INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 6, Number 3, Pages 355–374

NUMERICAL METHODS FOR NON-SMOOTH L^1 OPTIMIZATION : APPLICATIONS TO FREE SURFACE FLOWS AND IMAGE DENOISING

ALEXANDRE CABOUSSAT, ROLAND GLOWINSKI, AND VICTORIA PONS

Abstract. Non-smooth optimization problems based on L^1 norms are investigated for smoothing of signals with noise or functions with sharp gradients. The use of L^1 norms allows to reduce the blurring introduced by methods based on L^2 norms. Numerical methods based on over-relaxation and augmented Lagrangian algorithms are proposed. Applications to free surface flows and image denoising are presented.

Key Words. L^1 optimization, Over-relaxation algorithm, Augmented Lagrangian methods, Smoothing, Image Denoising.

1. Introduction

The need to smooth a given function is a problem that arises in many fields of science and engineering. A trade-off between the conservation of the accuracy and the regularity properties must be obtained. In volume-of-fluid methods pertaining to computational fluid dynamics, the smoothing of volume fractions of materials is required when calculating interfacial effects [2, 16]. In image treatment, noise can be removed by the application of appropriate filters, based on average mean calculations, low/high-pass filters or PDE-based techniques. Classical smoothing techniques range from kernel-based methods [2], to PDE-based techniques or wavelet-based methods [9]. However when using classical techniques, based on quadratic or L^2 norms, blurring of the sharp edges is often introduced. Recently, methods based on L^1 distances have received a lot more attention in various settings [4, 8, 9, 12, 19, 20]. More generally, smoothing is required when a numerical approximation of the derivatives of a non-smooth function is needed.

In this article, numerical methods for non-smooth optimization problems relying on L^1 norms are presented in order to reduce the blurring due to quadratic terms in classical methods. The solution methods for the smoothing of a given signal require advanced techniques since strict convexity and differentiability properties are not satisfied. Moreover, the uniqueness of the solution is not guaranteed, unless some regularization terms are introduced [15, 21].

The problems addressed here consist of the minimization of the distance between a given signal, typically with jumps or noise, and a smooth approximation whose first derivatives are regular. The L^1 distance is considered first. A smoothing term is introduced to add regularity. The regularization term is given either by the L^2 norm or the L^1 norm of the gradient of the approximated solution. Finally the L^2 distance is considered together with a L^1 smoothing term with bounded variation. Efficient numerical techniques are proposed for the solution of each of

Received by the editors October 30, 2008 and, in revised form, January 5, 2009.
2000 *Mathematics Subject Classification.* 65K10, 65N30, 68U10, 65D10, 93E14 .

these problems. The space discretization is addressed with piecewise linear finite elements. The discretized optimization problems are solved with either an over-relaxation algorithm [17], or an augmented Lagrangian approach [17, 18] when the strict convexity property is not satisfied, or a combination of both.

Numerical results are presented for two kinds of applications. First the smoothing of volume fractions in volume-of-fluid algorithms for multiphase flows is known to introduce artificial numerical errors near the boundaries of the physical domain (spurious currents) [2, 3, 16, 24, 27, 28]. The approximation of the surface tension effects near the boundaries requires for instance the introduction of *ghost cells* outside the domain [13]. This drawback can be corrected by the proposed approach.

On the other hand, image denoising and reconstruction is a very active field of research [6, 8, 10, 25]. The use of L^1 distance has two main properties: it allows to avoid the blurring of edges due to quadratic regularization terms, while being appropriate for removing the noise. Numerical examples based on a famous example (see *e.g.* [10]), are presented to compare the suggested approaches.

2. Non-Smooth Optimization Models

Let Ω be a bounded domain in $\mathbb{R}^2$ with a smooth boundary $\partial\Omega$. Let $f \in L^2(\Omega)$ be a given function (or *signal*), that contains either sharp interfaces, discontinuities along lines or points, or noise. We want to approximate the signal f by a smooth function u (typically $u \in H^1(\Omega)$) in order to (i) be able to approximate the derivatives of f through the derivatives of the function u, or (ii) remove the noise from the original signal.

Let $\Omega \subset \mathbb{R}^2$ be bounded with partition of the boundary $\Gamma_0 \cup \Gamma_1 = \partial\Omega$, $\Gamma_0 \cap \Gamma_1 = \emptyset$. Let us denote by V_0 and W_0 the spaces

$$\begin{aligned} V_0 &= \left\{ v \in H^1(\Omega) \,:\, v|_{\Gamma_0} = 0 \right\}, \\ W_0 &= \left\{ v \in W^{1,1}(\Omega) \,:\, v|_{\Gamma_0} = 0 \right\}. \end{aligned}$$

The Neumann case $\Gamma_0 = \emptyset$ and $\Gamma_1 = \partial\Omega$ is also included. We consider three possible approaches: first the L^1 distance between the original function and its smooth approximation is considered, together with a regularization term depending on the gradient of the approximation. This regularization term can be taken as the L^2 or the L^1 norm of the gradient. The use of the L^1 distance allows to conserve the sharp gradient (edges) of the original function. Finally, we consider the L^2 distance, together with a L^1 smoothing term, and design adequate numerical methods for each of these problems.

2.1. Optimization with L^1 Distance and L^2 Smoothing Term.

For $f \in L^2(\Omega)$, solve

$$\min_{v \in V_0} \int_\Omega |v - f| \, dx + \frac{\varepsilon}{2} \int_\Omega |\nabla v|^2 \, dx. \tag{1}$$

The distance term $\int_\Omega |v - f| \, dx$ is not differentiable, but the addition of the smoothing term $\frac{\varepsilon}{2} \int_\Omega |\nabla v|^2 \, dx$ forces uniqueness through (strict) convexity. The following theorem holds:

Theorem 1. *Problem (1) admits a unique solution $u \in V_0$ (also if $\Gamma_0 = \emptyset$). The solution is characterized by*

$$\varepsilon \int_\Omega \nabla u \cdot \nabla(v - u) dx + \int_\Omega |v - f| \, dx - \int_\Omega |u - f| \, dx \geq 0, \quad \forall v \in V_0. \tag{2}$$

Proof. See [17, 20]. □

Problem (1) is approximated with finite elements and solved with an over-relaxation iterative method in Section 3.1.

2.2. Optimization with L^1 Distance and L^1 Smoothing Term. For $f \in L^2(\Omega)$, solve

$$\inf_{v \in W_0} \int_\Omega |v - f| \, dx + \varepsilon \int_\Omega |\nabla v| \, dx. \tag{3}$$

Problem (3) does not necessarily admits a minimizer, and if this exists, the uniqueness is not guaranteed. Following [9], the formulation (3) is suitable for image denoising and in particular edge detection, to reduce the boundary layers introduced by the smoothing term.

Remark 1. *Since only L^1-norms are involved, problem (3) has to be solved in $W^{1,1}(\Omega)$. Its numerical approximation presented in the sequel is however defined in a $H^1(\Omega)$ framework.*

Remark 2. *In order to ensure uniqueness, a regularization term can be added to (3) as in* [21] *for instance, leading to the following problem:*

$$\min_{v \in V_0} \int_\Omega |v - f| \, dx + \varepsilon \int_\Omega |\nabla v| \, dx + \frac{\alpha}{2} \int_\Omega |\nabla v|^2 \, dx,$$

where α is a given (small) parameter. The corresponding solution has to be in V_0. Numerically, the regularization term is not needed when using the proposed augmented Lagrangian method.

Problem (3) is solved with an augmented Lagrangian approach [17, 18]. Let us define $\mathbf{q} = \nabla v \in L^1(\Omega)^2$. Problem (3) becomes

$$\inf_{(v,\mathbf{q}) \in \mathcal{K}} \int_\Omega |v - f| \, dx + \varepsilon \int_\Omega |\mathbf{q}| \, dx, \tag{4}$$

where $\mathcal{K} = \left\{(v, \mathbf{q}) \in W_0 \times L^1(\Omega)^2 : \nabla v - \mathbf{q} = 0\right\}$. In order to write an augmented Lagrangian, the solution to (4) has to be more regular. From now on, the space $\mathcal{K}$ in (4) is replaced by $\tilde{\mathcal{K}} = \left\{(v, \mathbf{q}) \in V_0 \times L^2(\Omega)^2 : \nabla v - \mathbf{q} = 0\right\}$.

Let $\boldsymbol{\mu} \in L^2(\Omega)^2$ be the Lagrange multiplier corresponding to the constraint $\nabla v - \mathbf{q} = 0$ and $r \geq 0$ a positive penalty constant. The *augmented Lagrangian functional* is defined as

$$\mathcal{L}_r(v, \mathbf{q}; \boldsymbol{\mu}) = \int_\Omega |v - f| \, dx + \varepsilon \int_\Omega |\mathbf{q}| \, dx + \frac{r}{2} \int_\Omega |\nabla v - \mathbf{q}|^2 \, dx + \int_\Omega \boldsymbol{\mu} \cdot (\nabla v - \mathbf{q}) dx. \tag{5}$$

Problem (4) consists in finding the saddle points of (5), namely looking for $\{u, \mathbf{p}, \boldsymbol{\lambda}\} \in V_0 \times L^2(\Omega)^2 \times L^2(\Omega)^2$ such that

$$\mathcal{L}_r(u, \mathbf{p}; \boldsymbol{\mu}) \leq \mathcal{L}_r(u, \mathbf{p}; \boldsymbol{\lambda}) \leq \mathcal{L}_r(v, \mathbf{q}; \boldsymbol{\lambda}), \tag{6}$$

for all $\{v, \mathbf{q}, \boldsymbol{\mu}\} \in V_0 \times L^2(\Omega)^2 \times L^2(\Omega)^2$. The augmented Lagrangian functional has the same saddle-points as the classical Lagrangian functional (*i.e.* when $r = 0$) [17]. An Uzawa-Douglas-Rachford algorithm is chosen to solve (6), which consists of the following iterative algorithm:
Let $u^{-1} \in V_0$ and $\boldsymbol{\lambda}^0 \in L^2(\Omega)^2$ be given. Then, for $n = 0, 1, 2, \ldots$

- Solve $\mathcal{L}_r(u^{n-1}, \mathbf{p}^n; \boldsymbol{\lambda}^n) \leq \mathcal{L}_r(u^{n-1}, \mathbf{q}; \boldsymbol{\lambda}^n)$ for all $\mathbf{q} \in L^2(\Omega)^2$. This corresponds to

$$\min_{\mathbf{q}\in L^2(\Omega)^2} \varepsilon \int_\Omega |\mathbf{q}|\, dx + \frac{r}{2}\int_\Omega |\mathbf{q}|^2\, dx - \int_\Omega (r\nabla u^{n-1} + \boldsymbol{\lambda}^n)\cdot \mathbf{q} dx \tag{7}$$

- Solve $\mathcal{L}_r(u^n, \mathbf{p}^n; \boldsymbol{\lambda}^n) \leq \mathcal{L}_r(v, \mathbf{p}^n; \boldsymbol{\lambda}^n)$ for all $v \in V_0$. This corresponds to

$$\min_{v\in V_0} \frac{r}{2}\int_\Omega |\nabla v|^2\, dx - \int_\Omega (r\mathbf{p}^n - \boldsymbol{\lambda}^n)\cdot \nabla v dx + \int_\Omega |v-f|\, dx \tag{8}$$

- Update the multiplier $\boldsymbol{\lambda}^n \in L^2(\Omega)^2$ by

$$\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^n + r(\nabla u^n - \mathbf{p}^n) \tag{9}$$

Remark 3. *Problem (8) is equivalent to (1) (except for the addition of a linear term), in which the augmented Lagrangian parameter r plays the role of the smoothing coefficient ε. Solution methods for (1) described in the following section can therefore be used inside the iterative method (7) (8) (9).*

The solution of (7) (8) (9) is addressed for the discretized version in Section 3, when approximations by finite elements are introduced.

2.3. Optimization with L^2 Distance and L^1 Smoothing Term. For $f \in L^2(\Omega)$, find $u \in W_0 \cap L^2(\Omega)$ satisfying

$$\min_{v\in W_0\cap L^2(\Omega)} \int_\Omega |v-f|^2\, dx + \varepsilon\int_\Omega |\nabla v|\, dx. \tag{10}$$

The distance term $\int_\Omega |v-f|^2\, dx$ is convex, while the addition of the smoothing term $\varepsilon\int_\Omega |\nabla v|\, dx$ is used to give more regularity to the derivatives of the smoothed function. Because of the Sobolev injection $W^{1,1}(\Omega) \hookrightarrow L^2(\Omega)$, $W_0 \cap L^2(\Omega) = W_0$.

Problem (10) is treated like (3) with an augmented Lagrangian approach, together with a Uzawa-Douglas-Rachford iterative scheme. As in the previous section, the solution to (10) is assumed to be in V_0 in order to write an augmented Lagrangian function. Let us define $\mathbf{q} = \nabla v \in L^2(\Omega)^2$. Problem (10) becomes

$$\inf_{(v,\mathbf{q})\in\mathcal{K}} \int_\Omega |v-f|^2\, dx + \varepsilon\int_\Omega |\mathbf{q}|\, dx, \tag{11}$$

where $\mathcal{K} = \{(v,\mathbf{q}) \in V_0 \times L^2(\Omega)^2 : \nabla v - \mathbf{q} = 0\}$. Let $\boldsymbol{\mu} \in L^2(\Omega)^2$ be the Lagrange multiplier corresponding to the constraint $\nabla v - \mathbf{q} = 0$ and $r \geq 0$ a positive penalty constant. The *augmented Lagrangian function* is defined as

$$\mathcal{L}_r(v,\mathbf{q};\boldsymbol{\mu}) = \int_\Omega |v-f|^2\, dx + \varepsilon\int_\Omega |\mathbf{q}|\, dx + \frac{r}{2}\int_\Omega |\nabla v - \mathbf{q}|^2\, dx + \int_\Omega \boldsymbol{\mu}\cdot(\nabla v - \mathbf{q}) dx. \tag{12}$$

Problem (11) consists in finding the saddle points of (12), namely looking for $\{u,\mathbf{p},\boldsymbol{\lambda}\} \in V_0 \times L^2(\Omega)^2 \times L^2(\Omega)^2$ such that

$$\mathcal{L}_r(u,\mathbf{p};\boldsymbol{\mu}) \leq \mathcal{L}_r(u,\mathbf{p};\boldsymbol{\lambda}) \leq \mathcal{L}_r(v,\mathbf{q};\boldsymbol{\lambda}), \tag{13}$$

for all $\{v,\mathbf{q},\boldsymbol{\mu}\} \in V_0 \times L^2(\Omega)^2 \times L^2(\Omega)^2$. An Uzawa-Douglas-Rachford algorithm to solve (13) consists in the following iterative algorithm:

let $u^{-1} \in V_0$ and $\boldsymbol{\lambda}^0 \in L^2(\Omega)^2$ be given. Then, for $n = 0, 1, 2, \ldots$

- Solve $\mathcal{L}_r(u^{n-1}, \mathbf{p}^n; \boldsymbol{\lambda}^n) \leq \mathcal{L}_r(u^{n-1}, \mathbf{q}; \boldsymbol{\lambda}^n)$ for all $\mathbf{q} \in L^2(\Omega)^2$. This corresponds to solving

$$\min_{\mathbf{q}\in L^2(\Omega)^2} \varepsilon \int_\Omega |\mathbf{q}|\, dx + \frac{r}{2}\int_\Omega |\mathbf{q}|^2\, dx - \int_\Omega (r\nabla u^{n-1} + \boldsymbol{\lambda}^n)\cdot \mathbf{q} dx \tag{14}$$

  Problem (14) is identical to (7).
- Solve $\mathcal{L}_r(u^n, \mathbf{p}^n; \boldsymbol{\lambda}^n) \leq \mathcal{L}_r(v, \mathbf{p}^n; \boldsymbol{\lambda}^n)$ for all $v \in V_0$. This corresponds to solving

$$\min_{v\in V_0} \frac{r}{2}\int_\Omega |\nabla v|^2\, dx + \int_\Omega |v|^2\, dx - \int_\Omega (r\mathbf{p}^n - \boldsymbol{\lambda}^n)\cdot \nabla v dx - 2\int_\Omega v f dx \tag{15}$$

  The elliptic problem (15) is well-posed, even if $\Gamma_0 = \emptyset$.
- Update the multipliers $\boldsymbol{\lambda}^n \in L^2(\Omega)^2$ as in (9):

$$\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^n + r(\nabla u^n - \mathbf{p}^n) \tag{16}$$

The discrete version of the iterative algorithm (14) (15) (16) is addressed in Section 3.

3. Finite Element Approximation and Numerical Algorithms

Finite element methods are used for the space discretization of (1), (3), and (10). Let $h > 0$ be the space discretization step. A family $\{\Omega_h\}_h$ of polygonal approximations of the domain Ω is introduced such that $\lim_{h\to 0}\Omega_h = \Omega$. Let $\mathcal{T}_h$ be a regular triangulation of Ω_h satisfying the compatibility conditions between triangles. Let us denote by N_e the number of elements of $\mathcal{T}_h$, N_n the number of vertices of $\mathcal{T}_h$ in $\Omega_h\backslash\Gamma_0$, and N the total number of vertices of $\mathcal{T}_h$ in $\overline{\Omega}_h\backslash\Gamma_0$. Let K denote a generic element (triangle) of $\mathcal{T}_h$ and P_j, $j = 1, \ldots, N$ the vertices of the triangulation. Let $\mathbb{P}_1$ be the space of polynomials of degree 1 and $\mathbb{P}_0$ the space of polynomials of degree 0. Define

$$\begin{aligned} V_h &= \{v_h \in C^0(\overline{\Omega}) : v|_K \in \mathbb{P}_1, \forall K \in \mathcal{T}_h\}, \\ V_{0h} &= \left\{q \in L^2(\Omega_h) \,:\, q|_K \in \mathbb{P}_0, \forall K \in \mathcal{T}_h\right\}. \end{aligned}$$

as the space of piecewise linear continuous functions and piecewise constant functions respectively. Let $\{\varphi_i\}_{i=1}^N$ be the finite element basis of V_h.

3.1. Numerical Approximation of the Optimization Problem with L^1 Distance and L^2 Smoothing Term.

Problem (1) is approximated by

$$\min_{v_h\in V_h} \int_{\Omega_h} |v_h - f_h|\, dx + \frac{\varepsilon}{2}\int_{\Omega_h} |\nabla v_h|^2\, dx. \tag{17}$$

Let us decompose the original signal in the finite element basis, such that $f(x) = \sum_{i=1}^N f_i\varphi_i(x)$, and $\mathbf{f} = [f_1, \ldots, f_N]^T \in \mathbb{R}^N$. Similarly $v_h(x) = \sum_{i=1}^N v_i\varphi_i(x)$, $\mathbf{v} = [v_1, \ldots, v_N]^T \in \mathbb{R}^N$, and the approximated problem reads

$$\min_{v_h\in V_h} \int_{\Omega_h} \left|\sum_{i=1}^N v_i\varphi_i - \sum_{i=1}^N f_i\varphi_i\right| dx + \frac{\varepsilon}{2}\int_{\Omega_h}\left(\sum_{i=1}^N v_i\nabla\varphi_i\right)\cdot\left(\sum_{j=1}^N v_j\nabla\varphi_j\right) dx.$$

Using numerical integration (namely trapezoidal formula) for the non-smooth term, the discretized problem reads:

$$\min_{\mathbf{v}\in\mathbb{R}^N} \sum_{i=1}^{N} \frac{|\Omega_i|}{3} |v_i - f_i| + \frac{\varepsilon}{2}\mathbf{v}^T\mathbf{A}\mathbf{v}, \tag{18}$$

where $|\Omega_i|$ is the area of the domain defined as the union of the triangles having P_i as a common vertex, $\mathbf{A}$ is the symmetric positive definite stiffness matrix defined as $\mathbf{A} = (A_{ij})_{i,j=1}^N$ and $A_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j dx$. Following [17], define $J(\mathbf{v}) = J_0(\mathbf{v}) + J_1(\mathbf{v})$, with

$$J_0(\mathbf{v}) = \frac{\varepsilon}{2}\mathbf{v}^T\mathbf{A}\mathbf{v}, \quad J_1(\mathbf{v}) = \sum_{i=1}^{N} \frac{|\Omega_i|}{3} |v_i - f_i|.$$

Problem (18) consists in finding $\mathbf{u} \in \mathbb{R}^N$, such that $J(\mathbf{u}) \le J(\mathbf{v})$, for all $\mathbf{v} \in \mathbb{R}^N$.

Theorem 2. *Problem (18) admits a unique solution $\mathbf{u} \in \mathbb{R}^N$. This solution approximates the solution of (1) and is characterized by*

$$\varepsilon\mathbf{u}^T\mathbf{A}(\mathbf{v}-\mathbf{u}) + J_1(\mathbf{v}) - J_1(\mathbf{u}) \ge 0, \quad \forall \mathbf{v} \in \mathbb{R}^N. \tag{19}$$

Proof. See [17, 20]. □

Following [17], *an over-relaxation algorithm* is proposed to construct an iterative sequence $\left\{\mathbf{u}^k\right\}_{k\ge 0}$ that converges to the solution of (18).

- Initialize $\mathbf{u}^0$ arbitrarily (typically $\mathbf{u}^0 = \mathbf{f}$). Let $\omega_i^n \in (0,2)$ be relaxation parameters, defined iteratively as explained later. Then, for $n = 0, 1, 2, \ldots$, compute the iterate $\mathbf{u}^{n+1} = [u_1^{n+1}, \ldots, u_N^{n+1}]^T \in \mathbb{R}^N$ as follows.
- For $i = 1, \ldots, N$, find $\hat{u}_i^{n+1}$ satisfying

$$J(u_1^{n+1}, \ldots, u_{i-1}^{n+1}, \hat{u}_i^{n+1}, u_{i+1}^n, \ldots, u_N^n) \le J(u_1^{n+1}, \ldots, u_{i-1}^{n+1}, v, u_{i+1}^n, \ldots, u_N^n), \tag{20}$$

  for all $v \in \mathbb{R}$. Problem (20) consist in a scalar optimization with respect to each component of the vector $\mathbf{u}^{n+1}$ successively.
- Set

$$u_i^{n+1} = u_i^n + \omega_i^n(\hat{u}_i^{n+1} - u_i^n) \tag{21}$$

The problem (20) has a unique solution $\hat{u}_i^{n+1} \in \mathbb{R}$ characterized by:

$$\left\{ \begin{array}{l} \dfrac{\partial J_0}{\partial u_i}(u_1^{n+1}, \ldots, u_{i-1}^{n+1}, \hat{u}_i^{n+1}, u_{i+1}^n, \ldots, u_N^n)(w_i - \hat{u}_i^{n+1}) \\ +\dfrac{|\Omega_i|}{3}\left(|w_i - f_i| - \left|\hat{u}_i^{n+1} - f_i\right|\right) \ge 0, \forall w_i \in \mathbb{R}. \end{array} \right. \tag{22}$$

This corresponds to the relation

$$\left\{ \begin{array}{l} \left(\varepsilon \displaystyle\sum_{j=1}^{i-1} A_{ij}u_j^{n+1} + \varepsilon A_{ii}\hat{u}_i^{n+1} + \varepsilon \sum_{j=i+1}^{N} A_{ij}u_j^n\right)(w_i - \hat{u}_i^{n+1}) \\ +\dfrac{|\Omega_i|}{3}\left(|w_i - f_i| - \left|\hat{u}_i^{n+1} - f_i\right|\right) \ge 0, \forall w_i \in \mathbb{R}. \end{array} \right. \tag{23}$$

The diagonal term A_{ii} is strictly positive since $\mathbf{A}$ is positive definite. The explicit solution to (23) is

$$\hat{u}_i^{n+1} = \min\left(\frac{b_i^{n+1} + \frac{|\Omega_i|}{3}}{\varepsilon A_{ii}}, \max\left(f_i, \frac{b_i^{n+1} - \frac{|\Omega_i|}{3}}{\varepsilon A_{ii}}\right)\right). \tag{24}$$

where $b_i^{n+1} := -\varepsilon \sum_{j=1}^{i-1} A_{ij} u_j^{n+1} - \varepsilon \sum_{j=i+1}^{N} A_{ij} u_j^n$.

The parameters ω_i^n in (21) are updated iteratively to improve the convergence speed of the algorithm [17]. Let $\omega_i^0 = \omega_f$, where $\omega_f \in [1, 2)$ is an optimal relaxation parameter for the solution of the linear systems associated to the matrix $\mathbf{A}$. The adaptive strategy to update ω_i^n is the following. If

$$\frac{\hat{u}_i^{n+1} - f_i}{u_i^n - f_i} \le 0, \quad \text{set} \ \ \omega_i^n = 1.$$

This leads to $u_i^{n+1} = \hat{u}_i^{n+1}$. Otherwise if

$$\frac{\hat{u}_i^{n+1} - f_i}{u_i^n - f_i} \ge 1, \quad \text{set} \ \ \omega_i = \omega_f,$$

Otherwise if

$$0 \le \frac{(\hat{u}_i^{n+1} - f_i)}{(u_i^n - f_i)} < 1, \quad \text{set} \ \ \omega_i^n = \min\left(\frac{u_i^n - f_i}{u_i^n - \hat{u}_i^{n+1}}, \omega_f\right).$$

Based on numerical experiments, we set $\omega_f = 1.5$. The over-relaxation algorithm converges to the solution of the variational inequality (2):

Theorem 3. *Consider the algorithm (20) (21) with adaptive choice of the parameter ω_i^n. Then, for all $\mathbf{u}^0 \in \mathbb{R}^N$,*

$$\lim_{n \to +\infty} \mathbf{u}^n = \mathbf{u}, \tag{25}$$

where $\mathbf{u}$ is the solution of (18).

Proof. See [17, 20]. □

3.2. Numerical Approximation of the Optimization Problem with L^1 Distance and L^1 Smoothing Term. The discrete version of the algorithm (7)-(9) consists in looking for approximations $u_h \in V_h$, $\mathbf{q}_h \in (V_{0h})^2$ and $\boldsymbol{\lambda}_h \in (V_{0h})^2$ of $u, \mathbf{q}$ and $\boldsymbol{\lambda}$ respectively. The subscripts h are omitted in the sequel.

Let $u^{-1} \in V_h$ and $\boldsymbol{\lambda}^0 \in (V_{0h})^2$ be arbitrary given functions. Then, for $n = 0, 1, 2, \ldots$ The discretized Uzawa-Douglas-Rachford algorithm to solve (6) reads as follows:

- Set $\mathbf{X}^n := r\nabla u^{n-1} + \boldsymbol{\lambda}^n$, with constant value $\mathbf{X}_i^n$ on each triangle, and solve

$$\min_{\mathbf{q} \in (V_{0h})^2} \varepsilon \int_{\Omega_h} |\mathbf{q}|\, dx + \frac{r}{2} \int_{\Omega_h} |\mathbf{q}|^2\, dx - \int_{\Omega_h} \mathbf{X}^n \cdot \mathbf{q} dx \tag{26}$$

  Locally on each element K, it corresponds to solving

$$\min_{\mathbf{q}_i \in \mathbb{R}^2} \left[\varepsilon |\mathbf{q}_i| + \frac{r}{2} |\mathbf{q}_i|^2 - \mathbf{X}_i^n \cdot \mathbf{q}_i\right], \quad i = 1, \ldots, N_e. \tag{27}$$

The minimum occurs when $\mathbf{q}_i = \alpha_i \mathbf{X}_i^n$, $\alpha_i \in \mathbb{R}$, $\alpha_i \geq 0$, and solving the first order optimality conditions leads to $\alpha_i = \dfrac{1}{r}\left(1 - \dfrac{\varepsilon}{|\mathbf{X}_i^n|}\right)^+$ [14], where

$$\left(1 - \frac{\varepsilon}{|\mathbf{X}_i^n|}\right)^+ = \begin{cases} \left(1 - \dfrac{\varepsilon}{|\mathbf{X}_i^n|}\right), & \text{when } \left(1 - \dfrac{\varepsilon}{|\mathbf{X}_i^n|}\right) \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore:

$$\mathbf{p}_i^n = \alpha_i \mathbf{X}_i^n = \begin{cases} \dfrac{1}{r}\left(1 - \dfrac{\varepsilon}{|\mathbf{X}_i^n|}\right)\mathbf{X}_i^n, & \text{if } |\mathbf{X}_i^n| > \varepsilon, \\ 0, & \text{if } |\mathbf{X}_i^n| \leq \varepsilon. \end{cases}$$

- Set $\mathbf{Y}^n := r\mathbf{p}^n - \boldsymbol{\lambda}^n$, with constant value $\mathbf{Y}_i^n$ on each triangle, and solve

$$\min_{v \in V_h} \frac{r}{2}\int_{\Omega_h} |\nabla v|^2 \, dx - \int_{\Omega_h} \nabla v \cdot \mathbf{Y}^n dx + \int_{\Omega_h} |v - f| \, dx \tag{28}$$

Following the steps of Section 3.1, let us define $f(x) = \sum_{i=1}^N f_i \varphi_i(x)$, $\mathbf{f} = [f_1, \ldots, f_N]^T \in \mathbb{R}^N$, and $v(x) = \sum_{i=1}^N v_i \varphi_i(x)$, $\mathbf{v} = [v_1, \ldots, v_N]^T \in \mathbb{R}^N$. The approximated problem reads

$$\begin{aligned} \min_{v \in V_h} \quad & \frac{r}{2}\int_{\Omega_h} \left(\sum_{i=1}^N v_i \nabla\varphi_i\right) \cdot \left(\sum_{j=1}^N v_j \nabla\varphi_j\right) dx - \sum_{i=1}^N v_i \int_{\Omega_h} \nabla\varphi_i \cdot \mathbf{Y}^n dx. \\ & + \int_{\Omega_h} \left|\sum_{i=1}^N v_i\varphi_i - \sum_{i=1}^N f_i\varphi_i\right| dx \end{aligned}$$

Using numerical integration (namely trapezoidal formula) for the non-smooth term, the discretized problem reads:

$$\min_{\mathbf{v} \in \mathbb{R}^N} \sum_{i=1}^N \frac{|\Omega_i|}{3} |v_i - f_i| + \frac{\varepsilon}{2}\mathbf{v}^T\mathbf{A}\mathbf{v} - \mathbf{v}^T\mathbf{b}, \tag{29}$$

where $\mathbf{b} = [b_i]_{i=1}^N \in \mathbb{R}^N$ is defined by $b_i = \displaystyle\int_{\Omega_h} \nabla\varphi_i \cdot \mathbf{Y}^n dx$. Starting from (29), we define

$$J(\mathbf{v}) := \tilde{J}_0(\mathbf{v}) + J_1(\mathbf{v}),$$

with $\tilde{J}_0(\mathbf{v}) = \frac{\varepsilon}{2}\mathbf{v}^T\mathbf{A}\mathbf{v} - \mathbf{v}^T\mathbf{b}$ and $J_1(\mathbf{v}) = \sum_{i=1}^N \frac{|\Omega_i|}{3}|v_i - f_i|$. An over-relaxation algorithm similar to the one detailed in Section 3.1 is advocated to determine the sequence $\{\mathbf{u}^n\}_{n=0}^\infty \subset \mathbb{R}^N$ such that $\mathbf{u}^n \to \mathbf{u}$ and $J(\mathbf{u}^n) \leq J(\mathbf{v})$, for all $\mathbf{v} \in \mathbb{R}^N$.

- Update the multipliers $\boldsymbol{\lambda}^n \in (V_{0h})^2$ as

$$\boldsymbol{\lambda}^{n+1}\big|_K = \boldsymbol{\lambda}^n\big|_K + r\left(\nabla u^n\big|_K - \mathbf{p}^n\big|_K\right), \quad \forall K \in \mathcal{T}_h. \tag{30}$$

The solution to the original problem (3) is not unique in W_0. However the use of augmented Lagrangian techniques allows to construct a converging sequence in V_0 by solving only well-posed problems (32). The use of relaxation terms as in [21] for instance, is therefore not necessary.

3.3. Numerical Approximation of the Optimization Problem with L^2 Distance and L^1 Smoothing Term. The discrete version of the algorithm (14)-(16) consists in looking for approximations $u_h \in V_h$, $\mathbf{q}_h \in (V_{0h})^2$ and $\boldsymbol{\lambda}_h \in (V_{0h})^2$ of $u, \mathbf{q}$ and $\boldsymbol{\lambda}$ respectively. The subscripts h are omitted in the sequel. Let $u^{-1} \in V_h$ and $\boldsymbol{\lambda}^0 \in (V_{0h})^2$ be arbitrary given functions. The Uzawa-Douglas-Rachford algorithm to solve (6) reads as follows: for $n = 0, 1, 2, \ldots$

- Set $\mathbf{X}^n := r\nabla u^{n-1} + \boldsymbol{\lambda}^n$, with constant value $\mathbf{X}_i^n$ on each triangle, and solve

$$\min_{\mathbf{q}\in(V_{0h})^2} \varepsilon \int_{\Omega_h} |\mathbf{q}|\, dx + \frac{r}{2}\int_{\Omega_h} |\mathbf{q}|^2\, dx - \int_{\Omega_h} \mathbf{X}^n \cdot \mathbf{q} dx \tag{31}$$

  This problem is equivalent to (26).
- Set $\mathbf{Y}^n := r\mathbf{p}^n - \boldsymbol{\lambda}^n$, with constant value $\mathbf{Y}_i^n$ on each triangle, and solve

$$\min_{v\in V_h} \frac{r}{2}\int_{\Omega_h} |\nabla v|^2\, dx + \int_{\Omega_h} |v|^2\, dx - \int_{\Omega_h} \mathbf{Y}^n \cdot \nabla v dx - 2\int_{\Omega_h} v f\, dx \tag{32}$$

  This classical elliptic problem is solved with piecewise linear finite elements. The corresponding linear system reads:

$$(r\mathbf{A} + \mathbf{M})\, \mathbf{v} = \mathbf{b}, \tag{33}$$

  where the rigidity matrix $\mathbf{A}$ is computed exactly, the mass matrix $\mathbf{M}$ is computed with the trapezoidal formula (mass lumping), and $\mathbf{b} = [b_i]_{i=1}^N$ is defined by $b_i = \int_{\Omega_h} \nabla\varphi_i \cdot \mathbf{Y}^n dx - 2\int_{\Omega_h} \varphi_i f dx$. The right-hand side is numerically approximated by $b_i \simeq \sum_{K\in\mathcal{T}_h} \nabla\varphi_i|_K \cdot \mathbf{Y}^n|_K\, |K| - \frac{2}{3}\, |\Omega_i|\, f(x_i)$. The linear systems (33) are solved with a sparse conjugate gradient method.
- Update the multipliers $\boldsymbol{\lambda}^n \in (V_{0h})^2$ as

$$\boldsymbol{\lambda}^{n+1}\big|_K = \boldsymbol{\lambda}^n|_K + r\left(\nabla u^n|_K - \mathbf{p}^n|_K\right), \quad \forall K \in \mathcal{T}_h. \tag{34}$$

4. Applications and Numerical Results

Let us consider $\Omega = (0,1)\times(0,1)$. Figure 1 illustrates the structured triangulation $\mathcal{T}_h$ of $\Omega_h = \Omega$ used for the two applications considered here, namely the smoothing of volume fractions in free surfaces flows, and the denoising of images/signals composed of pixels.

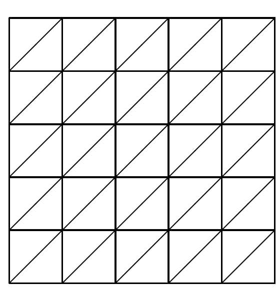

FIGURE 1. Domain $\Omega = (0,1)\times(0,1)$ and the corresponding structured triangulation $\mathcal{T}_h$ (for $N_n = 16 = 4^2$, $N = 36 = 6^2$ and $N_e = 50$).

4.1. Interfaces Reconstruction in Free Surface Flows. In volume-of-fluid methods for multiphase flow, the volume fractions of each material are typically approximated by piecewise constant functions on each cell of a finite volume or finite differences discretization. Consider a two-dimensional finite volume mesh composed by square cells. The triangulation of Figure 1 is embedded into such a finite volume discretization, by splitting each cell into two triangles along its first diagonal. The original piecewise constant approximation of the volume fraction can be projected onto the finite element space of piecewise linear functions by an L^2 projection [5], in order to address smoothing techniques in a finite elements framework.

The smoothing of volume fractions typically introduces a bias near $\partial\Omega$ when using kernel-based methods [2, 3, 16, 24]. The following results show that this drawback is removed with the proposed techniques (via the absence of deformations at the boundary).

Results are illustrated only for the first method (L^1 distance together with L^2 regularization terms); results are similar for the three approaches.

First, a planar interface between two fluids is considered. The original volume fraction is the step function given by zero on one side of the interface line $x = 1/2$, and by one on the other side (characteristic function). In this case, the curvature of the interface is zero and no interfacial force appear on the interface. Figure 2 shows the level lines of the original function f and the smoothed function u for various values of the smoothing parameter ε (dependent on the grid size h) and show that the level lines remain perfectly parallel near the boundaries of Ω. The larger ε, the larger the smoothing of the characteristic function, but no spurious curvature is created.

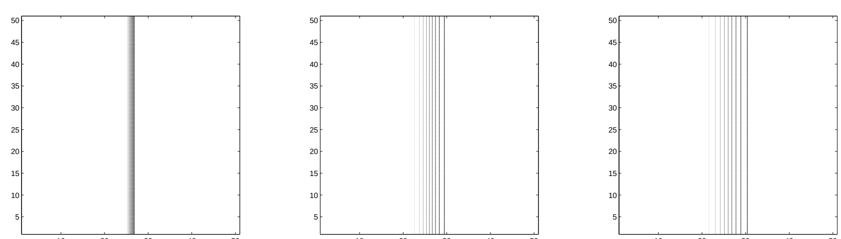

FIGURE 2. Smoothing of volume fractions in the case of planar interface. Left: Original volume fraction (step function); middle: smoothed volume fraction with $\varepsilon = 3h$; right: smoothed volume fraction with $\varepsilon = 5h$.

Consider the case of a sinusoidal interface, defined as $x = \frac{1}{4}\sin(2\pi y) + \frac{1}{2}$. The volume fraction is given by one on one side of the interface and by zero on the other side (characteristic function). Figure 3 shows the contour plot of the original volume fraction, as well as the smoothed function u for various values of ε. One can note still a small distorsion in the neighborhood of the boundary of Ω.

The regular function u can be used to approximate the normal vector to the interface $\mathbf{n} := \frac{\nabla u}{|\nabla u|}$, the curvature of the interface $\kappa := \nabla \cdot \frac{\nabla u}{|\nabla u|}$, and the surface tension effects, as in [2, 3]. Numerical results show good convergence properties of the approximation of κ based on a variational formulation as in [3].

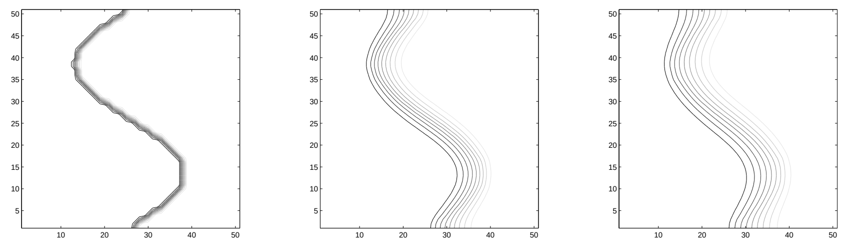

FIGURE 3. Smoothing of volume fractions in the case of sinusoidal interface. Left: Original volume fraction (characteristic function); middle: smoothed volume fraction with $\varepsilon = 3h$; right: smoothed volume fraction with $\varepsilon = 5h$.

4.2. Image Denoising. Images composed of pixels consist of an array of square cells that is similar to a piecewise constant approximation of a given signal (by shades of gray). A L^2-projection of such a signal onto the finite element space V_h related to the triangulation $\mathcal{T}_h$, is achieved as described in Section 4.1. Starting with a noisy signal, composed of an original signal artificially blurred with a given random noise, the goal is to reconstruct the original signal, with emphasis on recovering its sharp edges.

Let $\hat{f} \in L^2(\Omega)$ be the original signal (black/white image). Typically, $\hat{f}$ takes values between 0 and 245 (shades of gray), so $\hat{f}$ is normalized in the sequel to take values between 0 and 1 to be apparented to a volume fraction.

Let $f \in L^2(\Omega)$ be the modification of the original signal by the addition of *noise*. Two types of noise are considered in this article: (i) salt-and-pepper noise (see *e.g.* [1, 7, 11, 26, 29, 30]), and (ii) additive Gaussian noise (see *e.g.* [20, 22, 23]). The *amplitude* of the noise varies in the numerical experiments. The goal of the proposed algorithms is to reconstruct an approximation of the original signal $\hat{f}$ from f.

Salt-and-Pepper Noise. Salt-and-pepper noise represents the addition to f of randomly occuring white and black pixels, (*i.e.* of value 0 or 1). Unaffected pixels always remain unchanged. The noise is quantified by the percentage of pixels which are corrupted. The addition of salt-and-pepper noise can be formulated as follows. Let $\mathcal{U}(x)$ be a uniform random function, piecewise constant on each pixel x_i and taking values between 0 and 1. Let x_i be one pixel of the image (namely one vertex of the underlying mesh). A signal modified by $\alpha\%$ of noise is given by

$$f(x_i) = \begin{cases} 0, & \text{if } \mathcal{U}(x_i) \leq \dfrac{\alpha}{2}, \\ 1, & \text{if } \mathcal{U}(x_i) \geq 1 - \dfrac{\alpha}{2}, \\ \hat{f}(x_i), & \text{otherwise.} \end{cases}$$

The quantity α is the percentage of noise, implying that $\alpha\%$ of the pixels are modified, half of them taking value one (”salt”), the other half taking value zero (”pepper”).

Gaussian noise. Additive Gaussian noise is a modification of the signal $\hat{f}$ based on the probability density function of the normal distribution, with mean μ and variance σ^2, meaning that the values of the additional noise are normally distributed. The addition of Gaussian noise can be formulated as follows. Let $\mathcal{N}_{\mu,\sigma}(x)$ be a

Gaussian distribution function with mean μ and standard deviation σ, piecewise constant on each pixel x_i. A signal modified by $\alpha\%$ of noise is given by

$$f(x_i) = \hat{f}(x_i) + \alpha \mathcal{N}_{0,1}(x_i).$$

This means that, unlike the salt-and-pepper noise, *every* pixel of the original signal is modified by normally distributed values.

Results for the optimization problem with L^1 distance - L^2 smoothing term. Figures 4–6 illustrate the results of the algorithm presented in Section 3.1 for various settings. Figure 4 shows the original signal $\hat{f} \in L^2(\Omega)$, together with the modified signal $f \in L^2(\Omega)$ (modified with the salt-and-pepper noise of intensity 2%), and the reconstructed signal u obtained by the algorithm. For such small intensity of noise, one can see that the original signal $\hat{f}$ is completely reconstructed. Although the solution u is in $H^1(\Omega)$, and therefore smoother than $\hat{f}$, the edges are accurately conserved.

FIGURE 4. Image Denoising for salt-and-pepper noise. $L^1 - L^2$ smoothing algorithm for a noise level of 2% and $\varepsilon = 0.0003$. Left: original signal $\hat{f}$; middle: modified signal f; right: signal u after denoising.

Figure 5 illustrates the capacity of the algorithm in handling larger levels of noise, by showing the modified signals f together with the reconstructed functions u. As expected, the more noise the more difficult the denoising; however, a significant improvement is obtained, even if the algorithm is applied only once (*one-time filter*).

The smoothing parameter ε drives the amount of smoothing/denoising of the algorithm. When the smoothing parameter ε is too large, the resulting signal v is blurred and the edges are not preserved (*smoothed-out*), while the salt-and-pepper noise is completely removed. On the other hand, when the smoothing parameter ε is too small, the small peaks introduced by the noise are still visible, but the edges are preserved.

Figure 6 illustrates the denoising results when considering an additive Gaussian noise for the modification of the signal $\hat{f}$. Since such modification alters each pixel of the image, it is more difficult to recover the original signal. Results show significant denoising of the signal, while conserving edges, and exhibit the robustness of the algorithm.

Results for the optimization problem with L^1 distance - L^1 smoothing term. In a second step, results for the algorithm presented in Section 3.2 are presented. Figures 7–9 illustrate the results in various settings. Figure 7 illustrates the results of the algorithm under various levels of noise, by showing the modified signals f together with the reconstructed functions u. Results compare well with those in Figure 5.

FIGURE 5. Image Denoising for salt-and-pepper noise. $L^1 - L^2$ smoothing algorithm for various levels of noise and $\varepsilon = 0.0003$. Left: 10%, middle: 20%, right: 50%. First line: modified signal f; second line: signal u after denoising.

FIGURE 6. Image Denoising for Gaussian noise. $L^1 - L^2$ smoothing algorithm. Top row: modified signals f with 10% noise (left) and 25% noise (right). Bottom row: reconstructed signals with $\varepsilon = 0.002$.

Figure 8 shows the influence of the number of iterations of the augmented Lagrangian algorithm on the regularity of the solution. Let $\varepsilon = 1$ and $r = 10^{-6}$. The edges of the signal are blurred when the number of iterations increases, confirming than the augmented Lagrangian method acts as a smoother. When the parameter

FIGURE 7. Image Denoising for salt-and-pepper noise. $L^1 - L^1$ smoothing algorithm with $\varepsilon = 1$, $r = 10^{-4}$, one iteration of the augmented Lagrangian algorithm, for various levels of noise: left: 10%, middle: 20%, and right: 50%.

r is very small, the regularization effect induced by r is negligible, but is compensated by the number of iterations of the algorithm. If r is larger, a large number of iterations leads to a complete blurring of the signal.

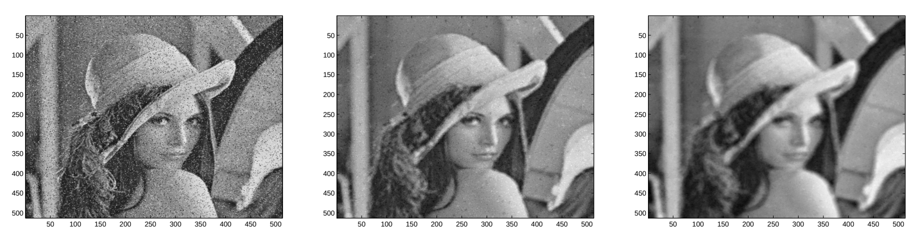

FIGURE 8. Image Denoising for salt-and-pepper noise 10%. $L^1 - L^1$ smoothing algorithm with $\varepsilon = 1$ and $r = 10^{-6}$. Regularized signal after 1 (left), 50 (middle) and 200 (right) iterations of the augmented Lagrangian algorithm.

Figure 9 shows the results of the denoising for additive Gaussian noise with the $L^1 - L^1$ numerical optimization algorithm after one iteration of the augmented Lagrangian. Results compare well with Figure 6. The signal is smoothed, and the noise is removed.

We can conclude that the $L^1 - L^1$ optimization algorithm introduces more denoising than the $L^1 - L^2$ optimization algorithm presented earlier; however this results in a loss of clarity of the signal and a tendency to blur the edges. On the other hand, this algorithm introduces more flexibility since it involves two smoothing parameters: r and ε. Results have shown that increasing the number of iterations of the augmented Lagrangian algorithm over a certain threshold blurs the signal. The

FIGURE 9. Image Denoising for Gaussian noise. $L^1 - L^1$ smoothing algorithm with $\varepsilon = 1$ and $r = 0.0009$. Top row: modified signals $\hat{f}$ with 10% noise (left) and 25% noise (right). Bottom row: reconstructed signals v.

parameter r (and the augmented Lagrangian iteration algorithm) plays the role of the smoother.

Results for the optimization problem with L^2 distance - L^1 smoothing term. In a last step, results for the $L^2 - L^1$ optimization algorithm presented in Section 3.3 are presented. Figure 10 illustrates the results of the algorithm when considering a salt-and-pepper noise of intensity 10%. Like the $L^1 - L^1$ method, the augmented Lagrangian iteration algorithm acts as a smoother. The behavior of the augmented Lagrangian algorithm is very close to the one for the problem with L^1 distance and L^1 smoothing term, but the L^2 distance produces more smoothing, with less iterations.

FIGURE 10. Image Denoising for Gaussian noise. $L^2 - L^1$ smoothing algorithm with $\varepsilon = 0.028$ and $r = 0.0002$. Regularized signal after 2 (left), 4 (middle) and 10 (right) iterations of the augmented Lagrangian.

Table 1. PSNR for the reconstruction of the 512×512 image "Lena" in Figures 4–10, for various levels of salt-and-pepper noise.

$\alpha\%$ (noise level)	PSNR*	PSNR$(L^1 - L^2)$ $(\varepsilon = 3 \cdot 10^{-4})$	PSNR$(L^1 - L^1)$ $(\varepsilon = 1, r = 10^{-4})$ $(1-2$ iterations$)$	PSNR$(L^2 - L^1)$ $(\varepsilon = 0.0028, r = 2 \cdot 10^{-5})$ $(2-4$ iterations$)$
10	23.38 db	29.01 db	29.07 db	26.16 db
20	18.79 db	26.47 db	26.79 db	23.68 db
50	11.71 db	20.31 db	20.43 db	19.45 db

Peak Signal-to-Noise ratios. The performance of algorithms can be quantified by considering the *peak signal-to-noise ratio* (PSNR), which represents the distance between two signals. In the literature [7, 23, 26], the classical distance to consider is the distance between the modified signal f and the reconstructed signal u. However, when the original signal $\hat{f}$ is known, it is more accurate to consider the PSNR comparing the original signal $\hat{f}$ and the reconstructed signal u. Keeping in mind that $\hat{f}$ is normalized in this paper and takes value between 0 and 1, the PSNR for a signal of $N \times M$ pixels is defined by

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{1}{\mathrm{MAE}^2} \right), \quad \mathrm{MAE} = \frac{\displaystyle\sum_{i=1}^{N} \sum_{j=1}^{M} \left| u_{ij} - \hat{f}_{ij} \right|}{MN}.$$

To evaluate the performance of the denoising algorithm, the PSNR comparing the original signal $\hat{f}$ and the noisy signal f is computed as a reference, namely

$$\mathrm{PSNR}^\star = 10 \log_{10} \left(\frac{1}{(\mathrm{MAE}^\star)^2} \right), \quad \mathrm{MAE}^\star = \frac{\displaystyle\sum_{i=1}^{N} \sum_{j=1}^{M} \left| f_{ij} - \hat{f}_{ij} \right|}{MN}.$$

Results for the three methods are presented in Table 1 for the salt-and-pepper noise at various intensities. The larger the PSNR, the better the approximation of the original signal $\hat{f}$. These results confirm the optimal choice of parameters and the performance of the three methods to denoise f. The results of the first two methods compare well, but the one relying on the L^2 distance is less efficient.

4.3. Multiphase Flow with Noise. Consider now the situation of a multiphase flows containing sharp edges and corners. Let $\Omega = (0,1)^2$ and $\hat{f} \in L^2(\Omega)$ be defined as the characteristic function of a V-shape domain, as follows:

$$\hat{f}(x,y) = \begin{cases} 1, & -2x + 1.2 \leq y \leq 0.8, \ 0.2 \leq x \leq 0.3 \\ 1, & -2x + 1.2 \leq y \leq -2x + 1.4, \ 0.3 \leq x \leq 0.5 \\ 1, & 2x - 0.8 \leq y \leq 2x - 0.6, \ 0.5 \leq x \leq 0.7 \\ 1, & 2x - 0.8 \leq y \leq 0.8, \ 0.7 \leq x \leq 0.8 \\ 0, & \text{otherwise}. \end{cases}$$

A salt-and-pepper noise with $\alpha\%$ of noise, as defined in the previous section, is added to the signal $\hat{f}$ to define the original signal f. The goals are now two-fold: first to smoothen the edges representing the interfaces between the various materials, in order for instance to compute curvatures and interfacial effects, and then to denoise the signal f. Consider the triangulation $\mathcal{T}_h$ illustrated in Figure 1 with $2N^2$ triangles and $N = 128$.

The signal f, after modification by a salt-and-pepper noise of intensity 10% or 20% is illustrated in Figure 11.

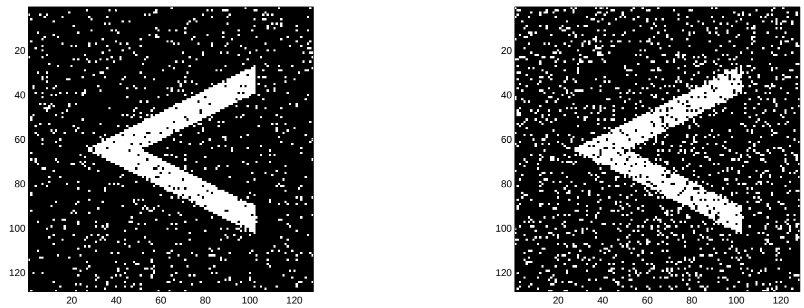

FIGURE 11. Smoothing of sharp interfaces. Signal f after alteration with a salt-and-pepper noise of intensity $\alpha = 10\%$ (left) or $\alpha = 20\%$ (right) ($N = 128$).

Figures 12 and 13 show the results of the smoothing procedures (1), (3), and (10) applied to the function f. In such as multi-objective optimization framework (smoothing edges and denoising), the choice of the smoothing parameter allows to emphasize one goal or the other. Figure 12 illustrates the results obtained with the three algorithms, for a level of noise of $\alpha = 10\%$. Figure 13 illustrates the results obtained with the three algorithms, for a level of noise of $\alpha = 20\%$. Comparisons show that the method with L^2 distance and L^1 smoothing term adds more blurring than the other two approaches.

5. Conclusions

Numerical methods for non-smooth optimization problems based on L^1 norms have been proposed for the smoothing of signals with noise or for the regularization of signals with sharp gradients. Decomposition techniques based on over-relaxation algorithms and augmented Lagrangian techniques allows to efficiently compute minimizing sequences.

Numerical results are presented for applications in free surfaces flows and image denoising. For the smoothing of volume fractions, the drawback encountered through other methods of creating artificial curvature near the boundaries of the physical domain is avoided. Although the first motivation of the work is from mechanical engineering and computational fluid dynamics, the methods have provided efficient results for the treatment of signals with salt-and-pepper noise.

Acknowledgements

The authors thank Dr M. M. Francois and J. M. Sicilian for helpful comments and having initiated the investigation of this problem. This research was partially supported by the Los Alamos National Laboratory, Los Alamos, NM, and by the National Science Foundation Grants NSF ATM-0417867 and NSF DMS-0412267.

References

[1] L. Bar, N.Sochen, and N. Karyati. Image deblurring in the presence of salt-and-pepper noise. *Lecture Notes in Computer Science, Springer Berlin/Heidelberg*, 3459:107–118, 2005.

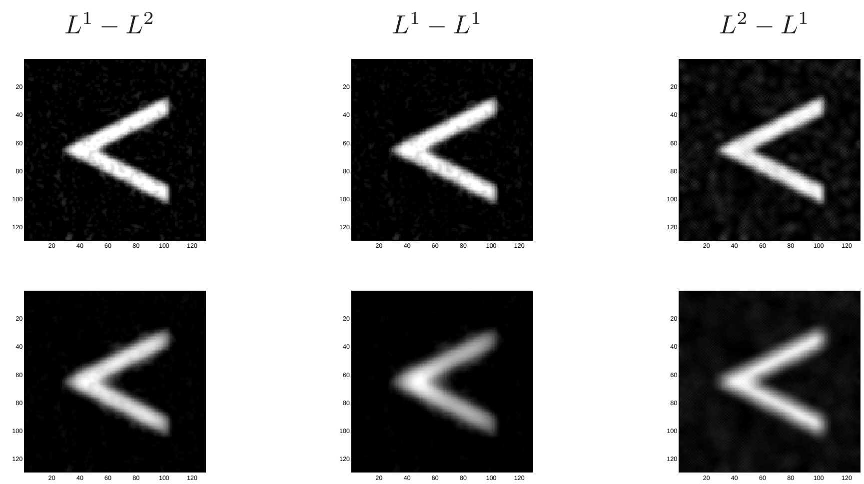

FIGURE 12. Multiphase Flow with Noise: reconstruction and smoothing of sharp interfaces for 10% of salt-and-pepper noise. Left: Smoothed solution with the algorithm with L^1 distance and L^2 smoothing term, with $\varepsilon = 3 \cdot 10^{-3}$ (first row) and $\varepsilon = 10^{-2}$ (second row). Middle: Smoothed solution with the algorithm with L^1 distance and L^1 smoothing term, with $\varepsilon = 1$, $r = 2 \cdot 10^{-3}$ and 1 iteration of the augmented Lagrangian (first row), and $\varepsilon = 1$, $r = 10^{-2}$ and 1 iteration (second row); Right: Smoothed solution with the algorithm with L^2 distance and L^1 smoothing term, with $\varepsilon = 1$, $r = 10^{-4}$ and 4 iterations of the augmented Lagrangian algorithm (first row) and $\varepsilon = 1$ and $r = 10^{-4}$ and 10 iterations (second row).

[2] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comp. Phys.*, 100:335–354, 1992.
[3] A. Caboussat. A numerical method for the simulation of free surface flows with surface tension. *Computers and Fluids*, 35(10):1205–1216, 2006.
[4] A. Caboussat, R. Glowinski, and V. Pons. An augmented Lagrangian approach to the numerical solution of a non-smooth eigenvalue problem. *SIAM J. Sci. Comp*, submitted, 2008.
[5] A. Caboussat, R. Glowinski, and J. M. Sicilian. Computation of the normal vector to a free surface by a finite element - finite volume mixed method. *Comptes Rendus Académie Sciences. Paris, Série I*, 343(6):431–436, 2006.
[6] T. Cecil, S. Osher, and L. Vese. Numerical methods for minimization problems constrained to S^1 and S^2. *J. of Computational Physics*, 198(2):567–579, 2004.
[7] R.H. Chan, C. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detectors detail-preserving regularization. *Image Processing*, 14(10):1479–1485, 2005.
[8] T. Chan and J. Shen. Variational restoration of nonflat image features: models and algorithms. *SIAM J. Appl. Math.*, 61(4):1338–1361, 2000.
[9] T. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia, 2005.
[10] T. F. Chan, J. Shen, and L. Vese. Variational PDE models in image processing. *Notices of the American Mathematical Society*, 50(1):14–26, 2003.
[11] K. Chinnasarn, Y. Rangsanseri, and P. Thitimajshima. Removing salt-and-pepper noise in text/graphics images. In *Circuits and Systems*, pages 459–462, 1998.
[12] B. A. Cipra. ℓ_1-magic. *SIAM News*, 39(9), 2006.
[13] S. J. Cummins, M. M. Francois, and D. B. Kothe. Estimating curvature from volume fractions. *Computers & Structures*, 83(6-7):425–434, 2005.

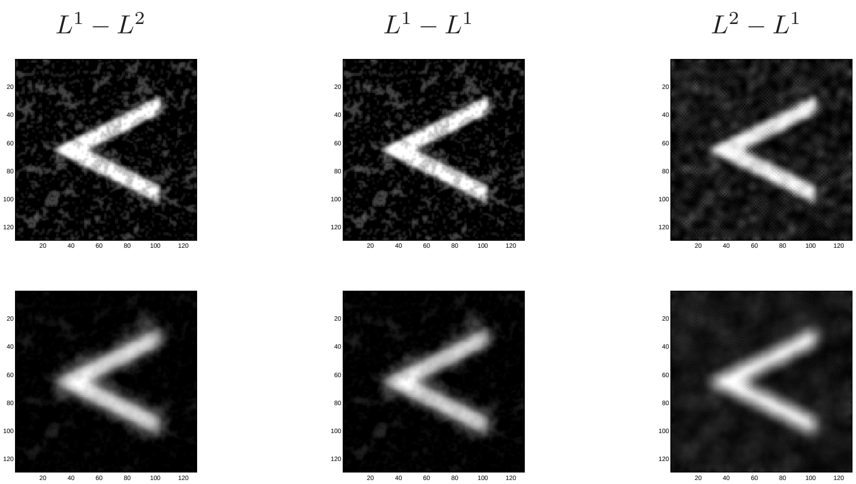

FIGURE 13. Multiphase Flow with Noise: reconstruction and smoothing of sharp interfaces for 20% of salt-and-pepper noise. Left: Smoothed solution with the algorithm with L^1 distance and L^2 smoothing term, with $\varepsilon = 3 \cdot 10^{-3}$ (first row) and $\varepsilon = 10^{-2}$ (second row). Middle: Smoothed solution with the algorithm with L^1 distance and L^1 smoothing term, with $\varepsilon = 1$, $r = 10^{-3}$ and 1 iterations of the augmented Lagrangian (first row), and $\varepsilon = 1$, $r = 4 \cdot 10^{-3}$ and 1 iterations (second row); Right: Smoothed solution with the algorithm with L^2 distance and L^1 smoothing term, with $\varepsilon = 1$, $r = 10^{-4}$ and 4 iterations of the augmented Lagrangian algorithm (first row) and $\varepsilon = 1$ and $r = 10^{-4}$ and 10 iterations (second row).

[14] E. J. Dean and R. Glowinski. Numerical methods for fully nonlinear elliptic equations of the Monge-Ampère type. *Comp. Meth. Appl. Mech. Engrg.*, 195:1344–1386, 2006.

[15] X. Feng, M. Neilan, and A. Prohl. Error analysis of finite element approximations of the inverse mean curvature flow arising from the general relativity. *Numer. Math.*, 108:93–119, 2007.

[16] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.*, 213(1):141–173, 2006.

[17] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems.* Springer-Verlag, New York, NY, 1984.

[18] R. Glowinski and P. Le Tallec. *Augmented Lagrangians and Operator-Splitting Methods in Nonlinear Mechanics.* SIAM, Philadelphia, 1989.

[19] J.-L. Guermond and B. Popov. Linear advection with ill-posed boundary conditions via L^1-minimization. *Numerical Analysis and Modeling*, 4:39–47, 2007.

[20] T. Kärkkäinen, K. Kunish, and K. Majava. Denoising of smooth images using L^1-fitting. *Computing*, 74(4):353–376, 2005.

[21] K. Majava, R. Glowinski, and T. Kärkkäinen. Solving a non-smooth eigenvalue problem using operator-splitting methods. *International Journal of Computer Mathematics*, 84(6):825–846, 2007.

[22] R. Pawula, S. Rice, and J.Roberts. Distribution of the phase angle between two vectors perturbed by Gaussian noise. *Communications, IEE Transactions*, 30(8):1828–1841, 1982.

[23] F. Ramirez-Mireles. On the performance of ultra-wide-band signals in Gaussian noise and dense multipath. *Vehicular Technology*, 50(1):244–249, 2001.

[24] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *J. Comput. Phys.*, 141:112–152, 1998.

[25] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60, 1992.

[26] S.M.Lu, S. F. Liang, and C. T. Lin. A HVS-directed neural-network-based approach for salt-pepper impulse noise removal. *Journal of Information Science and Engineering*, 22:925–939, 2006.

[27] M. Sussman, K. Smith, M.Y. Hussaini, M. Ohta, and R.-Z. Wei. A sharp interface method for incompressible two-phase flows. *J. Comput. Phys.*, 221(2):469–505, 2007.

[28] M. W. Williams, D. B. Kothe, and E. G. Puckett. Accuracy and convergence of continuum surface tension models. In Cambridge Cambridge Univ. Press, editor, *Fluid Dynamics at Interfaces*, pages 294–305. Fluid Dynamics at Interfaces, Gainesville, FL, 1998, 1999.

[29] B. Wohlberg and P. Rodríguez. An iteratively reweighted norm algorithm for minimization of total variation functionals. *IEEE Signal Processing Letters*, 14(12):948–951, 2007.

[30] C. Xing. An effective method for removing heavy salt-and-pepper noise. In *Image and Signal Processing CISP'08, Beijing University*, volume 1, pages 3–6, 2008.

Department of Mathematics, University of Houston, 4800 Calhoun Rd, Houston, TX 77204-3008, USA

E-mail: caboussat@math.uh.edu and roland@math.uh.edu and vpons@math.uh.edu

URL: http://www.math.uh.edu/∼caboussat/ and http://www.math.uh.edu/∼roland/