Industrial and
Systems Engineering

Accelerated Sparse Recovery via Gradient Descent with Nonlinear Conjugate Gradient Momentum

Mengqi Hu[1], Yifei Lou[2], Bao Wang[3], Ming Yan[4], Xiu Yang[1], and Qiang Ye[5]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

[2]Department of Mathematical Sciences, The University of Texas at Dallas 800 W. Campbell Rd, Richardson, 75080, TX, USA

[3]Department of Mathematics and Scientific Computing and Imaging Institute, The University of Utah 72 Central Campus Dr, Salt Lake City, 84102, Utah, USA

[4]Department of Computational Mathematics, Science and Engineering and Department of Mathematics, Michigan State University 428 South Shaw Lane, East Lansing, 48824, MI, USA

[5]Department of Mathematics, University of Kentucky, Lexington, 40513, Kentucky, USA

ISE Technical Report 22T-011

LEHIGH UNIVERSITY

Noname manuscript No.
(will be inserted by the editor)

Accelerated Sparse Recovery via Gradient Descent with Nonlinear Conjugate Gradient Momentum

Mengqi Hu · Yifei Lou · Bao Wang · Ming Yan · Xiu Yang · Qiang Ye

Received: date / Accepted: date

Abstract This paper applies an idea of adaptive momentum for the nonlinear conjugate gradient to accelerate optimization problems in sparse recovery. Specifically, we consider two types of minimization problems: a (single) differentiable function and the sum of a non-smooth function and a differentiable function. In the first case, we adopt a fixed step size to avoid the traditional line search and establish the convergence analysis of the proposed algorithm for a quadratic problem. This acceleration is further incorporated with an operator

M. Hu
Department of Industrial and Systems Engineering, Lehigh University
200 West Packer Avenue, Bethlehem, 18015, PA, USA
E-mail: meh621@lehigh.edu

Y. Lou
Department of Mathematical Sciences, The University of Texas at Dallas
800 W. Campbell Rd, Richardson, 75080, TX, USA
E-mail: yifei.lou@utdallas.edu

B. Wang
Department of Mathematics and Scientific Computing and Imaging Institute, The University of Utah
72 Central Campus Dr, Salt Lake City, 84102, Utah, USA
E-mail: bwang@sci.utah.edu

M. Yan
Department of Computational Mathematics, Science and Engineering and Department of Mathematics, Michigan State University
428 South Shaw Lane, East Lansing, 48824, MI, USA
E-mail: myan@msu.edu

X. Yang
Department of Industrial and Systems Engineering, Lehigh University
200 West Packer Avenue, Bethlehem, 18015, PA, USA
E-mail: xiy518@lehigh.edu

Q. Ye
Department of Mathematics, University of Kentucky, Lexington, 40513, Kentucky, USA
E-mail: qye3@uky.edu

splitting technique to deal with the non-smooth function in the second case. We use the convex ℓ_1 and the nonconvex $\ell_1 - \ell_2$ functionals as two case studies to demonstrate the efficiency of the proposed approaches over traditional methods.

Keywords Accelerated gradient momentum · operator splitting · fixed step size · convergence rate

Mathematics Subject Classification (2020) MSC 41A25 · MSC 41A25 · MSC 65K10

Declarations The MATLAB codes and datasets generated during and/or analysed during the current study will be available under `https://sites.google.com/site/louyifei/Software` after publication.

1 Introduction

Traditional methods for reconstructing signals from measured data follow the well-known Nyquist-Shannon sampling theorem [60], which guarantees the exact recovery if the sampling rate is at least twice the highest frequency of the underlying signal. Similarly, the fundamental theorem of linear algebra suggests that the number of linear measurements of a discrete finite-dimensional signal should be at least as large as its ambient dimension to ensure a stable reconstruction. Nyquist–Shannon theorem serves as the underlying principle of most devices [6] such as analog-to-digital conversion, medical imaging, and video processors, but it is a sufficient condition for the exact recovery of any signal that requires an overly large number of measurements to be collected.

To acquire and process data more economically, the paradigm of compressive sensing (CS) [15]—also known as compressed sensing, or compressive sampling—provides a fundamentally new approach that reconstructs certain signals from what was believed in the past to be highly incomplete measurements (information). CS relies on an empirical observation that most signals can be well approximated by a sparse expansion under a properly chosen basis, that is, by only a small number of non-zero coefficients. The number of non-zero entries of a vector $\mathbf{x} \in \mathbb{R}^N$ is denoted by $\|\mathbf{x}\|_0$. Note that $\|\cdot\|_0$ is named the "ℓ_0 norm" in [15], although it is not even a semi-norm. The vector $\mathbf{x}$ is called *s-sparse* if $\|\mathbf{x}\|_0 \leq s$, and it is considered a sparse vector if $s \ll N$. Note that few practical systems are truly sparse through direct observations, but rather *compressible*, i.e., only a few entries contribute significantly to its ℓ_1 norm under certain transformations.

For simplicity, we assume linear measurements; otherwise one can always linearize the data collection process. Consequently, we consider a data vector $\mathbf{b} \in \mathbb{R}^M$ obtained by

$$\mathbf{b} = A\mathbf{x} + \mathbf{n}, \tag{1}$$

where $A \in \mathbb{R}^{M \times N}$ is called a sensing matrix, $\mathbf{x} \in \mathbb{R}^N$ is an underlying signal to be recovered, and $\mathbf{n} \in \mathbb{R}^M$ is the noise term. To find a sparse vector $\mathbf{x}$ from (1), one formulates an unconstrained minimization problem,

$$\hat{\mathbf{x}}_0 = \arg\min_{\mathbf{x}} \lambda \|\mathbf{x}\|_0 + \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2, \quad (2)$$

where λ is a positive parameter to be tuned such that $\|A\hat{\mathbf{x}}_0 - \mathbf{b}\|_2 \le \epsilon$ for a pre-set error tolerance ϵ that often corresponds to the noise level $\|\mathbf{n}\|_2$. As the ℓ_0 minimization (2) is NP-hard [40], one replaces it by the convex ℓ_1 norm, i.e.,

$$\hat{\mathbf{x}}_1 = \arg\min_{\mathbf{x}} \lambda \|\mathbf{x}\|_1 + \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2. \quad (3)$$

In this paper, we consider a general formulation for sparse recovery

$$\min_{\mathbf{x}} \lambda f(\mathbf{x}) + g(\mathbf{x}), \quad (4)$$

where $f(\cdot)$ is a regularization term and $g(\cdot)$ is a (convex and differentiable) data fidelity term, e.g., $g(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$. We assume that f is a continuous (possibly non-differentiable) function that can enhance the sparsity of $\mathbf{x}$. For instance, the non-convex metric ℓ_p for $p \in (0,1)$ can be viewed as a continuation effort to approximate ℓ_0 as $p \to 0$. Another regularization that achieves a continuation from ℓ_0 to ℓ_1 is the error function (ERF) [22] by changing its internal parameter. Some non-convex regularizations derived from ℓ_1 include capped ℓ_1 [69,56,36], transformed ℓ_1 (TL1) [38,67,68,21], and sorted ℓ_1 [30]. A combination of different norms can also be served as a sparsity promoting sparsity, e.g., $\ell_1 - \ell_2$ [65,35,34] and ℓ_1/ℓ_2 [52,62]. To the best of our knowledge, only $\ell_1 - \ell_2$ and TL1 have the exact sparse recovery guarantees based on the RIP type of conditions [66,68], which are actually more strict compared to the one for the ℓ_1 model. As these RIP conditions are sufficient and unverifiable, many works reported the empirical advantages of non-convex regularizations over the convex ℓ_1 approach in promoting sparsity.

As $f(\cdot)$ is non-differentiable, gradient-based optimization methods can not be directly applied to minimize (4), not to mention some acceleration techniques by adaptive momentum [50,49,29,17]. One remedy involves a smooth approximation of f such as using the Huber function [31,27,58] to approximate the ℓ_1 norm. In general, several papers reported using smoothing to approximate non-smooth functions to improve the performance of non-linear conjugate algorithms [10,64,45,39]. Another alternative is based on operator splitting to deal with the non-smooth term $f(\cdot)$ and the smooth function $g(\cdot)$ separately, for example, forward-backward splitting (FBS) [11], the alternative direction method of multipliers (ADMM) [7], and iteratively reweighted L_1 [8, 37].

We propose to combine the operator splitting with the momentum acceleration. In particular, we incorporate the momentum update in the gradient descent when minimizing the data fitting term $g(\cdot)$ for speed-up, while relying on proximal operators [46] to deal with the non-differentiable function $f(\cdot)$. Starting by $f(\cdot) = \emptyset$, i.e., minimizing a single differentiable function $g(\cdot)$, we promote the choice of fixed step size in the momentum-based gradient descent algorithm and analyze its convergence rate for a quadratic problem. To deal with the non-smooth function $f(\cdot)$, we further adopt a splitting technique and consider two case studies when the proximal operator according to $f(\cdot)$ has a closed-form solution. We conduct experiments on a quadratic problem, ℓ_1 and $\ell_1 - \ell_2$ minimization problems to compare among different momentum update formulas and showcase the speed-up of the proposed approach with simple implementation over the traditional gradient-based approaches.

The remaining of this paper is organized as follows. Section 2 examines the case of minimizing a single differentiable function. In particular, we advocate a constant step size and prove the convergence for a quadratic problem. The proposed marriage of FBS and momentum acceleration is discussed in Section 3 with experiments on two case studies of ℓ_1 and $\ell_1 - \ell_2$ regularizations, showing the faster convergence of the proposed method than existing approaches. Finally, conclusions and future works are presented in Section 4.

2 Minimizing a single function

We review in Section 2.1 gradient-based algorithms that minimize a single function, including gradient descent, conjugate gradient, and adaptive momentum methods. We propose to combine the Fletcher-Reeve moment and gradient descent with a fixed step size in Section 2.2. The convergence of the proposed scheme can be established for a quadratic problem. Lastly, experimental comparison is presented in Section 2.3.

2.1 Literature review

Gradient descent is a class of first-order iterative optimization algorithms for finding a local minimum of a differentiable function. This type of algorithms involves repeated moving along the opposite direction of the gradient of the objective function at the current point, since it is the direction where function value decreases at the fastest rate. Given a differentiable function $g(\cdot)$, a general form of gradient descent (GD) that minimizes $g(\mathbf{x})$ can be described as,

$$\begin{cases} \mathbf{p}^{(l+1)} & = -\nabla g(\mathbf{x}^{(l)}) \\ \mathbf{x}^{(l+1)} & = \mathbf{x}^{(l)} + \alpha^{(l+1)}\mathbf{p}^{(l+1)}, \end{cases} \tag{5}$$

where l indexes the iteration number and $\alpha^{(l+1)} > 0$ is a step size that can be fixed or updated iteratively. There are many variations of GD depending on how the step size is determined and/or the descending direction is chosen. For example, steepest descent (SD) is perhaps one of the simplest variations, which goes as follows,

$$\begin{cases} \mathbf{p}^{(l+1)} & = -\nabla g(\mathbf{x}^{(l)}) \\ \alpha^{(l+1)} & = \arg\min_{\alpha} g(\mathbf{x}^{(l)} + \alpha\mathbf{p}^{(l+1)}) \\ \mathbf{x}^{(l+1)} & = \mathbf{x}^{(l)} + \alpha^{(l+1)}\mathbf{p}^{(l+1)}. \end{cases} \tag{6}$$

In each iteration, SD performs an exact line search to achieve the maximum descent along the gradient direction, i.e., the descent is the steepest. However, empirically it does not work well in most cases, since such a local descending property does not necessarily coincide with the overall descending of the original function.

Notice that the search direction in each iteration of (6) only utilizes the information at the current step $\mathbf{x}^{(l)}$ without any information from previous iterations. Adding them back leads to momentum-based algorithms, which are also called as heavy ball algorithms [50]. The term "momentum" is an analogy of a heavy ball sliding on the surface of values of the function being minimized when the update of each step is memorized in the process. To this end, we refer the following iteration

$$\begin{cases} \mathbf{p}^{(l+1)} & = -\nabla g(\mathbf{x}^{(l)}) + \beta^{(l+1)}\mathbf{p}^{(l)} \\ \mathbf{x}^{(l+1)} & = \mathbf{x}^{(l)} + \alpha^{(l+1)}\mathbf{p}^{(l+1)}, \end{cases} \tag{7}$$

as gradient descent with momentum (GDM). Both $\alpha^{(l+1)}$ and $\beta^{(l+1)}$ in (7) can be fixed or adaptively chosen according to a certain scheme. For instance, if we update $\alpha^{(l+1)}$ in the same way as SD (6), the corresponding algorithm

$$\begin{cases} \beta^{(l+1)} & = \frac{\|\nabla g(\mathbf{x}^{(l)})\|^2}{\|\nabla g(\mathbf{x}^{(l-1)})\|^2} \\ \mathbf{p}^{(l+1)} & = -\nabla g\left(\mathbf{x}^{(l)}\right) + \beta^{(l+1)}\mathbf{p}^{(l)} \\ \alpha^{(l+1)} & = \arg\min_{\alpha} g(\mathbf{x}^{(l)} + \alpha\mathbf{p}^{(l+1)}) \\ \mathbf{x}^{(l+1)} & = \mathbf{x}^{(l)} + \alpha^{(l+1)}\mathbf{p}^{(l+1)}, \end{cases} \tag{8}$$

is identical to the classic nonlinear conjugate gradient (CG). The β update for momentum coefficient is called Fletcher-Reeves (FR) momentum in nonlinear conjugate gradient algorithms [29, 17]. In addition to FR, other popular momentum updates include

– Polak-Ribière (PR) [49]

$$\beta_{PR}^{(l+1)} = \frac{\langle \nabla g(\mathbf{x}^{(l)}), \nabla g(\mathbf{x}^{(l)}) - \nabla g(\mathbf{x}^{(l-1)}) \rangle}{\|\nabla g(\mathbf{x}^{(l-1)})\|^2}; \tag{9}$$

– Hestenes-Stiefel (HS) [28]

$$\beta_{HS}^{(l+1)} = \frac{\langle \nabla g(\mathbf{x}^{(l)}), \nabla g(\mathbf{x}^{(l)}) - \nabla g(\mathbf{x}^{(l-1)}) \rangle}{-\langle \mathbf{x}^{(l)}, \nabla g(\mathbf{x}^{(l)}) - \nabla g(\mathbf{x}^{(l-1)}) \rangle}; \tag{10}$$

– Dai-Yuan (DY) [12]

$$\beta_{DY}^{(l+1)} = \frac{\|\nabla g(\mathbf{x}^{(l)})\|^2}{-\langle \mathbf{x}^{(l)}, \nabla g(\mathbf{x}^{(l)}) - \nabla g(\mathbf{x}^{(l-1)}) \rangle}. \tag{11}$$

Another type of momentum-based algorithms was developed by Yurii Nesterov [42,43]. Starting from $t^{(0)} = 1$, Nesterov's accelerated gradient (NAG) is expressed as,

$$\begin{cases} t^{(l+1)} &= \frac{1+\sqrt{4(t^{(l)})^2+1}}{2}, \\ \mathbf{p}^{(l+1)} &= -\nabla g\left(\mathbf{x}^{(l)}\right), \\ \mathbf{y}^{(l+1)} &= \mathbf{x}^{(l)} + \alpha^{(l+1)}\mathbf{p}^{(l+1)}, \\ \mathbf{x}^{(l+1)} &= \mathbf{y}^{(l+1)} + \frac{t^{(l)}-1}{t^{(l+1)}}(\mathbf{y}^{(l+1)} - \mathbf{y}^{(l)}). \end{cases} \tag{12}$$

Similarly to other gradient-based algorithms, the step size $\alpha^{(l+1)}$ in NAG can be fixed or updated during the iteration. For convex function $g(\cdot)$, NAG achieves a convergence rate of $O(\frac{1}{l^2})$, as opposed to $O(\frac{1}{l})$ obtained by standard gradient-based methods. This momentum scheme can be further accelerated by a proper restart with provable guarantees in certain circumstances [41,19,57,54].

2.2 The proposed approach

Exact line search is not necessarily optimal, as the gradient descent direction may not be a good search direction. As a result, the steepest descent algorithm (6) bounces back and forth in the valley formed by the objective function rather than down the valley. Similar conclusion can be drawn for certain momentum based algorithms. As pointed out in [51,24], exact line search would lead to a very small step size in such a way that two consecutive iterates do not vary too much; this phenomenon is called *jamming*. To avoid jamming, one can use a hybrid momentum scheme [13,2,44] or an inexact line search [23,53]. Instead of exact search as used in SD, an inexact search refers to finding a step size α that satisfies the Wolfe conditions [1,12,18,23], i.e.,

$$\begin{cases} g(\mathbf{x}^{(l)} + \alpha^{(l)}\mathbf{p}^{(l)}) \leqslant g(\mathbf{x}^{(l)}) + c_1\langle \alpha^{(l)}\mathbf{p}^{(l)}, \nabla g(\mathbf{x}^{(l)})\rangle, \\ \langle -\mathbf{p}^{(l)}, \nabla g(\mathbf{x}^{(l)} + \alpha^{(l)}\mathbf{p}^{(l)})\rangle \leqslant c_2\langle -\mathbf{p}^{(l)}, \nabla g(\mathbf{x}^{(l)})\rangle, \end{cases} \tag{13}$$

for two constants $0 < c_1 < c_2 < 1$. The first equation of (13) is also called Armijo-Goldenstein condition [3,5], which is usually used in back-tracking step sizes. These conditions play an important role in establishing a descent property and global convergence of conjugate descent.

Instead of designing a update scheme for α, we consider a fixed step size α in the gradient descent that is combined with the FR moment, i.e.,

$$\begin{cases} \beta^{(l+1)} &= \frac{\|\nabla g(\mathbf{x}^{(l)})\|^2}{\|\nabla g(\mathbf{x}^{(l-1)})\|^2} \\ \mathbf{p}^{(l+1)} &= -\nabla g\left(\mathbf{x}^{(l)}\right) + \beta^{(l+1)}\mathbf{p}^{(l)} \\ \mathbf{x}^{(l+1)} &= \mathbf{x}^{(l)} + \alpha\mathbf{p}^{(l+1)}, \end{cases} \tag{14}$$

which is referred to as FR gradient descent (FRGD). By fixing α, most properties used in the convergence analysis of conjugate gradient no longer hold. Fortunately, we can borrow a technique used in an inexact conjugate gradient (due to round off errors) or inexact preconditioning [20, 59] to analyze the convergence of FRGD (14). Theorem 1 characterizes the convergence analysis of the proposed FRGD for a quadratic problem,

$$\min_{\mathbf{x}} g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\mathsf{T} A\mathbf{x} + \mathbf{x}^\mathsf{T}\mathbf{b}, \tag{15}$$

where A is a strictly symmetric positive definite matrix. To this end, we define the condition number of A as $\kappa(A) = |\lambda_{\max}(A)/\lambda_{\min}(A)|$, i.e., the ratio between the largest and smallest eigenvalues.

Theorem 1 *Suppose* $\{\mathbf{x}^{(l)}, \mathbf{p}^{(l)}\}$ *be generated by* (14) *with a fixed step size* α *when minimizing* (15). *Let* $\mathbf{r}^{(l)} = \nabla g(\mathbf{x}^{(l)})$, $\rho = \max_{0\leqslant j\leqslant i\leqslant l-1} \|\mathbf{r}^{(i)}\|_2/\|\mathbf{r}^{(j)}\|_2$, $\mathbf{z}^{(l)} = \mathbf{r}^{(l)}/\|\mathbf{r}^{(l)}\|_2$ *and* $Z^{(l)} = [\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(l-1)}]$. *If* $\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(l)}$ *are linearly independent, then there exists a constant*

$$K_l \leqslant l(1 + \frac{l\rho}{2})\|A\|_2\kappa(Z^{(l+1)}), \tag{16}$$

such that

$$\|\mathbf{r}^{(l)}\|_2 \leqslant 2(1+K_l)\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^l \|\mathbf{r}^{(0)}\|_2. \tag{17}$$

Proof Denote $R^{(l)} = [\mathbf{r}^{(0)}, \cdots, \mathbf{r}^{(l-1)}]$ and $D^{(l)} = \mathrm{diag}\{\|\mathbf{r}^{(0)}\|_2, \cdots, \|\mathbf{r}^{(l-1)}\|_2\}$, then $Z^{(l)} = R^{(l)}(D^{(l)})^{-1}$. We further denote $P^{(l)} = [\mathbf{p}^{(0)}, \cdots, \mathbf{p}^{(l-1)}]$. It follows from $\mathbf{r}^{(l+1)} = \mathbf{r}^{(l)} - \alpha AP^{(l)}$ that

$$\begin{aligned}\alpha AP^{(l)} &= [\mathbf{r}^{(0)} - \mathbf{r}^{(1)}, \cdots, \mathbf{r}^{(l-1)} - \mathbf{r}^{(l)}]\\ &= R^{(l)}L^{(l)} - \mathbf{r}^{(l)}\mathbf{e}_{(l)}^\mathsf{T},\\ &= \mathbf{z}^{(l)}D^{(l)}L^{(l)} - \mathbf{r}^{(l)}\mathbf{e}_{(l)}^\mathsf{T},\end{aligned} \tag{18}$$

where $\mathbf{e}_{(l)} = [0, \cdots, 0, 1]^\mathsf{T}$ and $L^{(l)}$ is the $l \times l$ lower bidiagonal matrix with 1 on the diagonal and -1 on the subdiagonal. Similarly using the $\mathbf{p}$ update of $\mathbf{p}^{(l)} = \mathbf{r}^{(l)} + \beta^{(l)}\mathbf{p}^{(l-1)}$, we have

$$Z^{(l)} = R^{(l)}(D^{(l)})^{-1} = P^{(l)}U^{(l)}(D^{(l)})^{-1}, \tag{19}$$

where $U^{(l)}$ is the $l \times l$ upper bidiagonal matrix with 1 on the diagonal and $-\beta^{(1)}$, $\cdots$, $-\beta^{(l-1)}$ on the subdiagonal. Combining (18) (19), we obtain

$$AZ^{(l)} = Z^{(l)}T^{(l)} - \frac{\mathbf{r}^{(l)}\mathbf{e}_{(l)}^\mathsf{T}}{\hat{\alpha}\|\mathbf{r}^{(0)}\|}, \tag{20}$$

where $T^{(l)} = \frac{1}{\alpha}D^{(l)}L^{(l)}U^{(l)}(D^{(l)})^{-1}$ and $\hat{\alpha} = \alpha\|\mathbf{r}^{(l-1)}/\|\mathbf{r}^{(0)}\|$. It is straightforward to verify that $\hat{\alpha} = \mathbf{e}_{(l)}^\mathsf{T}(T^{(l)})^{-1}\mathbf{e}_{(1)}$. By [59, Theorem 3.5] we have

$$\|\mathbf{r}^{(l)}\|_2 \leqslant (1+K_l)\min_{p\in\mathcal{P}_l, p(0)=1}\|p(A)\mathbf{r}^{(0)}\|_2, \tag{21}$$

where $K_l = \|AZ^{(l)}T^{(l)}[I_{(l)}, 0]Z_{\dagger}^{(l+1)}\|_2 \leqslant \|A\|_2\|T^{(l)}\|_2\|Z^{(l)}\|_2\|Z_{\dagger}^{(l+1)}\|_2$, $Z_{\dagger}^{(l+1)}$ is the pseudo-inverse of $Z^{(l+1)}$, and $\mathcal{P}_l$ is the space of polynomials of degree l. By definition $\beta^{(l)} = \|\mathbf{r}^{(l)}\|_2^2/\|\mathbf{r}^{(l-1)}\|_2^2$, we can rewrite

$$T^{(l)} = \frac{1}{\alpha}D^{(l)}L^{(l)}(D^{(l)})^{-1}D^{(l)}U^{(l)}(D^{(l)})^{-1} = \frac{1}{\alpha}\tilde{L}^{(l)}\tilde{U}^{(l)}, \tag{22}$$

where

$$\begin{aligned}\tilde{L}^{(l)} &= D^{(l)}L^{(l)}(D^{(l)})^{-1} \\ &= \begin{bmatrix} 1 & & & & \\ -\frac{\|\mathbf{r}^{(1)}\|}{\|\mathbf{r}^{(0)}\|} & 1 & & & \\ & -\frac{\|\mathbf{r}^{(2)}\|}{\|\mathbf{r}^{(1)}\|} & 1 & & \\ & & \ddots & \ddots & \\ & & & -\frac{\|\mathbf{r}^{(l-1)}\|}{\|\mathbf{r}^{(l-2)}\|} & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & & & & \\ -\sqrt{\beta^{(1)}} & 1 & & & \\ & -\sqrt{\beta^{(2)}} & 1 & & \\ & & \ddots & \ddots & \\ & & & -\sqrt{\beta^{(l-1)}} & 1 \end{bmatrix},\end{aligned} \tag{23}$$

and

$$\begin{aligned}\tilde{U}^{(l)} &= D^{(l)}U^{(l)}(D^{(l)})^{-1} \\ &= \begin{bmatrix} 1 & -\beta^{(1)}\frac{\|\mathbf{r}^{(0)}\|}{\|\mathbf{r}^{(1)}\|} & & & \\ & 1 & -\beta^{(2)}\frac{\|\mathbf{r}^{(1)}}{\mathbf{r}^{(2)}\|} & & \\ & & \ddots & \ddots & \\ & & & 1 & -\beta^{(l-1)}\frac{\|\mathbf{r}^{(l-2)}\|}{\|\mathbf{r}^{(l-1)}\|} \\ & & & & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -\sqrt{\beta^{(1)}} & & & \\ & 1 & -\sqrt{\beta^{(2)}} & & \\ & & \ddots & \ddots & \\ & & & 1 & -\sqrt{\beta^{(l-1)}} \\ & & & & 1 \end{bmatrix} = (\tilde{L}^{(l)})^{\mathsf{T}}.\end{aligned} \tag{24}$$

We can see that $(\tilde{L}^{(l)})^{-1}$ is a lower triangular matrix with 1 on its diagonal and (i,j)-th entry be $\sqrt{\beta^{(j)}\beta^{(j+1)}\cdots\beta^{(i)}} = \|\mathbf{r}^{(i-1)}\|_2/\|\mathbf{r}^{(j-1)}\|_2$ for all $i > j$. Let ρ be an upper bound of $\|\mathbf{r}^{(i-1)}\|_2/\|\mathbf{r}^{(j-1)}\|_2$, then we have $\|(\tilde{L}^{(l)})^{-1}\|_F^2 \leqslant l + l(l-1)\rho/2$ and therefore

$$\|(T^{(l)})^{-1}\|_2 = \alpha\|(\tilde{L}^{(l)}(\tilde{L}^{(l)})^{\mathsf{T}})^{-1}\|_2 = \alpha\|(\tilde{L}^{(l)})^{-1}\|_2^2 \tag{25}$$

$$\leqslant \alpha\|(\tilde{L}^{(l)})^{-1}\|_F^2 \leqslant \alpha l(1 + l\rho/2). \tag{26}$$

Combining with the fact $\|Z^{(l)}\|_2 \|Z_{\dagger}^{(l+1)}\|_2 \leqslant \|Z^{(l+1)}\|_2 \|Z^{(l+1)}\|_2 = \kappa(U^{(l+1)})$ yields $K_l \leqslant l\alpha(1+l\rho/2)\|A\|_2 \kappa(\mathbf{Z}^{(l+1)})$. Finally it follows the standard conjugate gradient convergence bound [55] that gives

$$\begin{aligned}\min_{p \in \mathcal{P}_l, p(0)=1} \|p(A)\mathbf{r}^{(0)}\|_2 &\leqslant \min_{p \in \mathcal{P}_l, p(0)=1} \max_i |p(\lambda_i)| \|\mathbf{r}^{(0)}\|_2 \\ &\leqslant 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^l \|\mathbf{r}^{(0)}\|_2,\end{aligned}$$

where λ_i are the eigenvalues of matrix A and the result follows. $\square$

We see that this convergence rate (17) is similar to the one of the classic conjugate gradient algorithm, which is given by

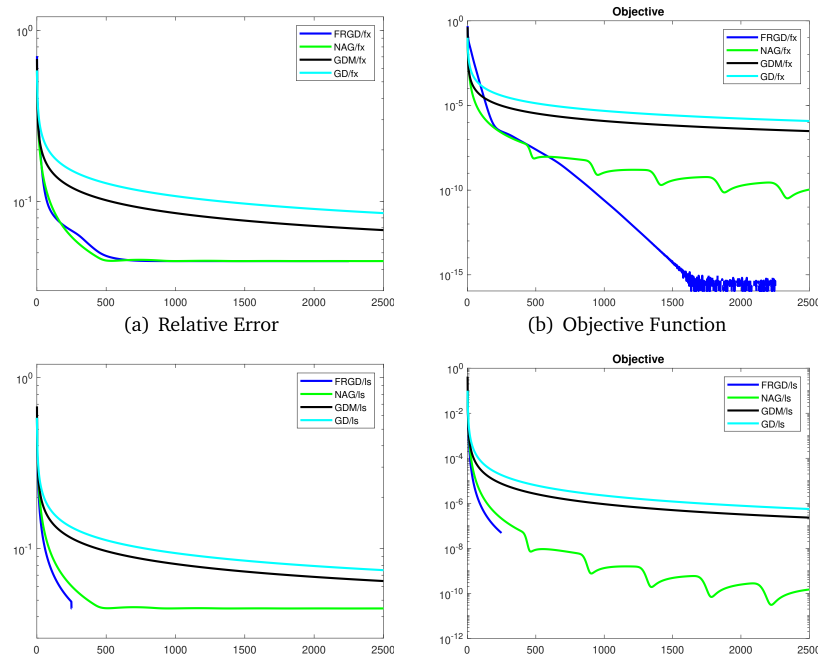
$$\|\mathbf{r}^{(l)}\|_2 \leqslant 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^l \|\mathbf{r}^{(0)}\|_2. \tag{27}$$

The difference between (17) (27) lies in the additional term of $1+K_l$ introduced by the fixed step size. Similar to other accelerated gradient schemes and Krylov subspace methods, this convergence rate in (17) (27) is achieved only when the current position is not in the neighborhood of a stationary point. As the iteration increases, especially when the number of iteration exceeds to the dimension of the problem, the linear independence of the columns in the matrix Z is no longer guaranteed, and hence the acceleration no longer works.

2.3 Experimental results

Following the work of [26], we consider the quadratic problem (15) with $A \in \mathbb{R}^{500\times 500}$ being the Laplacian matrix of a circular graph and $\mathbf{b} \in \mathbb{R}^{\mathbf{500}}$ being a vector whose first entry is 1 and the remaining entries are 0. It is straightforward to verify that $g(\mathbf{x})$ is convex with Lipschitz constant 4. We compare GD (5), GDM (7) with a fixed value of $\beta = 0.9$, NAG (12), and FRGD (14) in terms of relative errors to the ground truth and objective decay. For each competing method, we consider two ways to choose $\alpha^{(l+1)}$: a fixed value of 0.3 and an adaptive update via line search (6), indicated by “fx” and “ls” respectively. For example, FRGD/fx refers to FRGD method with a fixed value, and FRGD/ls refers to FRGD with updating $\alpha^{(l+1)}$ by an exact line search. Note that FRGD/ls is equivalent to the conjugate gradient for any quadratic problem.

We plot the relative errors and the objective functions with respect to iteration numbers of all the competing methods in Figure. 1. All the plots are in a logarithmic scale. As expected, GDM yields slightly better performance than SD/GD, while NAG converges significantly faster than GDM, but in an oscillatory manner. It is worth noting in Figure. 1 (b) that the objective values of GD/fx, GDM/fx, NAG/fx (with a fixed step size) become stagnant after 500 iterations, whereas FRGD/fx continues to decay until the machine accuracy. When the step size is adaptive, FRGD/ls reduces to the classic conjugate



(a) Relative Error (b) Objective Function

(c) Relative Error (d) Objective Function

Fig. 1 Comparison of gradient based methods on a quadratic problem with a fixed step size (top) and an adaptive step size by linear search (bottom).

gradient that quickly falls into a local minimum, while all the other algorithms require more iterations to converge. In summary, FRGD converges at a rate much faster than regular GD and its variants.

3 Minimizing the sum of two functions

In this section, we focus on minimizing the sum of two functions defined in (4). Specifically, we consider two different functions of f: the ℓ_1 norm $\|\mathbf{x}\|_1$ and the $\ell_1 - \ell_2$ regularization $\|\mathbf{x}\|_1 - \|\mathbf{x}\|_2$, to promote the sparsity of the vector $\mathbf{x}$. As f is not differentiable, we adopt the subdifferential for a general (not necessary convex) function, defined as

$$\partial f(\mathbf{x}) = \{\mathbf{p} | f(\mathbf{z}) \geqslant f(\mathbf{x}) + \mathbf{p}^{\mathsf{T}}(\mathbf{z} - \mathbf{x}) + o(\|\mathbf{z} - \mathbf{x}\|)\}, \tag{28}$$

instead of the standard gradient ∇f. We discretize the gradient flow

$$\frac{d}{dt}\mathbf{x}(t) \in -\lambda \partial f(\mathbf{x}(t)) - \nabla g(\mathbf{x}(t)), \tag{29}$$

that minimizes (4) by a semi-implicit scheme as follows,

$$\frac{\mathbf{x}^{(l+1)} - \mathbf{x}^{(l)}}{\delta} \in -\lambda \partial f(\mathbf{x}^{(l+1)}) - \nabla g(\mathbf{x}^{(l)}), \tag{30}$$

where $\delta > 0$ is a step size. The iteration of (30) is often referred to as forward-backward splitting [11], as one uses a forward solution in ∇g and a backward one in ∂f. After rearranging (30), we obtain

$$\mathbf{x}^{(l+1)} \in \big(I + \delta\lambda\partial f\big)^{-1}\big(\mathbf{x}^{(l)} - \nabla g(\mathbf{x}^{(l)})\big),$$

which implies that $\mathbf{x}^{(l+1)}$ is an optimal solution to

$$\mathbf{x}^{(l+1)} \in \arg\min_{\mathbf{x}} \delta\lambda f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}^{(l)} + \delta\nabla g(\mathbf{x}^{(l)})\|_2^2. \tag{31}$$

The solution to (31) can be expressed by the corresponding proximal operator. Recall that a proximal operator [46] of a functional $J(\cdot)$ with a positive parameter $\mu > 0$ is defined by

$$\mathbf{prox}_J(\mathbf{x};\mu) = \arg\min_{\mathbf{y}} \Big(\mu J(\mathbf{y}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2\Big). \tag{32}$$

Now by relating Equation. (32) (31), we have an iterative update,

$$\mathbf{x}^{(l+1)} \in \mathbf{prox}_f\left(\mathbf{x}^{(l)} - \delta\nabla g(\mathbf{x}^{(l)}); \delta\lambda\right). \tag{33}$$

For $f(\mathbf{x}) = \|\mathbf{x}\|_1$, its proximal operator is given by

$$\mathbf{prox}_{\ell_1}(\mathbf{x};\mu) = \text{sign}(\mathbf{x}) \circ \max(|\mathbf{x}| - \mu, 0), \tag{34}$$

where $\circ$ denotes the Hadamard operator for componentwise operation. As the proximal operator for ℓ_1 is called soft shrinkage, the corresponding iteration (33) is referred to as iterative soft-thresholding algorithm (ISTA) [9,16,14,25, 61,63]. One accelerated scheme of ISTA is called fast iterative soft-thresholding algorithm (FISTA) [4]. It is a momentum based algorithm that utilized the Nesterov's update (12) on the step size, having the form,

$$\begin{cases} t^{(l+1)} &= \frac{1+\sqrt{4(t^{(l)})^2+1}}{2} \\ \mathbf{y}^{(l+1)} &= \mathbf{x}^{(l)} + \frac{t^{(l)}-1}{t^{(l+1)}}(\mathbf{x}^{(l)} - \mathbf{x}^{(l-1)}) \\ \mathbf{x}^{(l+1)} &= \mathbf{prox}_{\ell_1}\left(\mathbf{y}^{(l+1)} - \delta\nabla g(\mathbf{y}^{(l+1)}); \delta\lambda\right). \end{cases} \tag{35}$$

The momentum term in FISTA is proven to be efficient, but the algorithm exhibits oscillatory patterns during the minimization process. To have a guaranteed descent, the accelerated proximal gradient (APG) algorithm [32] compares the objective function at two proximal solutions and selects the smaller one. In short, the APG algorithm goes as follows,

$$\begin{cases} t^{(l+1)} & = \frac{1+\sqrt{4(t^{(l)})^2+1}}{2} \\ \mathbf{y}^{(l+1)} & = \mathbf{x}^{(l)} + \frac{t^{(l)}}{t^{(l+1)}}(\mathbf{u}^{(l)} - \mathbf{x}^{(l)}) + \frac{t^{(l)}-1}{t^{(l+1)}}(\mathbf{x}^{(l)} - \mathbf{x}^{(l-1)}) \\ \mathbf{u}^{(l+1)} & = \mathbf{prox}_f\left(\mathbf{y}^{(l+1)} - \delta\nabla g(\mathbf{y}^{(l+1)}); \delta\lambda\right) \\ \mathbf{v}^{(l+1)} & = \mathbf{prox}_f\left(\mathbf{x}^{(l)} - \delta\nabla g(\mathbf{x}^{(l)}); \delta\lambda\right) \\ \mathbf{x}^{(l+1)} & = \underset{\mathbf{z}\in\{\mathbf{u}^{(l+1)},\mathbf{v}^{(l+1)}\}}{\arg\min} \quad \lambda f(\mathbf{z}) + g(\mathbf{z}). \end{cases} \tag{36}$$

We propose to combine adaptive momentum formula and FISTA for minimizing the general problem of (4). In particular, we replace the FISTA momentum update (35) in terms of FR, thus leading to

$$\begin{cases} \beta^{(l+1)} & = \frac{\|\nabla g(\mathbf{x}^{(l)})\|^2}{\|\nabla g(\mathbf{x}^{(l-1)})\|^2}, \\ \mathbf{y}^{(l+1)} & = \mathbf{x}^{(l)} + \beta^{(l+1)}(\mathbf{x}^{(l)} - \mathbf{x}^{(l-1)}), \\ \mathbf{x}^{(l+1)} & \in \mathbf{prox}_f\left(\mathbf{y}^{(l+1)} - \delta\nabla g(\mathbf{y}^{(l+1)}); \delta\lambda\right). \end{cases} \tag{37}$$

Similarly we can use other momentum terms given in (9)-(11). The proximal operator for the ℓ_1 norm is given in (34), while the proximal operator for $\ell_1 - \ell_2$ [34] can be defined separately into the following cases,

- If $\|\boldsymbol{y}\|_\infty > \lambda$, one has $\mathbf{prox}_{\ell_{1-2}}(\boldsymbol{y};\lambda) = \frac{\boldsymbol{z}(\|\boldsymbol{z}\|_2+\lambda)}{\|\boldsymbol{z}\|_2}$, where $\boldsymbol{z} = \mathbf{prox}_{\ell_1}(\boldsymbol{y};\lambda)$.
- If $\|\boldsymbol{y}\|_\infty \leq \lambda$, then $\boldsymbol{c}^* := \mathbf{prox}_{\ell_{1-2}}(\boldsymbol{y};\lambda)$ is an optimal solution if and only if $c_i^* = 0$ for $|y_i| < \|\boldsymbol{y}\|_\infty$, $\|\boldsymbol{c}^*\|_2 = \|\boldsymbol{y}\|_\infty$, and $c_i^* y_i \geq 0$ for all i. The optimality condition implies infinitely many solutions of $\boldsymbol{c}^*$, among which we choose $c_i^* = \text{sign}(y_i)\|\boldsymbol{y}\|_\infty$ for the smallest i satisfies $|y_i| = \|\boldsymbol{y}\|_\infty$ and the rest coefficients set to be zero.

In what follows, we present experimental results on the convex ℓ_1 minimization in Section. 3.1 and the non-convex $\ell_1 - \ell_2$ minimization in Section. 3.2, respectively.

3.1 Convex ℓ_1 Minimization

We test the performance of various methods to minimize the ℓ_1 norm with the least-squares fitting term, i.e.,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2. \tag{38}$$

We generate the sensing matrix A from Gaussian random matrices and a ground-truth sparse vector $\mathbf{x}$ of sparsity 5. Compressive sensing often involves an under-determined linear system, which implies that the matrix A has more columns than rows (a fat matrix). Here we examine both under-determined (fat) and over-determined (tall) matrices with size 256×1024 and 1024×256, respectively.

We consider two ways to generate the data vector $\mathbf{b}$. One is a standard *sparse recovery* setting, in which $\mathbf{b}$ is obtained by matrix-vector multiplication ($A\mathbf{x}$) with additive Gaussian noise of 30 dB. Another is referred to as a *constructed* case in a way that a given vector $\mathbf{x}$ corresponds to a stationary point of the problem (38) with a constructed data vector $\mathbf{b}$. Following the work of [33], we construct a data vector $\mathbf{b}$ such that a given sparse vector $\mathbf{x}^*$ is a stationary point of (38) for a given positive parameter λ and a given matrix A. Any non-zero stationary point satisfies the following first-order optimality condition:

$$\lambda \mathbf{p}^* + A^\intercal(A\mathbf{x}^* - \mathbf{b}) = \mathbf{0}, \tag{39}$$

where $\mathbf{p}^* \in \partial\|\mathbf{x}^*\|_1$. Denote Sign$(\cdot)$ as the multi-valued sign, *i.e.*,

$$\mathbf{y} \in \text{Sign}(\mathbf{x}) \iff y_i \begin{cases} = 1, & \text{if } x_i > 0, \\ = -1, & \text{if } x_i < 0, \\ \in [-1, 1], & \text{if } x_i = 0. \end{cases} \tag{40}$$

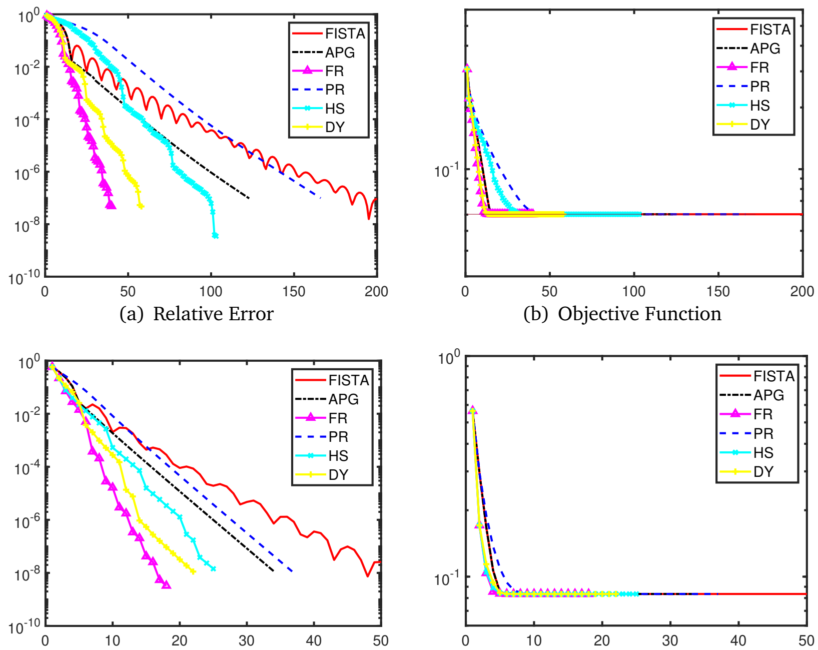
Given $A,\ \lambda$, and $\mathbf{x}^*$, we want to find $\mathbf{x} \in \text{Sign}(\mathbf{x}^*)$ and $\mathbf{x} \in \text{Range}(A^\intercal)$. If $\mathbf{y}$ satisfies $A^\intercal \mathbf{y} = \mathbf{x}$ and $\mathbf{b}$ is defined by $\mathbf{b} = \lambda \mathbf{y} + A\mathbf{x}^*$, then $\mathbf{x}^*$ is a stationary point to (38). To find $\mathbf{x} \in \mathbb{R}^N$, we adopt the iteration

$$\mathbf{x}^{(k+1)} = P_{\text{Sign}(\mathbf{x}^*)}\left(UU^\intercal\left(\mathbf{x}^{(k)}\right)\right), \tag{41}$$

until a stopping criterion is reached. Please refer to [33] for more details.

The constructed setting is examined in Figure. 2 that contains both fat and tall matrices. We use a fixed value of λ for all the algorithms so that they solve the same problem and we tune δ to achieve the fastest convergence. Specifically we choose the best δ among the set $\{10^{-4}, 10^{-3}, \cdots, 10^{1}\}$ that achieves the smallest objective function value when convergent. The proposed method with the FR momentum converges the fastest among all the other methods. FISTA initially converges faster than APG and PR/HS, while it always oscillates no matter whether the matrix A is fat or tall.

We examine a standard sparse recovery setting where the data $\mathbf{b}$ is obtained by matrix-vector multiplication with additive noise. Again two types of matrices are considered, a 256×1024 (fat) matrix and a 1024×256 (tall) one. We use the proposed algorithm with a small step size $\delta = 10^{-3}$ to find the optimal λ value among the set $\{10^{-4}, 10^{-3}, \cdots, 10^{1}\}$ that yields the smallest objective function value. Then we fix this optimal λ for all the competing algorithms while tuning the δ parameter in the same way as in the constructed case. The results are presetned in Figure. 3. We observe that our proposed algorithm has still some advantages over FISTA and APG. Interestingly, in Figure. 3 (a), we observe that the APG algorithm performs worse than FISTA until about 40th iteration due to the oscillatory nature of FISTA. This phenomenon implies that APG is less robust than FISTA in certain applications, which is somewhat counter-intuitive. In this set of experiments, all the methods can not reach the accuracy of 10^{-3} in terms of relative errors, as compared to the constructed cases (10^{-8}). This is because that the ground-truth solution may not be a stationary point to the corresponding minimization problem.



(a) Relative Error (b) Objective Function

(c) Relative Error (d) Objective Function

Fig. 2 Comparison of ℓ_1 minimization methods using a 256×1024 (top) and 1024×256 (bottom) matrix A in a constructed case.

3.2 Non-convex $\ell_1 - \ell_2$ Minimization

Lastly, we consider the non-convex $\ell_1 - \ell_2$ minimization problem,

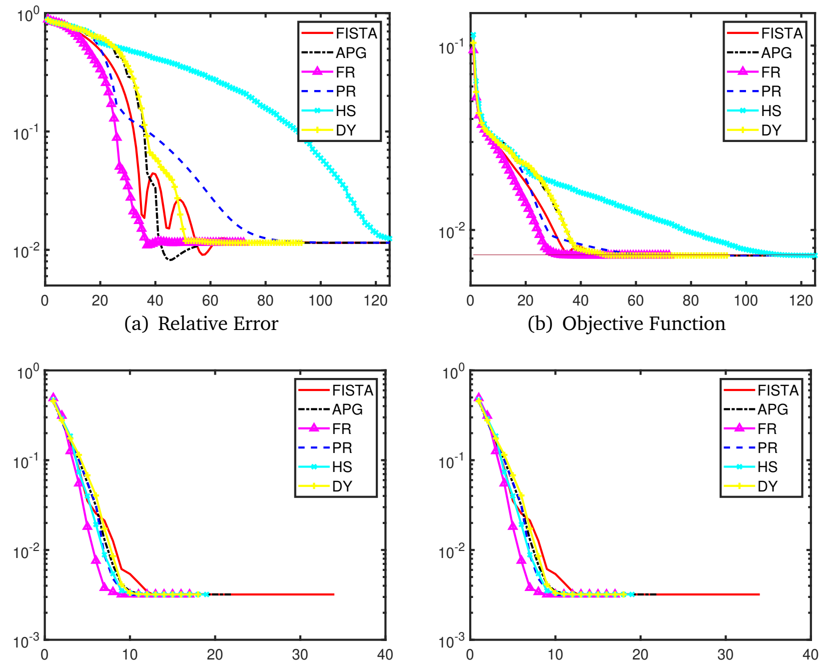
$$F(\mathbf{x}) = \lambda(\|\mathbf{x}\|_1 - \|\mathbf{x}\|_2) + \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \tag{42}$$

which was originally solved by the difference of convex algorithm (DCA) [47, 48]. As a baseline algorithm for comparison, we give a brief description of DCA. After decomposing F into a difference of two convex functions, DCA further relies on the linearization at the current step $\mathbf{x}^{(l)}$ to advance to the next one, i.e.,

$$\mathbf{x}^{(l+1)} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 - \left\langle \mathbf{x}, \frac{\lambda\mathbf{x}^{(l)}}{\|\mathbf{x}^{(l)}\|_2} \right\rangle. \tag{43}$$

To generate a constructed solution for the $\ell_1 - \ell_2$ problem, we only need to replace the iteration (41) for constructing the ℓ_1 solution by

$$\mathbf{x}^{(k+1)} = P_{\text{Sign}(\mathbf{x}^*)}\left(UU^{\mathsf{T}}\left(\mathbf{x}^{(k)} - \frac{\mathbf{x}^*}{\|\mathbf{x}^*\|_2}\right) + \frac{\mathbf{x}^*}{\|\mathbf{x}^*\|_2}\right). \tag{44}$$



(a) Relative Error (b) Objective Function

(c) Relative Error (d) Objective Function

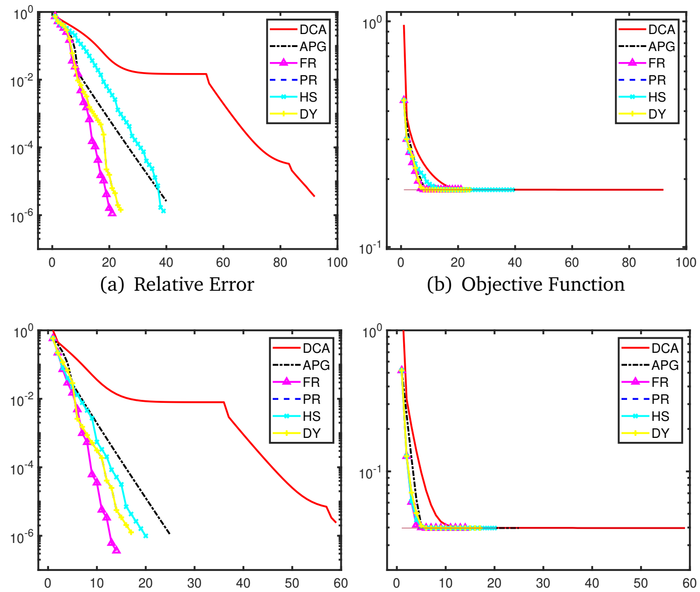
Fig. 3 Comparison of ℓ_1 minimization methods using a 256×1024 (top) and 1024×256 (bottom) matrix A in a standard sparse recovery setting.

Due to the non-convex nature of $F(\mathbf{x})$, the iteration (44) may not converge and $\mathbf{x}^*$ may not exist, especially when A is highly coherent. The results of $\ell_1 - \ell_2$ minimization methods on a constructed case are illustrated in Figure. 4 for matrix sizes of 256×1024 and 1024×256. Note that DCA is a doule-loop algorithm and its iteration number is counted as inner loop iterations, and yet the original DCA implementation [66,36] is the slowest, followed by APG. Our proposed algorithm is the fastest, having a clear advantage over all the other algorithms.

Figure. 5 shows the results for a sparse recovery problem. DCA is still the slowest, while our method is the fastest. The proposed method is worse than DCA for a tall matrix. This may attribute to the fact that the ground-truth signal is not the optimal solution to (42), and as a result, the performance is rather random.

4 Conclusion

In this paper, we leveraged adaptive momentum from nonlinear conjugate gradient algorithms for the purpose of acceleration. Unlike the existing works



(a) Relative Error (b) Objective Function

(c) Relative Error (d) Objective Function

Fig. 4 Comparison of $\ell_1 - \ell_2$ minimization methods using a 256×1024 (top) and 1024×256 (bottom) matrix A in a constructed case.

that rely on line search to establish convergence of gradient-based algorithms, we proposed the use of a fix step size and proved the convergence of FRGD on a quadratic problem. In addition, we combined the adaptive momentum with FISTA to deal with non-smooth objective function. The resulting algorithm has a relatively simple FISTA-like structure. We demonstrated the accelerated phenomena of the proposed approach over FISTA and APG on a convex ℓ_1 minimization and a nonconvex $\ell_1 - \ell_2$ problem for sparse recovery.

Acknowledgements YL acknowledges the support from NSF CAREER DMS-1846690. BW is supported by NSF DMS-1924935, DMS-1952339, DMS-2152762, and DMS-2208361. BW also acknowledges support from the office of Science of the department of energy under grant number DE-SC0021142. MY is supported by the NSF DMS-2012439. XY acknowledge the support from NSF CAREER 2143915. XY was also partly supported by DOE, Office of Science, Office of Advanced Scientific Computing Research (ASCR) as part of Multifaceted Mathematics for Rare, Extreme Events in Complex Energy and Environment Systems (MACSER). QY acknowledges the support from the NSF grant DMS-1821144.

References

1. Al-Baali, M.: Descent property and global convergence of the fletcher—reeves method with inexact line search. IMA Journal of Numerical Analysis **5**(1), 121–124 (1985)
2. Andrei, N.: Another hybrid conjugate gradient algorithm for unconstrained optimization. Numerical Algorithms **47**(2), 143–156 (2008)

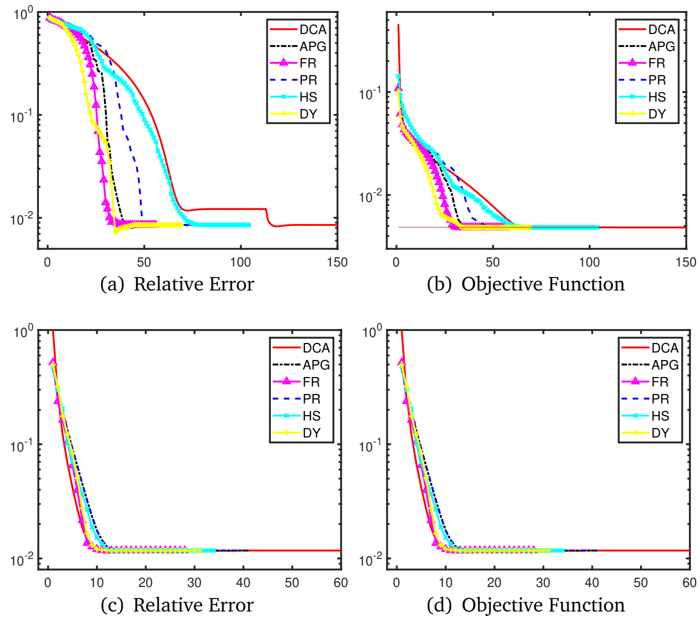


Fig. 5 Comparison of $\ell_1 - \ell_2$ minimization methods using a 256×1024 (top) and 1024×256 (bottom) matrix A in a standard sparse recovery setting.

3. Armijo, L.: Minimization of functions having lipschitz continuous first partial derivatives. Pacific Journal of mathematics **16**(1), 1–3 (1966)
4. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences **2**(1), 183–202 (2009)
5. Bertsekas, D.: Nonlinear Programming. Athena Scientific (1999)
6. Boggess, A., Narcowich, F.J.: A first course in wavelets with Fourier analysis. John Wiley & Sons (2015)
7. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
8. Candès, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted l1 minimization. J. Fourier Anal. Appl. **14**(5-6), 877–905 (2008)
9. Chambolle, A., De Vore, R.A., Lee, N.Y., Lucier, B.J.: Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. IEEE Transactions on Image Processing **7**(3), 319–335 (1998)
10. Chen, X., Zhou, W.: Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization. SIAM Journal on Imaging Sciences **3**(4), 765–790 (2010)
11. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Modeling & Simulation **4**(4), 1168–1200 (2005)
12. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on optimization **10**(1), 177–182 (1999)
13. Dai, Y.h., Yuan, Y.: An efficient hybrid conjugate gradient method for unconstrained optimization. Annals of Operations Research **103**(1), 33–47 (2001)
14. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences **57**(11), 1413–1457 (2004)

15. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
16. Figueiredo, M.A., Nowak, R.D.: An EM algorithm for wavelet-based image restoration. IEEE Transactions on Image Processing **12**(8), 906–916 (2003)
17. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. The computer journal **7**(2), 149–154 (1964)
18. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM Journal on optimization **2**(1), 21–42 (1992)
19. Giselsson, P., Boyd, S.: Monotonicity and restart in fast gradient methods. In: 53rd IEEE Conference on Decision and Control, pp. 5058–5063. IEEE (2014)
20. Golub, G.H., Ye, Q.: Inexact preconditioned conjugate gradient method with inner-outer iteration. SIAM Journal on Scientific Computing **21**(4), 1305–1320 (1999)
21. Guo, L., Li, J., Liu, Y.: Stochastic collocation methods via minimisation of the transformed l_1-penalty. East Asian Journal on Applied Mathematics **8**(3), 566–585 (2018)
22. Guo, W., Lou, Y., Qin, J., Yan, M.: A novel regularization based on the error function for sparse recovery. Journal of Scientific Computing **87**(1), 1–22 (2021)
23. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM Journal on optimization **16**(1), 170–192 (2005)
24. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pacific journal of Optimization **2**(1), 35–58 (2006)
25. Hale, E.T., Yin, W., Zhang, Y.: A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. CAAM TR07-07, Rice University **43**, 44 (2007)
26. Hardt, M.: Robustness versus acceleration (2014). URL `http://blog.mrtz.org/2014/08/18/robustness-versus-acceleration.html`
27. Hermey, D., Watson, G.A.: Fitting data with errors in all variables using the huber m-estimator. SIAM Journal on Scientific Computing **20**(4), 1276–1298 (1999)
28. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards **49**, 409–436 (1952)
29. Hestenes, M.R., Stiefel, E., et al.: Methods of conjugate gradients for solving linear systems, vol. 49. NBS Washington, DC (1952)
30. Huang, X.L., Shi, L., Yan, M.: Nonconvex sorted ell_1 minimization for sparse approximation. Journal of the Operations Research Society of China **3**(2), 207–229 (2015)
31. Huber, P.J.: The place of the l1-norm in robust estimation. Computational Statistics & Data Analysis **5**(4), 255–262 (1987)
32. Li, H., Lin, Z.: Accelerated proximal gradient methods for nonconvex programming. Advances in Neural Information Processing Systems **28**, 379–387 (2015)
33. Lorenz, D.A.: Constructing test instances for basis pursuit denoising. IEEE Trans. Signal Process. **61**(5), 1210–1214 (2013)
34. Lou, Y., Yan, M.: Fast l1-l2 minimization via a proximal operator. J. Sci. Comput. **74**(2), 767–785 (2018)
35. Lou, Y., Yin, P., He, Q., Xin, J.: Computing sparse representation in a highly coherent dictionary based on difference of L_1 and L_2. J. Sci. Comput. **64**(1), 178–196 (2015)
36. Lou, Y., Yin, P., Xin, J.: Point source super-resolution via non-convex l1 based methods. J. Sci. Comput. **68**, 1082–1100 (2016)
37. Lu, Z.: Iterative reweighted minimization methods for ℓ_p regularized unconstrained nonlinear programming. Mathematical Programming **147**(1), 277–307 (2014)
38. Lv, J., Fan, Y., et al.: A unified approach to model selection and sparse recovery using regularized least squares. Annals of Stat. **37**(6A), 3498–3528 (2009)
39. Narushima, Y.: A smoothing conjugate gradient method for solving systems of nonsmooth equations. Applied Mathematics and Computation **219**(16), 8646–8655 (2013)
40. Natarajan, B.K.: Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
41. Nemirovski, A.S., Nesterov, Y.E.: Optimal methods of smooth convex minimization. Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki **25**(3), 356–369 (1985)
42. Nesterov, Y.: A method of solving a convex programming problem with convergence rate o (1/k2). In: Soviet Mathematics Doklady, vol. 27, pp. 372–376 (1983)
43. Nesterov, Y.: Introductory lectures on convex programming volume i: Basic course. Lecture notes **3**(4), 5 (1998)

44. Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)
45. Pang, D., Du, S., Ju, J.: The smoothing fletcher-reeves conjugate gradient method for solving finite minimax problems. ScienceAsia **42**(1), 40–45 (2016)
46. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Opt. **1**(3), 127–239 (2014)
47. Pham-Dinh, T., Le-Thi, H.A.: A D.C. optimization algorithm for solving the trust-region subproblem. SIAM J. Optim. **8**(2), 476–505 (1998)
48. Pham-Dinh, T., Le-Thi, H.A.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals Oper. Res. **133**(1-4), 23–46 (2005)
49. Polak, E., Ribiere, G.: Note sur la convergence de méthodes de directions conjuguées. ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique **3**(R1), 35–43 (1969)
50. Polyak, B.: Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics **4**(5), 1–17 (1964)
51. Powell, M.J.D.: Restart procedures for the conjugate gradient method. Mathematical programming **12**(1), 241–254 (1977)
52. Rahimi, Y., Wang, C., Dong, H., Lou, Y.: A scale invariant approach for sparse signal recovery. SIAM J. Sci. Comput. **41**(6), A3649–A3672 (2019)
53. Rivaie, M., Mamat, M., Abashar, A.: A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches. Applied Mathematics and Computation **268**, 1152–1163 (2015)
54. Roulet, V., d'Aspremont, A.: Sharpness, restart, and acceleration. SIAM Journal on Optimization **30**(1), 262–289 (2020)
55. Saad, Y.: Iterative methods for sparse linear systems. SIAM (2003)
56. Shen, X., Pan, W., Zhu, Y.: Likelihood-based selection and sharp parameter estimation. J. Am. Stat. Assoc. **107**(497), 223–232 (2012)
57. Su, W., Boyd, S., Candes, E.: A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. Advances in neural information processing systems **27**, 2510–2518 (2014)
58. Sun, Q., Zhou, W.X., Fan, J.: Adaptive huber regression. Journal of the American Statistical Association **115**(529), 254–265 (2020)
59. Tong, C., Ye, Q.: Analysis of the finite precision bi-conjugate gradient algorithm for nonsymmetric linear systems. Mathematics of computation **69**(232), 1559–1575 (2000)
60. Unser, M.: Sampling — 50 years after shannon. In: Proceedings of the IEEE, pp. 569 – 587. IEEE (2000)
61. Vonesch, C., Unser, M.: A fast iterative thresholding algorithm for wavelet-regularized deconvolution. In: Wavelets XII, vol. 6701, p. 67010D. International Society for Optics and Photonics (2007)
62. Wang, C., Yan, M., Rahimi, Y., Lou, Y.: Accelerated schemes for the L_1/L_2 minimization. IEEE Trans. Signal Process. **68**, 2660–2669 (2020)
63. Wright, S.J., Nowak, R.D., Figueiredo, M.A.: Sparse reconstruction by separable approximation. IEEE Transactions on signal processing **57**(7), 2479–2493 (2009)
64. Wu, C., Zhan, J., Lu, Y., Chen, J.S.: Signal reconstruction by conjugate gradient algorithm based on smoothing l1-norm. Calcolo **56**(4), 1–26 (2019)
65. Yin, P., Esser, E., Xin, J.: Ratio and difference of l_1 and l_2 norms and sparse representation with coherent dictionaries. Comm. Inf. Syst. **14**(2), 87–109 (2014)
66. Yin, P., Lou, Y., He, Q., Xin, J.: Minimization of ℓ_{1-2} for compressed sensing. SIAM J. Sci. Comput. **37**(1), A536–A563 (2015)
67. Zhang, S., Xin, J.: Minimization of transformed L_1 penalty: Closed form representation and iterative thresholding algorithms. Comm. Math. Sci. **15**, 511–537 (2017)
68. Zhang, S., Xin, J.: Minimization of transformed L_1 penalty: theory, difference of convex function algorithm, and robust application in compressed sensing. Math. Program. **169**(1), 307–336 (2018)
69. Zhang, T.: Multi-stage convex relaxation for learning with sparse regularization. In: Adv. Neural Inf. Proces. Syst. (NIPS), pp. 1929–1936 (2009)