Calcolo (2019) 56:42
https://doi.org/10.1007/s10092-019-0340-5

Signal reconstruction by conjugate gradient algorithm based on smoothing l_1-norm

Caiying Wu[1] · Jiaming Zhan[1] · Yue Lu[2] · Jein-Shan Chen[3]

Received: 12 January 2019 / Accepted: 9 October 2019 / Published online: 14 October 2019

Abstract
The l_1-norm regularized minimization problem is a non-differentiable problem and has a wide range of applications in the field of compressive sensing. Many approaches have been proposed in the literature. Among them, smoothing l_1-norm is one of the effective approaches. This paper follows this path, in which we adopt six smoothing functions to approximate the l_1-norm. Then, we recast the signal recovery problem as a smoothing penalized least squares optimization problem, and apply the nonlinear conjugate gradient method to solve the smoothing model. The algorithm is shown globally convergent. In addition, the simulation results not only suggest some nice smoothing functions, but also show that the proposed algorithm is competitive in view of relative error.

Keywords l_1-norm regularization · Compressive sensing · Conjugate gradient algorithm · Smoothing function

Mathematics Subject Classification 90C33

Caiying Wu: The research is supported by the Natural Science Foundation of Inner Mongolia Autonomous Region (2018MS01016)
Jiaming Zhan: The research is supported by the Natural Science Foundation of Inner Mongolia Autonomous Region (2018MS01016)
Yue Lu: The research is supported by National Natural Science Foundation of China (Grant No. 11601389), Doctoral Foundation of Tianjin Normal University (Grant No. 52XB1513), 2017-Outstanding Young Innovation Team Cultivation Program of Tianjin Normal University (Grant No. 135202TD1703) and Program for Innovative Research Team in Universities of Tianjin (Grant No. TD13-5078).
Jein-Shan Chen: The research is supported by Ministry of Science and Technology, Taiwan.

✉ Jein-Shan Chen
jschen@math.ntnu.edu.tw

Extended author information available on the last page of the article

1 Introduction

In this paper, we focus on the topic of reconstructing signals, which is one of the most important applications in compressive sensing [8,9,16]. There exist numerous textbooks and articles related to this topic and there is no need to repeat its importance and applications here. Therefore, we get into its mathematical model directly and convey our new idea for tackling this problem. Mathematically, the noise-free signal reconstruction problem model is described as follows:

$$\begin{aligned} &\min\ \|x\|_0 \\ &\text{s.t.}\ \ b = Ax, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the original sparse signal that needs to be recovered, $A \in \mathbb{R}^{m\times n}$ $(m \ll n)$ is the measurement matrix, $b \in \mathbb{R}^m$ is the observation vector, and $\|x\|_0$ represents the l_0-norm of x, which is defined as the number of nonzero components of x. Without loss of generality, we assume that there exists a positive constant $K \le n$ such that $\|x\|_0 = K$. Besides this condition, if the measurement matrix A satisfies Restricted Isometry Property (RIP) of order K, then we can recover the signal x more accurately through the model (1), see [6–10] for more details.

Unfortunately, the l_0-norm minimization problem (1) is an NP-hard combinatorial optimization problem. In order to avoid this difficulty, researchers attempt to replace l_0-norm by l_1-norm in model (1) and obtain the following l_1 minimization problem

$$\begin{aligned} &\min\ \|x\|_1 \\ &\text{s.t.}\ \ b = Ax, \end{aligned} \tag{2}$$

where $\|x\|_1$ denotes the l_1 norm of x and $\|x\|_1 = \sum_{i=1}^n |x_i|$. Under the RIP condition, the model (2) has the same solution as (1). In practice, the probed signal b is usually impacted by noise, therefore there arises investigation on the noise signal reconstruction problem:

$$\begin{aligned} &\min\ \|x\|_1 \\ &\text{s.t.}\ \ b = Ax + e, \end{aligned} \tag{3}$$

where $e \in \mathbb{R}^m$ denotes the noise. In order to deal with (3), researchers prefer to consider the associated penalized least squares optimization problem

$$\min \lambda\|x\|_1 + \frac{1}{2}\|b - Ax\|_2^2 \tag{4}$$

where $\lambda > 0$ is the penalty factor. In the sequel, we call (4) the l_1-norm regularized problem.

Until now, there are plenty of numerical algorithms for solving the model (4) and some first-order algorithms have drawn much attention during the past decades. Due the huge amount of literature, we only outline and recall some approaches as below. The gradient projection algorithm [18] is one of the earliest gradient-based algorithms for solving (4), in which it was reformulated as a box constrained quadratic program and solved by the gradient projection algorithm. Recently, the most extensively investigated first order algorithm for the solution of (4) was the iterative shrinkage/thresholding (IST) algorithm [15,21]. Their results triggered off many contributions based on this method [17,24,25]. In light of the IST algorithm, many variants are proposed under different optimization reformulations and techniques. For instances, Hale, Yin and Zhang [19,20] presented an IST fixed point continuation (FPC) method by an operator splitting skill. Wright, Nowak and Figueiredo [29] studied the spaRSA for sparse reconstruction from solving nonsmooth convex problem. Experimental results show that the accelerated IST methods (two IST [4] and fast IST [1]) have better convergence properties. In addition, the famous NESTA [2] first proposed the smoothing function of the l_1-norm and then used Nesterov's gradient method to get the solution of (3). Besides these methods, the gradient projection technique and alternating direction ideas are also considered, see [3,5,30] for more details.

Because the simplicity and lower storage requirements, conjugate gradient algorithms are suitable for large scale problems. In this paper, like [28,31,33], we are interested in applying the conjugate gradient method for signal recovery. We use six smoothing functions, which are studied in [23,27] for absolute value function, to approximate the l_1-norm. Accordingly, we reformulate the signal recovery problem as a smoothing penalized least squares optimization problem, and apply the nonlinear conjugate gradient method [32] to solve the smoothing model. The proposed algorithm is shown globally convergent. Moreover, we report some numerical experiments to demonstrate the effectiveness of our method. Numerical comparisons with "NESTA", "FPCBB", and "FISTA", which are well-known open softwares, are presented as well.

2 Preliminary

This section recalls some ideas about smoothing technique which can be found in [11,12,23] and references therein.

It is well known that the absolute value function $|t|$ ($t \in \mathbb{R}$) is not smooth at zero. In order to overcome this difficulty, we introduce some smoothing functions of $|t|$ used in the sequel.

Definition 2.1 The function $\psi : \mathbb{R}_{++} \times \mathbb{R} \to \mathbb{R}$ is a smoothing function of $|t|$, if the following two conditions hold:

(a) ψ is continuously differentiable at $(\mu, t) \in \mathbb{R}_{++} \times \mathbb{R}$;
(b) $\lim_{\mu \downarrow 0} \psi(\mu, t) = |t|$ for any $t \in \mathbb{R}$.

How to construct smoothing functions for $|t|$? First, we observe that $|t|$ can be divided into two parts:

$$|t| = (t)_+ - (t)_- = (t)_+ + (-t)_+,$$

where $(t)_+$ denotes the plus function, i.e., $(t)_+ = \max\{0, t\}$, and $(t)_- = \min\{0, t\}$. As mentioned in [11,12,23], one can follow the below procedure to construct a smoothing function for $(t)_+$. More specifically, through importing a density (kernel) function $d(t)$ with finite number of pieces satisfying

$$d(t) \geq 0 \quad \text{and} \quad \int_{-\infty}^{+\infty} d(t)dt = 1,$$

one can define

$$\hat{s}(t,\mu) := \frac{1}{\mu} d\left(\frac{t}{\mu}\right),$$

where μ is a positive parameter. If the following condition holds

$$\int_{-\infty}^{+\infty} |t|\, d(t)\, dt < +\infty,$$

then the function $\hat{p}(t,\mu)$ defined as

$$\hat{p}(t,\mu) = \int_{-\infty}^{+\infty} (t-s)_+ \hat{s}(s,\mu)ds = \int_{-\infty}^{t} (t-s)\hat{s}(s,\mu)ds \approx (t)_+$$

is a smoothing approximation for $(t)_+$. There are existing well-known smoothing functions for the plus function [11,22,23], for example,

$$\hat{\psi}_1(\mu,t) = t + \mu \ln\left(1 + e^{-\frac{t}{\mu}}\right), \tag{5}$$

$$\hat{\psi}_2(\mu,t) = \begin{cases} t & \text{if} \quad t \geq \frac{\mu}{2}, \\ \frac{1}{2\mu}\left(t + \frac{\mu}{2}\right)^2 & \text{if} \ -\frac{\mu}{2} < t < \frac{\mu}{2}, \\ 0 & \text{if } t \leq -\frac{\mu}{2}, \end{cases} \tag{6}$$

$$\hat{\psi}_3(\mu,t) = \frac{\sqrt{4\mu^2 + t^2} + t}{2}, \tag{7}$$

$$\hat{\psi}_4(\mu,t) = \begin{cases} t - \frac{\mu}{2} & \text{if } t > \mu, \\ \frac{t^2}{2\mu} & \text{if } 0 \leq t \leq \mu, \\ 0 & \text{if } t < 0. \end{cases} \tag{8}$$

where the corresponding kernel functions are respectively given by

$$d_1(t) = \frac{e^{-x}}{(1+e^{-x})^2},$$

$$d_2(t) = \begin{cases} 1 & \text{if } -\frac{1}{2} \leq x \leq \frac{1}{2}, \\ 0 & \text{otherwise}, \end{cases}$$

$$d_3(t) = \frac{2}{(x^2+4)^{\frac{3}{2}}},$$

$$d_4(t) = \begin{cases} 1 & \text{if } 0 \le x \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

Likewise, we can achieve the smoothing function of $|t|$ via convolution as follows:

$$\hat{p}(|t|, \mu) = \hat{p}(t, \mu) + \hat{p}(-t, \mu) = \int_{-\infty}^{+\infty} |t - s| \, \hat{s}(s, \mu) ds.$$

Analogous to (5)–(8), we construct four smoothing functions for $|t|$ as [23]:

$$\psi_1(\mu, t) = \mu \left[\ln\left(1 + e^{-\frac{t}{\mu}}\right) + \ln\left(1 + e^{\frac{t}{\mu}}\right) \right], \tag{9}$$

$$\psi_2(\mu, t) = \begin{cases} t & \text{if } t \ge \frac{\mu}{2}, \\ \frac{t^2}{\mu} + \frac{\mu}{4} & \text{if } -\frac{\mu}{2} < t < \frac{\mu}{2}, \\ -t & \text{if } t \le -\frac{\mu}{2}, \end{cases} \tag{10}$$

$$\psi_3(\mu, t) = \sqrt{4\mu^2 + t^2}, \tag{11}$$

$$\psi_4(\mu, t) = \begin{cases} \frac{t^2}{2\mu} & \text{if } |t| \le \mu, \\ |t| - \frac{\mu}{2} & \text{if } |t| > \mu. \end{cases} \tag{12}$$

In particular, if we take a Epanechnikov kernel function

$$d(t) = \begin{cases} \frac{3}{4}(1 - t^2) & \text{if } |t| \le 1, \\ 0 & \text{if otherwise,} \end{cases}$$

then the associated smoothing function for $|t|$ is expressed by

$$\psi_5(\mu, t) = \begin{cases} t & \text{if } t > \mu, \\ -\frac{t^4}{8\mu^3} + \frac{3t^2}{4\mu} + \frac{3\mu}{8} & \text{if } -\mu \le t \le \mu, \\ -t & \text{if } t < -\mu. \end{cases} \tag{13}$$

Moreover, taking a Gaussian kernel function $d(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$ for all $t \in \mathbb{R}$ yields

$$\hat{s}(t, \mu) := \frac{1}{\mu} d\left(\frac{t}{\mu}\right) = \frac{1}{\sqrt{2\pi\mu^2}} e^{-\frac{t^2}{2\mu^2}},$$

which leads to another type of smoothing function [27] for $|t|$:

$$\psi_6(\mu, t) = t\,\mathrm{erf}\left(\frac{t}{\sqrt{2}\mu}\right) + \sqrt{\frac{2}{\pi}} \mu e^{-\frac{t^2}{2\mu^2}}, \tag{14}$$

where the error function is defined as follows:

$$\mathrm{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-u^2} du \quad \forall t \in \mathbb{R}.$$

In summary, we have obtained six smoothing functions and will employ them to approximate the l_1-norm for signal recovery problem.

3 Algorithm and convergence properties

This section is devoted to the detailed description and implementation of our algorithmic idea and its global convergence results. Basically, it is a type of conjugate gradient algorithm. To proceed, we briefly review the conjugate gradient method. Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and the target problem is

$$\min_{x \in \mathbb{R}^n} f(x).$$

Let d^k denote the search direction and α_k be a step length. Then, the general framework of conjugate gradient method is as below.

$$\begin{aligned} x^{k+1} &= x^k + \alpha_k d^k, \\ d^k &= \begin{cases} -\nabla f(x^k) & \text{if } k = 0, \\ -\nabla f(x^k) + \beta_k d^{k-1} & \text{if } k > 0, \end{cases} \end{aligned} \tag{15}$$

where ∇f is the gradient of f and β_k is a parameter. In general, various choices of parameters β_k represent (correspond to) different conjugate gradient algorithms. Among them, the three-term PRP (Polak–Ribiere–Polyak) conjugate gradient method is the most efficient, where

$$\begin{aligned} d^k &= \begin{cases} -\nabla f(x^k) & \text{if } k = 0, \\ -\nabla f(x^k) + \beta_k d^{k-1} - \theta_k y^{k-1} & \text{if } k > 0, \end{cases} \\ y^{k-1} &= \nabla f(x^k) - \nabla f(x^{k-1}), \\ \beta_k &= \frac{(\nabla f(x^k))^T y^{k-1}}{\|\nabla f(x^{k-1})\|^2}, \\ \theta_k &= \frac{(\nabla f(x^k))^T d^{k-1}}{\|\nabla f(x^{k-1})\|^2}. \end{aligned} \tag{16}$$

The utmost feature of this conjugate gradient method is to ensure the sufficient descent property of direction at each iteration, which plays a prominent role in the global convergence analysis. For more details, please refer to [32] and references therein.

For convenience, we denote

$$f_\mu(x) := \lambda \phi_i(\mu, x) + \frac{1}{2}\|b - Ax\|_2^2, \quad x \in \mathbb{R}^n, \tag{17}$$

where $\phi_i(\mu, x)$ is a smoothing function of l_1 norm $\|x\|_1$. In other words, it corresponds to the format as below:

$$\phi_i(\mu, x) := \sum_{j=1}^{n} \psi_i(\mu, x_j), \quad i = 1, 2, 3, 4, 5, 6. \tag{18}$$

Thus, the problem (4) is equivalent to the smoothing penalized least squares optimization problem:

$$\min_{x \in \mathbb{R}^n} f_\mu(x) \tag{19}$$

which is our new target problem. To deal with the unconstrained minimization (19), the following iterative scheme is employed, see [32].
The Algorithm

> *Step* 0 Given starting point $x^0 \in \mathbb{R}^n$ and $\mu_0 > 0$. Choose parameters ρ, δ, γ, and $\bar{\gamma} \in (0, 1)$. Let $d^0 = -\nabla f_{\mu_0}(x^0)$. Set $k = 0$.
> *Step* 1 If termination criterion is reached, then stop.
> *Step* 2 Determine $\alpha_k = \max\{\rho^j, j = 0, 1, 2, \ldots\}$ satisfying
>
> $$f_{\mu_k}(x^k + \alpha_k d^k) \le f_{\mu_k}(x^k) - 2\delta(1-\gamma)\alpha_k f_{\mu_k}(x^k).$$
>
> *Step* 3 Let $x^{k+1} = x^k + \alpha_k d^k$ and $\mu_{k+1} = \bar{\gamma}\mu_k$. Replace f with $f_{\mu_{k+1}}$ in formula (16) and compute d^{k+1} by (16).
> *Step* 4 Set $k = k + 1$ and return to step 1.

We also say a few words about how to update μ_k in Step 3. For Experiment 1, μ_k is updated at each iteration by $\mu_{k+1} = \bar{\gamma}\mu_k$, $\bar{\gamma} \in (0, 1)$ and we select $\bar{\gamma} = 0.4$. For Experiment 2 and Experiment 3, an inner loop is used to solve our relaxed problem for μ_k, and then reduces μ_k. This is indeed a technique appeared in NESTA. When μ_k is small enough, our algorithm no longer updates the parameter μ_k.

Lemma 3.1 *Suppose that the function $f_\mu(x)$ is defined by* (17). *Then, there exists a constant $L > 0$ such that $\left\|\nabla f_\mu(x) - \nabla f_\mu(y)\right\| \le L\|x - y\|$ for all $x, y \in \mathbb{R}^n$.*

Proof For any fixed $\mu > 0$, in order to prove the Lipschitz property of $\nabla f_\mu(x)$, we need to verify the Lipschitz condition of $\psi_i{}'$ $(i = 1, 2, 3, 4, 5, 6)$. To this end, we discuss two cases.
Case (i): $i = 1, 3, 5, 6$. For any $t_1, t_2 \in \mathbb{R}$, without losing of generality, let $t_1 < t_2$, by Lagrange Mean Value Theorem, we have

$$\left|\psi_i^{'}(\mu, t_1) - \psi_i^{'}(\mu, t_2)\right| = \left|\psi_i^{''}(\mu, \xi)\right| |t_1 - t_2|, \quad \xi \in (t_1, t_2).$$

For subsequent analysis, we need to estimate $\left|\psi_i^{''}(\mu, \xi)\right|$ for each $i = 1, 3, 5, 6$.
For $i = 1$, we know that

$$|\psi_1^{''}(\mu, \xi)| = \frac{1}{\mu}\left[\frac{e^{\frac{\xi}{\mu}}}{(1 + e^{\frac{\xi}{\mu}})^2} + \frac{e^{\frac{-\xi}{\mu}}}{(1 + e^{\frac{-\xi}{\mu}})^2}\right] < \frac{2}{\mu}.$$

For $i = 3$, it is clear that $\left|\psi_3^{''}(\mu, \xi)\right| = \frac{4\mu^2}{(4\mu^2+\xi^2)^{3/2}} < \frac{1}{2\mu}$.
For $i = 5$, we have

$$\psi_5^{'}(\mu, t) = \begin{cases} 1 & \text{if } t > \mu, \\ -\frac{t^3}{2\mu^3} + \frac{3t}{2\mu} & \text{if } -\mu \le t \le \mu, \\ -1 & \text{if } t < -\mu. \end{cases} \quad \psi_5^{''}(\mu, t) = \begin{cases} 0 & \text{if } t > \mu, \\ -\frac{3t^2}{2\mu^3} + \frac{3}{2\mu} & \text{if } -\mu \le t \le \mu, \\ 0 & \text{if } t < -\mu. \end{cases}$$

which yields

$$\left|\psi_5^{''}(\mu, \xi)\right| < \frac{3\mu^2}{2\mu^3} + \frac{3}{2\mu} = \frac{3}{\mu}.$$

For $i = 6$, we compute

$$\psi_6^{'}(\mu, t) = \frac{2}{\sqrt{\pi}} \int_0^{\frac{t}{\sqrt{2}\mu}} e^{-u^2} du, \quad \psi_6^{''}(\mu, t) = \frac{\sqrt{2}}{\sqrt{\pi}\mu} e^{\frac{-t^2}{2\mu^2}},$$

which imply $\left|\psi_6^{''}(\mu, \xi)\right| < \frac{\sqrt{2}}{\sqrt{\pi}\mu}$.
All the aforementioned results indicate that

$$\left|\psi_i^{'}(\mu, t_1) - \psi_i^{'}(\mu, t_2)\right| \le \frac{3}{\mu}|t_1 - t_2|, \quad i = 1, 3, 5, 6. \tag{20}$$

for any $t_1, t_2 \in \mathbb{R}$.
Case (ii): $i = 2, 4$. Indeed, we will provide a version like (20) for $\psi_2^{'}$ and $\psi_4^{'}$.
For $i = 2$, we know that

$$\psi_2^{'}(\mu, t) = \begin{cases} 1 & \text{if } t \ge \frac{\mu}{2}, \\ \frac{2t}{\mu} & \text{if } -\frac{\mu}{2} < t < \frac{\mu}{2}, \\ -1 & \text{if } t \le -\frac{\mu}{2}. \end{cases}$$

If $t_1 \ge \frac{\mu}{2}, t_2 \ge \frac{\mu}{2}, t_1 \le -\frac{\mu}{2}, t_2 \le -\frac{\mu}{2}$ or $t_1, t_2 \in (-\frac{\mu}{2}, \frac{\mu}{2})$, then

$$\left|\psi_2^{'}(\mu, t_1) - \psi_2^{'}(\mu, t_2)\right| \le \frac{2}{\mu}|t_1 - t_2|.$$

If $t_1 \ge \frac{\mu}{2}, t_2 \le -\frac{\mu}{2}$, then

$$\left|\psi_2^{'}(\mu, t_1) - \psi_2^{'}(\mu, t_2)\right| = 2 = \mu\frac{2}{\mu} \le \frac{2}{\mu}|t_1 - t_2|.$$

If $t_1 \ge \frac{\mu}{2}, t_2 \in (-\frac{\mu}{2}, \frac{\mu}{2})$, then

$$\left|\psi_2^{'}(\mu, t_1) - \psi_2^{'}(\mu, t_2)\right| = 1 - \frac{2t_2}{\mu} < \frac{2t_1}{\mu} - \frac{2t_2}{\mu} = \frac{2}{\mu}|t_1 - t_2|.$$

If $t_1 \le -\frac{\mu}{2}, t_2 \in (-\frac{\mu}{2}, \frac{\mu}{2})$, then

$$\left|\psi_2^{'}(\mu, t_1) - \psi_2^{'}(\mu, t_2)\right| = 1 + \frac{2t_2}{\mu} < -\frac{2t_1}{\mu} + \frac{2t_2}{\mu} = \frac{2}{\mu}|t_1 - t_2|.$$

Thus, it is clear to conclude that

$$\left|\psi_2^{'}(\mu, t_1) - \psi_2^{'}(\mu, t_2)\right| \le \frac{2}{\mu}|t_1 - t_2|, \quad \forall\, t_1, t_2 \in \mathbb{R}.$$

For $i = 4$, using the similar arguments for case of $i = 2$, it can be verified that

$$\left|\psi_4^{'}(\mu, t_1) - \psi_4^{'}(\mu, t_2)\right| \le \frac{1}{\mu}|t_1 - t_2|, \quad \forall\, t_1, t_2 \in \mathbb{R}.$$

In summary, we also achieve

$$\left|\psi_i^{'}(\mu, t_1) - \psi_i^{'}(\mu, t_2)\right| \le \frac{2}{\mu}|t_1 - t_2|, \quad i = 2, 4. \tag{21}$$

for any $t_1, t_2 \in \mathbb{R}$.
Now, applying (20) and (21), for any $x, y \in \mathbb{R}^n$, we have

$$\begin{aligned}
&\left\|\nabla f_\mu(x) - \nabla f_\mu(y)\right\| \\
&= \left\|\lambda \left[\nabla\phi_i(\mu, x) - \nabla\phi_i(\mu, y)\right] + A^T(Ax - b) - A^T(Ay - b)\right\| \\
&\le \frac{3}{\mu}\lambda n\|x - y\| + \|A\|^2\|x - y\| \\
&= L\|x - y\|,
\end{aligned}$$

where $L = \frac{3}{\mu}\lambda n + \|A\|^2$. Thus, the proof is complete. □

Lemma 3.2 *For $\mu > 0$, the level set $L(x^0) = \{x \in \mathbb{R}^n \mid f_\mu(x) \le f_\mu(x^0)\}$ is bounded.*

Proof We prove it by contradiction. Suppose that the set $L(x^0)$ is unbounded. Then, there exists an index set K_1, such that $\|x^k\| \to \infty$, $k \to \infty$, $k \in K_1$. Recalling the definition of $f_\mu(x)$, we have $f_\mu(x^k) \to \infty$, $k \to \infty$, $k \in K_1$. This contradicts $f_\mu(x^k) \le f_\mu(x^0)$. Thus, the level set $L(x^0)$ is bounded. □

In fact, since the continuity of $f_\mu(x)$, we find the level set $L(x^0)$ is compact. We point out that Lemmas 3.1 and 3.2 play key roles in our theoretical part. They were assumed as two assumptions in [32] and other literature like [14]. Here we assert them by showing that function (17) based on the proposed smoothing functions satisfies these assumptions. With these properties, we are ready to present the global convergence of our algorithm.

Theorem 3.1 *For any $\mu > 0$, consider the aforementioned algorithm with any starting point x^0. Let $\{x^k\}$ be the sequence generated by the algorithm. Then*

$$\liminf_{k\to\infty} \|\nabla f_\mu(x^k)\| = 0. \tag{22}$$

Proof Suppose that the conclusion (22) is not true. Then, there exists a constant $\varepsilon_0 > 0$, such that

$$\|\nabla f_\mu(x^k)\| \geq \varepsilon_0, \quad \forall\, k. \tag{23}$$

Since $(\nabla f_\mu(x^k))^T d^k = -\|\nabla f_\mu(x^k)\|^2$, there exists $\alpha_k > 0$, such that

$$f_\mu(x^k + \alpha_k d^k) \leq f_\mu(x^k) - 2\delta(1-\gamma)\alpha_k f_\mu(x^k). \tag{24}$$

This means $\{f_\mu(x^k)\}$ is decreasing and bounded, which implies that $x^k \in L(x^0)$ and $\{f_\mu(x^k)\}$ is convergent. We denote that $\lim_{k\to\infty} f_\mu(x^k) = f_*$. From (24), it can be verified that $\lim_{k\to\infty} \alpha_k f_\mu(x^k) = 0$. By the definition of $f_\mu(x)$, we know $f_* > 0$, and therefore $0 < \alpha_k = \frac{\alpha_k f_\mu(x^k)}{f_\mu(x^k)} \leq \frac{\alpha_k f_\mu(x^k)}{f_*}$. To sum up, we obtain

$$\lim_{k\to\infty} \alpha_k = 0. \tag{25}$$

Now, applying Lemma 3.2, there is a constant $\overline{r} > 0$, such that

$$\|\nabla f_\mu(x^k)\| \leq \overline{r}, \quad \forall\, k. \tag{26}$$

Next, we prove $\{\|d^k\|\}$ is bounded by a contradiction argument. If $\{\|d^k\|\}$ is unbounded, then there exists an index K_2, such that $\|d^k\| \to \infty$, $k \to \infty$, $k \in K_2$. Let θ_k be the angle between $-\nabla f_\mu(x^k)$ and d^k. Then, we see that

$$\cos\theta_k = \frac{-(\nabla f_\mu(x^k))^T d^k}{\|\nabla f_\mu(x^k)\|\|d^k\|} = \frac{\|\nabla f_\mu(x^k)\|}{\|d^k\|}. \tag{27}$$

This relation enables us to apply $\varepsilon_0 \leq \|\nabla f_\mu(x^k)\| \leq \overline{r}$ to conclude $\cos\theta_k \to 0$, $k \in K_2, k \to \infty$, which means $\theta_k \to \frac{\pi}{2}, k \in K_2, k \to \infty$. Considering this geometric relationship, we have $(\nabla f_\mu(x^k))^T d^k \to 0$, $k \in K_2$, $k \to \infty$, from which we find

$$-\|\nabla f_\mu(x^k)\|^2 = (\nabla f_\mu(x^k))^T d^k \to 0,\ k \in K_2, \quad k \to \infty,$$

which contradicts (23). Thus, $\{\|d^k\|\}$ is bounded, i.e., there exists a constant $M^* > \varepsilon_0$, such that

$$\|d^k\| \leq M^*, \quad \forall\, k. \tag{28}$$

Then, combining (25) together with (28) gives

$$\lim_{k\to\infty} \alpha_k \|d^k\| = 0. \tag{29}$$

In addition, from (27), we have $\cos\theta_k \geq \frac{\varepsilon_0}{M^*}$, which further yields

$$-(\nabla f_\mu(x^k))^T d^k = \|\nabla f_\mu(x^k)\| \|d^k\| \cos\theta_k \geq \frac{\varepsilon_0^2}{M^*}\|d^k\|. \tag{30}$$

Applying the mean value theorem, we obtain

$$\begin{aligned} f_\mu(x^k + \alpha_k d^k) &= f_\mu(x^k) + \alpha_k (\nabla f_\mu(\xi^k))^T d^k \\ &= f_\mu(x^k) + \alpha_k (\nabla f_\mu(x^k))^T d^k + \alpha_k (\nabla f_\mu(\xi^k) - \nabla f_\mu(x^k))^T d^k \\ &\leq f_\mu(x^k) + \alpha_k \|d^k\| \Big(\frac{(\nabla f_\mu(x^k))^T d^k}{\|d^k\|} + \|\nabla f_\mu(\xi^k) - \nabla f_\mu(x^k)\| \Big), \end{aligned} \tag{31}$$

where ξ^k is between x^k and $x^k + \alpha_k d^k$. Using Lemma 3.1 and the compact property of level set $L(x^0)$ imply that $\nabla f_\mu(x)$ is uniformly continuous on $L(x^0)$. This together and (29) implies that there exists a constant $\widehat{\alpha} > 0$, for sufficiently large k, such that $\alpha_k \|d^k\| < \widehat{\alpha}$ and

$$\|\nabla f_\mu(\xi^k) - \nabla f_\mu(x^k)\| < \frac{1}{2}\frac{\varepsilon_0^2}{M^*}. \tag{32}$$

Then, when k is sufficiently large, from (31), we have

$$\begin{aligned} f_\mu(x^k + \alpha_k d^k) &\leq f_\mu(x^k) + \widehat{\alpha}\Big(-\frac{\varepsilon_0^2}{M^*} + \frac{1}{2}\frac{\varepsilon_0^2}{M^*} \Big) \\ &= f_\mu(x^k) - \frac{\hat{\alpha}\varepsilon_0^2}{2M^*}, \end{aligned} \tag{33}$$

which contradicts $f_\mu(x^{k+1}) - f_\mu(x^k) \to 0$, $k \to \infty$. Therefore, there must hold

$$\liminf_{k\to\infty} \|\nabla f_\mu(x^k)\| = 0.$$

□

4 Numerical experiments

In this section, we conduct numerical experiments to show the performance of our algorithm for signal recovery. All tests are run in MATLAB R20116 on a 64-bit PC with an Intel (R) Core(TM) i7-6500U of 2.50 GHz CPU and 8.00 GB of RAM equipped with Windows 7 operating system.

Experiment 1
In what follows, we divide our tests into two groups: noise-free case and noise case. we report the numerical results of our algorithm for signal recovery. In all of our simulations, $A \in \mathbb{R}^{m\times n}$ is assumed to be Gaussian matrix and $m = \frac{n}{2}, \frac{n}{4}$, $x^* \in \mathbb{R}^n$ is a K-sparse original signal with $K = \frac{n}{40}$, whose nonzero component satisfies normal distribution $N(0, 1)$. Moreover, we set $b = Ax + e$ in the noisy case, where e is the Gaussian noise with zero mean and variance σ^2.

The parameters of our algorithm are summarized as follows:

$$x^0 = \text{zeros}(n, 1),\ \lambda = 0.001 * \|A^T b\|_\infty,\ \rho = 0.5,\ \delta = 0.002,\ \gamma = 0.2,\ \sigma^2 = 0.0001.$$

In Tables 1 and 2, we set $\mu_0 = 0.1$, $\mu_{k+1} = 0.4\mu_k$ and the algorithm terminates when the relative error

$$\frac{\|\overline{x} - x^*\|}{\|x^*\|} < 4 \times 10^{-3},$$

where $\overline{x}$ is the reconstruction signal. In Tables 3 and 4, F denotes the smoothing functions, CPU denotes cpu time, Re denotes the relative error and It denotes the iterations, we set $\mu = 1.0\,e^{-2}, 1.0\,e^{-3}, 1.0\,e^{-4}, 1.0\,e^{-5}$ and the algorithm terminates when

$$\frac{|f_\mu(x^{k+1}) - f_\mu(x^k)|}{|f_\mu(x^{k+1})|} < 1.0\,e^{-12}.$$

Note that the result from each test is obtained by running our algorithm 10 times and taking its average result.

In our experiments, we found that ψ_1 is very unstable, sometimes it is not applicable to solve the test problems. Similar concern had also been addressed in [13, page 135].

Table 1 Comparisons of five smoothing functions for noise-free case

CPU time							Iterations				
n	m	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6
2000	$n/2$	1.21	0.94	1.32	1.27	0.92	72	74	75	73	69
	$n/4$	1.37	0.90	1.45	1.59	1.00	129	135	142	148	147
4000	$n/2$	4.16	3.83	5.01	4.50	3.80	70	70	78	72	68
	$n/4$	4.68	4.12	5.19	5.58	4.50	131	143	143	148	146
6000	$n/2$	9.54	8.49	9.80	9.22	8.61	70	70	70	69	68
	$n/4$	9.69	10.35	10.71	12.91	10.54	134	163	145	159	164
8000	$n/2$	16.93	16.25	18.76	18.91	15.55	67	71	74	78	69
	$n/4$	15.76	15.65	18.65	19.49	15.75	126	137	145	144	138
10000	$n/2$	25.88	24.06	30.07	25.83	24.21	67	67	76	68	68
	$n/4$	27.94	28.28	29.96	34.48	27.60	139	138	151	160	149

Table 2 Comparisons of five smoothing functions for noisy case

CPU time							Iterations				
n	m	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6
2000	$n/2$	1.27	1.00	1.30	1.33	1.00	73	78	75	75	72
	$n/4$	1.54	0.92	1.64	1.55	0.91	145	151	154	150	144
4000	$n/2$	4.48	4.47	4.89	4.62	4.36	72	80	75	72	76
	$n/4$	4.94	5.16	5.77	5.09	4.65	137	159	154	143	158
6000	$n/2$	8.81	9.07	9.80	9.91	8.97	65	72	72	67	69
	$n/4$	11.65	11.61	12.10	12.38	10.12	160	167	152	165	154
8000	$n/2$	16.27	17.05	18.28	18.53	16.69	69	75	76	74	71
	$n/4$	15.76	18.04	19.27	17.99	16.91	132	155	145	145	143
10000	$n/2$	25.59	25.58	26.04	25.82	25.38	68	71	67	67	68
	$n/4$	25.82	27.23	37.00	29.52	27.73	136	150	175	150	155

Accordingly, we try to adopt the equivalent expression suggested in [13, formula (3.1)] and rewrite $\psi_1(t)$ as below:

$$\begin{aligned}\psi_1(t) &= \mu\left[\ln\left(e^{\frac{\lambda_1(A(t))}{\mu}}+e^{\frac{\lambda_2(A(t))}{\mu}}\right)+\ln\left(e^{\frac{\lambda_1(B(t))}{\mu}}+e^{\frac{\lambda_2(B(t))}{\mu}}\right)\right]\\ &= \lambda_1(A(t))+\mu\left(e^{\frac{\lambda_2(A(t))-\lambda_1(A(t))}{\mu}}\right)+\lambda_1(B(t))+\mu\left(e^{\frac{\lambda_2(B(t))-\lambda_1(B(t))}{\mu}}\right)\end{aligned}$$

where

$$A(t)=\begin{bmatrix}0 & 0\\ 0 & -t\end{bmatrix},\quad B(t)=\begin{bmatrix}0 & 0\\ 0 & t\end{bmatrix},$$
$$\lambda_1(A(t))=0,\quad \lambda_2(A(t))=-t,\quad \lambda_1(B(t))=0,\quad \lambda_2(B(t))=t.$$

Nonetheless, ψ_1 still does not work well along with the algorithm. We therefore omit it in our numerical reports.

From the experiments, we have the following observations:

(1) The sparse original signal can be recovered by our algorithm effectively. Tables 1, 2, 3, 4 and Figs. 1, 2, 3, 4, 5 illustrate that the five smoothing functions (except ψ_1) in our algorithm work quite well whether the given problem is noise-free or noisy.

(2) The convergence behavior with respect to CPU time, iterations, signal size, and the relative error are depicted in Figs. 1, 2, 3, 4, 5, respectively. From Tables 1, 2 and Figs. 1, 2, we see that our algorithm costs more cpu time when increasing the dimension n, while the changes of their iterations are very marginal. On the whole, the performance order can be summarized as below:

$$\psi_2 \geq \psi_4 \approx \psi_5 \approx \psi_6 > \psi_3$$

Table 3 Comparisons of five functions for different μ when $n = 5000$ and noise-free case

$\mu = 1.0e^{-2}$					$\mu = 1.0e^{-3}$			$\mu = 1.0e^{-4}$			$\mu = 1.0e^{-5}$		
F	m	CPU	Re	It	CPU	Re	It	CPU	Re	It	CPU	Re	It
ψ_2	$n/2$	8.80	0.0158	92	9.34	0.0040	89	11.03	0.0033	100	16.78	0.0025	141
	$n/4$	10.98	0.0365	172	11.49	0.0064	178	11.83	0.0029	184	17.06	0.0028	237
ψ_3	$n/2$	9.44	0.0567	104	8.14	0.0090	86	7.98	0.0032	86	10.93	0.0025	111
	$n/4$	7.44	0.1532	167	9.10	0.0195	179	8.36	0.0045	166	10.34	0.0030	188
ψ_4	$n/2$	8.22	0.0538	71	7.92	0.0059	73	10.58	0.0027	93	12.01	0.0025	104
	$n/4$	9.13	0.0157	139	11.11	0.0154	155	10.69	0.0035	179	13.43	0.0032	194
ψ_5	$n/2$	7.88	0.0419	68	7.59	0.0055	72	10.14	0.0025	91	12.77	0.0025	109
	$n/4$	10.26	0.1895	132	11.09	0.0117	156	10.51	0.0041	169	12.83	0.0034	193
ψ_6	$n/2$	8.95	0.0360	96	7.94	0.0063	84	9.69	0.0027	99	12.57	0.0023	121
	$n/4$	9.17	0.0926	166	9.16	0.0128	173	8.98	0.0038	174	11.75	0.0030	203

Table 4 Comparisons of five smoothing functions for different μ when $n = 5000$ and noisy case

$\mu = 1.0e^{-2}$					$\mu = 1.0e^{-3}$			$\mu = 1.0e^{-4}$			$\mu = 1.0e^{-5}$		
F	m	CPU	Re	It	CPU	Re	It	CPU	Re	It	CPU	Re	It
ψ_2	$n/2$	8.62	0.0159	89	8.77	0.0072	88	11.82	0.0026	100	15.80	0.0025	143
	$n/4$	11.67	0.0379	174	10.40	0.0063	180	10.26	0.0033	188	15.03	0.0033	242
ψ_3	$n/2$	8.54	0.0548	100	7.86	0.0085	88	8.75	0.0033	92	10.38	0.0025	109
	$n/4$	7.43	0.1506	169	8.43	0.0185	169	7.78	0.0045	177	9.82	0.0030	203
ψ_4	$n/2$	6.76	0.0491	64	8.11	0.0058	76	11.40	0.0028	95	11.93	0.0026	105
	$n/4$	11.10	0.1439	154	8.84	0.0137	152	9.67	0.0035	175	11.52	0.0035	184
ψ_5	$n/2$	7.36	0.0542	65	7.82	0.0059	75	9.57	0.0031	87	11.33	0.0025	104
	$n/4$	10.19	0.1492	143	10.00	0.0123	154	10.48	0.0039	177	11.37	0.0034	191
ψ_6	$n/2$	8.54	0.0355	94	8.78	0.0061	92	9.05	0.0029	96	11.57	0.0026	119
	$n/4$	7.77	0.0908	160	9.15	0.0122	180	8.11	0.0040	167	10.50	0.0030	207

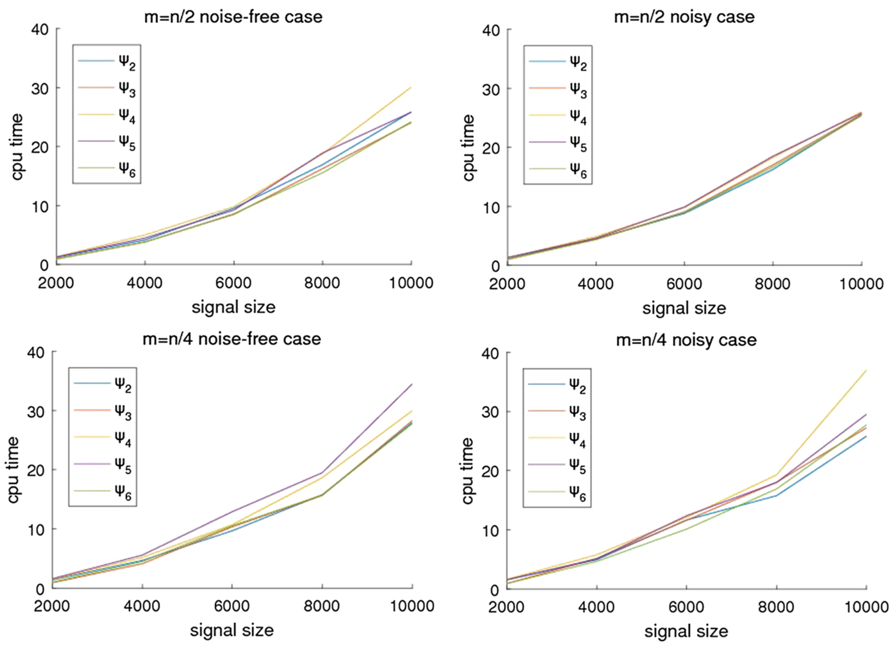

Fig. 1 The convergence behavior: cpu time versus signal size (n)when $\mu_{k+1} = 0.4\mu_k$

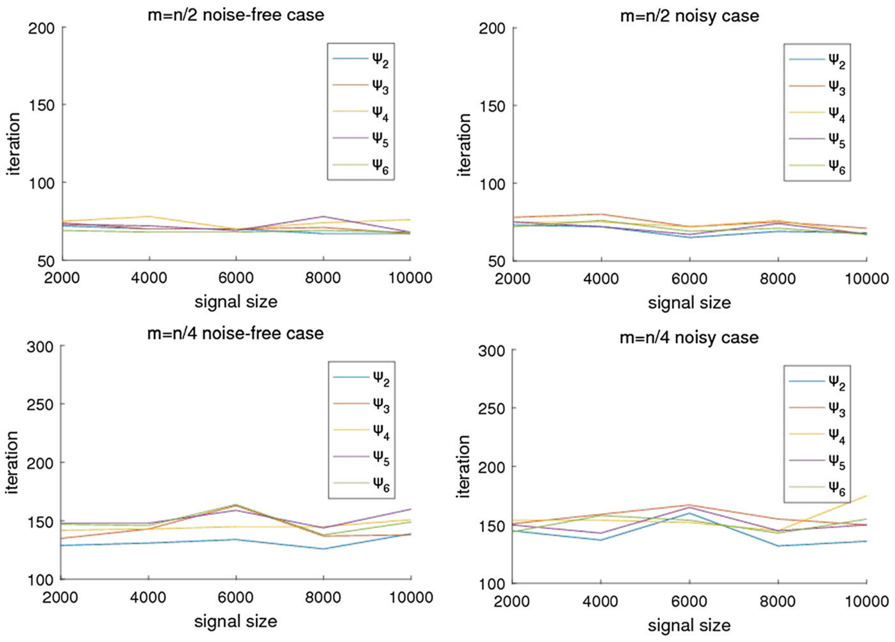

Fig. 2 The convergence behavior: iterations versus signal size (n) when $\mu_{k+1} = 0.4\mu_k$

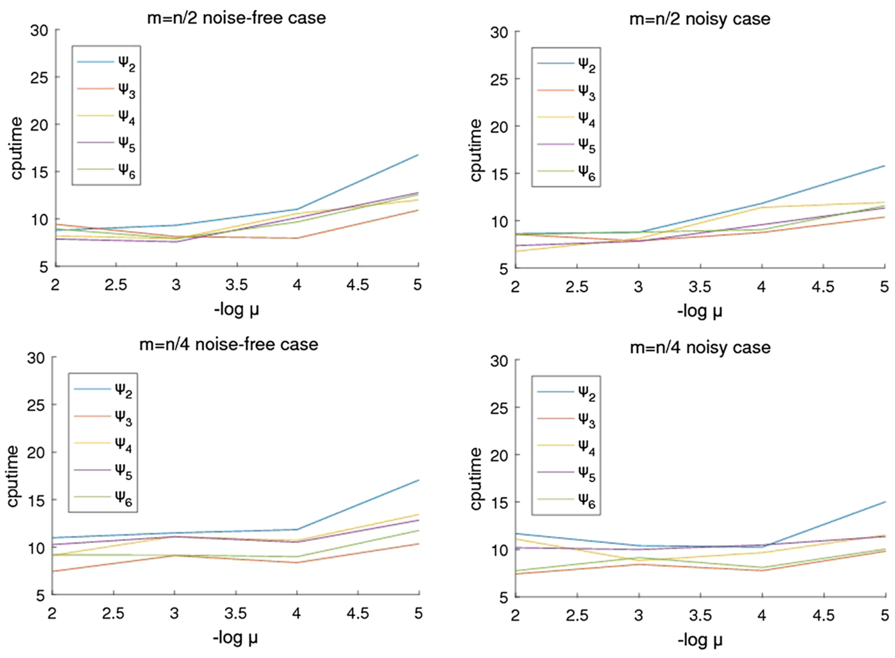

Fig. 3 The convergence behavior: cpu time versus $-\log \mu$ with $n = 5000$

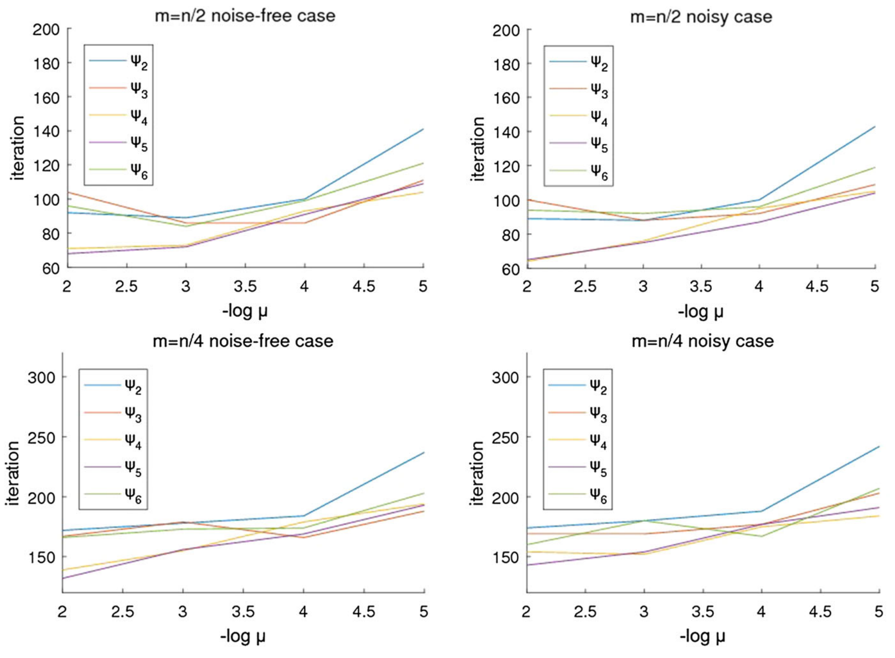

Fig. 4 The convergence behavior: iterations versus $-\log \mu$ with $n = 5000$

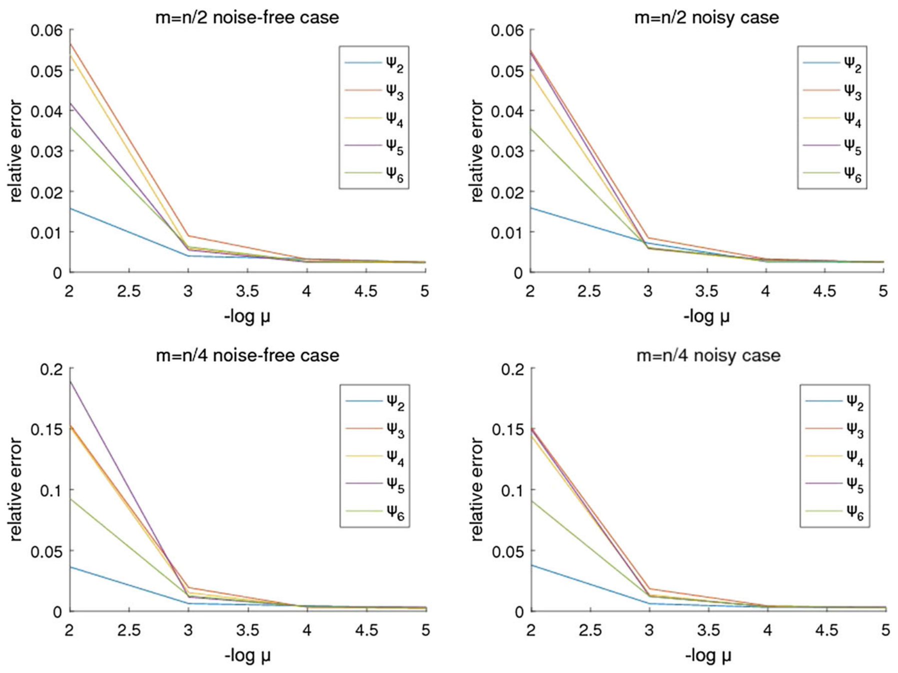

Fig. 5 The convergence behavior: relative error versus $-\log \mu$ with $n = 5000$

where "$\geq$" means performs better than and "$\approx$" means there is no difference in numerical performance.

(3) According to Tables 3, 4 and Figs. 3, 4, 5, the smaller the parameter μ, the better the signal to recover; and the more cpu time and iterations have to spend accordingly. For any fixed μ, although the function ψ_2 sometimes takes a bit more cpu time and iterations, it has a lower relative error. In view of this, we could conclude that the function ψ_2 works best along with the proposed algorithm. From Fig. 5, the function ψ_6 also has a lower relative error, which means ψ_6 is a possible good choice as well. To sum up, in view of relative error, we have

$$\psi_2 \geq \psi_4 \approx \psi_5 \approx \psi_6 > \psi_3.$$

Experiment 2
Next, we try to do comparison with "NESTA", which is a well-known, fast, open software proposed in [2]. In the noisy case, $b = Ax + e$, where e is the Gaussian noise with zero mean and variance σ^2. In all of our simulations, $A \in \mathbb{R}^{m\times n}$ is assumed to be Gaussian matrix, where $n = 2^{13}$ and $m = \frac{n}{4}$. We select K-sparse original signal $x^* \in \mathbb{R}^n$ with $K = \frac{n}{40}$, its nonzero component satisfies normal distribution $N(0, 1)$. In our test, we set

$$x^0 = A^T b,\ \lambda = 0.0048 * \|A^T b\|_\infty,\ \rho = 0.5,\ \delta = 0.002,\ \gamma = 0.2,\ \sigma^2 = 0.0001.$$

Table 5 Comparisons of five smoothing functions and NESTA for different tol when noise-free

tol = $1.0e^{-4}$				tol = $1.0e^{-5}$			tol = $1.0e^{-6}$			tol = $1.0e^{-7}$			tol = $1.0e^{-8}$		
F	CPU	Re	It	CPU	Re	It	CPU	Re	It	CPU	Re	It	CPU	Re	It
ψ_2	23.77	9.62E−04	150	28.30	1.98E−05	177	35.65	2.11E−05	213	46.62	1.91E−05	256	59.97	1.58E−05	342
ψ_3	21.82	6.37E−03	142	24.55	3.03E−05	171	30.88	2.29E−05	206	36.54	2.01E−05	241	40.39	1.76E−05	286
ψ_4	26.64	7.00E−03	146	27.03	4.32E−05	169	36.79	2.21E−05	215	45.12	2.04E−05	258	59.34	1.76E−05	338
ψ_5	26.32	2.21E−03	157	28.04	2.77E−05	174	34.73	2.12E−05	209	45.38	2.14E−05	256	49.89	1.88E−05	296
ψ_6	19.53	4.82E−03	138	24.83	3.01E−05	172	32.41	2.34E−05	214	38.80	2.06E−05	245	50.09	1.68E−05	301
NESTA	7.96	2.28E−02	555	8.30	2.95E−02	568	17.12	3.26E−04	1134	15.98	1.82E−05	1060	44.50	1.74E−05	2999

Table 6 Comparisons of five smoothing functions and NESTA for different tol when noisy case

$\text{tol} = 1.0e^{-4}$				$\text{tol} = 1.0e^{-5}$			$\text{tol} = 1.0e^{-6}$			$\text{tol} = 1.0e^{-7}$			$\text{tol} = 1.0e^{-8}$		
F	CPU	Re	It	CPU	Re	It	CPU	Re	It	CPU	Re	It	CPU	Re	It
ψ_2	23.25	2.49E−03	146	26.10	7.37E−04	165	36.67	5.91E−04	222	62.27	5.80E−04	332	104.44	5.95E−04	542
ψ_3	19.53	5.57E−03	138	19.02	4.24E−03	136	31.62	5.83E−04	215	53.04	5.99E−04	332	80.03	5.83E−04	459
ψ_4	25.11	3.52E−03	162	23.84	8.90E−03	153	37.37	5.39E−04	224	56.38	5.65E−04	314	104.55	5.68E−04	520
ψ_5	24.72	5.91E−03	161	26.40	7.22E−04	164	35.52	6.31E−04	214	57.01	5.75E−04	317	83.13	5.83E−04	429
ψ_6	20.45	1.23E−02	148	22.16	6.80E−04	156	33.50	5.63E−04	222	56.77	5.85E−04	308	94.43	5.75E−04	507
NESTA	7.67	2.13E−02	525	8.28	1.10E−02	549	15.81	7.85E−04	1086	16.45	7.47E−04	1073	85.89	6.16E−04	5886

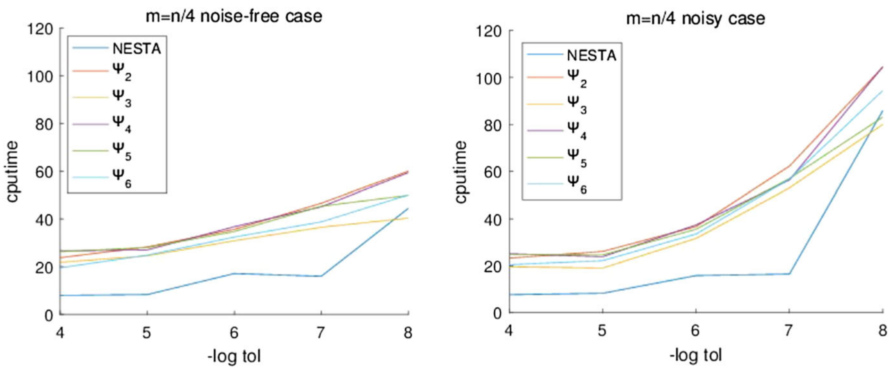

Fig. 6 The convergence behavior: CPU time versus accuracy

We also accelerate our algorithm by using

$$\mu_{k+1} = \omega\mu_k,\ 1 \le k \le t,\ \omega = (\frac{\mu_t}{\mu_0})^{1/t},\ \mu_t = 10^{-8}, \mu_0 = 10^{-2},\ \text{and}\ t = 4.$$

The algorithm terminates when

$$\frac{|f_\mu(x^k) - \bar{f}_\mu(x^k)|}{\bar{f}_\mu(x^k)} < \text{tol}, \quad \bar{f}_\mu(x^k) = \frac{1}{\min\{5,k\}} \sum_{l=1}^{\min\{5,k\}} f_\mu(x^{k-l}).$$

In our experiments, the accuracy tol $\in \{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$. We test the continuation version NESTA for comparison. In testing NESTA, other parameters are taken as default expect for the above parameters.

From the comparison experiments, we have the following observations:

(1) In the view of CPU time, the NESTA is really fast. However, from Tables 5, 6 and Fig. 6, as the "accuracy tol" gets smaller, the CPU time of our algorithm is not affected too much. To the contrast, when the "accuracy tol" gets smaller, the CPU time of NESTA goes up dramatically.

(2) From Tables 5, 6 and Fig. 7, we see that our algorithm has a small relative error when the "accuracy tol" is relatively large. Especially, when tol $= 10^{-4}$ and tol $= 10^{-5}$, the relative error of NESTA gets large, which indicates that the signal has not been recovered well. To the contrast, we see that the function ψ_2 keeps a small error, which means it has better performance.

(3) From Tables 5, 6 and Fig. 8, NESTA needs more iterations. In view of this and relative error, our proposed algorithm is competitive.

Experiment 3
In order to observe the efficiency on sparse recovery. We compare our algorithm with three other algorithms: the NESTA, the FPCBB and the FISTA, in which Bernoulli matrix, Partial Hadamard matrix and Gaussian matrix are chosen. In our implementations, we select $n = 2^{14}$, $m = \frac{n}{2}$ and sparsity is $K = \frac{n}{40}$. The parameters μ_k, δ and γ are the same as the Experiment 2. See Figs. 9, 10, 11.

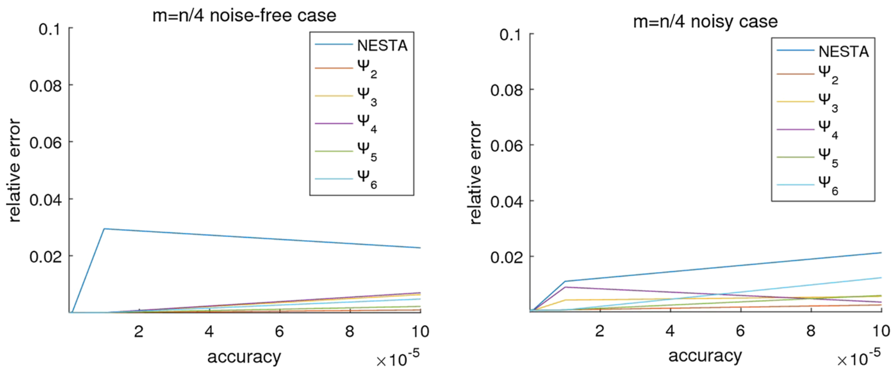

Fig. 7 The convergence behavior: relative error versus accuracy

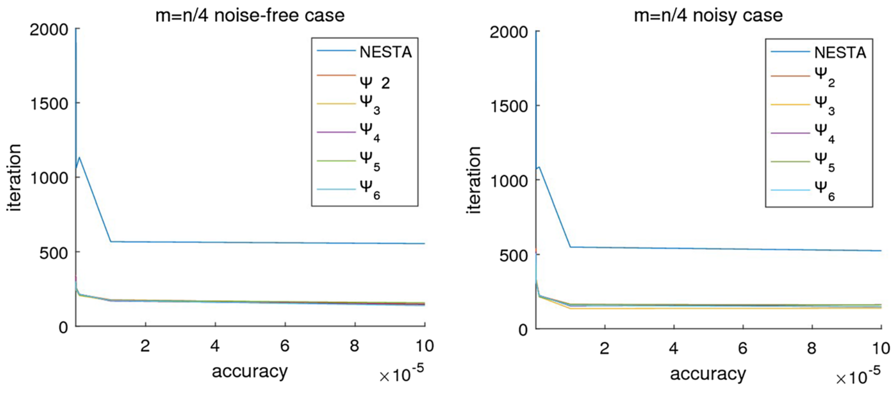

Fig. 8 The convergence behavior: iterations versus accuracy

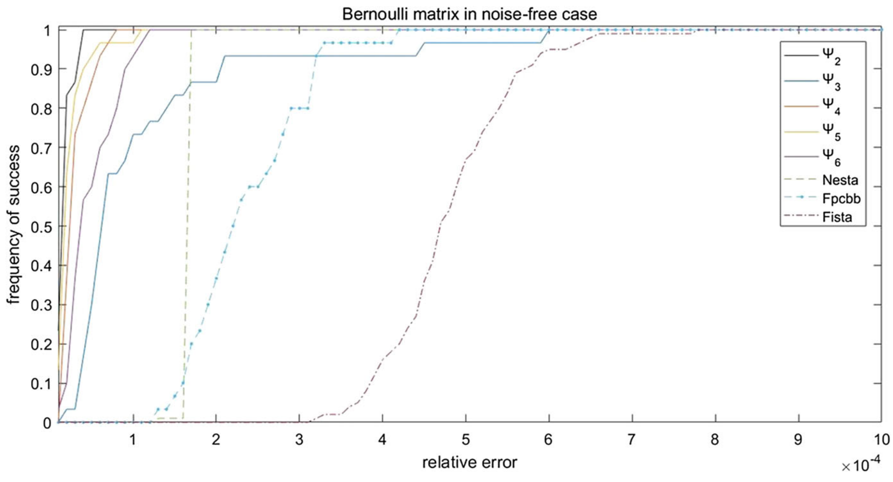

Fig. 9 Comparisons among algorithms with Bernoulli matrix

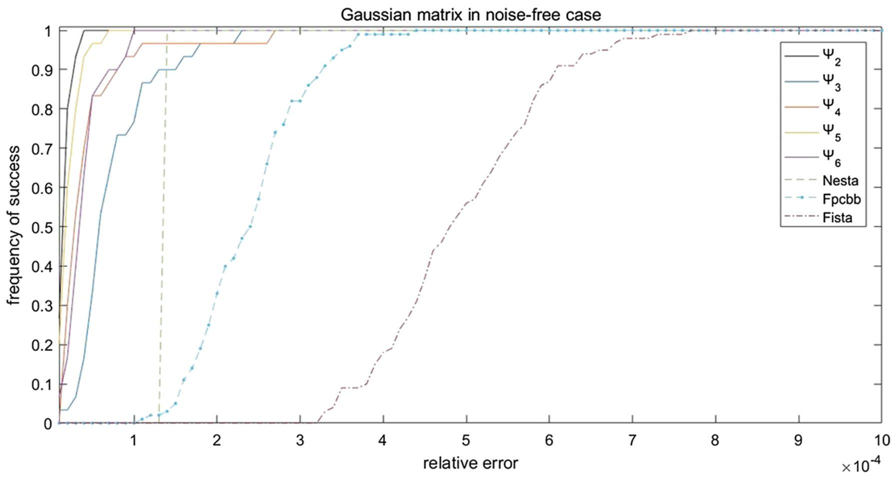

Fig. 10 Comparisons among algorithms with Gaussian matrix

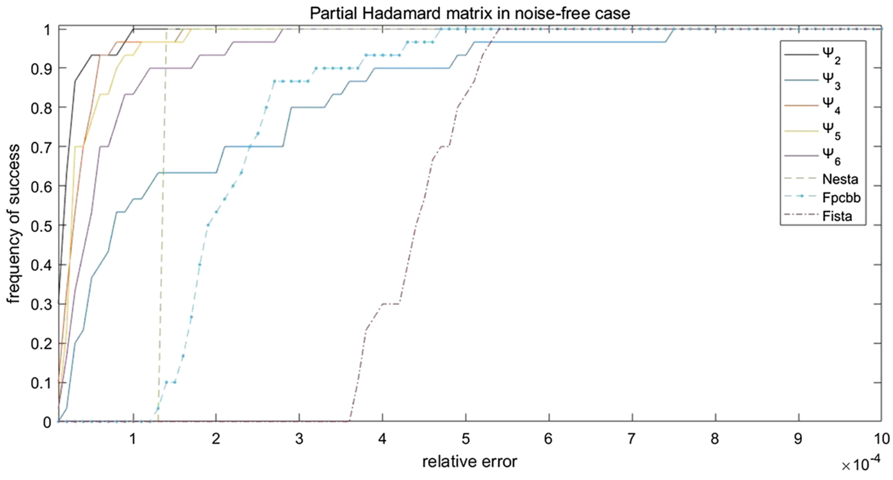

Fig. 11 Comparisons among algorithms with Partial Hadamard matrix

The result from each test is obtained by running our algorithm 30 times and taking its average result. Figures 9, 10, 11 show the frequency of successful reconstruction at different relative error. We can see that our algorithm gives relatively better behavior when relative error is small enough. This means that our algorithm has slightly higher efficiency on sparse recovery. To sum up, we obtain

$$\psi_2 \geq \psi_4 \approx \psi_5 \approx \psi_6 > \psi_3.$$

5 Conclusions

In this paper, the sparse signal recovery problem is investigated, in which we concentrate on the l_1-norm model. In light of convolution techniques in [23,27] and the framework of nonlinear conjugate gradient method [32], we propose an unified smoothing approach to solve the given sparse signal reconstruction problem. As a byproduct, the classical l_1 norm is approximated by six smoothing functions. Numerical results show that ψ_2 and ψ_6 are the better choices of smoothing function to work with the algorithm. Although the theoretical part is not very notable, the contribution of this paper lies in numerical comparisons among new smoothing functions. In particular, we suggest two nice smoothing functions to work along with nonlinear conjugate gradient method. In addition, we compare our algorithm with three other algorithms (well-known open softwares): the NESTA, the FPCBB and the FISTA, in which Bernoulli matrix, Partial Hadamard matrix and Gaussian matrix are chosen. It can be seen that our algorithm gives relatively better behavior when relative error is small enough. Under this sense, we provide a new choice of simple and effective approach to deal with the recovery of the original sparse signal problem.

Our attention was drawn to [14] by one reviewer. In [14], the authors proposed a smoothing conjugate gradient method, and then used this algorithm to solve the image restoration problem. More specifically, it used a smoothing function as below:

$$s_\mu(t) = \begin{cases} |t| & \text{if } |t| > \frac{\mu}{2}, \\ \frac{t^2}{\mu} + \frac{\mu}{4} & \text{if } |t| \leq \frac{\mu}{2}. \end{cases}$$

This smoothing function is exactly the $\psi_2(\mu, t)$ function in our paper. Our contribution lies in showing that this smoothing function is also a good choice for signal reconstruction problem comparing to other smoothing functions, which are not investigated in reference [14].

Acknowledgements The authors would like to thank the anonymous referee and the editor for their valuable comments, viewpoints, and suggestions, which help improve the manuscript a lot.

References

1. Beck, A., Teboulle, M.: A fast iterative shrinkage–thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
2. Becker, S., Bobin, J., Candes, E.: NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imaging Sci. **4**, 1–39 (2011)
3. Berg, E., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. **31**, 890–912 (2008)
4. Bioucas-Dias, J.M., Figueiredo, M.: A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Trans. Image Process. **16**, 2992–3004 (2007)
5. Broughton, R.L., Coope, I.D., Renaud, P.F., Tappenden, R.E.H.: A box constrained gradient projection algorithm for compressed sensing. Signal Process. **91**, 1985–1992 (2011)
6. Candes, E.J.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique **346**(9–10), 589–592 (2008)
7. Candes, E.J., Tao, T.: Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)

8. Candes, E.J., Tao, T.: Near optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006)
9. Candes, E.J., Romberg, J.K., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
10. Candes, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
11. Chen, C., Mangasarian, O.L.: A class of smoothing functions for nonlinear and mixed complementarity problems. Comput. Optim. Appl. **5**, 97–138 (1996)
12. Chen, X.: Smoothing methods for nonsmooth, nonconvex minimization. Math. Program. **134**, 71–99 (2012)
13. Chen, X., Womersley, R.S., Ye, J.: Minimizing the condition number of Gram matrix. SIAM J. Optim. **21**(1), 127–148 (2011)
14. Chen, X., Zhou, W.: Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization. SIAM J. Imaging Sci. **3**(4), 765–790 (2010)
15. De Mol, C., Defrise, M.: A note on wavelet-based inversion algorithms. Contemp. Math. **313**, 85–96 (2002)
16. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
17. Elad, M.: Why simple shrinkage is still relevant for redundant representations? IEEE Trans. Inf. Theory **52**(12), 5559–5569 (2006)
18. Figueiredo, M., Nowak, R., Wright, S.J.: Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Process. **1**, 586–597 (2007)
19. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for l_1-minimization: methodology and convergence. SIAM J. Optim. **19**, 1107–1130 (2008)
20. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation applied to compressed sensing: implementation and numerical experiments. J. Comput. Math. **28**, 170–194 (2010)
21. Nowak, R., Figueiredo, M.: Fast wavelet-based image deconvolution using the EM algorithm. In: Proceedings of the 35th Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 371–375 (2001)
22. Qi, L., Sun, D.: Smoothing functions and smoothing Newton method for complementarity and variational inequality problems. J. Optim. Theory Appl. **113**, 121–147 (2001)
23. Saheya, B., Yu, C.H., Chen, J.-S.: Numerical comparisons based on four smoothing functions for absolute value equation. J. Appl. Math. Comput. **56**(1–2), 131–149 (2018)
24. Starck, J.L., Candes, E., Donoho, D.: Astronomical image representation by the curvelet transform. Astron. Astrophys. **398**, 785–800 (2003)
25. Starck, J.L., Nguyen, M., Murtagh, F.: Wavelets and curvelets for image deconvolution: a combined approach. Signal Process. **83**, 2279–2283 (2003)
26. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. **58**, 267–268 (1996)
27. Voronin, S., Ozkaya, G., Yoshida, D.: Convolution based smooth approximations to the absolute value function with application to non-smooth regularization. arXiv:1408.6795v2 [math.NA] 1, (July 2015)
28. Wang, X., Liu, F., Jiao, L.C., Wu, J., Chen, J.: Incomplete variables truncated conjugate gradient method for signal reconstruction in compressed sensing. Inf. Sci. **288**, 387–411 (2014)
29. Wright, S., Nowak, R., Figueiredo, M.: Sparse reconstruction by separable approximation. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (October 2008)
30. Yang, J., Zhang, Y.: Alternating direction algorithms for l_1-problems in compressive sensing. SIAM J. Sci. Comput. **33**, 250–278 (2011)
31. Yin, K., Xiao, Y.H., Zhang, M.L.: Nonlinear conjugate gradient method for l_1-norm regularization problems in compressive sensing. J. Comput. Inf. Syst. **7**, 880–885 (2011)
32. Zhang, L., Zhou, W., Li, D.: A descent modified Polak–Ribiere–Polyak conjugate gradient method and its global convergence. IMA J. Numer. Anal. **26**, 629–640 (2006)
33. Zhu, H., Xiao, Y.H., Wu, S.Y.: Large sparse signal recovery by conjugate gradient algorithm based on smoothing technique. Comput. Math. Appl. **66**, 24–32 (2013)

Affiliations

Caiying Wu[1] · Jiaming Zhan[1] · Yue Lu[2] · Jein-Shan Chen[3]

Caiying Wu
wucaiyingyun@163.com

Jiaming Zhan
zhanjiaming1220@qq.com

Yue Lu
jinjin403@sina.com

1 College of Mathematics Science, Inner Mongolia University, Hohhot 010021, China

2 School of Mathematical Sciences, Tianjin Normal University, Tianjin 300387, China

3 Department of Mathematics, National Taiwan Normal University, Taipei 11677, Taiwan