

A New Operator Splitting Method for Euler's Elastica Model

Liang-Jian Deng^a, Roland Glowinski^b, Xue-Cheng Tai^c

^a*School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China.*

^b*Department of Mathematics, University of Houston, Houston, Texas, 77204, USA; Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong*

^c*Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong*

Abstract

Euler's elastica model has a wide range of applications in Image Processing and Computer Vision. However, the non-convexity, the non-smoothness 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 sub-problems 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 sub-problems 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.

Keywords: Euler elastica energy, Operator splitting, Space projection, Image smoothing

1. Introduction

The Euler elastica energy of a planar curve is defined as the following functional,

$$E(\mathcal{C}) = \int_{\mathcal{C}} (a + b\kappa^2) ds, \quad (1)$$

where \mathcal{C} represents the planar curve whose curvature is κ , s stands for arc length, a and b are two positive parameters. Especially, if $a = 0$, $E(\mathcal{C})$ measures the twisting energy of the curve that is related to the curvature; if $b = 0$, then $E(\mathcal{C})$ measures the total length of the curve.

[☆]This is only an example

Email addresses: liangjian.deng@uestc.edu.cn (Liang-Jian Deng), roland@math.uh.edu (Roland Glowinski), xuechengtai@hkbu.edu.hk (Xue-Cheng Tai)

For *imaging applications*, the generalized Euler elastica energy is defined by

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

where in (2), Ω is a bounded domain of \mathbf{R}^2 (a rectangle, typically), a and b are like in (1), v is a function of two variables belonging to an appropriate functional space containing (in principle) the underlying image, and $dx = dx_1 dx_2$.

Recently, the Euler elastica energy defined by (2) found applications in image processing, such as: denoising [1, 2], segmentation [3, 4, 5, 6], inpainting [7, 8, 1, 9], zooming [1], illusory contour [10, 11, 12], segmentation with depth [13, 14, 15]. In [7], Chan et al. discuss the mathematical foundation of the Euler elastica model and its mathematical properties, motivated by applications to image inpainting. In addition, the authors of [7] discussed also a numerical PDE method, in order to solve the associated nonlinear problem. In [1], Tai et al. 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. More recently, Zhang et al. proposed in [2] a linearized strategy to speed up ALM, and applied it to the solution of image denoising problems. In [9], two numerical algorithms were proposed to solve inpainting related problems involving the Euler elastica energy (2): The first algorithm is an improved variant of the ALM based algorithm discussed in [1]. The second algorithm is obtained by applying a split-Bregman method to a linearized elastica model proposed in [16]. Following an idea from [17], Masnou and Morel proposed in [10] a novel method to handle the elastica energy functional and applied it to the solution of illusory contour problems. In [11], Kang et al. used the Euler elastica energy as an effective tool to fuse the scattered corner bases. In [12], 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 [18], Bredies, Pock and Wirth suggested using as smoothing functional a convex, lower semi-continuous 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 [18] 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

$$\min_v \left[\int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla v}{|\nabla v|} \right|^2 \right) |\nabla v| dx + \frac{1}{2} \int_{\Omega} |f - v|^2 dx \right], \quad (3)$$

where v is as in (2), and f is the image we are trying to denoise. The first term in the functional in (3) 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 (3), and close variants of it.

The non-convexity, the non-smoothness, and the high-order of the derivatives it contains, make the fast and robust solution of problem (3) a very challenging task. So far, there are only few methods to solve problems such (3); let us mention among them: two graph-cut based methods ([19, 16]), an integer linear programming (ILP) method ([20]), a method based on the approximation of the Euler elastica energy ([21]), and the augmented Lagrangian method (ALM) (see, e.g., [1]). The method discussed in [1] is a

particular realization of the Alternating Method of Multipliers (ADMM), a well-known method from Mathematical Programming (see, e.g., [22] 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 [1], several extensions were proposed for solving, via the Euler elastica energy functional, a large variety of imaging problems (see [3, 23, 24, 6]). Actually, readers can find further curvature based methods in [25, 26, 27, 28, 29]. Primal-dual methods have been applied also to derive fast algorithms to handle the total variation (TV) imaging model, introduced in [30] by Rudin, Osher, and Fattorini (ROF). For instance, Droske and Bertozzi in [31] 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 more related literatures [32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48] 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 (3). 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 (3) 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 sub-problems 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 ALM method of [1], the method introduced in this article has the following advantages:

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

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 smoothing application, with some experiments designed to show the superiority of the new method. Some conclusions are drawn in Section 6.

2. A reformulation of problem (3)

From Section 1, Euler's elastica problem reads as

$$\min_{v \in \mathcal{V}} \left[\int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla v}{|\nabla v|} \right|^2 \right) |\nabla v| dx + \frac{1}{2} \int_{\Omega} |f - v|^2 dx \right], \quad (4)$$

with \mathcal{V} a functional space of the Sobolev's type, typically. If one is willing to replace \min by \inf in (4), one can take $\mathcal{V} = \mathcal{H}^2(\Omega)$ if we assume that no specific condition is imposed a priori to v on the boundary Γ of Ω . 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 (4) is that it makes no sense on those parts of Ω where v vanishes. 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., [49, 22, 50]) is to replace $\frac{\nabla v}{|\nabla v|}$ by a vector-valued function $\boldsymbol{\mu}$ verifying

$$\boldsymbol{\mu} \cdot \nabla v = |\nabla v|, \quad |\boldsymbol{\mu}| \leq 1, \quad (5)$$

with $|\boldsymbol{\mu}| = \sqrt{\mu_1^2 + \mu_2^2}$, $\forall \boldsymbol{\mu} = (\mu_1, \mu_2)$, and then problem (4) by

$$\min_{(v, \boldsymbol{\mu}) \in \mathcal{W}} \left[\int_{\Omega} (a + b |\nabla \cdot \boldsymbol{\mu}|^2) |\nabla v| dx + \frac{1}{2} \int_{\Omega} |f - v|^2 dx \right], \quad (6)$$

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:

Proposition 1 Suppose that $(u, \boldsymbol{\lambda})$ is solution of problem (6). We have then

$$\int_{\Omega} u dx = \int_{\Omega} f dx. \quad (7)$$

Proof. Consider the pair $(u + c, \boldsymbol{\lambda})$, where $c \in \mathcal{R}$. 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 (6). Since $\nabla(u + c) = \nabla u$ we have $J_1(u + c, \boldsymbol{\lambda}) = J_1(u, \boldsymbol{\lambda})$. On the other hand

$$J_2(u + c, \boldsymbol{\lambda}) = \frac{1}{2} \int_{\Omega} |u + c - f|^2 dx = J_2(u, \boldsymbol{\lambda}) + c \int_{\Omega} (u - f) dx + |\Omega| \frac{c^2}{2}, \quad (8)$$

with $|\Omega|$ = measure of Ω . The function u being fixed, the quadratic function of c in the right-hand side of (8) 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 $\int_{\Omega} u dx = \int_{\Omega} f dx$. \square

Remark 2.1: It is a common practice to assume periodicity when working with image processing problems. Proposition 1 still holds if one consider the minimization of the elastica functional in a space of sufficiently smooth periodic functions (functions defined over a two-dimensional torus). In this work, unless otherwise specified, we assume all the functions we are using are periodic in both the x_1 - and x_2 - directions.

Let us define the sets Σ_f and S by

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

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 equivalence between problem (6) and

$$\min_{(\mathbf{q}, \boldsymbol{\mu}) \in (\mathcal{L}^2(\Omega))^2 \times (\mathcal{L}^2(\Omega))^2} \left[\int_{\Omega} (a + b|\nabla \cdot \boldsymbol{\mu}|^2) |\mathbf{q}| dx + \frac{1}{2} \int_{\Omega} |v_{\mathbf{q}} - f|^2 dx + I_{\Sigma_f}(\mathbf{q}) + I_S(\mathbf{q}, \boldsymbol{\mu}) \right], \quad (9)$$

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

$$I_{\Sigma_f}(\mathbf{q}) = \begin{cases} 0, & \text{if } \mathbf{q} \in \Sigma_f, \\ +\infty, & \text{if } \mathbf{q} \in (\mathcal{L}^2(\Omega))^2 \setminus \Sigma_f, \end{cases} \quad (10)$$

and

$$I_S(\mathbf{q}, \boldsymbol{\mu}) = \begin{cases} 0, & \text{if } (\mathbf{q}, \boldsymbol{\mu}) \in S, \\ +\infty, & \text{if } (\mathbf{q}, \boldsymbol{\mu}) \in (\mathcal{L}^2(\Omega))^2 \times (\mathcal{L}^2(\Omega))^2 \setminus S, \end{cases} \quad (11)$$

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

$$\begin{cases} \nabla^2 v_{\mathbf{q}} = \nabla \cdot \mathbf{q} & \text{in } \Omega, \\ \int_{\Omega} v_{\mathbf{q}} dx = \int_{\Omega} f dx. \end{cases} \quad (12)$$

If we assume periodicity of all the functions, the function $v_{\mathbf{q}}$ is also periodic. Without periodicity, we will need to add a boundary condition to guarantee the uniqueness of the above problem (a typical one being $(\nabla v_{\mathbf{q}} - \mathbf{q}) \cdot \mathbf{n} = 0$ on $\partial\Omega$).

3. An Operator-Splitting Method for the Solution of Problem (9)

3.1. Optimality conditions and associated dynamical flow problem

Let us denote by J_1 and J_2 the functionals defined by

$$\begin{cases} J_1(\mathbf{q}, \boldsymbol{\mu}) = \int_{\Omega} (a + b|\nabla \cdot \boldsymbol{\mu}|^2) |\mathbf{q}| dx, \\ J_2(\mathbf{q}) = \frac{1}{2} \int_{\Omega} |v_{\mathbf{q}} - f|^2 dx, \end{cases} \quad (13)$$

and suppose that $(\mathbf{p}, \boldsymbol{\lambda})$ is a minimizer of the functional in (9). We have then $u = v_{\mathbf{p}}$ and the following system of (necessary) optimality conditions holds:

$$\begin{cases} \partial_{\mathbf{q}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + DJ_2(\mathbf{p}) + \partial I_{\Sigma_f}(\mathbf{p}) + \partial_{\mathbf{q}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0}, \\ D_{\boldsymbol{\mu}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0}, \end{cases} \quad (14)$$

where the Ds (resp., the ∂s) denotes classical differentials (resp., generalized differentials (subdifferentials in the case of non-smooth convex functionals, I_{Σ_f} being a typical one)). We associate with (14) the following initial value problem (dynamical flow):

$$\begin{cases} \frac{\partial \mathbf{p}}{\partial t} + \partial_{\mathbf{q}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + DJ_2(\mathbf{p}) + \partial I_{\Sigma_f}(\mathbf{p}) + \partial_{\mathbf{q}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} & \text{in } \Omega \times (0, +\infty), \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} + D_{\boldsymbol{\mu}} J_1(\mathbf{p}, \boldsymbol{\lambda}) + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} & \text{in } \Omega \times (0, +\infty), \\ (\mathbf{p}(0), \boldsymbol{\lambda}(0)) = (\mathbf{p}_0, \boldsymbol{\lambda}_0), \end{cases} \quad (15)$$

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

The rich structure of problems (14), (15) suggests to solve (14) via the computation of the steady state solutions of (15) using a time discretization method based on operator-splitting. This approach will be discussed in Section 3.2.

Remark 3.1: We advocate taking $(\mathbf{p}_0, \boldsymbol{\lambda}_0) \in S$ in (15).

3.2. An operator-splitting method for the solution of the dynamical flow problem (15)

Following [51] (see also [52] for applications of operator-splitting methods to Imaging) we will use a *Lie scheme* to time-discretize problem (15). 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 (15), we advocate the one below:

$$(\mathbf{p}^0, \boldsymbol{\lambda}^0) = (\mathbf{p}_0, \boldsymbol{\lambda}_0). \quad (16)$$

1st Fractional step: Solve

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \frac{\partial \mathbf{p}}{\partial t} + \partial_{\mathbf{q}} J_1(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} + D_{\boldsymbol{\mu}} J_1(\mathbf{p}, \boldsymbol{\lambda}) = \mathbf{0} \end{array} \right. \quad \text{in } \Omega \times (t^n, t^{n+1}), \\ (\mathbf{p}(t^n), \boldsymbol{\lambda}(t^n)) = (\mathbf{p}^n, \boldsymbol{\lambda}^n), \end{array} \right. \quad (17)$$

and set

$$(\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}) = (\mathbf{p}(t^{n+1}), \boldsymbol{\lambda}(t^{n+1})). \quad (18)$$

2nd Fractional step: Solve

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \frac{\partial \mathbf{p}}{\partial t} + \partial_{\mathbf{q}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}, \boldsymbol{\lambda}) \ni \mathbf{0} \end{array} \right. \quad \text{in } \Omega \times (t^n, t^{n+1}), \\ (\mathbf{p}(t^n), \boldsymbol{\lambda}(t^n)) = (\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}), \end{array} \right. \quad (19)$$

and set

$$(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}) = (\mathbf{p}(t^{n+1}), \boldsymbol{\lambda}(t^{n+1})). \quad (20)$$

3rd Fractional step: Solve

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \frac{\partial \mathbf{p}}{\partial t} + D J_2(\mathbf{p}) + \partial I_{\Sigma_f}(\mathbf{p}) \ni \mathbf{0} \\ \gamma \frac{\partial \boldsymbol{\lambda}}{\partial t} = \mathbf{0} \end{array} \right. \quad \text{in } \Omega \times (t^n, t^{n+1}), \\ (\mathbf{p}(t^n), \boldsymbol{\lambda}(t^n)) = (\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}), \end{array} \right. \quad (21)$$

and set

$$(\mathbf{p}^{n+1}, \boldsymbol{\lambda}^{n+1}) = (\mathbf{p}(t^{n+1}), \boldsymbol{\lambda}^{n+2/3}). \quad (22)$$

The Lie scheme (17)-(22) is only semi-discrete since we have not specified yet how to time-discretize the initial value problems (17), (19) and (21). In order to so, we suggest using the following time discretization scheme (of the Marchuk-Yanenko type):

$$(\mathbf{p}^0, \boldsymbol{\lambda}^0) = (\mathbf{p}_0, \boldsymbol{\lambda}_0). \quad (23)$$

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:

$$\begin{cases} \frac{\mathbf{p}^{n+1/3} - \mathbf{p}^n}{\tau} + \partial_{\mathbf{q}} J_1(\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^n) \ni \mathbf{0} \\ \gamma \frac{\boldsymbol{\lambda}^{n+1/3} - \boldsymbol{\lambda}^n}{\tau} + D_{\boldsymbol{\mu}} J_1(\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}) = \mathbf{0} \end{cases} \quad \text{in } \Omega \Rightarrow (\mathbf{p}^{n+1/3}, \boldsymbol{\lambda}^{n+1/3}), \quad (24)$$

$$\begin{cases} \frac{\mathbf{p}^{n+2/3} - \mathbf{p}^{n+1/3}}{\tau} + \partial_{\mathbf{q}} I_S(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}) \ni \mathbf{0} \\ \gamma \frac{\boldsymbol{\lambda}^{n+2/3} - \boldsymbol{\lambda}^{n+1/3}}{\tau} + \partial_{\boldsymbol{\mu}} I_S(\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}) \ni \mathbf{0} \end{cases} \quad \text{in } \Omega \Rightarrow (\mathbf{p}^{n+2/3}, \boldsymbol{\lambda}^{n+2/3}), \quad (25)$$

$$\begin{cases} \frac{\mathbf{p}^{n+1} - \mathbf{p}^{n+2/3}}{\tau} + D J_2(\mathbf{p}^{n+1}) + \partial I_{\Sigma_f}(\mathbf{p}^{n+1}) \ni \mathbf{0} \\ \gamma \frac{\boldsymbol{\lambda}^{n+1} - \boldsymbol{\lambda}^{n+2/3}}{\tau} = \mathbf{0} \end{cases} \quad \text{in } \Omega \Rightarrow (\mathbf{p}^{n+1}, \boldsymbol{\lambda}^{n+1}). \quad (26)$$

In the following subsections we are going to discuss the solution of the various sub-problems encountered when applying scheme (23)-(26) to the solution of problem (9).

3.3. Computing $\mathbf{p}^{n+1/3}$ from (24)

The multi-valued equation verified by $\mathbf{p}^{n+1/3}$ in (24) is nothing but the (formal) Euler-Lagrange equation of the following minimization problem

$$\mathbf{p}^{n+1/3} = \arg \min_{\mathbf{q} \in (\mathcal{L}^2(\Omega))^2} \left[\frac{1}{2} \int_{\Omega} |\mathbf{q} - \mathbf{p}^n|^2 dx + \tau \int_{\Omega} (a + b |\nabla \cdot \boldsymbol{\lambda}^n|^2) |\mathbf{q}| dx \right]. \quad (27)$$

Problems such as (27) are very common in Image Processing and Viscoplasticity. The closed form solution of problem (27) is given by (see [53, 54, 55, 56, 1]):

$$\mathbf{p}^{n+1/3} = \max \left\{ 0, 1 - \frac{c}{|\mathbf{p}^n|} \right\} \mathbf{p}^n, \quad (28)$$

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

3.4. Computing $\lambda^{n+1/3}$ from (24)

The equation verified by $\lambda^{n+1/3}$ in (24) is the (formal) Euler-Lagrange equation of the following minimization problem

$$\lambda^{n+1/3} = \arg \min_{\mu \in (\mathcal{L}^2(\Omega))^2} \left[\gamma \int_{\Omega} \frac{|\mu - \lambda^n|^2}{2\tau} dx + J_1(\mu, \mathbf{p}^{n+1/3}) \right], \quad (29)$$

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

From the Euler-Lagrangian equation of (29), we get that the solution $\lambda^{n+1/3}$ is the solution of following linear equation:

$$\gamma \frac{\lambda^{n+1/3} - \lambda^n}{\tau} - \nabla(2b|\mathbf{p}^{n+1/3}| \nabla \cdot \lambda^{n+1/3}) = 0 \text{ in } \Omega. \quad (30)$$

This is a vector equation. We shall use periodic boundary condition for the above equation and also other subproblems coming later. It is common to use this kind of boundary condition for image processing problems. It is easy to justify this approach by assuming that the image data is defined on a 2D torus, for example. Under the periodic boundary condition, (30) can be efficiently and easily solved by the FFT, see [1, §3.2.5] and [56].

3.5. Computing $(\mathbf{p}^{n+2/3}, \lambda^{n+2/3})$ from (25)

3.5.1. Decomposition of problem (25)

One can view system (25) as the Euler-Lagrange equation of the following minimization problem:

$$\min_{(\mathbf{q}, \mu) \in S} \left[\int_{\Omega} |\mathbf{q} - \mathbf{p}^{n+1/3}|^2 dx + \gamma \int_{\Omega} |\mu - \lambda^{n+1/3}|^2 dx \right]. \quad (31)$$

Problem (31) can be solved point-wise, reducing, a.e. on Ω , to the following finite dimensional constrained minimization problem:

$$(\mathbf{p}^{n+2/3}(x), \lambda^{n+2/3}(x)) = \operatorname{argmin}_{(\mathbf{q}, \mu) \in \sigma} j_{n+1/3}(\mathbf{q}, \mu; x), \quad (32)$$

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

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

Let us define σ_0 and σ_1 by

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

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

(i) Solve the following two uncoupled minimization problems

$$(\mathbf{p}_0^{n+2/3}(x), \lambda_0^{n+2/3}(x)) = \operatorname{argmin}_{(\mathbf{q}, \mu) \in \sigma_0} j_{n+1/3}(\mathbf{q}, \mu; x), \quad (33)$$

$$(\mathbf{p}_1^{n+2/3}(x), \lambda_1^{n+2/3}(x)) = \operatorname{argmin}_{(\mathbf{q}, \mu) \in \sigma_1} j_{n+1/3}(\mathbf{q}, \mu; x), \quad (34)$$

(ii) Choose the one that gives the smallest value to $j_{n+1/3}$ as the minimizer of (32), i.e.

$$(\mathbf{p}^{n+2/3}(x), \boldsymbol{\lambda}^{n+2/3}(x)) = \operatorname{argmin} \left[j_{n+1/3}(\mathbf{p}_0^{n+2/3}, \boldsymbol{\lambda}_0^{n+2/3}; x), \right. \\ \left. j_{n+1/3}(\mathbf{p}_1^{n+2/3}, \boldsymbol{\lambda}_1^{n+2/3}; x) \right], \text{ a.e. on } \Omega. \quad (35)$$

In what follows, we first introduce a strategy of adaptively choosing γ in Section 3.5.2. After that, in Section 3.5.3 and Section 3.5.4, we will discuss the minimization of the functional in (32) 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

$$\frac{\partial \boldsymbol{\lambda}}{\partial t} = \lim_{\tau \rightarrow 0} \frac{1}{\tau} \left(\frac{\mathbf{p}(t+\tau)}{|\mathbf{p}(t+\tau)|} - \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|} \right). \quad (36)$$

Due to the following relation:

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

one has

$$\left| \frac{\mathbf{p}(t+\tau)}{|\mathbf{p}(t+\tau)|} - \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|} \right| \leq \frac{\sqrt{2}|\mathbf{p}(t+\tau) - \mathbf{p}(t)|}{\sqrt{|\mathbf{p}(t+\tau)|^2 + |\mathbf{p}(t)|^2}}.$$

Let $\tau \rightarrow 0$, we get from (36) that

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

For small τ , the minimizer of (31) verifies

$$\frac{|\mathbf{p}^{n+2/3} - \mathbf{p}^{n+1/3}|^2}{2\tau} + \gamma \frac{|\boldsymbol{\lambda}^{n+2/3} - \boldsymbol{\lambda}^{n+1/3}|^2}{2\tau} \approx \frac{\tau}{2} \left(\left| \frac{\partial \mathbf{p}}{\partial t}(t^{n+1/3}) \right|^2 + \gamma \left| \frac{\partial \boldsymbol{\lambda}}{\partial t}(t^{n+1/3}) \right|^2 \right). \quad (37)$$

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

$$\gamma = \max(|\mathbf{p}^{n+1/3}|^2, \hat{\alpha}), \quad (38)$$

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

3.5.3. Minimizing the functional in (32) over σ_0

Over σ_0 the minimization problem (33) reduces to

$$\min_{\boldsymbol{\mu} \in \mathbf{R}^2, |\boldsymbol{\mu}| \leq 1} |\boldsymbol{\mu} - \boldsymbol{\lambda}^{n+1/3}(x)|. \quad (39)$$

Clearly, the solution of problem (39) is given by

$$\boldsymbol{\lambda}_0^{n+1/3}(x) = \frac{\boldsymbol{\lambda}^{n+1/3}(x)}{\max(1, |\boldsymbol{\lambda}^{n+1/3}(x)|)}. \quad (40)$$

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

$$\mathbf{p}_0^{n+1/3}(x) = \mathbf{0}. \quad (41)$$

3.5.4. Minimizing the functional in (32) over σ_1

Over σ_1 , the minimization problem (34) reduces to:

$$\inf_{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \neq \mathbf{0}, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| = 1} [|\mathbf{q} - \mathbf{p}^{n+1/3}(x)|^2 + \gamma |\boldsymbol{\mu} - \boldsymbol{\lambda}^{n+1/3}(x)|^2]. \quad (42)$$

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 (42) takes the following simplified formulation

$$\inf_{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \neq \mathbf{0}, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{q} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right]. \quad (43)$$

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

$$\mathbf{q} = \theta \boldsymbol{\mu}, \quad \theta > 0. \quad (44)$$

Relation (44) allows us to replace problem (43) by the following constrained minimization problem in \mathbf{R}^3

$$\inf_{(\theta, \boldsymbol{\mu}) \in \mathbf{R} \times \mathbf{R}^2, \theta > 0, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} \theta^2 - \theta \boldsymbol{\mu} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right]. \quad (45)$$

In order to solve problem (45), we observe that the above problem is equivalent to

$$\inf_{\theta > 0} \min_{\boldsymbol{\mu} \in \mathbf{R}^2, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} \theta^2 - \theta \boldsymbol{\mu} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right]. \quad (46)$$

In order to minimize on a closed set of \mathbf{R}^3 , the problem that we finally consider is the following variant of problem (46)

$$\min_{\theta \geq 0} \min_{\boldsymbol{\mu} \in \mathbf{R}^2, |\boldsymbol{\mu}| = 1} \left[\frac{1}{2} \theta^2 - \theta \boldsymbol{\mu} \cdot \mathbf{x} - \gamma \boldsymbol{\mu} \cdot \mathbf{y} \right]. \quad (47)$$

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 (47) reduces to

$$\min_{\theta \geq 0} \left[\frac{1}{2} \theta^2 - |\theta \mathbf{x} + \gamma \mathbf{y}| \right]. \quad (48)$$

There are many ways to solve problem (48), such as Newton's method, bisection or golden section methods, and a variety of fixed point methods ([57]). 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:

$$\begin{cases} \theta^0 = |\mathbf{x}| \\ \text{For } k \geq 0, \theta^{(k)} \rightarrow \theta^{(k+1)} \text{ as follows} \\ \theta^{(k+1)} = \max(0, \frac{\mathbf{x} \cdot (\theta^{(k)} \mathbf{x} + \gamma \mathbf{y})}{|\theta^{(k)} \mathbf{x} + \gamma \mathbf{y}|}). \end{cases} \quad (49)$$

A more detailed presentation of our fixed point method reads as:

Algorithm 1: Fixed point solution of problem (48)

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) $k = k + 1$

Endwhile.

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 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:

$$\boldsymbol{\lambda}_1^{n+2/3}(x) = \frac{\theta^* \mathbf{p}^{n+1/3}(x) + \gamma \boldsymbol{\lambda}^{n+1/3}(x)}{|\theta^* \mathbf{p}^{n+1/3}(x) + \gamma \boldsymbol{\lambda}^{n+1/3}(x)|}, \quad (50)$$

and

$$\mathbf{p}_1^{n+2/3}(x) = \theta^* \boldsymbol{\lambda}_1^{n+2/3}(x), \quad (51)$$

respectively. A more rigorous notation would have been to use $\theta_n^*(x)$ instead of θ^* , since the solution of problem (48) varies with x and n (we recall that, in (48), $\mathbf{x} = \mathbf{p}^{n+1/3}(x)$ and $\mathbf{y} = \boldsymbol{\lambda}^{n+1/3}(x)$).

Once we have gotten $(\mathbf{p}_1^{n+2/3}(x), \boldsymbol{\lambda}_1^{n+2/3}(x))$ from (40)–(41) and $(\mathbf{p}_1^{n+2/3}(x), \boldsymbol{\lambda}_1^{n+2/3}(x))$ from (50)–(51), we obtain the minimizer of (32) through (35).

3.6. Computing \mathbf{p}^{n+1} and λ^{n+1} from (26)

We clearly have

$$\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^{n+2/3}. \quad (52)$$

On the other hand, the multi-valued equation verified by \mathbf{p}^{n+1} in (26) is the Euler-Lagrange equation of the following minimization problem

$$\mathbf{p}^{n+1} = \arg \min_{\mathbf{q} \in \Sigma_f} \left[\frac{1}{2} \int_{\Omega} |\mathbf{q} - \mathbf{p}^{n+2/3}|^2 dx + \frac{\tau}{2} \int_{\Omega} |v_{\mathbf{q}} - f|^2 dx \right], \quad (53)$$

the function $v_{\mathbf{q}}$ being defined by (12).

As we have mentioned in Section 3.4, we shall use periodic boundary condition for all the subproblems. Suppose that Ω is the rectangle $(0, L) \times (0, H)$. Next, define $\mathcal{H}_p^1(\Omega)$, a space of doubly periodic functions, by

$$\mathcal{H}_p^1(\Omega) = \{v \in \mathcal{H}^1(\Omega); v(0, x_2) = v(L, x_2), \text{ a.e. on } (0, H); v(x_1, 0) = v(x_1, H) \text{ a.e. on } (0, L)\}$$

From the definition of Σ_f (see Section 2), problem (53) is equivalent to

$$\begin{cases} \mathbf{p}^{n+1} = \nabla u^{n+1} & \text{with} \\ u^{n+1} = \arg \min_{v \in \mathcal{H}_p^1(\Omega)} \left[\frac{1}{2} \int_{\Omega} |\nabla v|^2 dx + \frac{\tau}{2} \int_{\Omega} |v - f|^2 dx - \int_{\Omega} \mathbf{p}^{n+2/3} \cdot \nabla v dx \right]. \end{cases} \quad (54)$$

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

$$\begin{cases} u^{n+1} \in \mathcal{H}_p^1(\Omega), \\ \int_{\Omega} \nabla u^{n+1} \cdot \nabla v dx + \tau \int_{\Omega} u^{n+1} v dx = \int_{\Omega} \mathbf{p}^{n+2/3} \cdot \nabla v dx + \tau \int_{\Omega} f v dx, \quad \forall v \in \mathcal{H}^1(\Omega) \end{cases} \quad (55)$$

Problem (54), (55) has a unique solution which is the weak solution of the following problem:

$$\begin{cases} -\nabla^2 u^{n+1} + \tau u^{n+1} = -\nabla \cdot \mathbf{p}^{n+2/3} + \tau f, & \text{in } \Omega \\ u^{n+1}(0, x_2) = u^{n+1}(L, x_2) \text{ a.e. on } (0, H); u^{n+1}(x_1, 0) = u^{n+1}(x_1, H) \text{ a.e. on } (0, L), \\ \frac{\partial u^{n+1}}{\partial x_1}(0, x_2) = \frac{\partial u^{n+1}}{\partial x_1}(L, x_2) \text{ a.e. on } (0, H); \frac{\partial u^{n+1}}{\partial x_2}(x_1, 0) = \frac{\partial u^{n+1}}{\partial x_2}(x_1, H) \text{ a.e. on } (0, L). \end{cases} \quad (56)$$

In the particular case of rectangular domains Ω , many efficient solvers are available for the solution of the finite dimensional analogues of problem (56) obtained by finite difference discretization. Among these fast solvers let us mention those based on cyclic reduction and FFT.

3.7. Summary

The subproblems (24), (25) and (26) 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 (4)

Input: The inputted 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 (24) 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 (25).
3. Use the method discussed in Section 3.6 to obtain $(u^{n+1}, \mathbf{p}^{n+1}, \boldsymbol{\lambda}^{n+1})$ from (26).
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 L_2 norm. All 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. 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 will be provided in Section 6 about the choices of all these parameters.

4. Numerical Discretization

4.1. Synopsis

As in [1], we will assume that Ω is a rectangle. We assume 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 Fig. 1. In Fig. 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 sub-problems will be discussed in Sections 4.3–4.6.

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 $h = L/M_1 = H/N_1$. 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

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

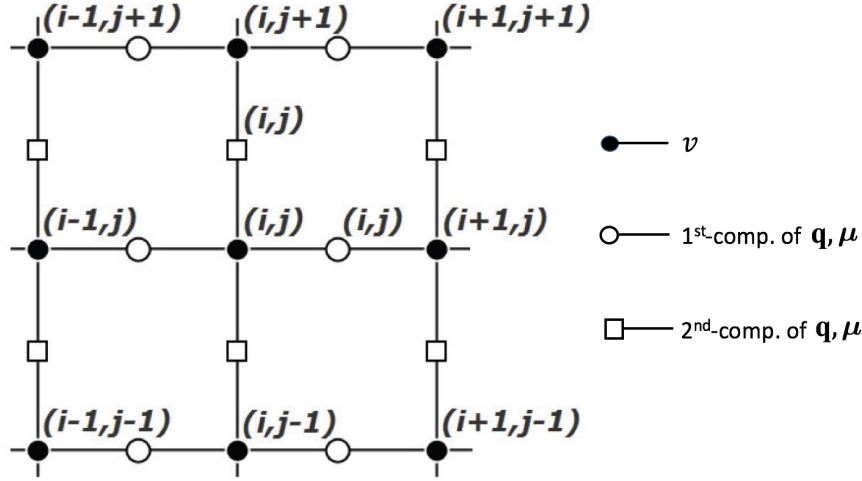


Figure 1: Indexation of the discrete analogues of the unknown functions v (at the \bullet -nodes) and of the first (at the \circ -nodes) and second (at the \square -nodes) components of the vector-valued functions \mathbf{q} and $\boldsymbol{\mu}$.

$$\begin{aligned}\partial_2^- v(i, j) &= \begin{cases} (v(i, j) - v(i, j-1))/h, & 1 < j \leq N_1 \\ (v(i, 1) - v(i, N_1))/h, & j = 1, \end{cases} \\ \partial_1^+ v(i, j) &= \begin{cases} (v(i+1, j) - v(i, j))/h, & 1 \leq i < M_1 \\ (v(1, j) - v(M_1, j))/h, & i = M_1, \end{cases} \\ \partial_2^+ v(i, j) &= \begin{cases} (v(i, j+1) - v(i, j))/h, & 1 \leq j < N_1 \\ (v(i, 1) - v(i, N_1))/h, & j = N_1. \end{cases}\end{aligned}$$

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

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

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) , it will be done using the following averaging operator:

$$\mathcal{A}_{i,j}^\square(\mu_1) = \frac{\mu_1(i, j+1) + \mu_1(i-1, j+1) + \mu_1(i, j) + \mu_1(i-1, j)}{4}, \quad (57)$$

(resp.

$$\mathcal{A}_{i,j}^\circ(\mu_2) = \frac{\mu_2(i+1, j) + \mu_2(i, j) + \mu_2(i+1, j-1) + \mu_2(i, j-1)}{4}, \quad (58)$$

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

$$|\mathcal{A}|_{i,j}^{\bullet}(\mathbf{q}) = \sqrt{\left(\frac{q_1(i,j) + q_1(i-1,j)}{2}\right)^2 + \left(\frac{q_2(i,j) + q_2(i,j-1)}{2}\right)^2}, \quad (59)$$

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

$$\text{div}_{i,j}^{\bullet}(\boldsymbol{\mu}) = [\mu_1(i, j) - \mu_1(i-1, j) + \mu_2(i, j) - \mu_2(i, j-1)]/h, \quad (60)$$

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

$$\mathcal{S}_1^{\pm}\varphi(i, j) = \varphi(i \pm 1, j), \quad \mathcal{S}_2^{\pm}\varphi(i, j) = \varphi(i, j \pm 1) \quad \text{and} \quad \mathcal{I}\varphi(i, j) = \varphi(i, j). \quad (61)$$

4.3. Computation of the discrete analogue of $\mathbf{p}^{n+1/3}$ in (28)

Let us recall that from (28) one has

$$\mathbf{p}^{n+1/3} = \max\left\{0, 1 - \frac{c}{|\mathbf{p}^n|}\right\}\mathbf{p}^n, \quad (62)$$

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 (62) (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:

$$p_1^{(1)}(i, j) = p_1(i, j); \quad p_2^{(1)}(i, j) = \mathcal{A}_{i,j}^{\circ}(p_2), \quad (63)$$

$$\begin{aligned} c^{(1)}(i, j) &= \tau [a + b |\partial_1 \lambda_1(i, j) + \partial_2 \lambda_2(i, j)|^2] \\ &= \tau \left[a + b \left| \frac{\lambda_1(i+1, j) - \lambda_1(i-1, j)}{2h} + \frac{\lambda_2(i+1, j) + \lambda_2(i, j)}{2h} - \frac{\lambda_2(i, j-1) + \lambda_2(i+1, j-1)}{2h} \right|^2 \right]. \end{aligned} \quad (64)$$

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

$$p_1^{(2)}(i, j) = \mathcal{A}_{i,j}^{\square}(p_1); \quad p_2^{(2)}(i, j) = p_2(i, j), \quad (65)$$

$$\begin{aligned} c^{(2)}(i, j) &= \tau [a + b |\partial_1 \lambda_1(i, j) + \partial_2 \lambda_2(i, j)|^2] \\ &= \tau \left[a + b \left| \frac{\lambda_1(i, j) + \lambda_1(i, j+1)}{2h} - \frac{\lambda_1(i-1, j) + \lambda_2(i-1, j+1)}{2h} + \frac{\lambda_2(i, j+1) - \lambda_2(i, j-1)}{2h} \right|^2 \right]. \end{aligned} \quad (66)$$

Finally,

$$p_{\alpha}^{n+1/3}(i, j) = \max\left\{0, 1 - \frac{c^{(\alpha)}(i, j)}{\sqrt{|p_1^{(\alpha)}(i, j)|^2 + |p_2^{(\alpha)}(i, j)|^2}}\right\} p_{\alpha}^{(\alpha)}(i, j), \quad \alpha = \{1, 2\}. \quad (67)$$

4.4. Computation of the discrete analogue of $\lambda^{n+1/3}$ in (30)

We recall that (30) reads as

$$\gamma \lambda^{n+1/3} - \tau \nabla (2b|\mathbf{p}^{n+1/3}| \nabla \cdot \lambda^{n+1/3}) = \gamma \lambda^n, \text{ in } \Omega, \quad (68)$$

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 [1], we discretize (68) as follows

$$\gamma \lambda - \tau \nabla^+ (2b|\tilde{\mathbf{p}}| \text{div}^- \lambda) = \gamma \tilde{\lambda}. \quad (69)$$

To solve (69), we will employ (as in [1]) a *frozen coefficient* approach where instead of solving (69) we solve

$$\gamma \lambda - c^* \nabla^+ (\text{div}^- \lambda) = \gamma \tilde{\lambda} - \nabla^+ \left[(c^* - 2\tau b|\tilde{\mathbf{p}}|) \text{div}^- \tilde{\lambda} \right], \quad (70)$$

with c^* properly chosen. Following [1], we advocate taking $c^* = \max_{\bullet\text{-nodes}(i,j)} 2\tau b|\mathcal{A}|_{i,j}^\bullet(\tilde{\mathbf{p}})$.

In matrix form, (70) can be written as, in Ω_h ,

$$\gamma h^2 \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} - c^* \begin{pmatrix} \partial_1^+ \\ \partial_2^+ \end{pmatrix} \begin{pmatrix} \partial_1^- & \partial_2^- \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \gamma h^2 \begin{pmatrix} \tilde{\lambda}_1 \\ \tilde{\lambda}_2 \end{pmatrix} - \begin{pmatrix} \partial_1^+ \\ \partial_2^+ \end{pmatrix} (c^* h - 2\tau b h |\tilde{\mathbf{p}}|) \text{div}^- \tilde{\lambda}, \quad (71)$$

or, equivalently,

$$\begin{cases} (\gamma h^2 - c^* \partial_1^+ \partial_1^-) \lambda_1 - c^* \partial_1^+ \partial_2^- \lambda_2 = \gamma h^2 \tilde{\lambda}_1 - \partial_1^+ (c^* h - 2\tau b h |\tilde{\mathbf{p}}|) \text{div}^- \tilde{\lambda}, \\ -c^* \partial_2^+ \partial_1^- \lambda_1 + (\gamma h^2 - c^* \partial_2^+ \partial_2^-) \lambda_2 = \gamma h^2 \tilde{\lambda}_2 - \partial_2^+ (c^* h - 2\tau b h |\tilde{\mathbf{p}}|) \text{div}^- \tilde{\lambda}. \end{cases} \quad (72)$$

Using the shifting and identity operator defined in Section 4.2, for each pair (i, j) the first equation in (72) reads as

$$[\gamma h^2 \mathcal{I} + c^* (\mathcal{I} - \mathcal{S}_1^+) (\mathcal{I} - \mathcal{S}_1^-)] \lambda_1(i, j) + c^* (\mathcal{I} - \mathcal{S}_1^+) (\mathcal{I} - \mathcal{S}_2^-) \lambda_2(i, j) = g_1(i, j), \quad (73)$$

where

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

Similarly, the second equation of (72) reads as

$$c^* (\mathcal{I} - \mathcal{S}_2^+) (\mathcal{I} - \mathcal{S}_1^-) \lambda_1(i, j) + [\gamma h^2 \mathcal{I} + c^* (\mathcal{I} - \mathcal{S}_2^+) (\mathcal{I} - \mathcal{S}_2^-)] \lambda_2(i, j) = g_2(i, j), \quad (74)$$

where

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

The boundary conditions we consider being the periodic ones, we may apply the *discrete Fourier transform* \mathcal{F} to equations (73), (74). We obtain then

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \mathcal{F} \begin{pmatrix} \lambda_1(y_i, y_j) \\ \lambda_2(y_i, y_j) \end{pmatrix} = \mathcal{F} \begin{pmatrix} g_1(y_i, y_j) \\ g_2(y_i, y_j) \end{pmatrix}, \quad (75)$$

where in (75), one has:

$$a_{11} = \gamma h^2 - 2c^* (\cos z_i - 1), \quad a_{12} = c^* (\cos z_i - 1 + \sqrt{-1} \sin z_i) (\cos z_j - 1 - \sqrt{-1} \sin z_j),$$

$$a_{21} = c^*(\cos z_j - 1 + \sqrt{-1} \sin z_j)(\cos z_i - 1 - \sqrt{-1} \sin z_i), \quad a_{22} = \gamma h^2 - 2c^*(\cos z_j - 1),$$

with

$$z_i = \frac{2\pi}{M_1} y_i, \quad y_i = 1, 2, \dots, M_1, \quad \text{and} \quad z_j = \frac{2\pi}{N_1} y_j, \quad y_j = 1, 2, \dots, N_1. \quad (76)$$

The determinant $D(i, j)$ of the coefficient matrix in (75) 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 follows then from (75), that (with obvious notation) the solution $\boldsymbol{\lambda}$ of problem (70) (that is the discrete analogue of $\boldsymbol{\lambda}^{n+1/3}$ in (30)) is given by

$$\begin{cases} \lambda_1 = \text{Real} \left[\mathcal{F}^{-1} \left(\frac{a_{22}\mathcal{F}(g_1) - a_{12}\mathcal{F}(g_2)}{D} \right) \right], \\ \lambda_2 = \text{Real} \left[\mathcal{F}^{-1} \left(\frac{-a_{21}\mathcal{F}(g_1) + a_{11}\mathcal{F}(g_2)}{D} \right) \right], \end{cases} \quad (77)$$

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

4.5. Computation of the discrete analogue of $(\mathbf{p}^{n+2/3}, \lambda^{n+2/3})$ in (25)

We need to solve problem (31) to get the solutions. In the following, we give the details of its discretization.

4.5.1. Solution of (40)

From Section 3.5.3, we see that the minimizer of the functional in (32) over σ_0 is given by:

$$\left(\mathbf{p}_0^{n+2/3}(x), \boldsymbol{\lambda}_0^{n+2/3}(x) \right) = \left(\mathbf{0}, \frac{\boldsymbol{\lambda}^{n+1/3}(x)}{\max[1, |\boldsymbol{\lambda}^{n+1/3}(x)|]} \right). \quad (78)$$

The discrete analogue of (78) reads as

$$\left(\mathbf{p}_0^{n+2/3}(i, j), \boldsymbol{\lambda}_0^{n+2/3}(i, j) \right) = \left(\mathbf{0}, \frac{\boldsymbol{\lambda}^{n+1/3}(i, j)}{\max \left[1, \sqrt{|\lambda_1^{n+1/3}(i, j)|^2 + |\lambda_2^{n+1/3}(i, j)|^2} \right]} \right), \quad (79)$$

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 (42)

Section 3.5.4 was dedicated to the solution of problem (42), a constrained minimization problem in \mathbf{R}^4 defined by

$$\inf_{(\mathbf{q}, \boldsymbol{\mu}) \in \mathbf{R}^2 \times \mathbf{R}^2, \mathbf{q} \neq \mathbf{0}, \mathbf{q} \cdot \boldsymbol{\mu} = |\mathbf{q}|, |\boldsymbol{\mu}| = 1} [|\mathbf{q} - \mathbf{p}^{n+1/3}(x)|^2 + \gamma |\boldsymbol{\mu} - \boldsymbol{\lambda}^{n+1/3}(x)|^2]. \quad (80)$$

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

$$\begin{aligned} \mathbf{x}_{i,j} &= \left(\mathbf{x}_{i,j}^{(1)}, \mathbf{x}_{i,j}^{(2)} \right) = \left(p_1^{n+1/3}(i, j), p_2^{n+1/3}(i, j) \right), \\ \mathbf{y}_{i,j} &= \left(\mathbf{y}_{i,j}^{(1)}, \mathbf{y}_{i,j}^{(2)} \right) = \left(\lambda_1^{n+1/3}(i, j), \lambda_2^{n+1/3}(i, j) \right). \end{aligned}$$

Then, we use the following discrete variant of algorithm (49) to compute $\theta_{i,j}^*$

$$\begin{cases} \theta_{i,j}^{(0)} = |\mathbf{x}_{i,j}| \\ \text{For } k \geq 0, \theta_{i,j}^{(k)} \rightarrow \theta_{i,j}^{(k+1)} \text{ as follows} \\ \theta_{i,j}^{(k+1)} = \max \left[0, \frac{\mathbf{x}_{i,j} \cdot (\theta_{i,j}^{(k)} \mathbf{x}_{i,j} + \gamma \mathbf{y}_{i,j})}{|\theta_{i,j}^{(k)} \mathbf{x}_{i,j} + \gamma \mathbf{y}_{i,j}|} \right]. \end{cases} \quad (81)$$

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 (50), (51):

$$\begin{cases} \boldsymbol{\lambda}^{n+2/3}(i, j) = \frac{\theta_{i,j}^* \mathbf{x}_{i,j} + \gamma \mathbf{y}_{i,j}}{\sqrt{|\theta_{i,j}^* x_{i,j}^{(1)} + \gamma y_{i,j}^{(1)}|^2 + |\theta_{i,j}^* x_{i,j}^{(2)} + \gamma y_{i,j}^{(2)}|^2}}, \\ \mathbf{p}^{n+2/3}(i, j) = \theta_{i,j}^* \boldsymbol{\lambda}^{n+2/3}(i, j). \end{cases} \quad (82)$$

4.6. Discretization of problem (56)

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

$$-\nabla^2 u^{n+1} + \tau u^{n+1} = -\nabla \cdot \mathbf{p}^{n+2/3} + \tau f, \quad \text{in } \Omega, \quad (83)$$

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 (83):

$$\begin{pmatrix} \partial_1^- & \partial_2^- \end{pmatrix} \left[\begin{pmatrix} \partial_1^+ \\ \partial_2^+ \end{pmatrix} u^{n+1} - h \begin{pmatrix} \tilde{p}_1 \\ \tilde{p}_2 \end{pmatrix} \right] + \tau h^2 (f - u^{n+1}) = 0, \quad \text{in } \Omega_h \quad (84)$$

Problem (84) is equivalent to

$$(\partial_1^- \partial_1^+ + \partial_2^- \partial_2^+ - \tau h^2) u^{n+1} = h(\partial_1^- \tilde{p}_1 + \partial_2^- \tilde{p}_2) - \tau h^2 f, \quad (85)$$

Relation (85) can be written also as

$$[(\mathcal{I} - \mathcal{S}_1^-)(\mathcal{S}_1^+ - \mathcal{I}) + (\mathcal{I} - \mathcal{S}_2^-)(\mathcal{S}_2^+ - \mathcal{I}) - \tau h^2 \mathcal{I}] u^{n+1}(i, j) = g(i, j), \quad (86)$$

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 (86). We obtain then

$$w_{i,j} \mathcal{F}(u^{n+1}(i, j)) = \mathcal{F}(g(i, j)), \quad (87)$$

where $w(i, j) = \left[(1 - e^{-\sqrt{-1}z_i})(e^{\sqrt{-1}z_i} - 1) + (1 - e^{-\sqrt{-1}z_j})(e^{\sqrt{-1}z_j} - 1) - \tau h^2 \right]$, with z_i and z_j as in (76). From (87), we obtain (with obvious notation)

$$u^{n+1} = \text{Real} \left[\mathcal{F}^{-1} \left(\frac{\mathcal{F}(g)}{w} \right) \right], \quad (88)$$

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

$$\mathbf{p}^{n+1} = \nabla^+ u^{n+1} = \begin{pmatrix} \partial_1^+ u^{n+1} \\ \partial_2^+ u^{n+1} \end{pmatrix}, \quad (89)$$

(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

$$\begin{cases} \lambda_1^{n+1}(i, j) = \lambda_1^{n+2/3}(i, j), \\ \lambda_2^{n+1}(i, j) = \lambda_2^{n+2/3}(i, j). \end{cases} \quad (90)$$

4.7. Further comments

In Sections 4.3 to 4.6, we have supplied the details for the discretization for the sub-problems associated with the operator-splitting scheme (23)-(26). 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. These 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(R) Core(TM) 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 $h = 1$.

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.,

$$\text{ReErr} = \frac{\|u^{n+1} - u^n\|_2}{\|u^{n+1}\|_2} < tol, \quad (91)$$

where tol is a pre-defined positive value. In particular, a bigger tol may result in a faster stopping of the proposed iterative method.

One of the main advantage 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 ([1]). 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, c.f. [1, 58, 2], the Euler elastica model was applied to image denoising. We found, however, that “edge-preserving smoothing” describes better than “denoising” the properties of the proposed method. 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 non-edge regions, while preserving and smoothing the edges. The “denoising” effect is just an intermediate result, “smoothing” being actually 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 Chambolle’s method [34], the THC method [1] and the proposed method to the ROF model which is actually a special case of the Euler elastica energy when setting $b = 0$ in (3). In Fig. 2, we set $b = 0$ and fix $a = 0.1$ for the Euler elastica energy based image restoration problem (3), which is just the ROF model. The results of Chambolle’s method, of the THC method and of the proposed method for the ROF model are shown in Fig. 2. In particular, we implemented our method with $\tau = 0.1$ and $\gamma^n = \max(|\mathbf{p}^{n+1/3}|, \sqrt{\tau})$, c.f. (38). All three algorithms are solving the same ROF based problem and their energy converges to the same value. The restored images are also shown in Fig. 2. We use this example to show that our algorithm works also for the ROF model.

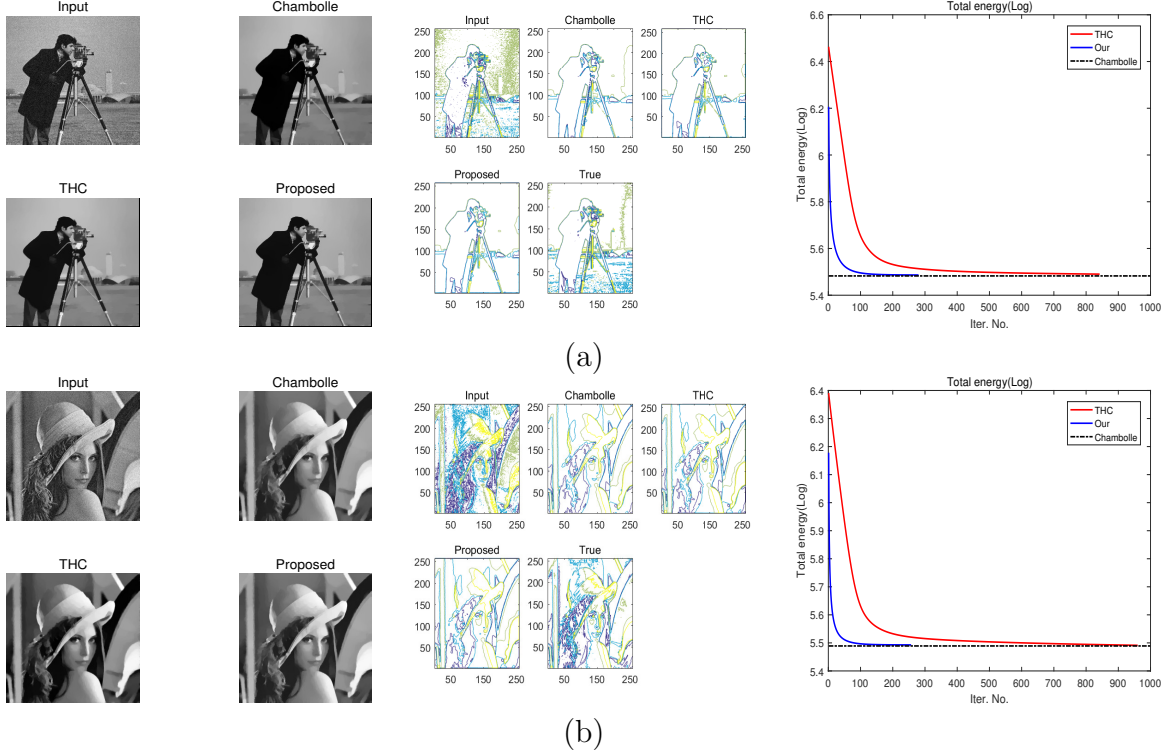


Figure 2: We use Chambolle’s algorithm, the THC method and the proposed method to apply the ROF model when setting $b = 0$ to see if all methods converge to the same solution and same energy. Note that we set $a = 0.1$, and $tol = 1 \times 10^{-5}$ for all methods, and $\tau = 0.1$ for the proposed method. The experiments are tested on the image “cameraman” (a) and “Lena” (b). The results (the first two columns), the contour plots (the middle column), and energy plots (the right column) by the three compared methods are shown. Note that the black dashed lines in the energy plots represent the final energy of the ROF model solved by the Chambolle’s method. From this figure, we observe that the results of three compared approaches for the ROF model converge finally to the same energy value, which is a preliminary validation of the proposed method.

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 Chambolle’s method [34] as well. The results demonstrate the competitive ability of edge-preserving image smoothing of the Euler elastica model.

Fig. 3 shows the results of the proposed algorithm for Euler’s elastica model and Chambolle’s algorithm for ROF model on four synthetic images. The noisy images are shown in the left column, and the smoothed images by 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 [1].

From Fig. 3, the ROF model is able to well preserve image discontinuous jumps, e.g., 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 Fig. 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. An analysis of these properties can be found in [59].

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

In Fig. 4, we also monitor the energy changes of the subproblems and of the original problem (9). From this figure, it is clear that the energies of the $\mathbf{p}^{n+1/3}$ subproblem (27), of the $\boldsymbol{\lambda}^{n+1/3}$ subproblem (29), of $(n + 2/3)$ subproblem (31) (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 (53) 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 (9) 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 [1].

The THC method proposed in [1] is an efficient approach to solve the Euler’s elastica problem. As shown by the results reported in [1], the THC method can solve the Euler elastica problem hundredfold times faster than the Chan-Kang-Shen (CKS) method in [7]. After the THC method, some promising approaches, see e.g., [58, 4, 2, 6], were proposed for the Euler elastica problem. In [58], Duan et al. proposed another fast augmented Lagrangian method to solve the Euler elastica problem based on the framework of the THC method. Afterwards, Duan et al. in [4] applied the THC based method in to solve the Euler elastica regularized Mumford-Shah problem, aiming to deal with two-

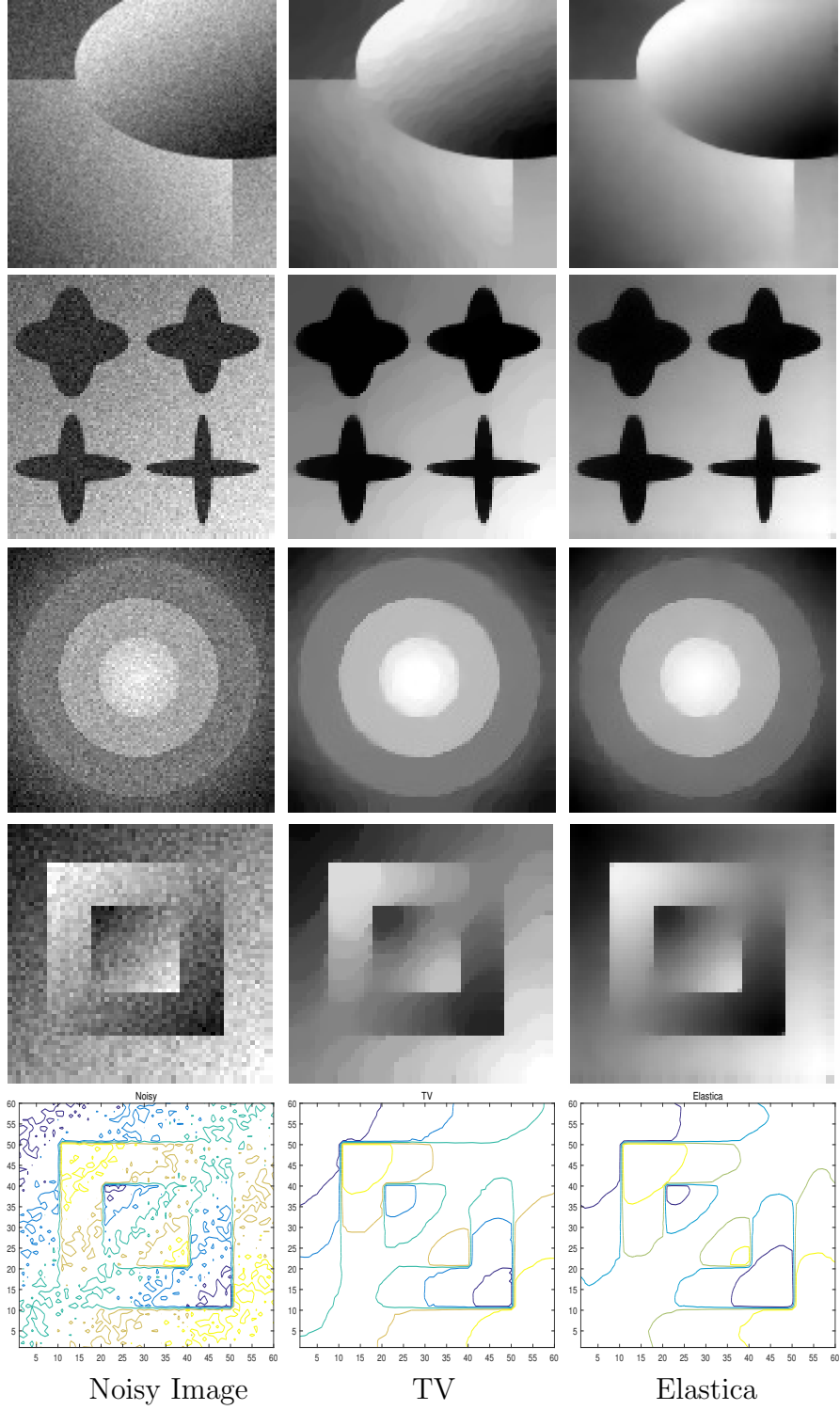


Figure 3: Image smoothing using the ROF and Euler elastica models. Left: Noisy images; Middle: ROF model treated by Chambolle’s method; Right: the Euler elastica model treated by the proposed algorithm. The bottom row shows the contour of the last image. We can see that the elastica model gives images 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 $\gamma^n = \max(|\mathbf{p}^{n+1/3}|, \sqrt{\tau})$.

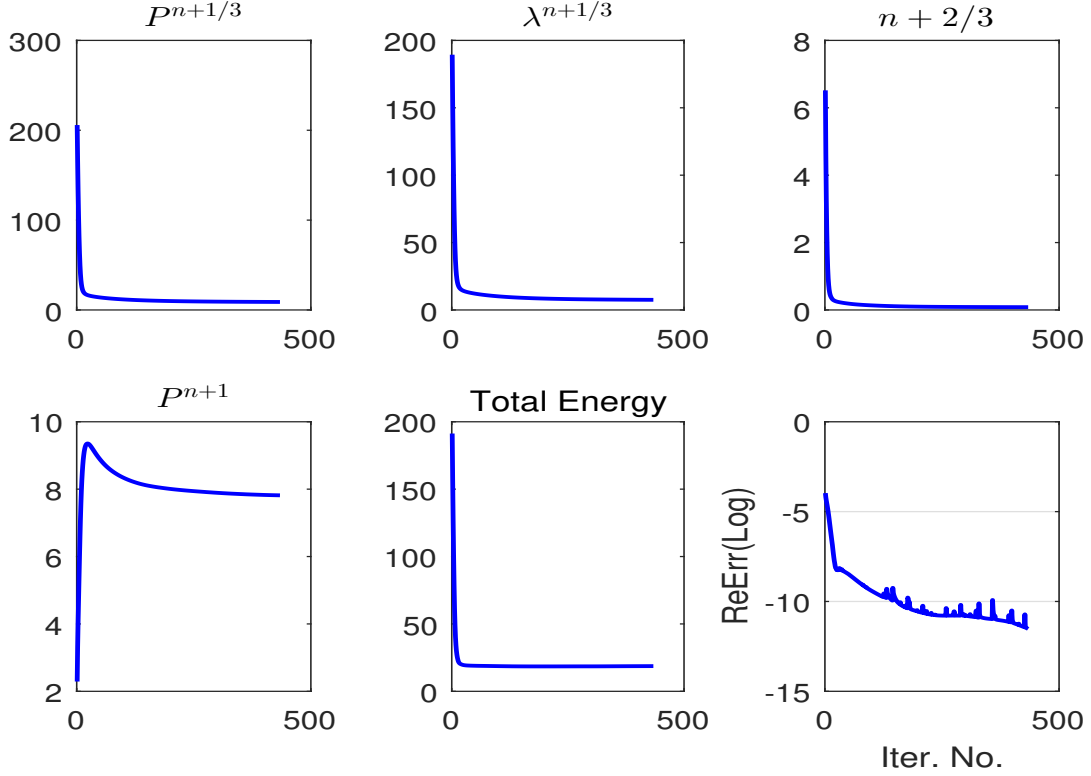


Figure 4: We take the image “square” which is the last one in Fig. 3 as an example to show the energy changes of each subproblem, i.e., the $\mathbf{p}^{n+1/3}$ -subproblem (27), the $\lambda^{n+1/3}$ -subproblem (29), the $(n + 2/3)$ -subproblem (31), the \mathbf{p}^{n+1} -subproblem (53) and the total energy (9).

stage image segmentation. Also, Zhu et al. [3] applied the THC method for the Euler elastica regularized Chan-Vese problem, which gets excellent segmentation results. In [2], 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 only need to verify the second and third advantages.

5.2.1. Parameter sensibility: 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 the time-discretization step τ of our method. Finally,

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

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 Fig. 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 insure 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 (38) for the proposed method), these values producing high quality smoothing of the noisy “ball” image (see [1] 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 results reported in Fig. 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 Fig. 5 and Tab. 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 sub-problems, these sub-problems either having closed form solutions or being solvable by fast algorithms such as FFT.

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 Fig. 6) or modifying the augmentation parameters (see Fig. 7). We see in particular on Fig. 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 Fig. 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.

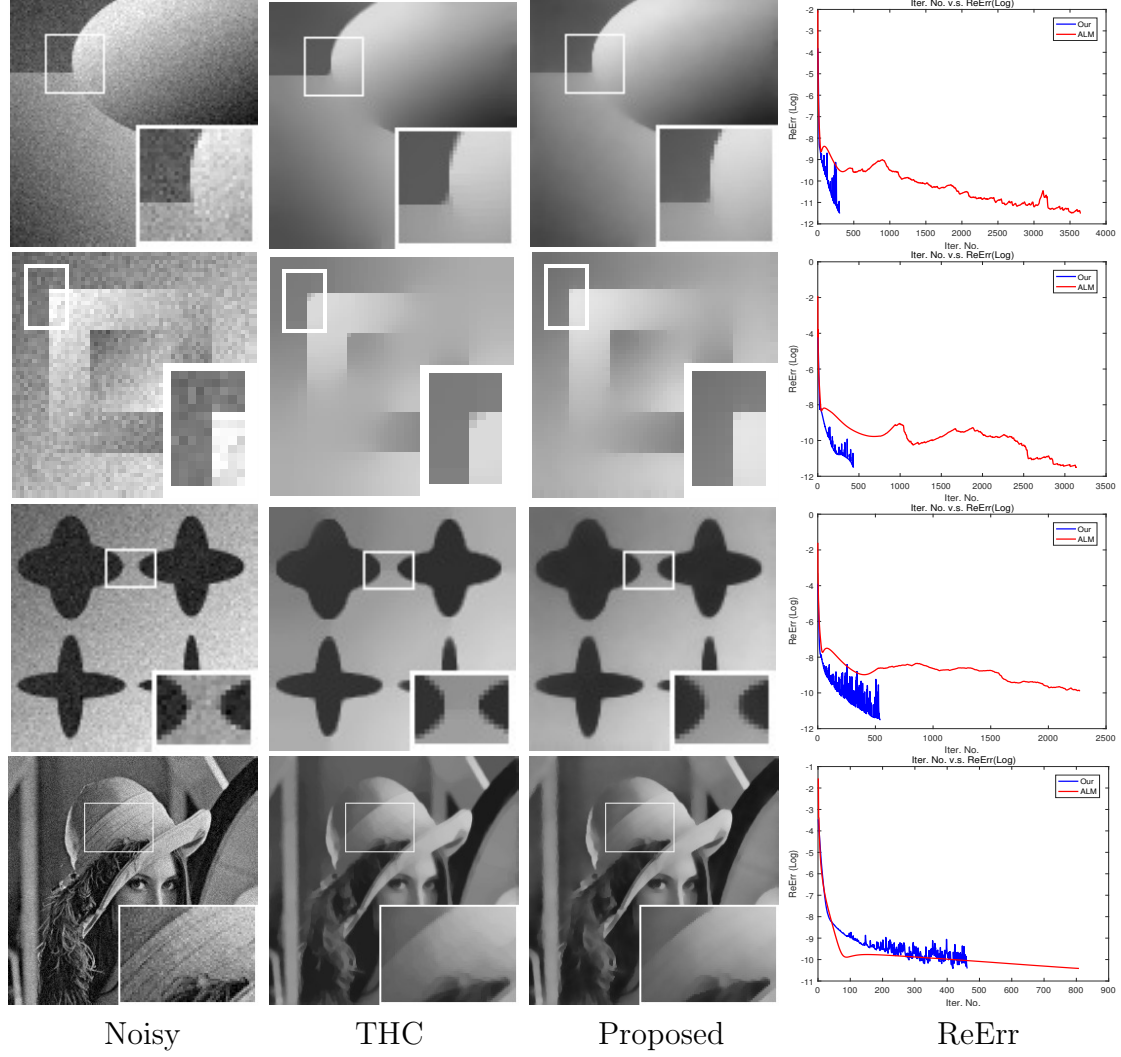


Figure 5: First row: The figure shows that THC method and the method proposed in this article obtain both good results for $r_1 = 0.01$, $r_2 = 10$, $r_3 = 100$ and $\tau = 0.1$, respectively. For both methods we took $a = b = 0.1$ and $tol = 1 \times 10^{-5}$ (resp., 3×10^{-5}) for the first three examples (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).

5.2.2. Speed of convergence comparisons

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

On Figures 10 to 12 we have reported for $tol = 1 \times 10^{-5}$ (Fig. 10), 5×10^{-5} (Fig. 11) and 1×10^{-4} (Fig. 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 Tab. 2, which displays averaged performances), not to mention its greater simplicity and robustness.

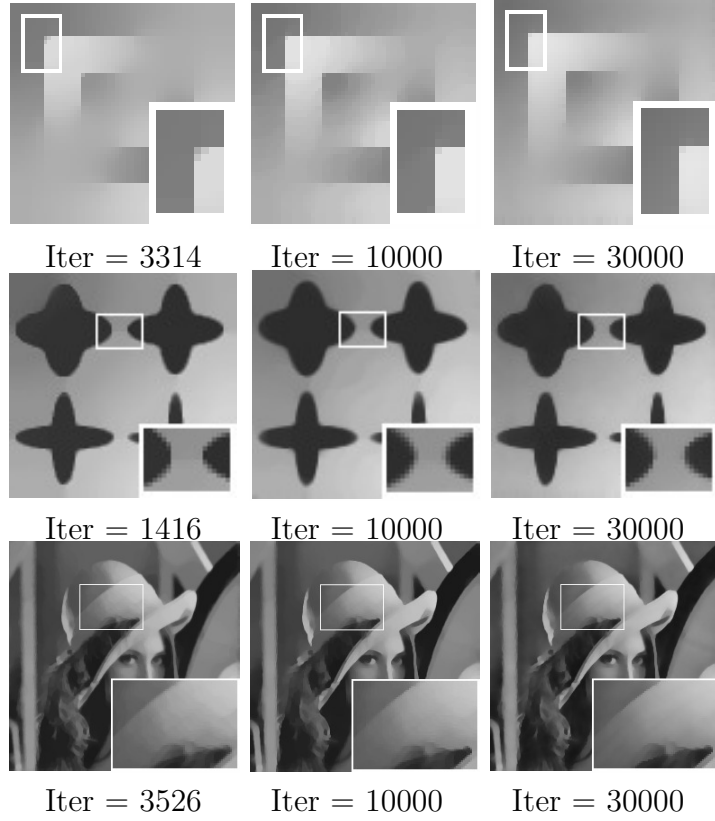


Figure 6: This figure shows that one can improve the smoothing qualities of the THC method (for a non-optimal choice of the augmentation parameters r_1 , r_2 and r_3) by requiring more iterations. We did the computations with $r_1 = 0.01$, $r_2 = 10$ and $r_3 = 100$, an augmentation parameter choice 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 Fig. 5.

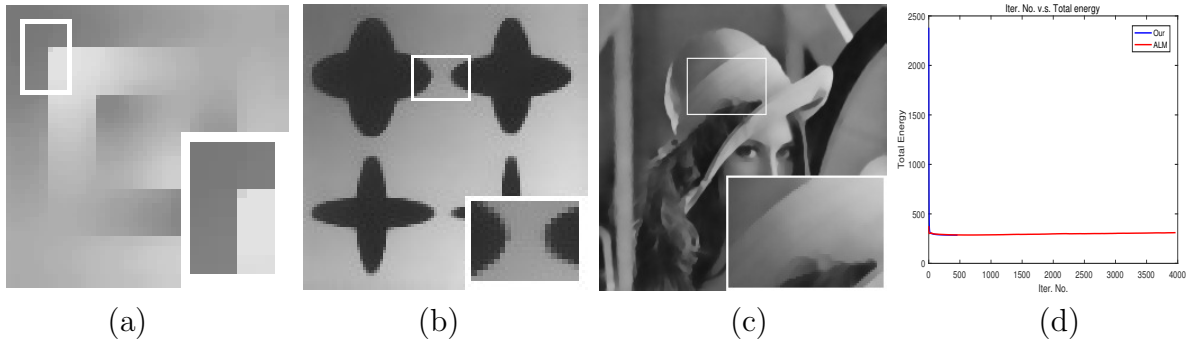


Figure 7: This figure shows that by a proper tuning of the augmentation parameters r_1 , r_2 and r_3 one can significantly improve the smoothing properties of the THC method. Good choices are: (a) $r_1 = 0.05$, $r_2 = 10$ and $r_3 = 100$ for “square”. (b) $r_1 = 0.005$, $r_2 = 10$ and $r_3 = 100$ for “star”. (c) $r_1 = 0.01$, $r_2 = 10$ and $r_3 = 300$ for “Lena”. In Fig. 7(d) we have visualized for the “Lena” image, the variations of the elastica energy v.s. 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 .

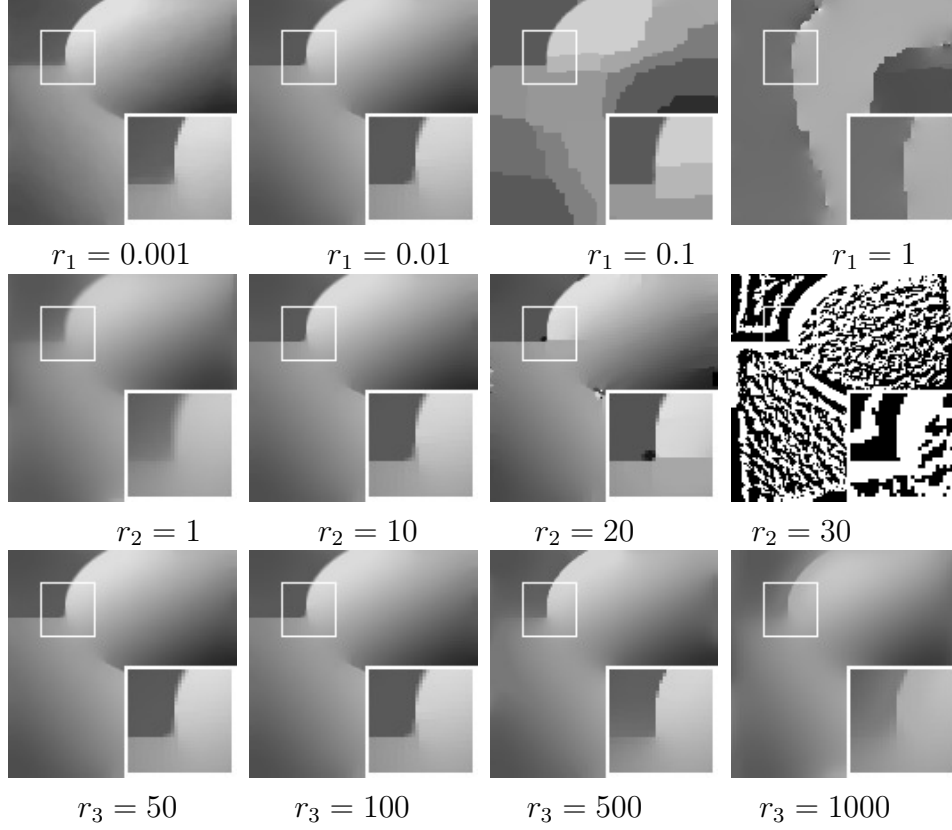


Figure 8: This figure shows the dependence of the THC method computed solution to the augmentation parameters r_1 , r_2 and r_3 . The default choice being $r_1 = 0.01$, $r_2 = 10$ and $r_3 = 100$, one varied each time only one parameter leaving the other two unchanged (here $a = b = 0.1$ and $tol = 1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

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 Fig. 9.

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

6. Conclusions

In the paper, 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 demonstrated also that the proposed method is fast, stable and robust.

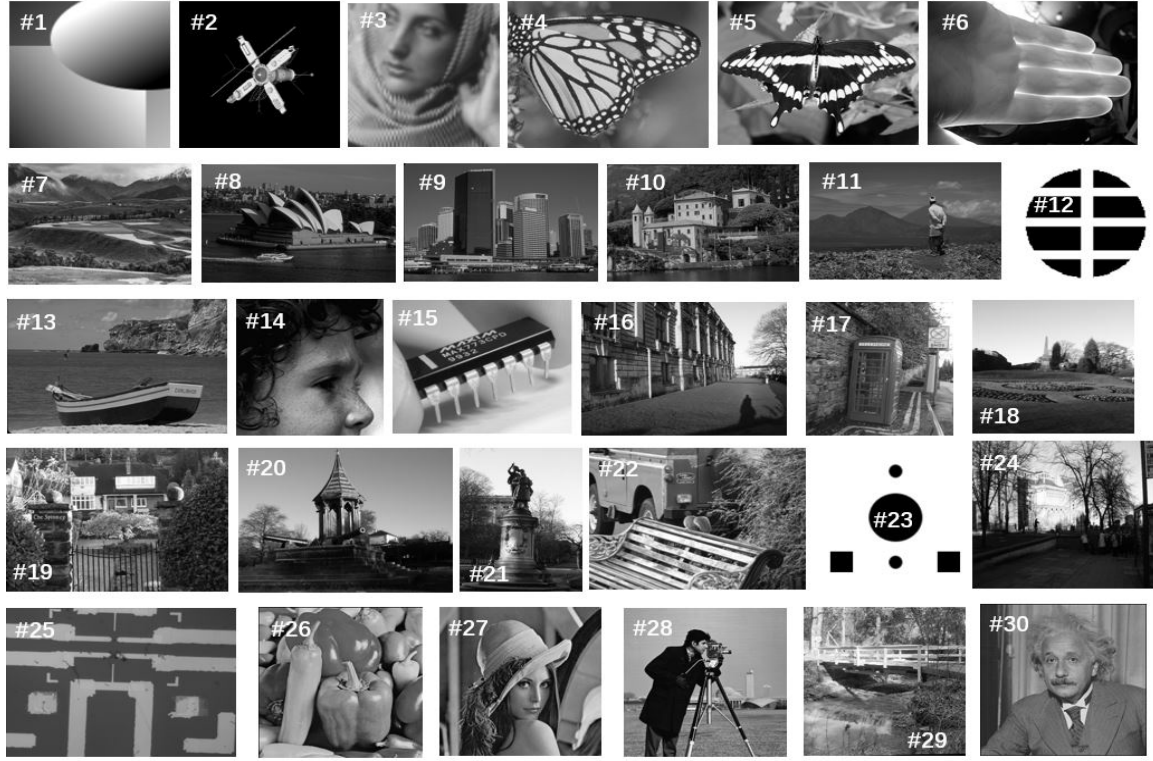


Figure 9: Display of the 30 gray images we used to evaluate and compare the convergence properties of both the method we proposed in this article and the THC method. In order to facilitate the display, we managed to have same size images although they were originally of different sizes.

Acknowledgments

The first author would like to thank the supports by NSFC (61702083, 61772003, 61876203). The third author would like to acknowledge the support from HKBU startup grant, RG(R)- RC/17-18/02-MATH, and FRG2/17-18/033.

References

- [1] 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.*, vol. 4, pp. 313–344, 2011.
- [2] J. Zhang, R. Chen, C. Deng, and S. Wang, “Fast Linearized Augmented Lagrangian Method for Euler’s Elastica Model,” *Numer. Math. Theor. Meth. Appl.*, vol. 10, pp. 98–115, 2017.
- [3] W. Zhu, X. C. Tai, and T. Chan, “Image Segmentation Using Eulers Elastica as the Regularization,” *Journal of Scientific Computing*, vol. 57, pp. 414–438, 2013.
- [4] 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)*, pp. 118–123, 2014.
- [5] J. Zhang and C. Ken, “A new augmented Lagrangian primal dual algorithm for elastica regularization,” *Journal of Alg. and Comput. Tech.*, vol. 10, pp. 325–338, 2016.
- [6] 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. Imag.*, vol. 11, pp. 1–23, 2017.
- [7] T. Chan, S. H. Kang, and J. Shen, “Eulers elastica and curvature-based inpainting,” *SIAM J. Appl. Math.*, vol. 63, pp. 564–592, 2002.
- [8] C. Brito-Loeza and C. Ken, “Fast numerical algorithms for Eulers elastica inpainting model,” *Int. J. Modern Math.*, vol. 5, pp. 157–182, 2010.

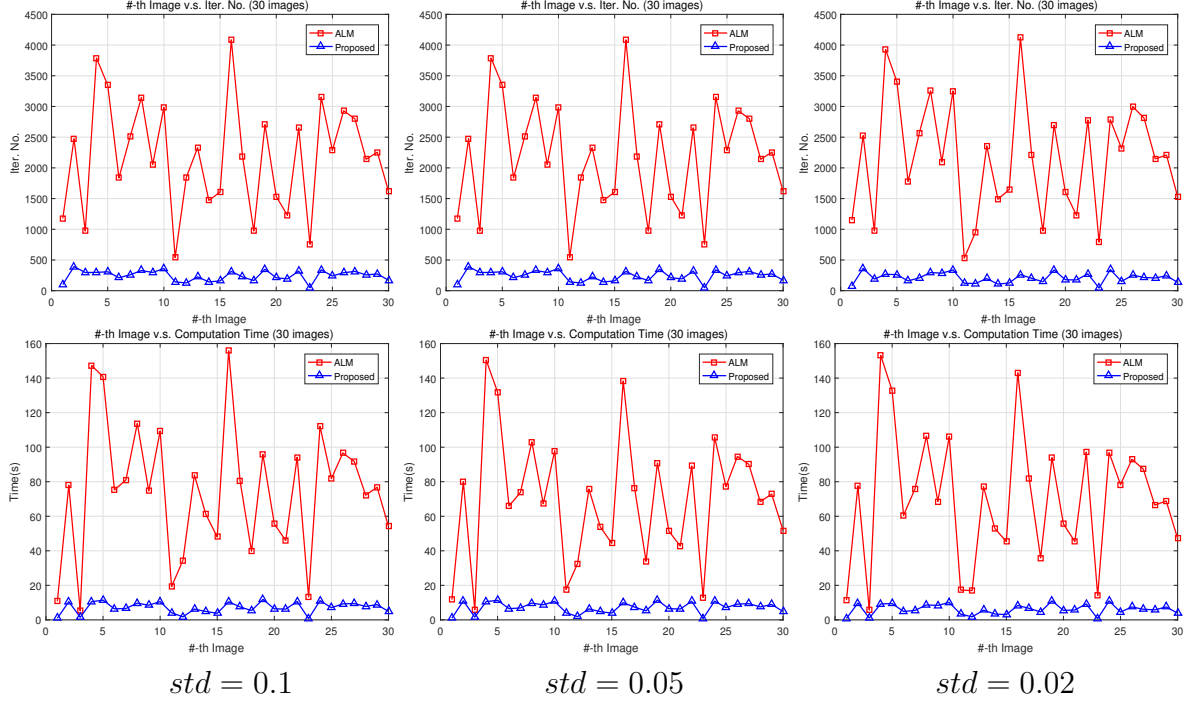


Figure 10: Number of iterations (first row) and corresponding computational time (second row) for the THC method (red curves) and the method introduced in this article (blue curves), both methods using $tol = 1 \times 10^{-5}$ for their respective stopping criterion. The two methods are applied to the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

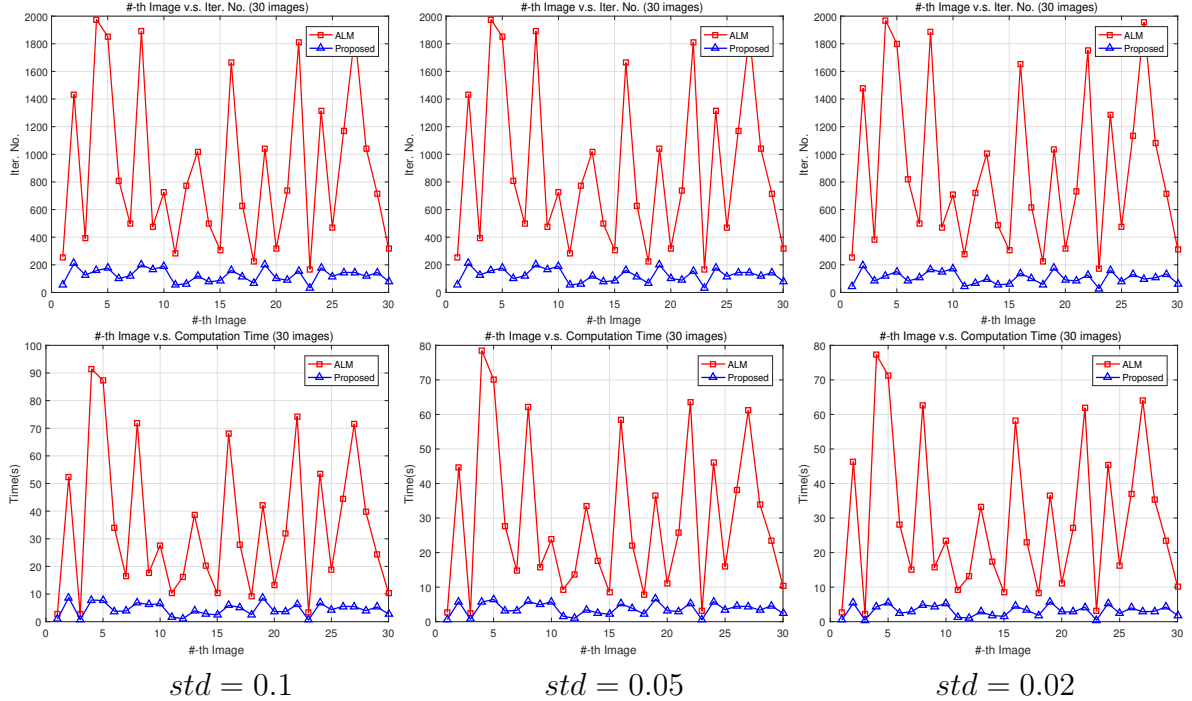


Figure 11: Number of iterations (first row) and corresponding computational time (second row) for the 30 images of Figure 9, using the THC method (red curves) and the method introduced in this article (blue curves). Both methods use $tol = 1 \times 10^{-5}$ for their respective stopping criterion.

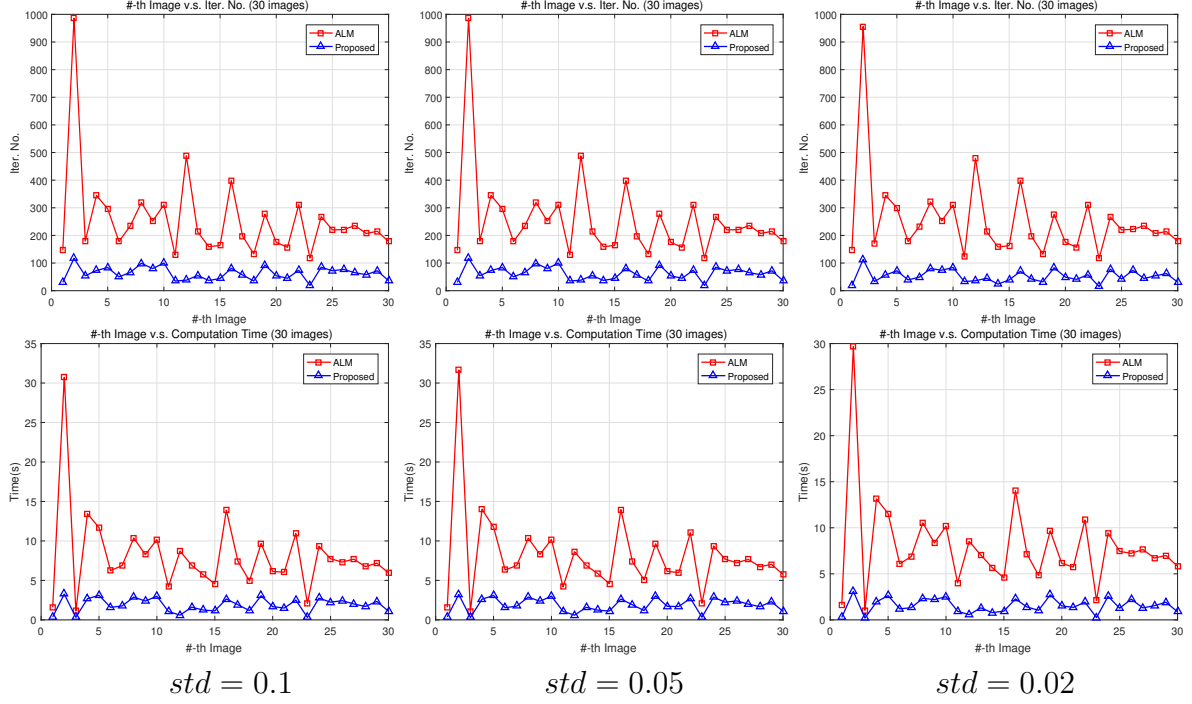


Figure 12: Number of iterations (first row) and corresponding computational time (second row) for the 30 images of Figure 9, using the THC method (red curves) and the method introduced in this article (blue curves). Both methods use $tol = 1 \times 10^{-4}$ for their respective stopping criterion.

- [9] 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.*, vol. 9, pp. 1552–1581, 2016.
- [10] S. Masnou and J.-M. Morel, “On a variational theory of image amodal completion,” *Rend. Sem. Mat. Univ. Padova*, vol. 116, pp. 211–252, 2006.
- [11] S. H. Kang, W. Zhu, and J. Shen, “Illusory shapes via corner fusion,” *SIAM J. Imaging Sci.*, vol. 7, pp. 1907–1936, 2014.
- [12] 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. Mod.*, vol. 14, pp. 809–821, 2017.
- [13] M. Nitzberg, D. Mumford, and T. Shiota, “Filtering, Segmentation and Depth,” *Lecture Notes in Comput. Sci. 662*, Springer-Verlag, 1993.
- [14] S. Esedoglu and R. March, “Segmentation with depth but without detecting junctions,” *J. Math. Imaging Vision*, vol. 18, pp. 7–15, 2003.
- [15] W. Zhu, T. Chan, and S. Esedoglu, “Segmentation with depth: A level set approach,” *SIAM J. Sci. Comput.*, vol. 28, pp. 1957–1973, 2006.
- [16] E. Bae, J. Shi, and X.-C. Tai, “Graph cuts for curvature based image denoising,” *IEEE Trans. Image Process.*, vol. 20, pp. 1199–1210, 2011.
- [17] S. Masnou and J.-M. Morel, “Level lines based disocclusion,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 259–263, 1998.
- [18] K. Bredies, T. Pock, and B. Wirth, “A convex, lower semicontinuous approximation of Euler’s elastica energy,” *SIAM J. Math. Anal.*, vol. 47, pp. 566–613, 2015.
- [19] 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)*, pp. 3257–3264, 2010.
- [20] 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.
- [21] K. Bredies, T. Pock, and B. Wirth, “A Convex, Lower Semicontinuous Approximation of Euler’s Elastica Energy,” *SIAM J. Math. Anal.*, vol. 47, pp. 566–613, 2015.
- [22] E. Dean, R. Glowinski, and G. Guidoboni, “On the numerical simulation of Bingham visco-plastic

- flow: old and new results,” *Journal of non-newtonian fluid mechanics*, vol. 142, pp. 36–62, 2007.
- [23] Y. Duan, Y. Wang, and J. Hahn, “A Fast Augmented Lagrangian Method for Eulers Elastica Models,” *Numerical Mathematics: Theory, Methods and Applications*, vol. 6, pp. 47–71, 2013.
 - [24] M. Yashtini and S. H. Kang, “Alternating Direction Method of Multiplier for Eulers Elastica-Based Denoising,” *In Scale Space and Variational Methods in Computer Vision (SSVM)*, vol. Lecture Notes in Computer Science, vol 9087. Springer, Cham, pp. 690–701, 2015.
 - [25] A. Bruckstein, R. Holt, and A. Netravali, “Discrete elastica,” *Appl. Anal.*, vol. 78, pp. 453–485, 2001.
 - [26] J. Greer and A. Bertozzi, “Traveling wave solutions of fourth order PDEs for image processing,” *SIAM Journal on Mathematical Analysis*, vol. 36, pp. 38–68, 2004.
 - [27] S. Esedoglu, S. Ruuth, and R. Tsai, “Threshold dynamics for shape reconstruction and disocclusion,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 502–505, 2005.
 - [28] 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.*, vol. 10, pp. 1200–1211, 2001.
 - [29] A. Chambolle and T. Pock, “Total Roto-Translational Variation,” *In arXiv:1709.09953*, 2017.
 - [30] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Phys. D*, vol. 60, pp. 259–268, 1992.
 - [31] M. Droske and A. Bertozzi, “Higher-order feature-preserving geometric regularization,” *SIAM Journal on Imaging Sciences*, vol. 3, pp. 21–51, 2010.
 - [32] T. Chan, G. H. Golub, and P. Mulet, “A nonlinear primal-dual method for total variation-based image restoration,” *SIAM J. Sci. Comput.*, vol. 20, pp. 1964–1977, 1999.
 - [33] J. Weickert, B. M. ter Harr Romeny, and M. A. Viergever, “Efficient and reliable schemes for nonlinear diffusion filtering,” *IEEE Trans. Image Process.*, vol. 7, pp. 398–410, 2001.
 - [34] A. Chambolle, “An algorithm for total variational minimization and applications,” *J. Math. Imaging Vision*, vol. 20, pp. 89–97, 2004.
 - [35] 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.
 - [36] 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.*, vol. 1, pp. 143–168, 2008.
 - [37] E. Esser, “Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman,” *UCLA CAM Report 09-31*, UCLA, 2009.
 - [38] 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, 2009.
 - [39] 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, pp. 502–513, 2009.
 - [40] P. Weiss, L. Blanc-Feraud, and G. Aubert, “Efficient schemes for total variation minimization under constraints in image processing,” *SIAM J. Sci. Comput.*, vol. 31, pp. 2047–2080, 2009.
 - [41] G. Steidl and T. Teuber, “Removing multiplicative noise by Douglas-Rachford splitting methods,” *J. Math. Imaging Vision*, vol. 36, pp. 168–184, 2010.
 - [42] 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.*, vol. 3, pp. 300–339, 2010.
 - [43] K. Bredies, K. Kunisch, and T. Pock, “Total generalized variation,” *SIAM J. Imaging Sci.*, vol. 3, pp. 492–526, 2010.
 - [44] X. Zhang, M. Burger, and S. Osher, “A Unified Primal-Dual Algorithm Framework Based on Bregman Iteration,” *J. Sci. Computing*, vol. 46, pp. 20–46, 2011.
 - [45] A. Chambolle, S. Levine, and B. Lucier, “An upwind finite-difference method for total variationbased image smoothing,” *SIAM J. Imaging Sci.*, vol. 4, pp. 277–299, 2011.
 - [46] V. Caselles, A. Chambolle, and M. Novaga, “Total variation in imaging,” *Handbook of Mathematical Methods in Imaging*, pp. 1455–1499, 2015.
 - [47] A. Chambolle, V. Duval, G. Peyre, and C. Poon, “Geometric properties of solutions to the total variation denoising problem,” *Inverse Problems*, vol. 33, pp. 015002, 2016.
 - [48] V. Kolmogorov, T. Pock, and M. Rolinek, “Total variation on a tree,” *SIAM J. Imaging Sci.*, vol. 9, pp. 605–636, 2016.
 - [49] G. Duvaut and J.L. Lions, “Inequalities in mechanics and physics,” *Springer-Verlag*, 1976.
 - [50] R. Glowinski and A. Wachs, “On the numerical simulation of viscoplastic fluid flow,” *Handbook of Numerical Analysis*, vol. 16, pp. 483–717, 2011.

- [51] R. Glowinski, S. Osher, and W. Yin, “Splitting Methods in Communication, Imaging, Science, and Engineering,” *Springer Press*, 2017.
- [52] 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*, pp. 19–94, 2017.
- [53] M Fortin and R. Glowinski, “Augmented Lagrangian methods,” *Volume 15 of Studies in Mathematics and its Applications*, vol. North-Holland Publishing Co., Amsterdam, 1983.
- [54] R. Glowinski and P. Le Tallec, “Augmented Lagrangian and operator-splitting methods in nonlinear mechanics,” *SIAM, Philadelphia*, 1989.
- [55] D. L. Donoho, “De-noising by soft-thresholding,” *IEEE T. Inform. Theory*, vol. 41, pp. 613–627, 1995.
- [56] Y. Wang, J. Yang, W. Yin, and Y. Zhang, “A new alternating minimization algorithm for total variation image reconstruction,” *SIAM J. Imaging Sci.*, vol. 1, pp. 248–272, 2008.
- [57] R. L. Burden and J. D. Faires, “Fixed-Point Iteration,” *Numerical Analysis (3rd ed.)*, PWS Publishers, 1985.
- [58] Y. Duan, Y. Wang, and J. Hahn, “A Fast Augmented Lagrangian Method for Eulers Elastica Models,” *Numer. Math. Theor. Meth. Appl.*, vol. 6, pp. 47–71, 2013.
- [59] W. Zhu and T. Chan, “Image Denoising Using Mean Curvature of Image Surface,” *SIAM J. Imaging Sci.*, vol. 5, pp. 1–32, 2012.