



AFISTA: Accelerated FISTA for sparse signal recovery and compressive sensing

Shahab Babapour¹ · Mehrdad Lakestani¹ · Abolfazl Fatholahzadeh¹

Received: 2 April 2020 / Revised: 27 October 2020 / Accepted: 10 February 2021 /
Published online: 9 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

This paper presents a new fast iterative shrinkage-thresholding algorithm, termed AFISTA. The essential idea is to improve the convergence rate of FISTA using a new continuation strategy leading to a less number of iterations compared to FISTA. The convergence theorem of the AFISTA is proposed. In order to further accelerate the AFISTA method, it is equipped with the Barzilai-Borwein (BB) method. Also, for applications with orthogonal sensing matrix A , we proposed a specialized version of the AFISTA method. AFISTA is tailored for solving the basis pursuit problem which can be applied successfully on a variety of problems arising in signal and image processing issues such as sparse signal recovery, signal and image denoising, image restoration, and compressive sensing. To show the efficiency of the method, we compare our results with generalizations of linearized Bregman and fixed-point continuation (FPC) methods in sparse signal recovery applications, with split Bregman method in compressive sensing for sparse MRI and with Gradient projection for sparse reconstruction (GPSR) method in image deconvolution. Numerical results demonstrate that AFISTA overcomes all of the compared methods in convergence rate and some of them in both convergence rate and quality of reconstructed results.

Keywords Proximal algorithms · Compressed sensing · L_1 minimization · Basis pursuit

1 Introduction

Let $A \in \mathbb{R}^{m \times n}$ with $m < n$ or $m \ll n$ (in compressed sensing), $b \in \mathbb{R}^m$, and $u \in \mathbb{R}^n$. A basis pursuit (BP) problem is a constrained minimization problem as follows [11]:

$$\min_{u \in \mathbb{R}^n} \{ \|u\|_1 : Au = b \}, \quad (1)$$

✉ Mehrdad Lakestani
lakestani@tabrizu.ac.ir; lakestani@gmail.com

Shahab Babapour
sh.falcon32@gmail.com

Abolfazl Fatholahzadeh
abfzadeh@gmail.com

¹ Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

which gives the solution of the underdetermined linear system with minimal L_1 norm. This formulation has been suggested by Chen and Donoho [11] to find a sparse solution among the many possible solutions to $Au = b$. Basis pursuit problem emerges in many applications such as statistics [15], image compression [22], spectral decomposition [23] and compressed sensing [9, 14, 33, 38], image reconstruction [39], specially its applications in medical images [36, 37]. Instead of problem (1), one can consider the unconstrained problem:

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} \|Au - b\|_2^2 + \lambda \|u\|_1, \quad (2)$$

for $\lambda \geq 0$ named LASSO (Least Absolute Shrinkage And Selection Operator) which is closely related to that. Indeed, in the [32], the authors proved that a solution of (1) is either $u = 0$, or else is a solution of problem (2), for some $\lambda > 0$. In (1), one can use another norm instead of L_1 norm. Especially, considering sparsity for the solution, L_0 norm can be replaced with L_1 norm since it gives the sparsest solution for the problem at hand (note that L_0 is not actually a norm) [24, 26]. But, it is an NP-hard problem and so it is difficult to solve [24, 26]. In [29], it is mentioned that under suitable circumstances, the solution to the problem (2) gives the sparsest solution satisfying $Au = b$. Indeed, the problem (2) is a convex approximation to the L_0 regularizing problem [17]:

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} \|Au - b\|_2^2 + \lambda \|u\|_0, \quad (3)$$

where $\|u\|_0$ corresponds to the total number of nonzero elements in u . The problem (2) can be considered as a second order cone programming problem [17] and therefore can be solved by conventional methods such as an interior point method [5]. However, because of its computational complexity, these conventional methods are not suitable for large scale data [29] which arises in the real applications such as CS reconstruction of Magnetic Resonance Imaging (MRI). Many algorithms solve efficiently the problem (2). Some of these methods can be listed as follows: infeasible-point subgradient algorithm (ISA) [25], homotopy based methods [27, 28], fixed - point continuation (FPC) [19, 40], L_1 -Magic algorithm [6], linearized Bregman and its generations [21, 29, 41, 42] and proximal gradient based algorithms such as alternating direction method (ADM) [30], alternating direction method of multipliers (ADMM) [30], iterative shrinkage thresholding algorithm (ISTA) and its modification, fast iterative shrinkage thresholding algorithm (FISTA) [2, 3, 30, 34, 35]. Compared to the traditional methods, the above methods, had been suggested to be suitable for problems with large - scale data and sparse solutions and had been used successfully for the LASSO problem. Most of these methods are proposed to solving LASSO (2) which has nothing to do with the BP problem. Among these methods, fixed - point continuation and an active set method can also solve the BP problem directly only if λ is set to be a tiny value (e.g., $1e-10$).

In this paper, we proposed a method based on FISTA which solves the BP problem directly. Indeed, we modified FISTA using a continuation strategy to lead to a less number of iterations compared to FISTA and to solve the BP problem instead of the LASSO problem. Given that the only difference between the FISTA and AFISTA methods is related to the continuation strategy added to the AFISTA algorithm, which has little computational cost compared to the matrix-vector multiplication, the computational cost of each of these two algorithms at each iteration is almost the same. But the number of iterations for AFISTA is considerably less than FISTA. Also, we presented the convergence analysis of the proposed method. In order to further accelerate the proposed method, we used the Barzilai-Borwein (BB) method [4]. Also, for applications with orthogonal sensing matrix A , we proposed a

specialized version of the AFISTA method. The efficiency of the proposed method is illustrated through a variety of numerical experiments such as sparse signal recovery, recovery of a noisy sinusoidal wave, compressive sensing for MRI images, and image deconvolution.

This paper is organized as follows: Section 2 recalls ISTA and FISTA methods which is the core of the proposed method, in Section 3, we proposed the AFISTA method and its generations, also, we presented its convergence analysis, in Section 4, some numerical experiments were presented. Finally, in the conclusion section, we summarize the whole text of this paper.

2 ISTA and FISTA methods

In this section, the ideas of ISTA and FISTA algorithms, the core of our proposed method, are shortly described In [2], the authors considered the following general formulation to solve

$$\min_{u \in \mathbb{R}^n} f(u) + g(u), \tag{4}$$

where $g : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous convex function which can be both smooth or non-smooth. And $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex and continuously differentiable function with Lipschitz continuous gradient, i.e.,

$$\|\nabla f(u_1) - \nabla f(u_2)\| \leq L\|u_1 - u_2\|, \quad \forall u_1, u_2 \in \mathbb{R}^n,$$

where $\|\cdot\|$ denotes the Euclidean norm in the space \mathbb{R}^n . The basic iterative scheme of the ISTA for solving the convex optimization problem (4) is as follows

$$\begin{aligned} u^k &= \operatorname{argmin}_{u \in \mathbb{R}^n} f(u^{k-1}) + \left\langle u - u^{k-1}, \nabla f(u^{k-1}) \right\rangle + \frac{L}{2} \|u - u^{k-1}\|^2 + g(u) \\ &= \operatorname{argmin}_{u \in \mathbb{R}^n} g(u) + \frac{L}{2} \left\| u - \left(u^{k-1} - \frac{1}{L} \nabla f(u^{k-1}) \right) \right\|^2. \end{aligned}$$

Now by setting $f(u) = \frac{1}{2} \|Au - b\|^2$ and $g(u) = \lambda \|u\|_1$, we have $\nabla f(u) = A^T(Au - b)$ and $L(f) = \|A^T A\|$ and also [2]

$$u^k = \mathcal{T}_{\lambda/L} \left(u^{k-1} - \frac{1}{L} A^T(Au^{k-1} - b) \right),$$

where $\mathcal{T}_\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the shrinkage (soft thresholding) operator defined by

$$\mathcal{T}_\alpha(\mathbf{x})_i = \operatorname{sgn}(x_i) \max\{|x_i| - \alpha, 0\}.$$

Finally, algorithm of the ISTA method for solving LASSO problem can be written as follows [2]

Algorithm 1: ISTA with constant stepsize.

- 1 **Inputs:** $L = L(f)$ A Lipschitz constant of ∇f , $u^0 \in \mathbb{R}^n$;
 - 2 **while** *stopping conditions not satisfied* **do**
 - 3 $u^k = \mathcal{T}_{\lambda/L} \left(u^{k-1} - \frac{1}{L} A^T(Au^{k-1} - b) \right)$;
 - 4 **end**
-

Algorithm 1 shows the ISTA algorithm exactly uses two matrix-vector multiplications involving A and A^T followed by a shrinkage step. So, it is suitable for large-scale

applications. Note that, ISTA is the special case of the proximal forward-backward method proposed in [8] and [31]. A general convergence results for the proximal forward-backward method can be found in [13]. A sufficient but not necessary condition to ensure the convergence of the sequence $\{u^k\}$ generated by the ISTA for the optimal point of the LASSO problem is that the stepsize L satisfies $\frac{1}{L} \in \left(0, \frac{1}{\|A^T A\|}\right)$ [16] or equivalently $L \geq \|A^T A\|$. The authors in [7] proved that the sequence $\{u^k\}$ can converge to minimizer of the LASSO problem with $L > \frac{1}{2}\|A^T A\|$. However, it is not a necessary condition too.

The main drawback of the ISTA algorithm is its rate of convergence. Indeed, it converges slowly to the optimal objective function value. To fix this drawback of the ISTA, in [2] a faster algorithm called FISTA is proposed. The main difference between FISTA and ISTA algorithms is that FISTA uses y^k composed of u^{k-1} and u^{k-2} , whereas ISTA is only based on u^{k-1} . Algorithm 2 shows FISTA method for LASSO problem [2]

Algorithm 2: FISTA with constant stepsize.

```

1 Inputs:  $L = L(f)$  A Lipschitz constant of  $\nabla f$ ,  $y^1 = u^0 \in \mathbb{R}^n$ ,  $t_1 = 1$ 
2 while stopping conditions not satisfied do
3    $u^k = \mathcal{T}_{\lambda/L} \left( y^k - \frac{1}{L} A^T (A y^k - b) \right)$ ;
4    $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ ;
5    $y^{k+1} = u^k + \left( \frac{t_k - 1}{t_{k+1}} \right) (u^k - u^{k-1})$ ;
6 end

```

As appears, the cost of calculation of Algorithms 1 and 2 are almost the same, this is because the execution times of lines 4 and 5 are obviously marginal. Now, we report two theorems of the convergence rate of the ISTA and FISTA methods presented in [2]. Suppose that $F(u) = \frac{1}{2}\|Au - b\|^2 + \lambda\|u\|_1$ and BP problem is solvable, i.e., $U^* = \operatorname{argmin} F \neq \emptyset$ and also F^* be the optimal function value of F at an optimal point u^* .

Theorem 1 [2] *Let $\{u^k\}$ be the sequence generated by ISTA method. Then for any $k \geq 1$*

$$F(u^k) - F(u^*) \leq \frac{C}{k}, \quad \forall u^* \in U^*,$$

where C is a positive constant.

Theorem 2 [2] *Let $\{u^k\}$ be the sequence generated by FISTA method. Then for any $k \geq 1$*

$$F(u^k) - F(u^*) \leq \frac{C}{(k + 1)^2}, \quad \forall u^* \in U^*,$$

where C is a positive constant.

In [35], the authors note that the ISTA and FISTA methods have local linear convergence rate, i.e., the convergence of these methods is linear when they reach close to the optimal point.

3 AFISTA method

Considering the importance of the convergence rate in large-scale problems, we advocate in favor of using a continuation strategy along with the BB stepsize for accelerating the FISTA method.

3.1 Accelerating FISTA with continuation

Here, the AFISTA method is presented. In [19], the authors introduced a fixed point continuation method that solves a sequence of LASSO problems with different values of the regularization parameter λ . Indeed, for a decreasing sequence $\{\lambda_k\}$, as a warm start for the next LASSO problem, they used the solution of the i -th LASSO problem associated with the regularization parameter λ_k , which was obtained using fixed - point algorithm. Similar continuation strategies have also been used in [1, 17, 20, 28].

Here, we have used the continuation technique to accelerate the FISTA method in a **different way**. Compared with previous works, that produce decreasing sequence $\{\lambda_k\}$ with a fixed descent rate $0 < \beta < 1$, we used the convergence of the sequence $\{u_k\}$ generated by FISTA method to produce a decreasing sequence of the regularization parameters which converges to zero. Indeed, here we produced the decreasing sequence $\{\lambda_k\}$, first by setting λ_1 as $\lambda_1 = \tilde{\lambda} \|b\|_2 \|A^T b\|_2 / L$ and then at iteration k , by setting λ_{k+1} as follows

$$\begin{aligned}
 \lambda' &= \tilde{\lambda} \times \|b\|_2 \times \|u^k - u^{k-1}\|_2, \\
 \text{if } \lambda' &< \gamma \lambda_k, \\
 \lambda_{k+1} &= \lambda', \\
 \text{else} \\
 \lambda_{k+1} &= \lambda_k,
 \end{aligned} \tag{5}$$

where $\tilde{\lambda}$ and γ are positive constants. It is notable that, the ISTA and FISTA are proposed to solve LASSO problem which has nothing to do with BP problem. Whereas, the proposed method, by converging λ_k to 0, solves BP problem. Also, to ensure the convergence of the sequence λ_k to 0, we need to take $\gamma < 1$. Numerical results show how much the use of this method to generate a sequence of regularization parameters in the continuation technique can be successful. Now, we are ready to present the algorithm of the AFISTA for the BP problem

Algorithm 3: AFISTA with constant stepsize

- 1 **Inputs:** $L = L(f)$ A Lipschitz constant of ∇f , $y^1 = u^0 \in \mathbb{R}^n$, $t_1 = 1$ and $a > 2$;
 - 2 **Set:** $\lambda_1 = \tilde{\lambda} \times \|b\|_2 \times \|A^T b\|_2 / L$;
 - 3 **while** *stopping conditions not satisfied* **do**
 - 4 $u^k = \mathcal{T}_{\lambda_k/L} \left(y^k - \frac{1}{L} A^T (A y^k - b) \right)$;
 - 5 $t_{k+1} = \frac{k+a}{a}$;
 - 6 $y^{k+1} = u^k + \left(\frac{t_k-1}{t_{k+1}} \right) (u^k - u^{k-1})$;
 - 7 update λ_{k+1} using update rule (5);
 - 8 **end**
-

Given that the only difference between the FISTA and AFISTA methods is in the fifth and seventh lines of the AFISTA algorithm, which has little computational cost compared to the matrix-vector multiplication, the computational cost of each of these two algorithms at each iteration is almost the same. But the number of iterations for AFISTA is **considerably less than** FISTA. This fact is clearly shown in Fig. 1, where the total number of iterations of the FISTA and AFISTA methods to reach the same stopping criterion are **7651** and **155**, respectively. In this experiment, we take the sensing matrix A to be a 300×1000 Gaussian random matrix, the total number of nonzeros of u^* to be 50 and $\lambda = 0.005$ for FISTA method and $\tilde{\lambda} = 2$ and $\gamma = 0.9$ for AFISTA method and we use the stopping criterion $\frac{\|Au^k - b\|}{\|b\|} < 10^{-5}$ for these two methods.

Table 1 shows the difference between the convergence rate of the AFISTA and FISTA methods in terms of the number of iterations ($N^o Iter$ for short) along with stoping criterion $\frac{\|u^k - u^*\|}{\|u^*\|} \leq \varepsilon$ for different values of ε . Here, Func. Err. denotes objective function error $F(u^k) - F(u^*)$. Observe the big difference between AFISTA and FISTA: the former needs **715** iterations with $\gamma = 0.5$, **733** iterations with $\gamma = 0.7$ and **708** iterations with $\gamma = 0.9$ for $\varepsilon = 10^{-14}$ whereas the latter requires **3282** iterations for $\varepsilon = 10^{-1}$. In this test, we take the sensing matrix A to be a 300×1000 Gaussian random matrix, the total number of nonzeros of u^* to be 50 and $\tilde{\lambda} = 2.5$. It is notable that, numerical results are obtained for 10 random samples.

3.2 Convergence analysis of the AFISTA

In this section, first we present the convergence result of the FISTA. Then, we propose the convergence theorem of the AFISTA. Note that, the Authors in [10], proved the convergence of the objective function. However, the convergence of the sequence $\{u_k\}$ given by FISTA was not known. In [10], the Authors show that with a small modification, one can ensure the convergence of the objective function as well as the convergence of the sequence $\{u_k\}$ itself. Here, we present the theorem proved in [10].

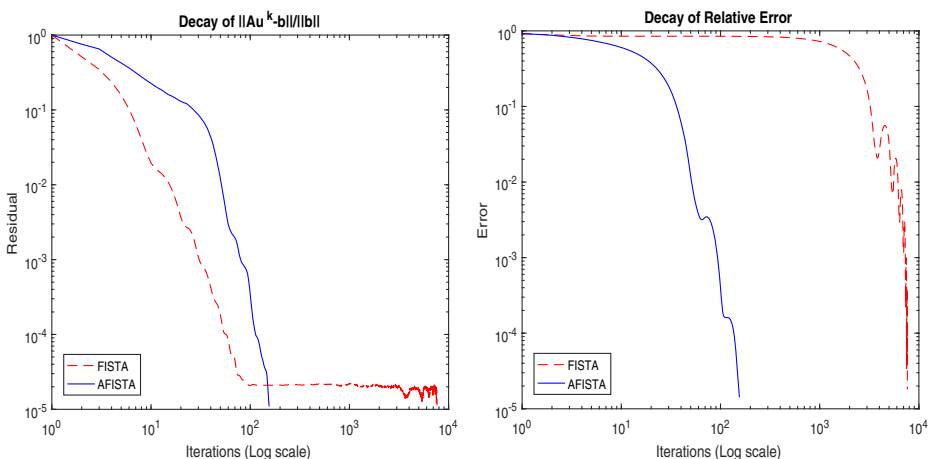


Fig. 1 Plot of residual and relative error as a function of iteration counts in the left and right hand side, respectively. Solid line: AFISTA. Dashed line: FISTA

Table 1 Iteration numbers of AFISTA .vs. FISTA to reach the stoping criterion $\frac{\|u^k - u^*\|}{\|u^*\|} \leq \varepsilon$

ε	AFISTA			FISTA					
	$\gamma = 0.5$		$\gamma = 0.7$		$\gamma = 0.9$		$N^\circ Iter$	Func. Err.	λ
	$N^\circ Iter$	Func. Err.	$N^\circ Iter$	Func. Err.	$N^\circ Iter$	Func. Err.			
1e-1	33	9.64e-02	28	9.50e-02	31	9.40e-02	3282	1.60e-02	5e-3
1e-2	57	9.43e-03	51	9.49e-03	47	9.41e-03	5089	5.31e-03	5e-3
1e-3	92	9.67e-04	89	9.53e-04	84	9.63e-04	7036	1.31e-03	5e-3
1e-4	138	9.61e-05	138	9.79e-05	136	9.64e-05	7431	8.60e-04	5e-3
1e-5	197	9.68e-06	192	9.73e-06	185	9.73e-06	7524	4.33e-04	5e-3
1e-6	242	9.64e-07	243	9.80e-07	239	9.70e-07	32188	4.74e-05	5e-4
1e-7	303	9.75e-08	304	9.66e-08	283	9.44e-08	142699	5.26e-06	5e-5
1e-8	350	9.68e-09	352	9.64e-09	354	9.54e-09	682023	5.39e-07	5e-6
1e-9	419	9.66e-10	402	9.59e-10	407	9.53e-10	3010636	4.99e-08	5e-7
1e-10	468	9.63e-11	464	9.73e-11	474	9.51e-11	26354427	4.86e-09	5e-8
1e-11	539	9.74e-12	551	9.79e-12	527	9.71e-12	-----	-----	-----
1e-12	599	9.79e-13	595	9.63e-13	585	9.68e-13	-----	-----	-----
1e-13	649	9.67e-14	652	9.61e-14	636	9.66e-14	-----	-----	-----
1e-14	715	9.66e-15	733	9.62e-15	708	9.65e-15	-----	-----	-----

Theorem 3 [10] Suppose that $a > 2$ be a real number, and $\forall k \in \mathbb{N}$ let $t_k = \frac{k+a-1}{a}$. Then, the sequence $\{u_k\}$ produced by FISTA converges to a minimizer of objective function F .

Note that, here we set objective function $F(u) = \frac{1}{2}\|Au - b\|^2 + \lambda\|u\|_1$, which is the objective function of the LASSO problem. Now, we are ready to present the following theorem of the convergence of sequence $\{u_k\}$ given by AFISTA.

Theorem 4 Suppose that $\{u_k\}$ is the sequence given by AFISTA. Then, the sequence $\{u_k\}$ converges to the minimizer of the Basis Pursuit problem.

Proof Based on updating rule (5), it is obvious that the sequence $\{\lambda_k\}$ is a non-increasing sequence. Also it is not a fixed sequence. For some $k \geq 1$, suppose that at iteration k , we have $\lambda_k = \lambda_{k+1} = \dots$. It means that, from iteration k , we are solving LASSO problem with regularization parameter λ_k using FISTA starting from initial point u_k . Thus, the convergence of FISTA method yields that for a given $0 < \gamma < 1$

$$\begin{aligned}
 &\exists N' \in \mathbb{N} \text{ s. t. } \|u_{k+N'-1} - u_{k+N'-2}\| < \gamma \|u_{k-1} - u_{k-2}\| \\
 &\Rightarrow \tilde{\lambda} \|b\| \|u_{k+N'-1} - u_{k+N'-2}\| < \gamma \tilde{\lambda} \|b\| \|u_{k-1} - u_{k-2}\| \\
 &\Rightarrow \lambda_{k+N'} < \gamma \lambda_k.
 \end{aligned}
 \tag{6}$$

So, the sequence $\{\lambda_k\}$ is a positive decreasing sequence, and converges to a point. Also, from (6), one can conclude that the sequence $\{\lambda_k\}$ has a subsequence which satisfies $\lambda_{k_{i+1}} < \gamma \lambda_{k_i} \quad \forall i \in \mathbb{N}$. This means that, the sequence $\{\lambda_k\}$ is a convergent sequence which has a subsequence that converges to zero. Thus, we have $\{\lambda_k\} \rightarrow 0$ which will force the equality $Au = b$ constraint. This means that eventually AFISTA solves the Basis Pursuit problem. □

Note that, throughout this paper we set $a = 3$ in formula $t_k = \frac{k+a-1}{a}$.

3.3 AFISTA with BB stepsize

The Barzilai-Borwein (BB) method was proposed in [4]. The motivation of the BB method is the same idea behind quasi-Newton’s method. In fact, BB method provides an approximation to the inverse of the Hessian matrix of the problem at hand, and this approximation simply is set to be a multiple of the identity matrix, that is, the step size α_k is chosen so that $\alpha_k I$ approximates the inverse of the Hessian at iteration k .

Let $S^{k-1} = u^k - u^{k-1}$ and $G^{k-1} = g^k - g^{k-1}$, then the BB stepsize can be computed as follows

$$\alpha_k = \operatorname{argmin}_{\alpha \in \mathbb{R}} \|S^{k-1} - \alpha G^{k-1}\|^2,$$

and so by differentiating and setting it to zero, we obtain

$$\alpha_k = \frac{(G^{k-1})^T S^{k-1}}{(G^{k-1})^T G^{k-1}}. \tag{7}$$

Now, in order to further accelerate the AFISTA method, we intend to use the BB method, which has been successfully used in similar works for l_1 -minimization issues [12, 17, 40, 41]. The AFISTA algorithm contains two basic steps in each iteration. The first step has one iteration of the gradient descent method with fixed stepsize $\frac{1}{L}$, which performs the minimization of the function $f(u) = \frac{1}{2} \|Au - b\|^2$. The second step uses a shrinkage operator, which imposes the sparsity assumption on the obtained solution. Therefore, to improve the convergence rate of the first step, we changed the value of stepsize furnished by BB stepsizes. The above discussions lead to an accelerated version of AFISTA, namely AFISTA-BB (see Algorithm 4).

Algorithm 4: Accelerated FISTA with BB stepsize.

- 1 **Inputs:** given $A, b, L, \tilde{\lambda}$, and $a > 2$;
 - 2 **Set:** $k = 1, \quad y^1 = u^0 = \mathbf{0}, g^1 = A^T(Ay^1 - b), \quad t_1 = 1,$
 $\lambda_1 = \tilde{\lambda} \times \|b\|_2 \times \|A^T b\|_2 / L;$
 - 3 **while** *stopping conditions not satisfied* **do**
 - 4 **if** $k = 1$ **then**
 - 5 $u^k = \mathcal{T}_{\frac{\lambda_k}{L}} \left(y^k - \frac{1}{L} g^k \right);$
 - 6 **else**
 - 7 $\alpha_{k-1} = \frac{(S^{k-1})^T G^{k-1}}{(G^{k-1})^T G^{k-1}};$
 - 8 $u^k = \mathcal{T}_{\frac{\lambda_k \alpha_{k-1}}{2}} \left(y^k - \alpha_{k-1} g^k \right);$
 - 9 **end**
 - 10 $t_{k+1} = \frac{k+a}{a};$
 - 11 $y^{k+1} = u^k + \binom{t_k-1}{t_{k+1}} (u^k - u^{k-1});$
 - 12 $g^{k+1} = A^T(Ay^{k+1} - b);$
 - 13 $G^k = g^{k+1} - g^k;$
 - 14 $S^k = y^{k+1} - y^k;$
 - 15 update λ_{k+1} using update rule (5);
 - 16 **end**
-

Each iteration of the AFISTA-BB algorithm uses two matrix-vector multiplications as in the previous methods, but in practice, its convergence rate is better than the AFISTA. Left and right - hand side of Fig. 2 shows the residual and relative error as a function of iteration counts, respectively. In this figure, to reach the stopping criterion $\frac{\|Au^k - b\|}{\|b\|} < 10^{-5}$, the total number of iterations of the AFISTA and AFISTA-BB methods are **122** and **71**, respectively. Figure 2 shows also the AFISTA-BB method has significantly better performance than the AFISTA. The sensing matrix A is a 600×2000 Gaussian matrix. All other parameters used in this figure are the same as used in Fig. 1.

3.4 Orthogonal case

Let us consider the specific case of $AA^T = I$ which appears in the most applications. In this case, all of the stepsizes are equal to one. For steepest descent stepsize, we have

$$\alpha_k = \frac{(g^k)^T g^k}{(g^k)^T A^T A g^k} = \frac{(Au^k - b)^T}{\text{brace}AA^T(Au^k - b)} (Au^k - b)^T \underbrace{AA^T}_{AA^T} (Au^k - b) = 1.$$

And for BB stepsize, we have

$$S^{k-1} = u^k - u^{k-1}$$

$$G^{k-1} = g^k - g^{k-1} = A^T A(u^k - u^{k-1}),$$

thus

$$\alpha_k = \frac{(G^{k-1})^T S^{k-1}}{(G^{k-1})^T G^{k-1}} = \frac{(u^k - u^{k-1})^T A^T A(u^k - u^{k-1})}{(u^k - u^{k-1})^T A^T \underbrace{AA^T}_{AA^T} A(u^k - u^{k-1})} = 1.$$

Therefore, for this particular case, the AFISTA algorithm can be rewritten as follows

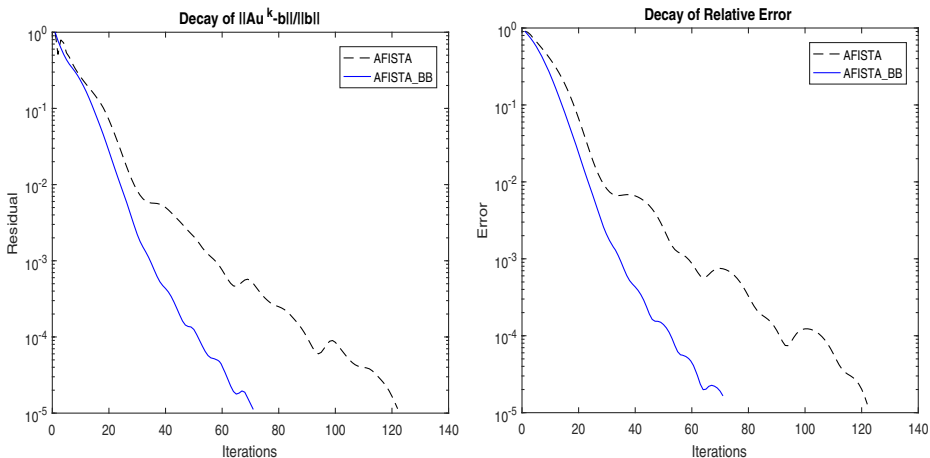


Fig. 2 Plot of residual (left) and relative error (right) as a function of iteration. Solid line: AFISTA-BB. Dashed line: AFISTA

Algorithm 5: Specialized AFISTA for orthogonal case (SAFISTA).

1 **Inputs:** given $A, b, \tilde{\lambda}, \alpha \in [0.9, 1]$, and $a > 2$;
 2 **Set:** $k = 1, \quad y^1 = u^0 = \mathbf{0}, g^1 = A^T(Ay^1 - b), \quad t_1 = 1,$
 $\lambda_1 = \tilde{\lambda} \times \|b\|_2 \times \|A^T b\|_2 / L$;
 3 **while** *stopping conditions not satisfied* **do**
 4 $u^k = \mathcal{T}_{\frac{\lambda_k \alpha}{2}}(y^k - \alpha g^k)$;
 5 $t_{k+1} = \frac{k+a}{a}$;
 6 $y^{k+1} = u^k + \left(\frac{t_k-1}{t_{k+1}}\right)(u^k - u^{k-1})$;
 7 $g^{k+1} = A^T(Ay^{k+1} - b)$;
 8 update λ_{k+1} using update rule (5);
 9 **end**

4 Numerical experiments

In this section, the efficiency of the AFISTA algorithm is shown for solving the BP problem. We applied the AFISTA on several examples. First, it is done on sparse signal recovery, second on noisy sinusoidal wave recovery, third on CS for sparse MRI, and finally on image deconvolution problem. All of the discussed algorithms were implemented using Lenovo B590 laptop with Intel(R) Celeron(R) CPU 1005M @ 1.90GHz Processor in MATLAB. Throughout this section, the accuracy of the methods is measured by relative error $\frac{\|u^k - \bar{u}\|}{\|\bar{u}\|}$, where \bar{u} and u^k stand for the original signal and the reconstructed one at iteration k , respectively.

4.1 Sparse signal recovery

In this subsection, our goal is to reconstruct the original sparse signal \bar{u} from the sensing matrix A and the observation b . To generate the original signal \bar{u} , the position of the non-zero components of signal \bar{u} were randomly selected using the **randsample** command in MATLAB and then each of these non-zero components were sampled from the standard Gaussian (**randn** in MATLAB) or from $[-1, 1]$ uniformly at random (**2*rand-1** in MATLAB) or from $\{-1, 1\}$ standard Gaussian random (**sign(randn)** in MATLAB). And we use three types of measurement matrices: orthogonalized Gaussian matrices that its entries were generated using Gaussian distribution and its rows were orthogonalized by QR decompositions, standard Gaussian matrices, and partial discrete cosine transform (DCT) matrices that its m rows were randomly selected from the $n \times n$ DCT matrix. Note that, when we deal with the partial DCT matrix, we can compute matrix-vector multiplications involving A and A^T using the discrete cosine transform and the inverse discrete cosine transform, respectively. Here, we have done two types of tests: Noise - free Test: its results are given in Fig. 3 and Table 2, and Noisy Test: its results are given in Figs. 4, 5, 6 and Table 3. In noisy case, we take the measurement vector b as follows

$$b = A\bar{u} + \eta,$$

where η is the white Gaussian noise of variance σ^2 . To detect the noise level, instead of using σ , we use the signal to noise ratio (SNR), which is defined as follows

$$SNR = 20 \log_{10} \left(\frac{\|\bar{u}\|}{\|\eta\|} \right)$$

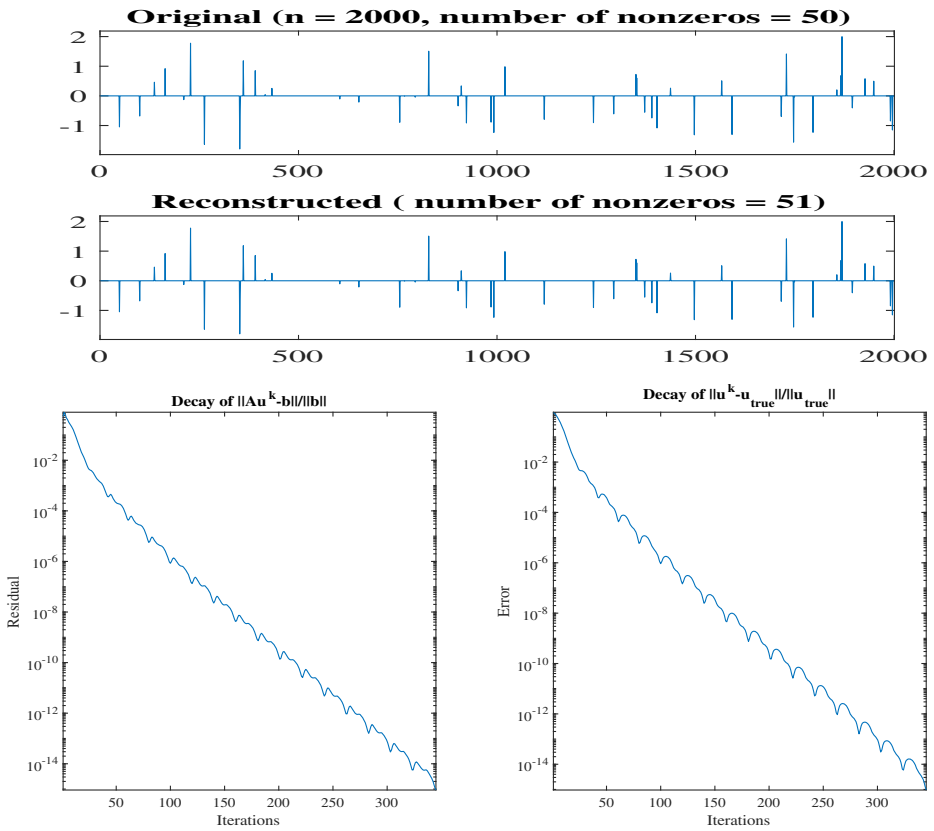


Fig. 3 Noise - free test. Top: Original signal with $\|\vec{u}\|_0 = 50$. Middle: Reconstructed signal using AFISTA-BB algorithm with $\|u^k\|_0 = 51$ after 346 iterations. Bottom left: Graph of residual values with respect to iteration counts. Bottom right: Graph of relative error with respect to iteration counts

Here, we used AFISTA-BB method along with the following parameters, sensing matrix $A = 600 \times 2000$ standard Gaussian matrix, $\|\vec{u}\|_0 = 50$ and $\tilde{\lambda} = 2$. Figure 3 shows the residual and the relative error. AFISTA-BB algorithm converges to a solution with 10^{-15} residual and relative error in less than **350** iterations. The stopping criterion is $\frac{\|Au^k - b\|}{\|b\|} < 10^{-15}$. Figures 4–6 show the results of the noisy tests with three different levels of noise. These figures show, even for huge noise levels with negative SNRs, the AFISTA-BB algorithm works well.

In Table 2, we compare specialized AFISTA for orthogonal case, termed SAFISTA, with two different methods, namely kicking+BB line search (KB) and L-BFGS (LB) of the linearized Bergman [41]. In this Table, following [41], we use orthogonalized Gaussian matrices as sensing matrices and we report average results which are obtained using 20 random instances. Also, we let $\tilde{\lambda} = 0.1$ for Gaussian signals and $\tilde{\lambda} = 0.05$ for uniformly random $[-1, 1]$ signals, $\alpha = 0.95$ and the stopping criterion equal to $\frac{\|Au^k - b\|}{\|b\|} < 10^{-5}$. From Table 2, one can observe that the number of matrix-vector multiplications required for the SAFISTA method is less than linearized Bergman’s methods, so SAFISTA is faster

Table 2 Noiseless test: Comparison of the SAFISTA for orthogonal case with KB and LB methods [41]. Numerical results are obtained for 20 random samples

Signal type		A and A ^T mult's			Relative error			Time(sec.)
m	n	KB	LB	proposed	KB	LB	proposed	proposed
Standard Gaussian								
\bar{u} ₀ = 50								
300	1000	324	193	172	7.71e-06	6.62e-06	1.72e-05	4.89e-02
600	2000	1297	286	138	1.58e-05	7.16e-06	1.06e-05	1.75e-01
1200	4000	1018	339	118	6.11e-06	6.94e-06	9.68e-06	5.87e-01
\bar{u} ₀ = 20								
300	1000	102	128	122	5.14e-06	3.75e-06	1.73e-05	3.86e-02
600	2000	145	152	106	5.67e-06	5.46e-06	1.19e-05	1.32e-01
1200	4000	275	215	98	5.39e-06	6.43e-06	9.35e-06	4.74e-01
Uniformly random [-1, 1]								
\bar{u} ₀ = 50								
300	1000	343	214	194	7.71e-06	6.60e-06	1.48e-05	6.24e-02
600	2000	968	282	158	8.40e-06	5.87e-06	1.06e-05	2.02e-01
1200	4000	1183	370	138	9.74e-06	5.94e-06	1.00e-05	6.32e-01
\bar{u} ₀ = 20								
300	1000	119	141	120	4.77e-06	4.42e-06	1.57e-05	3.92e-02
600	2000	269	167	106	6.33e-06	4.82e-06	9.96e-06	1.45e-01
1200	4000	435	257	100	5.83e-06	5.42e-06	9.28e-06	4.64e-01

than these tow methods. Notably, the quality of reconstructions for all three methods are the same.

Table 3 shows the comparison of AFISTA and the linearized Bregman with kicking (KO) [29]. Let *std* denotes the standard deviation. For standard Gaussian matrices and partial DCT ones we use the AFISTA-BB with stopping criterion $std(Au^k - b) < \sigma$ and SAFISTA with $std(Au^k - b) < 0.5\sigma$, respectively. We report mean results obtained for 10 instances. Numerical results show that the AFISTA method overcomes KO method from the perspective of convergence rate and accuracy.

Table 4 shows the comparison of the SAFISTA with FPC-AS (i.e. FPC Active Set [40]). We used latest version of the FPC-AS Matlab code, which is available at http://www.caam.rice.edu/~optimization/L1/FPC_AS/. For a fair comparison, first, we run the FPC-AS code and then, having the FPC-AS's solution u_F , we terminated the SAFISTA method at iteration k when the following solution u^k is satisfied

$$10^{-5} \|u^k\|_1 + \frac{1}{2} \|Au^k - b\|_2^2 \leq 10^{-5} \|u_F\|_1 + \frac{1}{2} \|Au_F - b\|_2^2.$$

Table 4 shows also the average results which are obtained for 20 random instances using orthogonalized Gaussian matrices as sensing matrices. Also, we take $\alpha = 0.95$ and $\tilde{\lambda} = 0.04$. Note that, in this table, we have both noise - free and noisy cases. From this table, one can see that for the noise free cases FPC-AS works better than SAFISTA, but in noisy cases, SAFISTA works better and required fewer matrix-vector multiplications than FPC-AS. In the last column of this table we present the time ratio of FPC-AS and proposed

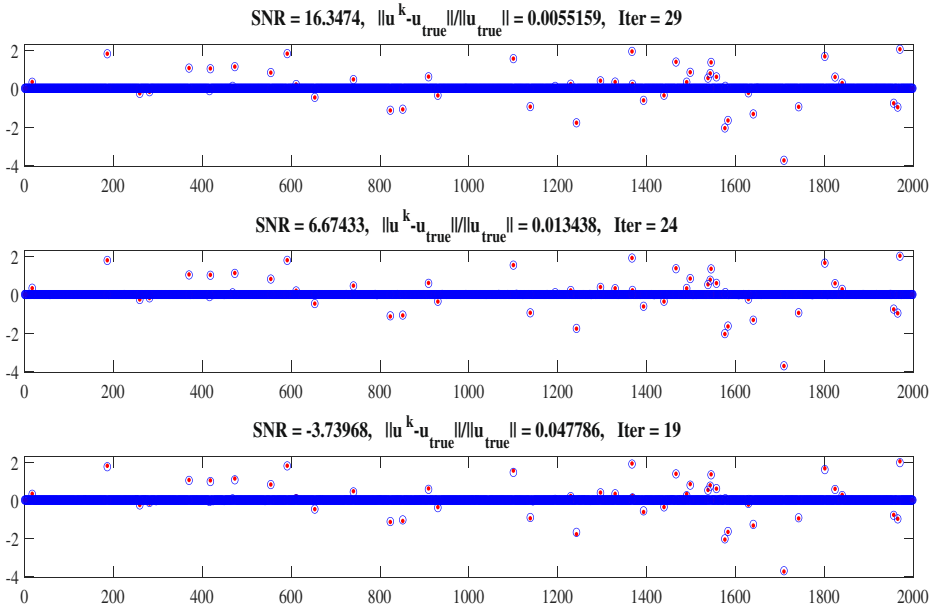


Fig. 4 Noisy test for three different noise levels with SNRs = 16.35, 6.67, -3.74 and for standard Gaussian signal with $\|\bar{u}\|_0 = 50$. Original signal: red dots. Reconstructed signal: blue circles

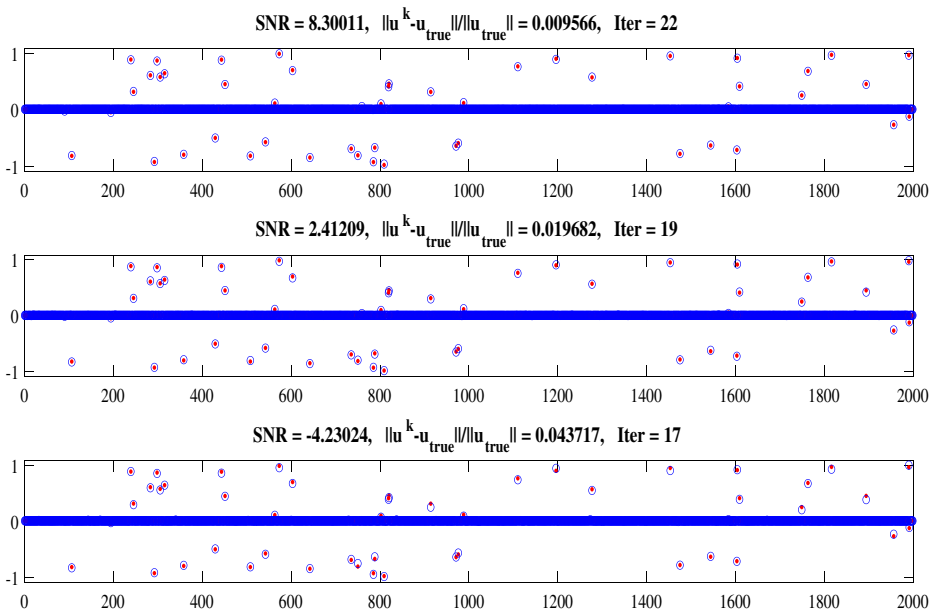


Fig. 5 Noisy test for three different noise levels with SNRs = 8.30, 2.41, -4.23 and for uniformly random $[-1, 1]$ signal with $\|\bar{u}\|_0 = 50$. Original signal: red dots. Reconstructed signal: blue circles

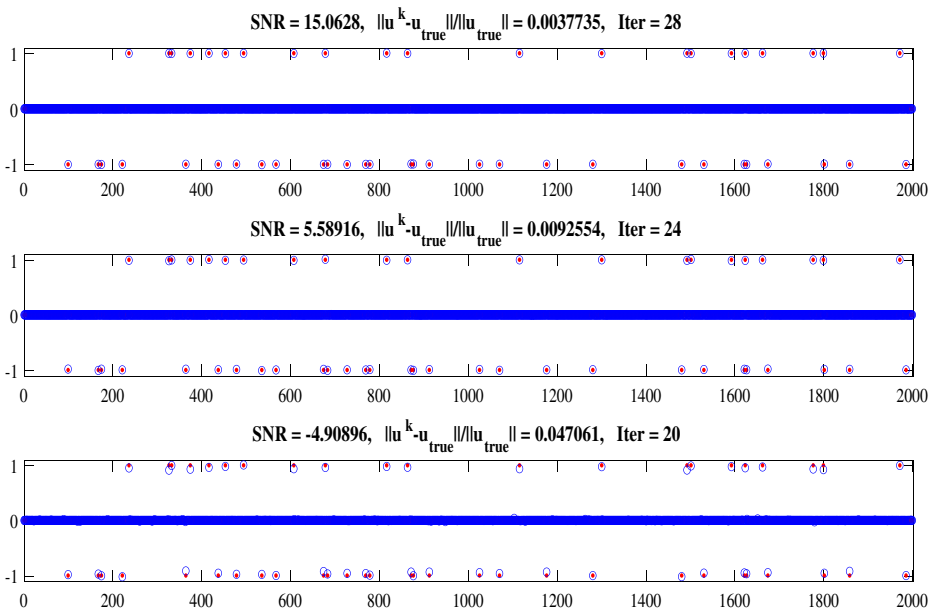


Fig. 6 Noisy test for three different noise levels with SNRs = 15.06, 5.59, -4.91 and for uniformly random $\{-1, 1\}$ signal with $\|\bar{u}\|_0 = 50$. Original signal: red dots. Reconstructed signal: blue circles

methods, which the average of this column shows that for noise - free cases stopping time of these two methods is approximately the same but for noisy cases, the proposed method is approximately 2 times faster than FPC-AS.

4.2 Recovery of noisy sinusoidal wave

For the first compressive sensing application in the frequency domain, we consider a sinusoidal wave that is sparse in frequency domain:

$$\bar{x}(t) = a_1 \sin(\beta_1 t) + a_2 \cos(\beta_2 t),$$

where a_i and β_i denote magnitudes and frequencies of the signal. The observed signal \tilde{x} , is a noisy signal with the form $\tilde{x} = \bar{x} + \eta$, where η is the white Gaussian noise of variance σ^2 . Note that, in CS application, only a random measurement of observed signal \tilde{x} is available. So, the problem is to reconstruct the original signal \bar{x} from a random measurement of the observed signal \tilde{x} . Then, the problem can be written as $Au = b$, where $b = \tilde{x}(I)$, I is a random subset of $\{1, 2, \dots, n\}$, n is total signal length and the sensing matrix A is the partial inverse Fourier transform matrix. Note that, signal u is in the Fourier domain, thus we need to take an inverse Fourier transform to get the reconstructed signal in the physical domain. In numerical experiments, to generate original signal \bar{x} , the magnitudes a_i are generated uniformly at random from $[-1, 1]$ and β_i s are taken to be random multiples of $\frac{2\pi}{n}$, i.e., $\beta_i = k_i \frac{2\pi}{n}$, where k_i is randomly selected from $\{0, 1, \dots, n - 1\}$.

In this test, AFISTA-BB algorithm is used with parameters $\tilde{\lambda} = 0.01$, $L = 1$ and the stopping criterion $\frac{\|Au^k - b\|}{\|b\|} < 0.01$. Numerical results for this test for three different levels of noise with SNRs = **9.5369**, **4.0767** and **-0.9420** are shown in Figs. 7, 8 and 9. Details

Table 3 Noisy test: Comparison of the AFISTA and the KO [29]. Standard Gaussian and partial DCT matrices are used for AFISTA-BB and SAFISTA algorithms, respectively. Numerical results are obtained for 10 random samples

Stopping criteria			Avg. SNR		$N^o\ Iter$		Relative error		Time(sec.)
m	n	$\tilde{\lambda}$	KO	proposed	KO	proposed	KO	proposed	proposed
Gaussian matrices									
$std(Au^k - b) < \sigma$									
$\ \tilde{u}\ _0 = 0.05n$									
300	1000	3	26.12	25.46	420	41	0.0608	0.0053	3.45e-02
600	2000	3	25.44	25.47	206	51	0.0636	0.0032	1.45e-01
1200	4000	3	26.02	25.47	114	77	0.0622	0.0024	7.47e-01
$\ \tilde{u}\ _0 = 0.02n$									
156	1000	5	27.48	26.32	890	59	0.0456	0.0098	3.29e-02
312	2000	5	25.06	24.92	404	55	0.0638	0.0059	9.11e-02
468	4000	5	26.04	25.90	216	96	0.0557	0.0060	4.12e-01
DCT matrices									
$std(Au^k - b) < 0.5\sigma$									
$\ \tilde{u}\ _0 = 0.05n$									
2000	4000	0.01	23.97	23.68	151	12	0.0300	0.0607	2.77e-02
10000	20000	0.001	24.00	24.04	250	11	0.0300	0.0805	8.30e-02
25000	50000	0.001	24.09	24.02	274	14	0.0304	0.0609	2.53e-01
$\ \tilde{u}\ _0 = 0.02n$									
1327	4000	0.05	24.29	23.87	130	21	0.0223	0.0717	4.23e-02
7923	20000	0.005	24.37	24.07	223	14	0.0204	0.0559	1.06e-01
21640	50000	0.005	24.16	24.01	283	24	0.0193	0.0637	4.34e-01

of the results obtained in this test are presented in the Table 5. These results show that to have a reliable reconstruction at the higher level of noise, we need to use a greater number of measurements.

4.3 CS for sparse MRI

To illustrate the efficiency of our method in practical applications, we tested it on MRI images. All images used in this section are of the size 720×960 which are available at <https://usa.healthcare.siemens.com/magnetic-resonance-imaging/>. After the vectorization of these images, the original signal \tilde{u} will be of length $n = 691200$. In the CS application for sparse MRI, we need to take the sensing matrix A to be a partial discrete Fourier transform matrix. Note that, we compute matrix-vector multiplications involving A and A^T using the fast Fourier transform and the inverse fast Fourier transform, respectively. Throughout this part, we use the AFISTA-BB algorithm with parameters $\tilde{\lambda} = 1 \times 10^{-10}$, $L = 1$ and the stopping criterion $\frac{\|Au^k - b\|}{\|b\|} < 5 \times 10^{-3}$.

Considering the importance of the sparsity assumption, we first start with angiogram images, which is usually sparse in the spatial domain. We used four angiogram images

Table 4 Comparison of the SAFISTA and the FPC active set [40]. Numerical results are obtained for 20 random samples

Sensing matrix			Orthogonalized Gaussian matrix						
Test			A and A^T mult's		Relative error		Time(sec.)		
m	n	SNR	FPC-AS	proposed	FPC-AS	proposed	FPC-AS	proposed	ratio
Without noise									
$\ \bar{u}\ _0 = 50$									
300	1000		158	132	1.46e-05	8.69e-05	1.06e-01	6.17e-02	1.72
600	2000		108	101	4.18e-05	8.11e-05	1.41e-01	1.71e-01	0.82
1200	4000		91	90	1.24e-05	5.56e-05	3.48e-01	5.60e-01	0.62
$\ \bar{u}\ _0 = 20$									
300	1000		115	145	2.07e-05	6.78e-05	8.75e-02	6.72e-02	1.30
600	2000		101	90	2.10e-05	5.70e-05	1.63e-01	1.47e-01	1.11
1200	4000		78	85	5.94e-06	2.90e-05	3.52e-01	4.96e-01	0.71
Noisy									
$\ \bar{u}\ _0 = 50$									
300	1000	27.08	89	41	8.74e-02	8.85e-02	8.13e-02	2.28e-02	3.57
600	2000	25.97	77	37	7.37e-02	5.99e-02	1.38e-01	7.90e-02	1.75
1200	4000	23.92	55	38	6.89e-02	6.65e-02	2.47e-01	3.13e-01	0.79
$\ \bar{u}\ _0 = 20$									
300	1000	25.88	76	34	7.36e-02	6.69e-02	6.64e-02	1.87e-02	3.55
600	2000	23.55	52	29	7.83e-02	7.13e-02	1.06e-01	5.47e-02	1.94
1200	4000	24.74	45	27	5.97e-02	4.41e-02	2.66e-01	1.83e-01	1.45

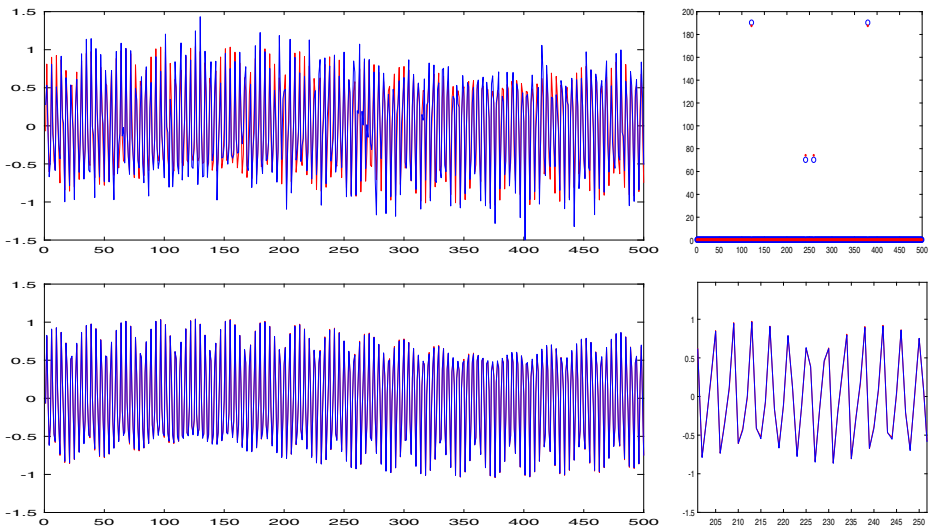


Fig. 7 Reconstruction using 30% of the noisy signal. Top left: Original (red) and noisy (blue) signals. Top right: The absolute magnitude of original (red dots) and reconstructed (blue circles) signals in frequency domain. Bottom left: Original (red) and reconstructed (blue) signals. Bottom right: A close-up view of the bottom left figure

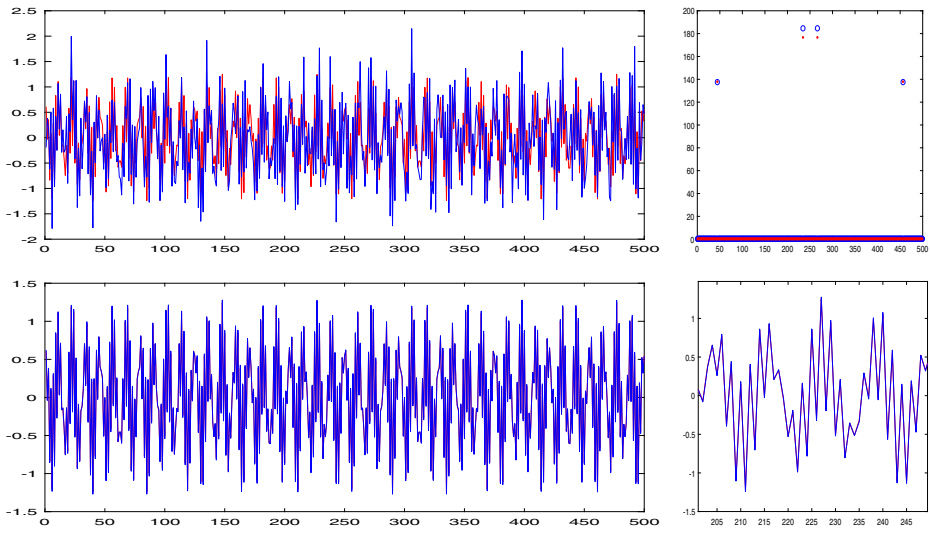


Fig. 8 Reconstruction using 50% of the noisy signal. Top left: Original (red) and noisy (blue) signals. Top right: The absolute magnitude of original (red dots) and reconstructed (blue circles) signals in frequency domain. Bottom left: Original (red) and reconstructed (blue) signals. Bottom right: A close-up view of the bottom left figure

with different sparsity levels. We call these four images *angio1*, *angio2*, *angio3* and *angio4*, which are used for producing Fig. 10, from top to bottom, and Fig. 11, respectively. Figure 10 shows the results of our first test on angiography images. In this figure, we

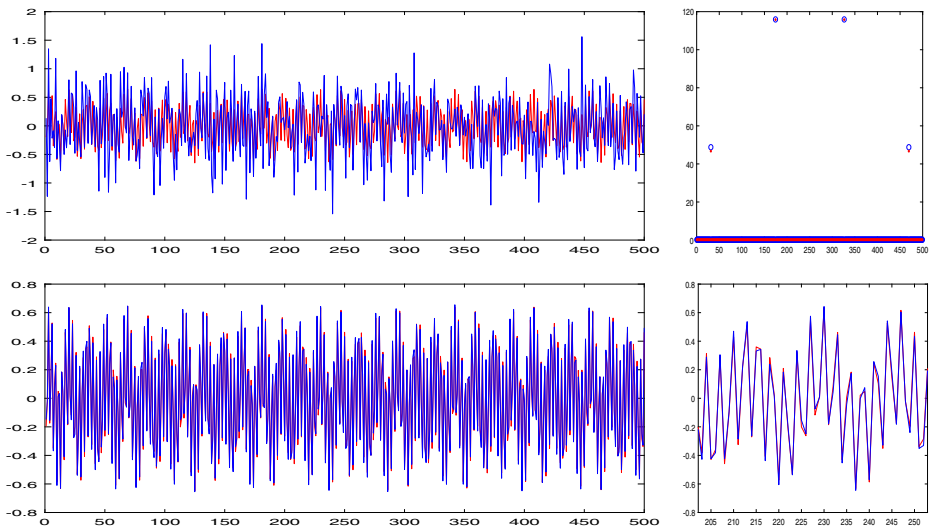


Fig. 9 Reconstruction using 70% of the noisy signal. Top left: Original (red) and noisy (blue) signals. Top right: The absolute magnitude of original (red dots) and reconstructed (blue circles) signals in frequency domain. Bottom left: Original (red) and reconstructed (blue) signals. Bottom right: A close-up view of the bottom left figure

Table 5 Details of the results obtained in Figs. 7, 8 and 9

Figure	Data used	SNR	$N^{\circ}Iter$	Relative error	Time(sec.)
Figure 7	30%	9.5369	24	2.79e-02	3.37e-02
Figure 8	50%	4.0767	32	3.74e-02	5.21e-02
Figure 9	70%	-0.9420	20	7.55e-02	5.77e-02

compare the AFISTA-BB algorithm with the split Bregman method [18]. The Matlab code for the split Bregman method can be downloaded from http://tag7.web.rice.edu/Split_Bregman.html. The obtained results show that for enough sparse images, the AFISTA-BB works better than split Bregman in convergence rate and accuracy of the solution. In the other words, Whereas AFISTA-BB is faster than split Bregman, the quality of reconstructed images using AFISTA-BB is better than split Bregman. Details of the results obtained in this test are presented in the Table 6. Figure 11 shows the results of our second test on angiography images. In this test, we use image *angio4* that is 52% sparse. Compared to the images used in Fig. 10, which are about 70% sparse, this image is not sparse enough to reconstruct using 30% of its frequency domain data. From Fig. 11, we can see that the AFISTA-BB algorithm failed to have a proper reconstruction using 30% of frequency domain data.

This figure also shows that the AFISTA-BB algorithm works well in reconstructing the image *angio4* using 50% of its frequency domain data. Details of the results obtained in this test are presented in the Table 7. Figure 11 shows the importance of the sparsity assumption

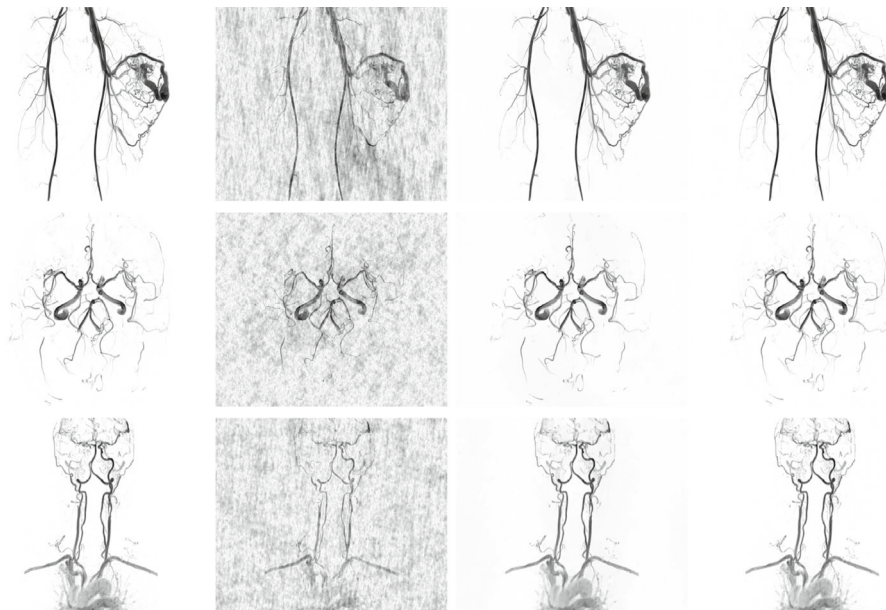


Fig. 10 CS reconstruction of MR Angiography images using 30% of frequency domain data. Columns from left to right: Original images, images reconstructed using 30% of the frequency domain data, in which missing samples are filled with zero, images reconstructed by split Bregman algorithm after 8 inner iterations (3 outer iterations) and images reconstructed by AFISTA-BB using the stopping criterion $\frac{\|Au^k - b\|}{\|b\|} \leq 5 \times 10^{-3}$, respectively

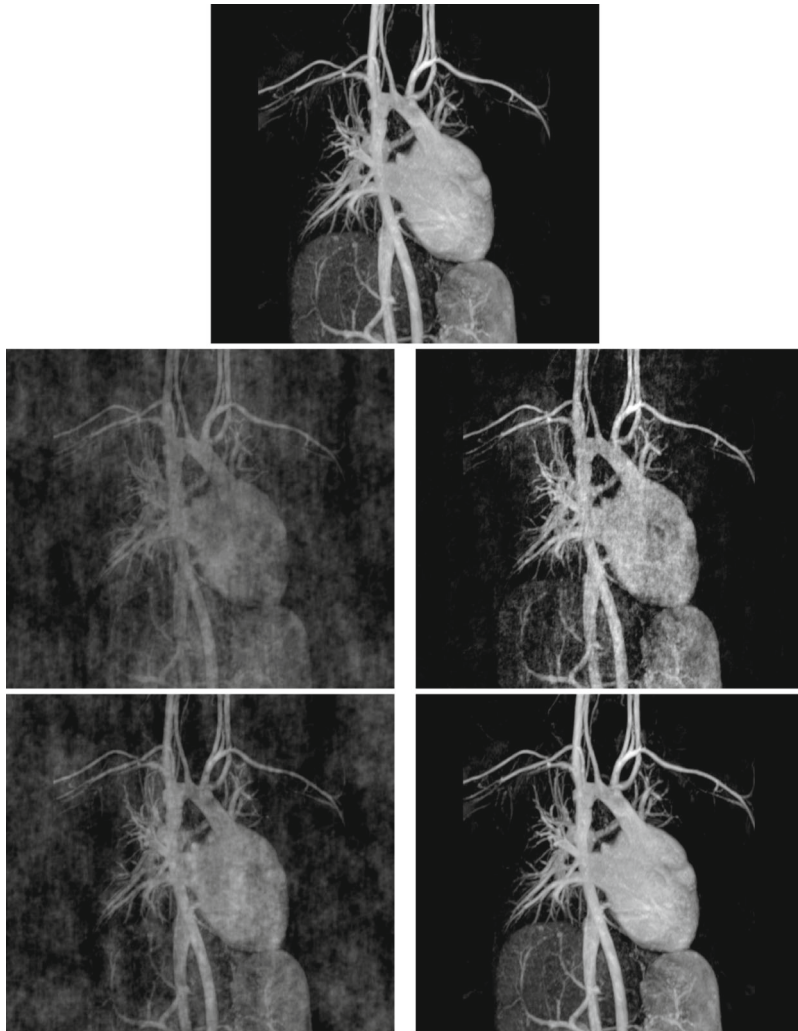


Fig. 11 CS reconstruction of a 50% sparse MR Angiography image. Top: Original image. Middle left: image reconstructed using 30% of the frequency domain data, in which missing samples are filled with zero. Middle right: Result of AFISIA-BB algorithm using 30% of frequency domain data. Bottom left: image reconstructed using 50% of the frequency domain data, in which missing samples are filled with zero. Bottom right: Result of AFISIA-BB algorithm using 50% of frequency domain data

Table 6 Details of the results obtained in Fig. 10

Images	Sparsity	$N^o Iter$		Relative error		Time(sec.)	
		SB	proposed	SB	proposed	SB	proposed
angio1	74%	3(8)	22	1.39e-01	1.89e-02	6.57	2.83
angio2	69%	3(8)	22	9.74e-02	1.36e-02	6.15	2.63
angio3	72%	3(8)	26	1.35e-01	2.04e-02	6.48	3.24

Table 7 Details of the results obtained in Fig. 11

Frec. data	$N^{\circ} Iter$	Relative error	Time(sec.)
30%	43	2.20e-01	5.27
50%	32	9.30e-03	4.31

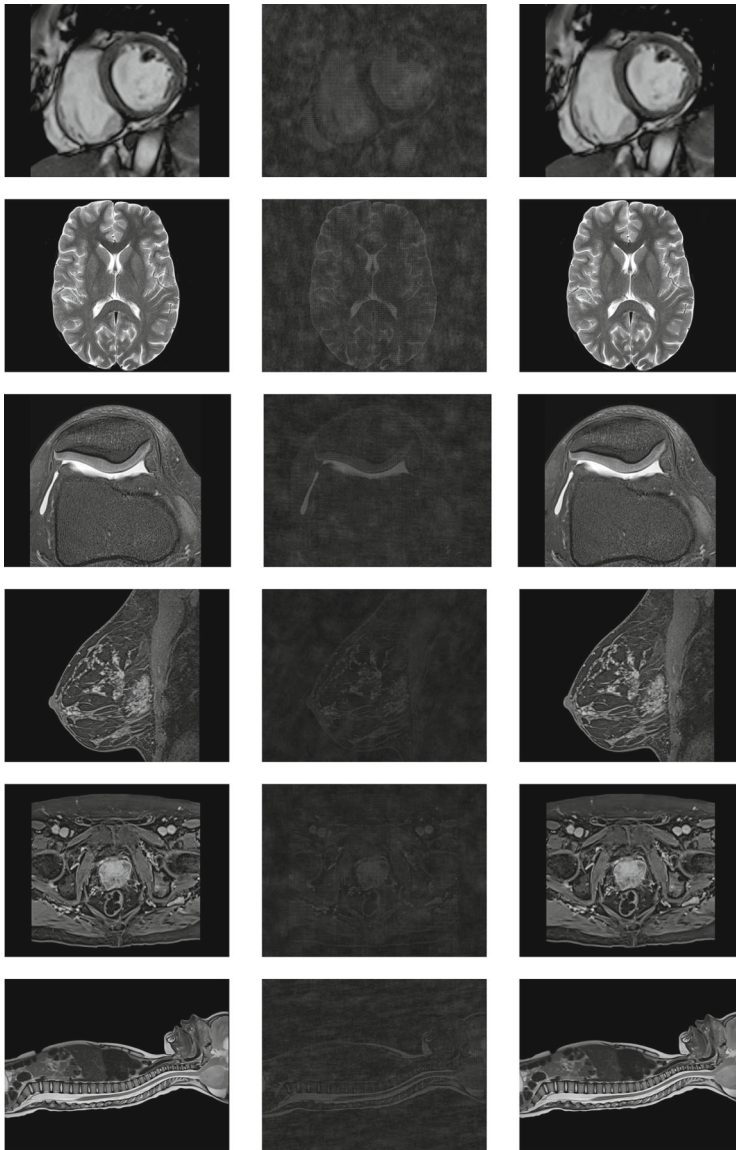


Fig. 12 CS reconstruction of MRI images in wavelet domain (The original signal is considered as the wavelet coefficients of the MRI images) using 25% of frequency domain data. Columns from left to right: Original images, Images reconstructed using 25% of the frequency domain data, in which missing samples are filled with zero and Images reconstructed by AFISIA-BB algorithm, which are obtained by using the stopping criterion $\frac{\|Au^k - b\|}{\|b\|} \leq 5 \times 10^{-3}$, respectively

Table 8 Details of the results presented in Fig. 12

Images	Wav. Dom. Sparsity	Spat. Dom. Sparsity	SNR	Wav. Dom. Relerr	Spat. Dom. Relerr	Iter	Time(s)
Cardiac	95%	37%	3.1282	4.54e-03	2.68e-02	16	1.84
Neurology	90%	52%	3.3644	6.54e-03	3.30e-02	21	2.48
Orthopedics	90%	25%	0.9697	8.29e-03	4.97e-02	21	2.52
Health	91%	48%	-1.7118	7.51e-03	5.69e-02	19	2.44
Oncology	91%	35%	-0.3860	6.92e-03	5.04e-02	19	2.65
Pediatrics	92%	62%	0.8107	5.50e-03	3.43e-02	18	2.11

in BP problem. It is obvious that for other MRI images with significantly little sparsity level the AFISTA method will not work.

As the last test, we applied AFISTA-BB on MRI images. Here, the original signal is considered as the wavelet coefficients of the MRI images, in which the Haar wavelet is used to decompose images up to five resolution levels. The results of this test are presented in Fig. 12 and its details are presented in Table 8.

4.4 Image deconvolution

For our final experiment, we test the proposed AFISTA-BB on the Image Deconvolution problem. In other words, we set matrix A to be RW , in which, R is a matrix representation of the blur operation, and W denotes the inverse wavelet transform. Following [17], the blur kernel is setted to be $h_{ij} = \frac{1}{i^2 + j^2}$. Here, we use two well - known test images, Lena and Cameraman. And we compare the proposed method With GPSR-BB algorithm [17]. Both the paper and the code of the GPSR-BB algorithm are available at <http://www.lx.it.pt/~mtf/GPSR/>. For the GPSR algorithm, we use the demo "demo_image_deblur.m" with default settings for its parameters except parameter "tolA". For implementation of the discrete wavelet transform, both algorithms use the Rice wavelet toolbox that it can be freely downloaded from <http://www-dsp.rice.edu/software/rwt.shtml>. Throughout this part, we use the AFISTA-BB algorithm with parameters $\tilde{\lambda} = 1 \times 10^{-7}$, $L = 1 \times 10^{+3}$ and

Table 9 Image deconvolution test: Comparison of the AFISTA-BB with the GPSR-BB [17] for different tolerances

GPSR [17]				Proposed			
"tolA"	Psnr	$N^o Iter$	Time(sec.)	ϵ	Psnr	$N^o Iter$	Time(sec.)
Cameraman Test Image							
1e-03	26.96	21	0.875	1e-03	28.99	20	1.098
1e-05	27.88	50	2.000	1e-04	39.83	103	3.922
1e-07	28.09	144	5.953	1e-05	57.60	402	14.27
Lena Test Image							
1e-03	33.02	13	2.531	1e-03	33.48	17	2.797
1e-05	33.58	28	5.469	1e-04	39.50	81	12.55
1e-07	33.69	82	15.28	1e-05	52.35	440	66.44



Fig. 13 Image deconvolution test: Size of the blur kernel is $[9, 9]$. Top left: Original image. Top right: Blurred Image. Bottom left: reconstructed using GPSR-BB algorithm with Psnr 28.09. Bottom right: reconstructed using AFISTA-BB with Psnr 57.60



Fig. 14 Image deconvolution test: Size of the blur kernel is $[17, 17]$. Top left: Original image. Top right: Blurred Image. Bottom left: reconstructed using GPSR-BB algorithm with Psnr 32.49. Bottom right: reconstructed using AFISTA-BB with Psnr 50.54

the stopping criterion $\frac{\|Au^k - b\|}{\|b\|} < \epsilon$. The results of Image Deconvolution experiments are reported at Table 9 and Figs. 13 and 14, which show the efficiency of the proposed method. Note that, in this test, the quality of reconstructed images using the GPSR method is less than the proposed method. The psnr of reconstructed images shows this fact. For example, from Table 9, we can see that, for Lena test image GPSR-BB reached psnr=33.69 in 15.28 seconds whereas AFISTA-BB reached psnr=39.50 in 12.55 seconds. So, AFISTA-BB can reach a better quality of reconstruction in less time compared to GPSR-BB. Thus, from this test, we can see that the AFISTA method overcomes the GPSR-BB method from the perspective of convergence rate and accuracy.

5 Conclusion

This paper proposes AFISTA to accelerate the FISTA method using a specific continuation strategy. Also, the convergence analysis of the proposed method is presented. Although the main computational cost of these two methods at each iteration remains almost the same, however thanks to the continuation strategy AFISTA method is significantly faster than the FISTA method. Furthermore, to improve its convergence rate, we furnished the AFISTA method with BB stepsize. In the sequel, AFISTA is specialized for the orthogonal case i.e. $AA^T = I$. A fundamental difference between the proposed method and the FISTA method is that the proposed method solves BP problem while the FISTA method solves LASSO problem. As mentioned above, AFISTA is applied on various tasks of signal and image processing and compared with other methods such as generalizations of linearized Bregman, fixed - point continuation (FPC), split Bregman, and Gradient projection for sparse reconstruction (GPSR) methods. Numerical results show that in most cases, the proposed method in both the convergence rate and the quality of reconstruction is better than other methods. The main limitation of the proposed method is that its parameters need to set beforehand. The application of AFISTA on medical image processing is our future work.

References

1. Becker S, Bobin J, Candes EJ (2011) NESTA: A fast and accurate first-order method for sparse recovery. *SIAM J Imaging Sci* 4(1):1–39
2. Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J Imaging Sci* 2(1):183–202
3. Beck A, Teboulle M (2010) Gradient-based algorithms with applications to signal recovery problems. *Convex optimization in signal processing and communications*. Cambridge University Press, Cambridge, pp 42–88
4. Barzilai J, Borwein J (1988) Two point step size gradient methods. *IMA J Numer Anal* 8:141–148
5. Ben-Tal A, Nemirovski A (2001) Lectures on modern convex optimization: Analysis, algorithms, and engineering applications. *MPS/SIAM Ser. Optim*. SIAM
6. Boyd S, Vandenberghe L. (2004) *Convex optimization*. Cambridge University Press, New York
7. Bredies K, Lorenz D (2008) Linear convergence of iterative soft-thresholding. *J Fourier Anal Appl* 14:813–837
8. Bruck RJ (1977) On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in Hilbert space. *J Math Anal Appl* 61:159–164
9. Candes E, Romberg J, Tao T (2006) Stable signal recovery from incomplete and inaccurate information. *SPure Appl Math* 59:1207–1233
10. Chambolle A, Dossal Ch (2015) On the convergence of the iterates of the “Fast Iterative Shrinkage/Thresholding Algorithm” *J Opt Theory Appl* 166(3):968–982

11. Chen S, Donoho D Basis pursuit. IEEE, Proceedings of 1994 28th Asilomar conference on signals, systems and computers
12. Chen S, Yang M (2017) An adaptive fast iterative shrinkage threshold algorithm. In: 2017 29th Chinese control and decision conference (CCDC). IEEE, pp 2190–2194
13. Combettes PL, Wajs VR (2005) Signal recovery by proximal forward-backward splitting. *Multiscale Model Simul* 4:1168–1200
14. Donoho D (2006) Compressed sensing. *IEEE Trans Inform Theory* 52:1289–1306
15. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. *Ann Statist* 32:407–499
16. Facchinei F, Pang JS (2003) Finite-dimensional variational inequalities and complementarity problems. Vol. II, Springer Ser. Oper. Res. Springer, New York
17. Figueiredo MA, Nowak R, Wright SJ (2007) Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing* 1(4):586–597
18. Goldstein T, Osher S (2009) The split Bregman method for L1-regularized problems. *SIAM J Imaging Sciences* 2(2):323–343
19. Hale ET, Yin W, Zhang Y (2007) A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. Technical Report-Rice University
20. Hale ET, Yin W, Zhang Y (2008) Fixed-point continuation for l_1 -minimization: Methodology and convergence. *SIAM J Optim* 19:1107–1130
21. Huang B, Ma S. h., Goldfarb D (2013) Accelerated linearized Bregman method. *J Sci Comput* 54:428–453
22. Islam SMR, Huang X, Ou KL (2015) Image compression based on compressive sensing using wavelet lifting scheme. *The International Journal of Multimedia & Its Applications* 7(1)
23. Jiajun H, Russell B, Downton J (2017) Seismic spectral decomposition via basis pursuit. SEG Technical program expanded abstracts 2017. Society of exploration geophysicists: 2174–2179
24. Karimi S, Vavasis S (2017) IMRO A proximal quasi-Newton method for solving l1-regularized least squares problems. *SIAM J Optim* 27(2):583–615
25. Lorenz DA, Pfetsch ME, Tillmann AM (2014) An infeasible-point subgradient method using adaptive approximate projections. *Comput Optim Appl* 57(2):271–306
26. Lorenz DA, Pfetsch ME, Tillmann AM (2015) Solving basis pursuit: Heuristic optimality check and solver comparison. *ACM Trans Math Softw (TOMS)* 41(2):1–29
27. Malioutov DM, Cetin M, Willsky AS (2005) Homotopy continuation for sparse signal representation. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing, vol 5, pp 733–736
28. Osborne MR, Presnell B, Turlach BA (2000) A new approach to variable selection in least squares problems. *IMA J Numer Anal* 20(3):389–402
29. Osher S, Mao Y, Dong B, Yin W (2010) Fast linearized Bregman iteration for compressive sensing and sparse denoising. *Commun Math Sci* 8(1):93–111
30. Parikh N, Boyd S (2013) Proximal algorithms. *Foundations and Trends in Optimization* 1(3):123–231
31. Passty GB (1979) Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *J Math Anal Appl* 72:383–390
32. Rockafellar RT (1970) *Convex analysis*. Princeton University Press, Princeton
33. Tanay S, Srivastava S. h., Khare S, Stanimirovic PS, Petkovic MD (2019) An improved algorithm for basis pursuit problem and its applications. *Appl Math Comput* 355:385–398
34. Tao Z. h. (2020) Accelerating monotone fast iterative shrinkage–thresholding algorithm with sequential subspace optimization for sparse recovery. *SIViP* 14(4):771–780
35. Tao S, Boley D, Zhang S (2016) Local linear convergence of ISTA and FISTA on the LASSO problem. *SIAM J OPTIM* 26(1):313–336
36. Tesfamicael SA, Barzideh F (2015) Clustered compressive sensing: Application on medical imaging. *Int J Electr Comput Eng* 5(1)
37. Tesfamicael S, Barzideh F (2014) Clustered compressed sensing in fMRI data analysis using a Bayesian framework. *Int J Inf Electr Eng* 4(2)
38. Vahi I, Shahri PK, Ahani H (2020) A compressed-sensing-based compressor for ECG. *Biomed Eng Lett*: 1–9
39. van den Berg E (2020) A hybrid quasi-Newton projected-gradient method with application to lasso and basis-pursuit denoising. *Math Program Comput* 12(1):1–38
40. Wen Z, Yin W, Goldfarb D, Zhang Y (2010) A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. *SIAM J Sci Comput* 32:1832–1857
41. Yin W (2010) Analysis and generalizations of the linearized Bregman method. *SIAM J Imaging Sciences* 3(4):856–877

42. Yin W, Osher S, Goldfarb D, Darbon J (2008) Bregman iterative algorithms for L_1 -minimization with applications to compressed sensing. *SIAM J Imaging Sci* 1:143–168

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.