Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 619123, 13 pages
http://dx.doi.org/10.1155/2013/619123

Research Article

An Adaptive Prediction-Correction Method for Solving Large-Scale Nonlinear Systems of Monotone Equations with Applications

Gaohang Yu,[1] Shanzhou Niu,[2] Jianhua Ma,[2] and Yisheng Song[3]

[1] *School of Mathematics and Computer Sciences, Gannan Normal University, Ganzhou 341000, China*
[2] *School of Biomedical Engineering, Southern Medical University, Guangzhou 510515, China*
[3] *Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong*

Correspondence should be addressed to Yisheng Song; songyisheng123@yahoo.com.cn

Received 21 February 2013; Accepted 10 April 2013

Academic Editor: Guoyin Li

Combining multivariate spectral gradient method with projection scheme, this paper presents an adaptive prediction-correction method for solving large-scale nonlinear systems of monotone equations. The proposed method possesses some favorable properties: (1) it is progressive step by step, that is, the distance between iterates and the solution set is decreasing monotonically; (2) global convergence result is independent of the merit function and its Lipschitz continuity; (3) it is a derivative-free method and could be applied for solving large-scale nonsmooth equations due to its lower storage requirement. Preliminary numerical results show that the proposed method is very effective. Some practical applications of the proposed method are demonstrated and tested on sparse signal reconstruction, compressed sensing, and image deconvolution problems.

1. Introduction

Considering the problem to find solutions of the following nonlinear monotone equations:

$$g(x) = 0, \tag{1}$$

where $g : \Re^n \to \Re^n$ is a continuous and monotone, that is, $\langle g(x) - g(y), x - y\rangle \geq 0$ for all $x, y \in \Re^n$.

Nonlinear monotone equations arise in many practical applications such as ballistic trajectory computation [1] and vibration systems [2], the first-order necessary condition of the unconstrained convex optimization problem, and the subproblems in the generalized proximal algorithms with Bregman distances [3]. Moreover, we can convert some monotone variational inequality into systems of nonlinear monotone equations by means of fixed point maps or normal maps [4] if the underlying function satisfies some coercive conditions. Solodov and Svaiter [5] proposed a projection method for solving (1). A nice property of the projection method is that the whole sequence of iterates is always globally convergent to a solution of the system without any additional regularity assumptions. Moreover, Zhang and Zhou [6] presented a spectral gradient projection (SG) method for solving systems of monotone equations which combines a modified spectral gradient method and projection method. This method is shown to be globally convergent if the nonlinear monotone equations is Lipschitz continuous. Xiao et al. [7] proposed a spectral gradient method to minimize a nonsmooth minimization problem, arising from spare solution recovery in compressed sensing, consisting of a least-squares data-fitting term and a ℓ_1-norm regularization term. This problem is firstly formulated as a convex quadratic program (QP) problem and then reformulated to an equivalent nonlinear monotone equation. Furthermore, Yin et al. [8] developed a nonlinear conjugate gradient method for ℓ_1-norm regularization problems in compressed sensing. Yu [9, 10] extended the spectral gradient method and conjugate gradient-type method to solve large-scale nonlinear system of equations, respectively. Recently, the authors in [11] proposed a multivariate spectral gradient projection method for solving nonlinear monotone equations with convex constraints. Numerical results show that

multivariate spectral gradient method (MSG) could improve its performance very well.

Following this line, based on multivariate spectral gradient method (MSG), we present an adaptive prediction-correction method for solving nonlinear monotone equations (1) in the next section. Its global convergence result is established, which is independent of the merit function and Lipschitz continuity. Section 3 presents some numerical experiments to demonstrate and test its practical performance on compressed sensing and image deconvolution problems. Finally, we have a conclusion section.

2. Adaptive Prediction-Correction Method

Considering the projection method [5] for solving nonlinear monotone equations (1), suppose that we have obtained a direction d_k. By performing some kind of line search procedure along the direction d_k, a point $z_k = x_k + \alpha_k d_k$ can be computed such that

$$\langle g(z_k), x_k - z_k \rangle > 0. \tag{2}$$

By the monotonicity of g, for any $\overline{x}$ such that $g(\overline{x}) = 0$, we have

$$\langle g(z_k), \overline{x} - z_k \rangle \leq 0. \tag{3}$$

Thus, the hyperplane

$$H_k = \{x \in \Re^n \mid \langle g(z_k), x - z_k \rangle = 0\} \tag{4}$$

strictly separates the current iterate x_k from solutions of the systems of monotone equations. Once we get the separating hyperplane, the next iterate x_{k+1} is computed by projecting x_k on it.

Recalling the multivariate spectral gradient (MSG) method [12] for minimization problem $\min\{f(x) \mid x \in \Re^n\}$, its iterative formula is defined by $x_{k+1} = x_k - \operatorname{diag}\{1/\lambda_k^1, 1/\lambda_k^2, \ldots, 1/\lambda_k^n\} g_k$, where g_k is the gradient of f at x_k and $\operatorname{diag}\{\lambda_k^1, \lambda_k^2, \ldots, \lambda_k^n\}$ is obtained by minimizing

$$\left\| \operatorname{diag}\{\lambda^1, \lambda^2, \ldots, \lambda^n\} s_{k-1} - y_{k-1} \right\|_2 \tag{5}$$

with respect to $\{\lambda^i\}_{i=1}^n$, where $s_{k-1} = x_k - x_{k-1}$, $y_k = g_k - g_{k-1}$. In particular, when $f(x)$ has positive definite diagonal Hessian matrix, multivariate spectral gradient method will be convergent quadratically [12].

Let the ith column of y_k and s_k denoted by s_k^i and y_k^i, respectively. Combining multivariate spectral gradient method with projection scheme, we can present an adaptive prediction-correction method for solving monotone equations (1) as follows.

Algorithm 1 (multivariate spectral gradient (MSG) method). Given $x_0 \in \Re^n$, $\beta \in (0,1)$, $\sigma \in (0,1)$, $0 < \varepsilon < 1$, $r \geq 0$, $\delta > 0$. Set $k = 0$.

Step 1. If $\|g_k\| = 0$, stop.

Step 2. (a) If $k = 0$, set $d_k = -g(x_k)$.

(b) else if $y_{k-1}^i / s_{k-1}^i > 0$, then set $\lambda_k^i = y_{k-1}^i / s_{k-1}^i$; otherwise set $\lambda_k^i = (s_{k-1}^T y_{k-1}) / (s_{k-1}^T s_{k-1})$ for $i = 1, 2, \ldots, n$, where $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = g(x_k) - g(x_{k-1}) + r s_{k-1}$.

(c) else if $\lambda_k^i \leq \varepsilon$ or $\lambda_k^i \geq 1/\varepsilon$, set $\lambda_k^i = \delta$ for $i = 1, 2, \ldots, n$.

Set $d_k = -\operatorname{diag}\{1/\lambda_k^1, 1/\lambda_k^2, \ldots, 1/\lambda_k^n\} g_k$.

Step 3 (prediction step). Compute step length α_k, set $z_k = x_k + \alpha_k d_k$, where $\alpha_k = \beta^{m_k}$ with m_k being the smallest nonnegative integer m such that

$$-\langle g(x_k + \beta^m d_k), d_k \rangle \geq \sigma \beta^m \|d_k\|^2. \tag{6}$$

Step 4 (correction step). Compute

$$x_{k+1} = x_k - \frac{\langle g(z_k), x_k - z_k \rangle}{\|g(z_k)\|^2} g(z_k). \tag{7}$$

Step 5. Set $k = k + 1$ and go to Step 1.

By using multivariate spectral gradient method, we obtain prediction sequence $\{z_k\}$, and then we get correction sequence $\{x_k\}$ via projection. It follows from (17) that x_{k+1} will be more close to the solution x^* than x_k, that is, the sequence $\{x_k\}$ makes progress iterate by iterate. From Step 2(c), we have

$$\min\left\{\epsilon, \frac{1}{\delta}\right\} \|g_k\| \leq \|d_k\| \leq \max\left\{\frac{1}{\epsilon}, \frac{1}{\delta}\right\} \|g_k\|. \tag{8}$$

In what follows, we assume that $g(x_k) \neq 0$ for all $k \geq 0$; otherwise we have got the solution of the problem (1). The following lemma states that Algorithm 1 is well defined.

Lemma 2. *There exists a nonnegative number m_k satisfying (6) for all $k \geq 0$.*

Proof. Suppose that there exists a $k_0 \geq 0$ such that (6) is not satisfied for any nonnegative integer m, that is,

$$-\left\langle g\left(x_{k_0} + \beta^m d_{k_0}\right), d_{k_0} \right\rangle < \sigma \beta^m \left\| d_{k_0} \right\|^2, \quad \forall m \geq 1. \tag{9}$$

Let $m \to \infty$ and using the continuity of g yields

$$-\left\langle g\left(x_{k_0}\right), d_{k_0} \right\rangle \leq 0. \tag{10}$$

From Steps 1, 2, and 5, we have

$$g(x_k) \neq 0, \quad d_k \neq 0, \quad \forall k \geq 0. \tag{11}$$

Thus,

$$\begin{aligned} -\langle g(x_0), d_0 \rangle &= \langle g(x_0), g(x_0) \rangle > 0, \\ -\langle g(x_k), d_k \rangle &= \left\langle g(x_k), \operatorname{diag}\left\{\frac{1}{\lambda_k^1}, \frac{1}{\lambda_k^2}, \ldots, \frac{1}{\lambda_k^n}\right\} g(x_k) \right\rangle \\ &\geq \min\left\{\epsilon, \frac{1}{\delta}\right\} \|g_k\|^2 > 0, \quad \forall k \geq 1. \end{aligned} \tag{12}$$

The last inequality contradicts (10). Hence the statement is proved. □

Lemma 3. *Let $\{x_k\}$ and $\{z_k\}$ be any sequence generated by Algorithm 1. Suppose that g is monotone and that the solution set of (1) is not empty, then $\{x_k\}$ and $\{z_k\}$ are both bounded. Furthermore, it holds that*

$$\lim_{k\to\infty} \|x_k - z_k\| = 0, \tag{13}$$

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{14}$$

Proof. From (6), we have

$$\begin{aligned} \langle g(z_k), x_k - z_k\rangle &= -\alpha_k \langle g(z_k), d_k\rangle \\ &\geq \sigma\alpha_k^2 \|d_k\|^2 \\ &= \sigma\|x_k - z_k\|^2. \end{aligned} \tag{15}$$

Let x^* be an arbitrary point such that $g(x^*) = 0$. Taking account of the monotonicity of g, we have

$$\begin{aligned} \langle g(z_k), x_k - x^*\rangle &= \langle g(z_k), x_k - z_k\rangle + \langle g(z_k), z_k - x^*\rangle \\ &\geq \langle g(z_k), x_k - z_k\rangle + \langle g(x^*), z_k - x^*\rangle \\ &= \langle g(z_k), x_k - z_k\rangle. \end{aligned} \tag{16}$$

From (7), (14), and (16), it follows that

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \left\| x_k - \frac{\langle g(z_k), x_k - z_k\rangle}{\|g(z_k)\|^2} g(z_k) - x^* \right\|^2 \\ &= \|x_k - x^*\|^2 - \frac{\langle g(z_k), x_k - z_k\rangle^2}{\|g(z_k)\|^2} \\ &\leq \|x_k - x^*\|^2 - \frac{\sigma^2\|x_k - z_k\|^4}{\|g(z_k)\|^2}. \end{aligned} \tag{17}$$

Hence the sequence $\{\|x_k - x^*\|\}$ is decreasing and convergent; moreover, the sequence $\{\|x_k\|\}$ is bounded. Since the g is continuous, there exists a constant $C > 0$ such that

$$\|g(z_k)\| \leq C. \tag{18}$$

By the Cauchy-Schwarz inequality, the monotonicity of g and (15), we have

$$\begin{aligned} \|g(x_k)\| &\geq \frac{\langle g(x_k), x_k - z_k\rangle}{\|x_k - z_k\|} \\ &\geq \frac{\langle g(z_k), x_k - z_k\rangle}{\|x_k - z_k\|} \\ &\geq \sigma\|x_k - z_k\|. \end{aligned} \tag{19}$$

From (18) and (19), we obtain that $\{z_k\}$ is also bounded. It follows from (17) and (18) that

$$\frac{\sigma^2}{C^2}\sum_{k=1}^{\infty}\|x_k - z_k\|^4 \leq \sum_{k=1}^{\infty}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right) < \infty, \tag{20}$$

which implies

$$\lim_{k\to\infty} \|x_k - z_k\| = 0. \tag{21}$$

From (7), using the Cauchy-Schwarz inequality, we obtain that

$$\|x_{k+1} - x_k\| = \frac{\langle g(z_k), x_k - z_k\rangle}{\|g(z_k)\|} \leq \|x_k - z_k\|. \tag{22}$$

Thus $\lim_{k\to\infty}\|x_{k+1} - x_k\| = 0$.

The proof is complete. □

Now we can establish the global convergence of Algorithm 1.

Theorem 4. *Let x_k be generated by Algorithm 1; then $\{x_k\}$ converges to an $\overline{x}$ such that $g(\overline{x}) = 0$.*

Proof. Since $z_k = x_k + \alpha_k d_k$, it follows from Lemma 3 that

$$\lim_{k\to\infty} \alpha_k \|d_k\| = \lim_{k\to\infty} \|x_k - z_k\| = 0. \tag{23}$$

From (8) and (18), it holds that $\{d_k\}$ is bounded.

Now we consider the following two possible cases:

(i) $\liminf_{k\to\infty}\|d(x_k)\| = 0$.

(ii) $\liminf_{k\to\infty}\|d(x_k)\| > 0$.

If (i) holds, from (8), we have $\liminf_{k\to\infty}\|g(x_k)\| = 0$. By the continuity of g and the boundedness of $\{x_k\}$, it is clear that the sequence $\{x_k\}$ has some accumulation point $\overline{x}$ such that $g(\overline{x}) = 0$. From (17), we also have that the sequence $\{\|x_k - \overline{x}\|\}$ converges. Therefore, $\{x_k\}$ converges to $\overline{x}$.

If (ii) holds, from (8), we have $\liminf_{k\to\infty}\|g(x_k)\| > 0$. By (23), it holds that

$$\lim_{k\longrightarrow\infty} \alpha_k = 0. \tag{24}$$

By the line search rule, we have for all k sufficiently large, $m_k - 1$ will not satisfy (6). This means

$$-\left\langle g\left(x_k + \beta^{m_k-1}d_k\right), d_k\right\rangle < \sigma\beta^{m_k-1}\|d_k\|^2. \tag{25}$$

Since the sequences $\{x_k\}$, $\{d_k\}$ are bounded, we choose a subsequence, let $k \to \infty$ in (25), we obtain that

$$-\left\langle g(\widetilde{x}), \widetilde{d}\right\rangle \leq 0, \tag{26}$$

where $\widetilde{x}$, $\widetilde{d}$ are limits of corresponding subsequences. On the other hand, by (8), it holds that

$$-\left\langle g(\widetilde{x}), \widetilde{d}\right\rangle > 0, \tag{27}$$

which contradicts (26). Hence, $\liminf_{k\to\infty}\|d(x_k)\| > 0$ is impossible.

The proof is complete. □

3. Numerical Experiments

In this section, we report some preliminary numerical experiments to test our algorithms with comparison to spectral gradient projection method [6]. Firstly, in Section 3.1 we test these algorithms on solving nonlinear systems of monotone equations. Secondly, in Section 3.2, we apply HSG-V algorithm to solve ℓ_1-norm regularization problem arising from compressed sensing. All of numerical experiments were performed under Windows XP and MATLAB 7.0 running on a personal computer with an Intel Core 2 Duo CPU at 2.2 GHz and 2 GB of memory.

3.1. Test on Nonlinear Systems of Monotone Equations. We test the performance of our algorithms for solving some monotone equations (see details in the appendix). The termination condition is $\|g(x_k)\| \le 10^{-6}$. The parameters are specified as follows. For MSG method, we set $\beta = 0.5$, $\sigma = 0.01$, $\epsilon = 10^{-10}$, $r = 0.01$. In Step 2, the parameter δ is chosen in the following way:

$$\delta = \begin{cases} 1 & \text{if } \|g(x_k)\| > 1, \\ \|g(x_k)\|^{-1} & \text{if } 10^{-5} \le \|g(x_k)\| \le 1, \\ 10^{5} & \text{if } \|g(x_k)\| < 10^{-5}. \end{cases} \tag{28}$$

Firstly, we test the performance of the MSG method on the Problem 1 with $n = 1000$, the initial point $x_0 = (1, 1, \ldots, 1)^T$. Figure 1 displays the performance of MSG method for Problem 1 which indicates that prediction sequences $\{z_k\}$ are better than correction sequences $\{x_k\}$ at most time. Taking this into account, we relax the MSG method such that Step 4 in Algorithm 1 is replaced by the following:

if mod

$$x_{k+1} = x_k - \frac{\langle g(z_k), x_k - z_k \rangle}{\|g(z_k)\|^2} g(z_k),$$

eslse

$$x_{k+1} = z_k,$$

end.

In this case, we refer to this modification as "MSG-V" method. When $M \equiv 1$, the above algorithm will reduce to Algorithm 1. The performance of those methods on the Problem (1) is shown in Figure 1, from which we can see that the MSG-V method is preferable quite frequently to the SG method while it also outperforms the MSG method. Furthermore, motivated to accelerate the performance of MSG-V method, we present a hybrid spectral gradient (HSG-V) algorithm. The main idea of the HSG-V algorithm is to run MSG-V algorithm when $y_{k-1}^i / s_{k-1}^i > 0$ for $i = 1, 2, \ldots, n$; otherwise switch to spectral gradient projection (SG) method.

And then we compare the performance of MSG method, MSG-V method, and HSG-V method with the spectral gradient projection (SG) method in [6] on test problems with different initial points. We set $\beta = 0.5$, $\sigma = 0.01$, $r = 0.01$ in the spectral gradient projection (SG) method in [6], and $M = 10$ for MSG-V method and HSG-V method.

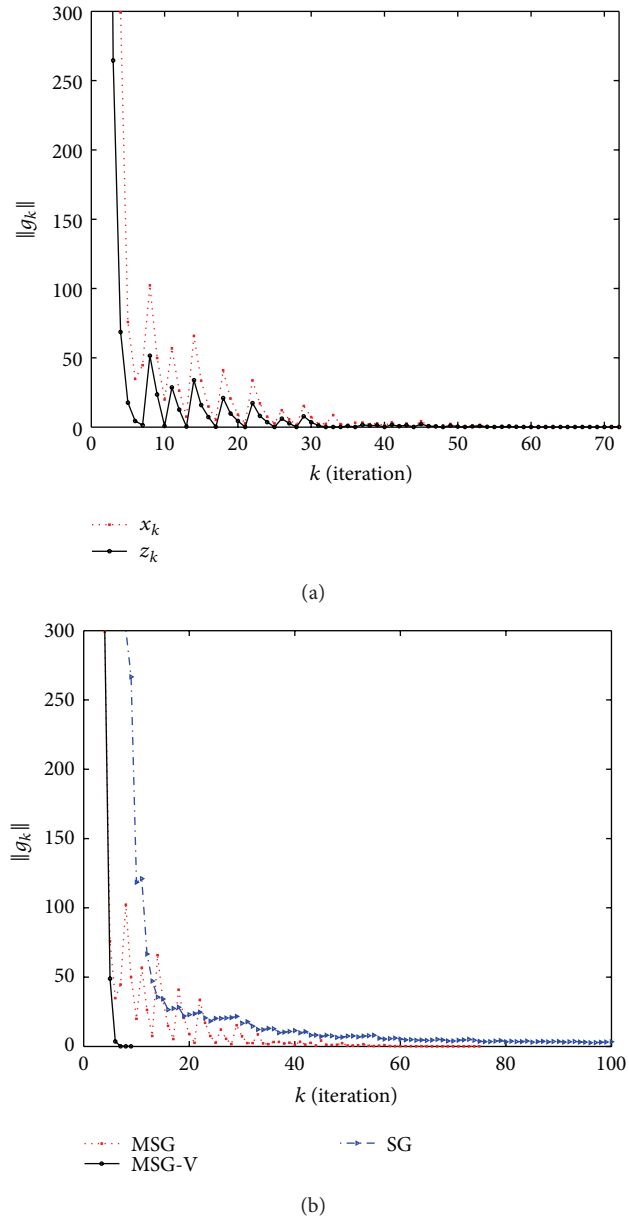

FIGURE 1: (a) Norm of sequence versus iteration for MSG algorithm. (b) MSG-V algorithm versus MSG and SG algorithm, where the iteration has been cut to 100 for the SG algorithm.

Numerical results are shown in Tables 1, 2, 3, 4, 5, and 6 with the form NI/NF/T/BK, where we report the dimension of the problem (n), the initial points (Init), the number of iteration (NI), the number of function evaluations (NF), and the CPU time (Time) in seconds and the number of backtracking (BK). The symbol "F" denotes that the method fails for this test problem, or the number of the iterations is greater than 10000.

As we can see from Tables 1–6 that the HSG-V algorithm is preferable quite frequently to the SG method and also outperforms the MSG algorithm and MSG-V algorithm, since

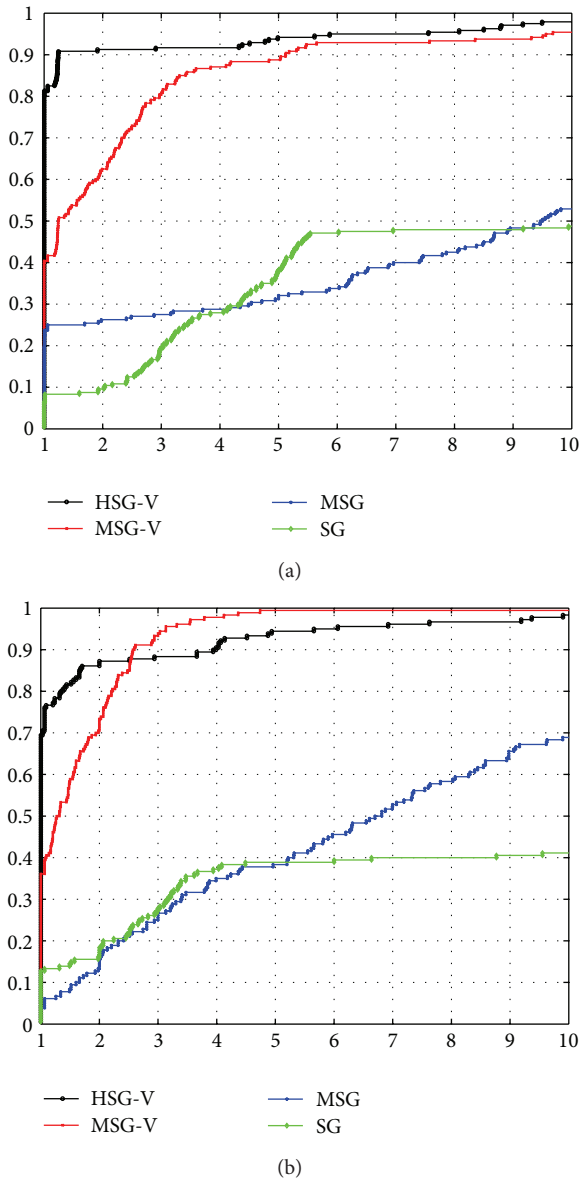

Figure 2: (a) Performance profiles for the number of function evaluations. (b) Performance profiles for the CPU time.

it can solve about 80% and 70% of the problems with the best time and the smallest number of function evaluations, respectively. We also find that the SG algorithm seems more sensitive to the initial points.

Figure 2 shows the performance of these algorithms relative to the number of function evaluations and CPU time, respectively, which were evaluated using the profiles of Dolan and Moré [13]. That is, for each algorithm, we plot the fraction P of problems for which the method is within a factor t of the smallest number of function evaluations/CPU time. Clearly, the left side of the figure gives the percentage of the test problems for which a method is the best one according to the number of function evaluations or CPU time, respectively. As we can see from Figure 2, "HSG-V" algorithm has the best performance.

Table 1: Numerical results for SG/MSG methods on Problem 1.

Init (n)	SG	MSG
	NI/NF/Time/BK	NI/NF/Time/BK
x_1 (100)	7935/56267/7.375/6	1480/9774/1.609/1
x_2 (100)	4365/25627/3.125/2	981/6223/0.906/1
x_3 (100)	3131/18028/2.11/7	1139/8240/1.266/1
x_4 (100)	2287/13025/1.453/4	294/1091/0.172/1
x_5 (100)	1685/9535/1.188/3	212/640/0.093/1
x_6 (100)	1788/10238/1.156/3	243/745/0.11/1
x_7 (100)	1608/9236/1.047/2	220/664/0.109/1
x_8 (100)	1629/9283/1.172/4	185/558/0.078/1
x_9 (100)	1478/8407/0.953/4	8/20/0.016/0
x_{10} (100)	1611/9131/1.031/3	184/555/0.078/1
x_{11} (100)	1475/8404/0.938/4	39/99/0.016/0
x_{12} (100)	1226/6938/0.797/5	19/46/0.016/0
x_1 (200)	F	2846/20922/6.328/1
x_2 (200)	8506/50896/11.985/4	1535/11707/3.5/1
x_3 (200)	6193/37063/8.687/7	1826/15256/4.5/0
x_4 (200)	4563/27333/6.5/7	266/1055/0.312/1
x_5 (200)	3343/19760/5.078/4	376/1133/0.422/1
x_6 (200)	3620/21536/6.11/7	200/617/0.172/1
x_7 (200)	3249/19340/4.531/6	148/444/0.125/0
x_8 (200)	3253/19383/4.5/5	323/973/0.344/1
x_9 (200)	2974/17649/4.109/4	8/21/0.015/0
x_{10} (200)	3256/19214/5.062/4	308/928/0.391/1
x_{11} (200)	2995/17784/5.266/4	42/110/0.047/0
x_{12} (200)	2483/14698/3.453/3	27/63/0.047/0
x_1 (300)	F	F
x_2 (300)	F	3374/29158/12.782/1
x_3 (300)	9334/57185/20.985/2	1293/10136/4.656/1
x_4 (300)	6734/41348/14.735/4	406/1601/0.687/1
x_5 (300)	5011/30857/11.265/7	235/706/0.282/1
x_6 (300)	5380/33167/11.875/6	300/919/0.546/1
x_7 (300)	4812/29977/10.657/6	187/559/0.235/0
x_8 (300)	4825/29770/10.656/4	158/466/0.187/0
x_9 (300)	4396/27551/10.062/3	8/21/0.016/0
x_{10} (300)	4774/29731/10.969/3	203/610/0.266/1
x_{11} (300)	4411/27366/9.859/8	52/144/0.062/0
x_{12} (300)	3656/23021/8.36/6	32/75/0.031/0
x_1 (500)	F	F
x_2 (500)	F	4915/46911/37.906/1
x_3 (500)	F	1754/15152/12.375/1
x_4 (500)	F	489/1905/1.547/1

Table 1: Continued.

Init (n)	SG	MSG
	NI/NF/Time/BK	NI/NF/Time/BK
x_5 (500)	8360/51414/37.485/6	269/803/0.797/1
x_6 (500)	8996/56185/39.75/6	295/900/0.734/1
x_7 (500)	8128/50525/35.703/2	256/766/0.672/1
x_8 (500)	8089/50688/36.484/4	387/1163/1.156/0
x_9 (500)	7453/46083/33.75/2	8/22/0.016/0
x_{10} (500)	8118/50185/35.985/4	403/1211/1.187/1
x_{11} (500)	7525/46275/33.14/4	55/149/0.11/0
x_{12} (500)	6235/38408/28.047/9	38/95/0.11/0
x_1 (1000)	F	F
x_2 (1000)	F	8998/93619/200.78/0
x_3 (1000)	F	2939/26376/55.86/0
x_4 (1000)	F	783/3016/6.438/0
x_5 (1000)	F	364/1085/2.75/0
x_6 (1000)	F	429/1309/3.266/0
x_7 (1000)	F	499/1500/3.687/1
x_8 (1000)	F	481/1444/3.969/0
x_9 (1000)	F	8/23/0.047/0
x_{10} (1000)	F	346/1032/2.515/0
x_{11} (1000)	F	74/201/0.391/0
x_{12} (1000)	F	50/124/0.234/0

3.2. Test on ℓ_1-Norm Regularization Problem in Compressed Sensing. There has been considerable interest in solving the ℓ_1-norm regularized least-square problem

$$\min_{x\in\Re^n} f(x) \equiv \frac{1}{2}\|Ax - y\|_2^2 + \mu\|x\|_1, \tag{29}$$

where $A \in \Re^{m\times n}(m \ll n)$ is a linear operator, $y \in \Re^m$ is an observation, and μ is a nonnegative parameter. Equation (29) mainly appears in compressed sensing: an emerging methodology in digital signal processing and has attracted intensive research activities over the past few years. Compressed sensing is based on the fact that if original signal is sparse or approximately sparse in some orthogonal basis, then an exact restoration can be produced by solving (29).

Recently, Figueiredo et al. [14] proposed gradient projection method for sparse reconstruction (GPSR). The first key step of GPSR method is to express (29) as a quadratic program. For any $x \in \Re^n$ it can be formulated as $x = u - v$, $u \geq 0$, $v \geq 0$, where $u \in \Re^n, v \in \Re^n$, and $u_i = (x_i)_+, v_i = (-x_i)_+$ for $i = 1, 2, \ldots, n$ with $(\cdot)_+ = \max\{0, \cdot\}$. We thus have $\|x\|_1 = e_n^T u + e_n^T v$, where $e_n = (1, 1, \ldots, 1)^T$ is the

Table 2: Numerical results for MSG-V/HSG-V methods on Problem 1.

Init (n)	MSG-V	HSG-V
	NI/NF/Time/BK	NI/NF/Time/BK
x_1 (100)	48/162/0.031/0	139/471/0.14/0
x_2 (100)	37/120/0.016/0	165/572/0.079/7
x_3 (100)	29/93/0.015/0	158/522/0.062/2
x_4 (100)	22/61/0.016/0	134/461/0.063/0
x_5 (100)	8/22/0.016/0	8/22/0.015/0
x_6 (100)	9/26/0.015/0	9/26/0.015/0
x_7 (100)	8/22/0.016/0	8/22/0.016/0
x_8 (100)	8/21/0.015/0	8/21/0.016/0
x_9 (100)	8/20/0.015/0	8/20/0.016/0
x_{10} (100)	24/75/0.031/1	24/75/0.031/1
x_{11} (100)	8/21/0.015/0	8/21/0.016/0
x_{12} (100)	6/16/0.016/0	6/16/0.016/0
x_1 (200)	43/134/0.047/0	174/587/0.172/0
x_2 (200)	39/142/0.047/0	184/640/0.172/0
x_3 (200)	35/118/0.031/1	205/693/0.188/0
x_4 (200)	19/59/0.031/0	148/519/0.125/0
x_5 (200)	8/23/0.296/0	8/23/0.015/0
x_6 (200)	9/27/0.016/0	9/27/0.016/0
x_7 (200)	8/23/0.016/0	8/23/0.016/0
x_8 (200)	8/22/0.015/0	8/22/0.016/0
x_9 (200)	8/21/0.016/0	8/21/0.015/0
x_{10} (200)	25/79/0.046/1	25/79/0.031/1
x_{11} (200)	8/22/0.016/0	8/22/0.016/0
x_{12} (200)	6/17/0.015/0	6/17/0.016/0
x_1 (300)	50/200/0.094/0	246/865/0.375/4
x_2 (300)	56/196/0.093/1	265/977/0.375/8
x_3 (300)	33/119/0.047/0	345/1253/0.515/7
x_4 (300)	28/90/0.031/0	244/825/0.329/0
x_5 (300)	8/23/0.016/0	8/23/0.015/0
x_6 (300)	9/27/0.015/0	9/27/0.016/0
x_7 (300)	8/23/0.016/0	8/23/0.015/0
x_8 (300)	8/22/0.016/0	8/22/0.016/0
x_9 (300)	8/21/0.016/0	8/21/0.016/0
x_{10} (300)	26/82/0.031/1	26/82/0.031/1
x_{11} (300)	8/23/0.016/0	8/23/0.016/0
x_{12} (300)	6/18/0.015/0	6/18/0.015/0
x_1 (500)	51/218/0.188/0	290/1054/0.765/8
x_2 (500)	41/132/0.125/0	330/1159/0.953/0
x_3 (500)	47/164/0.14/1	383/1394/0.969/0

Table 2: Continued.

Init (n)	MSG-V NI/NF/Time/BK	HSG-V NI/NF/Time/BK
x_4 (500)	28/91/0.125/0	246/864/0.609/0
x_5 (500)	9/26/0.047/0	9/26/0.032/0
x_6 (500)	9/28/0.031/0	9/28/0.015/0
x_7 (500)	9/26/0.047/0	9/26/0.032/0
x_8 (500)	8/23/0.032/0	8/23/0.015/0
x_9 (500)	8/22/0.031/0	8/22/0.016/0
x_{10} (500)	27/86/0.062/1	27/86/0.062/1
x_{11} (500)	8/23/0.016/0	8/23/0.032/0
x_{12} (500)	6/19/0.016/0	6/19/0.015/0
x_1 (1000)	49/205/0.547/0	437/1656/3.094/2
x_2 (1000)	41/138/0.344/0	506/1824/3.406/0
x_3 (1000)	49/181/0.437/1	607/2137/4.094/2
x_4 (1000)	37/121/0.282/1	426/1582/3/0
x_5 (1000)	9/27/0.062/0	9/27/0.062/0
x_6 (1000)	9/29/0.063/0	86/308/0.579/6
x_7 (1000)	9/27/0.046/0	9/27/0.046/0
x_8 (1000)	8/24/0.063/0	8/24/0.047/0
x_9 (1000)	8/23/0.047/0	12/44/0.078/2
x_{10} (1000)	29/93/0.172/1	29/93/0.188/1
x_{11} (1000)	8/24/0.047/0	8/24/0.078/0
x_{12} (1000)	6/20/0.031/0	6/20/0.078/0

Table 3: Numerical results for SG/MSG methods on Problem 2.

Init (n)	SG NI/NF/Time/BK	MSG NI/NF/Time/BK
x_1 (100)	F	349/712/0.094/0
x_2 (100)	F	352/711/0.094/0
x_3 (100)	F	351/711/0.094/0
x_4 (100)	F	353/717/0.109/0
x_5 (100)	F	346/696/0.094/0
x_6 (100)	F	347/698/0.078/0
x_7 (100)	F	345/694/0.109/0
x_8 (100)	F	320/644/0.078/0
x_9 (100)	359/719/0.031/0	359/719/0.047/0
x_{10} (100)	22/67/0.015/1	22/67/0.015/1
x_{11} (100)	434/869/0.047/0	434/869/0.063/0
x_{12} (100)	424/849/0.031/0	424/849/0.047/0
x_1 (200)	F	437/888/0.218/0
x_2 (200)	F	439/885/0.219/0

Table 3: Continued.

Init (n)	SG NI/NF/Time/BK	MSG NI/NF/Time/BK
x_3 (200)	F	438/885/0.219/0
x_4 (200)	F	440/891/0.219/0
x_5 (200)	F	433/870/0.343/0
x_6 (200)	F	434/872/0.203/0
x_7 (200)	F	432/868/0.219/0
x_8 (200)	F	358/720/0.172/0
x_9 (200)	372/745/0.047/0	371/743/0.063/0
x_{10} (200)	22/66/0.015/0	23/70/0.015/1
x_{11} (200)	544/1089/0.063/0	544/1089/0.094/0
x_{12} (200)	534/1069/0.047/0	534/1069/0.109/0
x_1 (300)	F	498/1010/0.375/0
x_2 (300)	F	500/1007/0.375/0
x_3 (300)	F	500/1009/0.375/0
x_4 (300)	F	501/1013/0.36/0
x_5 (300)	F	494/992/0.375/0
x_6 (300)	F	495/994/0.375/0
x_7 (300)	F	494/992/0.359/0
x_8 (300)	F	368/740/0.266/0
x_9 (300)	373/747/0.047/0	373/747/0.093/0
x_{10} (300)	22/66/0.015/0	23/70/0.016/1
x_{11} (300)	621/1243/0.078/0	621/1243/0.157/0
x_{12} (300)	611/1223/0.078/0	611/1223/0.25/0
x_1 (500)	F	588/1190/0.781/0
x_2 (500)	F	590/1187/0.781/0
x_3 (500)	F	589/1187/0.781/0
x_4 (500)	F	591/1193/0.797/0
x_5 (500)	F	584/1172/0.782/0
x_6 (500)	F	585/1174/0.906/0
x_7 (500)	F	583/1170/0.797/0
x_8 (500)	F	372/748/0.468/0
x_9 (500)	373/747/0.078/0	373/747/0.125/0
x_{10} (500)	22/66/0.015/0	23/70/0.016/1
x_{11} (500)	734/1469/0.141/0	734/1469/0.234/0
x_{12} (500)	724/1449/0.14/0	724/1449/0.25/0
x_1 (1000)	F	736/1486/2.563/0
x_2 (1000)	F	739/1485/2.406/0
x_3 (1000)	F	738/1485/2.578/0
x_4 (1000)	F	740/1491/2.407/0
x_5 (1000)	F	733/1470/2.562/0
x_6 (1000)	F	734/1472/2.531/0
x_7 (1000)	F	732/1468/2.407/0
x_8 (1000)	F	372/748/1.125/0
x_9 (1000)	373/747/0.125/0	373/747/0.343/0
x_{10} (1000)	24/73/0.016/1	24/73/0.032/1
x_{11} (1000)	921/1843/0.281/0	921/1843/0.562/0
x_{12} (1000)	912/1825/0.266/0	912/1825/0.547/0

Table 4: Numerical results for MSG-V/HSG-V methods on Problem 2.

Init (n)	MSG-V	HSG-V
	NI/NF/Time/BK	NI/NF/Time/BK
x_1 (100)	27/82/0.015/1	27/82/0.016/1
x_2 (100)	435/872/0.047/0	435/872/0.094/0
x_3 (100)	416/838/0.047/0	416/838/0.062/0
x_4 (100)	434/872/0.047/0	434/872/0.047/0
x_5 (100)	424/850/0.047/0	424/850/0.047/0
x_6 (100)	347/697/0.047/0	347/697/0.047/0
x_7 (100)	346/695/0.047/0	346/695/0.032/0
x_8 (100)	318/640/0.078/0	318/640/0.062/0
x_9 (100)	355/711/0.031/0	355/711/0.031/0
x_{10} (100)	22/67/0.016/1	22/67/0.016/1
x_{11} (100)	434/869/0.063/0	434/869/0.062/0
x_{12} (100)	424/849/0.046/0	424/849/0.047/0
x_1 (200)	27/82/0.015/1	27/82/0.016/1
x_2 (200)	545/1092/0.094/0	545/1092/0.078/0
x_3 (200)	525/1056/0.094/0	525/1056/0.078/0
x_4 (200)	544/1092/0.094/0	544/1092/0.078/0
x_5 (200)	533/1068/0.093/0	533/1068/0.078/0
x_6 (200)	435/873/0.063/0	435/873/0.078/0
x_7 (200)	433/869/0.078/0	433/869/0.063/0
x_8 (200)	355/714/0.172/0	356/716/0.078/0
x_9 (200)	367/735/0.062/0	367/735/0.047/0
x_{10} (200)	23/70/0.015/1	23/70/0.016/1
x_{11} (200)	544/1089/0.094/0	544/1089/0.078/0
x_{12} (200)	534/1069/0.094/0	534/1069/0.078/0
x_1 (300)	28/85/0.016/1	28/85/0.016/1
x_2 (300)	622/1246/0.14/0	622/1246/0.11/0
x_3 (300)	602/1210/0.156/0	602/1210/0.125/0
x_4 (300)	621/1246/0.25/0	621/1246/0.109/0
x_5 (300)	610/1222/0.125/0	610/1222/0.125/0
x_6 (300)	496/995/0.11/0	496/995/0.094/0
x_7 (300)	494/991/0.125/0	494/991/0.094/0
x_8 (300)	365/734/0.25/0	365/734/0.109/0
x_9 (300)	368/737/0.094/0	368/737/0.063/0
x_{10} (300)	23/70/0.031/1	23/70/0.015/1
x_{11} (300)	621/1243/0.141/0	621/1243/0.11/0
x_{12} (300)	611/1223/0.125/0	611/1223/0.125/0
x_1 (500)	28/85/0.015/1	28/85/0.015/1
x_2 (500)	735/1472/0.25/0	735/1472/0.203/0
x_3 (500)	715/1436/0.219/0	715/1436/0.203/0
x_4 (500)	734/1472/0.25/0	734/1472/0.188/0
x_5 (500)	723/1448/0.219/0	723/1448/0.187/0
x_6 (500)	586/1175/0.187/0	586/1175/0.156/0
x_7 (500)	584/1171/0.204/0	584/1171/0.141/0
x_8 (500)	369/742/0.453/0	369/742/0.156/0
x_9 (500)	368/737/0.125/0	368/737/0.094/0
x_{10} (500)	23/70/0.015/1	23/70/0.015/1
x_{11} (500)	734/1469/0.219/0	734/1469/0.203/0
x_{12} (500)	724/1449/0.234/0	724/1449/0.235/0
x_1 (1000)	28/85/0.016/1	28/85/0.047/1
x_2 (1000)	923/1848/0.531/0	923/1848/0.485/0
x_3 (1000)	902/1810/0.641/0	902/1810/0.437/0
x_4 (1000)	922/1848/0.516/0	922/1848/0.453/0
x_5 (1000)	911/1824/0.531/0	911/1824/0.453/0
x_6 (1000)	737/1477/0.406/0	737/1477/0.344/0
x_7 (1000)	733/1469/0.422/0	733/1469/0.344/0
x_8 (1000)	369/742/1.125/0	369/742/0.297/0
x_9 (1000)	368/737/0.219/0	368/737/0.172/0
x_{10} (1000)	24/73/0.015/1	24/73/0.015/1
x_{11} (1000)	921/1843/0.625/0	921/1843/0.438/0
x_{12} (1000)	912/1825/0.563/0	912/1825/0.453/0

Table 5: Numerical results for SG/MSG methods on Problem 3.

Init (n)	SG	MSG
	NI/NF/Time/BK	NI/NF/Time/BK
x_1 (100)	161/753/0.094/2	246/1062/0.296/3
x_2 (100)	103/424/0.062/3	185/794/0.219/2
x_3 (100)	118/517/0.063/2	204/935/0.266/5
x_4 (100)	63/224/0.031/1	194/894/0.25/2
x_5 (100)	63/234/0.031/2	149/662/0.187/3
x_6 (100)	81/307/0.047/2	191/831/0.235/1
x_7 (100)	64/237/0.031/2	168/738/0.203/1
x_8 (100)	53/194/0.032/2	94/378/0.109/1
x_9 (100)	53/192/0.015/2	178/808/0.219/3
x_{10} (100)	76/288/0.047/2	229/1008/0.281/1
x_{11} (100)	73/273/0.031/2	203/924/0.25/4
x_{12} (100)	60/225/0.016/3	195/888/0.266/1
x_1 (200)	216/1203/0.468/2	251/1129/0.515/2
x_2 (200)	147/728/0.282/2	168/702/0.407/3
x_3 (200)	157/759/0.312/2	180/821/0.422/1
x_4 (200)	58/206/0.094/3	214/956/0.453/1

Table 5: Continued.

Init (n)	SG NI/NF/Time/BK	MSG NI/NF/Time/BK
x_5 (200)	64/238/0.094/1	174/793/0.359/2
x_6 (200)	78/294/0.125/2	194/882/0.422/2
x_7 (200)	44/156/0.062/2	170/757/0.359/1
x_8 (200)	48/173/0.078/1	93/368/0.172/1
x_9 (200)	54/192/0.078/2	191/890/0.391/2
x_{10} (200)	84/314/0.125/3	152/627/0.282/1
x_{11} (200)	61/222/0.094/2	193/903/0.422/2
x_{12} (200)	63/236/0.093/3	121/531/0.25/2
x_1 (300)	541/5455/20.282/3	247/1066/4.187/5
x_2 (300)	142/724/2.734/2	135/512/2.063/1
x_3 (300)	195/1084/4.031/2	197/892/3.5/1
x_4 (300)	55/196/0.734/2	193/868/3.328/2
x_5 (300)	68/254/0.953/2	154/693/2.703/4
x_6 (300)	77/295/1.11/2	241/1168/4.484/2
x_7 (300)	76/281/1.094/2	186/851/3.313/5
x_8 (300)	57/207/0.765/2	127/528/2/1
x_9 (300)	59/210/0.797/1	190/939/3.578/1
x_{10} (300)	91/342/1.266/3	142/664/2.563/1
x_{11} (300)	79/288/1.109/3	168/767/2.906/1
x_{12} (300)	72/266/0.984/1	114/448/1.75/1
x_1 (500)	488/4021/42.907/1	212/927/9.985/1
x_2 (500)	240/1440/15.343/3	166/712/7.64/4
x_3 (500)	254/1544/16.485/1	228/1050/11.313/2
x_4 (500)	59/210/2.266/3	273/1564/16.843/3
x_5 (500)	70/261/2.75/2	189/905/9.781/2
x_6 (500)	82/313/3.328/2	190/866/9.266/1
x_7 (500)	76/284/3.078/3	149/642/6.984/1
x_8 (500)	59/215/2.282/2	97/381/4.141/1
x_9 (500)	51/185/1.985/2	439/2832/30.391/3
x_{10} (500)	84/319/3.421/2	66/238/2.562/1
x_{11} (500)	77/280/2.969/2	74/266/2.891/1
x_{12} (500)	67/250/2.719/3	57/189/2.078/1
x_1 (1000)	2780/36160/1510.7/2	199/853/35.969/3
x_2 (1000)	331/2242/94.734/2	160/656/27.656/1
x_3 (1000)	352/2532/106.25/1	197/891/37.359/2
x_4 (1000)	71/260/10.891/1	182/794/33.985/2
x_5 (1000)	71/268/11.234/2	190/891/37.546/3
x_6 (1000)	84/319/13.422/3	160/698/29.188/4
x_7 (1000)	73/271/11.391/2	157/663/27.547/1
x_8 (1000)	50/182/7.578/2	112/446/18.641/4
x_9 (1000)	49/173/7.328/2	232/1162/48.656/1
x_{10} (1000)	85/318/13.375/2	56/172/7.391/1
x_{11} (1000)	81/297/12.485/3	61/185/7.781/1
x_{12} (1000)	69/255/10.781/1	49/154/6.578/1

Table 6: Numerical results for MSG-V/HSG-V methods on Problem 3.

Init (n)	MSG-V NI/NF/Time/BK	HSG-V NI/NF/Time/BK
x_1 (100)	92/377/0.078/1	55/172/0.047/6
x_2 (100)	93/374/0.079/1	59/177/0.031/0
x_3 (100)	86/360/0.078/1	40/120/0.031/0
x_4 (100)	83/363/0.062/1	40/111/0.032/1
x_5 (100)	45/173/0.047/1	20/56/0.015/0
x_6 (100)	57/204/0.047/2	42/121/0.016/2
x_7 (100)	53/202/0.031/1	30/87/0.031/0
x_8 (100)	47/181/0.047/2	22/61/0.016/0
x_9 (100)	49/194/0.047/1	33/100/0.015/1
x_{10} (100)	48/165/0.031/1	30/97/0.032/3
x_{11} (100)	49/181/0.031/1	26/75/0.015/0
x_{12} (100)	37/125/0.032/0	29/93/0.016/0
x_1 (200)	88/335/0.172/1	53/173/0.078/1
x_2 (200)	83/330/0.14/1	50/142/0.078/0
x_3 (200)	76/290/0.141/1	42/126/0.047/2
x_4 (200)	69/266/0.125/1	35/101/0.063/3
x_5 (200)	48/190/0.094/1	25/72/0.031/3
x_6 (200)	69/294/0.156/2	34/99/0.047/0
x_7 (200)	55/215/0.109/5	30/81/0.047/0
x_8 (200)	51/217/0.11/1	22/61/0.031/0
x_9 (200)	46/153/0.078/1	24/64/0.031/0
x_{10} (200)	42/129/0.078/1	33/91/0.031/0
x_{11} (200)	49/180/0.094/0	30/92/0.047/1
x_{12} (200)	37/125/0.062/2	30/85/0.047/0
x_1 (300)	89/322/1.266/1	53/168/0.625/0
x_2 (300)	91/348/1.282/1	55/162/0.609/1
x_3 (300)	72/278/1.046/1	37/109/0.422/0
x_4 (300)	78/339/1.297/1	28/81/0.297/1
x_5 (300)	45/178/0.735/2	20/55/0.281/0
x_6 (300)	63/248/0.937/3	38/113/0.453/2
x_7 (300)	53/208/0.797/1	33/91/0.344/0
x_8 (300)	63/295/1.109/1	22/61/0.234/0
x_9 (300)	45/177/0.672/1	27/71/0.266/0
x_{10} (300)	45/148/0.641/1	40/108/0.406/0
x_{11} (300)	43/144/0.531/2	32/89/0.344/1
x_{12} (300)	36/129/0.5/0	28/83/0.312/2
x_1 (500)	85/289/3.125/1	50/158/1.672/0
x_2 (500)	78/245/2.625/0	56/157/1.75/2
x_3 (500)	74/289/3.156/2	40/118/1.235/1
x_4 (500)	62/223/2.375/1	47/131/1.437/1
x_5 (500)	49/194/2.11/1	25/74/0.813/3

Table 6: Continued.

Init (n)	MSG-V NI/NF/Time/BK	HSG-V NI/NF/Time/BK
x_6 (500)	55/181/1.922/2	40/116/1.218/0
x_7 (500)	49/163/1.797/1	26/74/0.781/0
x_8 (500)	53/219/2.328/1	22/61/0.657/0
x_9 (500)	49/184/2/0	25/69/0.781/2
x_{10} (500)	41/158/1.656/2	35/99/1.062/6
x_{11} (500)	48/173/1.922/1	31/83/0.891/0
x_{12} (500)	35/126/1.359/2	27/76/0.812/0
x_1 (1000)	97/376/15.688/3	50/165/6.922/3
x_2 (1000)	78/263/11.015/1	52/151/6.235/2
x_3 (1000)	80/303/12.766/1	44/128/5.343/1
x_4 (1000)	73/291/12.219/0	36/101/4.235/0
x_5 (1000)	43/165/6.968/1	22/66/2.75/2
x_6 (1000)	59/213/9.094/4	44/120/5.016/0
x_7 (1000)	55/201/8.438/1	27/78/3.281/3
x_8 (1000)	57/250/10.5/1	22/60/2.547/0
x_9 (1000)	46/166/7/0	29/80/3.266/4
x_{10} (1000)	39/126/5.234/0	31/87/3.64/0
x_{11} (1000)	47/165/6.969/1	35/96/4.032/3
x_{12} (1000)	36/114/4.797/2	34/91/3.812/0

vector consisting of n ones. Hence (29) can be rewritten as the following quadratic program:

$$\begin{aligned} \min_{u,v} \quad & \frac{1}{2}\|y - A(u-v)\|_2^2 + \mu e_n^T u + \mu e_n^T v, \\ \text{s.t.} \quad & u \geq 0, \\ & v \geq 0. \end{aligned} \tag{30}$$

Furthermore, from [14], (30) can be written in following form

$$\begin{aligned} \min_{u,v} \quad & \frac{1}{2} z^T B z + c^T z, \\ \text{s.t.} \quad & z \geq 0, \end{aligned} \tag{31}$$

where

$$z = \begin{pmatrix} u \\ v \end{pmatrix}, \quad b = A^T y, \quad c = \mu e_{2n} + \begin{pmatrix} -b \\ b \end{pmatrix}, \quad B = \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix}. \tag{32}$$

It is obvious that B is a positive semidefinite matrix, hence, (30) is a convex QP problem. Figueiredo et al. [14] proposed a gradient projection method with BB step length for solving this problem.

Xiao et al. [7] indicated that the QP problem (30) is equivalent to the linear complementary problem: find $z \in \mathfrak{R}^{2n}$ such that

$$z \geq 0, \qquad Bz + c \geq 0, \quad \langle Bz + c, z \rangle = 0. \tag{33}$$

It is obvious that z is a solution of (33) if and only if it is a solution of the following nonlinear systems of equation

$$g(z) = \min\{z, Bz + c\} = 0. \tag{34}$$

The function g is vector valued, and the "min" is interpreted as componentwise minimum. Xiao et al. [7] proved that g is monotone. Hence, (34) can be solved effectively by the HSG-V algorithm.

Firstly, we consider a typical CS scenario that goal is to reconstruct a length-n sparse signal from m observations. We measure the quality of restoration by means of squared error (MSE) to the original signal $\overline{x}$, that is,

$$\text{MSE} = \frac{1}{n}\|\overline{x} - x^*\|^2, \tag{35}$$

where x^* is the restored signal. We test a small size signal with $n = 2^{12}$, $m = 2^{10}$, and the original contains 2^7 randomly nonzero elements. A is the Gaussian matrix which is generated by command $randn(m, n)$ in MATLAB. In this test, the measurement y is usually contaminated by noise, that is,

$$y = A\overline{x} + \omega, \tag{36}$$

where ω is the Gaussian noise distributed as $N(0, 0.0001)$. The parameters are taken as $\beta = 0.1$, $\sigma = 0.01$, $\epsilon = 10^{-10}$, $r = 1.2$, $M = 2$, μ is forced in decrease as the measure of [14]. To get better quality estimated signals, the process is terminated when the relative change of the objective function is below 10^{-5}, that is,

$$\frac{\|f_k - f_{k-1}\|}{\|f_{k-1}\|} < 10^{-5}, \tag{37}$$

where f_k denotes the function value at x_k.

Figures 3 and 4 report the results of HSG-V for a signal sparse reconstruction from its limited measurement. Comparing the first and last plot in Figure 3, we can find that the original sparse signal is restored almost exactly from the limited measurement. From the right plot in Figure 4, we observe that all the blue dots are circled by the red circles, which shows that the original signal has been found almost exactly. All together, this simple experiment shows that HSG-V algorithms perform well, and it is an efficient method to denoise sparse signals.

In the next experiment, we compare the performance of our algorithm with the SGCS algorithm for image deconvolution, in which A is a partial DWT matrix whose m rows are chosen randomly from $n \times n$ DWT matrix. To measure the quality of restoration, we use the SNR (signal to noise ratio) defined as $\text{SNR} = 20\log_{10}(\|x\|/\|x - \overline{x}\|)$. Figure 5 shows the original test images, and Figure 6 shows the restoration results by the SGCS and HSG-V algorithm, respectively. These results show that the HSG-V algorithm can restore blurred image quite well and obtain better quality reconstructed images in an efficient manner.

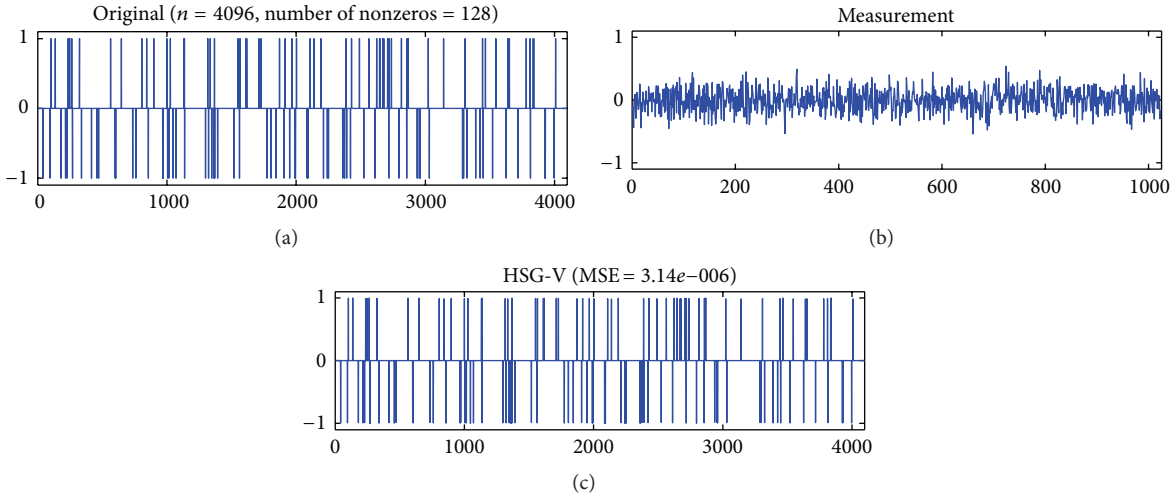

Figure 3: (a) Original signal with length 4096 and 128 nonzero elements. (b) The noisy measurement with length 1024. (c) Recovery signal by HSG-V with 232 iterations, 15.16 s CPU time in seconds, and 3.14e-06 error.

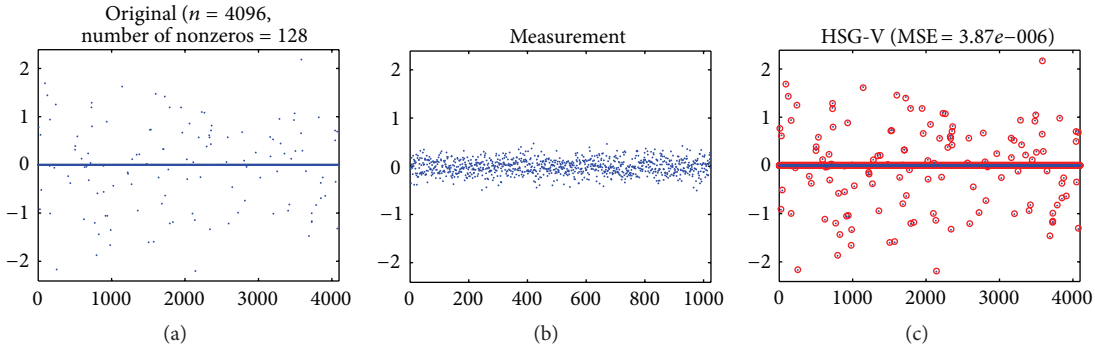

Figure 4: (a) Original signal with 128 nonzero elements, (b) noisy measurement, (c) restored signal (red circles) versus original signal (blue dots) with MSE = 6.72e − 06, 12.66 CPU time in seconds, and 188 iterations.

Figure 5: The original images: cameraman/barbara/bridge.

4. Conclusion

In this paper, we develop an adaptive prediction-correction method for solving nonlinear monotone equations. Under some assumptions, we establish its global convergence. Base on the prediction-correction method, an efficient hybrid spectral gradient (HSG-V) algorithm is proposed, which is composite of MSG-V, algorithm and SG algorithm. Numerical results show that the HSG-V algorithm is preferable and outperforms the MSG, MSG-V and SG algorithm. Moreover,

Figure 6: The blurred image (first column), the restored image by SGCS algorithm (second column), and HSG-V algorithm.

HSG-V algorithm is applied to solve ℓ_1-norm regularized problems arising from sparse signal reconstruction. Numerical experiments show that HSG-V algorithm works well, and it provides an efficient approach for compressed sensing and image deconvolution.

Appendix

The Test Problems

In this appendix, we list the test functions and the associated initial guess as follows.

Problem 1. $g : \Re^n \to \Re^n$,

$$g_i(x) = \frac{i}{10}\left(e^{x_i} - 1\right), \quad i = 1, 2, \ldots, n, \tag{A.1}$$

$g(x) = (g_1(x), g_2(x), \ldots, g_n(x))^T$, $x = (x_1, x_2, \ldots, x_n)^T$.

Problem 2. $g : \Re^n \to \Re^n$,

$$g_i(x) = x_i - \sin\left|x_i\right|, \quad i = 1, 2, \ldots, n, \tag{A.2}$$

$g(x) = (g_1(x), g_2(x), \ldots, g_n(x))^T$, $x = (x_1, x_2, \ldots, x_n)^T$.

Table 7: The initial points used in our test.

x_1	$(-20, 20, -20, 20, \ldots, -20, 20, -20, 20)^T$
x_2	$(-15, 15, -15, 15, \ldots, -15, 15, -15, 15)^T$
x_3	$(-10, 10, -10, 10, \ldots, -10, 10, -10, 10)^T$
x_4	$(-5, 5, -5, 5, \ldots, -5, 5, -5, 5)^T$
x_5	$(-1.2, 1, -1.2, 1, \ldots, -1.2, 1, -1.2, 1)^T$
x_6	$(-2, 1, -2, 1, \ldots, -2, 1, -2, 1)^T$
x_7	$(-1, 1, -1, 1, \ldots, -1, 1, -1, 1)^T$
x_8	$(-1/1, 1/2, \ldots, -1/(n-1)), 1/n)^T$
x_9	$(1/1, 1/2, \ldots, 1/(n-1)), 1/n)^T$
x_{10}	$(-1, -1, \ldots, -1, -1)^T$
x_{11}	$(1, 1, \ldots, 1, 1)^T$
x_{12}	$(0.1, 0.1 \ldots, 0.1, 0.1)$

Problem 3. $g : \mathfrak{R}^n \to \mathfrak{R}^n$ is given by

$$g(x) = Ax + f(x), \tag{A.3}$$

$f(x) = (e^{x_1} - 1, e^{x_2} - 1, \ldots, e^{x_n} - 1)^T$ and

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}. \tag{A.4}$$

It is noticed that Problems 1 and 3 are smooth at $x = 0$, while Problem 2 is nonsmooth (Table 7).

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (no. 11001060, 61262026, 81000613, 81101046), the JGZX Programme of Jiangxi Province (20112BCB23027), and the Science and Technology Programme of Jiangxi Education Committee (LDJH12088).

References

[1] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, USA, 1970.

[2] E. Zeidler, *Nonlinear functional analysis and its applications. II/B: Nonlinear monotone operators*, Springer, New York, NY, USA, 1990.

[3] A. N. Iusem and M. V. Solodov, "Newton-type methods with generalized distances for constrained optimization," *Optimization*, vol. 41, no. 3, pp. 257–278, 1997.

[4] Y.-B. Zhao and D. Li, "Monotonicity of fixed point and normal mappings associated with variational inequality and its application," *SIAM Journal on Optimization*, vol. 11, no. 4, pp. 962–973, 2001.

[5] M. V. Solodov and B. F. Svaiter, "A globally convergent inexact Newton method for systems of monotone equations," in *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, vol. 22, pp. 355–369, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[6] L. Zhang and W. Zhou, "Spectral gradient projection method for solving nonlinear monotone equations," *Journal of Computational and Applied Mathematics*, vol. 196, no. 2, pp. 478–484, 2006.

[7] Y. Xiao, Q. Wang, and Q. Hu, "Non-smooth equations based method for l_1-norm problems with applications to compressed sensing," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 74, no. 11, pp. 3570–3577, 2011.

[8] K. Yin, Y. Xiao, and M. Zhang, "Nonlinear conjugate gradient method for l_1-norm regularization problems in compressive sensing," *Journal of Computational Information Systems*, vol. 7, no. 3, pp. 880–885, 2011.

[9] G. Yu, "A derivative-free method for solving large-scale nonlinear systems of equations," *Journal of Industrial and Management Optimization*, vol. 6, no. 1, pp. 149–160, 2010.

[10] G. Yu, "Nonmonotone spectral gradient-type methods for large-scale unconstrained optimization and nonlinear systems of equations," *Pacific Journal of Optimization*, vol. 7, no. 2, pp. 387–404, 2011.

[11] G. Yu, S. Niu, and J. Ma, "Multivariate spectral gradient projection method for nonlinear monotone equations with convex constraints," *Journal of Industrial and Management Optimization*, vol. 9, no. 1, pp. 117–129, 2013.

[12] L. Han, G. Yu, and L. Guan, "Multivariate spectral gradient method for unconstrained optimization," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 621–630, 2008.

[13] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[14] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.