A Cyclic Projected Gradient Method

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

November 13, 2012

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 [24] 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 projected gradient 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* projected gradient 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 Euclidean 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 projected gradient methods introduced in [23, 27], see [6] for further references, and their variants such as FISTA [30, 5], Barzilai-Borwein techniques [3, 16] and primal-dual methods [15, 41].

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 [24]. 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. [21, 22] 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, 11, 20]: 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 projected gradient 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. One way to guarantee convergence is to use a line search strategy.

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 projected gradient method and call the resulting approach the cyclic projected gradient method. Then, we examine the convergence of the method in a simple case study and show how a line search strategy can be employed to guarantee convergence. Section 4 compares our cyclic projected gradient 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 ℓ_∞-ball, the second method involves projections onto the (mixed) ℓ_1-ball.

2 Modified Cyclic Richardson Method

In this section we briefly explain the idea of so-called "super-time stepping" [21, 22] which is closely related to the nonstationary (cyclic) Richardson method [2, 11, 20] 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 δt 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

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

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

Here, $\otimes$ denotes the Kronecker product. 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 [21, 22] 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

$$\max_{y\in[0,1]} (-1)^n \frac{1}{2n+1} P_n(y) \;=\; (-1)^n \frac{1}{2n+1} P_n(0) \;=\; 1, \tag{4}$$

$$\min_{y\in[0,1]} (-1)^n \frac{1}{2n+1} P_n(y) \;>\; -1. \tag{5}$$

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 [21, 22] 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 [11, 20, 37] which solves the linear system of equations $Au = b$ by

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

Here, A is assumed to be a symmetric, positive definite matrix with eigenvalues in $[d_1, d_2]$, $0 < d_1 < d_2$ and ν_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 ν_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)

1: **Initialization** $u^{(0)}$, A symmetric, b, $\alpha \geq \|A\|_2$
2: **for** $s = 0, 1, \ldots$ until a convergence criterion is reached **do**
3: **for** $i = 0, \ldots, n-1$ **do**
4: $u^{(sn+i+1)} = u^{(sn+i)} + \frac{\tau_i}{\alpha}(b - Au^{(sn+i)})$

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 τ_i, resp. ν_i, see [1]. The so-called "Lebedev-Finogenov ordering" of ν_i which makes the cyclic Richardson iteration computationally stable was first proposed by Lebedev-Finogenov [26] and a stability analysis for cycles of lengths n which are powers of two was given in [38].
In [21, 22], the following heuristic procedure was suggested to order the values τ_i. Let $1 < \kappa < n$ be an integer having no common divisors with n. Then, we permute the order of the τ_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 κ lead to the best stability results.

3 Cyclic Projected Gradient Method

3.1 Supersteps in the Projected Gradient Method

Recently, projected gradient algorithms were applied in various image processing tasks, in particular when minimizing functionals containing the Rudin-Osher-Fatemi regularization term [14, 32] or in sparse approximation and compressed sensing. To improve the convergence of the projected gradient algorithm various first-order algorithms as Nesterov's algorithm [31] and the related FISTA [30, 5], Barzilai-Borwein techniques [3, 16] or primal-dual methods [15, 41] were developed. Here, we propose a very simple speed up by incorporating supersteps into the projected gradient 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 ι_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 ι_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 projected gradient method reads as follows:

Algorithm (C-PG – Cyclic Projected Gradient Method)

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

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 interval $(-1, 1]$.

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 $C = \{x \in \mathbb{R}^2 : \|x\|_2 \leq 1\}$ be the closed ℓ_2-ball in $\mathbb{R}^2$ so that

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

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 to give a convergence proof for the cyclic projected gradient method even under stronger conditions on α. We demonstrate the difficulties by a case study in the following subsection.

3.2 A Case Study

In this subsection, we consider the ℓ_2-ball in $\mathbb{R}^N$, i.e., $C := \{x \in \mathbb{R}^N : \|x\|_2 \leq 1\}$ and the corresponding projection

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

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

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

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*

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

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 Λ_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 $P_C U x = U P_C x$. 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^{T} is a one-to-one mapping, we obtain the assertion. □

In the rest of this section, we consider the cyclic projected gradient 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

$$\Lambda_i := I - \tau_i \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix} = \begin{pmatrix} \lambda_{i0} & 0 \\ 0 & \lambda_{i1} \end{pmatrix} = \frac{1}{c_i^2}\begin{pmatrix} c_i^2 - \lambda_0 & 0 \\ 0 & c_i^2 - \lambda_1 \end{pmatrix}, \qquad \lambda_i \in [0, 1). \tag{9}$$

The matrix Λ_0 has eigenvalues in $(-0.1056, 1]$ and the matrix Λ_1 in $(-1.8944, 1]$. Note that by Lemma 3.2 we can restrict our attention to diagonal matrices Λ_i. Then we can claim the following proposition which "proof" contains a numerical component.

Proposition 3.3. *Let Λ_i, $i = 0, 1$ be given by (9), where $\lambda_i \in [0, 1-\varepsilon]$, $\varepsilon \geq 0.16$. Then, for all $u, v \in \mathbb{R}^2$ the relation*

$$\|P_C\Lambda_0 P_C\Lambda_1 u - P_C\Lambda_0 P_C\Lambda_1 v\|_2 \leq \|u - v\|_2 \tag{10}$$

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 Λ_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 Λ_0 and P_C are nonexpansive, the assertion (10) can be rewritten as

$$\|\Lambda_0 P_C x - \Lambda_0 P_C y\|_2 \leq \|\Lambda_1^{-1}(x - y)\|_2. \tag{11}$$

We distinguish three cases.
1. If $\|x\|_2 \leq 1$ and $\|y\|_2 \leq 1$, then (11) 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, (11) becomes

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

and using (9) 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

$$\begin{aligned} 0 \quad \leq \quad & (c_1^2 - \lambda_1)^2 \left((x_0 - y_0)^2 - \gamma_0 \left(x_0 - \frac{y_0}{\|y\|_2} \right)^2 \right) \\ & + (c_1^2 - \lambda_0)^2 \left((x_1 - y_1)^2 - \gamma_1 \left(x_1 - \frac{y_1}{\|y\|_2} \right)^2 \right), \end{aligned} \tag{12}$$

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 (12) 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, (12) reads

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

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 (12) 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 (12) becomes

$$\begin{aligned} 0 \quad \leq \quad & (c_1^2 - \lambda_1)^2 \left((\cos\psi - \|y\|_2 \cos\varphi)^2 - \gamma_0 (\cos\psi - \cos\varphi)^2 \right) \\ & + (c_1^2 - \lambda_0)^2 \left((\sin\psi - \|y\|_2 \sin\varphi)^2 - \gamma_1 (\sin\psi - \sin\varphi)^2 \right). \end{aligned} \tag{13}$$

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 (13) 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 (11) becomes

$$\|\Lambda_0 \left(\frac{x}{\|x\|_2} - \frac{y}{\|y\|_2} \right) \|_2 \leq \|\Lambda_1^{-1}(x - y)\|_2. \tag{14}$$

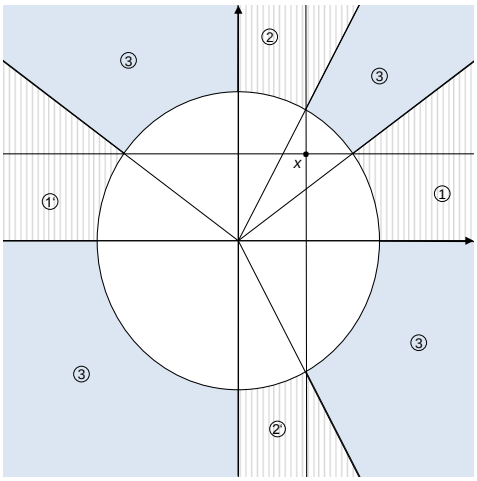

Figure 1: Areas for the study of case 2.

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

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

which implies (14). □

3.3 A convergence guarantee via nonmonotone backtracking line search

As described above, the convergence properties of the C-PG algorithm remain an open question. One way to circumvent this problem is to use a line search strategy. This is a well-known approach for projected gradient methods and different line search techniques exist. Below we show how to adapt to our setting the nonmonotone backtracking line search method which was proposed in [8], see also [9, 10], for a projected gradient method with Barzilai-Borwein step sizes. As in [8], we obtain the following convergence result.

Theorem 3.4. *Every accumulation point of the sequence $(u^{(k)})_k$ generated by the C-PG algorithm with nonmonotone backtracking line search is a solution of* (8).

Proof: The proof follows from [8, Theorem 2.3], cf. also [7, Proposition 2.3.3], and the convexity of both the quadratic function and the set C in (8).

Observe that if the set C is bounded in addition to being closed and convex, we can conclude that an accumulation point exists. Moreover, in this case every subsequence contains itself a subsequence which converges to a solution.

Algorithm (C-PG with nonmonotone backtracking line search)

Initialization $u^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\alpha \geq \|B\|_2^2$, $\xi \in (0,1)$, $K \in \mathbb{N}$
for $s = 0, 1, \ldots$ until a convergence criterion is reached **do**
  for $i = 0, \ldots, n-1$ **do**
    $u^{(sn+i+1)} = P_C\left(u^{(sn+i)} + \frac{\tau_i}{\alpha} B^{\mathrm{T}}(f - Bu^{(sn+i)})\right)$
    $t_{\max} = \max\limits_{0 \leq j \leq \min(sn+i-1, K-1)} \frac{1}{2}\|Bu^{(sn+i-j)} - f\|_2^2$
    $d = u^{(sn+i+1)} - u^{(sn+i)}$
    $\theta = 1$
    while $\frac{1}{2}\|B(u^{(sn+i)} + \theta d) - f\|_2^2 > t_{\max} + \xi\theta\langle d, B^{\mathrm{T}}(Bu^{(sn+i)} - f)\rangle$ **do** $\theta = \sigma\theta$
    $u^{(sn+i+1)} = u^{(sn+i)} + \theta d$

4 Numerical Comparison

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

$$\min_{u \in \mathbb{R}^M} \{\frac{1}{2}\|Bu - f\|_2^2 + \iota_C(u)\}, \tag{15}$$

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 and in image processing. We consider two groups of first-order algorithms: variants of the projected gradient algorithm and first-order primal-dual methods.

Variants of the Projected Gradient Algorithm Recall that the main idea of the projected gradient algorithm is to perform in each iteration a gradient descent step on the quadratic part of (15) followed by projecting the resulting point back onto the feasible set C. We consider the following versions of the projected gradient algorithm:

i) Projected gradient algorithm with fixed step size (PG),

ii) Cyclic projected gradient algorithm (C-PG),

iii) Projected gradient algorithm with Barzilai-Borwein step sizes (BB-PG),

iv) Fast iterative threshold algorithm (FISTA).

The PG algorithm has the form

Algorithm (PG)

Initialization $u^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\gamma < 2/\|B\|_2^2$
for $k = 0, 1, \ldots$ until a convergence criterion is reached **do**
  $u^{(k+1)} = P_C(u^{(k)} - \gamma B^{\mathrm{T}}(Bu^{(k)} - f))$

Convergence is guaranteed for any $\gamma < 2/\|B\|_2^2$. Note that $\|B\|_2^2$ is the Lipschitz constant of the gradient of quadratic function in (15).
As we will see in the experiments below, our cyclic version C-PG of this algorithm performs much better. Hence, we want to compare our algorithm C-PG to acceleration schemes of PG 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-PG)

Initialization $u^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\gamma_0 > 0$
for $k = 0, 1, \ldots$ until a convergence criterion is reached **do**
  $u^{(k+1)} = P_C(u^{(k)} - \gamma_k B^{\mathrm{T}}(Bu^{(k)} - f))$
  $s^{(k+1)} = u^{(k+1)} - u^{(k)}$, $y^{(k+1)} = B^{\mathrm{T}} B s^{(k+1)}$
  $\gamma_{k+1} = \frac{\langle s^{(k+1)}, s^{(k+1)} \rangle}{\langle s^{(k+1)}, y^{(k+1)} \rangle}$

Observe that we can easily reformulate BB-PG so that we have to compute $B^{\mathrm{T}} B u^{(k)}$ only once in each iteration. Hence, BB-PG uses the same number of matrix multiplications as PG. 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 [19] for an overview and further references. Note that in general, one needs to incorporate a line search to guarantee convergence of BB-PG. We apply the spectral projected gradient method SPG2 of [8] here which uses a nonmonotone backtracking line search, cf. Section 3.3.

Algorithm (BB-PG with nonmonotone backtracking line search (SPG2))

Initialization $u^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\gamma_0 > 0$, $\xi \in (0,1)$, $\rho \in (0,1)$, $0 < \alpha_{\min} < \alpha_{\max}$, $K \in \mathbb{N}$
for $k = 0, 1, \ldots$ until a convergence criterion is reached **do**
  $u^{(k+1)} = P_C\left(u^{(k)} + \gamma_k B^{\mathrm{T}}(f - Bu^{(k)})\right)$
  $t_{\max} = \max\limits_{0 \le j \le \min(k, K-1)} \frac{1}{2}\|Bu^{(k-j)} - f\|_2^2$
  $d = u^{(k+1)} - u^{(k)}$
  $\theta = 1$
  while $\frac{1}{2}\|B(u^{(k)} + \theta d) - f\|_2^2 > t_{\max} + \xi\theta\langle d, B^{\mathrm{T}}(Bu^{(k)} - f)\rangle$ **do** $\theta = \sigma\theta$
  $u^{(k+1)} = u^{(k)} + \theta d$
  $s^{(k+1)} = u^{(k+1)} - u^{(k)}$, $y^{(k+1)} = B^{\mathrm{T}} B s^{(k+1)}$
  if $\langle s^{(k+1)}, y^{(k+1)} \rangle \le 0$ **then** $\gamma_{k+1} = \alpha_{\max}$
  else $\gamma_{k+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, \frac{\langle s^{(k+1)}, s^{(k+1)} \rangle}{\langle s^{(k+1)}, y^{(k+1)} \rangle}\}\}$

Another method designed to improve the convergence speed of PG is the fast iterative shrinkage thresholding algorithm (FISTA) of [5] which builds on the method of Nesterov [30]. It uses a fixed step length but combines preceding iterations in a clever way to achieve a significant speed-up. It can be shown that the convergence rate measured as the difference of

the current function value to the optimal function value decreases as $\mathcal{O}(1/k^2)$ instead of only $\mathcal{O}(1/k)$ for the standard PG method.

Algorithm (FISTA)

Initialization $u^{(0)} = w^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\gamma = \|B\|_2^2$, $t_0 = 1$
for $k = 0, 1, \ldots$ until a convergence criterion is reached **do**
  $u^{(k+1)} = P_C(w^{(k)} - \gamma B^{\mathrm{T}}(Bw^{(k)} - f))$
  $t_{k+1} = \frac{1}{2}(1 + \sqrt{1 + 4t_k^2})$
  $w^{(k+1)} = u^{(k)} + \frac{t_k - 1}{t_{k+1}}(u^{(k+1)} - u^{(k)})$

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 [15].

ii) The primal-dual hybrid gradient algorithm (PDHG) with dynamic step sizes of Zhu and Chan, cf., [41].

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 $k = 0, 1, \ldots$ until a convergence criterion is reached **do**
  $u^{(k+1)} = P_C(u^{(k)} + \sigma B^{\mathrm{T}}\tilde{v}^{(k)})$
  $v^{(k+1)} = \frac{1}{1+\tau}(v^{(k)} - \tau B u^{(k+1)} + \tau f)$
  $\tilde{v}^{(k+1)} = v^{(k+1)} + \theta(v^{(k+1)} - v^{(k)})$

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 $u^{(0)} \in \mathbb{R}^N$, $v^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$, $\sigma_0\tau_0 < 1/\|B\|_2^2$, $\gamma > 0$
for $k = 0, 1, \ldots$ until a convergence criterion is reached **do**
  $u^{(k+1)} = P_C(u^{(k)} + \sigma_k B^{\mathrm{T}}\tilde{v}^{(k)})$
  $v^{(k+1)} = \frac{1}{1+\tau_k}(v^{(k)} - \tau_k B u^{(k+1)} + \tau_k f)$
  $\theta_k = 1/\sqrt{1 + 2\gamma\tau_k},\ \tau_{k+1} = \theta_k/\tau_k,\ \sigma_{k+1} = \sigma_k\theta_k$
  $\tilde{v}^{(k+1)} = v^{(k+1)} + \theta_k(v^{(k+1)} - v^{(k)})$

It was shown in [15] that if the step length parameters in CP-I/II are chosen as stated above, the algorithms converge. Moreover, the convergence rate measured in terms of the squared distance to the limit $v^* := \lim_{k\to\infty} v^{(k)}$, i.e., $\|v^{(k)} - v^*\|_2^2$, decreases as $\mathcal{O}(1/k^2)$, cf. [15].

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. For a recent convergence proof, see [12]. The step-size strategy makes PDHG very fast for solving the Rudin-Osher-Fatemi model which we consider in Subsection 4.2. However, it is tailored to this application and does not converge in the other test settings we examine here.

Algorithm (PDHG)

1: **Initialization** $u^{(0)} \in \mathbb{R}^N$, $v^{(0)} \in \mathbb{R}^M$, $B \in \mathbb{R}^{N,M}$, $f \in \mathbb{R}^N$
2: **for** $k = 0, 1, \ldots$ until a convergence criterion is reached **do**
3: $\quad u^{(k+1)} = P_C(u^{(k)} + \tau_k B^{\mathrm{T}} v^{(k)})$
4: $\quad v^{(k+1)} = (1 - \theta_k) v^{(k)} + \theta_k (f - B u^{(k+1)})$
5: $\quad \tau_{k+1} = 0.2 + 0.08k$
6: $\quad \theta_{k+1} = \frac{1}{\tau_{k+1}} \left(0.5 - \frac{5}{15+k}\right)$

In the following numerical experiments, we consider two different sets C. We start with the ℓ_1-ball and then consider a generalization of the ℓ_∞-ball.

4.1 Projection onto the ℓ_1-Ball

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

$$\operatorname*{argmin}_{u \in \mathbb{R}^M} \|u\|_1 \quad \text{subject to} \quad Bu = f \tag{16}$$

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 because of its interesting theoretical properties and because of its importance for sparse approximation and compressive sensing, cf., e.g., [13, 18]. Since in most applications noise is present, different problems related to (16) where proposed which relax the linear constraints. We refer to [35, 36] 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 [34]. It has the form

$$\operatorname*{argmin}_{u \in \mathbb{R}^M} \frac{1}{2} \|Bu - f\|_2^2 \quad \text{subject to} \quad \|u\|_1 \leq \xi, \tag{17}$$

where $B \in \mathbb{R}^{N,M}$ with $N \ll M$ and $f \in \mathbb{R}^N$. Recall that by solving (17) our goal is 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 [28]. For a given B and a given sparse u^* it computes a parameter ξ and a right-hand side f such that u^* is a solution of (17). We choose a matrix B with $M = 1000$ and $N = 200$ whose entries are independent realization of a Gaussian random variable with mean zero and standard deviation one. The vector u^* of length 1000 has 25 nonzero elements which are also independent realizations of a Gaussian random variable with mean zero and standard deviation one.
Table 1 summarizes the results of our experiments. As a performance measure, we choose the number of matrix multiplication needed to reach two different accuracies in terms of the

value of the objective function $F(u) := \frac{1}{2}\|Bu - f\|_2^2$. 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.
Table 1 confirms the observation of other papers that the Barzilai-Borwein step length rule is very effective for sparse recovery problems. Our C-PG algorithm is outperformed by BB-PG (SPG2) in the high-accuracy case. For the moderate-accuracy case, however, our method is faster. Furthermore, it outperforms all the other methods considered here for both stopping criteria.

Choice of parameters: We optimized the parameters of every method by hand in order to be independent of the performance of application-specific parameter heuristics. All the methods except BB-PG were applied without an additional line search and require the knowledge of $\|B\|_2$. Although estimating this norm can be costly, we exclude the computation of $\|B\|_2$ from the performance measure since for some matrices used in compressive sensing, e.g., partial DCT matrices, this value is immediately known. Here, we simply normalize B such that its spectral norm is equal to one.
It turns out that for this experiment C-PG converges without a line search. We found that the line search does not increase the number of matrix multiplications if we choose $K > n$. However, it also does not make the algorithm faster if we use a smaller value of K.
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 [39], the ABBmin2 strategy proposed in [19], the cyclic Barzilai-Borwein method of [17, 25] and the PG-SS algorithm of [29]. For all these methods, we also optimized the parameters by hand and obtained results which are very similar to BB-PG so that we show only the results of the latter here. We suspect that the hand-tuning of the parameters is the reason for this result. Observe that the SPGL1 software of [35] also uses the Barzilai-Borwein method applied here.
In our experiment, BB-PG combined with the backtracking line search of [8, 9, 10] not only comes with a convergence guarantee but is also faster than standard BB-PG. We found that the following line search parameters give the fastest convergence: $\xi = \rho = 0.5$, $\alpha_{\min} = 10^{-12}$ and $\alpha_{\max} = 10^{12}$. Although the nonmonotonicity allowed by this line search is very important in many applications, for our experiment the monotone version, i.e., setting $K = 1$, yields the best results.
For CP-I, we found that for a given τ it is optimal to choose $\sigma = 1/\tau$. The same holds true for τ_0 and σ_0 in CP-II. Hence, we only state the best τ for CP-I and the optimal τ_0 and γ for CP-II in Table 1.

4.2 Projection onto the Mixed ℓ_∞-Ball

Next we compare the convergence speed of the algorithms for two image denoising problems which can be written in the form (15). First, we consider the Rudin-Osher-Fatemi model for edge-preserving image denoising, cf. [32]. For a noisy function $f : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^2$ the

	$\lvert F(u)-F(u^*)\rvert < 10^{-3}$		$\lvert F(u)-F(u^*)\rvert < 10^{-9}$	
Method	Parameters	Matrix mult.	Parameters	Matrix mult.
PG	$\gamma = 1$	299	$\gamma = 1$	686
C-PG	$n = 19$, $\kappa = 8$	44	$n = 18$, $\kappa = 5$	130
BB-PG (SPG2)	$\gamma_0 = 2$, $K = 1$	54	$\gamma_0 = 5$, $K = 1$	103
FISTA	$L = 1$	76	$L = 1$	340
CP-I	$\tau = 0.19$	60	$\tau = 0.18$	196
CP-II	$\tau_0 = 0.2$, $\gamma = 0.1$	51	$\tau_0 = 0.17$, $\gamma = 0.005$	187

Table 1: Comparison of first-order algorithms to solve the LASSO problem (17). The parameters are hand-tuned and the results averaged over 100 experiments. The stopping criterion is measured in terms of the difference of the function value $F(u) = \frac{1}{2}\|Bu-f\|_2^2$ to the optimal value $F(u^*)$.

Rudin-Osher-Fatemi model describes the denoised image as the solution of

$$\underset{v\in \mathrm{BV}(\Omega)}{\operatorname{argmin}} \{\frac{1}{2}\|v-f\|_{L_2(\Omega)}^2 + \lambda \|v\|_{TV}\}, \tag{18}$$

where $\mathrm{BV}(\Omega)$ is the space of functions of bounded variation and $\|\cdot\|_{TV}$ is the total-variation semi-norm

$$\|v\|_{TV} = \sup\{\int_\Omega v \operatorname{div} g \, dx \,:\, g \in C_0^1(\Omega, \mathbb{R}^2) \text{ and } \sqrt{g_1^2 + g_2^2} \leq 1\}.$$

If v is smooth, it holds that

$$\|v\|_{TV} = \int_\Omega \sqrt{(\partial_x v)^2 + (\partial_y v)^2}\, dxdy. \tag{19}$$

In order to discretize (18), we use the gradient matrix ∇ 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 version of (18)

$$\underset{v\in\mathbb{R}^N}{\operatorname{argmin}} \{\frac{1}{2}\|v-f\|_2^2 + \lambda \| \, |\nabla v| \, \|_1\}, \tag{20}$$

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 (15), i.e.,

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

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

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

In Table 2, we show the number of iterations and runtimes needed by the algorithms to meet two different stopping criteria for $\lambda = 25$ and $\lambda = 50$, respectively. The noisy image we use here is depicted in Figure 2 as well as the denoising result using the regularization parameter $\lambda = 25$. The experiments were performed on a laptop with an Intel Core Duo processor with 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 $\|B\|_2^2 < 8$. In Figure 3, we show the behaviour of C-PG with the two sets of parameters used for $\lambda = 25$, cf. Table 2 (top), for a large numbers of iterations. So, even without the backtracking line search described in Subsection 3.3 the algorithm seems to converge. We observe the same for $\lambda = 50$ and all the other experiments presented in this paper. Hence, Table 2 shows the results of C-PG without a line search. Note that we have tested several BB-PG variants, including those considered in [19, 40], but this did not improve the speed of convergence. Moreover, the backtracking line search of BB-PG (SPG2) did not lead to faster convergence so that we report the performance of standard BB-PG here. Concerning CP-I and CP-II, we found that it is best to choose $\sigma = 1/(8\,\tau)$ and $\sigma_0 = 1/(8\,\tau_0)$. The optimal values for τ, τ_0 and γ are given in Table 2.
We see that our method C-PG outperforms all other algorithms if moderate accuracy is required. If we use a more restrictive stopping criterion, CP-II and PDHG have advantages and for the harder problem with $\lambda = 50$ also FISTA and CP-I are faster. Note that FISTA now performs much better compared to what we have seen in Subsection 4.1 whereas BB-PG is less efficient for this experiment.

Figure 2: Top: Original image of size 256×256 with values in $[0, 255]$ and noisy image (Gaussian noise with standard deviation 25). Bottom: Reconstruction via the Rudin-Osher-Fatemi model (21) and regularization parameter $\lambda = 25$ (left) and model (24) with $\lambda = 15$ (right).

Finally, we consider the following variant of the Rudin-Osher-Fatemi model. We substitute

	$\lambda = 25$					
	$\|v - v^*\|_\infty < 1$			$\|v - v^*\|_\infty < 0.1$		
Method	Parameters	Iterations	Time	Parameters	Iterations	Time
PG	$\gamma = 0.249$	253	1.52	$\gamma = 0.249$	5073	28.74
C-PG	$n = 19$, $\kappa = 11$	41	0.27	$n = 49$, $\kappa = 19$	272	1.62
BB-PG	$\gamma_0 = 6$	86	0.81	$\gamma_0 = 0.8$	1017	9.85
FISTA	$\gamma = 1/8$	64	0.54	$\gamma = 1/8$	279	2.45
CP-I	$\tau = 2.3$	78	0.52	$\tau = 0.6$	287	1.93
CP-II	$\tau = 0.15$, $\gamma = 0.2$	67	0.45	$\tau = 0.28$, $\gamma = 0.44$	221	1.55
PDHG		46	0.28		194	1.14

	$\lambda = 50$					
	$\|v - v^*\|_\infty < 1$			$\|v - v^*\|_\infty < 0.1$		
Method	Parameters	Iterations	Time	Parameters	Iterations	Time
PG	$\gamma = 0.249$	1179	6.61	$\gamma = 0.249$	18596	104.31
C-PG	$n = 37$, $\kappa = 8$	86	0.50	$n = 55$, $\kappa = 12$	829	4.72
BB-PG	$\gamma_0 = 5$	255	2.56	$\gamma_0 = 2$	4289	48.60
FISTA	$\gamma = 1/8$	148	1.29	$\gamma = 1/8$	469	4.03
CP-I	$\tau = 2.4$	118	0.81	$\tau = 0.9$	409	2.69
CP-II	$\tau = 0.1$, $\gamma = 0.28$	102	0.70	$\tau = 0.04$, $\gamma = 0.2$	367	2.48
PDHG		91	0.57		350	2.01

Table 2: Comparison of first-order algorithms to solve the dual Rudin-Osher-Fatemi problem (21) for two different regularization parameters, $\lambda = 25$ (top) and $\lambda = 50$ (bottom). Runtime is given in seconds and as stopping criterion we use the maximal pixel difference to a reference solution (obtained after a large number of FISTA iterations) smaller than 1.0 (left) and 0.1 (right).

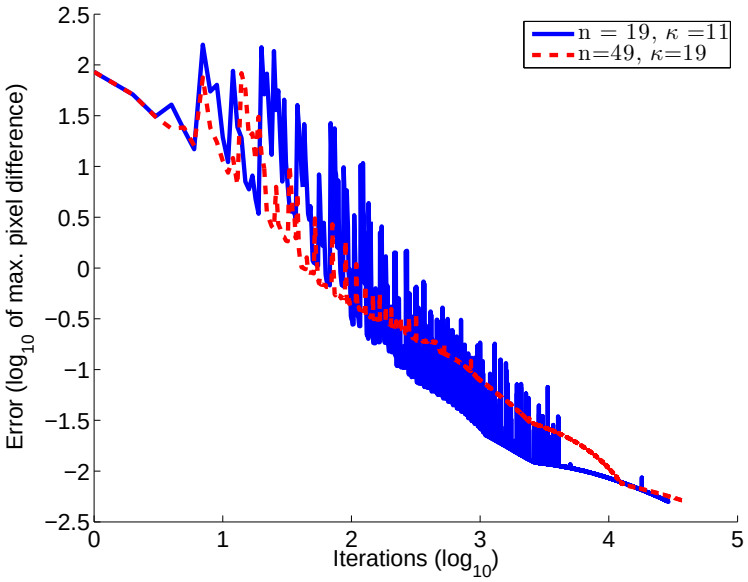

Figure 3: Behaviour of C-PG for large iteration numbers. The dual Rudin-Osher-Fatemi problem (21) with $\lambda = 25$ is solved using C-PG with two different sets of parameters. Shown is the error $\log_{10} \|u^{(k)} - u^*\|_\infty$ as a function of $\log_{10} k$ where $k = sn+i$ is the iteration number.

the norm of the gradient in (19) by the Frobenius norm of the Hessian, cf. [33]. This yields for the case of smooth functions

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

We obtain a discrete version of (22) as follows

$$\operatorname*{argmin}_{v\in\mathbb{R}^N} \{\frac{1}{2}\|v-f\|_2^2 + \lambda \| \, |B^{\mathrm{T}}v| \, \|_1\}, \tag{23}$$

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 (15), i.e.,

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

As before, we can recover a solution v^* of (23) from a solution u^* of (24) via

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

Table 3 shows the performance of the first-order methods for solving (24). We use the regularization parameter $\lambda = 15$ and $\lambda = 30$ and for each case two different stopping criteria. For $\lambda = 15$, the denoised image is depicted in Figure 2. Observe that we have now $\|B\|_2^2 < 64$. The observations we made above for (21) concerning the choice of parameters and the use of the line search also hold true for this experiment. Note that 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 either.
Our method C-PG is now the fastest method in three of the four test settings. Furthermore, we notice a clearer advantage of C-PG over the other methods than for the case $B = \nabla^{\mathrm{T}}$. Only in the hardest case where we use $\lambda = 30$ and the strict stopping criterion $\|v-v^*\|_\infty < 0.1$ it is outperformed by FISTA, CP-I and CP-II.

High-accuray case: In the experiments presented in Table 2 and Table 3, we restrict our attention to relatively modest accuracy requirements. These stopping criteria are appropriate for the denoising problem presented here as well as many other image processing tasks since a higher precision will not lead to a visually different result. For high-accuracy settings, it turns out that our step-length strategy does not perform very well. If we solve (21) with $\lambda = 25$ and stopping criterion $\|v - v^*\|_\infty < 0.001$, e.g., the fastest method is PDHG which needs 678 iterations (5.08 sec.) whereas C-PG with 84983 iteration (504.55 sec) has a similar runtime than the standard projected gradient method (PG).

	$\lambda = 15$					
	$\|v - v^*\|_\infty < 1$			$\|v - v^*\|_\infty < 0.1$		
Method	Parameters	Iterations	Time	Parameters	Iterations	Time
PG	$\gamma = 0.249$	511	6.61	$\gamma = 0.249$	4544	61.69
C-PG	$n = 19$, $\kappa = 11$	58	0.77	$n = 49$, $\kappa = 19$	241	3.29
BB-PG	$\gamma_0 = 2$	86	0.81	$\gamma_0 = 1.5$	1017	9.85
FISTA	$\gamma = 1/8$	96	2.15	$\gamma = 0.125$	319	7.29
CP-I	$\tau = 2.3$	103	1.45	$\tau = 0.6$	349	5.09
CP-II	$\tau = 0.15$, $\gamma = 0.2$	94	1.33	$\tau = 0.28$, $\gamma = 0.44$	249	4.08

	$\lambda = 30$					
	$\|v - v^*\|_\infty < 1$			$\|v - v^*\|_\infty < 0.1$		
Method	Parameters	Iterations	Time	Parameters	Iterations	Time
PG	$\gamma = 0.0312$	3191	43.14	$\gamma = 0.0312$	27680	377.31
C-PG	$n = 55$, $\kappa = 21$	156	2.12	$n = 59$, $\kappa = 11$	1058	13.99
BB-PG	$\gamma_0 = 0.3$	752	19.78	$n = 19$, $\gamma_0 = 0.3$	6642	176.56
FISTA	$\gamma = 1/8$	225	5.14	$\gamma = 1/8$	684	15.21
CP-I	$\tau = 0.02$	255	3.75	$\tau = 0.01$	775	10.81
CP-II	$\tau = 0.09$, $\gamma = 0.3$	224	3.39	$\tau = 0.1$, $\gamma = 0.4$	600	10.61

Table 3: Comparison of first-order algorithms to solve problem (24) using the regularization parameters $\lambda = 15$ (top) and $\lambda = 30$ (bottom). Stopping criterion: maximal pixel difference to a reference solution smaller than 1 (left) and smaller than 0.1 (right).

5 Conclusions

We introduced a projected gradient algorithm which uses a step length strategy based on so-called superstep cycles. The performance of the algorithm was tested for the LASSO problem and two version of the Rudin-Osher-Fatemi model which is popular in image processing. These numerical experiments show that our method is competitive to recent first-order optimization algorithms. Convergence can be guaranteed by applying a nonmonotone backtracking line search. Experimentally, however, convergence was observed even without this line search. Although the proof of this observation remains an open problem, we show for a simple case in two dimensions that the corresponding operator which is applied in each step of the algorithm is nonexpansive.

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.

[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] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

[8] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000.

[9] E. G. Birgin, J. M. Martínez, and M. Raydan. Algorithm 813 – Software for convex-constrained optimization. *ACM Transactions on Mathematical Software*, 27:340–349, 2001.

[10] E. G. Birgin, J. M. Martínez, and M. Raydan. Inexact spectral projected gradient methods on convex sets. *IMA Journal on Numerical Analysis*, 23:539–559, 2003.

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

[12] S. Bonettini and V. Ruggiero. On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration. *Journal of Mathematical Imaging and Vision*, 44(3):236–253, 2012.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

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

[26] 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.

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

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

[29] 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.

[30] Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

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

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

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

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

[35] 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.

[36] 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.

[37] D. Young. On Richardson's method for solving linear systems with positive definite matrices. *Journal of Mathematical Physics*, 32:243–255, 1954.

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

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

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

[41] 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.