Deutsche Forschungsgemeinschaft
Priority Programme 1962

# *The Proximal Map of the Weighted Mean Absolute Error*

Lukas Baumgärtner, Roland Herzog, Stephan Schmidt, Manuel Weiß

SPP 1962 | Non-smooth and Complementarity-based Distributed Parameter Systems: Simulation and Hierarchical Optimization

Preprint Number SPP1962-195

received on September 27, 2022

Edited by
SPP1962 at Weierstrass Institute for Applied Analysis and Stochastics (WIAS)
Leibniz Institute in the Forschungsverbund Berlin e.V.
Mohrenstraße 39, 10117 Berlin, Germany
E-Mail: spp1962@wias-berlin.de

World Wide Web: http://spp1962.wias-berlin.de/

# The Proximal Map of the Weighted Mean Absolute Error*

Lukas Baumgärtner† Roland Herzog‡ Stephan Schmidt† Manuel Weiß‡

We investigate the proximal map for the weighted mean absolute error function. An algorithm for its efficient and vectorized evaluation is presented. As a demonstration, this algorithm is applied as part of a checkerboard algorithm to solve a total-variation image denoising (ROF) problem.

**Keywords.** proximal map, weighted mean absolute error, multi-thresholding

**AMS subject classifications (MSC2010).** 90C25, 68U10, 94A08, 46N10

## 1 Introduction

The proximity operator or proximal map plays a fundamental role in non-smooth optimization; see for instance Chambolle, Pock, 2011; Combettes, Pesquet, 2011; Parikh, Boyd, 2014. Given a function $f\colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, the proximal map $\operatorname{prox}_f\colon \mathbb{R}^n \to \mathbb{R}^n$ is defined as the solution of the problem

$$\text{Minimize} \quad f(y) + \frac{1}{2}\|y - x\|_2^2, \quad \text{where } y \in \mathbb{R}^n. \tag{1.1}$$

Under the mild condition that $f$ is proper, lower semicontinuous and convex, $\operatorname{prox}_f$ is well-defined. We refer the reader to Bauschke, Combettes, 2011, Ch. 12.4 for details and further properties.

In this paper we present theory and an efficient algorithm for the evaluation of prox $f$, where $f\colon \mathbb{R} \to \mathbb{R}$ is defined as

$$f(x) \coloneqq \sum_{i=1}^{N} w_i\,|x - d_i|. \tag{1.2}$$

*This work was supported by DFG grants HE 6077/10–2 and SCHM 3248/2–2 within the Priority Program SPP 1962 (Non-smooth and Complementarity-based Distributed Parameter Systems: Simulation and Hierarchical Optimization), which is gratefully acknowledged.

†Institut für Mathematik, Humboldt-Universität zu Berlin, 10099 Berlin, Germany (lukas.baumgaertner@hu-berlin.de, https://www.mathematik.hu-berlin.de/en/people/mem-vz/1693318, ORCID 0000-0003-1007-4815, s.schmidt@hu-berlin.de, https://www.mathematik.hu-berlin.de/en/people/mem-vz/1693090, ORCID 0000-0002-4888-0794).

‡Interdisciplinary Center for Scientific Computing, Heidelberg University, 69120 Heidelberg, Germany (roland.herzog@iwr.uni-heidelberg.de, https://scoop.iwr.uni-heidelberg.de, ORCID 0000-0003-2164-6575, manuel.weiss@iwr.uni-heidelberg.de, https://scoop.iwr.uni-heidelberg.de, ORCID 0000-0002-6098-9725).

Here $w_i > 0$ are given, positive weights and $d_i \in \mathbb{R}$ are given data, $i = 1, \dots, N$ for some $N \in \mathbb{N}$. We refer to (1.2) as the weighted mean absolute error. Any of its minimizers is known as a weighted median of the data $\{d_i\}$. Clearly, $f$ is proper, continuous and convex, and so is $\gamma f$ for any $\gamma > 0$.

By definition, the proximal map $\operatorname{prox}_{\gamma f} \colon \mathbb{R} \to \mathbb{R}$ for $f$ as in (1.2) is given by

$$\operatorname{prox}_{\gamma f}(x) \coloneqq \operatorname*{arg\,min}_{y \in \mathbb{R}} \gamma \sum_{i=1}^{N} w_i \, |y - d_i| + \frac{1}{2}(y - x)^2. \tag{1.3}$$

In case $N = 1$, problem (1.3) reduces to the well-known problem

$$\operatorname*{arg\,min}_{y \in \mathbb{R}} \gamma \, w \, |y - d| + \frac{1}{2}(y - x)^2 \tag{1.4}$$

with $w > 0$ and $d \in \mathbb{R}$, whose unique solution is explicitly given in terms of the soft-thresholding operator $S_r(x) \coloneqq \max\{0, \, |x| - r\} \operatorname{sgn}(x)$. In this case, we have

$$\operatorname{prox}_{\gamma f}(x) = d + S_{\gamma w}(x - d) = d + \max\{0, \, |x - d| - \gamma \, w\} \operatorname{sgn}(x - d). \tag{1.5}$$

This map, often with $d = 0$, arises in many iterative schemes for the solution of problems involving the 1-norm; see for instance Daubechies, Defrise, De Mol, 2004; Goldstein, Osher, 2009. We therefore refer to (1.3) as a multi-thresholding operation.

We wish to point out that our problem of interest (1.3) is different from the LASSO problem

$$\text{Minimize} \quad \|Ay - d\|_2^2 + \lambda \, \|y\|_1 \quad \text{where } y \in \mathbb{R}^n, \tag{1.6}$$

see Tibshirani, 1996; Chen, Donoho, Saunders, 1998. In the latter, $y$ is multi-dimensional and the deviation of its image under a linear map $A$ from a data vector $d$ is measured. By contrast, in (1.3) we measure the deviation of a scalar $y$ from multiple data points $d_i$. Moreover, the roles of the 1-norm and the 2-norm are reversed in (1.3) and (1.6).

We point out that Li, Osher, 2009 have considered the slightly more general problem

$$\operatorname*{arg\,min}_{y \in \mathbb{R}} \sum_{i=1}^{N} w_i \, |y - d_i| + F(y) \tag{1.7}$$

with $F$ strictly convex, differentiable and $F'$ bijective. The prototypical examples are functions $F(y) = \lambda \, |y - x|^\alpha$ with $\alpha > 1$.

We concentrate on the case $F(y) = \frac{1}{2\gamma}(y - x)^2$, which agrees with (1.3). In contrast to Li, Osher, 2009, we provide a vectorized, open-source implementation of (1.3). We also demonstrate the utility of our implementation of (1.3) by solving, similarly as in Li, Osher, 2009, an image denoising problem using a block coordinate descent (checkerboard) algorithm. In order to overcome the generic failure of convergence of such a method to the unique minimizer, we combine it with restarts based on the steepest descent direction. The emphasis in this paper, however, is on the efficient solution of (1.3).

This paper is structured as follows: We establish an algorithm for the evaluation of the proximal map of the weighted mean absolute error (1.3) in Section 2 and prove its correctness in Theorem 2.1. In

Section 3 we briefly discuss the structural properties of the proximal map. We conclude this paper by showing an application of the proposed algorithm to an image denoising problem, using the ROF model Rudin, Osher, Fatemi, 1992.

## 2 Algorithm for the Evaluation of the Proximal Map

In this section, we derive an efficient algorithm for the evaluation of the proximal map $\operatorname{prox}_{\gamma f}(x)$ (1.3) and prove its correctness in Theorem 2.1. To this end, we assume that the points $d_i$ have been sorted and duplicates have been removed and their weights added. As a result, we can assume $d_1 < d_2 < \ldots < d_N$. Moreover, we assume $\gamma > 0$, $N \geq 1$ and $w_i > 0$ for all $i = 1, \ldots, N$. Summands with $w_i = 0$ can obviously be dropped from the sum in (1.3).

We divide the real line into the intervals

$$I_1 := (-\infty, d_1], \quad I_i := [d_{i-1}, d_i] \text{ for } i = 2, \ldots, N \quad \text{and} \quad I_{N+1} := [d_N, \infty), \tag{2.1}$$

which overlap in the given data points. It is also useful to set $d_0 := -\infty$ and $d_{N+1} := \infty$. We further introduce the forward and reverse cumulative weights as

$$\mu_i := \sum_{\ell=1}^{i} w_\ell, \quad \nu_i := \sum_{\ell=i}^{N} w_\ell, \quad i = 1, \ldots, N. \tag{2.2}$$

We extend these definitions by setting $\mu_0 := 0$, $\mu_{N+1} := \mu_N$ and $\nu_{N+1} := \nu_{N+2} := 0$. We therefore have $\nu_i = \mu_N - \mu_{i-1}$ for all $i = 1, \ldots, N+2$. Using this notation, we can rewrite the derivative of $f$ as

$$f'(y) = \mu_{i-1} - \nu_i \quad \text{for } y \in \operatorname{int} I_i = (d_{i-1}, d_i), \quad i = 1, \ldots, N+1. \tag{2.3}$$

This formula reflects the fact that $f$ is piecewise linear and convex since $\mu_{i-1} - \nu_i$ is monotone increasing with $i$. Moreover, $f'(y) = \mu_0 - \nu_1 = -\sum_{\ell=1}^{N} w_\ell < 0$ holds for all $y \in \operatorname{int} I_1$ (points to the left of smallest data point $d_1$), and $f'(y) = \mu_N - \nu_{N+1} = \sum_{\ell=1}^{N} w_\ell > 0$ holds for all $y \in \operatorname{int} I_{N+1}$ (points to the right of the largest data point $d_N$). At $y = d_i$, $i = 1, \ldots, N$, $f$ is non-differentiable but we can specify its subdifferential, which is

$$\partial f(d_i) = [\mu_{i-1} - \nu_i, \mu_i - \nu_{i+1}], \quad i = 1, \ldots, N. \tag{2.4}$$

The objective

$$\Phi(y) := \gamma f(y) + \frac{1}{2}(y - x)^2 = \gamma \sum_{i=1}^{N} w_i |y - d_i| + \frac{1}{2}(y - x)^2 \tag{2.5}$$

of (1.3) is piecewise quadratic and strongly convex. Its derivative is thus strongly monotone and it satisfies

$$\Phi'(y) < 0 \quad \text{for all } y < \min\{d_1, x\} \quad \text{and} \quad \Phi'(y) > 0 \text{ for all } y > \max\{d_N, x\}. \tag{2.6}$$

Consequently, the unique minimizer of $\Phi$ lies between these bounds.

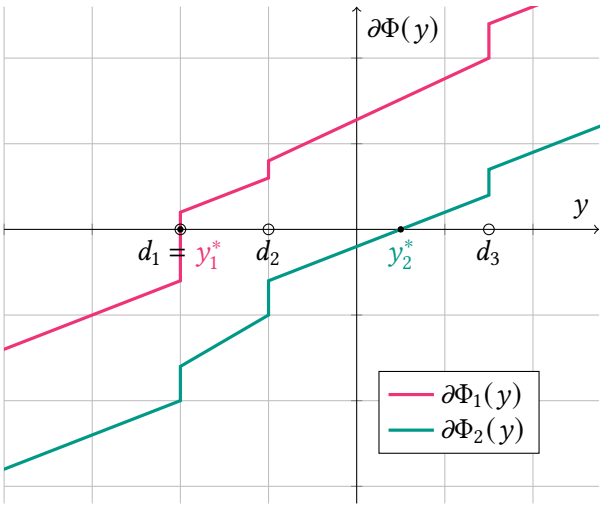


Figure 2.1: Visualization of the subdifferential $\partial\Phi$ of the objective $\Phi$. In the first (upper) case, $y_1^* = d_1$ holds. In the second (lower) case, we have $y_2^* \in \operatorname{int} I_3$.

The idea to finding the unique minimizer $y^*$ of (2.5) is to locate the smallest index $1 \le k \le N+1$ such that $y^* \le d_k$ holds, i. e., the nearest data point to the right of $y^*$. In other words, we need to find $1 \le k \le N+1$ such that

$$\lim_{y \searrow d_{k-1}} \Phi'(y) = \gamma\,(\mu_{k-1} - \nu_k) + d_{k-1} - x < 0 \tag{2.7a}$$

$$\text{and} \quad \lim_{y \searrow d_k} \Phi'(y) = \gamma\,(\mu_k - \nu_{k+1}) + d_k - x \ge 0 \tag{2.7b}$$

holds. Now we can distinguish two cases: $y^* = d_k$ and $y^* < d_k$. The first case applies if and only if

$$\gamma\,(\mu_{k-1} - \nu_k) + d_k - x \le 0. \tag{2.8}$$

Otherwise, $y^*$ lies in $\operatorname{int} I_k$ and thus it is the unique minimizer $x - \gamma\,(\mu_{k-1} - \nu_k)$ of the locally quadratic objective $\Phi$. In either case, once the index $1 \le k \le N+1$ has been identified, $y^*$ is given by

$$y^* = \min\{d_k, x - \gamma\,(\mu_{k-1} - \nu_k)\}. \tag{2.9}$$

Both cases are also depicted in Figure 2.1.

The considerations above lead to Algorithm 1. In our implementation, we evaluate (2.10) for all $k$ simultaneously and benefit from the quantities being monotone increasing with $k$ when finding the first non-negative entry.

Let us prove the correctness of Algorithm 1.

**Theorem 2.1.** *Under the assumptions stated in Algorithm 1, this algorithm returns $y^* = \operatorname{prox}_{\gamma f}(x)$, the unique solution of (1.3).*

*Proof.* Let $1 \le k \le N+1$ be the index found in Line 1. First suppose $2 \le k \le N$. Then $d_{k-1}$ and $d_k$ are both finite, and (2.7), (2.10) imply

$$\max \partial\Phi(d_{k-1}) = \lim_{y \searrow d_{k-1}} \Phi'(y) < 0 \quad \text{and} \quad \max \partial\Phi(d_k) = \lim_{y \searrow d_k} \Phi'(y) \ge 0.$$

**Algorithm 1** Evaluation of (1.3), the proximal map of the weighted mean absolute error.

**Input:** data points $d_1 < d_2 < \ldots < d_N \in \mathbb{R}$, $N \geq 1$, and $d_0 := -\infty$, $d_{N+1} := \infty$
**Input:** weight vector $w \in \mathbb{R}^N$ with entries $w_i > 0$
**Input:** prox parameter $\gamma > 0$ and point of evaluation $x \in \mathbb{R}$
**Output:** $y = \operatorname{prox}_{\gamma f}(x)$, the unique solution of (1.3)

1: Find the smallest index $1 \leq k \leq N+1$ that satisfies

$$\gamma\,(\mu_k - \nu_{k+1}) + d_k - x \geq 0 \tag{2.10}$$

2: **return** $y := \min\{d_k, x - \gamma\,(\mu_{k-1} - \nu_k)\}$

Owing to the properties of the subdifferential of strongly convex functions, there exists a unique point $y^* \in (d_{k-1}, d_k]$ such that $0 \in \partial\Phi(y^*)$, i. e., $y^*$ is the unique minimizer of (1.3). This point either belongs to int $I_k$, or else $y^* = d_k$ holds. In the first case, $\Phi$ is differentiable, so that $\Phi'(y^*) = 0$ holds, yielding

$$x - \gamma\,(\mu_{k-1} - \nu_k) = y^* < d_k.$$

Otherwise, we have $y^* = d_k$, and $0 \in \partial\Phi(d_k)$ implies

$$d_k \leq x - \gamma\,(\mu_{k-1} - \nu_k). \tag{2.11}$$

In either case, the unique solution $y^*$ of (1.3) is determined by

$$y^* = \min\{d_k, x - \gamma\,(\mu_{k-1} - \nu_k)\},$$

which is the quantity returned in Line 2.

It remains to verify the marginal cases $k = 1$ and $k = N+1$. In case $k = N+1$, we have $\lim_{y \searrow d_N} \Phi'(y) < 0$ due to the minimality of $k$. Hence,

$$d_N < y^* = x - \gamma\,(\mu_N - \nu_{N+1}) < d_{N+1} = \infty.$$

A similar reasoning applies in the case $k = 1$. □

We provide an efficient and vectorized Python implementation of Algorithm 1 in ….[1] It allows the simultaneous evaluation of (1.3) for multiple values of $x$, provided that each instance of (1.3) has the same number $N$ of data points. The weights $w_i$ and data points $d_i$ as well as the prox parameter $\gamma$ may vary between instances. The discussion so far assumed positive weights for simplicity, but the case $w_i = 0$ is a simple extension and it is allowed in our implementation. This is convenient in order to solve problem instances simultaneously which differ with respect to the number of data points $N$. Using zero weights, we can easily pad all instances to the same number of data points. In addition, data points are allowed to be duplicate, i. e., we only require $d_1 \leq d_2 \leq \ldots \leq d_N \in \mathbb{R}$, $N \geq 1$. Notice that since the data points are sorted, finding the index in Line 1 is of complexity $\log(N)$.

[1] The repository will be made public upon a first successful round of reviews.

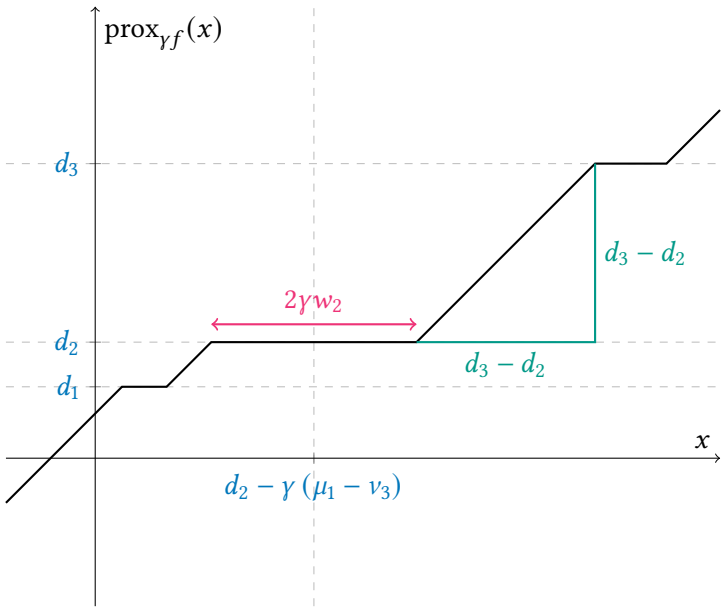


Figure 3.1: Example of the proximal map $\mathrm{prox}_{\gamma f}$ in case $N = 3$.

## 3 Structure of $\mathrm{prox}_{\gamma f}$

In this section, we briefly discuss the structure of the map $x \mapsto \mathrm{prox}_{\gamma f}(x)$. Since it generalizes the soft-thresholding operation (1.5), it is not surprising that we obtain a graph which features a staircase pattern. An illustrative plot for certain choices of weights $w_i$, data points $d_i$, and the parameter $\gamma > 0$ is shown in Figure 3.1. Each of the $N$ distinct data points provides one plateau in the graph.

Two alternating regimes occur for $y^* = \mathrm{prox}_{\gamma f}(x)$, as $x$ ranges over $\mathbb{R}$. First, when $y^* \in \operatorname{int} I_k$ holds, then (2.9) implies that $y^*$ is an affine function of $x$ with slope 1. This is the case for $x$ whose associated index $k$ is constant, i. e.,

$$x \in \gamma\,(\mu_k - \nu_{k+1}) + [d_k, d_{k+1}].$$

As $x$ increases beyond the upper bound, $y^*$ enters a constant regime which applies to

$$x \in \gamma\,(\mu_{k-1} - \nu_{k+1}) + d_k + [-\gamma w_k, \gamma w_k].$$

Notice that the case $N = 1$ reduces to the soft-thresholding map (1.5) with only one plateau.

## 4 Application To Image Denoising

In this section, we present the application of Algorithm 1 to a classical (ROF) total-variation image denoising problem going back to Rudin, Osher, Fatemi, 1992. Given noisy image data $f_{i,j}$ of size $D_1 \times D_2$,

we seek $u_{i,j}$ of the same dimension which solves

$$\text{Minimize} \quad \mathcal{H}(u) := \frac{1}{2}\sum_{i=1}^{D_1}\sum_{j=1}^{D_2}(u_{i,j} - f_{i,j})^2 + \beta\sum_{i=1}^{D_1-1}\sum_{j=1}^{D_2}|u_{i+1,j} - u_{i,j}| + \beta\sum_{i=1}^{D_1}\sum_{j=1}^{D_2-1}|u_{i,j+1} - u_{i,j}|, \quad u \in \mathbb{R}^{D_1\times D_2}. \tag{4.1}$$

Well-known solution approaches to (4.1) include the primal-dual hybrid gradient method Chambolle, Pock, 2011 and the split Bregman iteration Goldstein, Osher, 2009. The latter requires the solution of a Laplacian problem for $u_{i,j}$ in each iteration. A simpler approach, considered in Li, Osher, 2009, is to partition the unknowns in (4.1) into two disjoint subsets, according to a checkerboard pattern. In this case, problem (4.1) with only one subset of unknowns decouples into independent problems, each of which is of type (1.3) with weights $w_i = 1$ and $\gamma = \beta$ and can be solved efficiently and in parallel using Algorithm 1. We give further implementation details below.

Alternating over both subsets of unknowns, one obtains a block-coordinate descent method as proposed in Li, Osher, 2009, Sec. 3. Unfortunately, such a method does not necessarily converge towards the global minimizer for non-smooth objectives; see for instance Friedman et al., 2007, Sec. 2. Therefore, Li, Osher, 2009 proposed to restart the block-coordinate descent algorithm using a random perturbation of the final iterate upon stagnation.

We depart from this restarting strategy in the following way. Upon stagnation of the block-coordinate descent method, we evaluate the steepest descent direction by orthogonally projecting (w.r.t. the Euclidean norm $\|\cdot\|$) the zero vector onto the subdifferential of the objective $\mathcal{H}$ from (4.1):

$$d = -\operatorname{proj}_{\partial\mathcal{H}(u)}(\mathbf{0}) = -\operatorname*{arg\,min}_{s\in\partial\mathcal{H}(u)}\|s\|^2. \tag{4.2}$$

Due to the structure of the subdifferential of the absolute value function $|\cdot|$, this amounts to solving a quadratic optimization problem with $D_1D_2$ unknowns and sparse linear equality as well as bound constraints, which describe the condition $s \in \partial\mathcal{H}(u)$. We employ the QP solver OSQP (Stellato et al., 2020, https://github.com/osqp/osqp) for the purpose of solving (4.2).

The steepest descent direction (4.2) is used in the following way in Algorithm 2. First, it serves as a perturbation direction upon stagnation, in contrast to the random perturbation proposed in Li, Osher, 2009; see Line 10. Second, the norm $\|d\|$ can serve as a stopping criterion; see Line 14.

In Lines 4 and 5, we use subvector indexing. That is, $u_w$ refers to the subvector of $u$ with "white" indices copied and "black" indices zeroed. The roles are reversed for $u_b$. Consequently, $u = u_b + u_w$ holds. Subvector indexing can be conveniently done in Python using logical indexing.

Notice that Lines 4 and 5 require the evaluation of a function of the (1.3) for many arguments in parallel, where we benefit from our vectorized implementation of Algorithm 1. All norms in Theorem 2.1 are Frobenius norms for matrices. Line 10 is implemented using a backtracking strategy starting with initial step size $\alpha = 0.5$, which is then halved until the condition $\mathcal{H}(u^k + \alpha d) < \mathcal{H}(u^k)$ is met.

We wish to emphasize that we do not propose Algorithm 2 as a novel solver for image denoising problems. We rather consider it here as a source of problems of type (1.3), which can be solved efficiently

**Algorithm 2** Checkerboard scheme to approximately solve (4.1).

**Input:** tolerances $\mathrm{TOL}_{\mathrm{inner}}$, $\mathrm{TOL}_{\mathrm{outer}}$
**Output:** $u^k$, an approximate solution of (4.1)

1: $k := 0$
2: **repeat**
3: **repeat**
4: $u_w^{k+1} := \arg\min_{u_w} \mathcal{H}\left(u_b^k + u_w\right)$ using Algorithm 1
5: $u_b^{k+1} := \arg\min_{u_b} \mathcal{H}\left(u_b + u_w^{k+1}\right)$ using Algorithm 1
6: Set $k := k + 1$
7: **until** $\|u^k - u^{k-1}\| \leq \mathrm{TOL}_{\mathrm{inner}}$
8: $d := -\operatorname{proj}_{\partial\mathcal{H}(u^k)}(\mathbf{0})$
9: **if** $\|d\| > \mathrm{TOL}_{\mathrm{outer}}$ **then**
10: Choose $\alpha > 0$ with $\mathcal{H}(u^k + \alpha d) < \mathcal{H}(u^k)$
11: Set $u^{k+1} := u^k + \alpha d$
12: Set $k := k + 1$
13: **end if**
14: **until** $\|d\| \leq \mathrm{TOL}_{\mathrm{outer}}$

by our proposed Algorithm 1. In practice, Algorithm 2 can also be used effectively as a preliminary solver stage for (4.1), before one switches to, e. g., the split Bregman or Chambolle-Pock iteration.

For verification purposes, we show the outcome of Algorithm 2. As a test case, we choose the well-known cameraman image of size $D_1 = D_2 = 256$. We add zero-mean Gaussian noise with standard deviation $\sigma = 50$ independently pixel by pixel. We then truncate the values to the range $[0, 255]$ to obtain the noisy image shown in Figure 4.1a. We apply Algorithm 2 to the image denoising problem (4.1) with parameter $\beta = 10$. The inner tolerance is set to $\mathrm{TOL}_{\mathrm{inner}} = 10^{-4}$. We observe that even for a coarse outer tolerance of $\mathrm{TOL}_{\mathrm{outer}} = 300$, a good reconstruction is obtained; see Figure 4.1b. This tolerance is reached after $k = 42$ iterations, of which 37 are iterations of the loop in Line 4–Line 6 and 5 are executions of Line 8–Line 13. In particular, the subdifferential projection step in Line 8, which amounts to solving a quadratic optimization problem, is carried out 5 times. For comparison, we include an "exact" solution of (4.1) obtained by a split Bregman iteration with very tight tolerances in Figure 4.1.

Each call to Algorithm 1 (Lines 4 and 5 in Algorithm 2) evaluates the proximity operator (1.3) in parallel for half the number of pixels, i. e., $2^{15} = 32\,768$ instances. Using padding with zero weights for instances of (1.3) pertaining to points on the boundary, all instances have $N = 4$ data points $d_i$ (the current values for all neighbors north, south, east and west) and weights $w_i \in \{0, 1\}$. Timing results are reported in Table 4.1. They were obtained on a laptop with an 8-core Intel Core i5 CPU with 1.6 GHz and 16 GiB RAM, running Ubuntu 22.04 and Python 3.10.

| subroutine | number of calls | total time | time per call |
|---|---|---|---|
| Algorithm 1 (32 768 solves of (1.3) in parallel) | 74 | 10.7 s | 0.14 s |
| solution of QP (Line 8) using OSQP | 5 | 11.7 s | 2.34 s |

Table 4.1: Timing results.

(a) Noisy cameraman image.

(b) Approximate solution obtained with Algorithm 2.

(c) Exact solution of (4.1) obtained by a split Bregman iteration.

Figure 4.1: Numerical results to obtain timings for Algorithm 1.

## References


Bauschke, H. H.; P. L. Combettes (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. With a foreword by Hédy Attouch. Springer, New York. DOI: 10.1007/978-1-4419-9467-7.

Chambolle, A.; T. Pock (2011). "A first-order primal-dual algorithm for convex problems with applications to imaging". *Journal of Mathematical Imaging and Vision* 40.1, pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

Chen, S. S.; D. L. Donoho; M. A. Saunders (1998). "Atomic decomposition by basis pursuit". *SIAM Journal on Scientific Computing* 20.1, pp. 33–61. DOI: 10.1137/s1064827596304010.

Combettes, P. L.; J.-C. Pesquet (2011). "Proximal splitting methods in signal processing". *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Vol. 49. Springer Optimization and Its Applications. Springer, New York, pp. 185–212. DOI: 10.1007/978-1-4419-9569-8_10.

Daubechies, I.; M. Defrise; C. De Mol (2004). "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". *Communications on Pure and Applied Mathematics* 57.11, pp. 1413–1457. DOI: 10.1002/cpa.20042.

Friedman, J.; T. Hastie; H. Höfling; R. Tibshirani (2007). "Pathwise coordinate optimization". *The Annals of Applied Statistics* 1.2, pp. 302–332. DOI: 10.1214/07-AOAS131.

Goldstein, T.; S. Osher (2009). "The split Bregman method for $L1$-regularized problems". *SIAM Journal on Imaging Sciences* 2.2, pp. 323–343. DOI: 10.1137/080725891.

Li, Y.; S. Osher (2009). "A new median formula with applications to PDE based denoising". *Communications in Mathematical Sciences* 7.3, pp. 741–753. DOI: 10.4310/cms.2009.v7.n3.a11.

Parikh, N.; S. Boyd (2014). "Proximal algorithms". *Foundations and Trends in Optimization* 1.3, pp. 127–239. DOI: 10.1561/2400000003.

Rudin, L. I.; S. Osher; E. Fatemi (1992). "Nonlinear total variation based noise removal algorithms". *Physica D* 60.1–4, pp. 259–268. DOI: 10.1016/0167-2789(92)90242-F.

Stellato, B.; G. Banjac; P. Goulart; A. Bemporad; S. Boyd (2020). "OSQP: an operator splitting solver for quadratic programs". *Mathematical Programming Computation* 12.4, pp. 637–672. DOI: 10.1007/s12532-020-00179-2.

Tibshirani, R. (1996). "Regression shrinkage and selection via the Lasso". *Journal of the Royal Statistical Society. Series B. Methodological* 58.1, pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.