nabla u_{2,h}| d\mathbf{x} = 1.25 \cdot 10^{-9}.
\end{aligned}
$$

We obtained the results reported in Table 8 using structured asymmetric meshes. These results show a convergence of order 0.9–1.0, approximately. One observes also that the number of time steps required to achieve quasi-stationarity increases drastically with $1/h$; this is due to the oscillatory behavior of function $t \to \mathbf{u}_h(t)$. Various geometrical aspects of the computed approximate solution have been visualized in Figure 7. In particular, Figure 7 (bottom left) shows that $\det \nabla \mathbf{u}$ satisfies accurately the angle condition; however, accuracy deteriorates close to the singular point $(0.5, 0.5)$.

TABLE 8

Nonsmooth example with point singularity. (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns* 2, 3, *and* 4*).* (ii) *Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns* 5 *and* 6*).* (iii) *Variations with respect to h of the number of time steps necessary to achieve convergence (column* 7*). ($\Omega = (0,1)^2$, structured asymmetric meshes.)*

h	$\int_\Omega \lvert\nabla u_{1,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{2,h}\rvert d\mathbf{x}$	$\int_\Omega \lvert\nabla u_{1,h} \cdot \nabla u_{2,h}\rvert d\mathbf{x}$	$\lvert\lvert\mathbf{u}-\mathbf{u}_h\rvert\rvert_{L^2}$	Rate	Iter
0.02	0.9703	0.9667	0.0422	5.71e-03	-	333
0.01	0.9852	0.9831	0.0219	2.89e-03	0.98	675
0.005	0.9926	0.9915	0.0113	1.52e-03	0.93	1438
0.0025	0.9963	0.9957	0.0058	7.88e-04	0.94	3316

5.5. Curved boundaries. Finally, in order to investigate the capabilities of our methodology at handling domains Ω with curved boundaries, we consider the particular problem (2.1), where Ω is the open unit disk centered at $(0, 0)$, and $\mathbf{g}$ is the restriction to $\partial\Omega$ of the vector-valued function $\mathbf{u}$ defined by

$$
\begin{aligned}
u_1(x_1,x_2) &= \begin{cases} x_1 & \text{if } x_1 < 0.5,\\ 1-x_1 & \text{if } x_1 \geq 0.5,\end{cases}\\
u_2(x_1,x_2) &= \begin{cases} x_2 & \text{if } x_2 < 0.5,\\ 1-x_2 & \text{if } x_2 \geq 0.5,\end{cases}
\end{aligned} \tag{5.7}
$$

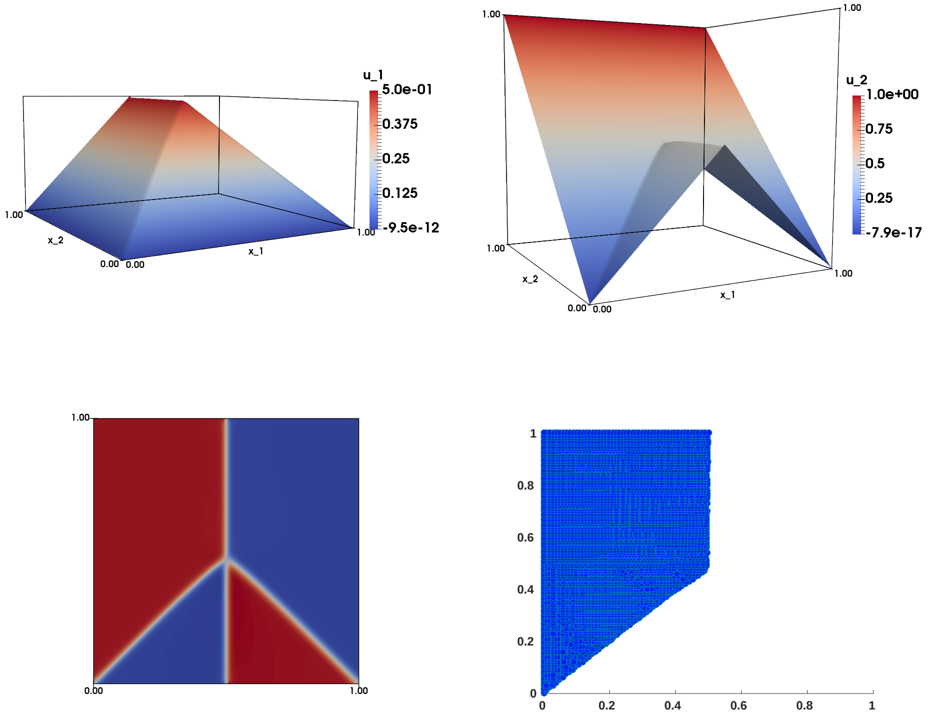

FIG. 7. *Nonsmooth example with point singularity. Visualization of the solution* $\mathbf{u}_h$ *of problem* (2.1) *with* $\mathbf{g}$ *given by* (5.5)*. Top left: Component* $u_{1,h}$*. Top right: Component* $u_{2,h}$*. Bottom left: Visualization of* $\det \nabla \mathbf{u}_h$*. Bottom right: Visualization of the domain* $\mathbf{u}_h(\Omega)$*.*

with $(x_1, x_2) \in \Omega$. The function $\mathbf{u}$ we just defined is an exact solution to the above problem. Actually, this solution corresponds to a double folding: one folding along the horizontal diameter of Ω (Ox_1 axis) and one folding along the vertical diameter of Ω (Ox_2 axis). We have reported in Table 9, for various values of h, the results we obtained, taking $\varepsilon_1 = \frac{h^2}{5\Delta t}$. These results (obtained using isotropic unstructured meshes) show first-order convergence (actually, slightly better than first). They show also that the gradients of the components $u_{1,h}$ and $u_{2,h}$ of $\mathbf{u}_h$ verify accurately the orthogonality properties of $\nabla u_{1,h}$ and $\nabla u_{2,h}$. Geometrical aspects of $\mathbf{u}_h$ have been visualized in Figure 8, $\mathbf{u}_h(\Omega_h)$ in particular (in Figure 8 (bottom right)), Ω_h being a polygonal approximation of Ω.

6. The Dirichlet problem with homogeneous boundary conditions. In this section, we will elaborate on the solution of the orthogonal maps problem (2.1) with homogeneous Dirichlet boundary conditions: Find a vector-valued function $\mathbf{u}(= [u_1, u_2]^T) : \Omega \to \mathbb{R}^2$ verifying

$$(6.1) \qquad \begin{cases} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega. \end{cases}$$

The choice of homogeneous boundary conditions actually corresponds to the case where the image of the whole boundary $\partial\Omega$ is the single point $\mathbf{0} = (0, 0)$. If the

TABLE 9

Double folding of the unit disk. (i) *Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns* 2, 3, *and* 4*).* (ii) *$L^2(\Omega)$ norm of the approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence rates (columns* 5 *and* 6*).* (iii) *Number of iterations necessary to achieve convergence (column* 7*). These results have been obtained using an isotropic unstructured triangulation of the unit disk.*

h	$\int_\Omega \|\nabla u_{1,h}\|d\mathbf{x}$	$\int_\Omega \|\nabla u_{2,h}\|d\mathbf{x}$	$\int_\Omega \|\nabla u_{1,h} \cdot \nabla u_{2,h}\|d\mathbf{x}$	$\|\|\mathbf{u} - \mathbf{u}_h\|\|_{L^2}$	Rate	Iter
0.05	0.9623	0.9623	0.0719	1.27e-02	-	75
0.025	0.9807	0.9807	0.0430	5.76e-03	1.03	379
0.013	0.9902	0.9902	0.0217	2.72e-03	1.08	557
0.006	0.9949	0.9949	0.0110	1.11e-03	1.29	427

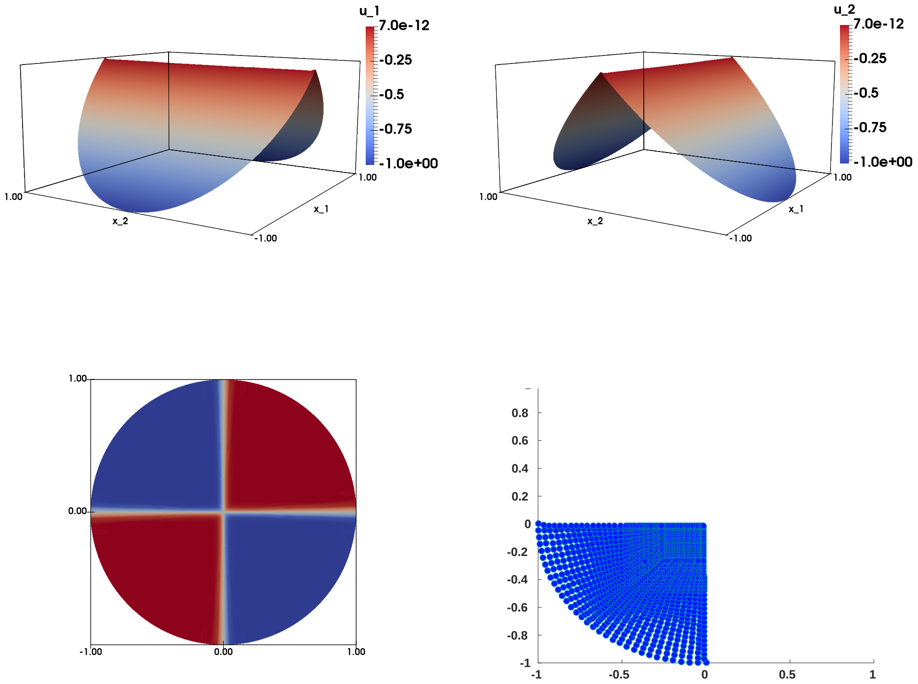

FIG. 8. *Double folding of the unit disk. Visualization of the computed approximate solution $\mathbf{u}_h = (u_{1,h}, u_{2,h})$ of problem* (2.1), (5.3), *with Ω being the unit disk centered at* (0, 0)*. Top left: Graph of $u_{1,h}$. Top right: Graph of $u_{2,h}$. Bottom left: Visualization of* $\det \nabla \mathbf{u}_h$*. Bottom right: Visualization (postprocessing) of the domain $\mathbf{u}_h(\Omega_h)$. These results have been obtained using an isotropic unstructured triangulation with $h = 0.08$.*

condition $\nabla\mathbf{u}\nabla\mathbf{u}^T = \mathbf{I}$ is incompatible with the homogeneous boundary condition (as is the case if Ω is a rectangle), the solution of the regularized least-squares problem (3.1) will develop a fractal behavior near the boundary Ω [16, 20], in a similar fashion as in [8, 14, 30] for a scalar Eikonal equation. Such a fractal behavior makes the convergence of our time-stepping method very slow or can even prevent it, which makes this example a very stringent one. Actually, in this case, the introduction of the regularization term with $\varepsilon_1 \neq 0$ becomes crucial to ensure convergence of the algorithm, smooth the fractal behavior of the solution, and to avoid the blow-up of the number of time steps.

One first approach would be to fold the domain Ω into a segment, e.g., $\{0\}\times[0,1]$. This could be achieved by repeatedly folding along the axis Ox_2 (*sequential folding*). Then the procedure can be completed by repeatedly folding the obtained segment along the other axis Ox_1. However, let us remark that the limit solution obtained when proceeding like this iteratively is not a solution to (6.1), as the solution mapping is the null mapping, with a range restricted to the point $(0, 0)$.

Figure 9 illustrates snapshots of the results (first component, $u_{1,h}$; second component, $u_{2,h}$) when folding once, twice, and three times along the axis Ox_2 only. We use a uniform asymmetric triangulation with $h = 1/160$, $\varepsilon_2 = 5\times 10^{-12}$, and $\varepsilon_1 = h^2/(5\Delta t)$. We note that the numerical solution starts to present some instabilities when folding several times. This issue will be addressed in section 7.

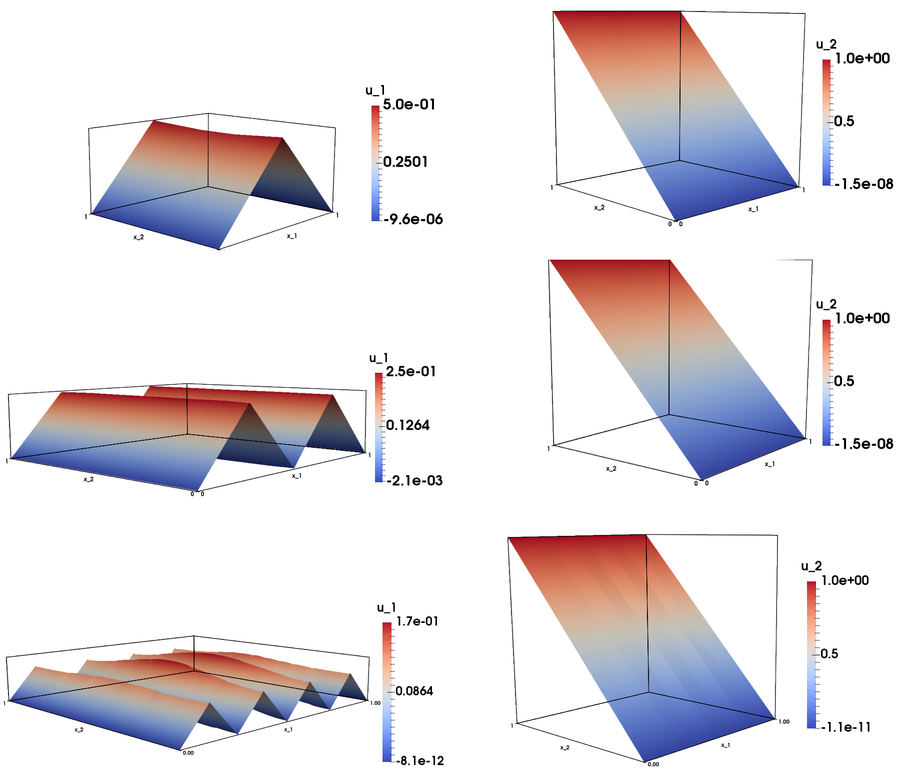

FIG. 9. *Sequential folding for the construction of a sequence of numerical approximations of* (2.1)*, with boundary data converging to* $\mathbf{g} = \mathbf{0}$*. Snapshots of the numerical stationary solution (left: first component,* $u_{1,h}$*; right: second component,* $u_{2,h}$*), when folding once, twice, or three times along one axis (first through third row).*

In a second step, we focus on the case with homogeneous Dirichlet boundary conditions (6.1). Figure 10 visualizes the results obtained by the methodology discussed in sections 2–4, using a uniform asymmetric triangulation with $h = 1/200$ (thus one has 40,401 vertices and 80,000 triangles). A close inspection shows that the angle condition is not satisfied everywhere and that ε_1 induces a smoothing of the lines of discontinuities of the singular set.

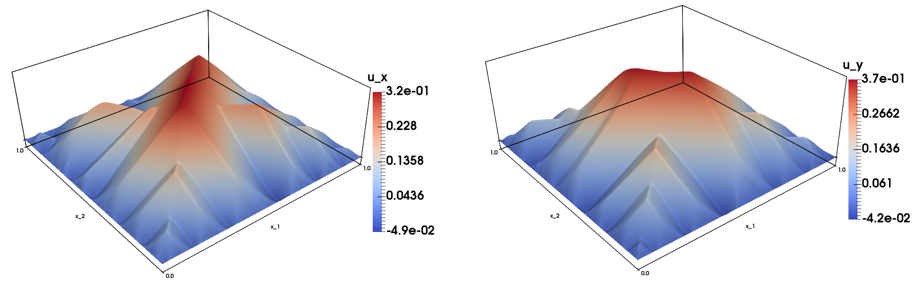

FIG. 10. *Homogeneous orthogonal maps-Dirichlet problem* (6.1) *with* $\Omega = (0,1)^2$. *Visualization of the approximate solutions (left: component* $u_{1,h}$*; right: component* $u_{2,h}$*). Results obtained using a uniform asymmetric triangulation with* 40,401 *vertices and* 80,000 *triangles.*

7. On the perspective of using adaptivity methods. Finally, in this section, we will illustrate the interest of using adaptivity techniques for the solution of the orthogonal maps problem (2.1). In order to improve convergence (and other properties), one can consider modifying the finite element approximation. A possible (and classical) approach, based on error estimates for the Poisson problem introduced in [38, 39, 40], is to adapt the finite element mesh to better follow the line singularities and better approximate, via mesh refinement, the oscillatory behavior of the solution near the boundary $\partial\Omega$.

Figure 11 visualizes the results obtained via mesh adaptivity for the sequential folding problem introduced in the previous section, when folding three times along the axis Ox_2 and once along the axis Ox_1 ($\varepsilon_2 = 5 \times 10^{-12}$ and $\varepsilon_1 = 0$). We start from a coarse isotropic unstructured mesh with 540 vertices and 998 triangles and end up with an unstructured triangulation with 501 vertices and 895 triangles. One can observe the better accuracy (sharpness of the folding lines) due to the space adaptivity approach.

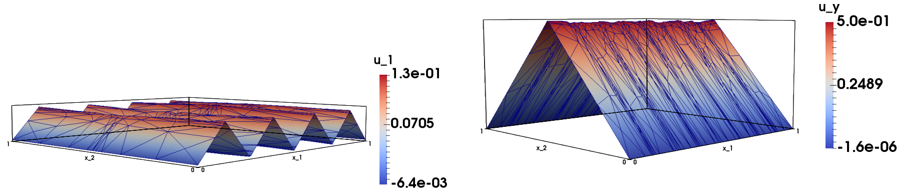

FIG. 11. *Sequential folding for the construction of a sequence of numerical approximations of* (2.1)*, with boundary data converging to* $\mathbf{g} = \mathbf{0}$ *(Figure* 9 *continued). Snapshots of the numerical stationary solution (left: first component,* $u_{1,h}$*; right: second component,* $u_{2,h}$*) when folding three times along one axis and once along the other axis.*

Figure 12 visualizes the results obtained with homogeneous Dirichlet boundary conditions (6.1), via mesh adaptivity, starting from a coarse isotropic unstructured mesh with 266 vertices and 154 triangles; we end up with an unstructured triangulation with 22,871 vertices and 42,987 triangles. The adaptive method smooths some of the oscillations and allows a sharper tracking of the lines of discontinuities of the singular set.

Therefore, mesh adaptation looks promising for orthogonal maps problems, and

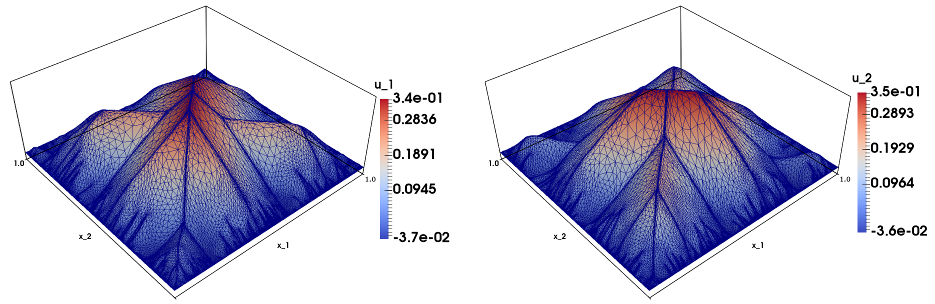

FIG. 12. *Homogeneous orthogonal maps-Dirichlet problem* (6.1) *with* $\Omega = (0,1)^2$. *Visualization of the approximate solutions computed with mesh adaptation. Left column: Component* $u_{1,h}$. *Right column: Component* $u_{2,h}$. *Results obtained using an unstructured adapted triangulation with* 22,871 *vertices and* 42,987 *triangles.*

dedicated (anisotropic) adaptive methods will be investigated in the future.

8. Conclusions and perspectives.

We have discussed an operator-splitting/finite element methodology for the numerical solution of the Dirichlet problem for orthogonal maps. This methodology is based on a variational principle, the introduction of the associated flow problem, and a time-stepping splitting algorithm. The results reported in section 5 show the robustness and the flexibility of this methodology and its ability to approximate solutions with line singularities on convex domains, with convergence orders close to 1 for the $L^2(\Omega)$ norm of the approximation error.

Preliminary numerical experiments have illustrated that for some stringent instances (e.g., with homogeneous boundary data), a fractal behavior of the solution next to the domain boundary appears, and the approximation of the solution is more difficult to obtain. An adaptive mesh algorithm approach allows us to obtain a more accurate solution, but also ensures the convergence of the time-stepping algorithm. Such nontrivial approaches will be investigated in the future.

Acknowledgments. The authors thank Prof. B. Dacorogna (EPFL) and Prof. Marcellini (University of Firenze) for fruitful discussions, and the two anonymous referees for constructive remarks.

REFERENCES

[1] Z. Abel, E. D. Demaine, M. L. Demaine, D. Eppstein, A. Lubiw, and R. Uehara, *Flat foldings of plane graphs with prescribed angles and edge lengths*, J. Comput. Geom., 9 (2018), pp. 74–93, https://doi.org/10.20382/jocg.v9i1a3.

[2] B. M. Afkham, A. Bhatt, B. Haasdonk, and J. S. Hesthaven, *Symplectic Model-Reduction with a Weighted Inner Product*, preprint, https://arxiv.org/abs/1803.07799, 2018.

[3] S. Bartels, A. Bonito, and R. H. Nochetto, *Bilayer plates: Model reduction, Γ-convergent finite element approximation, and discrete gradient flow*, Comm. Pure Appl. Math., 70 (2017), pp. 547–589, https://doi.org/10.1002/cpa.21626.

[4] S. Basterrechea and B. Dacorogna, *Existence of solutions for Jacobian and Hessian equations under smallness assumptions*, Numer. Funct. Anal. Optim., 35 (2014), pp. 868–892, https://doi.org/10.1080/01630563.2014.895746.

[5] A. Caboussat and R. Glowinski, *A penalty-regularization-operator splitting method for the numerical solution of a scalar Eikonal equation*, Chin. Ann. Math. Ser. B, 36 (2015), pp. 659–688, https://doi.org/10.1007/s11401-015-0930-8.

[6] A. Caboussat and R. Glowinski, *An alternating direction method of multipliers for the numerical solution of a fully nonlinear partial differential equation involving the Jacobian*

determinant, SIAM J. Sci. Comput., 40 (2018), pp. A52–A80, https://doi.org/10.1137/16M1094075.

[7] A. Caboussat, R. Glowinski, and D. Gourzoulidis, *A least-squares/relaxation method for the numerical solution of the three-dimensional elliptic Monge-Ampère equation*, J. Sci. Comput., 77 (2018), pp. 53–78, https://doi.org/10.1007/s10915-018-0698-6.

[8] A. Caboussat, R. Glowinski, and T.-W. Pan, *On the numerical solution of some Eikonal equations: An elliptic solver approach*, Chin. Ann. Math. Ser. B, 36 (2015), pp. 689–702, https://doi.org/10.1007/s11401-015-0971-z.

[9] 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, https://doi.org/10.1051/cocv/2012033.

[10] L. A. Caffarelli and M. G. Crandall, *Distance functions and almost global solutions of Eikonal equations*, Comm. Partial Differential Equations, 35 (2010), pp. 391–414, https://doi.org/10.1080/03605300903253927.

[11] L. A. Caffarelli and X. Cabré, *Fully Nonlinear Elliptic Equations*, American Mathematical Society, 1995, https://doi.org/10.1090/coll/043.

[12] L. A. 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, https://doi.org/10.1515/JNUM.2008.009.

[13] M. G. Crandall, L. C. Evans, and P.-L. Lions, *Some properties of viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc., 282 (1984), pp. 487–502, https://doi.org/10.1090/S0002-9947-1984-0732102-X.

[14] B. Dacorogna, R. Glowinski, Y. Kuznetzov, and T.-W. Pan, *On a conjugate gradient/Newton/penalty method for the solution of obstacle problems. Application to the solution of an Eikonal system with Dirichlet boundary conditions*, in Conjugate Gradient Algorithms and Finite Element Methods, M. Křížek, P. Neittaanmäki, S. Korotov, and R. Glowinski, eds., Springer-Verlag, Berlin, Heidelberg, 2004, pp. 263–283, https://doi.org/10.1007/978-3-642-18560-1_17.

[15] B. Dacorogna, R. Glowinski, and T.-W.Pan, *Numerical methods for the solution of a system of Eikonal equations with Dirichlet boundary conditions*, C. R. Acad. Sci. Paris Sér. I, 336 (2003), pp. 511–518, https://doi.org/10.1016/S1631-073X(03)00024-4.

[16] B. Dacorogna and P. Marcellini, *Implicit Partial Differential Equations*, Birkhaüser, Basel, 1999, https://doi.org/10.1007/978-1-4612-1562-2.

[17] B. Dacorogna, P. Marcellini, and E. Paolini, *An explicit solution to a system of implicit differential equations*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 25 (2008), pp. 163–171, https://doi.org/10.1016/j.anihpc.2006.11.007.

[18] B. Dacorogna, P. Marcellini, and E. Paolini, *Lipschitz-continuous local isometric immersions: Rigid maps and origami*, J. Math. Pures Appl. (9), 90 (2008), pp. 66–81, https://doi.org/10.1016/j.matpur.2008.02.011.

[19] B. Dacorogna, P. Marcellini, and E. Paolini, *Functions with orthogonal Hessian*, Differential Integral Equations, 23 (2010), pp. 51–60.

[20] B. Dacorogna, P. Marcellini, and E. Paolini, *Origami and partial differential equations*, Notices Amer. Math. Soc., 57 (2010), pp. 598–606.

[21] B. Dacorogna, P. Marcellini, and E. Paolini, *On the n-dimensional Dirichlet problem for isometric maps*, J. Funct. Anal., 255 (2008), pp. 3274–3280, https://doi.org/10.1016/j.jfa.2008.10.010.

[22] 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: Modeling and Numerical Simulation, R. Glowinski and P. Neittaanmäki, eds., Comput. Methods Appl. Sci. 16, Springer, 2008, pp. 43–63, https://doi.org/10.1007/978-1-4020-8758-5_3.

[23] E. D. Demaine and T. Tachi, *Origamizer: A practical algorithm for folding any polyhedron*, in Proceedings of the 33rd International Symposium on Computational Geometry, SoCG 2017, Brisbane, Australia, 2017, pp. 34:1–34:16, https://doi.org/10.4230/LIPIcs.SoCG.2017.34.

[24] L. H. Dudte, E. Vouga, T. Tachi, and L. Mahadevan, *Programming curvature using origami tessellations*, Nature Mater., 15 (2016), pp. 583-588, https://doi.org/10.1038/nmat4540.

[25] L. C. Evans, *Partial Differential Equations*, Grad. Stud. Math. 19, American Mathematical Society, 1998, https://doi.org/10.1090/gsm/019.

[26] 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.

[27] A. Figalli, *The Monge-Ampère Equation and Its Applications*, Zur. Lect. Adv. Math., European Mathematical Society, 2017, https://doi.org/10.4171/170.

[28] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, 2nd ed., Springer-Verlag, Berlin, 2008, https://doi.org/10.1007/978-3-662-12613-4.

[29] R. Glowinski, *Numerical methods for fully nonlinear elliptic equations*, in Invited Lectures, 6th International Congress on Industrial and Applied Mathematics, Zürich, Switzerland, 2007, Eur. Math. Soc., 2009, pp. 155–192, https://doi.org/10.4171/056-1/9.

[30] R. Glowinski, *Variational Methods for the Numerical Solution of Nonlinear Elliptic Problems*, SIAM, Philadelphia, 2015, https://doi.org/10.1137/1.9781611973785.

[31] R. Glowinski, Y. Kuznetzov, and T.-W.Pan, *A penalty/Newton/conjugate gradient method for the solution of obstacle problems*, C. R. Acad. Sci. Paris Sér. I, 336 (2003), pp. 435–440, https://doi.org/10.1016/S1631-073X(03)00025-6.

[32] R. Glowinski, S. Leung, and J. Qian, *A penalization-regularization-operator splitting method for eikonal based traveltime tomography*, SIAM J. Imaging Sci., 8 (2015), pp. 1263–1292, https://doi.org/10.1137/140992072.

[33] R. Glowinski and X. Niu, *On the Numerical Solution of a Vector Eikonal Equation*, private communication, 2017.

[34] R. Glowinski, S. J. Osher, and W. Yin, eds., *Splitting Methods in Communication, Imaging, Science, and Engineering*, Sci. Comput., Springer International, 2016, https://doi.org/10.1007/978-3-319-41589-5.

[35] P. A. Gremaud and C. M. Kuster, *Computational study of fast methods for the Eikonal equation*, SIAM J. Sci. Comput., 27 (2006), pp. 1803–1816, https://doi.org/10.1137/040605655.

[36] S.-R. Hysing and S. Turek, *The Eikonal Equation: Numerical Efficiency vs. Algorithmic Complexity on Quadrilateral Grids*, in Proceedings of Algorithmy, 2005, pp. 22–31.

[37] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey, libMesh: *A C++ library for parallel adaptive mesh refinement/coarsening simulations*, Eng. Comput., 22 (2006), pp. 237–254, https://doi.org/10.1007/s00366-006-0049-3.

[38] M. Picasso, *An anisotropic error indicator based on Zienkiewicz–Zhu error estimator: Application to elliptic and parabolic problems*, SIAM J. Sci. Comput., 24 (2003), pp. 1328–1355, https://doi.org/10.1137/S1064827501398578.

[39] M. Picasso, *Numerical study of the effectivity index for an anisotropic error indicator based on Zienkiewicz-Zhu error estimator*, Comm. Numer. Methods Engrg., 19 (2003), pp. 13–23, https://doi.org/10.1002/cnm.546.

[40] M. Picasso, *Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives*, Comput. Methods Appl. Mech. Engrg., 196 (2006), pp. 14–23, https://doi.org/10.1016/j.cma.2005.11.018.

[41] F. Qin, Y. Luo, K. Olsen, W. Cai, and G. Schuster, *Finite-difference solution of the Eikonal equation along expanding wavefronts*, Geophys., 57 (1992), pp. 478–487.

[42] D. Qiu, K.-C. Lam, and L.-M. Lui, *Computing quasiconformal folds*, submitted.

[43] H. Zhao, *A fast sweeping method for Eikonal equations*, Math. Comp., 74 (2005), pp. 603–627, https://doi.org/10.1090/S0025-5718-04-01678-3.