# On Cyclic Gradient Descent Reprojection

S. Setzer [*] G. Steidl[†] J. Morgenthaler[†]

September 14, 2011

**Abstract**

In recent years, convex optimization methods were successfully applied for various image processing tasks and a large number of first-order methods were designed to minimize the corresponding functionals. Interestingly, it was shown recently in [19] that the simple idea of so-called “superstep cycles” leads to very efficient schemes for time-dependent (parabolic) image enhancement problems as well as for steady state (elliptic) image compression tasks. The ”superstep cycles” approach is similar to the nonstationary (cyclic) Richardson method which has been around for over sixty years.

In this paper, we investigate the incorporation of superstep cycles into the gradient descent reprojection method. We show for two problems in compressive sensing and image processing, namely the LASSO approach and the Rudin-Osher-Fatemi model that the resulting simple *cyclic* gradient descent reprojection algorithm can numerically compare with various state-of-the-art first-order algorithms. However, due to the nonlinear projection within the algorithm convergence proofs even under restrictive assumptions on the linear operators appear to be hard. We demonstrate the difficulties by studying the simplest case of a two-cycle algorithm in $\mathbb{R}^2$ with projections onto the Euclidian ball.

## 1 Introduction

Many sparse recovery problems as well as image processing tasks such as denoising, deblurring, inpainting and image segmentation can be formulated as convex optimization problems. To minimize the corresponding functionals, first-order methods, i.e., methods which only use gradient information of the functional were extensively exploited in recent years. The most popular ones are gradient descent reprojection methods introduced in [18, 22], see [6] for further references, and their variants such as FISTA [5], Barzilai-Borwein techniques [3, 11] and primal-dual methods [10, 36].
On the other hand, the idea of so-called “super-time stepping” was recently revitalized from another point of view within *fast explicit diffusion* (FED) schemes in [19]. More precisely, the authors provided very efficient schemes for time-dependent (parabolic) image enhancement problems as well as for steady state (elliptic) image compression. In the latter case, FED schemes were speeded up by embedding them in a cascadic coarse-to-fine approach. Indeed the idea of “super-time stepping” proposed by Gentzsch et al. [16, 17] for the explicit solution of parabolic partial differential equations is very similar to those of the nonstationary

[*] Saarland University, Dept. of Mathematics and Computer Science, Campus E1.1, 66041 Saarbrücken, Germany

[†] University of Kaiserslautern, Dept. of Mathematics, Felix-Klein-Center, 67653 Kaiserslautern, Germany

(cyclic) Richardson method [2, 7, 15]: zeros of Tschebyscheff polynomials were used as varying acceleration parameters in the algorithm in a cyclic way. Although these nonstationary acceleration parameters violate the convergence restrictions on an iterative algorithm in 50 percent of all cases, the overall cycle is still in agreement with these restrictions. Hence the theoretical convergence of the algorithm is ensured. However, practical implementation of these cyclic methods require a proper ordering of the acceleration parameters to avoid the accumulation of round-off errors in case of larger cycles.
In this paper, we are interested in incorporating cyclic supersteps in gradient descent reprojection algorithms. Indeed our numerical experiments show that this simple idea can speed up the fixed step-length version of the algorithm significantly and can even compare with various state-of-the-art first-order algorithms. However, due to the nonlinear projection operator involved in the algorithm it seems to be hard to provide any convergence analysis as a simple case study underlines.

The rest of the paper is organized as follows. In Section 2, we review the basic idea of the method of "super-time stepping" and of the nonstationary (cyclic) Richardson method. In Section 3 we incorporate cyclic supersteps within the gradient descent reprojection method and call the resulting approach the cyclic gradient descent reprojection method. Then, we examine the convergence of the method in a simple case study. Section 4 compares our cyclic gradient descent reprojection method with various first-order algorithms for two sparse recovery and image processing tasks, namely for the LASSO problem and the Rudin-Osher-Fatemi approach. While the first one requires projections onto the $\ell_\infty$-ball, the second method involves projections onto the (generalized) $\ell_1$-ball.

## 2 Modified Cyclic Richardson Method

In this section we briefly explain the idea of so-called "super-time stepping" [16, 17] which is closely related to the nonstationary (cyclic) Richardson method [2, 7, 15] so that we call the first one a modified cyclic Richardson method. Consider the standard example of the heat equation

$$u_t = \triangle u = u_{xx} + u_{yy} \tag{1}$$

on $[0,1]^2$ with Neumann boundary conditions and initial condition $u(x,y,0) = f(x,y)$. A simple explicit scheme to approximate the solution of (1) on the spatial-temporal grid with spatial mesh size $\delta x = \frac{1}{N}$ and time step size $\delta t$ is given by

$$\begin{aligned} u^{(0)} &= f, \\ u^{(k+1)} &= \big(I - \frac{\delta t}{(\delta x)^2} L\big)\, u^{(k)}, \quad k = 0, 1, \ldots, \end{aligned} \tag{2}$$

where $u^{(k)}$ is the column vector obtained by columnwise reshaping $(u_{i,j}^{(k)})_{i,j=0}^{N-1}$, and $u_{i,j}^{(k)} \approx u((i+\frac{1}{2})\delta x, (j+\frac{1}{2})\delta x, k\delta t)$. The matrix $L$ results from the approximation of the derivatives in the Laplacian by symmetric finite differences. More precisely, we have that $L = \nabla^{\mathrm{T}} \nabla$, where

$\nabla$ is the discrete gradient operator $\nabla : u \mapsto \begin{pmatrix} u_x \\ u_y \end{pmatrix}$ given by

$$\nabla := \begin{pmatrix} I \otimes D \\ D \otimes I \end{pmatrix} \quad \text{with} \quad D := \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{N,N}. \tag{3}$$

The matrix $L$ is a symmetric, positive semi-definite matrix which eigenvalues are given by $\lambda_{i,j} = 4\left(\sin(i\pi/(2N))^2 + \sin(j\pi/(2N))^2\right)$, $i,j = 0,\dots,N-1$ so that $0 \le \lambda_{i,j} < 8$. Let $\lambda_{max}(L) = \|L\|_2$ denote the largest eigenvalue of $L$. Then the above scheme converges if and only if the eigenvalues of $I - \frac{\delta t}{(\delta x)^2} L$ given by $1 - \frac{\delta t}{(\delta x)^2}\lambda_{i,j}$ are within the interval $(-1,1]$ which is the case if and only if $\frac{\delta t}{(\delta x)^2} \le \frac{1}{4}$. Note that in this case $u^{(k)}$ converges to a constant vector whose entries are equal to the mean value of $f$. In [16, 17] the authors suggested to speed up the algorithm by incorporating "superstep cycles". To understand the basic idea we provide the following proposition.

**Proposition 2.1.** *Let $c_i := \cos\left(\frac{\pi(2i+1)}{2(2n+1)}\right)$ and $\tau_i := 1/c_i^2$, $i = 0,\dots,n-1$. Then we have for a symmetric matrix $A$ with eigenvalues in $[0,1]$ that*

$$\mathcal{A} := \prod_{i=0}^{n-1} (I - \tau_i A)$$

*has eigenvalues in $(-1,1]$.*

**Proof:** First note that $\{0, \pm c_i : i = 0,\dots,n-1\}$ are the zeros of the Tschebyscheff polynomial of first kind $T_{2n+1}$. Using Vieta's theorem, we see that

$$\prod_{i=0}^{n-1} c_i^2 = 2^{-2n}(2n+1).$$

Let

$$P_n(x^2) := 2^{2n} \prod_{i=0}^{n-1} (x^2 - c_i^2) = T_{2n+1}(x)/x.$$

Then, we have that

$$\begin{aligned} \max_{y\in[0,1]} (-1)^n \frac{1}{2n+1} P_n(y) &= (-1)^n \frac{1}{2n+1} P_n(0) = 1, & (4)\\ \min_{y\in[0,1]} (-1)^n \frac{1}{2n+1} P_n(y) &> -1. & (5) \end{aligned}$$

Next, we rewrite $\mathcal{A}$ as

$$\begin{aligned} \mathcal{A} &= (-1)^n \prod_{l=0}^{n-1} \tau_l \prod_{i=0}^{n-1} (A - c_i^2 I) \\ &= (-1)^n \frac{2^{2n}}{2n+1} \prod_{i=0}^{n-1} (A - c_i^2 I) = (-1)^n \frac{1}{2n+1} P_n(A). \end{aligned}$$

By (4) and (5) this yields the assertion. □

In [16, 17] the following algorithm was proposed.

$$\begin{aligned} u^{(0)} &= f, \\ u^{(sn+i+1)} &= (I - \frac{\tau_i}{8} L) u^{(sn+i)}, \quad i = 0, 1, \ldots, n-1, \; s = 0, 1, \ldots. \end{aligned} \tag{6}$$

This iteration scheme has an inner cycle of length $n$ whose iteration matrices can have eigenvalues with absolute values much larger than 1. However, by Proposition 2.1 the overall iteration matrix of the inner cycle has again eigenvalues in $(-1, 1]$ so that the convergence of the whole algorithm is assured in exact arithmetic. In the ordinary explicit scheme (2), we arrive after $nS$ steps of maximal length $\delta t = \frac{(\delta x)^2}{4}$ at $nS \frac{(\delta x)^2}{4}$. Since

$$\sum_{i=0}^{n-1} \tau_i = \frac{2}{3} n(n+1),$$

we have after $nS$ steps in (6) the time length $\frac{2}{3} n(n+1) S \frac{(\delta x)^2}{8}$ which is a larger time interval for $n \geq 3$.

The recursion (6) is closely related to the following nonstationary (cyclic) Richardson algorithm [7, 15, 32] which solves the linear system of equations $Au = b$ by

$$\begin{aligned} u^{(sn+i+1)} &= u^{(sn+i)} + \nu_i (b - A u^{(sn+i)}) \\ &= (I - \nu_i A) u^{(sn+i)} + \nu_i b, \qquad i = 0, 1, \ldots, n-1, \; s = 0, 1, \ldots. \end{aligned}$$

Here, $A$ is assumed to be a symmetric, positive definite matrix with eigenvalues in $[d_1, d_2]$, $0 < d_1 < d_2$ and $\nu_i$ are the reciprocals of the zeros of the Tschebyscheff polynomials $T_n$ on $[d_1, d_2]$, i.e.,

$$\nu_i = \frac{2}{d_2 + d_1 - (d_2 - d_1) \cos\left(\frac{\pi(2i+1)}{2n}\right)}.$$

Although Richardson's original method was a stationary one with fixed $\nu_i = \nu$ he always observed that better convergence can be obtained for varying $\nu_i$. In subsequent papers, numerical properties of the nonstationary Richardson methods and various applications were discussed. For an overview see the preprint [2].

Note that for $d_1 = 0$ and $d_2 = 1$ which was our setting in Proposition 2.1, we obtain that $\nu_i = 1/\sin^2\left(\frac{\pi(2i+1)}{4n}\right)$. Of course, assuming $d_1 = 0$ neglects that $A$ has to be positive definite. We call the following algorithm the modified cyclic Richardson method.

**Algorithm (Modified Cyclic Richardson Method)**

```
Initialization: $u^{(0)}$, $A$ symmetric, $b$, $\alpha \geq \|A\|_2$
For $s = 0, 1, \ldots$ repeat until a convergence criterion is reached
    For $i = 0, \ldots, n-1$ repeat
        $u^{(sn+i+1)} = u^{(sn+i)} + \frac{\tau_i}{\alpha}(b - A u^{(sn+i)})$.
```

All the above algorithms converge in exact arithmetic which is of course not provided by a computer. In practice, round-off errors can accumulate throughout the cycles and cause numerical instabilities for larger $n$. This is in particular the case if we apply the acceleration parameters within the algorithm in ascending or descending order. Indeed, the success of the cyclic algorithms depends on the proper ordering of the acceleration parameters $\tau_i$, resp. $\nu_i$, see [1]. The so-called "Lebedev-Finogenov ordering" of $\nu_i$ which makes the cyclic Richardson iteration computationally stable was first proposed by Lebedev-Finogenov [21] and a stability analysis for cycles of lengths $n$ which are powers of two was given in [33].

In [16, 17], the following heuristic procedure was suggested to order the values $\tau_i$. Let $1 < \kappa < n$ be an integer having no common divisors with $n$. Then, we permute the order of the $\tau_i$ by $\tau_{\pi(i)}$ with

$$\pi(i) := i \cdot \kappa \bmod n, \quad i = 0, \ldots, n-1. \tag{7}$$

Up to now it is not clear which values of $\kappa$ lead to the best stability results.

# 3 Cyclic Gradient Descent Reprojection Method

## 3.1 Supersteps in Gradient Descent Reprojection

Recently, gradient descent reprojection algorithms were applied in various image processing tasks, in particular when minimizing functionals containing the Rudin-Osher-Fatemi regularization term [9, 26] or in sparse approximation and compressed sensing. To improve the convergence of the gradient descent reprojection algorithm various first-order algorithms as Nesterov's algorithm [25] and the related FISTA [5], Barzilai-Borwein techniques [3, 11] or primal dual methods [10, 36] were developed. Here, we propose a very simple speed up by incorporating supersteps into the gradient descent reprojection algorithm. In Section 4, we will see that the resulting algorithm can compete with the other state-of-the-art algorithms.

We are interested in minimizers of the convex functional

$$\operatorname*{argmin}_{u \in \mathbb{R}^M} \left\{ \frac{1}{2} \|Bu - f\|_2^2 + \iota_C(u) \right\}, \tag{8}$$

where $f \in \mathbb{R}^N$, $B \in \mathbb{R}^{N,M}$, $C$ is a closed, convex set and $\iota_C$ is the indicator function of the set $C$ defined by $\iota_C(u) := 0$ for $u \in C$ and $\iota_C(u) := +\infty$ for $u \notin C$.

Note that without the term $\iota_C$ the solutions of (8) are given by the solutions of $B^{\mathrm{T}}Bu = B^{\mathrm{T}}f$ which can be computed by the cyclic Richardson method with $A := B^{\mathrm{T}}B$ and $b := B^{\mathrm{T}}f$. Denoting by $P_C$ the orthogonal projection onto $C$, our cyclic gradient descent reprojection method reads as follows:

**Algorithm (Cyclic Gradient Descent Reprojection Method)**

```
Initialization: $u^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\alpha \geq \|B\|_2^2$
For $s = 0, 1, \ldots$ repeat until a convergence criterion is reached
    For $i = 0, \ldots, n-1$ repeat
        $u^{(sn+i+1)} = P_C \left( u^{(sn+i)} + \frac{\tau_i}{\alpha} B^{\mathrm{T}}(f - Bu^{(sn+i)}) \right).$
```

An operator $T : \mathbb{R}^N \to \mathbb{R}^N$ is called *firmly nonexpansive* if

$$\|Tx - Ty\|_2^2 \leq \langle Tx - Ty, x - y \rangle \qquad \forall x, y \in \mathbb{R}^N.$$

A firmly nonexpansive operator is nonexpansive, i.e., a linear symmetric operator (in matrix form) is firmly nonexpansive if and only if all its eigenvalues lie within the intervall $(-1,1]$. If $T$ is firmly nonexpansive and has at least one fixed point, then the sequence $\left(T^k u^{(0)}\right)_{k\in\mathbb{N}}$ converges for any starting point $u^{(0)} \in \mathbb{R}^N$ to a fixed point of $T$. For more information on firmly nonexpansive operators or more general averaged operators, see [4].
It is well-known that $P_C$ is a firmly nonexpansive operator. However, we cannot apply Proposition 2.1 to prove convergence of the algorithm since we do not have in general that $P_C A_1 P_C A_0$ is nonexpansive if $A_1 A_0$ is nonexpansive as the following example shows.

**Example.** Let $C \subset \mathbb{R}^2$ be the closed $\ell_2$-ball so that $P_C$ is given by (9). Then we obtain for

$$x := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad y := \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}, \quad 0 < \varepsilon < 1$$

that $\|x - y\|_2 = \varepsilon$. Further, we have for

$$A_0 := \begin{pmatrix} 1 & 0 \\ 0 & a \end{pmatrix}, \quad A_1 := \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{a} \end{pmatrix}, \quad a \geq 1$$

that $A_1 A_0$ is nonexpansive. We compute

$$A_0 x = P_C A_0 x = A_1 P_C A_0 x = P_C A_1 P_C A_0 x = x$$

and

$$A_0 y = \begin{pmatrix} 1 \\ a\varepsilon \end{pmatrix}, \; P_C A_0 y = \frac{1}{c} \begin{pmatrix} 1 \\ a\varepsilon \end{pmatrix}, \; A_1 P_C A_0 y = P_C A_1 P_C A_0 y = \frac{1}{c} \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}$$

with $c := \sqrt{1 + (a\varepsilon)^2}$ and get

$$\|P_C A_1 P_C A_0 x - P_C A_1 P_C A_0 y\|_2^2 = \left\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \frac{1}{c} \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix} \right\|_2^2 = \frac{(c-1)^2 + \varepsilon^2}{c^2}.$$

Using this relation we conclude for $c > 2/(1 - \varepsilon^2)$ that

$$\|P_C A_1 P_C A_0 x - P_C A_1 P_C A_0 y\|_2 > \|x - y\|_2$$

so that $P_C A_1 P_C A_0$ is not nonexpansive.

Indeed, it seems to be hard give a convergence proof for the cyclic gradient descent reprojection method even under stronger conditions on $\alpha$. We demonstrate the difficulties by a case study in the following subsection.

### 3.2 A Case Study

In this subsection, let $C := \{x \in \mathbb{R}^N : \|x\|_2 \leq 1\}$ so that

$$P_C x = \begin{cases} x & \text{if } x \in C, \\ x/\|x\|_2 & \text{otherwise.} \end{cases} \tag{9}$$

We are interested in the cyclic gradient descent reprojection method with $f = 0$, more precisely, in the nonlinear operator

$$T := \prod_{i=1}^{n} (P_C A_{n-i}) = P_C A_{n-1} \ldots P_C A_0,$$

where $A_i := I - \tau_i A$ and $A$ is a symmetric matrix with eigenvalues in $[0, 1)$.

**Remark 3.1.** *In one dimension, i.e., if $N = 1$ it is easy to check that $T : \mathbb{R} \to \mathbb{R}$ is nonexpansive since*

$$\begin{aligned} |Tx - Ty| &= |P_C A_{n-1} \dots P_C A_0 x - P_C A_{n-1} \dots P_c A_0 y| \\ &\leq |A_{n-1} \dots P_C A_0 x - A_{n-1} \dots P_C A_1 y| \\ &= |A_{n-1}||P_C A_{n-2} \dots P_C A_0 x - P_C A_{n-2} \dots P_C A_0 y| \\ &\leq \dots \\ &\leq |\prod_{i=1}^{n} A_{n-i}| \, |x - y| \ \leq \ |x - y|, \end{aligned}$$

*where the last inequality follows by Proposition 2.1.*

By the following lemma we can restrict our attention also in higher dimensions to diagonal matrices $A_i$.

**Lemma 3.2.** *Let $A_i = U \Lambda_i U^T$, $i = 0, \dots, n-1$ be the eigenvalue decompositions of $A_i$ with an orthonormal matrix $U$ and diagonal matrices $\Lambda_i$. Then the operator $T$ is firmly nonexpansive if and only if $S := \prod_{i=1}^{n} (P_C \Lambda_{n-i})$ is firmly nonexpansive.*

**Proof:** Since $\|Ux\|_2 = \|x\|_2$ it follows that $P_C Ux = UP_C x$. Consequently, we obtain

$$\begin{aligned} T = \prod_{i=1}^{n} (P_C A_{n-i}) x &= P_C U \Lambda_{n-1} U^{\mathrm{T}} \, \dots \, P_C U \Lambda_2 U^{\mathrm{T}} \, P_C U \Lambda_0 U^{\mathrm{T}} x \\ &= P_C U \Lambda_{n-1} U^{\mathrm{T}} \, \dots \, P_C U \Lambda_2 \underbrace{U^{\mathrm{T}} U}_{I} \, P_C \Lambda_0 U^{\mathrm{T}} x \\ &= \dots \\ &= U \prod_{i=1}^{n} (P_C \Lambda_{n-i}) \, U^{\mathrm{T}} x. \end{aligned}$$

Hence it follows with $u := U^{\mathrm{T}} x$ and $v := U^{\mathrm{T}} y$ that

$$\|Tx - Ty\|_2^2 = \|USU^{\mathrm{T}} x - USU^{\mathrm{T}} y\|_2^2 = \|Su - Sv\|_2^2$$

and

$$\langle Tx - Ty, x - y \rangle = \langle USu - USv, x - y \rangle = \langle USu - USv, Uu - Uv \rangle = \langle Su - Sv, u - v \rangle.$$

Since $U^{\mathrm{T}}$ is a one-to-one mapping, we obtain the assertion. □

In the rest of this section, we consider the cyclic gradient descent reprojection method for the case $N = 2$ and $n = 2$. More precisely, we are interested if the operator $P_C \Lambda_0 P_C \Lambda_1$ is nonexpansive, where $c_0 := \cos(\pi/10)$, $c_1 := \cos(3\pi/10)$, $\tau_i := 1/c_i^2$, $i = 0, 1$ and

$$\Lambda_i := I - \tau_i \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix} = \begin{pmatrix} \lambda_{i0} & 0 \\ 0 & \lambda_{i1} \end{pmatrix} = \frac{1}{c_i^2} \begin{pmatrix} c_i^2 - \lambda_0 & 0 \\ 0 & c_i^2 - \lambda_1 \end{pmatrix}, \qquad \lambda_i \in [0, 1). \qquad (10)$$

The matrix $\Lambda_0$ has eigenvalues in $(-0.1056, 1]$ and the matrix $\Lambda_1$ in $(-1.8944, 1]$. Note that by Lemma 3.2 we can restrict our attention to diagonal matrices $\Lambda_i$. Then we can claim the following proposition which “proof” contains a numerical component.

**Proposition 3.3.** *Let $\Lambda_i$, $i = 0, 1$ be given by (10), where $\lambda_i \in [0, 1 - \varepsilon]$, $\varepsilon \geq 0.16$. Then the relation*

$$\|P_C \Lambda_0 P_C \Lambda_1 u - P_C \Lambda_0 P_C \Lambda_1 v\|_2 \leq \|u - v\|_2 \tag{11}$$

*holds true, i.e., $P_C \Lambda_0 P_C \Lambda_1$ is nonexpansive.*

**"Proof"** (with numerical computation): By Remark 3.1, we can restrict our attention to invertible matrices $\Lambda_i$, $i = 0, 1$ i.e., matrices without zero eigenvalues, since we are otherwise in the one-dimensional setting. Using $x := \Lambda_1 u$ and $y := \Lambda_1 v$ and regarding that $\Lambda_0$ and $P_C$ are nonexpansive, the assertion (11) can be rewritten as

$$\|\Lambda_0 P_C x - \Lambda_0 P_C y\|_2 \leq \|\Lambda_1^{-1}(x - y)\|_2. \tag{12}$$

We distinguish three cases.
1. If $\|x\|_2 \leq 1$ and $\|y\|_2 \leq 1$, then (12) is equivalent to $\|\Lambda_0 \Lambda_1 (u - v)\|_2 \leq \|u - v\|_2$ which holds true by Proposition 2.1.
2. Let $\|x\|_2 \leq 1$ and $\|y\|_2 > 1$. W.l.o.g. we assume that $x_0, x_1 \geq 0$, i.e., $x$ lies within the first quadrant. Then, (12) becomes

$$\|\Lambda_0 \left( x - \frac{y}{\|y\|_2} \right) \|_2 \leq \|\Lambda_1^{-1}(x - y)\|_2$$

and using (10) further

$$\lambda_{00}^2 \left( x_0 - \frac{y_0}{\|y\|_2} \right)^2 + \lambda_{01}^2 \left( x_1 - \frac{y_1}{\|y\|_2} \right)^2 \leq \frac{1}{\lambda_{10}^2}(x_0 - y_0)^2 + \frac{1}{\lambda_{11}^2}(x_1 - y_1)^2$$

and

$$0 \leq \frac{1}{\lambda_{10}^2}(x_0 - y_0)^2 - \lambda_{00}^2 \left( x_0 - \frac{y_0}{\|y\|_2} \right)^2 + \frac{1}{\lambda_{11}^2}(x_1 - y_1)^2 - \lambda_{01}^2 \left( x_1 - \frac{y_1}{\|y\|_2} \right)^2.$$

Multiplying by $\frac{(c_1^2 - \lambda_0)^2 (c_1^2 - \lambda_1)^2}{c_1^4}$ yields

$$\begin{aligned} 0 \quad \leq \quad & (c_1^2 - \lambda_1)^2 \left( (x_0 - y_0)^2 - \gamma_0 \left( x_0 - \frac{y_0}{\|y\|_2} \right)^2 \right) \\ & + (c_1^2 - \lambda_0)^2 \left( (x_1 - y_1)^2 - \gamma_1 \left( x_1 - \frac{y_1}{\|y\|_2} \right)^2 \right), \end{aligned} \tag{13}$$

where by the proof of Proposition 2.1

$$\gamma_i := \frac{(c_0^2 - \lambda_i)^2 (c_1^2 - \lambda_i)^2}{c_0^4 c_1^4} = \left( \frac{1}{5} P_2(\lambda_i) \right)^2 \leq 1.$$

We consider the following cases for $y$.
2.1. If $y$ lies within the area denoted by 3 in Fig. 1, then $(x_i - y_i)^2 \geq \left( x_i - \frac{y_i}{\|y\|_2} \right)^2$ for $i = 0, 1$ so that (13) holds true.

2.2. Let $y$ lie within the areas denoted by 1 and $1'$ in Fig. 1. Any element in the area $1'$ can be written as $y = (-y_0, y_1)^{\mathrm{T}}$, where $(y_0, y_1)^{\mathrm{T}}$ lies within area 1. Then, (13) reads

$$\begin{aligned} 0 \quad &\leq \quad (c_1^2 - \lambda_1)^2 \left( (x_0 + y_0)^2 - \gamma_0 \left( x_0 + \frac{y_0}{\|y\|_2} \right)^2 \right) \\ &\quad + (c_1^2 - \lambda_0)^2 \left( (x_1 - y_1)^2 - \gamma_1 \left( x_1 - \frac{y_1}{\|y\|_2} \right)^2 \right). \end{aligned}$$

By straightforward computation we see that for $1/\|y\|_2 < 1$ the relation

$$(x_0 - y_0)^2 - \gamma_0 \left( x_0 - \frac{y_0}{\|y\|_2} \right)^2 \leq (x_0 + y_0)^2 - \gamma_0 \left( x_0 + \frac{y_0}{\|y\|_2} \right)^2$$

holds true. Therefore, we can restrict our attention to area 1.

Let $y$ lie within area 1. By the following argument, we may assume that $\|x\|_2 = 1$. If $\|x\|_2 < 1$, we shift it to $\tilde{x} := x + (\delta, 0)^{\mathrm{T}}$ such that $\|\tilde{x}\|_2 = 1$. We have that $\delta \in (0, e_0]$, where $e_0 := y_0/\|y\|_2 - x_0$. Then, the second summand on the right-hand side of (13) is the same for $x$ and $\tilde{x}$. Concerning the first summand, we obtain with $d_0 := y_0 - x_0$ that

$$(x_0 + \delta - y_0)^2 - \gamma_0 \left( x_0 + \delta - \frac{y_0}{\|y\|_2} \right)^2 = (d_0 - \delta)^2 - \gamma_0 (e_0 - \delta)^2 \leq d_0^2 - \gamma_0 e_0^2$$

if $\delta \leq \frac{2(d_0 - \gamma_0 e_0)}{1 - \gamma_0}$ which holds true since $e_0 \leq \frac{2(d_0 - \gamma_0 e_0)}{1 - \gamma_0}$. Therefore it remains to consider the case $\|x\|_2 = 1$. Changing our setting to polar coordinates

$$x := \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix}, \; y := \|y\|_2 \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix}$$

where $0 \leq \varphi \leq \psi \leq \frac{\pi}{2}$, inequality (13) becomes

$$\begin{aligned} 0 \quad &\leq \quad (c_1^2 - \lambda_1)^2 \left( (\cos\psi - \|y\|_2 \cos\varphi)^2 - \gamma_0 (\cos\psi - \cos\varphi)^2 \right) \\ &\quad + (c_1^2 - \lambda_0)^2 \left( (\sin\psi - \|y\|_2 \sin\varphi)^2 - \gamma_1 (\sin\psi - \sin\varphi)^2 \right). \end{aligned} \tag{14}$$

The right-hand side is a convex, quadratic function in $\|y\|_2$ and we can compute the values where this function is zero. Now we have checked *numerically* if the largest of these (real) values is less or equal than 1. In this case (14) is valid since $\|y\|_2 > 1$. To this end, we have used the grid $\lambda_i := 0 : 0.001 : 0.84$ for $i = 0, 1$ and $\psi := 0 : 0.001\pi : \pi/2$, $\varphi \leq \psi$. The desired property follows for $\lambda_i \in [0, 0.84]$, $i = 1, 2$.

2.3. If $y$ lies within the area denoted by 2 or $2'$ in Fig. 1, then we can argue as in the case 2.2 by exchanging the roles of the coordinates.

3. If $1 < \|x\|_2 \leq \|y\|_2$, then (12) becomes

$$\|\Lambda_0 \left( \frac{x}{\|x\|_2} - \frac{y}{\|y\|_2} \right) \|_2 \leq \|\Lambda_1^{-1} (x - y)\|_2. \tag{15}$$

Since $\frac{y}{\|y\|_2} = P_C \left( \frac{y}{\|x\|_2} \right)$ and by case 2 we obtain

$$\begin{aligned} \frac{1}{\|x\|_2} \|\Lambda_0 \left( \frac{x}{\|x\|_2} - \frac{y}{\|y\|_2} \right) \|_2 \quad &\leq \quad \|\Lambda_0 \left( P_C \left( \frac{x}{\|x\|_2} \right) - P_C \left( \frac{y}{\|x\|_2} \right) \right) \|_2 \\ &\leq \quad \|\Lambda_1^{-1} \left( \frac{x}{\|x\|_2} - \frac{y}{\|x\|_2} \right) \|_2 \\ &= \quad \frac{1}{\|x\|_2} \|\Lambda_1^{-1} (x - y)\|_2 \end{aligned}$$

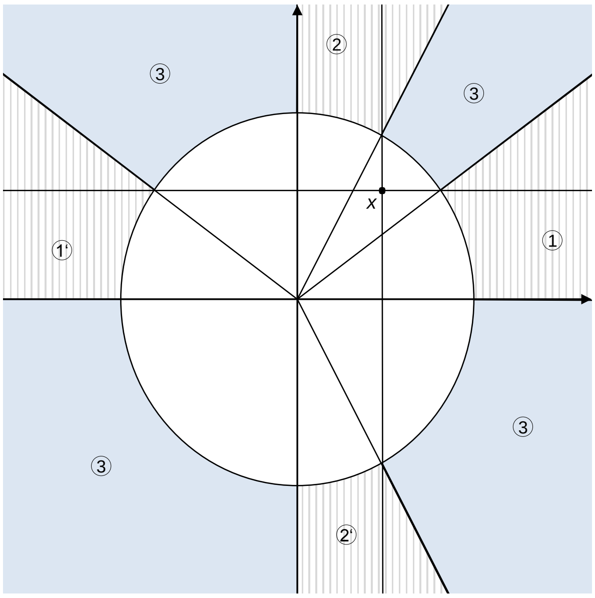

Figure 1: Areas for the study of case 2.

which implies (15). □

# 4 Numerical Comparison

In this section, we show how the cyclic gradient descent reprojection algorithm compares with other state-of-the-art algorithms. We consider the minimization problem

$$\min_{u \in \mathbb{R}^M} \{ \frac{1}{2} \|Bu - f\|_2^2 + \iota_C(u) \}, \tag{16}$$

where $B \in \mathbb{R}^{N,M}$ and $f \in \mathbb{R}^N$ are given and $C \subset \mathbb{R}^M$ denotes the set of feasible points. We restrict our attention to first-order methods, i.e., methods which only use gradient information. Algorithms of this type have become popular recently, e.g., for sparse recovery problems, see Subsection 4.1, and in image processing, cf. Subsection 4.2. We consider two groups of first-order algorithms: variants of the gradient descent reprojection algorithm and first-order primal-dual methods.

**Variants of the Gradient Descent Reprojection Algorithm** Recall that the main idea of the gradient descent reprojection algorithm, often called the gradient projection algorithm, is to perform in each iteration a gradient descent step on the quadratic part of (16) followed by projecting the resulting point back onto the feasible set $C$. We consider the following versions of the gradient projection algorithm:

i) Gradient descent reprojection algorithm with fixed step size (GP),

ii) Cyclic gradient descent reprojection algorithm (C-GP),

iii) Gradient descent reprojection algorithm with Barzilai-Borwein step sizes (BB-GP),

iv) Fast iterative threshold algorithm (FISTA) of [5].

The GP algorithm has the form

**Algorithm (GP)**
Initialization: $u^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\gamma < 2/\|B\|_2^2$
For $k = 0, 1, \ldots$ repeat until a convergence criterion is reached

$$u^{(k+1)} = P_C(u^{(k)} - \gamma B^{\mathrm{T}}(Bu^{(k)} - f)).$$

Convergence is guaranteed for any $\gamma < 2/\|B\|_2^2$. Note that $\|B\|_2^2$ is the Lipschitz constant of the quadratic part of (16).
As we will see in the experiments below, our cyclic version C-GP of this algorithm performs much better. We want to compare our algorithm C-GP to acceleration schemes of GP which have become popular recently. In [3], Barzilai and Borwein proposed to use a Quasi-Newton method with the simplest matrix $\gamma_k^{-1} I$ fulfilling the Quasi-Newton condition

$$\gamma_k^{-1} I(u^{(k)} - u^{(k-1)}) = B^{\mathrm{T}} B(u^{(k)} - u^{(k-1)}).$$

This results in the following algorithm.

**Algorithm (BB-GP)**
Initialization: $u^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\gamma_0 > 0$, $u^{(1)} = P_C(u^{(0)} - \gamma_0 B^{\mathrm{T}}(Bu^{(0)} - f))$
For $k = 1, \ldots$ repeat until a convergence criterion is reached

$$\begin{aligned} s^{(k)} &= u^{(k)} - u^{(k-1)},\ y^{(k)} = B^{\mathrm{T}} B s^{(k)}, \\ \gamma_k &= \frac{\langle s^{(k)}, s^{(k)} \rangle}{\langle s^{(k)}, y^{(k)} \rangle}, \\ u^{(k+1)} &= P_C(u^{(k)} - \gamma_k B^{\mathrm{T}}(Bu^{(k)} - f)). \end{aligned}$$

Observe that we can easily reformulate BB-GP so that we have to compute $B^{\mathrm{T}} B u^{(k)}$ only once in each iteration. Hence, BB-GP uses the same number of matrix multiplications as GP. The above form was chosen for the sake of better readability. It should be mentioned that many related Barzilai-Borwein step-size rules have been proposed in recent years. We refer to [14] for an overview and further references. Note that in general, one needs to incorporate a line search to guarantee convergence of BB-GP. However, in our experiments, it turned out that a line search was neither necessary nor beneficial for the convergence of BB-GP.
Another method designed to improve the convergence speed of GP is the fast iterative shrinkage thresholding algorithm (FISTA) in [5]. It uses a fixed step-length but combines preceding iterations in a clever way to achieve a significant speed-up for some problems which was also be shown analytically.

**Algorithm (FISTA)**
Initialization: $u^{(0)} = w^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\gamma = \|B\|_2^2$

For $k = 0, 1, \ldots$ repeat until a convergence criterion is reached

$$\begin{aligned} u^{(k+1)} &= P_C(w^{(k)} - \gamma B^{\mathrm{T}}(Bw^{(k)} - f)), \\ t_{k+1} &= \frac{1}{2}(1 + \sqrt{1 + 4t_k^2}), \\ w^{(k+1)} &= u^{(k)} + \frac{t_k - 1}{t_{k+1}}(u^{(k+1)} - u^{(k)}). \end{aligned}$$

**First-Order Primal-Dual Algorithms** An increasingly important class of algorithms are first-order methods based on the primal-dual Lagrangian formulation of the given optimization problem. We consider the following three methods:

i) Two primal-dual algorithms (CP-I/II) proposed by Chambolle and Pock in [10].

ii) The primal-dual hybrid gradient algorithm (PDHG) with dynamic step sizes of Zhu and Chan, cf., [36].

More specifically, CP-I has the following form:

**Algorithm (CP-I)**
Initialization: $u^{(0)} \in \mathbb{R}^N$, $v^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\sigma\tau < 1/\|B\|_2^2$
For $k = 0, 1, \ldots$ repeat until a convergence criterion is reached

$$\begin{aligned} u^{(k+1)} &= P_C(u^{(k)} + \sigma B^{\mathrm{T}}\tilde{v}^{(k)}), \\ v^{(k+1)} &= \frac{1}{1+\tau}(v^{(k)} - \tau B u^{(k+1)} + \tau f), \\ \tilde{v}^{(k+1)} &= v^{(k+1)} + \theta(v^{(k+1)} - v^{(k)}). \end{aligned}$$

In our experiments, we will always choose $\theta = 1$. Algorithm CP-II shown below is a variant of CP-I with dynamic step-sizes.

**Algorithm (CP-II)**
Initialization: $u^{(0)} \in \mathbb{R}^N$, $v^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\sigma_0\tau_0 < 1/\|B\|_2^2$
For $k = 0, 1, \ldots$ repeat until a convergence criterion is reached

$$\begin{aligned} u^{(k+1)} &= P_C(u^{(k)} + \sigma_k B^{\mathrm{T}}\tilde{v}^{(k)}), \\ v^{(k+1)} &= \frac{1}{1+\tau_k}(v^{(k)} - \tau_k B u^{(k+1)} + \tau_k f), \\ &\theta_k = 1/\sqrt{1 + 2\gamma\tau_k},\ \tau_{k+1} = \theta_k/\tau_k,\ \sigma_{k+1} = \sigma_k\theta_k \\ \tilde{v}^{(k+1)} &= v^{(k+1)} + \theta_k(v^{(k+1)} - v^{(k)}). \end{aligned}$$

It was shown in [10] that if the step-length parameters in CP-I/II are chosen as indicated above, the algorithms converge.
The following PDHG algorithm differs from CP-II in that $\theta_k = 0$ for all $k$ and a special dynamic step-size rule is used. Although no convergence proof exists up to now, this strategy is very fast for solving the Rudin-Osher-Fatemi model we consider in Subsection 4.2. However, it cannot be applied for the other experiments presented here since the setting is tailored for the Rudin-Osher-Fatemi model and we have no convergence for the other tasks.

**Algorithm (PDHG)**
Initialization: $u^{(0)} \in \mathbb{R}^N$, $v^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$
For $k = 0, 1, \dots$ repeat until a convergence criterion is reached

$$\begin{aligned}
u^{(k+1)} &= P_C(u^{(k)} + \tau_k B^{\mathrm{T}} v^{(k)}),\\
v^{(k+1)} &= (1-\theta_k)v^{(k)} + \theta_k(f - Bu^{(k+1)}),\\
\tau_{k+1} &= 0.2 + 0.08k,\\
\theta_{k+1} &= \frac{1}{\tau_{k+1}}\left(0.5 - \frac{5}{15+k}\right).
\end{aligned}$$

In the following experiments, we consider two different sets $C$. We start with the $\ell_1$-ball and then consider a generalization of the $\ell_\infty$-ball.

### 4.1 Projection onto the $\ell_1$-Ball

The basis pursuit problem consists of finding a *sparse* solution of an underdetermined system via the *convex* minimization problem

$$\operatorname*{argmin}_{u \in \mathbb{R}^M} \|u\|_1 \quad \text{subject to} \quad Bu = f \tag{17}$$

with $B \in \mathbb{R}^{N,M}$, $N \ll M$ and $f \in \mathbb{R}^N$ being the measured signal. This model has attracted a lot of attention recently both from a theoretical point of view as well as because of its importance for sparse approximation and compressed sensing, cf., e.g., [8, 13]. Since in most application noise is present, different problems related to (17) where proposed which relax the linear constraint. We refer to [29, 31] for comparisons of these models and further references. The noise-robust model we want to consider here is the following convex problem called LASSO (least absolute shrinkage and selection operator) which was originally proposed by Tibshirani in [28]. It has the form

$$\operatorname*{argmin}_{u \in \mathbb{R}^M} \frac{1}{2}\|Bu - f\|_2^2 \quad \text{subject to} \quad \|u\|_1 \le \xi, \tag{18}$$

with $C := \{u \in \mathbb{R}^M : \|u\|_1 \le \lambda\}$ being the closed $\ell_1$-ball, $f \in \mathbb{R}^N$ and $B \in \mathbb{R}^{N,M}$ with $N \ll M$. Recall that by solving (18) we are trying to find a *sparse* vector $u^*$ which is an *approximate* solution to the underdetermined system $Bu = f$.
For our numerical tests, we use the software described in [23]. For given $B$ and $u^*$ it computes a parameter $\xi$ and a right-hand side $f$ such that $u^*$ is a solution of (18). We choose a matrix $B \in \mathbb{R}^{200,1000}$ whose entries are independent realization of a Gaussian random variable with mean zero and standard deviation one. The vector $u^* \in \mathbb{R}^{1000}$ has 25 nonzero elements which are also independent realizations of a Gaussian random variable with mean zero and standard deviation one.

**Choice of parameters:** All the methods except BB-GP are designed to work without an additional line-search but require knowledge of $\|B\|_2^2$. Although estimating this norm can be costly, we exclude the computation of $\|B\|_2$ from the performance measure below since for some matrices used in compressed sensing, e.g., partial DCT matrices, this value is immediately known to be 1. Here, we simply normalize $B$ such that its spectral norm is equal

|  | $\|u-u^*\|_\infty < 10^{-4}$ |  | $\|u-u^*\|_\infty < 10^{-8}$ |  |
|---|---|---|---|---|
| Method | Parameters | Matrix multipl. | Parameters | Matrix multipl. |
| GP | $\gamma = 1$ | 308 | $\gamma = 1$ | 520 |
| C-GP | $n = 8$, $\kappa = 7$ | 85 | $n = 7$, $\kappa = 5$ | 158 |
| BB-GP | $\gamma_0 = 10$ | 46 | $\gamma_0 = 10$ | 68 |
| FISTA | $L = 1$ | 313 | $L = 1$ | 892 |
| CP-I | $\sigma = 5$, $\tau = 1/5$ | 145 | $\sigma = 4.2$, $\tau = 1/4.2$ | 274 |
| CP-II | $\sigma_0 = 4.6$, $\tau_0 = 1/4.6$ | 168 | $\sigma_0 = 3.9$, $\tau_0 = 1/3.9$ | 368 |

Table 1: Comparison of first-order algorithms to solve the LASSO problem (18). The parameters are hand-tuned and the results averaged over 100 experiments.

to one. In order to guarantee convergence of BB-GP, one has to use a line-search in general, cf., [30]. In our experiments, however, BB-GP did convergence without any line-search.
We optimized the parameters of all methods by hand in order to be independent from the performance of application-specific parameter strategies.

As already mentioned, there exist various variants of the Barzilai-Borwein step-length rule presented above. We tested several of them, including the Adaptive Barzilai-Borwein method (ABB) of [34], the ABBmin2 strategy proposed in [14], the cyclic Barzilai-Borwein method of [12, 20] and the GP-SS algorithm of [24]. For all these methods, we also optimized the parameters by hand but obtained results which where very similar to BB-GP so that we show only the results of the latter here. We suspect that the hand-tuning itself is the reason for this result. Observe that the SPGL1 algorithm of [29] uses the Barzilai-Borwein method applied here.
Table 1 summarizes the results of our experiments. As a performance measure, we choose the number of matrix multiplication needed to reach two different values of the maximal difference to the exact solution $u^*$. Comparing matrix multiplications allows us to be independent of the implementation, hardware and programming language used and takes into account that the matrix multiplications with the fully populated matrix $B$ are by far be the most expensive part of the algorithm. Observe that we have averaged the results of 100 experiments.
Our results confirm the observation of other papers that the Barzilai-Borwein step-length rule is very effective for sparse recovery problems. Although our C-GP algorithm is outperformed by BB-GP, we still see that it is superior to the other methods considered here.

## 4.2 Projection onto the Generalized $\ell_\infty$-Ball

Next we compare the convergence speed of the algorithms for two image denoising problems which can be written in the form (16). First, we consider the Rudin-Osher-Fatemi model for edge-preserving image denoising, cf. [26]. For (weakly) differentiable functions $v : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^2$ and a noisy image $f$, the Rudin-Osher-Fatemi model has the form

$$\operatorname*{argmin}_v \{ \frac{1}{2} \|v - f\|_{L_2(\Omega)} + \lambda \int_\Omega \sqrt{(\partial_x v)^2 + (\partial_y v)^2}\, dxdy \}. \tag{19}$$

In order to discretize (19), we use the gradient matrix $\nabla$ defined in (3). So, if we reorder the discrete noisy image columnwise into a vector $f \in \mathbb{R}^N$ we obtain the following discrete

|  | $\|v - v^*\|_\infty < 1$ |  |  |
|---|---|---|---|
| Method | Parameters | Iterations | Time [sec] |
| GP | $\gamma = 0.249$ | 253 | 1.64 |
| C-GP | $n = 19$, $\kappa = 11$ | 41 | 0.29 |
| BB-GP | $\gamma_0 = 6$ | 86 | 0.96 |
| FISTA | $\gamma = 0.125$ | 59 | 0.72 |
| CP-I | $\sigma = 1.4$, $\tau = 0.125/1.4$ | 77 | 0.78 |
| CP-II | $\sigma_0 = 1.2$, $\tau_0 = 0.125/1.2$ | 75 | 0.67 |
| PDHG |  | 46 | 0.34 |

Table 2: Comparison of first-order algorithms to solve the dual Rudin-Osher-Fatemi problem (21). Stopping criterion: maximal pixel difference to a reference solution (obtained after a large number of FISTA iterations) smaller than 1.0 in the primal variable.

version of (19)

$$\operatorname*{argmin}_{v\in\mathbb{R}^N}\{\|v - f\|_2^2 + \lambda\| \, |\nabla v| \, \|_1\}, \tag{20}$$

where we use the notation $(|\nabla v|)_i := (((I \otimes D)v)_i^2 + ((D \otimes I)v)_i^2)^{1/2}$. The dual problem of (20) has the form of (16), i.e.,

$$\operatorname*{argmin}_{u\in\mathbb{R}^{2N}}\{\frac{1}{2}\|Bu - f\|_2^2 + \iota_{\{\| \, |\cdot| \, \| \le \lambda\}}(u)\} \tag{21}$$

with $B = \nabla^{\mathrm{T}}$. Note that we can recover the solution $v^*$ of (20) from a solution $u^*$ of (21) as follows

$$v^* = f - Bu^*.$$

We show the number of iterations and runtimes of several first-order methods in Tables 2 and 3. The noisy image of size 256 × 256 we use here is shown in Figure 2 as well as the denoising result using the regularization parameter $\lambda = 25$. The experiments were conducted using a laptop with an Intel Core Duo processor 2.66 GHz running Matlab R2008b.

As in Subsection 4.1, we hand-tuned the parameters of all the methods so that they yield fastest convergence. Observe that we use the bound $\|B\|_2^2 < 8$. We see that our method C-GP outperforms the others for the first experiment where a moderate accuracy is required. For the second experiment, which uses a more restrictive stopping criterion it is only outperformed by the PDHG algorithm. Moreover, we see that the results for FISTA are much better compared to what we have seen in Subsection 4.1 whereas BB-GP is now much less efficient. Note that we have tested several BB-GP variants, including those considered in [14, 35], but this did not improve the speed of convergence.

Finally, we consider the following variant of the Rudin-Osher-Fatemi model. We substitute the norm of the gradient in (19) by the Frobenius norm of the Hessian, cf. [27]. This yields

$$\operatorname*{argmin}_{v}\{\frac{1}{2}\|v - f\|_{L_2(\Omega)} + \lambda \int_\Omega \sqrt{(\partial_{xx}v)^2 + (\partial_{xy}v)^2 + (\partial_{yx}v)^2 + (\partial_{yy}v)^2} \, dxdy\}. \tag{22}$$

We obtain a discrete version of (22) as follows

$$\operatorname*{argmin}_{v\in\mathbb{R}^N}\{\|v - f\|_2^2 + \lambda\| \, |B^{\mathrm{T}}v| \, \|_1\}, \tag{23}$$

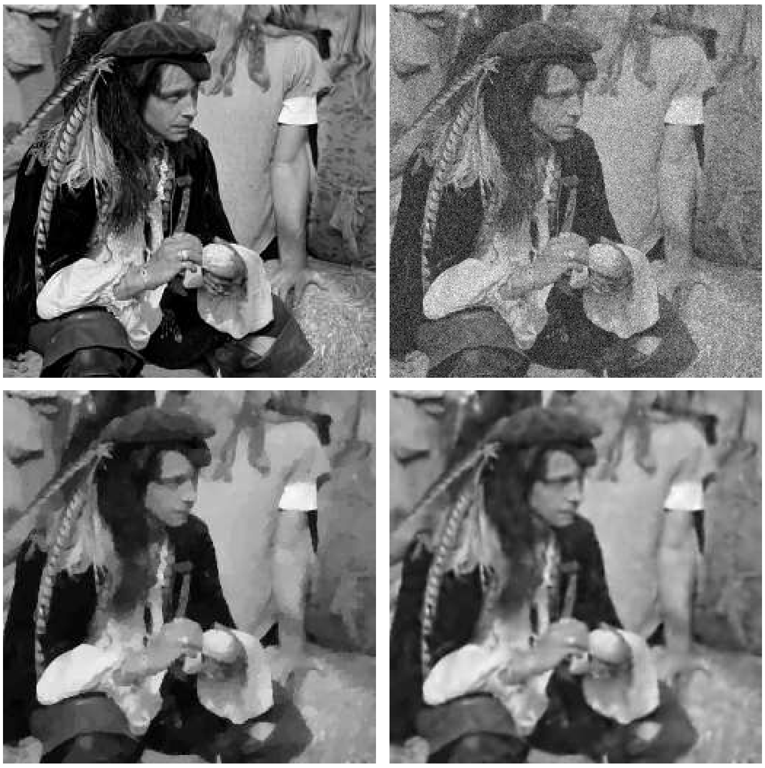

Figure 2: Top: Original image with values in $[0, 255]$ and noisy image (Gaussian noise with standard deviation 25). Bottom: Reconstruction via the Rudin-Osher-Fatemi model (21) (left) and model (24) (right).

| | $\Vert v - v^*\Vert_\infty < 0.1$ | | |
|---|---|---|---|
| Method | Parameters | Iterations | Time [sec] |
| GP | $\gamma = 0.249$ | 5073 | 32.90 |
| C-GP | $n = 38$, $\kappa = 11$ | 297 | 1.95 |
| BB-GP | $\gamma_0 = 6$ | 1066 | 12.23 |
| FISTA | $\gamma = 0.125$ | 279 | 3.10 |
| CP-I | $\sigma = 5$, $\tau = 0.125/5$ | 278 | 2.95 |
| CP-II | $\sigma_0 = 4.4$, $\tau_0 = 0.125/4.4$ | 274 | 2.55 |
| PDHG | | 194 | 1.40 |

Table 3: Comparison of first-order algorithms to solve the dual Rudin-Osher-Fatemi problem (21). Stopping criterion: maximal pixel difference to a reference solution smaller than 0.1 in the primal variable.

|  | $\lVert v - v^* \rVert_\infty < 1$ |  |  |
|---|---|---|---|
| Method | Parameters | Iterations | Time [sec] |
| GP | $\gamma = 0.0312$ | 511 | 7.57 |
| C-GP | $n = 27$, $\kappa = 19$ | 58 | 0.88 |
| BB-GP | $\gamma_0 = 6$ | 142 | 3.45 |
| FISTA | $\gamma = 1/64$ | 96 | 1.95 |
| CP-I | $\sigma = 0.2$, $\tau = 1/(64 \cdot 0.2)$ | 104 | 1.75 |
| CP-II | $\sigma_0 = 0.2$, $\tau_0 = 1/(64 \cdot 0.2)$ | 101 | 1.68 |

Table 4: Comparison of first-order algorithms to solve problem (24). Stopping criterion: maximal pixel difference to a reference solution smaller than 1 in the primal variable.

where $B^{\mathrm{T}} = \begin{pmatrix} D_{xx} \\ D_{xy} \\ D_{yx} \\ D_{yy} \end{pmatrix} = \begin{pmatrix} I \otimes D^{\mathrm{T}} D \\ D^{\mathrm{T}} D \otimes I \\ D^{\mathrm{T}} \otimes D \\ D \otimes D^{\mathrm{T}} \end{pmatrix}$ and

$$(|B^{\mathrm{T}} v|)_i := ((D_{xx} v)_i^2 + (D_{xy} v)_i^2 + (D_{yx} v)_i^2 + (D_{yy} v)_i^2)^{1/2}.$$

As above, the dual problem to (23) has the form of (16), i.e.,

$$\operatorname*{argmin}_{u \in \mathbb{R}^{4N}} \{ \frac{1}{2} \| Bu - f \|_2^2 + \iota_{\{\| \, |\cdot| \, \| \le \lambda\}}(u) \}. \tag{24}$$

Note that we can recover a solution $v^*$ of (20) from a solution and $u^*$ of (21) as follows

$$v^* = f - Bu^*.$$

Tables 4 and 5 show the performance of the first-order methods for solving (24). We use the regularization parameter $\lambda = 15$ and again two different stopping criteria. The resulting denoised image is depicted in Figure 2. Observe that we have now $\|B\|_2^2 < 64$.
PDHG using the dynamic step-length strategy described above does not converge for this problem and a simple rescaling of the parameters does not yield an efficient method. So, since we cannot apply PDHG any more, C-GP is now the fastest method for both stopping criteria. Furthermore, we notice a clearer advantage of C-GP over the remaining methods than for the case $B = \nabla^{\mathrm{T}}$.

# References

[1] V. Alexiades, G. Amiez, and P. A. Gremaud. Super-time-stepping acceleration of explicit schemes for parabolic problems. *Communications in Numerical Methods in Engineering*, 12:31–42, 1996.

[2] R. S. Anderssen and G. H. Golub. Richardson's non-stationary matrix iterative procedure. Technical report, Stanford University, Stanford, CA, USA, 1972.

[3] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, January 1988.

|  | $\|v - v^*\|_\infty < 0.1$ |  |  |
|---|---|---|---|
| Method | Parameters | Iterations | Time [sec] |
| GP | $\gamma = 0.0312$ | 4544 | 71.57 |
| C-GP | $n = 49$, $\kappa = 31$ | 241 | 3.76 |
| BB-GP | $\gamma_0 = 6$ | 961 | 24.12 |
| FISTA | $\gamma = 1/64$ | 319 | 6.57 |
| CP-I | $\sigma = 0.44$, $\tau = 1/(64 * 0.44)$ | 349 | 5.87 |
| CP-II | $\sigma_0 = 0.4$, $\tau_0 = 1/(64 * 0.4)$ | 313 | 5.31 |

Table 5: Comparison of first-order algorithms to solve problem (24). Stopping criterion: maximal pixel difference to a reference solution smaller than 0.1 in the primal variable.

[4] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, 2011.

[5] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009.

[6] D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21:174–183, 1976.

[7] M. S. Birman. On a variant of the method of successive approximations. *Vestnik LGU*, 9:69–76, 1952.

[8] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52:489–509, 2006.

[9] A. Chambolle. Total variation minimization and a class of binary MRF models. In A. Rangarajan, B. C. Vemuri, and A. L. Yuille, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition, EMMCVPR*, volume 3757 of *LNCS*, pages 136–152. Springer, 2005.

[10] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.

[11] Y.-H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, 100:21–47, 2005.

[12] Y.-H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang. The cyclic Barzilai-Borwein method for unconstrained optimization. *IMA Journal of Numerical Analysis*, 26:604–627, 2006.

[13] D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52:1289–1306, 2006.

[14] G. Frassoldati, L. Zanni, and G. Zanghirati. New adaptive stepsize selections in gradient methods. *Journal of Industrial and Management Optimization*, 4(2):299–312, 2008.

[15] M. K. Gavurin. The use of polynomials of best approximation for the improvement of the convergence of iteration processes. *Uspekhi Matem. Nauk*, 5:156–160, 1950.

[16] W. Gentzsch. Numerical solution of linear and non-linear parabolic differential equations by a time discretisation of third order accuracy. In E. H. Hirschel, editor, *Proceedings of the Third GAMM Conference on Numerical Methods in Fluid Dynamics*, pages 109–117. Vieweg&Sohn, 1979.

[17] W. Gentzsch and A. Schlüter. Über ein Einschrittverfahren mit zyklischer Schrittweitenänderung zur Lösung parabolischer Differentialgleichungen. *Zeitschrift für Angewandte Mathematik und Mechanik*, 58:415–416, 1978.

[18] A. A. Goldstein. Convex programming in Hilbert space. *Bull. Amer. Math. Soc.*, 70:709–710, 1964.

[19] S. Grewenig, J. Weickert, and A. Bruhn. From Box filtering to fast explicit diffusion. In M. Goesele, S. Roth, A. Kuijper, B. Schiele, and K. Schindler, editors, *Pattern Recognition. Lecture Notes in Computer Science, Vol. 6376*, pages 533–542. Springer, Berlin, 2010.

[20] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization*, 17:526–557, 2006.

[21] V. Lebedev and S. Finogenov. Ordering of the iterative parameters in the cyclical Chebyshev iterative method. *USSR Computational Mathematics and Mathematical Physics*, 11(2):155–170, 1971.

[22] E. S. Levitin and B. T. Polyak. Constrained minimization problems. *USSR Comput. Math. Math. Phys.*, 6:1–50, 1966.

[23] D. A. Lorenz. Constructing test instances for basis pursuit denoising. Technical report, TU Braunschweig, 2011. `http://arxiv.org/abs/1103.2897`.

[24] I. Loris, M. Bertero, C. D. Mol, R. Zanella, and L. Zanni. Accelerating gradient projection methods for l1-constrained signal recovery by steplength selection rules. *Applied and Computational Harmonic Analysis*, 27(2):247–254, 2009.

[25] Y. E. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

[26] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[27] G. Steidl. A note on the dual treatment of higher order regularization functionals. *Computing*, 76:135–148, 2006.

[28] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[29] E. van den Berg and M. P. Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31:890–912, 2008.

[30] Z. Wen, W. Yin, and D. Goldfarb. On the convergence of an active set method for l1-minimization. *Optimization Methods and Software*, 2010. To appear.

[31] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. *SIAM Journal on Scientific Computing*, 32:1832–1857, 2009.

[32] D. Young. On Richardson's method for solving linear systems with positive definite matrices. *Journal of Mathematical Physics*.

[33] A. Zawilski. Numerical stability of the cyclic Richardson iteration. *Numerische Mathematik*, 60:251–290, 1991.

[34] B. Zhou, L. Gao, and Y. H. Dai. Gradient methods with adaptive step-sizes. *Computational Optimization and Applications*, 35:69–86, 2005.

[35] M. Zhu. *Fast numerical algorithms for total variation based image restoration.* PhD thesis, University of California, Los Angeles, USA, 2008.

[36] M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. Technical report, UCLA, Center for Applied Math., 2008.