

A New Operator Splitting Method for the Euler Elastica Model for Image Smoothing*

Liang-Jian Deng[†], Roland Glowinski[‡], and Xue-Cheng Tai[§]

Abstract. Euler's elastica model has a wide range of applications in image processing and computer vision. However, the nonconvexity, the nonsmoothness, and the nonlinearity of the associated energy functional make its minimization a challenging task, further complicated by the presence of high order derivatives in the model. In this article we propose a new operator-splitting algorithm to minimize the Euler elastica functional. This algorithm is obtained by applying an operator-splitting based time discretization scheme to an initial value problem (dynamical flow) associated with the optimality system (a system of multivalued equations). The subproblems associated with the three fractional steps of the splitting scheme have either closed form solutions or can be handled by fast dedicated solvers. Compared with earlier approaches relying on ADMM (Alternating Direction Method of Multipliers), the new method has, essentially, only the time discretization step as free parameter to choose, resulting in a very robust and stable algorithm. The simplicity of the subproblems and its modularity make this algorithm quite efficient. Applications to the numerical solution of smoothing test problems demonstrate the efficiency and robustness of the proposed methodology.

Key words. Euler's elastica energy, operator splitting, image smoothing, space projection

AMS subject classifications. 68U10, 94A08

DOI. 10.1137/18M1226361

1. Introduction. In *imaging applications*, the generalized Euler elastica energy is defined by

$$(1.1) \quad E(v) = \int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla v}{|\nabla v|} \right|^2 \right) |\nabla v| dx,$$

where in (1.1), Ω is a bounded domain of \mathbf{R}^2 (a rectangle, typically), a and b are two positive parameters, v is a function of two variables belonging to an appropriate functional space containing (in principle) the underlying image, and $dx = dx_1 dx_2$.

*Received by the editors November 13, 2018; accepted for publication (in revised form) April 16, 2019; published electronically June 25, 2019.

<http://www.siam.org/journals/siims/12-2/M122636.html>

Funding: The work of the first author was supported by National Natural Science Foundation of China grants 61702083, 61772003, and 61876203. The work of the second author was supported by the Hong Kong Baptist University and by the Kennedy Wong Foundation. The work of the third author was supported by Hong Kong Baptist University startup grants RG(R)-RC/17-18/02-MATH and FRG2/17-18/033.

[†]School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China (liangjian.deng@uestc.edu.cn), and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

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

[§]Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (xuechengtai@hkbu.edu.hk).

The Euler elastica energy defined by (1.1) has found applications in image processing, such as denoising [48, 57, 36], segmentation [63, 20, 58, 2], inpainting [45, 6, 48, 55], zooming [48], illusory contour [41, 37, 47], and segmentation with depth [42, 24, 62]. In [45], Shen, Kang, and Chan discussed the mathematical foundation of the Euler elastica model and its mathematical properties, motivated by applications to image inpainting. In addition, the authors of [45] also discussed a numerical PDE method, in order to solve the associated nonlinear problem. In [48], Tai, Hahn, and Chung proposed an augmented Lagrangian method (ALM) to handle the Euler elastica energy and applied the resulting algorithm to the solution of imaging problems in denoising, inpainting, and zooming (we denote this method as the THC method). In [21], Duan et al. proposed a fast augmented Lagrangian method (FALM) to solve the Euler elastica problem for image denoising, inpainting, and zooming based on the framework of the THC method. More recently, in [57] Zhang et al. proposed a linearized augmented Lagrangian method (LALM) to simplify the THC method and applied it to the solution of image denoising problems. In [55], two numerical algorithms were proposed to solve inpainting related problems involving the Euler elastica energy (1.1): The first algorithm is an improved variant of the ALM based algorithm discussed in [48]. The second algorithm is obtained by applying a split-Bregman method to a linearized elastica model proposed in [1]. Following an idea from [40], Masnou and Morel proposed in [41] a novel method to handle the elastica energy functional and applied it to the solution of illusory contour problems. In [37], Kang, Zhu, and Shen used the Euler elastica energy as an effective tool to fuse the scattered corner bases. In [47], Tai and Duan combined level set and binary representation of interfaces to solve, via the Euler elastica model, inpainting, segmentation and illusory control problems. In [5], Bredies, Pock, and Wirth suggested using as smoothing functional a convex, lower semicontinuous approximation of the Euler elastica energy and applied this approximation to the solution of some imaging problems; combined with tailored discretization of measures, the functional introduced in [5] has produced promising results. Taking *image restoration* as an illustration, in order to solve the image restoration problem, via Euler's elastica energy, we need to solve the following minimization problem:

$$(1.2) \quad \min_v \left[\int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla v}{|\nabla v|} \right|^2 \right) |\nabla v| dx + \frac{1}{2} \int_{\Omega} |f - v|^2 dx \right],$$

where v is as in (1.1), and f is the image we are trying to denoise. The first term in the functional in (1.2) is a *regularizing* one; it captures the image geometrical features. The second term is the *fidelity* one; it enforces the underlying image to be close to f .

The main goal of this article is to develop a robust, stable, and “almost” parameter free method to solve problem (1.2), and close variants of it.

The nonconvexity, the nonsmoothness, and the high-order of the derivatives it contains make the fast and robust solution of problem (1.2) a very challenging task. So far, there are only a few methods to solve problems such as (1.2); let us mention among them two graph-cut based methods [23, 1], an integer linear programming (ILP) method [44], a method based on the approximation of the Euler elastica energy [5], and the THC method [48]. The THC method in [48] is a particular realization of the alternating direction method of multipliers (ADMM), a well-known method from mathematical programming (see, e.g., [17] and references

therein for more details). ADMM is a primal-dual method, closely related to the Douglas–Rachford alternating direction method (a well-known operator-splitting method). Following the THC method [48], several extensions were proposed for solving, via the Euler elastica energy functional, a large variety of imaging problems (see [63, 21, 54, 2]). Actually, readers can find further curvature-based methods in [7, 35, 25, 3, 14]. Primal-dual methods have been applied also to derive fast algorithms to handle the total variation (TV) imaging model, introduced in [43] by Rudin, Osher, and Fattorini (ROF). For instance, Droske and Bertozzi in [19] combined the regularization techniques with active contour models to segment polygonal objects in aerial images. This method could avoid losing features by using TV-based inverse scale-space techniques on the input data. See [15, 51, 10, 60, 56, 26, 27, 49, 52, 46, 53, 4, 59, 12, 9, 11, 38] and the references therein for more details.

In this article, we propose a novel and (relatively) simple operator-splitting method for the solution of problem (1.2). The principle of the method is very simple: (i) We introduce the vector-valued functions $\mathbf{q} (= \nabla v)$ and $\boldsymbol{\mu} (= \mathbf{q}/|\mathbf{q}|)$. (ii) Using appropriate indicator functionals, we reformulate problem (1.2) as an unconstrained minimization problem with respect to the triple $(v, \mathbf{q}, \boldsymbol{\mu})$. (iii) We derive an optimality system and associate with it an initial-value problem (gradient flow). (iv) We use the Lie scheme to time-discretize the above initial value problem and capture its steady state solutions. The subproblems associated with the Lie scheme fractional steps have either closed form solutions or can be solved by fast dedicated algorithms (such as FFT). Numerous applications to image smoothing show the efficiency of the proposed method.

When compared to the THC method in [48], the method introduced in this article has the following advantages:

- The time-discretization step is, essentially, the only parameter one has to choose, while the THC method requires the balancing of three augmentation parameters.
- The results produced by the new method are less sensitive to parameter choice than those obtained by the THC method.
- For the same stopping criterion tolerance, the new method needs less iterations than the THC counterpart. Moreover, the new method has a lower cost per iteration than the THC method.

This article is structured as follows: The novel method is described in sections 2 and 3, while its finite difference implementation is discussed in section 4. Section 5 is dedicated to image smoothing, with some experiments designed to show the superiority of the new method. Some conclusions are drawn in section 6. Finally, an appendix dedicated to the Lie and Marchuk–Yanenko operator-splitting schemes is added. Indeed, we feel justified adding this appendix since these two schemes are highly popular in computational fluid dynamics but much less in imaging science.

To conclude this section we would like to mention that various derivations in the following sections are largely mathematically formal (that is, lacking sometimes rigorous mathematical foundations). This follows, in particular, from the fact that, to the best of our knowledge, one has not identified, yet, the proper functional framework to formulate problem (1.2). Accordingly, existence of minimizers and convergence of the proposed schemes are tasks for further studies.

2. A reformulation of problem (1.2). From section 1, Euler's elastica problem reads as

$$(2.1) \quad \min_{v \in \mathcal{V}} \left[\int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla v}{|\nabla v|} \right|^2 \right) |\nabla v| dx + \frac{1}{2} \int_{\Omega} |f - v|^2 dx \right],$$

where \mathcal{V} is a functional space that needs to be chosen properly. As already mentioned in section 1, formulation (2.1) is largely formal since we do not know much about space \mathcal{V} , which has to be, obviously, a subspace of $L^2(\Omega)$. At any rate, the discrete problems largely ignore these functional analysis considerations, and we will say no more about the proper choice of \mathcal{V} . An important issue with formulation (2.1) is that it makes no sense on those parts of Ω where ∇v vanishes (implying that (2.1) is a typical formal mathematical formulation). An obvious (and once popular) way to overcome this difficulty is to replace $|\nabla v|$ by $\sqrt{\epsilon^2 + |\nabla v|^2}$, ϵ being a small parameter. A more sophisticated way we borrow from *viscoplasticity* (see, e.g., [22, 17, 34]) is to replace $\frac{\nabla v}{|\nabla v|}$ by a vector-valued function $\boldsymbol{\mu}$ verifying

$$(2.2) \quad \boldsymbol{\mu} \cdot \nabla v = |\nabla v|, \quad |\boldsymbol{\mu}| \leq 1,$$

with $|\boldsymbol{\mu}| = \sqrt{\mu_1^2 + \mu_2^2}$ $\forall \boldsymbol{\mu} = (\mu_1, \mu_2)$, which is used in some imaging works; see, e.g., [3], and then problem (2.1) by

$$(2.3) \quad \min_{(v, \boldsymbol{\mu}) \in \mathcal{W}} \left[\int_{\Omega} \left(a + b |\nabla \cdot \boldsymbol{\mu}|^2 \right) |\nabla v| dx + \frac{1}{2} \int_{\Omega} |f - v|^2 dx \right],$$

where (formally)

$$\mathcal{W} = \{(v, \boldsymbol{\mu}) \in \mathcal{H}^1(\Omega) \times \mathcal{H}(\Omega, \text{div}), \quad \boldsymbol{\mu} \cdot \nabla v = |\nabla v|, \quad |\boldsymbol{\mu}| \leq 1\},$$

with

$$\mathcal{H}(\Omega, \text{div}) = \{\boldsymbol{\mu} \in (\mathcal{L}^2(\Omega))^2, \nabla \cdot \boldsymbol{\mu} \in \mathcal{L}^2(\Omega)\}.$$

A simple, but computationally important, result is provided by the following (semiformal) proposition.

Proposition 1. Suppose that $(u, \boldsymbol{\lambda})$ is a solution of problem (2.3), then u and f have the same average grey value, that is,

$$(2.4) \quad \int_{\Omega} u dx = \int_{\Omega} f dx.$$

Proof. Consider the pair $(u + c, \boldsymbol{\lambda})$, where $c \in \mathbf{R}$. Since $\nabla(u + c) = \nabla u$, the pair $(u + c, \boldsymbol{\lambda})$ belongs also to \mathcal{W} . Let us denote by J_1 (resp., J_2) the left (resp., right) integral in (2.3). Since $\nabla(u + c) = \nabla u$ we have $J_1(u + c, \boldsymbol{\lambda}) = J_1(u, \boldsymbol{\lambda})$. On the other hand,

$$(2.5) \quad J_2(u + c, \boldsymbol{\lambda}) = \frac{1}{2} \int_{\Omega} |u + c - f|^2 dx = J_2(u, \boldsymbol{\lambda}) + c \int_{\Omega} (u - f) dx + |\Omega| \frac{c^2}{2},$$

with $|\Omega| = \text{measure of } \Omega$. The function u being fixed, the quadratic function of c in the right-hand side of (2.5) takes its minimal value for $c = c_m = \frac{1}{|\Omega|} \int_{\Omega} (f - u) dx$. Suppose that $\int_{\Omega} (f - u) dx \neq 0$; then

$$J_2(u + c_m, \boldsymbol{\lambda}) < J_2(u, \boldsymbol{\lambda}),$$

implying that $(u, \boldsymbol{\lambda})$ is not a minimizer of $J_1 + J_2$. We have, thus, necessarily $\int_{\Omega} u dx = \int_{\Omega} f dx$. ■

Remark 2.1. As we do not have the existence of the minimizer of problem (2.3), thus the proof above is largely formal mathematically. However, if one is willing to consider the discretized problems in finite dimensions, the existence is not a problem and the proof is correct. This remark is also applicable to some similar issues related to existence of minimizers later.

Remark 2.2. It is a common practice to assume periodicity when working with image processing problems. Proposition 1 still holds if one considers the minimization of the elastica functional in a space of sufficiently smooth periodic functions (functions defined over a two-dimensional (2D) torus). In section 3.8 we will return to the case where \mathcal{V} is a space of smooth functions periodic in the Ox_1 and Ox_2 directions

Let us define the sets Σ_f and S by

$$\Sigma_f = \left\{ \mathbf{q} \in (\mathcal{L}^2(\Omega))^2, \exists v \in \mathcal{H}^1(\Omega) \text{ such that } \mathbf{q} = \nabla v \text{ and } \int_{\Omega} (v - f) dx = 0 \right\}$$

and

$$S = \{(\mathbf{q}, \boldsymbol{\mu}) \in (\mathcal{L}^2(\Omega))^2 \times (\mathcal{L}^2(\Omega))^2, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| \leq 1\}.$$

There is then (formal) equivalence between problem (2.3) and

$$(2.6) \quad \min_{(\mathbf{q}, \boldsymbol{\mu}) \in (\mathcal{L}^2(\Omega))^2 \times \mathcal{H}(\Omega, \text{div})} \left[\int_{\Omega} (a + b|\nabla \cdot \boldsymbol{\mu}|^2) |\mathbf{q}| dx + \frac{1}{2} \int_{\Omega} |v_{\mathbf{q}} - f|^2 dx + I_{\Sigma_f}(\mathbf{q}) + I_S(\mathbf{q}, \boldsymbol{\mu}) \right],$$

where I_{Σ_f} and I_S are indicator functionals defined by

$$(2.7) \quad I_{\Sigma_f}(\mathbf{q}) = \begin{cases} 0 & \text{if } \mathbf{q} \in \Sigma_f, \\ +\infty & \text{if } \mathbf{q} \in (\mathcal{L}^2(\Omega))^2 \setminus \Sigma_f \end{cases}$$

and

$$(2.8) \quad I_S(\mathbf{q}, \boldsymbol{\mu}) = \begin{cases} 0 & \text{if } (\mathbf{q}, \boldsymbol{\mu}) \in S, \\ +\infty & \text{if } (\mathbf{q}, \boldsymbol{\mu}) \in (\mathcal{L}^2(\Omega))^2 \times (\mathcal{L}^2(\Omega))^2 \setminus S \end{cases}$$

$v_{\mathbf{q}}$ being the unique solution of the following problem:

$$(2.9) \quad \begin{cases} \nabla^2 v_{\mathbf{q}} = \nabla \cdot \mathbf{q} \text{ in } \Omega, \\ (\nabla v_{\mathbf{q}} - \mathbf{q}) \cdot \mathbf{n} = 0 \text{ on } \partial\Omega, \\ \int_{\Omega} v_{\mathbf{q}} dx = \int_{\Omega} f dx, \end{cases}$$

where in (the Neuman) problem (2.9), \mathbf{n} denotes the outward unit normal vector on the boundary $\partial\Omega$ of Ω . If $\mathbf{q} \in (\mathcal{L}^2(\Omega))^2$, then problem (2.9) has a unique solution in $\mathcal{H}^1(\Omega)$.

3. An operator-splitting method for the solution of problem (2.6).

3.1. Optimality conditions and associated dynamical flow problem. Let us denote by J_1 and J_2 the functionals defined by

$$(3.1) \quad \begin{cases} J_1(\mathbf{q}, \boldsymbol{\mu}) = \int_{\Omega} (a + b|\nabla \cdot \boldsymbol{\mu}|^2) |\mathbf{q}| dx, \\ J_2(\mathbf{q}) = \frac{1}{2} \int_{\Omega} |v_{\mathbf{q}} - f|^2 dx, \end{cases}$$

and suppose that $(\mathbf{p}, \boldsymbol{\lambda})$ is a minimizer of the functional in (2.6). We have that $u = v_{\mathbf{p}}$ (see the definition of $v_{\mathbf{q}}$ given in (2.9)) is a solution to problem (2.1), and the following system of (necessary) optimality conditions holds (formally, at least):

$$(3.2) \quad \begin{cases} \partial_{\mathbf{q}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + D J_2(\mathbf{p}) + \partial I_{\Sigma_f}(\mathbf{p}) + \partial_{\mathbf{q}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0}, \\ D_{\boldsymbol{\mu}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0}, \end{cases}$$

where the Ds (resp., the ∂s) denotes classical differentials (resp., generalized differentials (subdifferentials in the case of nonsmooth convex functionals, I_{Σ_f} being a typical one)). We associate with (3.2) the following initial value problem (dynamical flow):

$$(3.3) \quad \begin{cases} \frac{\partial \mathbf{p}}{\partial t} + \partial_{\mathbf{q}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + D J_2(\mathbf{p}) + \partial I_{\Sigma_f}(\mathbf{p}) + \partial_{\mathbf{q}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} \text{ in } \Omega \times (0, +\infty), \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} + D_{\boldsymbol{\mu}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} \text{ in } \Omega \times (0, +\infty), \\ (\mathbf{p}(0), \boldsymbol{\lambda}(0)) = (\mathbf{p}_0, \boldsymbol{\lambda}_0), \end{cases}$$

with $\gamma > 0$ (the choice of γ will be discussed in section 3.5).

Let us denote the pair $(\mathbf{p}, \boldsymbol{\lambda})$ by \mathbf{X} . Problem (3.3) is clearly of the following form:

$$\begin{cases} \frac{\partial \mathbf{X}}{\partial t} + \sum_{j=1}^4 A_j(\mathbf{X}) \ni \mathbf{0} \text{ in } (0, +\infty), \\ \mathbf{X}(0) = \mathbf{X}_0 (= (\mathbf{p}_0, \boldsymbol{\lambda}_0)), \end{cases}$$

implying (see Appendix A) that the initial value problem (3.3) is a natural candidate to a solution method of the operator-splitting type, *the Lie and Marchuk–Yanenko schemes, in particular*. The idea is to capture the steady state solutions of (3.3) (necessarily solutions of (3.2)) by integrating (approximately) (3.3) over the time interval $(0, +\infty)$. This approach will be discussed in section 3.2.

Remark 3.1. For the initial value, we advocate taking $(\mathbf{p}_0, \boldsymbol{\lambda}_0) \in S$ in (3.3). Related to subdifferentials, it is known that the subdifferentials of the sum of two functionals may not equal the sum of the subdifferentials of each functional. Due to the complexity of the problem, we will not dwell upon this issue here.

3.2. An operator-splitting method for the solution of the dynamical flow problem (3.3)

. Following [32] (and the Appendix A; see also [33] for applications of operator-splitting to imaging), we will use a *Lie scheme* to time-discretize problem (3.3). Let $\tau(>0)$ be a time discretization step; we denote $(n + \alpha)\tau$ by $t^{n+\alpha}$. Among the many possible splitting schemes of the Lie type one can employ to solve problem (3.3), we advocate the following:

$$(3.4) \quad (\mathbf{p}^0, \boldsymbol{\lambda}^0) = (\mathbf{p}_0, \boldsymbol{\lambda}_0).$$

1st fractional step. Solve

$$(3.5) \quad \begin{cases} \begin{cases} \frac{\partial \mathbf{p}}{\partial t} + \partial_{\mathbf{q}} J_1(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0}, \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} + D_{\boldsymbol{\mu}} J_1(\mathbf{p}, \boldsymbol{\lambda}) = \mathbf{0} \end{cases} & \text{in } \Omega \times (t^n, t^{n+1}), \\ (\mathbf{p}(t^n), \boldsymbol{\lambda}(t^n)) = (\mathbf{p}^n, \boldsymbol{\lambda}^n), \end{cases}$$

and set

$$(3.6) \quad (\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}) = (\mathbf{p}(t^{n+1}), \boldsymbol{\lambda}(t^{n+1})).$$

2nd fractional step. Solve

$$(3.7) \quad \begin{cases} \begin{cases} \frac{\partial \mathbf{p}}{\partial t} + \partial_{\mathbf{q}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0}, \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} \end{cases} & \text{in } \Omega \times (t^n, t^{n+1}), \\ (\mathbf{p}(t^n), \boldsymbol{\lambda}(t^n)) = (\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}), \end{cases}$$

and set

$$(3.8) \quad (\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}) = (\mathbf{p}(t^{n+1}), \boldsymbol{\lambda}(t^{n+1})).$$

3rd fractional step. Solve

$$(3.9) \quad \begin{cases} \begin{cases} \frac{\partial \mathbf{p}}{\partial t} + D J_2(\mathbf{p}) + \partial I_{\Sigma_f}(\mathbf{p}) \ni \mathbf{0}, \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} = \mathbf{0} \end{cases} & \text{in } \Omega \times (t^n, t^{n+1}), \\ (\mathbf{p}(t^n), \boldsymbol{\lambda}(t^n)) = (\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}), \end{cases}$$

and set

$$(3.10) \quad (\mathbf{p}^{n+1}, \boldsymbol{\lambda}^{n+1}) = (\mathbf{p}(t^{n+1}), \boldsymbol{\lambda}^{n+2/3}).$$

The Lie scheme (3.5)–(3.10) is only semidiscrete since we have not yet specified how to time-discretize the initial value problems (3.5), (3.7), and (3.9). In order to do so, we suggest using the following time discretization scheme (of the Marchuk–Yanenko type):

$$(3.11) \quad (\mathbf{p}^0, \boldsymbol{\lambda}^0) = (\mathbf{p}_0, \boldsymbol{\lambda}_0).$$

Then, for $n \geq 0$, $(\mathbf{p}^n, \boldsymbol{\lambda}^n) \rightarrow (\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}) \rightarrow (\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}) \rightarrow (\mathbf{p}^{n+1}, \boldsymbol{\lambda}^{n+1})$ as follows:

$$(3.12) \quad \begin{cases} \frac{\mathbf{p}^{n+1/3} - \mathbf{p}^n}{\tau} + \partial_{\mathbf{q}} J_1(\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^n) \ni \mathbf{0}, \\ \gamma \frac{\boldsymbol{\lambda}^{n+1/3} - \boldsymbol{\lambda}^n}{\tau} + D_{\boldsymbol{\mu}} J_1(\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}) = \mathbf{0} \end{cases} \quad \text{in } \Omega \Rightarrow (\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}),$$

$$(3.13) \quad \begin{cases} \frac{\mathbf{p}^{n+2/3} - \mathbf{p}^{n+1/3}}{\tau} + \partial_{\mathbf{q}} I_S(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}) \ni \mathbf{0}, \\ \gamma \frac{\boldsymbol{\lambda}^{n+2/3} - \boldsymbol{\lambda}^{n+1/3}}{\tau} + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}) \ni \mathbf{0} \end{cases} \quad \text{in } \Omega \Rightarrow (\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}),$$

$$(3.14) \quad \begin{cases} \frac{\mathbf{p}^{n+1} - \mathbf{p}^{n+2/3}}{\tau} + D J_2(\mathbf{p}^{n+1}) + \partial I_{\Sigma_f}(\mathbf{p}^{n+1}) \ni \mathbf{0}, \\ \gamma \frac{\boldsymbol{\lambda}^{n+1} - \boldsymbol{\lambda}^{n+2/3}}{\tau} = \mathbf{0} \end{cases} \quad \text{in } \Omega \Rightarrow (\mathbf{p}^{n+1}, \boldsymbol{\lambda}^{n+1}).$$

In the following subsections we are going to discuss the solution of the various subproblems encountered when applying scheme (3.11)–(3.14) to the solution of problem (2.6).

Remark 3.2. The *nonconvexity* of problem (2.1) implies the *nonmonotonicity* of some of the operators encountered in the (formal) necessary conditions (3.2) and the associated initial value problem (3.3) (nondifferentiability further complicates the situation). Due to these difficulties, the convergence of algorithms (3.4)–(3.10) and (3.11)–(3.14) (and of their finite dimensional analogues) as $n \rightarrow +\infty$, are questions that cannot be answered at this moment. These difficult mathematical issues are beyond the scope of this article.

3.3. Computing $\mathbf{p}^{n+1/3}$ from (3.12). The multivalued equation verified by $\mathbf{p}^{n+1/3}$ in (3.12) is nothing but the (formal) Euler–Lagrange equation of the following minimization problem:

$$(3.15) \quad \mathbf{p}^{n+1/3} = \arg \min_{\mathbf{q} \in (\mathcal{L}^2(\Omega))^2} \left[\frac{1}{2} \int_{\Omega} |\mathbf{q} - \mathbf{p}^n|^2 dx + \tau \int_{\Omega} (a + b|\nabla \cdot \boldsymbol{\lambda}^n|^2) |\mathbf{q}| dx \right].$$

Problems such as (3.15) are very common in image processing and viscoplasticity. The closed form solution of problem (3.15) is given by (see [28, 31, 18, 50, 48])

$$(3.16) \quad \mathbf{p}^{n+1/3} = \max \left\{ 0, 1 - \frac{c}{|\mathbf{p}^n|} \right\} \mathbf{p}^n,$$

where $c = \tau a + \tau b |\nabla \cdot \boldsymbol{\lambda}^n|^2$.

3.4. Computing $\boldsymbol{\lambda}^{n+1/3}$ from (3.12). The equation verified by $\boldsymbol{\lambda}^{n+1/3}$ in (3.12) is the (formal) Euler–Lagrange equation of the following minimization problem:

$$(3.17) \quad \boldsymbol{\lambda}^{n+1/3} = \arg \min_{\boldsymbol{\mu} \in \mathcal{H}(\Omega, \text{div})} \left[\gamma \int_{\Omega} \frac{|\boldsymbol{\mu} - \boldsymbol{\lambda}^n|^2}{2\tau} dx + J_1(\boldsymbol{\mu}, \mathbf{p}^{n+1/3}) \right],$$

where $\boldsymbol{\lambda}^n$ and $\mathbf{p}^{n+1/3}$ are known.

From the Euler–Lagrange equation of (3.17), we get that the solution $\boldsymbol{\lambda}^{n+1/3}$ is the solution of following linear elliptic system with variable coefficients:

$$(3.18) \quad \begin{cases} \gamma \frac{\boldsymbol{\lambda}^{n+1/3} - \boldsymbol{\lambda}^n}{\tau} - 2b\nabla(|\mathbf{p}^{n+1/3}| \nabla \cdot \boldsymbol{\lambda}^{n+1/3}) = \mathbf{0} & \text{in } \Omega, \\ b|\mathbf{p}^{n+1/3}| \nabla \cdot \boldsymbol{\lambda}^{n+1/3} = 0 & \text{on } \partial\Omega. \end{cases}$$

Problem (3.18) is (formally) well posed. Properly approximated by either finite difference or finite element methods, problem (3.18) leads to linear systems associated with symmetric positive definite matrices making these systems solvable by a large variety of efficient linear solvers. For those cases where Ω is a rectangle (the most common situation), an alternative to the above mentioned approximation methods is provided by cosine expansions-based spectral methods.

In section 3.8, we will encounter the variant of system (3.18) associated with periodic boundary conditions in the Ox_1 and Ox_2 directions. Indeed, it is common to use periodic boundary conditions for image processing problems. One can justify this approach by assuming that the image is defined on a 2D torus, for example. As shown in [48, section 3.2.5] and [50], periodic boundary conditions simplify the efficient solution of the periodic variant of problem (3.18) by fast Fourier transform (FFT). We will assume periodic boundary conditions in sections 4 and 5.

3.5. Computing $(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3})$ from (3.13).

3.5.1. Decomposition of problem (3.13). One can view system (3.13) as the Euler–Lagrange equation of the following minimization problem:

$$(3.19) \quad \min_{(\mathbf{q}, \boldsymbol{\mu}) \in S} \left[\int_{\Omega} |\mathbf{q} - \mathbf{p}^{n+1/3}|^2 dx + \gamma \int_{\Omega} |\boldsymbol{\mu} - \boldsymbol{\lambda}^{n+1/3}|^2 dx \right].$$

Problem (3.19) can be solved pointwise, reducing, a.e. on Ω , to the following finite dimensional constrained minimization problem:

$$(3.20) \quad (\mathbf{p}^{n+2/3}(x), \boldsymbol{\lambda}^{n+2/3}(x)) = \operatorname{argmin}_{(\mathbf{q}, \boldsymbol{\mu}) \in \sigma} j_{n+1/3}(\mathbf{q}, \boldsymbol{\mu}; x),$$

where $\sigma = \{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| \leq 1\}$ and

$$j_{n+1/3}(\mathbf{q}, \boldsymbol{\mu}; x) = \left| \mathbf{q} - \mathbf{p}^{n+1/3}(x) \right|^2 + \gamma \left| \boldsymbol{\mu} - \boldsymbol{\lambda}^{n+1/3}(x) \right|^2 \forall (\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2.$$

Let us define σ_0 and σ_1 by

$$\sigma_0 = \{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} = \mathbf{0}, |\boldsymbol{\mu}| \leq 1\}, \quad \sigma_1 = \{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \neq \mathbf{0}, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| = 1\}.$$

We clearly have $\sigma = \sigma_0 \cup \sigma_1$, implying that to compute $(\mathbf{p}^{n+2/3}(x), \boldsymbol{\lambda}^{n+2/3}(x))$, we may proceed as follows:

(i) Solve the following two uncoupled minimization problems:

$$(3.21) \quad \left(\mathbf{p}_0^{n+2/3}(x), \boldsymbol{\lambda}_0^{n+2/3}(x) \right) = \operatorname{argmin}_{(\mathbf{q}, \boldsymbol{\mu}) \in \sigma_0} j_{n+1/3}(\mathbf{q}, \boldsymbol{\mu}; x),$$

$$(3.22) \quad \left(\mathbf{p}_1^{n+2/3}(x), \boldsymbol{\lambda}_1^{n+2/3}(x) \right) = \operatorname{argmin}_{(\mathbf{q}, \boldsymbol{\mu}) \in \sigma_1} j_{n+1/3}(\mathbf{q}, \boldsymbol{\mu}; x),$$

(ii) Choose the one that gives the smallest value to $j_{n+1/3}$ as the minimizer of (3.20), i.e.,

$$(3.23) \quad \left(\mathbf{p}^{n+2/3}(x), \boldsymbol{\lambda}^{n+2/3}(x) \right) = \operatorname{argmin} \left[j_{n+1/3}(\mathbf{p}_0(x)^{n+2/3}, \boldsymbol{\lambda}_0(x)^{n+2/3}; x), j_{n+1/3}(\mathbf{p}_1(x)^{n+2/3}, \boldsymbol{\lambda}_1(x)^{n+2/3}; x) \right], \text{a.e. on } \Omega.$$

In what follows, we first introduce a strategy of adaptively choosing γ in section 3.5.2. After that, in sections 3.5.3 and 3.5.4, we will discuss the minimization of the functional in (3.20) over σ_0 and σ_1 , respectively.

3.5.2. Selection of the parameter γ . We intend to select the parameter γ so that the two terms in $j_{n+1/3}$ are balanced. We note that

$$\boldsymbol{\lambda}(t) = \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|}.$$

Thus

$$(3.24) \quad \frac{\partial \lambda}{\partial t} = \lim_{\tau \rightarrow 0} \frac{1}{\tau} \left(\frac{\mathbf{p}(t + \tau)}{|\mathbf{p}(t + \tau)|} - \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|} \right).$$

Due to the relation

$$\left| \frac{\mathbf{p}}{|\mathbf{p}|} - \frac{\mathbf{q}}{|\mathbf{q}|} \right|^2 = \frac{|\mathbf{p}|^2}{|\mathbf{p}|^2} + \frac{|\mathbf{q}|^2}{|\mathbf{q}|^2} - 2 \frac{\mathbf{p} \cdot \mathbf{q}}{|\mathbf{p}||\mathbf{q}|} = 2 \left(1 - \frac{\mathbf{p} \cdot \mathbf{q}}{|\mathbf{p}||\mathbf{q}|} \right) \leq \frac{|\mathbf{p}|^2 + |\mathbf{q}|^2 - 2\mathbf{p} \cdot \mathbf{q}}{|\mathbf{p}||\mathbf{q}|} = \frac{|\mathbf{p} - \mathbf{q}|^2}{|\mathbf{p}||\mathbf{q}|}$$

one has

$$\left| \frac{\mathbf{p}(t + \tau)}{|\mathbf{p}(t + \tau)|} - \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|} \right| \leq \frac{|\mathbf{p}(t + \tau) - \mathbf{p}(t)|}{\sqrt{|\mathbf{p}(t + \tau)||\mathbf{p}(t)|}}$$

Let $\tau \rightarrow 0$, we get from (3.24) that

$$\left| \frac{\partial \lambda}{\partial t} \right| \leq \frac{1}{|\mathbf{p}|} \left| \frac{\partial \mathbf{p}}{\partial t} \right|.$$

For small τ , the minimizer of (3.19) verifies

$$(3.25) \quad \frac{|\mathbf{p}^{n+2/3} - \mathbf{p}^{n+1/3}|^2}{2\tau} + \gamma \frac{|\lambda^{n+2/3} - \lambda^{n+1/3}|^2}{2\tau} \approx \frac{\tau}{2} \left(\left| \frac{\partial \mathbf{p}}{\partial t}(t^{n+1/3}) \right|^2 + \gamma \left| \frac{\partial \lambda}{\partial t}(t^{n+1/3}) \right|^2 \right).$$

According to the above estimate, to balance these two terms, we just need to choose

$$\gamma = |\mathbf{p}^{n+1/3}|^2.$$

In order to avoid the case $|\mathbf{p}^{n+1/3}| \approx 0$, we choose, in practice,

$$(3.26) \quad \gamma = \max(|\mathbf{p}^{n+1/3}|^2, \hat{\alpha}),$$

where $\hat{\alpha}$ is a given positive small number. In this work, we empirically choose $\hat{\alpha} = \sqrt{\tau}$.

3.5.3. Minimizing the functional in (3.20) over σ_0 . Over σ_0 the minimization problem (3.21) reduces to

$$(3.27) \quad \min_{\mu \in \mathbf{R}^2, |\mu| \leq 1} |\mu - \lambda^{n+1/3}(x)|.$$

Clearly, the solution of problem (3.27) is given by

$$(3.28) \quad \lambda_0^{n+1/3}(x) = \frac{\lambda^{n+1/3}(x)}{\max(1, |\lambda^{n+1/3}(x)|)}.$$

Concerning $\mathbf{p}_0^{n+1/3}(x)$, we have, obviously,

$$(3.29) \quad \mathbf{p}_0^{n+1/3}(x) = \mathbf{0}.$$

3.5.4. Minimizing the functional in (3.20) over σ_1 . Over σ_1 , the minimization problem (3.22) reduces to

$$(3.30) \quad \inf_{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \neq 0, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| = 1} \left[|\mathbf{q} - \mathbf{p}^{n+1/3}(x)|^2 + \gamma |\boldsymbol{\mu} - \boldsymbol{\lambda}^{n+1/3}(x)|^2 \right].$$

For notational simplicity, we introduce \mathbf{x} and \mathbf{y} defined by $\mathbf{x} = \mathbf{p}^{n+1/3}(x)$ and $\mathbf{y} = \boldsymbol{\lambda}^{n+1/3}(x)$, respectively. Using this notation and taking relation $|\boldsymbol{\mu}| = 1$ into account, problem (3.30) takes the following simplified formulation:

$$(3.31) \quad \inf_{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \neq 0, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{q} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right].$$

Let us denote $|\mathbf{q}|$ by θ ; since $\boldsymbol{\mu} = \mathbf{q}/|\mathbf{q}|$, the above relations imply that

$$(3.32) \quad \mathbf{q} = \theta \boldsymbol{\mu}, \quad \theta > 0.$$

Relation (3.32) allows us to replace problem (3.31) by the following constrained minimization problem in \mathbf{R}^3 :

$$(3.33) \quad \inf_{(\theta, \boldsymbol{\mu}) \in \mathbf{R} \times \mathbf{R}^2, \theta > 0, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} \theta^2 - \theta \boldsymbol{\mu} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right].$$

In order to solve problem (3.33), we observe that the above problem is equivalent to

$$(3.34) \quad \inf_{\theta > 0} \min_{\boldsymbol{\mu} \in \mathbf{R}^2, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} \theta^2 - \theta \boldsymbol{\mu} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right].$$

In order to minimize on a closed set of \mathbf{R}^3 , the problem that we finally consider is the following variant of problem (3.34):

$$(3.35) \quad \min_{\theta \geq 0} \min_{\boldsymbol{\mu} \in \mathbf{R}^2, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} \theta^2 - \theta \boldsymbol{\mu} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right].$$

With the parameter θ being fixed, the solution $\boldsymbol{\mu}^*(\theta)$ of problem

$$\min_{\boldsymbol{\mu} \in \mathbf{R}^2, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} \theta^2 - \theta \boldsymbol{\mu} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right]$$

is given by $\boldsymbol{\mu}^*(\theta) = \frac{\theta \mathbf{x} + \gamma \mathbf{y}}{|\theta \mathbf{x} + \gamma \mathbf{y}|}$, implying that problem (3.35) reduces to

$$(3.36) \quad \min_{\theta \geq 0} \left[\frac{1}{2} \theta^2 - |\theta \mathbf{x} + \gamma \mathbf{y}| \right].$$

There are many ways to solve problem (3.36), such as Newton's method, bisection or golden section methods, and a variety of fixed point methods ([8]). The method we have chosen is a fixed point one and has shown fast convergence properties. Let us denote by E the function defined by

$$E(\theta) = \frac{1}{2} \theta^2 - |\theta \mathbf{x} + \gamma \mathbf{y}|.$$

We clearly have

$$\frac{dE}{d\theta}(\theta) = \theta - \frac{\mathbf{x} \cdot (\theta\mathbf{x} + \gamma\mathbf{y})}{|\theta\mathbf{x} + \gamma\mathbf{y}|}.$$

In order to solve equation $\frac{dE}{d\theta}(\theta) = 0$, we advocate the simple following *fixed point* method:

$$(3.37) \quad \begin{cases} \theta^0 = |\mathbf{x}|, \\ \text{for } k \geq 0, \theta^{(k)} \rightarrow \theta^{(k+1)}, \\ \theta^{(k+1)} = \max \left(0, \frac{\mathbf{x} \cdot (\theta^{(k)}\mathbf{x} + \gamma\mathbf{y})}{|\theta^{(k)}\mathbf{x} + \gamma\mathbf{y}|} \right). \end{cases}$$

A more detailed presentation of our fixed point method reads as follows.

Algorithm 1. Fixed point solution of problem (3.36)

Input: $\mathbf{x}, \mathbf{y}, \gamma$

Output: θ^*

Initialization: $\theta^{(0)} = |\mathbf{x}|, k = 0$

While: $|\theta^{(k+1)} - \theta^{(k)}| > tol$ and $k < M_{it}$

(1) compute $\theta^{(k+1)}$ by

$$\theta^{(k+1)} = \max \left(0, \frac{\mathbf{x} \cdot (\theta^{(k)}\mathbf{x} + \gamma\mathbf{y})}{|\theta^{(k)}\mathbf{x} + \gamma\mathbf{y}|} \right),$$

(2) $k = k + 1$

End While.

(3) One gets the final θ^* when iterations stop

In Algorithm 1, tol and M_{it} denote a positive tolerance value and the maximum number of iterations, respectively. Actually, Algorithm 1 is not sensitive to these values. For all of the experiments reported in this article we took $tol = 10^{-3}$ and $M_{it} = 100$.

Once θ^* is known, we obtain the vectors $\boldsymbol{\lambda}_1^{n+2/3}(x)$ and $\mathbf{p}_1^{n+2/3}(x)$ (we defined them in section 3.5.1) via the following relations:

$$(3.38) \quad \boldsymbol{\lambda}_1^{n+2/3}(x) = \frac{\theta^* \mathbf{p}^{n+1/3}(x) + \gamma \boldsymbol{\lambda}^{n+1/3}(x)}{|\theta^* \mathbf{p}^{n+1/3}(x) + \gamma \boldsymbol{\lambda}^{n+1/3}(x)|}$$

and

$$(3.39) \quad \mathbf{p}_1^{n+2/3}(x) = \theta^* \boldsymbol{\lambda}_1^{n+2/3}(x),$$

respectively. A more rigorous notation would have been to use $\theta_n^*(x)$ instead of θ^* , since the solution of problem (3.36) varies with x and n (we recall that, in (3.36), $\mathbf{x} = \mathbf{p}^{n+1/3}(x)$ and $\mathbf{y} = \boldsymbol{\lambda}^{n+1/3}(x)$).

Once we compute $(\mathbf{p}_1^{n+2/3}(x), \boldsymbol{\lambda}_1^{n+2/3}(x))$ from (3.28)–(3.29) and $(\mathbf{p}_1^{n+2/3}(x), \boldsymbol{\lambda}_1^{n+2/3}(x))$ from (3.38)–(3.39), we obtain the minimizer of (3.20) through (3.23).

3.6. Computing \mathbf{p}^{n+1} and λ^{n+1} from (3.14). We clearly have

$$(3.40) \quad \boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^{n+2/3}.$$

On the other hand, the multivalued equation verified by \mathbf{p}^{n+1} in (3.14) is the Euler–Lagrange equation of the following minimization problem:

$$(3.41) \quad \mathbf{p}^{n+1} = \arg \min_{\mathbf{q} \in \Sigma_f} \left[\frac{1}{2} \int_{\Omega} |\mathbf{q} - \mathbf{p}^{n+2/3}|^2 dx + \frac{\tau}{2} \int_{\Omega} |v_{\mathbf{q}} - f|^2 dx \right],$$

the function $v_{\mathbf{q}}$ being defined by (2.9).

From the definition of Σ_f (see section 2), problem (3.41) is equivalent to

$$(3.42) \quad \begin{cases} \mathbf{p}^{n+1} = \nabla u^{n+1} & \text{with} \\ u^{n+1} = \arg \min_{v \in \mathcal{H}^1(\Omega)} \left[\frac{1}{2} \int_{\Omega} |\nabla v|^2 dx + \frac{\tau}{2} \int_{\Omega} |v - f|^2 dx - \int_{\Omega} \mathbf{p}^{n+2/3} \cdot \nabla v dx \right]. \end{cases}$$

Function u^{n+1} is the unique solution of the following well-posed linear variational problem in $\mathcal{H}^1(\Omega)$:

$$(3.43) \quad \begin{cases} u^{n+1} \in \mathcal{H}^1(\Omega), \\ \int_{\Omega} \nabla u^{n+1} \cdot \nabla v dx + \tau \int_{\Omega} u^{n+1} v dx = \int_{\Omega} \mathbf{p}^{n+2/3} \cdot \nabla v dx + \tau \int_{\Omega} f v dx \quad \forall v \in \mathcal{H}^1(\Omega). \end{cases}$$

Problems (3.42) and (3.43) have a unique solution which is the weak solution of the following problem:

$$(3.44) \quad \begin{cases} -\nabla^2 u^{n+1} + \tau u^{n+1} = -\nabla \cdot \mathbf{p}^{n+2/3} + \tau f \text{ in } \Omega, \\ (\nabla u^{n+1} - \mathbf{p}^{n+2/3}) \cdot \mathbf{n} = 0 \text{ on } \partial\Omega. \end{cases}$$

The linear elliptic problem (3.44) is of the Neumann type with constant coefficients. The numerical solution of this type of problem has motivated a very large number of methods and associated software. In the particular case of rectangular domains, many efficient solvers are available for the solution of the discrete finite element analogues of problem (3.44) obtained by symmetry preserving finite difference discretization (sparse Cholesky, conjugate gradient, cyclic reduction, etc.). In section 3.8, we will encounter the variant of (3.44) associated with periodic boundary conditions. Its discrete analogues are particularly well suited to FFT based solvers.

3.7. Summary. The subproblems (3.12), (3.13), and (3.14) encountered in our splitting method aim at minimizing consecutively the various components of the elastica cost functional. Our proposed algorithm is summarized in Algorithm 2.

Algorithm 2. A schematic description of the algorithm solving problem (2.1)

Input: The input image f , the parameters a , b , and τ .

Output: The computed image u^* .

Initialization: $n = 0$, $u^0 = f$, $\mathbf{p}^0 = \nabla f$, $\boldsymbol{\lambda}^0(x) = \begin{cases} \mathbf{p}^0(x)/|\mathbf{p}^0(x)| & \text{if } \mathbf{p}^0(x) \neq 0, \\ \mathbf{0} & \text{otherwise.} \end{cases} x \in \Omega$.

While: $\|u^{n+1} - u^n\|/\|u^{n+1}\| > tol$ and $n < M_{iter}$

1. Using the methods discussed in sections 3.3 and 3.4, solve system (3.12) to obtain $(\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3})$.
2. Use the method discussed in section 3.5 to obtain $(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3})$ from (3.13).
3. Use the method discussed in section 3.6 to obtain $(u^{n+1}, \mathbf{p}^{n+1}, \boldsymbol{\lambda}^{n+1})$ from (3.14).
4. Check convergence and go to the next iteration or stop.

End While.

If iterations stop, take $u^* = u^{n+1}$.

In Algorithm 2, tol is the stopping criterion tolerance, M_{iter} is the maximum of iterations and the norm $\|\cdot\|$ is the L_2 norm. All of the subproblems encountered when using Algorithm 2 have either closed form solutions or can be solved by dedicated fast solvers. Due to the semi-implicit nature of the operator-splitting scheme, we can use (relatively) large values of τ and our numerical experiments show that the overall iteration number is (relatively) low. We have, however, to keep τ small enough so that the resulting splitting error is small as well (see Appendix A). The model parameters a and b have to be given. Finally, the time-discretization step τ also needs to be provided. We want to say that τ is easy to tune. The selection of γ was addressed in section 3.5.2; further information about the choice of all these parameters will be provided in section 5.

3.8. On the handling of periodic boundary conditions. In the preceding sections (section 3.4, in particular) we mentioned quite a few times the possibility of using periodic boundary conditions when Ω is a rectangular domain (a very common situation). The changes that choice requires are minimal and will be discussed below (we will assume that $\Omega = (0, L) \times (0, H)$).

The *first* modification one encounters is to replace the space \mathcal{W} in (2.3) by \mathcal{W}_P defined by

$$\mathcal{W}_P = \{(v, \boldsymbol{\mu}) \in \mathcal{H}_P^1(\Omega) \times \mathcal{H}_P(\Omega, \text{div}), \quad \boldsymbol{\mu} \cdot \nabla v = |\nabla v|, \quad |\boldsymbol{\mu}| \leq 1\},$$

where (with obvious notation)

$$\mathcal{H}_P^1(\Omega) = \{v \in \mathcal{H}^1(\Omega), v(0, \cdot) = v(L, \cdot), v(\cdot, 0) = v(\cdot, H)\}$$

and

$$\mathcal{H}_P(\Omega, \text{div}) = \{\boldsymbol{\mu} = (\mu_1, \mu_2) | \boldsymbol{\mu} \in \mathcal{H}(\Omega, \text{div}), \mu_1(0, \cdot) = \mu_1(L, \cdot), \mu_2(\cdot, 0) = \mu_2(\cdot, H)\}.$$

The *second* modification is to define Σ_f by

$$\Sigma_f = \left\{ \mathbf{q} \in (\mathcal{L}^2(\Omega))^2, \exists v \in \mathcal{H}_P^1(\Omega) \text{ such that } \mathbf{q} = \nabla v \text{ and } \int_{\Omega} (v - f) dx = 0 \right\},$$

and we replace (2.6) and (2.9) by

$$(3.45) \quad \min_{(\mathbf{q}, \boldsymbol{\mu}) \in (\mathcal{L}^2(\Omega))^2 \times \mathcal{H}_P(\Omega, \text{div})} \left[\int_{\Omega} (a + b|\nabla \cdot \boldsymbol{\mu}|^2) |\mathbf{q}| dx + \frac{1}{2} \int_{\Omega} |v_{\mathbf{q}} - f|^2 dx + I_{\Sigma_f}(\mathbf{q}) + I_S(\mathbf{q}, \boldsymbol{\mu}) \right],$$

and

$$(3.46) \quad \begin{cases} \nabla^2 v_{\mathbf{q}} = \nabla \cdot \mathbf{q} \text{ in } \Omega, \\ v_{\mathbf{q}}(0, \cdot) = v_{\mathbf{q}}(L, \cdot), \quad v_{\mathbf{q}}(\cdot, 0) = v_{\mathbf{q}}(\cdot, H), \\ \left(\frac{\partial v_{\mathbf{q}}}{\partial x_1} - q_1 \right)(0, \cdot) = \left(\frac{\partial v_{\mathbf{q}}}{\partial x_1} - q_1 \right)(L, \cdot), \quad \left(\frac{\partial v_{\mathbf{q}}}{\partial x_2} - q_2 \right)(\cdot, 0) = \left(\frac{\partial v_{\mathbf{q}}}{\partial x_2} - q_2 \right)(\cdot, H), \\ \int_{\Omega} v_{\mathbf{q}} dx = \int_{\Omega} f dx, \end{cases}$$

respectively (above, $(q_1, q_2) = \mathbf{q}$).

The *third* modification is to replace (3.17) and (3.18) by

$$(3.47) \quad \boldsymbol{\lambda}^{n+1/3} = \arg \min_{\boldsymbol{\mu} \in \mathcal{H}_P(\Omega, \text{div})} \left[\gamma \int_{\Omega} \frac{|\boldsymbol{\mu} - \boldsymbol{\lambda}^n|^2}{2\tau} dx + J_1(\boldsymbol{\mu}, \mathbf{p}^{n+1/3}) \right]$$

and

$$(3.48) \quad \begin{cases} \gamma \frac{\boldsymbol{\lambda}^{n+1/3} - \boldsymbol{\lambda}^n}{\tau} - 2b\nabla(|\mathbf{p}^{n+1/3}| \nabla \cdot \boldsymbol{\lambda}^{n+1/3}) = \mathbf{0} \text{ in } \Omega, \\ \boldsymbol{\lambda}_1(0, \cdot) = \boldsymbol{\lambda}_1(L, \cdot), \quad \boldsymbol{\lambda}_2(\cdot, 0) = \boldsymbol{\lambda}_2(\cdot, H), \\ \left(|\mathbf{p}^{n+1/3}| \nabla \cdot \boldsymbol{\lambda}^{n+1/3} \right)(0, \cdot) = \left(|\mathbf{p}^{n+1/3}| \nabla \cdot \boldsymbol{\lambda}^{n+1/3} \right)(L, \cdot) \\ \left(|\mathbf{p}^{n+1/3}| \nabla \cdot \boldsymbol{\lambda}^{n+1/3} \right)(\cdot, 0) = \left(|\mathbf{p}^{n+1/3}| \nabla \cdot \boldsymbol{\lambda}^{n+1/3} \right)(\cdot, H), \end{cases}$$

respectively. The periodic boundary conditions in (3.48) make the above linear elliptic problem well suited to FFT-based solution methods, after appropriate finite difference discretization (see section 4).

Finally, replace (3.42), (3.43), and (3.44) by

$$(3.49) \quad \begin{cases} \mathbf{p}^{n+1} = \nabla u^{n+1} \quad \text{with} \\ u^{n+1} = \arg \min_{v \in \mathcal{H}_P^1(\Omega)} \left[\frac{1}{2} \int_{\Omega} |\nabla v|^2 dx + \frac{\tau}{2} \int_{\Omega} |v - f|^2 dx - \int_{\Omega} \mathbf{p}^{n+2/3} \cdot \nabla v dx \right], \end{cases}$$

$$(3.50) \quad \begin{cases} u^{n+1} \in \mathcal{H}_P^1(\Omega), \\ \int_{\Omega} \nabla u^{n+1} \cdot \nabla v dx + \tau \int_{\Omega} u^{n+1} v dx = \int_{\Omega} \mathbf{p}^{n+2/3} \cdot \nabla v dx + \tau \int_{\Omega} f v dx \quad \forall v \in \mathcal{H}_P^1(\Omega), \end{cases}$$

and

$$(3.51) \quad \begin{cases} -\nabla^2 u^{n+1} + \tau u^{n+1} = -\nabla \cdot \mathbf{p}^{n+2/3} + \tau f \text{ in } \Omega, \\ u^{n+1}(0, \cdot) = u^{n+1}(L, \cdot), \quad u^{n+1}(\cdot, 0) = u^{n+1}(\cdot, H), \\ \left(\frac{\partial u^{n+1}}{\partial x_1} - p_1^{n+2/3} \right)(0, \cdot) = \left(\frac{\partial u^{n+1}}{\partial x_1} - p_1^{n+2/3} \right)(L, \cdot), \\ \left(\frac{\partial u^{n+1}}{\partial x_2} - p_2^{n+2/3} \right)(\cdot, 0) = \left(\frac{\partial u^{n+1}}{\partial x_2} - p_2^{n+2/3} \right)(\cdot, H), \end{cases}$$

respectively.

System (3.51) is an elliptic problem with constant coefficients, and periodic boundary conditions, taking place on a rectangle. In section 4, we will show how to solve its finite difference discrete analogues by FFT.

4. Numerical discretization.

4.1. Synopsis. As with the THC method in [48], we will assume that Ω is a rectangle. We assume also that all functions are periodic in both the x_1 and x_2 directions. To discretize the Euler elastica variational problem, we will use staggered grids as visualized in Figure 1. In Figure 1, the unknown function v is discretized at the \bullet -nodes, while the first (resp., second) components of \mathbf{q} and $\boldsymbol{\mu}$ are discretized at the \circ -nodes (resp., \square -nodes). Useful notation will be introduced in section 4.2. The solution of the discrete subproblems will be discussed in sections 4.3–4.6.

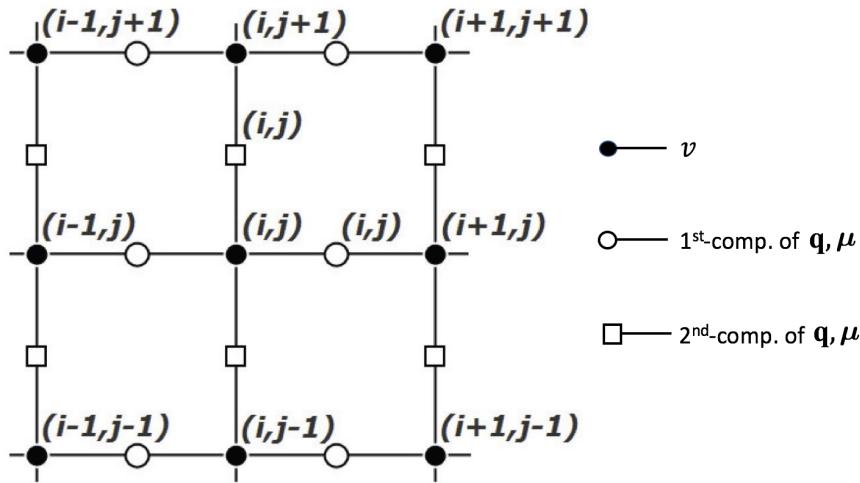


Figure 1. Indexation of the discrete analogues of the unknown functions v (at the \bullet -nodes) and of the first (at the \circ -nodes) and second (at the \square -nodes) components of the vector-valued functions \mathbf{q} and $\boldsymbol{\mu}$.

4.2. Some useful discrete operators. After discretization, we denote by Ω_h the discrete image domain $\Omega_h = [1, M_1]h \times [1, N_1]h$, where $h = L/M_1 = H/N_1$, which indicates the image size is $M_1 \times N_1$. Note that Ω_h is a set of $M_1 N_1$ points in \mathbf{R}^2 . Taking periodicity into account,

we define the backward (–) and forward (+) discrete analogues of $\frac{\partial v}{\partial x_1}$ and $\frac{\partial v}{\partial x_2}$ by

$$\partial_1^- v(i, j) = \begin{cases} (v(i, j) - v(i - 1, j))/h, & 1 < i \leq M_1 \\ (v(1, j) - v(M_1, j))/h, & i = 1, \end{cases}$$

$$\partial_2^- v(i, j) = \begin{cases} (v(i, j) - v(i, j - 1))/h, & 1 < j \leq N_1 \\ (v(i, 1) - v(i, N_1))/h, & j = 1, \end{cases}$$

$$\partial_1^+ v(i, j) = \begin{cases} (v(i + 1, j) - v(i, j))/h, & 1 \leq i < M_1 \\ (v(1, j) - v(M_1, j))/h, & i = M_1, \end{cases}$$

$$\partial_2^+ v(i, j) = \begin{cases} (v(i, j + 1) - v(i, j))/h, & 1 \leq j < N_1 \\ (v(i, 1) - v(i, N_1))/h, & j = N_1. \end{cases}$$

With obvious notation, the discrete forward (+) and backward (–) gradient operators ∇^+ and ∇^- are defined by

$$\nabla^\pm v(i, j) = (\partial_1^\pm v(i, j), \partial_2^\pm v(i, j)).$$

The associated discrete forward (+) and backward (–) divergence operators div^+ and div^- are defined (again with obvious notation) by

$$\operatorname{div}^\pm \mathbf{q}(i, j) = \partial_1^\pm q_1(i, j) + \partial_2^\pm q_2(i, j).$$

If, in particular, a variable defined at the \circ -nodes (resp., \square -nodes) needs to be evaluated at the \square -node (resp., \circ -node) (i, j) , they will be done, respectively, using the following averaging operators:

$$(4.1) \quad \mathcal{A}_{i,j}^\square(\mu_1) = \frac{\mu_1(i, j + 1) + \mu_1(i - 1, j + 1) + \mu_1(i, j) + \mu_1(i - 1, j)}{4},$$

$$(4.2) \quad \mathcal{A}_{i,j}^\circ(\mu_2) = \frac{\mu_2(i + 1, j) + \mu_2(i, j) + \mu_2(i + 1, j - 1) + \mu_2(i, j - 1)}{4},$$

where μ_1 (resp., μ_2) is defined at the \circ -nodes (resp., \square -nodes). In order to evaluate the magnitude of $\mathbf{q} = (q_1, q_2)$ at the \bullet -node (i, j) we will use an additional averaging operator, namely

$$(4.3) \quad |\mathcal{A}|_{i,j}^\bullet(\mathbf{q}) = \sqrt{\left(\frac{q_1(i, j) + q_1(i - 1, j)}{2}\right)^2 + \left(\frac{q_2(i, j) + q_2(i, j - 1)}{2}\right)^2},$$

where q_1 and q_2 are defined on \circ -nodes and \square -nodes, respectively. Similarly, the discrete divergence $\text{div}_{i,j}^\bullet(\boldsymbol{\mu})$ of $\boldsymbol{\mu} = (\mu_1, \mu_2)$ at the \bullet -node (i, j) is defined by

$$(4.4) \quad \text{div}_{i,j}^\bullet(\boldsymbol{\mu}) = [\mu_1(i, j) - \mu_1(i-1, j) + \mu_2(i, j) - \mu_2(i, j-1)]/h,$$

where μ_1 (resp., μ_2) is defined at the \circ -nodes (resp., \square -nodes). Finally, we define shifting and identity operators by

$$(4.5) \quad \mathcal{S}_1^\pm \varphi(i, j) = \varphi(i \pm 1, j), \quad \mathcal{S}_2^\pm \varphi(i, j) = \varphi(i, j \pm 1), \quad \text{and} \quad \mathcal{I}\varphi(i, j) = \varphi(i, j).$$

4.3. Computation of the discrete analogue of $\mathbf{p}^{n+1/3}$ in (3.16). Let us recall that from (3.16) one has

$$(4.6) \quad \mathbf{p}^{n+1/3} = \max\left\{0, 1 - \frac{c}{|\mathbf{p}^n|}\right\} \mathbf{p}^n,$$

where $c = \tau a + \tau b |\nabla \cdot \boldsymbol{\lambda}^n|^2$. In the discrete setting, the first (resp., second) component of \mathbf{p}^n and $\boldsymbol{\lambda}^n$ is defined at \circ -nodes (resp., \square -nodes), we need to discuss the two situations we will encounter when discretizing (4.6) (for simplicity, we will denote $\boldsymbol{\lambda}^n$ by $\boldsymbol{\lambda}$ and \mathbf{p}^n by \mathbf{p}).

(1) If (i, j) is a \circ -node, the corresponding discretization of \mathbf{p} and c is given as follows:

$$(4.7) \quad p_1^{(1)}(i, j) = p_1(i, j); \quad p_2^{(1)}(i, j) = \mathcal{A}_{i,j}^\circ(p_2),$$

$$(4.8) \quad \begin{aligned} c^{(1)}(i, j) &= \tau [a + b |\partial_1 \lambda_1(i, j) + \partial_2 \lambda_2(i, j)|^2] \\ &= \tau \left[a + b \left| \frac{\lambda_1(i+1, j) - \lambda_1(i-1, j)}{2h} + \frac{\lambda_2(i+1, j) + \lambda_2(i, j)}{2h} - \frac{\lambda_2(i, j-1) + \lambda_2(i+1, j-1)}{2h} \right|^2 \right]. \end{aligned}$$

(2) If (i, j) is a \square -node, the corresponding discretization of \mathbf{p} and c is given as follows:

$$(4.9) \quad p_1^{(2)}(i, j) = \mathcal{A}_{i,j}^\square(p_1); \quad p_2^{(2)}(i, j) = p_2(i, j),$$

$$(4.10) \quad \begin{aligned} c^{(2)}(i, j) &= \tau [a + b |\partial_1 \lambda_1(i, j) + \partial_2 \lambda_2(i, j)|^2] \\ &= \tau \left[a + b \left| \frac{\lambda_1(i, j) + \lambda_1(i, j+1)}{2h} - \frac{\lambda_1(i-1, j) + \lambda_2(i-1, j+1)}{2h} + \frac{\lambda_2(i, j+1) - \lambda_2(i, j-1)}{2h} \right|^2 \right]. \end{aligned}$$

Finally,

$$(4.11) \quad p_\alpha^{n+1/3}(i, j) = \max \left\{ 0, 1 - \frac{c^{(\alpha)}(i, j)}{\sqrt{|p_1^{(\alpha)}(i, j)|^2 + |p_2^{(\alpha)}(i, j)|^2}} \right\} p_\alpha^{(\alpha)}(i, j), \quad \alpha = \{1, 2\}.$$

4.4. Computation of the discrete analogue of $\lambda^{n+1/3}$ in (3.18). We recall that (3.18) reads as

$$(4.12) \quad \gamma\lambda^{n+1/3} - \tau\nabla(2b|\mathbf{p}^{n+1/3}|\nabla \cdot \lambda^{n+1/3}) = \gamma\lambda^n, \text{ in } \Omega,$$

It is completed by periodic boundary conditions. For simplicity, we denote the (known) vector $(\mathbf{p}^{n+1/3}, \lambda^n)$ by $(\tilde{\mathbf{p}}, \tilde{\lambda})$ and $\lambda^{n+1/3}$ (an unknown one) by λ . Following [48], we discretize (4.12) as follows:

$$(4.13) \quad \gamma\lambda - \tau\nabla^+(2b|\tilde{\mathbf{p}}|\operatorname{div}^-\lambda) = \gamma\tilde{\lambda}.$$

To solve (4.13), we will employ (as in [48]) a *frozen coefficient* approach where instead of solving (4.13) we solve

$$(4.14) \quad \gamma\lambda - c^*\nabla^+(\operatorname{div}^-\lambda) = \gamma\tilde{\lambda} - \nabla^+\left[(c^* - 2\tau b|\tilde{\mathbf{p}}|)\operatorname{div}^-\tilde{\lambda}\right],$$

with c^* properly chosen. Following [48], we advocate taking $c^* = \max_{\bullet\text{-nodes}(i,j)} 2\tau b|\mathcal{A}|_{i,j}^\bullet(\tilde{\mathbf{p}})$.

In matrix form, (4.14) can be written as, in Ω_h ,

$$(4.15) \quad \gamma h^2 \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} - c^* \begin{pmatrix} \partial_1^+ \\ \partial_2^+ \end{pmatrix} \begin{pmatrix} \partial_1^- & \partial_2^- \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \gamma h^2 \begin{pmatrix} \tilde{\lambda}_1 \\ \tilde{\lambda}_2 \end{pmatrix} - \begin{pmatrix} \partial_1^+ \\ \partial_2^+ \end{pmatrix} (c^*h - 2\tau bh|\tilde{\mathbf{p}}|)\operatorname{div}^-\tilde{\lambda},$$

or, equivalently,

$$(4.16) \quad \begin{cases} (\gamma h^2 - c^* \partial_1^+ \partial_1^-) \lambda_1 - c^* \partial_1^+ \partial_2^- \lambda_2 = \gamma h^2 \tilde{\lambda}_1 - \partial_1^+ (c^*h - 2\tau bh|\tilde{\mathbf{p}}|)\operatorname{div}^-\tilde{\lambda}, \\ -c^* \partial_2^+ \partial_1^- \lambda_1 + (\gamma h^2 - c^* \partial_2^+ \partial_2^-) \lambda_2 = \gamma h^2 \tilde{\lambda}_2 - \partial_2^+ (c^*h - 2\tau bh|\tilde{\mathbf{p}}|)\operatorname{div}^-\tilde{\lambda}. \end{cases}$$

Using the shifting and identity operator defined in section 4.2, for each pair (i, j) the first equation in (4.16) reads as

$$(4.17) \quad [\gamma h^2 \mathcal{I} + c^*(\mathcal{I} - \mathcal{S}_1^+)(\mathcal{I} - \mathcal{S}_1^-)] \lambda_1(i, j) + c^*(\mathcal{I} - \mathcal{S}_1^+)(\mathcal{I} - \mathcal{S}_2^-) \lambda_2(i, j) = g_1(i, j),$$

where

$$g_1(i, j) = \gamma h^2 \tilde{\lambda}_1(i, j) - \left[(c^*h - 2\tau bh|\mathcal{A}|_{i+1,j}^\bullet(\tilde{\mathbf{p}})) \operatorname{div}_{i+1,j}^\bullet \tilde{\lambda} - (c^*h - 2\tau bh|\mathcal{A}|_{i,j}^\bullet(\tilde{\mathbf{p}})) \operatorname{div}_{i,j}^\bullet \tilde{\lambda} \right].$$

Similarly, the second equation of (4.16) reads as

$$(4.18) \quad c^*(\mathcal{I} - \mathcal{S}_2^+)(\mathcal{I} - \mathcal{S}_1^-) \lambda_1(i, j) + [\gamma h^2 \mathcal{I} + c^*(\mathcal{I} - \mathcal{S}_2^+)(\mathcal{I} - \mathcal{S}_2^-)] \lambda_2(i, j) = g_2(i, j),$$

where

$$g_2(i, j) = \gamma h^2 \tilde{\lambda}_2(i, j) - \left[(c^*h - 2\tau bh|\mathcal{A}|_{i,j+1}^\bullet(\tilde{\mathbf{p}})) \operatorname{div}_{i,j+1}^\bullet \tilde{\lambda} - (c^*h - 2\tau bh|\mathcal{A}|_{i,j}^\bullet(\tilde{\mathbf{p}})) \operatorname{div}_{i,j}^\bullet \tilde{\lambda} \right].$$

For the boundary conditions we consider to be the periodic ones, we may apply the *discrete Fourier transform* \mathcal{F} to (4.17), (4.18). We then obtain

$$(4.19) \quad \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \mathcal{F} \begin{pmatrix} \lambda_1(y_i, y_j) \\ \lambda_2(y_i, y_j) \end{pmatrix} = \mathcal{F} \begin{pmatrix} g_1(y_i, y_j) \\ g_2(y_i, y_j) \end{pmatrix},$$

where in (4.19) one has

$$\begin{aligned} a_{11} &= \gamma h^2 - 2c^*(\cos z_i - 1), & a_{12} &= c^*(\cos z_i - 1 + \sqrt{-1} \sin z_i)(\cos z_j - 1 - \sqrt{-1} \sin z_j), \\ a_{21} &= c^*(\cos z_j - 1 + \sqrt{-1} \sin z_j)(\cos z_i - 1 - \sqrt{-1} \sin z_i), & a_{22} &= \gamma h^2 - 2c^*(\cos z_j - 1), \end{aligned}$$

with

$$(4.20) \quad z_i = \frac{2\pi}{M_1} y_i, \quad y_i = 1, 2, \dots, M_1, \quad \text{and} \quad z_j = \frac{2\pi}{N_1} y_j, \quad y_j = 1, 2, \dots, N_1.$$

The determinant $D(i, j)$ of the coefficient matrix in (4.19) is given by

$$D(i, j) = \gamma^2 h^4 + 2\gamma h^2 c^*(2 - \cos z_i - \cos z_j),$$

implying $D(i, j) > 0$ if $\gamma > 0$. It then follows from (4.19) that (with obvious notation) the solution $\boldsymbol{\lambda}$ of problem (4.14) (the discrete analogue of $\boldsymbol{\lambda}^{n+1/3}$ in (3.18)) is given by

$$(4.21) \quad \begin{cases} \lambda_1 = \text{Real} \left[\mathcal{F}^{-1} \left(\frac{a_{22}\mathcal{F}(g_1) - a_{12}\mathcal{F}(g_2)}{D} \right) \right], \\ \lambda_2 = \text{Real} \left[\mathcal{F}^{-1} \left(\frac{-a_{21}\mathcal{F}(g_1) + a_{11}\mathcal{F}(g_2)}{D} \right) \right], \end{cases}$$

where $\text{Real}(x + \sqrt{-1}y) = x$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$.

4.5. Computation of the discrete analogue of $(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3})$ in (3.13). We need to solve problem (3.19) to get the solutions. In the following, we give the details of its discretization.

4.5.1. Solution of (3.28). From section 3.5.3, we see that the minimizer of the functional in (3.20) over σ_0 is given by

$$(4.22) \quad \left(\mathbf{p}_0^{n+2/3}(x), \boldsymbol{\lambda}_0^{n+2/3}(x) \right) = \left(\mathbf{0}, \frac{\boldsymbol{\lambda}^{n+1/3}(x)}{\max[1, |\boldsymbol{\lambda}^{n+1/3}(x)|]} \right).$$

The discrete analogue of (4.22) reads as

$$(4.23) \quad \left(\mathbf{p}_0^{n+2/3}(i, j), \boldsymbol{\lambda}_0^{n+2/3}(i, j) \right) = \left(\mathbf{0}, \frac{\boldsymbol{\lambda}^{n+1/3}(i, j)}{\max \left[1, \sqrt{|\lambda_1^{n+1/3}(i, j)|^2 + |\lambda_2^{n+1/3}(i, j)|^2} \right]} \right),$$

with $\boldsymbol{\lambda}^{n+1/3}(i, j) = (\lambda_1^{n+1/3}(i, j), \lambda_2^{n+1/3}(i, j))$.

4.5.2. Discretization of problem (3.30). Section 3.5.4 was dedicated to the solution of problem (3.30), a constrained minimization problem in \mathbf{R}^4 defined by

$$(4.24) \quad \inf_{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \neq 0, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| = 1} \left[|\mathbf{q} - \mathbf{p}^{n+1/3}(x)|^2 + \gamma |\boldsymbol{\mu} - \boldsymbol{\lambda}^{n+1/3}(x)|^2 \right].$$

Proceeding as in section 3.5.4, we define $\mathbf{x}_{i,j}$ and $\mathbf{y}_{i,j}$ by

$$\mathbf{x}_{i,j} = \left(\mathbf{x}_{i,j}^{(1)}, \mathbf{x}_{i,j}^{(2)} \right) = \left(p_1^{n+1/3}(i, j), p_2^{n+1/3}(i, j) \right),$$

$$\mathbf{y}_{i,j} = \left(\mathbf{y}_{i,j}^{(1)}, \mathbf{y}_{i,j}^{(2)} \right) = \left(\lambda_1^{n+1/3}(i, j), \lambda_2^{n+1/3}(i, j) \right).$$

Then, we use the following discrete variant of algorithm (3.37) to compute $\theta_{i,j}^*$:

$$(4.25) \quad \begin{cases} \theta_{i,j}^{(0)} = |\mathbf{x}_{i,j}|, \\ \text{for } k \geq 0, \theta_{i,j}^{(k)} \rightarrow \theta_{i,j}^{(k+1)} \\ \theta_{i,j}^{(k+1)} = \max \left(0, \frac{\mathbf{x}_{i,j} \cdot (\theta_{i,j}^{(k)} \mathbf{x}_{i,j} + \gamma \mathbf{y}_{i,j})}{|\theta_{i,j}^{(k)} \mathbf{x}_{i,j} + \gamma \mathbf{y}_{i,j}|} \right). \end{cases}$$

Once $\theta_{i,j}^*$ is computed we obtain the discrete analogues of $(\mathbf{p}_1^{n+2/3}(x), \boldsymbol{\lambda}_1^{n+2/3}(x))$ from the following formula which is the discrete analogue of (3.38), (3.39):

$$(4.26) \quad \begin{cases} \boldsymbol{\lambda}^{n+2/3}(i, j) = \frac{\theta_{i,j}^* \mathbf{x}_{i,j} + \gamma \mathbf{y}_{i,j}}{\sqrt{\left| \theta_{i,j}^* x_{i,j}^{(1)} + \gamma y_{i,j}^{(1)} \right|^2 + \left| \theta_{i,j}^* x_{i,j}^{(2)} + \gamma y_{i,j}^{(2)} \right|^2}}, \\ \mathbf{p}^{n+2/3}(i, j) = \theta_{i,j}^* \boldsymbol{\lambda}^{n+2/3}(i, j). \end{cases}$$

4.6. Discretization of problem (3.44). From section 3.6, we have $\boldsymbol{\lambda}^{n+1/3} = \boldsymbol{\lambda}^{n+2/3}$ and $\mathbf{p}^{n+1} = \nabla u^{n+1}$, where u^{n+1} is the solution of the following linear elliptic problem:

$$(4.27) \quad -\nabla^2 u^{n+1} + \tau u^{n+1} = -\nabla \cdot \mathbf{p}^{n+2/3} + \tau f \quad \text{in } \Omega,$$

completed by periodic boundary conditions. We need to discretize this problem. Denoting $\mathbf{p}^{n+2/3}$ by $\tilde{\mathbf{p}}$, we employ the following finite difference scheme to approximate (4.27):

$$(4.28) \quad \left(\begin{array}{cc} \partial_1^- & \partial_2^- \end{array} \right) \left[\left(\begin{array}{c} \partial_1^+ \\ \partial_2^+ \end{array} \right) u^{n+1} - h \left(\begin{array}{c} \tilde{p}_1 \\ \tilde{p}_2 \end{array} \right) \right] + \tau h^2 (f - u^{n+1}) = 0 \quad \text{in } \Omega_h$$

Problem (4.28) is equivalent to

$$(4.29) \quad (\partial_1^- \partial_1^+ + \partial_2^- \partial_2^+ - \tau h^2) u^{n+1} = h(\partial_1^- \tilde{p}_1 + \partial_2^- \tilde{p}_2) - \tau h^2 f.$$

Relation (4.29) can also be written as

$$(4.30) \quad [(\mathcal{I} - \mathcal{S}_1^-)(\mathcal{S}_1^+ - \mathcal{I}) + (\mathcal{I} - \mathcal{S}_2^-)(\mathcal{S}_2^+ - \mathcal{I}) - \tau h^2 \mathcal{I}] u^{n+1}(i, j) = g(i, j),$$

where $g(i, j) = h(\partial_1^- \tilde{p}_1(i, j) + \partial_2^- \tilde{p}_2(i, j)) - \tau h^2 f(i, j)$. From the periodicity of the boundary conditions, it makes sense to use FFT to solve problem (4.30). We then obtain

$$(4.31) \quad w_{i,j} \mathcal{F}(u^{n+1}(i, j)) = \mathcal{F}(g(i, j)),$$

where $w(i, j) = [(1 - e^{-\sqrt{-1}z_i})(e^{\sqrt{-1}z_i} - 1) + (1 - e^{-\sqrt{-1}z_j})(e^{\sqrt{-1}z_j} - 1) - \tau h^2]$, with z_i and z_j as in (4.20). From (4.31), we obtain (with obvious notation)

$$(4.32) \quad u^{n+1} = \text{Real} \left[\mathcal{F}^{-1} \left(\frac{\mathcal{F}(g)}{w} \right) \right],$$

with $\text{Real}(\cdot)$ as in section 4.4. Once u^{n+1} is known we compute \mathbf{p}^{n+1} by

$$(4.33) \quad \mathbf{p}^{n+1} = \nabla^+ u^{n+1} = \begin{pmatrix} \partial_1^+ u^{n+1} \\ \partial_2^+ u^{n+1} \end{pmatrix}$$

(operators have been defined in section 4.2). Finally, the discrete analogue of $\lambda^{n+1}(x)$, for a.e. $x \in \Omega$, is given by

$$(4.34) \quad \begin{cases} \lambda_1^{n+1}(i, j) = \lambda_1^{n+2/3}(i, j), \\ \lambda_2^{n+1}(i, j) = \lambda_2^{n+2/3}(i, j). \end{cases}$$

4.7. Further comments. In sections 4.3 to 4.6, we have provided the details for the discretization for the subproblems associated with the operator-splitting scheme (3.11)–(3.14). In section 5, we will apply the above methodology to the solution of image smoothing problems. It will allow us to demonstrate that with our approach, one can handle the elastica energy functional efficiently and accurately. In addition, we will use further experiments to show the good properties of the proposed method, which include modularity, good stability, and the low cost of the algorithm.

5. Numerical results. In this section, the proposed method is applied to image smoothing to test its effectiveness. All experiments are implemented in MATLAB(R2016a) on a laptop of 8GB RAM and Intel Core i7-7500 CPU: @2.70 GHz 2.90GHz. Note that the intensities of all images are in the range of $[0, 1]$. For simplicity, we also use mesh size $h = 1$. Readers can download the source code of this work from the link <https://ww2.mathworks.cn/matlabcentral/fileexchange/71550-dgt-a-new-operator-splitting-method-for-the-euler-elastica>.

In our experiments, it is reasonable to stop the iteration if the following defined relative error (ReErr) of the solution is smaller than the predefined tolerance tol , i.e.,

$$(5.1) \quad \text{ReErr} = \frac{\|u^{n+1} - u^n\|_2}{\|u^{n+1}\|_2} < tol,$$

where tol is a predefined positive value. In particular, a larger tol may result in a faster stopping of the proposed iterative method.

One of the main advantages of the new method is that it only involves the time step τ as free algorithm parameter to be chosen. The fast speed and robustness of the proposed method are also verified in this section by some specially designed experiments.

In what follows, we apply, in section 5.1, the proposed method to image smoothing. Then, in section 5.2, we compare the speed of convergence and stability properties of this method with those of the THC algorithm [48]. In section 5.2 we further discuss various aspects of the new method and draw some conclusions concerning its ability at solving smoothing problems.

Remark 5.1. In some earlier works (cf. [48, 21, 57]), the Euler elastica model was applied to image denoising. We found, however, that “edge-preserving smoothing” better describes the properties of the proposed method than “denoising”. Indeed, minimizing the Euler elastica energy functional is actually a way to enforce the curvature of an image to be small, a property leading to the smoothing of image details in nonedge regions, while preserving and smoothing the edges. The “denoising” effect is just an intermediate result, “smoothing” actually being the final result of the elastica energy functional minimization. Therefore, in this article, we will use “smoothing” instead of “denoising,” a departure from the terminology we used in previous works.

5.1. Image smoothing. In this section, we first apply (in section 5.1.1) the proposed method to the ROF model (i.e., $b = 0$) and then show, in section 5.1.2, some results of image smoothing with the Euler elastica model.

5.1.1. The proposed method for the ROF model. We apply the primal-dual (PD) approach [13], Chambolle fixed-point (CFP) algorithm proposed in [10], the THC method [48], and the proposed method to the ROF model which is actually a special case of the Euler elastica energy when setting $b = 0$ in (1.2). In Figure 2, we set $b = 0$ and fix $a = 0.1$ for the Euler elastica energy based image restoration problem (1.2), which is just the ROF model. In particular, we implemented our method with $\tau = 0.1$ and $\gamma^n = \max(|\mathbf{p}^{n+1/3}|^2, \sqrt{\tau})$; cf. (3.26). The results of the PD method, the CFP method, the THC method, and the proposed method for the ROF model are shown in Figure 2. All four algorithms are solving the same ROF based problem and their energy converges to the same value. Besides, the restored images and contour maps shown in Figure 2 are also quite similar. We use this example to show that our algorithm also works for the ROF model.

5.1.2. Application of the proposed method to image smoothing . In what follows, we show the capability of the new method at image smoothing. In addition, we also demonstrate the superiority of the Euler elastica model when compared with the ROF model.

We report the results of image smoothing by the Euler elastica model solved by the proposed method, and by the ROF model solved by the CFP method [10] as well. The results demonstrate the competitive ability of edge-preserving image smoothing of the Euler elastica model.

Figure 3 shows the results of the proposed algorithm for Euler's elastica model and the CFP algorithm for the ROF model on four synthetic images. The noisy images are shown in the left column, and the smoothed images by the ROF model and the Euler elastica model are shown in the middle and right columns, respectively. Gaussian white noise with zero mean and a 20 standard deviation is used for the first three images, i.e., “ball,” “star,” and “circle,” a 10 standard deviation being used for the fourth image, i.e., “square.” We acknowledge that all test images in this figure are taken from [48].

From Figure 3, the ROF model is able to well preserve image discontinuous jumps, e.g.,

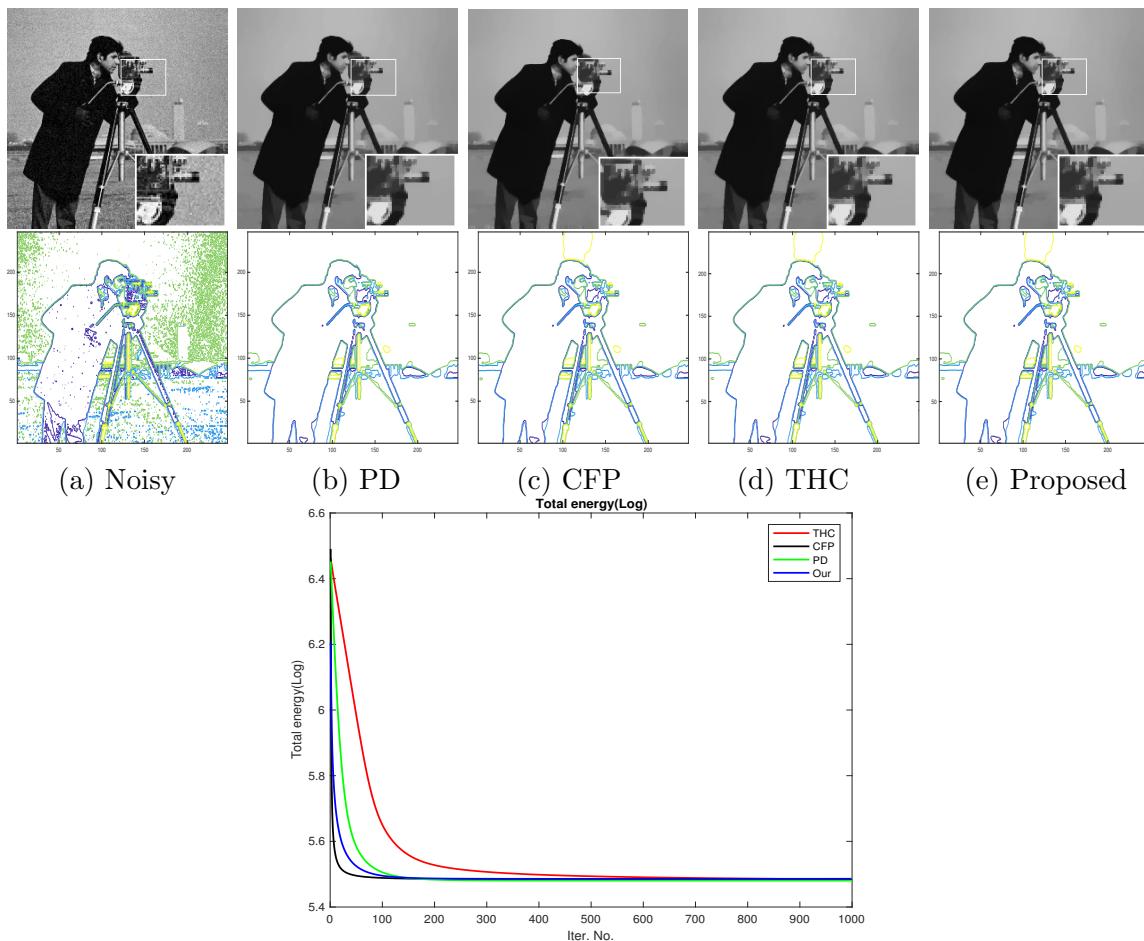


Figure 2. We use the PD approach (b), CFP algorithm (c), THC method (d), and the proposed method (e) to solve the ROF model, which is a special case of the elastica model when $b = 0$, to see if all of methods converge to the same solution and same energy. We use $a = 0.1$ and set $\text{tol} = 1 \times 10^{-5}$ for all methods, and $\tau = 0.1$ for the proposed method. The experiments are tested on the image ‘‘cameraman.’’ The visual results and the contour plots are shown in the first and second rows, respectively. Additionally, when reaching the stopping criterion, the running times are about 1.96 seconds (PD), 2.42 seconds (CFP), 18.57 seconds (THC), and 6.98 seconds (Proposed), respectively. To see the energy changes, we fix 1,000 iterations for all methods and show the energy results in the third row (logarithmic axis). From this figure, we observe that the results of four compared approaches for the ROF model finally converge to the same energy level.

sharp edges, but it leads to some undesired artifacts, for example, the staircase effect in the smooth regions. The Euler elastica model applied via our method not only well preserves the jumps, but also removes the noise without leading to undesired artifacts in the smooth regions. In the last row of Figure 3, we have visualized the contours of the image ‘‘square’’ (noisy on the right, after ROF smoothing in the center, after elastica smoothing on the right). The smoothest contours are the ones obtained by the elastica model via our method due to minimizing the total image curvature and length, while the contours by the CFP method for ROF are unsMOOTH. An analysis of these properties can be found in [61].

Note that for all the experiments reported in Figure 3, all of the involved parameters have the same values, i.e., $a = b = 0.1$, $\tau = 0.1$, $\gamma^n = \max(|\mathbf{p}^{n+1/3}|^2, \sqrt{\tau})$, and $\text{tol} = 1 \times 10^{-5}$. This shows that our method is stable with respect to parameter choice, this property being one of its main advantages.

In Figure 4, we also monitor the energy changes of the subproblems and of the original problem (2.6). From this figure, it is clear that the energies of the $\mathbf{p}^{n+1/3}$ subproblem (3.15), the $\boldsymbol{\lambda}^{n+1/3}$ subproblem (3.17), and of $(n + 2/3)$ subproblem (3.19) (including $\mathbf{p}^{n+2/3}$ and $\boldsymbol{\lambda}^{n+2/3}$ subproblems) all decrease as n increases, while the energy of the \mathbf{p}^{n+1} subproblem (3.41) increases to a stable value. This is because the role of the $(n + 1/3)$ subproblem is to minimize the value of the elastica energy term, without taking the fidelity term into account. Nevertheless, the total energy of the original problem (2.6) always decreases as n increases.

5.2. Advantages of the proposed method. In section 5.1, we applied our method to the Euler elastica model. In what follows, some special experiments will be designed and implemented to illustrate the superiority of the proposed method compared with the THC method [48].

The THC method proposed in [48] is an efficient approach to solve the Euler's elastica problem. As shown by the results reported in [48], the THC method can solve the Euler elastica problem hundredfold times faster than the Chan–Kang–Shen (CKS) method in [45]. After the THC method, some promising approaches (see e.g., [21, 20, 57, 2]) were proposed for the Euler elastica problem. In [21], Duan et al. proposed another FALM to solve the Euler elastica problem based on the framework of the THC method. Afterwards, Duan et al. in [20] applied the THC based method to solve the Euler elastica regularized Mumford–Shah problem, aiming to deal with two-stage image segmentation. Also, Zhu et al. [63] applied the THC method for the Euler elastica regularized Chan–Vese problem, which gets excellent segmentation results. In [57], Zhang et al. proposed a fast linearized augmented Lagrangian approach to solve the Euler elastica problem and applied it to image denoising.

However, the ALM method has some limitations. First, it needs three Lagrange multipliers and three augmentation parameters. For practical applications, it is rather difficult to tune these parameters. We have observed, as shown later in this section, that ALM has a fast convergence and produces very good results when these parameters are chosen correctly. However, if we just change these parameters slightly from their “correct” values, the algorithm will slow down dramatically. Moreover, these parameters are often image dependent and need to be chosen properly for different images.

The method proposed in this article is a simple and new operator splitting approach. It requires only the solution of few simple subproblems. Moreover, it requires the tuning of only one parameter, namely the time-discretization step τ .

In what follows, we will design some numerical experiments to verify the above-mentioned advantages. In particular, the first advantage, i.e., fewer parameters, holds obviously. Thus we need only verify the second and third advantages.

5.2.1. Parameter sensitivity: A numerical testing approach. In order to assess the stability properties, with respect to parameter variations, of the method we introduced in this article, we will proceed as follow. First, we will fix the model parameters a and b . Next, we will tune the augmentation parameters of the augmented Lagrangian in the THC method and

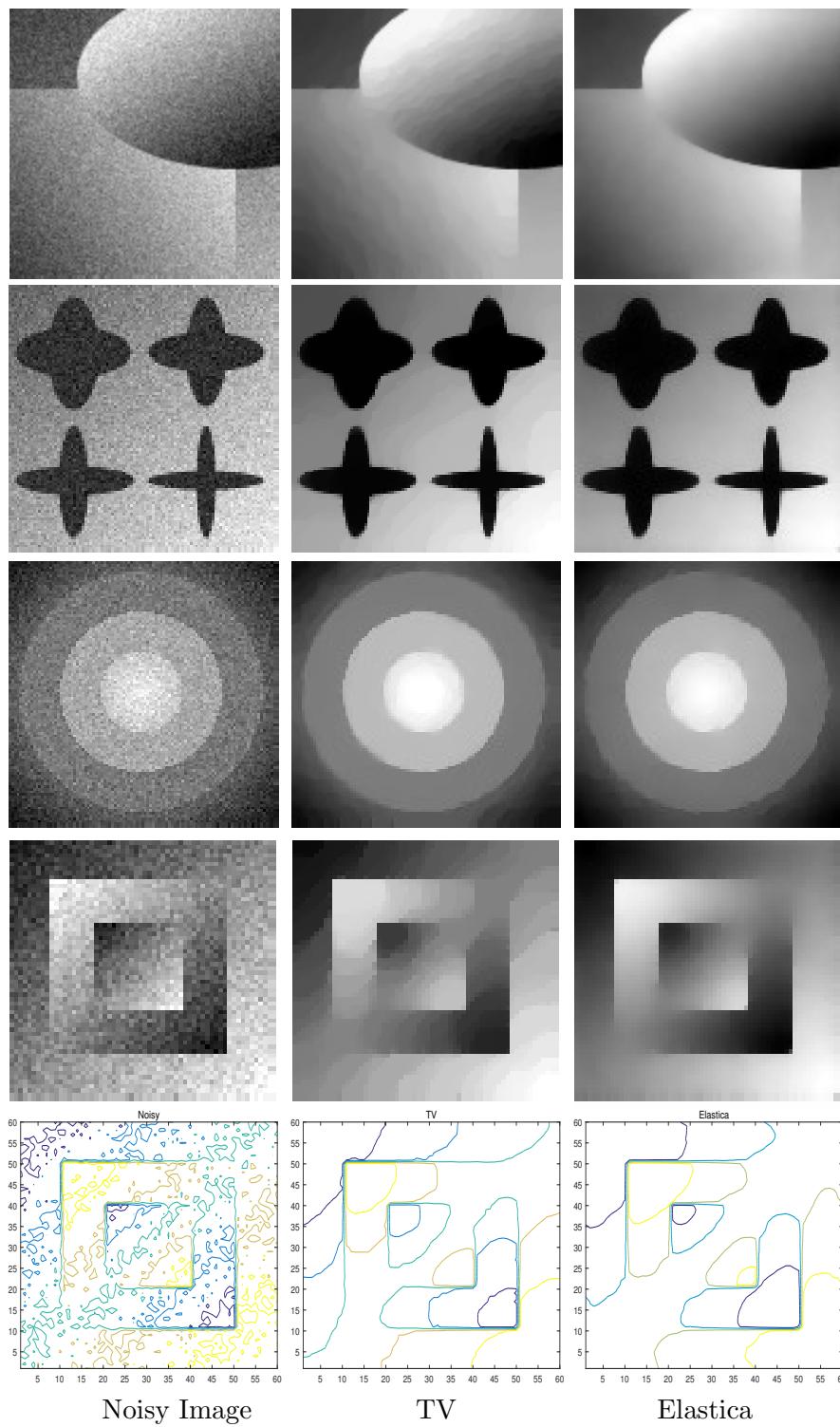


Figure 3. Image smoothing using the ROF and Euler elastica models. Left: Noisy images; Middle: ROF model treated by the CFP method; Right: the Euler elastica model treated by the proposed algorithm. The bottom row shows the contour of the last image. We can see that the elastica model gives images with smoother contours than the ROF model. From the figure, we see that the ROF model creates undesired staircase effects, while the elastica model overcomes it. Note that the parameters in our method for these four test examples are all set as $a = b = 0.1$, $\tau = 0.1$, and $\gamma^n = \max(|\mathbf{p}^{n+1/3}|^2, \sqrt{\tau})$.

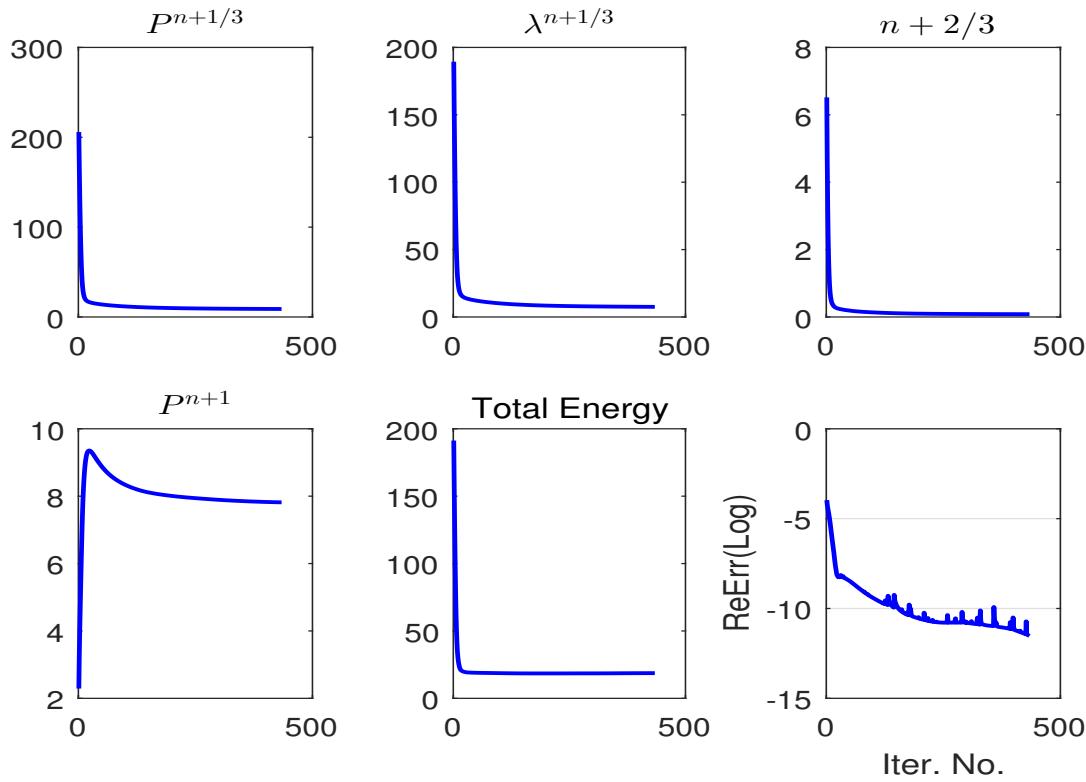


Figure 4. We take the image “square” which is the last one in Figure 3 as an example to show the energy changes of each subproblem, i.e., the $\mathbf{P}^{n+1/3}$ -subproblem (3.15), the $\lambda^{n+1/3}$ -subproblem (3.17), the $(n + 2/3)$ -subproblem (3.19), the \mathbf{P}^{n+1} -subproblem (3.41), and the total energy (2.6).

the time-discretization step τ of our method. Finally, we will compare the results obtained by both methods.

Our intention with the first experiment we performed was testing the sensitivity to one set of parameters for multiple images. For instance, a and b being fixed, we selected one specific image, then tuned the THC method (resp., the proposed method) augmentation parameters (resp., time-discretization step τ) in order to obtain high quality image smoothing. Then, leaving the augmentation parameters and τ unchanged, we applied both methods to the smoothing of other images to see if one still obtains good results. In Figure 5 we have reported the results of the experiment described hereafter: (i) One considers four noisy images, namely “ball” (128×128), “square” (60×60), “star” (100×100), and “Lena” (256×256). (ii) We take $a = b = 0.1$ for both methods to ensure that they solve the same problem. (iii) The tolerance of the stopping criterion is set at $tol = 1 \times 10^{-5}$ (resp., 3×10^{-5}) for “ball,” “square,” and “star” (resp., “Lena”). (iv) Taking “ball” as image of reference, we selected $r_1 = 0.01$, $r_2 = 10$, and $r_3 = 100$ for the THC method (resp., $\tau = 0.1$ and γ given by (3.26) for the proposed method), these values producing high quality smoothing of the noisy “ball” image (see [48] for details about the THC method augmentation parameters r_1 , r_2 , r_3). (v) Finally, keeping the same values for the above parameters, we applied both methods to the other three images. The

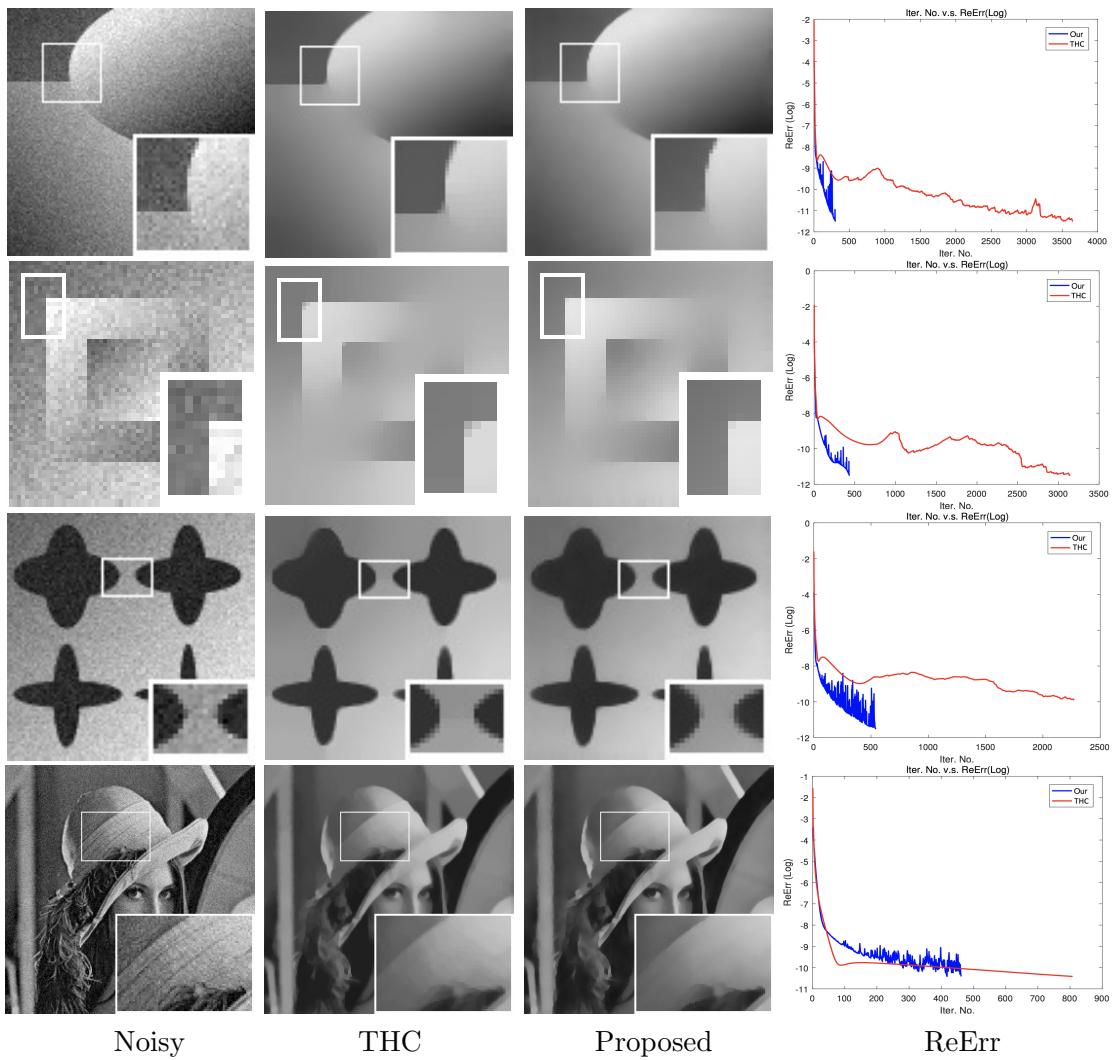


Figure 5. First row: The figure shows that the THC method and the method proposed in this article both obtain good results for $r_1 = 0.01$, $r_2 = 10$, $r_3 = 100$, and $\tau = 0.1$, respectively. For both methods we took $a = b = 0.1$ and $\text{tol} = 1 \times 10^{-5}$ (resp., 3×10^{-5}) for the first three examples (resp., for “Lena” our fourth example). The column on the right shows that the proposed method is significantly faster than the THC one (although its convergence is more oscillatory).

results reported in Figure 5 show that the method we propose in this article is still operational, unlike the THC method that leads to undesired image artifacts. Moreover, the right column of Figure 5 and Table 1 show that the new method requires significantly less iterations than the THC one to verify the stopping criterion. Besides, the average computational time per iteration of the proposed method is also smaller than the one of the THC method, a possible explanation being that the method we propose in this article has fewer subproblems, these subproblems either having closed form solutions or being solvable by fast algorithms such as FFT.

Table 1

In this table we have compared the following performances of the method introduced in this article and of the THC method. Third column: Number of iterations necessary to achieve convergence. Fourth column: Total computational time (in seconds). Fifth column: Averaged computational time per iteration (in seconds).

Image	Method	Iterations	Time (s)	Average time (s)/per iteration
ball (128 × 128)	Proposed	306	2.57	0.008
	THC	3648	37.19	0.010
square (60 × 60)	Proposed	434	1.16	0.002
	THC	3339	10.27	0.003
star(100 × 100)	Proposed	562	3.70	0.006
	THC	2234	17.58	0.007
Lena (256 × 256)	Proposed	462	15.21	0.033
	THC	808	31.18	0.039

As already mentioned the THC method gives poor results when it reaches the stopping criteria given above. As expected, one can make the THC method operational again by either increasing the number of iterations (see Figure 6) or modifying the augmentation parameters (see Figure 7). We see, in particular, on Figure 7(d) that, for the “Lena” image, the THC method with properly tuned augmentation parameters converges to a solution with the same energy than the one we obtain via the method proposed in this article.

In Figure 8, we reported the performances of the THC method for different values of r_1 , r_2 , and r_3 . It is clear from this figure that the THC method is quite sensitive to the values of the augmentation parameters, r_1 and r_2 in particular, implying that augmentation parameter tuning is necessary for the THC method to have good convergence properties.

5.2.2. Speed of convergence comparisons. In this subsection, we further compare the speeds of convergence of the THC and proposed methods. To have fair comparisons, we collected 30 gray images (see Figure 9), either synthetic or natural, and added Gaussian white noise with zero mean and various standard deviations (*std*) to these images.

In Figures 10 to 12 we have reported for $\text{tol} = 1 \times 10^{-5}$ (Figure 10), 5×10^{-5} (Figure 11), and 1×10^{-4} (Figure 12), the averaged number of iterations needed to achieve convergence (first row) and the corresponding computational time (second row). These figures leave no doubt about the superiority of the method we introduced in this article over the augmented Lagrangian based THC method. Indeed, the new method outperforms THC's in terms of number of iterations and computational time per iteration (as shown by Table 2, which displays averaged performances), not to mention its greater simplicity and robustness.

5.3. More comparisons. In this subsection, we compare the proposed method with two state-of-the-art methods in the field of image smoothing using the elastica model. One method is a linearized augmented Lagrangian method (LALM) proposed by Zhang et al. [57], the other is a fast augmented Lagrangian method (FALM) proposed by Duan et al. [21]. The three compared methods are all based on alternating direction methods of multipliers (ADMM) through the augmented Lagrangian approach. Figure 13 shows the computed results for image smoothing by LALM, FALM, and our proposed method for one synthetic image and two natural images. We use the same essential parameters $a = b = 0.1$. After careful tuning of the algorithmic parameters for LALM and FALM, the three compared methods

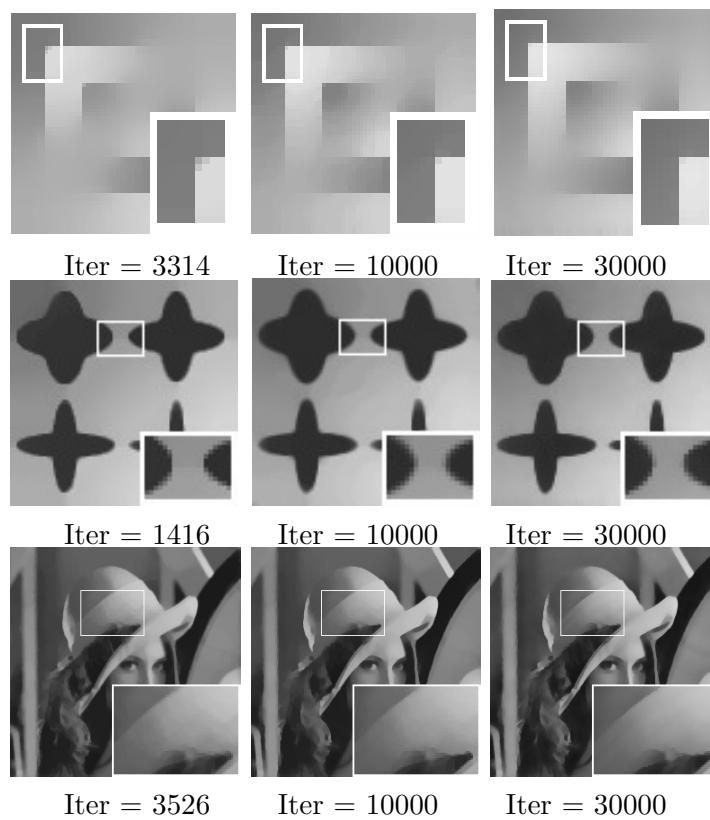


Figure 6. This figure shows that one can improve the smoothing qualities of the THC method (for a nonoptimal choice of the augmentation parameters r_1 , r_2 , and r_3) by requiring more iterations. We did the computations with $r_1 = 0.01$, $r_2 = 10$, and $r_3 = 100$, an augmentation parameter choice which is optimal (or near optimal) for the “ball” image, but not for “square,” “star,” and “Lena.” Note that the left images are identical to the corresponding ones in the second column of Figure 5.

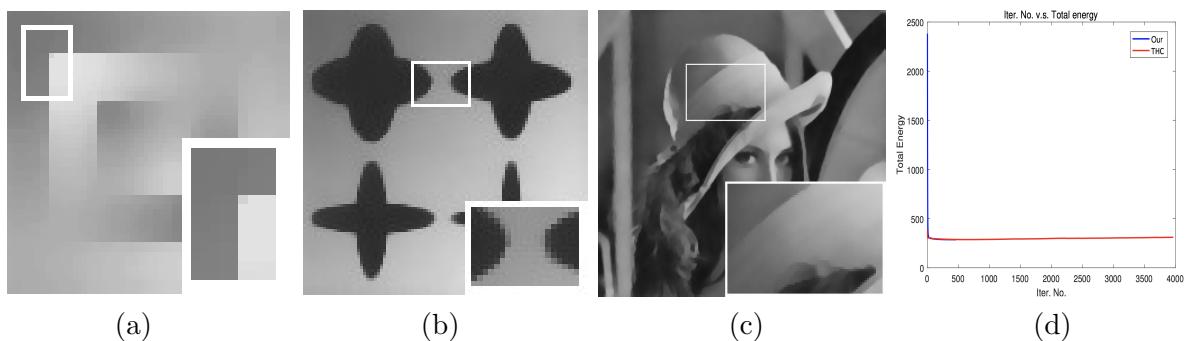


Figure 7. This figure shows that by a proper tuning of the augmentation parameters r_1 , r_2 , and r_3 one can significantly improve the smoothing properties of the THC method. Good choices are: (a) $r_1 = 0.05$, $r_2 = 10$, and $r_3 = 100$ for “square”. (b) $r_1 = 0.005$, $r_2 = 10$, and $r_3 = 100$ for “star”. (c) $r_1 = 0.01$, $r_2 = 10$, and $r_3 = 300$ for “Lena”. In Figure 7(d) we have visualized for the “Lena” image, the variations of the elastica energy versus the iteration number for the proposed method (blue curve) and for the THC method (red curve): both methods reached the same limit. These results show that the THC method suffers from a strong image dependence concerning a good choice for the augmentation parameters r_1 , r_2 , and r_3 .

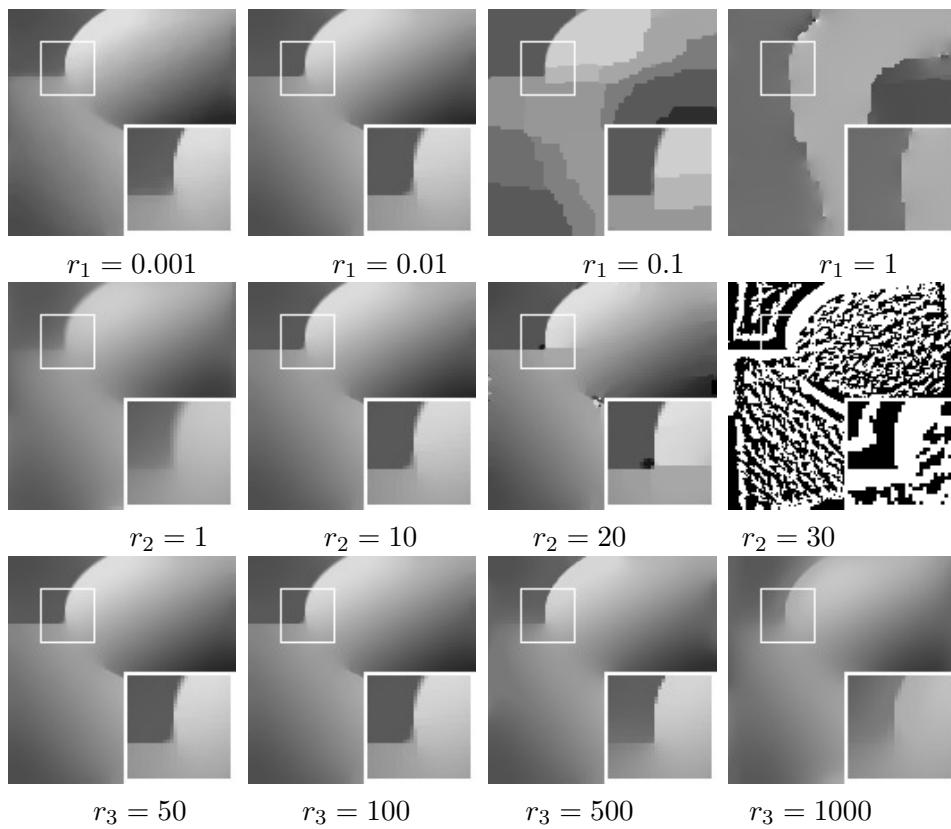


Figure 8. This figure shows the dependence of the THC method computed solution to the augmentation parameters r_1 , r_2 , and r_3 . The default choice being $r_1 = 0.01$, $r_2 = 10$, and $r_3 = 100$, one varied each time only one parameter leaving the other two unchanged (here $a = b = 0.1$ and $\text{tol} = 1 \times 10^{-5}$).

Table 2

Averaged computational time (in seconds and per image) of the method we propose in this article and of the THC method, the images being the 30 gray images displayed in Figure 9.

tol	Method	$std = 0.1$	$std = 0.05$	$std = 0.02$
$\text{tol} = 1 \times 10^{-5}$	Proposed	7.2	7.2	6.1
	THC	74.9	70.3	70.4
$\text{tol} = 5 \times 10^{-5}$	Proposed	4.5	3.7	3.1
	THC	34.4	29.4	29.6
$\text{tol} = 1 \times 10^{-4}$	Proposed	1.9	1.9	1.6
	THC	8.1	8.2	8.0

all obtain very similar results (see the close-up of the images). Especially, since LALM and FALM methods are based on the framework of augmented Lagrangian method, they are very sensitive to penalization parameters as THC; see discussions in section 5.2. It seems that the LALM and FALM are more sensitive to the penalization parameters than THC. For instance, the LALM method could get competitive results for the augmentation parameters $r_1 = 0.01$, $r_2 = 1$, $r_3 = 10$, but if we change r_2 slightly, e.g., setting $r_2 = 2.5$, the result deteriorates

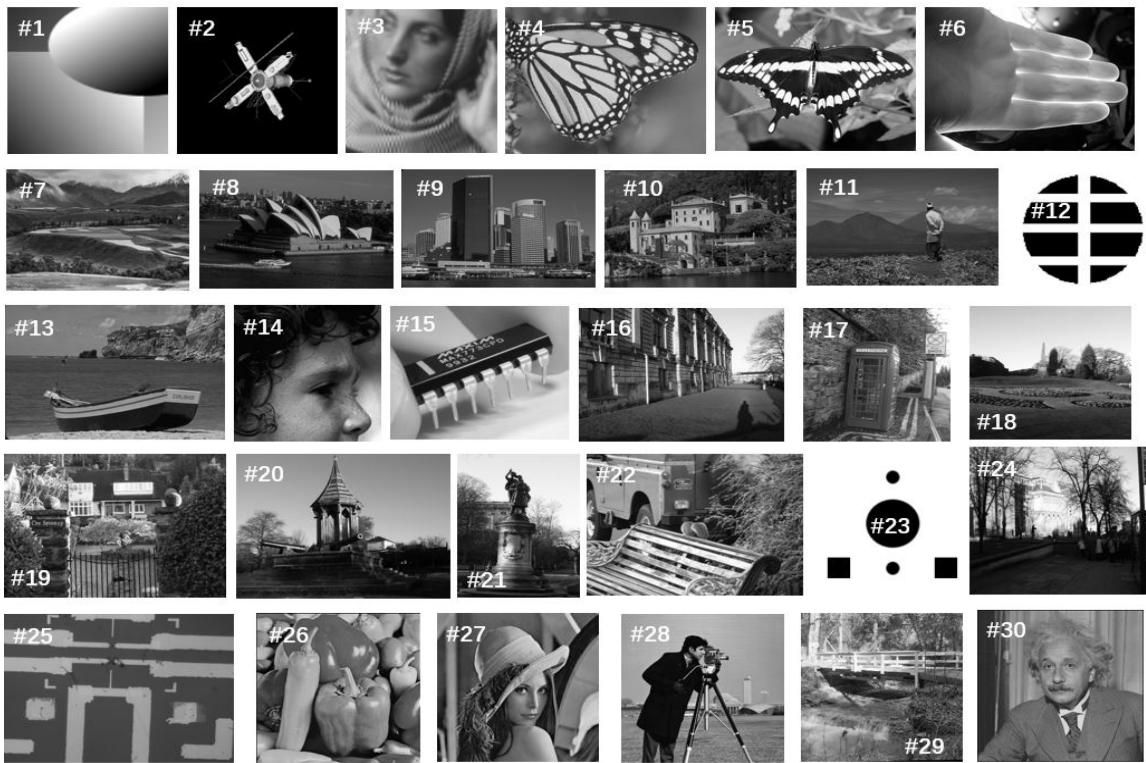


Figure 9. Display of the 30 gray images we used to evaluate and compare the convergence properties of both the method we proposed in this article and the THC method. In order to facilitate the display, we managed to have same size images although they were originally of different sizes.

for the same stopping criterion. In contrast with these methods, the proposed method is nearly parameter free. We do not compare the computational time, since the work in [21] only provides C++ code, but the algorithm in [57] and our method are implemented with MATLAB codes. Moreover, the optimization of the implementation of the codes is also a rather complicated task. It is not easy to give a fair comparison.

6. Conclusions. In this article, we proposed a simple and efficient operator splitting approach to solve the Euler elastica model, and applied the proposed method to image smoothing. Different from the ALM method, the proposed method only needs to tune one parameter, i.e., the time step. Numerical experiments demonstrated that the proposed method works well for the Euler elastica energy and produces good results for image smoothing. Moreover, extensive test experiments were also designed and implemented to assess the stability and effectiveness of the proposed method. Furthermore, the comparisons with the THC method also demonstrated that the proposed method is fast, stable, and robust.

Appendix A. On the Lie scheme for the time-discretization of initial value problems: A brief introduction.

A.1. Generalities. Let us consider the following autonomous initial value problem:

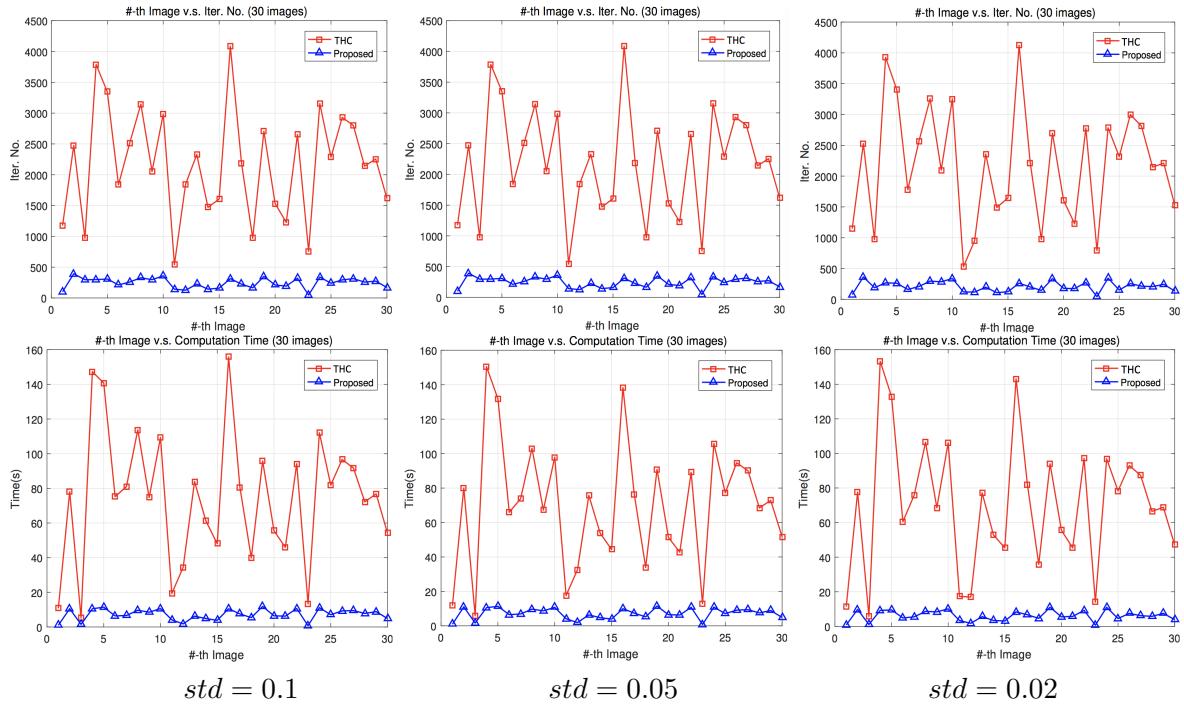


Figure 10. Number of iterations (first row) and corresponding computational time (second row) for the THC method (red curves) and the method introduced in this article (blue curves), both methods using $tol = 1 \times 10^{-5}$ for their respective stopping criterion. The two methods are applied to the 30 images of Figure 9, with added zero mean Gaussian noise, and various standard deviations (std) ($std = 0.1, 0.05$, and 0.02).

$$(A.1) \quad \begin{cases} \frac{d\phi}{dt} + A(\phi) = 0 & \text{on } (0, T) \text{ (with } 0 < T \leq +\infty\text{)}, \\ \phi(0) = \phi_0. \end{cases}$$

Operator A maps the vector space \mathcal{V} into itself (modulo a linear duality map if necessary) and we suppose that $\phi_0 \in \mathcal{V}$. We suppose also that operator A has a *nontrivial decomposition* such as

$$(A.2) \quad A = \sum_{j=1}^J A_j,$$

with $J \geq 2$ (by *nontrivial* we mean that the operators A_j are individually simpler than A).

Many schemes have been designed to take advantage of decomposition (A.2) when solving (A.1), the simplest one, namely the *Lie scheme* will be briefly discussed below.

A.2. Time-discretization of the initial value problem (A.1) by the Lie scheme. Let $\tau (> 0)$ be a *time-discretization step* (we suppose τ uniform, for simplicity): we denote $n\tau$ by t^n . With ϕ^n denoting an approximation of $\phi(t^n)$, the Lie scheme reads as follows (see, e.g.,

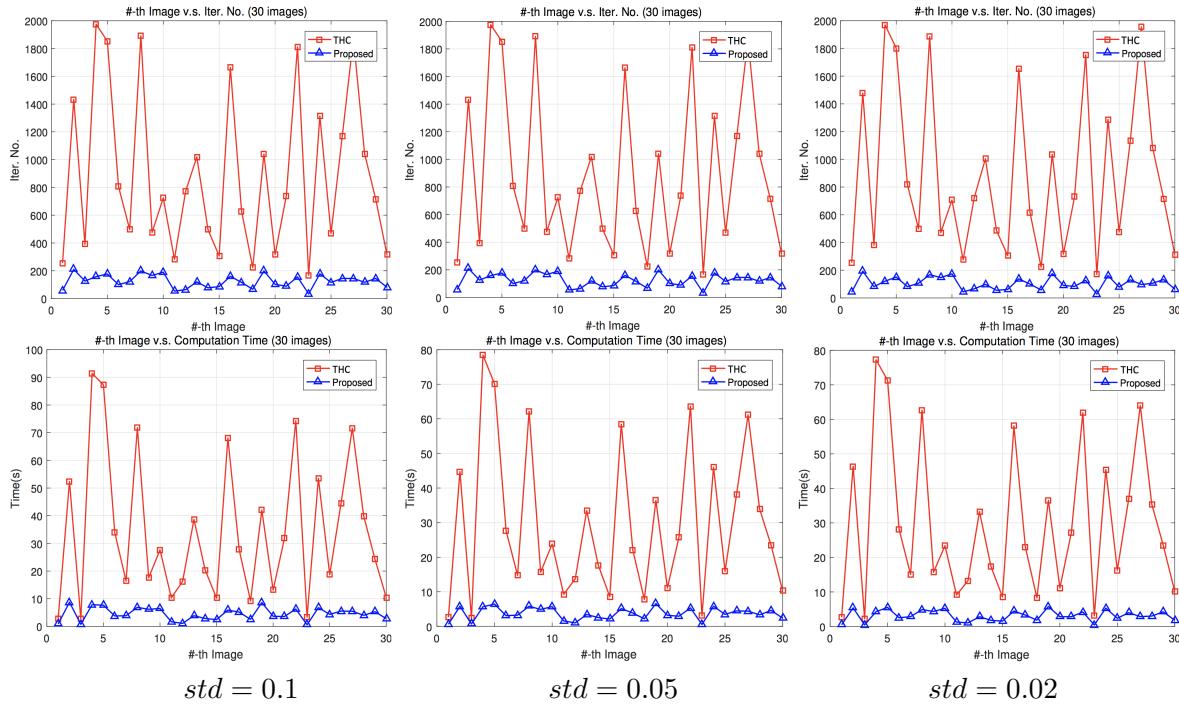


Figure 11. Number of iterations (first row) and corresponding computational time (second row) for the 30 images of Figure 9, using the THC method (red curves) and the method introduced in this article (blue curves). Both methods use $tol = 1 \times 10^{-5}$ for their respective stopping criterion.

[16] and Chapter 6 of [30] for its derivation):

$$(A.3) \quad \phi^0 = \phi_0,$$

then, for $n \geq 0$, $\phi^n \rightarrow \phi^{n+1}$ are updated as follows:

$$(A.4) \quad \begin{cases} \frac{d\phi_j}{dt} + A_j(\phi_j) = 0 & \text{on } (t^n, t^{n+1}), \\ \phi_j(t^n) = \phi^{n+(j-1)/J}; \quad \phi^{n+j/J} = \phi_j(t^{n+1}) \end{cases}$$

for $j = 1, \dots, J$.

If problem (A.1) is taking place in a finite-dimensional space and if the operators A_j are smooth enough, then $\|\phi^n - \phi(t^n)\| = O(\tau)$, function ϕ being the solution of (A.1).

Remark A.1. The above scheme applies also for *multivalued* operators (such as the subdifferential of proper, lower semicontinuous, convex functionals, but in such a case first order accuracy is not guaranteed anymore). We encounter this situation in section 3 of this article.

Remark A.2. Scheme (A.3), (A.4) is semiconstructive in the sense that one still has to solve the subinitial value problems in (A.4) for every j (unless some of them are sufficiently simple to have a closed-form solution). Suppose that one discretizes these subproblems using just *one step of the backward Euler scheme*. The resulting scheme reads as follows:

$$(A.5) \quad \phi^0 = \phi_0,$$

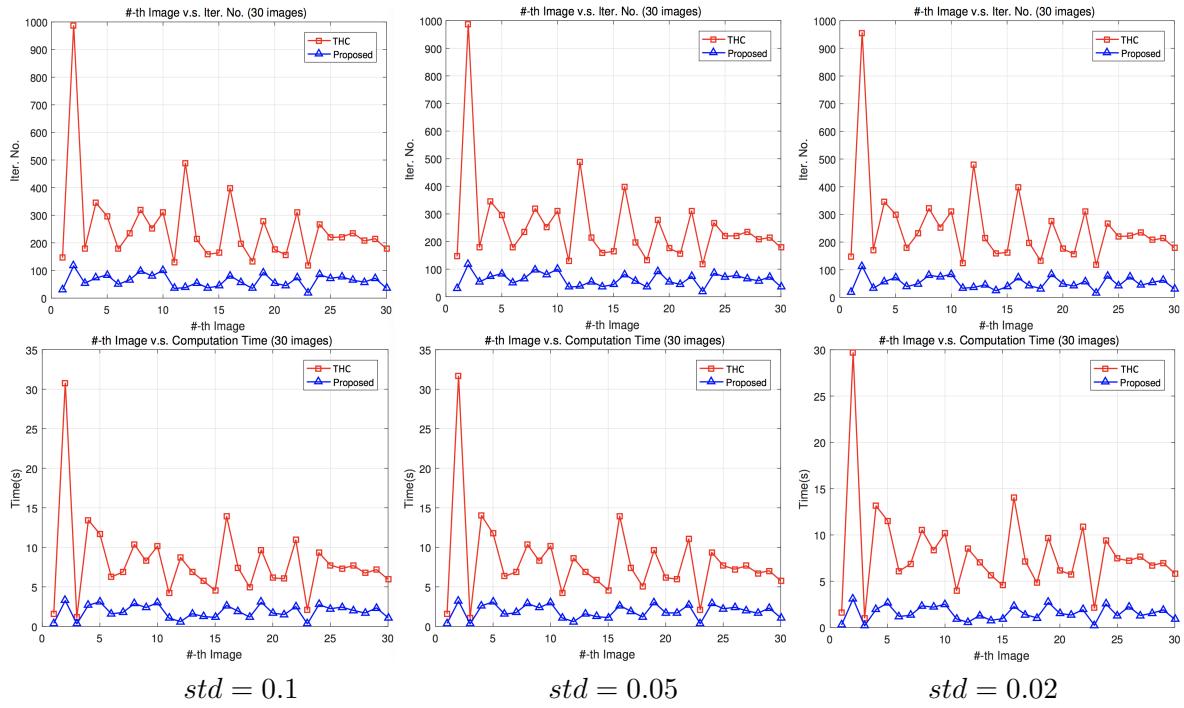


Figure 12. Number of iterations (first row) and corresponding computational time (second row) for the 30 images of Figure 9, using the THC method (red curves) and the method introduced in this article (blue curves). Both methods use $tol = 1 \times 10^{-4}$ for their respective stopping criterion.

then, for $n \geq 0$, assuming that $\phi^{n+(j-1)/J}$ is known, compute $\phi^{n+j/J}$ via

$$(A.6) \quad \frac{\phi^{n+j/J} - \phi^{n+(j-1)/J}}{\tau} + A_j(\phi^{n+j/J}) = 0,$$

for $j = 1, \dots, J$. Especially, if operator A_j is multivalued, one should replace (A.6) by

$$\phi^{n+j/J} + \tau A_j(\phi^{n+j/J}) \ni \phi^{n+(j-1)/J}.$$

Scheme (A.5), (A.6) is known as the Marchuk–Yanenko scheme (see, e.g., [30, 39] for more details, and applications (in computational fluid dynamics, in particular)). As the Lie scheme, the Marchuk–Yanenko scheme is *generically first order accurate at best*. However, this low order accuracy is compensated by this scheme simplicity, flexibility, and robustness, making it a method of choice for those situations where J is large and some of the operators A_j are poorly differentiable and even nonsmooth (the numerical simulation of time dependent *visco-plastic flows* [17]) and the elastica energy minimization problem considered in this work provides such situations.

A.3. Asymptotic properties of the Lie and Marchuk–Yanenko schemes. Consider now the *steady-state problem* associated with (A.1), namely

$$(A.7) \quad A(\phi) = 0.$$

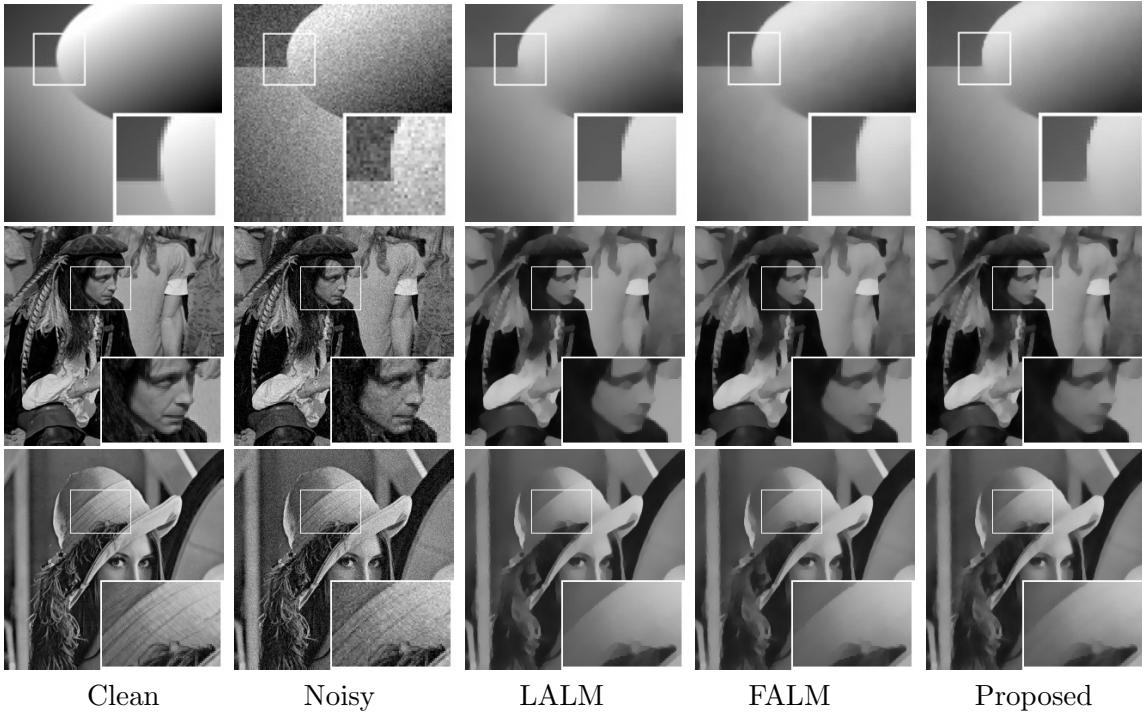


Figure 13. The results of image smoothing by LALM, FALM, and the proposed method for the images “ball” (first row), “man” (second row), and “Lena” (third row). All methods took $a = b = 0.1$ and $\text{tol} = 1 \times 10^{-5}$. For the LALM method, we set $r_1 = 0.01$, $r_2 = 2$, $r_3 = 10$, and $\delta_1 = 0.1$, $\delta_2 = 3 \times 10^{-4}$ for all examples. For the FALM method, we set $r_1 = 0.1$, $r_2 = 100$, $r_3 = 500$, and $\eta = 100$ in the code.

Assume that problem (A.1) has steady-state solutions. These solutions are necessarily solutions of problem (A.7). A classical method to solve (A.7) is then to integrate (A.1) on the time interval $(0, +\infty)$, with (if possible) ϕ_0 properly chosen. Suppose that decomposition (A.2) holds. It is then tempting to use either the Lie or Marchuk–Yanenko scheme to capture these steady-state solutions. It is where one encounters the *main drawback* of the Lie and Marchuk schemes, namely their *asymptotic inconsistency*. Indeed, one observes the following phenomena, generically:

1. If converging for $j = 1, \dots, J$, the sequences $(u^{n+j/J})_{n \geq 0}$ converge to different limits than the solution of (A.7), the distance between them being $O(\tau)$ at best.
2. None of the above limits is a steady-state solution, but their distance to a steady-state solution converges to 0 as $\tau \rightarrow 0$ (if a steady-state solution does exist).

This behavior has been proved in, e.g., Chapter 6 of [30], for simple situations where the space \mathcal{V} is finite dimensional and the operators A_j affine. These properties for the A_j operators are not satisfied in the applications to elastica energy minimization. So the convergence of the schemes proposed in this work for elastica energy minimization, we cannot use existing convergence theory and need to study them separately.

There are several ways to overcome asymptotic inconsistency: among them, *averaging* and *symmetrization*. Actually, the simplest way to reduce the so-called *splitting error* is to take a

small time step τ . That is the approach followed by the authors of this article.

Remark A.3. The *Douglas–Rachford* and *Peaceman–Rachford* alternating direction schemes are asymptotically consistent, but they have drawbacks of their own, their (relatively) slow convergence to steady state, in particular. They also have difficulties at handling large values of J , particularly if some of the operators A_j are nonsmooth or multivalued. We suggest the interested readers look at [29], which is dedicated to the numerical solution of

$$(A.8) \quad |\nabla\Psi| - \frac{|1 - \mathbf{V} \cdot \nabla\Psi|}{c} = 0,$$

an *Eikonal type equation* from *acoustics*, modeling *sound propagation in moving media* (in (A.8), $c(> 0)$ is the medium at rest local speed of sound, and \mathbf{V} is the medium motion velocity). Article [29] contains a thorough comparison of operator-splitting methods applied to the solution of problem (A.8). Among these splitting methods, let us mention the Lie and Marchuk–Yanenko splitting schemes, and the Peaceman–Rachford and Douglas–Rachford alternating direction methods.

Remark A.4. The choice of the *time-discretization step* τ is a delicate issue a priori. Indeed a small τ is required to have a small splitting error. On the other hand, a small τ may imply a large number of time steps to achieve convergence. A classical way to overcome this difficulty is to use a sequence $(\tau_n)_{n \geq 0}$ of time steps, this sequence verifying

$$(A.9) \quad \left\{ \begin{array}{l} \forall n \geq 0, \tau_n > 0 \text{ and } \tau_n > \tau_{n+1}, \\ \lim_{n \rightarrow +\infty} \tau_n = 0, \\ \sum_{n=0}^{+\infty} \tau_n = +\infty. \end{array} \right.$$

Actually, we never used (A.9), or other variable time-step strategies. The main reason for that being that we obtained good results with “not so small” fixed time steps τ (by *good*, we mean that these results were quantitatively comparable to results obtained by other methods (THC and variants of it, for example) but much faster in terms of the number of iterations and CPU time per iteration). These comparisons are further discussed in section 5.

Acknowledgments. The authors would like to thank Dr. Zhang in [57] and Dr. Duan in [21] for providing codes for the comparisons. The authors also thank the anonymous referees of this article for helpful comments and suggestions.

REFERENCES

- [1] E. BAE, J. SHI, AND X.-C. TAI, *Graph cuts for curvature based image denoising*, IEEE Trans. Image Process., 20 (2011), pp. 1199–1210.
- [2] E. BAE, X. C. TAI, AND W. ZHU, *Augmented Lagrangian method for an Euler's elastica based segmentation model that promotes convex contours*, Inverse Probl. Imaging, 11 (2017), pp. 1–23.
- [3] C. BALLESTER, M. BERTALMIO, V. CASELLES, G. SAPIRO, AND J. VERDERA, *Filling-in by joint interpolation of vector fields and gray levels*, IEEE Trans. Image Process., 10 (2001), pp. 1200–1211.
- [4] K. BREDIES, K. KUNISCH, AND T. POCK, *Total generalized variation*, SIAM J. Imaging Sci., 3 (2010), pp. 492–526, <https://doi.org/10.1137/090769521>.

- [5] K. BREDIES, T. POCK, AND B. WIRTH, *A convex, lower semicontinuous approximation of Euler's elastica energy*, SIAM J. Math. Anal., 47 (2015), pp. 566–613, <https://doi.org/10.1137/130939493>.
- [6] C. BRITO-LOEZA AND C. KEN, *Fast numerical algorithms for Euler's elastica inpainting model*, Int. J. Mod. Math., 5 (2010), pp. 157–182.
- [7] A. BRUCKSTEIN, R. HOLT, AND A. NETRAVALI, *Discrete elastica*, Appl. Anal., 78 (2001), pp. 453–485.
- [8] R. L. BURDEN AND J. D. FAIRES, *Fixed-Point Iteration*, 3rd ed., Numerical Analysis, PWS Publishers, Boston, 1985.
- [9] V. CASELLES, A. CHAMBOLLE, AND M. NOVAGA, *Total variation in imaging*, Handbook of Mathematical Methods in Imaging, Springer, New York, 2015, pp. 1455–1499.
- [10] A. CHAMBOLLE, *An algorithm for total variational minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.
- [11] A. CHAMBOLLE, V. DUVAL, G. PEYRE, AND C. POON, *Geometric properties of solutions to the total variation denoising problem*, Inverse Problems, 33 (2016), 015002.
- [12] A. CHAMBOLLE, S. E. LEVINE, AND B. J. LUCIER, *An upwind finite-difference method for total variation-based image smoothing*, SIAM J. Imaging Sci., 4 (2011), pp. 277–299, <https://doi.org/10.1137/090752754>.
- [13] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
- [14] A. CHAMBOLLE AND T. POCK, *Total Roto-Translational Variation*, preprint, <https://arxiv.org/abs/1709.09953>, 2017.
- [15] T. F. CHAN, G. H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1977, <https://doi.org/10.1137/S1064827596299767>.
- [16] A. J. CHORIN, T. J. HUGHES, M. F. McCACKEN, AND J. E. MARSDEN, *Product formulas and numerical algorithms*, Comm. Pure Appl. Math., 31 (1978), pp. 205–256.
- [17] E. J. DEAN, R. GLOWINSKI, AND G. GUIDOBONI, *On the numerical simulation of Bingham visco-plastic flow: Old and new results*, J. Non-Newton. Fluid Mech., 142 (2007), pp. 36–62.
- [18] D. L. DONOHO, *De-noising by soft-thresholding*, IEEE Trans. Inform. Theory, 41 (1995), pp. 613–627.
- [19] M. DROSKE AND A. BERTOZZI, *Higher-order feature-preserving geometric regularization*, SIAM J. Imaging Sci., 3 (2010), pp. 21–51, <https://doi.org/10.1137/090751694>.
- [20] Y. DUAN, W. HUANG, J. ZHOU, H. CHANG, AND T. ZENG, *A two-stage image segmentation method using Euler's elastica regularized Mumford-Shah model*, International Conference on Pattern Recognition (ICPR), 2014, pp. 118–123.
- [21] Y. DUAN, Y. WANG, X.-C. TAI, AND J. HAHN, *A fast augmented Lagrangian method for Euler's elastica models*, Numer. Math. Theory Methods Appl., 6 (2013), pp. 47–71.
- [22] G. DUVAUT AND J. LIONS, *Inequalities in Mechanics and Physics*, Springer-Verlag, Berlin, New York, 1976.
- [23] N. EL-ZEHIRY AND L. GRADY, *Fast global optimization of curvature*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3257–3264.
- [24] S. ESEDOGLU AND R. MARCH, *Segmentation with depth but without detecting junctions*, J. Math. Imaging Vision, 18 (2003), pp. 7–15.
- [25] S. ESEDOGLU, S. RUUTH, AND R. TSAI, *Threshold dynamics for shape reconstruction and disocclusion*, Proceedings of the IEEE International Conference on Image Processing (ICIP), 2005, pp. 502–505.
- [26] E. ESSER, *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman*, UCLA CAM Report 09-31, UCLA, Los Angeles, CA, 2009.
- [27] E. ESSER, X. ZHANG, AND T. CHAN, *A General Framework for a Class of First Order Primal-Dual Algorithms for TV Minimization*, UCLA CAM Report 09-67, UCLA, Los Angeles, CA, 2009.
- [28] M. FORTIN AND R. GLOWINSKI, *Augmented Lagrangian Methods*, Stud. Math. Appl. 15, North-Holland, Amsterdam, 1983.
- [29] G. GLOWINSKI, Y. LEUNG, AND J. QIAN, *Operator-splitting based fast sweeping Methods for isotropic wave propagation wave propagation in a moving fluid*, SIAM J. Sci. Comput., 38 (2016), pp. A1195–A1223, <https://doi.org/10.1137/15M1043868>.
- [30] R. GLOWINSKI, *Finite element methods for incompressible viscous flow*, In Handbook of Numerical Analysis, Vol. IX, P. G. Ciarlet and J. L. Lions, eds., North-Holland, Amsterdam, 2003, pp. 3–1176.

- [31] R. GLOWINSKI AND P. LE TALLEC, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia, 1989, <https://doi.org/10.1137/1.9781611970838>.
- [32] R. GLOWINSKI, S. OSHER, AND W. YIN, *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer, Cham, 2017.
- [33] R. GLOWINSKI, T.-W. PAN, AND X.-C. TAI, *Some facts about operator-splitting and alternating direction methods*, Splitting Methods in Communication, Imaging, Science, and Engineering, Springer, Cham, 2016, pp. 19–94.
- [34] R. GLOWINSKI AND A. WACHS, *On the numerical simulation of viscoplastic fluid flow*, Handbook of Numerical Analysis 16, Elsevier/North-Holland, Amsterdam, 2011, pp. 483–717.
- [35] J. B. GREER AND A. L. BERTOZZI, *Traveling wave solutions of fourth order PDEs for image processing*, SIAM J. Math. Anal., 36 (2004), pp. 38–68, <https://doi.org/10.1137/S0036141003427373>.
- [36] V. GRIMM, R. I. MCLACHLAN, R. I. MCLAREN, G. R. W. QUISPTEL, AND C. B. SCHÖNLIEB, *Discrete gradient methods for solving variational image regularisation models*, J. Phys. A, 50 (2017).
- [37] S. H. KANG, W. ZHU, AND J. J. SHEN, *Illusory shapes via corner fusion*, SIAM J. Imaging Sci., 7 (2014), pp. 1907–1936, <https://doi.org/10.1137/140959043>.
- [38] V. KOLMOGOROV, T. POCK, AND M. ROLINEK, *Total variation on a tree*, SIAM J. Imaging Sci., 9 (2016), pp. 605–636, <https://doi.org/10.1137/15M1010257>.
- [39] G. I. MARCHUK, *Splitting and Alternating Direction Methods*, in Handbook of Numerical Analysis, Vol. I, P. G. Ciarlet and J. L. Lions, eds., North-Holland, Amsterdam, (1990), pp. 197–462.
- [40] S. MASNOU AND J.-M. MOREL, *Level lines based disocclusion*, Proceedings of the IEEE International Conference on Image Processing (ICIP), 1998, pp. 259–263.
- [41] S. MASNOU AND J.-M. MOREL, *On a variational theory of image amodal completion*, Rend. Sem. Mat. Univ. Padova, 116 (2006), pp. 211–252.
- [42] M. NITZBERG, D. MUMFORD, AND T. SHIOTA, *Filtering, Segmentation and Depth*, Lecture Notes in Comput. Sci. 662, Springer-Verlag, Berlin, 1993.
- [43] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
- [44] T. SCHOENEMANN, F. KAHL, AND D. CREMERS, *Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation*, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2009.
- [45] J. SHEN, S. H. KANG, AND T. F. CHAN, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63 (2002), pp. 564–592, <https://doi.org/10.1137/S0036139901390088>.
- [46] G. STEIDL AND T. TEUBER, *Removing multiplicative noise by Douglas-Rachford splitting methods*, J. Math. Imaging Vision, 36 (2010), pp. 168–184.
- [47] X. C. TAI AND J. DUAN, *A simple fast algorithm for minimization of the elastica energy combining binary and level set representations*, Int. J. Numer. Anal. Model., 14 (2017), pp. 809–821.
- [48] X.-C. TAI, J. HAHN, AND G. J. CHUNG, *A fast algorithm for Euler's Elastica model using augmented Lagrangian method*, SIAM J. Imaging Sci., 4 (2011), pp. 313–344, <https://doi.org/10.1137/100803730>.
- [49] X. C. TAI AND C. WU, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF model*, in SSVM'09: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 502–513.
- [50] Y. WANG, J. YANG, W. YIN, AND Y. ZHANG, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272, <https://doi.org/10.1137/080724265>.
- [51] J. WEICKERT, B. M. TER HARR ROMENY, AND M. A. VIERGEVER, *Efficient and reliable schemes for nonlinear diffusion filtering*, IEEE Trans. Image Process, 7 (2001), pp. 398–410.
- [52] P. WEISS, L. BLANC-FÉRAUD, AND G. AUBERT, *Efficient schemes for total variation minimization under constraints in image processing*, SIAM J. Sci. Comput., 31 (2009), pp. 2047–2080, <https://doi.org/10.1137/070696143>.
- [53] C. WU AND X.-C. TAI, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models*, SIAM J. Imaging Sci., 3 (2010), pp. 300–339, <https://doi.org/10.1137/090767558>.
- [54] M. YASHTINI AND S. H. KANG, *Alternating Direction Method of Multiplier for Euler's Elastica-Based Denoising*, in Scale Space and Variational Methods in Computer Vision (SSVM), Lecture Notes in

- Comput. Sci., 9087, Springer, Cham, 2015, pp. 690–701.
- [55] M. YASHTINI AND S. H. KANG, *A Fast relaxed normal two split method and an effective weighted TV approach for Euler's elastica image inpainting*, SIAM J. Imaging Sci., 9 (2016), pp. 1552–1581, <https://doi.org/10.1137/16M1063757>.
 - [56] W. YIN, S. OSHER, D. GOLDFARB, AND F. DARBON, *Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168, <https://doi.org/10.1137/070703983>.
 - [57] J. ZHANG, R. CHEN, C. DENG, AND S. WANG, *Fast linearized augmented Lagrangian method for Euler's elastica model*, Numer. Math. Theory Methods Appl., 10 (2017), pp. 98–115.
 - [58] J. ZHANG AND C. KEN, *A new augmented Lagrangian primal dual algorithm for elastica regularization*, J. Algorithms Comput. Technol., 10 (2016), pp. 325–338.
 - [59] X. ZHANG, M. BURGER, AND S. OSHER, *A unified primal-dual algorithm framework based on Bregman iteration*, J. Sci. Comput., 46 (2011), pp. 20–46.
 - [60] M. ZHU AND T. CHAN, *An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration*, UCLA CAM Report 08-34, UCLA, Los Angeles, CA, 2008.
 - [61] W. ZHU AND T. CHAN, *Image denoising using mean curvature of image surface*, SIAM J. Imaging Sci., 5 (2012), pp. 1–32, <https://doi.org/10.1137/110822268>.
 - [62] W. ZHU, T. CHAN, AND S. ESEDOĞLU, *Segmentation with depth: A level set approach*, SIAM J. Sci. Comput., 28 (2006), pp. 1957–1973, <https://doi.org/10.1137/050622213>.
 - [63] W. ZHU, X. C. TAI, AND T. CHAN, *Image segmentation using Euler's elastica as the regularization*, J. Sci. Comput., 57 (2013), pp. 414–438.