Nonlinear Convex Analysis and Optimization
Vol. 1, No. 1, 2022, pp. 47–65
https://ncao.design.blog

NONLINEAR CONVEX ANALYSIS AND OPTIMIZATION
An International Journal on Numerical, Computation and Applications

A Family of Conjugate Gradient Projection Method for Nonlinear Monotone Equations with Applications to Compressive Sensing

Aliyu Ibrahim Kiri [1], Auwal Bala Abubakar [2,*]

Numerical Optimization Research Group, Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano. Kano, Nigeria
[1]sssaly2014@gmail.com ; [2]ababubakar.mth@buk.edu.ng
* corresponding author

ABSTRACT

In this work, we propose a family of conjugate gradient projection method for nonlinear monotone equations with convex constraints. Under some appropriate assumptions, the global convergence of the method is established. Numerical examples reported shows that the method is competitive and efficient for solving monotone nonlinear equations. Furthermore, we apply the proposed algorithm to solve the sparse signal reconstruction problem in compressive sensing.

Article History
Received 21 Feb 2022
Accepted 31 May 2022
Keywords:
non-linear equations;
nonotone equation;
projection method;
compressive sensing
MSC
65K05; 90C52; 90C56

1. Introduction

Consider finding a point $x \in \Omega$ such that

$$F(x) = 0, \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and monotone, that is, $\langle F(x)-F(y), (x-y)\rangle \geq 0, \quad \forall x, y \in \mathbb{R}^n$, $\Omega \subset \mathbb{R}^n$ is nonempty and convex. The corresponding unconstrained problem when $\Omega = \mathbb{R}^n$ have been discussed extensively, and many iterative methods have been proposed by many researchers. Some examples are; Newton method, quasi-Newton method, Gauss-Newton, Levenberg-Marquardt method and their variants (see[1, 5, 6, 17, 7, 9, 14, 15, 16, 19, 21, 22, 24, 27, 28]). With a good initial guess, these algorithms are very attractive as they have fast convergence rate. However, there are relatively scanty literatures on constrained problem (1.1).

Constrained problem (1.1) has so many practical applications, for example in chemical equilibrium systems and economic equilibrium problems (see[20, 8]). Iterative methods for solving constrained monotone nonlinear equations have recently receive relatively high attention [18, 26, 34, 30, 32, 36, 33]. For example, in [26] Wang et al. proposed a projection method which requires no differentiability and regularity conditions for solving (1.1). Numerical experiments presented in the paper indicates the efficiency of the method. Ma and Wang [18] proposed a modified extragradient method for solving constrained monotone equations. A spectral gradient approach and a projection technique was presented by Yu et al. [33] for convex constrained problems. Using similar projection technique approach, Zheng [36] proposed a spectral gradient method for constrained problems. Also, Yu et al. in [32] proposed a multivariate spectral gradient

Please cite this article as: A.I. Kiri and A.B. Abubakar, A Family of Conjugate Gradient Projection Method for Nonlinear Monotone Equations with Applications to Compressive Sensing, Nonlinear Convex Anal. & Optim., Vol. 1 No. 1, 2022, 47–65.

projection (SGP) for solving problems of the form (1.1). A remarkable property of these gradient-type algorithms is that the direction does not depend on the gradient information, therefore can be applied to solve nonsmooth equations. However, Xiao and Zhu [30] proposed a projected conjugate gradient (CGD) to solve constrained problems. This method can be viewed as an extension of the CG−Descent method for solving convex constrained problems.

Motivated by these methods, we propose a family of conjugate gardient projection method for constrained nonlinear monotone equations, which is an extension of the method of Feng et al. [10] for solving convex constrained problems. The method possesses some properties, which are; (1) the method is derivative-free which implies its applicability in handling nonsmooth equations; (2) the global convergence was established without differentiability assummption and (3) it is independent of any merit function.

The remaining part of the paper is organized as follows. Section 2 provides the proposed method and its algorithm. Section 3 gives the global convergence and in Section 4 we report numerical results to show its practical performance, and apply it to solve the sparse signal reconstruction in compressive sensing.

2. Preliminaries and algorithm

In this section, we first give some basic concepts and properties. Let Ω be a nonempty closed convex subset of $\mathbb{R}^n$. Then for all $x \in \mathbb{R}^n$, its projection onto Ω is defined as

$$P_\Omega(x) = \arg\min\{\|x - y\| \ : y \in \Omega\}.$$

The map $P_\Omega : \mathbb{R}^n \to \Omega$ is called a projection operator and has the nonexpansive property, that is, for all $x, y \in \mathbb{R}^n$,

$$\|P_\Omega(x) - P_\Omega(y)\| \leq \|x - y\| \qquad \forall x, y \in \mathbb{R}^n. \tag{2.1}$$

The following propositions [31, 35] give some basic properties of the projection operator P_Ω.

Proposition 2.1. *Let $\Omega \subset \mathbb{R}^n$ be nonempty, closed and convex. Then for all $x \in \mathbb{R}^n$ and $y \in \Omega$,*

$$(P_\Omega(x) - x)^T(y - P_\Omega(x)) \geq 0.$$

Proposition 2.2. *Let $\Omega \subset \mathbb{R}^n$ be nonempty, closed and convex. Then for all $x, d \in \mathbb{R}^n$ and $\alpha \geq 0$, define $x(\alpha) := P_\Omega(x - \alpha d)$. Then $d^T(x(\alpha) - x)$ is nonincreasing with respect to $\alpha \geq 0$.*

The following assumptions hold throughout this paper.

Assumption A (i) The solution set of problem (1.1) is nonempty. (ii) The function F is Lipschitz continuous, that is there exists a positive constant L such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \tag{2.2}$$

for all $x, y \in \mathbb{R}^n$.

Assumption (ii) implies there is a positive constant τ such that

$$\|F(x_k)\| \leq \tau \quad \forall k \geq 0. \tag{2.3}$$

Now all is set to describe our proposed algorithm, which is an extension of the method in [10] to solve convex constrained problems.

Algorithm 1: Family of Conjugate Gradient Projection Method (FCG)

Step 0. Given an arbitrary initial point $x_0 \in \mathbb{R}^n$, parameters $0 < r < 1$, $\eta \geq 0$, $\sigma > 0$, $t > 0$, $\rho > 0$, $\epsilon > 0$, and set $k := 0$.

Step 1. If $\|F(x_k)\| \leq \epsilon$, stop, otherwise go to **Step 2**.

Step 2. Compute

$$d_k = \begin{cases} -F(x_k), & \text{if } k = 0, \\ -\left(1 + \beta_k \frac{F(x_k)^T d_{k-1}}{\|F(x_k)\|^2}\right) F(x_k) + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{2.4}$$

where β_k is such that

$$|\beta_k| \leq t \frac{\|F(x_k)\|}{\|d_{k-1}\|}, \quad \forall k \geq 1, \quad t > 0. \tag{2.5}$$

Step 3. Find the trial point $y_k = x_k + \alpha_k d_k$, where $\alpha_k = \rho r^{m_k}$ and m_k is the smallest nonnegative integer m such that

$$-\langle F(x_k + \rho r^m d_k), d_k \rangle \geq \sigma \rho r^m \|d_k\|. \tag{2.6}$$

Step 4. If $y_k \in \Omega$ and $\|F(y_k)\| \leq \epsilon$, stop. Else compute the next iterate

$$x_{k+1} = P_\Omega[x_k - \zeta_k F(y_k)]$$

where

$$\zeta_k = \frac{F(y_k)^T (x_k - y_k)}{\|F(y_k)\|^2}.$$

Step 5. Let $k = k + 1$ and go to **Step 1**.

Remark 2.3. From the definition of d_k, we have

$$\langle F(x_k), d_k \rangle = -F(x_k)^T F(x_k) - \frac{\beta_k F(x_k)^T F(x_k) F(x_k)^T d_{k-1}}{\|F(x_k)\|^2} + \beta_k F(x_k)^T d_{k-1} = -\|F(x_k)\|^2 \tag{2.7}$$

which means the direction d_k is sufficiently descent.

Remark 2.4. Remark 2.3 together with the Cauchy-Schwartz inequality implies that $\|d_k\| \geq \|F(x_k)\|$. Furthermore, by (2.4) and (2.5), we get

$$\begin{aligned} \|d_k\| &\leq \|F(x_k)\| + |\beta_k| \frac{\|F(x_k)\| \|d_{k-1}\|}{\|F(x_k)\|^2} \|F(x_k)\| + |\beta_k| \|d_{k-1}\| \\ &\leq \|F(x_k)\| + t\|F(x_k)\| + t\|F(x_k)\| \\ &\leq (1 + 2t)\|F(x_k)\|. \end{aligned}$$

Therefore,

$$\|F(x_k)\| \leq \|d_k\| \leq (1 + 2t)\|F(x_k)\|, \quad \forall k \geq 0, \tag{2.8}$$

which implies boundedness of the search direction.

3. Convergence analysis

To prove the global convergence of **Algorithm 1**, the following lemmas are needed. The following lemma shows that **Algorithm 1** is well-defined.

Lemma 3.1. *Suppose F is continuous, monotone and **Assumption A** (i) hold, then there exists a step-length α_k satisfying the line search (2.6) $\forall k \geq 0$.*

Proof. Suppose there exists $k_0 \geq 0$ such that (2.6) does not hold for any nonnegative integer i, i.e.,

$$-\langle F(x_k + \rho r^i d_k), d_k\rangle < \sigma \rho r^i \|d_k\|.$$

Using **Assumption A** and allowing $i \to \infty$, we get

$$-\langle F(x_{k_0}), d_{k_0}\rangle \leq 0. \tag{3.1}$$

Also from (2.7), we have

$$-\langle F(x_{k_0}), d_{k_0}\rangle \geq \|F(x_k)\|^2 > 0,$$

which contradicts (3.1). The proof is complete. ∎

The following theorem establishes the global convergence of **Algorithm 1**.

Theorem 3.2. *Let F be continuous and monotone, then the sequence $\{x_k\}$ generated by **Algorithm 1** converges globally to a solution of (1.1).*

Proof. We start by showing that the sequences $\{x_k\}$ and $\{y_k\}$ are bounded. Let x_* be an arbitrary solution of (1.1), then by monotonicity of F, we get

$$\langle F(y_k), x_k - x_*\rangle \geq \langle F(y_k), x_k - y_k\rangle. \tag{3.2}$$

Also by definition of y_k and the line search (2.6), we have

$$\langle F(y_k), x_k - y_k\rangle \geq \sigma \alpha_k \|d_k\|^2 \geq 0. \tag{3.3}$$

So, we have

$$\begin{aligned}
\|x_{k+1} - x_*\|^2 &= \|P_\Omega[x_k - \zeta_k F(y_k)] - x_*\|^2 \leq \|x_k - \zeta_k F(y_k) - x_*\| \\
&= \|x_k - x_*\|^2 - 2\zeta\langle F(y_k), x_k - x_*\rangle + t\|\zeta F(y_k)\|^2 \\
&\leq \|x_k - x_*\|^2 - 2\zeta\langle F(y_k), x_k - y_k\rangle + t\|\zeta F(y_k)\|^2 \\
&= \|x_k - x_*\|^2 - \frac{\langle F(y_k), x_k - y_k\rangle^2}{\|F(y_k)\|^2} \\
&= \|x_k - x_*\|^2 - \frac{\sigma^2 \alpha_k^4 \|d_k\|^4}{\|F(y_k)\|^4}.
\end{aligned}$$

Thus the sequence $\{\|x_k - x_*\|\}$ is non increasing and convergent, and hence $\{x_k\}$ is bounded. On the other hand (2.8) implies $\{d_k\}$ is bounded. Then, by $y_k = x_k + \alpha_k d_k$, the sequence $\{y_k\}$

is also bounded. Now, since F is continuous, there exists $M > 0$ such that $\|F(y_k)\| \leq M$ for all k. So,

$$\|x_{k+1} - x_*\|^2 \leq \|x_k - x_*\|^2 - \frac{\sigma^2 \alpha_k^4 \|d_k\|^4}{M^4}, \tag{3.4}$$

and we can deduce that

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0. \tag{3.5}$$

If $\liminf_{k\to\infty} \|d_k\| = 0$, we have $\liminf_{k\to\infty} \|F(x_k)\| = 0$. By continuity of F, the sequence $\{x_k\}$ has some accumulation point $\bar{x}$ such that $F(\bar{x}) = 0$. Since $\{\|x_k - x_*\|\}$ converges and $\bar{x}$ is an accumulation point of $\{x_k\}$, it follows that $\{x_k\}$ converges to $\bar{x}$.

If $\liminf_{k\to\infty} \|d_k\| > 0$, we have $\liminf_{k\to\infty} \|F(x_k)\| > 0$. By (3.5), it holds that $\lim_{k\to\infty} \alpha_k = 0$. Using the line search (2.6), $-F(x_k + \rho r^{m_i - 1} d_k)^T d_k < \sigma \rho r^{m_i - 1} \|d_k\|^2$ and the boundedness of $\{x_k\}, \{d_k\}$, we can choose a subsequence such that allowing k to go to infinity in the above inequality results

$$-\langle F(\bar{x}), \bar{d}\rangle \leq 0. \tag{3.6}$$

On the other hand, from (2.7) we have

$$-\langle F(\bar{x}), \bar{d}\rangle = \|F(\bar{x})\|^2 > 0. \tag{3.7}$$

Clearly, (3.6) contradicts (3.7). Therefore, $\liminf_{k\to\infty} \|F(x_k)\| > 0$ does not hold and the proof is complete. ∎

4. Numerical Experiment

In this section, for convenience sake, we denote **Algorithm 1** by FCG method. We also divided this section into two. First we compare FCG method with CGD method [30] by solving some monotone nonlinear equations with convex constraints using different initial points and several dimensions. Secondly, the FCG method is applied to solve the ℓ_1-regularization problem that arises from compressive sensing. All codes were written in MATLAB R2017a and run on a PC with intel COREi5 processor with 4GB of RAM and CPU 2.3GHZ.

4.1. Experiment on some convex constrained nonlinear monotone equations

FCG and CGD methods have same line search implementation. The specific parameters for each method are as follows:

FCG method: $\rho = 1,\ r = 0.5,\ \sigma = 0.01,\ t = 1$ and $\beta_k = \frac{\|F(x_k)\|}{\|d_{k-1}\|}$.

CGD method: $\rho = 1,\ r = 0.39,\ \sigma = 0.0001$.

All runs were stopped whenever

$\|F(x_k)\| < 10^{-5}$.

We test problems 1 to 6 with dimensions of $n = 1000, 5000, 10,000, 50,000, 100,000$ and different initial points: $x_1 = (1, 1, ..., 1)^T$, $x_2 = (2, 2, ..., 2)^T$, $x_3 = (3, 3, ..., 3)^T$, $x_4 = (5, 5, ..., 5)^T$, $x_5 = (8, 8, ..., 8)^T$, $x_6 = (0.5, 0.5, ...0.5)^T$, $x_7 = (0.1, 0.1, ..., 0.1)^T$, $x_8 = (10, 10, ..., 10)^T$. The results of experiment reported in Tables 1-6, which contain the number of iterations (ITER), number of function evaluations (FVAL), CPU time in seconds (TIME) and the norm at the approximate solution (NORM). The symbol '$-$' is used to indicate that the number of iterations exceeds 1000 and/or the number of function evaluations exceeds 2000.

The problems $F(x) = (f_1(x), f_2(x), ..., f_n(x))^T$, where $x = (x_1, x_2, ..., x_n)^T$, tested are listed as follows:

Problem 1 Modified exponential function

$$F_1(x) = e^{x_1} - 1$$
$$F_i(x) = e^{x_i} + x_{i-1} - 1 \text{ for } i = 2, 3, ..., n$$
$$\text{and} \quad \Omega = \mathbb{R}^n_+.$$

Problem 2 Logarithmic Function

$$F_i(x) = \ln(|x_i| + 1) - \frac{x_i}{n}, \text{ for } i = 2, 3, ..., n \quad \text{and} \quad \Omega = \mathbb{R}^n_+.$$

Problem 3 [37]

$$F_i(x) = 2x_i - \sin|x_i|, \ i = 1, 2, 3, ..., n \quad \text{and} \quad \Omega = \mathbb{R}^n_+.$$

Problem 4 Strictly convex function [26]

$$F_i(x) = e^{x_i} - 1, \text{ for } i = 2, 3, ..., n \quad \text{and} \quad \Omega = \mathbb{R}^n_+.$$

Problem 5 Linear monotone problem

$$F_1(x) = 2.5x_1 + x_2 - 1$$
$$F_i(x) = x_{i-1} + 2.5x_i + x_{i+1} - 1 \text{ for } i = 2, 3, ..., n-1$$
$$F_n(x) = x_{n-1} + 2.5x_n - 1$$
$$\text{and} \quad \Omega = \mathbb{R}^n_+.$$

Problem 6 Tridiagonal Exponential Problem [3]

$$F_1(x) = x_1 - e^{\cos(h(x_1+x_2))}$$
$$F_i(x) = x_i - e^{\cos(h(x_{i-1}+x_i+x_{i+1}))} \text{ for } i = 2, 3, ..., n-1$$
$$F_n(x) = x_n - e^{\cos(h(x_{n-1}+x_n))},$$
$$\text{where } h = \frac{1}{n+1}$$
$$\text{and} \quad \Omega = \mathbb{R}^n_+.$$

The results of the numerical performance indicate that the FCG method is more efficient than the CGD method for the given test problems as it solves more problems than CGD method which fails to solve most of the problems. In particular CGD method fails to solve problems 5 and 6 completely while FCG was able to solve the problems. Thus, FCG method is an effective tool for solving nonlinear monotone equations with convex constraints, especially for large-scale problems.

Table 1. Numerical Results for FCG and CGD for Problem 1 with given initial points and dimensions

DIMENSION	INITIAL POINT	FCG				CGD			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	30	151	0.029034	9.97E-06	-	-	-	-
	x_2	32	162	0.032225	8E-06	-	-	-	-
	x_3	29	147	0.024743	8.25E-06	-	-	-	-
	x_4	30	154	0.024078	8.98E-06	-	-	-	-
	x_5	32	169	0.024838	9.5E-06	-	-	-	-
	x_6	29	146	0.020169	9.09E-06	3	21	0.002236	0
	x_7	26	131	0.019619	6.87E-06	3	21	0.001834	0
	x_8	117	519	0.058816	9.57E-06	3	21	0.001666	0
5000	x_1	27	136	0.07062	9.27E-06	-	-	-	-
	x_2	29	147	0.069378	6.35E-06	-	-	-	-
	x_3	26	132	0.062323	9.74E-06	-	-	-	-
	x_4	27	139	0.056432	7.86E-06	-	-	-	-
	x_5	29	154	0.073783	9.28E-06	-	-	-	-
	x_6	26	131	0.057241	9.8E-06	3	21	0.00279	0
	x_7	24	121	0.061208	6.53E-06	3	21	0.001789	0
	x_8	225	945	0.359843	6.08E-06	3	21	0.001659	0
10000	x_1	27	136	0.110866	6.55E-06	-	-	-	-
	x_2	28	142	0.105131	6.99E-06	-	-	-	-
	x_3	26	132	0.095317	7.65E-06	-	-	-	-
	x_4	26	134	0.096757	9.22E-06	-	-	-	-
	x_5	29	154	0.10975	6.59E-06	-	-	-	-
	x_6	26	131	0.091813	7.43E-06	3	21	0.002557	0
	x_7	23	116	0.081306	9.74E-06	3	21	0.001912	0
	x_8	207	872	0.667213	9.86E-06	3	21	0.002028	0
50000	x_1	27	136	0.416897	5.45E-06	-	-	-	-
	x_2	28	142	0.4511	5.54E-06	-	-	-	-
	x_3	26	132	0.41667	7.7E-06	-	-	-	-
	x_4	26	134	0.42627	8.15E-06	-	-	-	-
	x_5	29	154	0.504924	5.43E-06	-	-	-	-
	x_6	26	131	0.414027	7.1E-06	3	21	0.002425	0
	x_7	24	121	0.44835	6.42E-06	3	21	0.001936	0
	x_8	193	816	2.51444	9.92E-06	3	21	0.001939	0
100000	x_1	27	136	0.991751	6.37E-06	-	-	-	-
	x_2	28	142	1.260811	6.4E-06	-	-	-	-
	x_3	26	132	1.424801	9.41E-06	-	-	-	-
	x_4	26	134	1.530526	9.67E-06	-	-	-	-
	x_5	29	154	1.381487	6.31E-06	-	-	-	-
	x_6	26	131	1.004984	8.59E-06	3	21	0.002641	0
	x_7	24	121	0.854936	8.11E-06	3	21	0.003401	0
	x_8	78	370	2.377291	9.74E-06	3	21	0.001946	0

Table 2. Numerical Results for FCG and CGD for Problem 2 with given initial points and dimensions

DIMENSION	INITIAL POINT	FCG				CGD			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	6	19	0.004914	3.6E-08	3	10	0.004752	0
	x_2	7	22	0.006036	1.74E-08	-	-	-	-
	x_3	7	22	0.004458	2.21E-06	-	-	-	-
	x_4	8	25	0.004995	5.45E-06	-	-	-	-
	x_5	10	31	0.009144	8.47E-08	-	-	-	-
	x_6	5	16	0.004309	4.37E-07	12	37	0.003022	0
	x_7	4	13	0.003884	5.17E-07	10	31	0.003156	0
	x_8	11	34	0.008439	2.64E-08	10	31	0.002395	0
5000	x_1	6	19	0.013848	6.26E-09	3	10	0.010011	0
	x_2	7	22	0.015027	2.36E-09	-	-	-	-
	x_3	7	22	0.01877	8.93E-07	-	-	-	-
	x_4	8	25	0.01957	2.58E-06	-	-	-	-
	x_5	10	31	0.023509	1.74E-08	6	19	0.018264	0
	x_6	5	16	0.011425	1.42E-07	12	37	0.003034	0
	x_7	4	13	0.008892	1.75E-07	10	31	0.002234	0
	x_8	11	34	0.019901	3.7E-09	10	31	0.002247	0
10000	x_1	6	20	0.025059	3.62E-09	3	10	0.017026	0
	x_2	7	23	0.027807	1.24E-09	-	-	-	-
	x_3	7	22	0.0271	6.86E-07	-	-	-	-
	x_4	8	25	0.024772	2.22E-06	-	-	-	-
	x_5	10	32	0.034177	1.07E-08	12	48	0.056075	0
	x_6	5	17	0.021084	9.73E-08	12	37	0.004599	0
	x_7	4	13	0.016566	1.21E-07	10	31	0.002371	0
	x_8	11	35	0.038309	2E-09	10	31	0.002892	0
50000	x_1	8	29	0.113023	8.3E-06	3	10	0.066789	0
	x_2	7	24	0.092315	1E-05	-	-	-	-
	x_3	17	64	0.238282	5.77E-06	-	-	-	-
	x_4	19	71	0.27681	7.15E-06	-	-	-	-
	x_5	14	49	0.198167	6.54E-06	-	-	-	-
	x_6	12	46	0.167172	7.79E-06	12	37	0.003578	0
	x_7	11	43	0.154196	9.67E-06	10	31	0.002085	0
	x_8	12	40	0.161888	8.52E-06	10	31	0.002224	0
100000	x_1	9	33	0.237879	5.71E-06	3	10	0.123028	0
	x_2	8	28	0.211838	6.74E-06	-	-	-	-
	x_3	17	64	0.507595	8.14E-06	-	-	-	-
	x_4	20	75	0.580962	5.05E-06	-	-	-	-
	x_5	14	49	0.394332	9.09E-06	-	-	-	-
	x_6	13	50	0.380856	5.48E-06	12	37	0.003665	0
	x_7	12	47	0.348441	6.8E-06	10	31	0.002801	0
	x_8	13	44	0.392105	5.8E-06	10	31	0.002108	0

Table 3. Numerical Results for FCG and CGD for Problem 3 with given initial points and dimensions

DIMENSION	INITIAL POINT	FCG				CGD			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	23	93	0.010923	6.1E-06	13	42	0.010427	0
	x_2	23	93	0.013051	6.55E-06	-	-	-	-
	x_3	20	81	0.01293	8.5E-06	9	39	0.009423	0
	x_4	24	98	0.014461	5.63E-06	-	-	-	-
	x_5	23	93	0.01694	7.07E-06	-	-	-	-
	x_6	22	89	0.021464	7.14E-06	7	26	0.001876	0
	x_7	20	81	0.014054	6.02E-06	10	42	0.002722	0
	x_8	25	103	0.017473	5.94E-06	9	39	0.002248	0
5000	x_1	24	97	0.041347	6.82E-06	-	-	-	-
	x_2	24	97	0.040248	7.32E-06	-	-	-	-
	x_3	21	85	0.037918	9.51E-06	-	-	-	-
	x_4	25	102	0.043526	6.29E-06	-	-	-	-
	x_5	24	97	0.04303	7.9E-06	-	-	-	-
	x_6	23	93	0.03983	7.98E-06	7	26	0.002126	0
	x_7	21	85	0.038976	6.73E-06	10	42	0.003248	0
	x_8	26	107	0.044616	6.64E-06	9	39	0.002423	0
10000	x_1	24	97	0.069348	9.65E-06	-	-	-	-
	x_2	25	101	0.077087	5.18E-06	-	-	-	-
	x_3	22	89	0.065376	6.72E-06	-	-	-	-
	x_4	25	102	0.079566	8.9E-06	-	-	-	-
	x_5	25	101	0.072858	5.59E-06	-	-	-	-
	x_6	24	97	0.070883	5.64E-06	7	26	0.002492	0
	x_7	21	85	0.058127	9.52E-06	10	42	0.00411	0
	x_8	26	107	0.086316	9.39E-06	9	39	0.002452	0
50000	x_1	26	105	0.326713	5.39E-06	-	-	-	-
	x_2	26	105	0.300343	5.79E-06	-	-	-	-
	x_3	23	93	0.271562	7.52E-06	-	-	-	-
	x_4	26	106	0.308558	9.95E-06	-	-	-	-
	x_5	26	105	0.342744	6.25E-06	-	-	-	-
	x_6	25	101	0.310528	6.31E-06	7	26	0.002338	0
	x_7	23	93	0.266319	5.32E-06	10	42	0.002626	0
	x_8	28	115	0.3389	5.25E-06	9	39	0.00371	0
100000	x_1	26	105	0.609266	7.63E-06	-	-	-	-
	x_2	26	105	0.640604	8.19E-06	-	-	-	-
	x_3	24	97	0.604267	5.31E-06	-	-	-	-
	x_4	27	110	0.666098	7.04E-06	-	-	-	-
	x_5	26	105	0.622149	8.84E-06	-	-	-	-
	x_6	25	101	0.621465	8.92E-06	7	26	0.002508	0
	x_7	23	93	0.567894	7.52E-06	10	42	0.00269	0
	x_8	28	115	0.724637	7.42E-06	9	39	0.002607	0

Table 4. Numerical Results for FCG and CGD for Problem 4 with given initial points and dimensions

DIMENSION	INITIAL POINT	**FCG**				**CGD**			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	21	85	0.010375	7.37E-06	-	-	-	-
	x_2	22	90	0.01171	7.98E-06	11	45	0.008779	0
	x_3	22	91	0.009893	9.46E-06	3	21	0.004732	0
	x_4	22	93	0.016824	7.88E-06	3	21	0.004487	0
	x_5	23	102	0.012653	5.28E-06	3	21	0.005234	0
	x_6	21	85	0.016972	8.87E-06	3	21	0.001853	0
	x_7	20	81	0.012323	5.45E-06	3	21	0.00177	0
	x_8	2	18	0.005313	0	3	21	0.001968	0
5000	x_1	22	89	0.037557	8.24E-06	-	-	-	-
	x_2	23	94	0.033656	8.93E-06	-	-	-	-
	x_3	24	99	0.033578	5.29E-06	3	21	0.010306	0
	x_4	23	97	0.035841	8.81E-06	3	21	0.012105	0
	x_5	24	106	0.047643	5.9E-06	3	21	0.012359	0
	x_6	22	89	0.03202	9.91E-06	3	21	0.00166	0
	x_7	21	85	0.050125	6.1E-06	3	21	0.001721	0
	x_8	2	18	0.009803	0	3	21	0.001635	0
10000	x_1	23	93	0.065522	5.83E-06	-	-	-	-
	x_2	24	98	0.053947	6.31E-06	-	-	-	-
	x_3	24	99	0.060434	7.48E-06	3	21	0.018915	0
	x_4	24	101	0.05745	6.23E-06	3	21	0.017853	0
	x_5	24	106	0.074314	8.35E-06	3	21	0.016733	0
	x_6	23	93	0.054277	7.01E-06	3	21	0.001641	0
	x_7	21	85	0.057827	8.62E-06	3	21	0.00238	0
	x_8	2	18	0.018646	0	3	21	0.001563	0
50000	x_1	24	97	0.264969	6.51E-06	-	-	-	-
	x_2	25	102	0.242806	7.06E-06	-	-	-	-
	x_3	25	103	0.253747	8.36E-06	3	21	0.072756	0
	x_4	25	105	0.242377	6.96E-06	3	21	0.071677	0
	x_5	25	110	0.24953	9.33E-06	3	21	0.072242	0
	x_6	24	97	0.235073	7.84E-06	3	21	0.001643	0
	x_7	22	89	0.214897	9.64E-06	3	21	0.002218	0
	x_8	2	18	0.064875	0	3	21	0.002332	0
100000	x_1	24	97	0.450026	9.21E-06	-	-	-	-
	x_2	25	102	0.496804	9.98E-06	-	-	-	-
	x_3	26	107	0.529191	5.91E-06	3	21	0.155644	0
	x_4	25	105	0.515341	9.85E-06	3	21	0.146987	0
	x_5	26	114	0.553523	6.6E-06	3	21	0.141577	0
	x_6	25	101	0.465119	5.54E-06	3	21	0.001987	0
	x_7	23	93	0.420296	6.82E-06	3	21	0.001702	0
	x_8	2	18	0.135172	0	3	21	0.002578	0

Table 5. Numerical Results for FCG and CGD for Problem 5 with given initial points and dimensions

DIMENSION	INITIAL POINT	FCG				CGD			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	152	802	0.075299	9.2E-06	-	-	-	-
	x_2	139	736	0.073897	7.3E-06	-	-	-	-
	x_3	136	726	0.072044	6.7E-06	-	-	-	-
	x_4	165	872	0.093292	7.4E-06	-	-	-	-
	x_5	165	873	0.084202	8.2E-06	-	-	-	-
	x_6	150	793	0.080311	7.2E-06	-	-	-	-
	x_7	153	808	0.077297	7.4E-06	-	-	-	-
	x_8	183	964	0.09986	7E-06	-	-	-	-
5000	x_1	151	796	0.302958	9.1E-06	-	-	-	-
	x_2	108	580	0.222843	8.5E-06	-	-	-	-
	x_3	133	711	0.244279	9.8E-06	-	-	-	-
	x_4	155	822	0.268972	9.7E-06	-	-	-	-
	x_5	177	934	0.331852	7.1E-06	-	-	-	-
	x_6	143	758	0.247833	1E-05	-	-	-	-
	x_7	145	768	0.260931	9.8E-06	-	-	-	-
	x_8	177	934	0.376048	9.4E-06	-	-	-	-
10000	x_1	151	796	0.582134	9.1E-06	-	-	-	-
	x_2	101	545	0.405084	7.5E-06	-	-	-	-
	x_3	150	796	0.59398	9.5E-06	-	-	-	-
	x_4	172	908	0.666833	9.1E-06	-	-	-	-
	x_5	158	839	0.60491	9.3E-06	-	-	-	-
	x_6	159	839	0.594092	9.9E-06	-	-	-	-
	x_7	150	793	0.561624	9.7E-06	-	-	-	-
	x_8	171	905	0.651769	7.7E-06	-	-	-	-
50000	x_1	147	776	2.228341	9E-06	-	-	-	-
	x_2	89	486	1.463798	8.6E-06	-	-	-	-
	x_3	147	780	2.378053	9.1E-06	-	-	-	-
	x_4	156	829	2.491981	6.7E-06	-	-	-	-
	x_5	176	930	2.74415	8.2E-06	-	-	-	-
	x_6	177	930	2.722724	8.6E-06	-	-	-	-
	x_7	151	799	2.321288	7.7E-06	-	-	-	-
	x_8	159	845	2.448193	9.8E-06	-	-	-	-
100000	x_1	148	782	5.294027	7.9E-06	-	-	-	-
	x_2	89	486	3.293977	7.9E-06	-	-	-	-
	x_3	132	707	4.714172	9.5E-06	-	-	-	-
	x_4	144	769	5.166183	9.4E-06	-	-	-	-
	x_5	165	875	5.898258	7.3E-06	-	-	-	-
	x_6	171	900	5.94172	8E-06	-	-	-	-
	x_7	152	805	5.387028	6.7E-06	-	-	-	-
	x_8	165	875	5.867254	9.7E-06	-	-	-	-

Table 6. Numerical Results for FCG and CGD for Problem 6 with given initial points and dimensions

DIMENSION	INITIAL POINT	FCG				CGD			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	x_1	24	97	0.018177	6.47E-06	-	-	-	-
	x_2	23	93	0.020801	5.41E-06	-	-	-	-
	x_3	21	85	0.018842	8.49E-06	-	-	-	-
	x_4	24	97	0.017617	8.59E-06	-	-	-	-
	x_5	25	101	0.018456	9.94E-06	-	-	-	-
1000	x_6	24	97	0.017399	8.35E-06	-	-	-	-
	x_7	24	97	0.018177	9.86E-06	-	-	-	-
	x_8	26	105	0.028836	6.85E-06	-	-	-	-
	x_1	25	101	0.071229	7.24E-06	-	-	-	-
	x_2	24	97	0.069072	6.05E-06	-	-	-	-
	x_3	22	89	0.062428	9.5E-06	-	-	-	-
	x_4	25	101	0.093955	9.62E-06	-	-	-	-
	x_5	27	109	0.072351	5.56E-06	-	-	-	-
5000	x_6	25	101	0.065859	9.35E-06	-	-	-	-
	x_7	26	105	0.090732	5.52E-06	-	-	-	-
	x_8	27	109	0.071772	7.67E-06	-	-	-	-
	x_1	26	105	0.154804	5.12E-06	-	-	-	-
	x_2	24	97	0.117887	8.56E-06	-	-	-	-
	x_3	23	93	0.114087	6.72E-06	-	-	-	-
	x_4	26	105	0.148228	6.8E-06	-	-	-	-
	x_5	27	109	0.144108	7.87E-06	-	-	-	-
10000	x_6	26	105	0.129431	6.61E-06	-	-	-	-
	x_7	26	105	0.130249	7.8E-06	-	-	-	-
	x_8	28	113	0.149137	5.43E-06	-	-	-	-
	x_1	27	109	0.635653	5.73E-06	-	-	-	-
	x_2	25	101	0.533873	9.57E-06	-	-	-	-
	x_3	24	97	0.550562	7.51E-06	-	-	-	-
	x_4	27	109	0.605892	7.6E-06	-	-	-	-
	x_5	28	113	0.636995	8.8E-06	-	-	-	-
50000	x_6	27	109	0.613244	7.39E-06	-	-	-	-
	x_7	27	109	0.602482	8.72E-06	-	-	-	-
	x_8	29	117	0.645397	6.07E-06	-	-	-	-
	x_1	27	109	1.250775	8.1E-06	-	-	-	-
	x_2	26	105	1.246817	6.77E-06	-	-	-	-
	x_3	25	101	1.158025	5.31E-06	-	-	-	-
	x_4	28	113	1.301028	5.38E-06	-	-	-	-
	x_5	29	117	1.338954	6.22E-06	-	-	-	-
100000	x_6	28	113	1.324453	5.23E-06	-	-	-	-
	x_7	28	113	1.299459	6.17E-06	-	-	-	-
	x_8	29	117	1.355021	8.58E-06	-	-	-	-

4.2. Experiments on the ℓ_1-norm regularization problem in compressive sensing

There are many problems in signal processing and statistical inference involving finding sparse solutions to ill-conditioned linear systems of equations. Among popular approach is minimizing an objective function which contains quadratic (ℓ_2) error term and a sparse ℓ_1-regularization term, i.e.,

$$\min_{x} \frac{1}{2}\|y - Ax\|_2^2 + \omega\|x\|_1, \tag{4.1}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^k$ is an observation, $A \in \mathbb{R}^{k\times n}$ ($k << n$) is a linear operator, ω is a nonnegative parameter, $\|x\|_2$ denotes the Euclidean norm of x and $\|x\|_1 = \sum_{i=1}^{n}|x_i|$ is the ℓ_1-norm of x. It is easy to see that problem (4.1) is a convex unconstrained minimization problem. Due to the fact that if the original signal is sparse or approximately sparse in some orthogonal basis, problem (4.1) frequently appears in compressive sensing, and hence an exact restoration can be produced by solving (4.1).

Iterative methods for solving (4.1) have been been presented in many literatures, (see [11, 13, 2, 12, 25, 4]). The most popular method among these methods is the gradient based method and the earliest gradient projection method for sparse reconstruction (GPRS) was proposed by Figueiredo et al. [12]. The first step of the GPRS method is to express (4.1) as a quadratic problem using the following process. Let $x \in \mathbb{R}^n$ and splitting it into its positive and negative parts. Then x can be formulated as

$$x = u - v, \qquad u \geq 0, \quad v \geq 0,$$

where $u_i = (x_i)_+$, $v_i = (-x_i)_+$ for all $i = 1, 2, ..., n$, and $(.)_+ = \max\{0, .\}$. By definition of ℓ_1-norm, we have $\|x\|_1 = e_n^T u + e_n^T v$, where $e_n = (1, 1, ..., 1)^T \in \mathbb{R}^n$. Now (4.1) can be written as

$$\min_{u,v} \frac{1}{2}\|y - A(u - v)\|_2^2 + \omega e_n^T u + \omega e_n^T v, \qquad u \geq 0, \quad v \geq 0, \tag{4.2}$$

which is a bound-constrained quadratic program. However, from [12], equation (4.2) can be written in standard form as

$$\min_{z} \frac{1}{2} z^T Bz + c^T z, \qquad \text{such that} \quad z \geq 0, \tag{4.3}$$

where $z = \begin{pmatrix} u \\ v \end{pmatrix}$, $\quad c = \omega e_{2n} + \begin{pmatrix} -b \\ b \end{pmatrix}$, $\quad b = A^T y$, $\quad B = \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix}$.
Clearly, B is a positive semidefinite matrix, which implies that equation (4.3) is a convex quadratic problem.

Xiao et al. [30] translated (4.3) into a linear variable inequality problem which is equivalent to a linear complementarity problem. Furthermore, they pointed out that z is a solution of the linear complementarity problem if and only if it is a solution of the nonlinear equation:

$$F(z) = \min\{z, Bz + c\} = 0. \tag{4.4}$$

It was proved in [29, 23] that $F(z)$ is continuous and monotone. Therefore problem (4.1) can be translated into problem (1.1) and thus FCG method can be applied to solve (4.1).

In this experiment, we consider a simple compressive sensing possible situation, where our goal is to reconstruct a sparse signal of length n from k observations. The quality of restoration is assessed by mean of squared error (MSE) to the original signal $\bar{x}$,

$$MSE = \frac{1}{n}\|\bar{x} - x_*\|^2,$$

where x_* is the recovered or restored signal. The signal size is choosen as $n = 2^{12}$, $k = 2^{10}$ and the original signal contains 2^7 randomly nonzero elements. A is the Gaussian matrix generated by the command $rand(m, n)$ in MATLAB. In addition, the measurement y is distributed with noise, that is, $y = Ax + \mu$, where μ is the Gaussian noise distributed normally with mean 0 and variance 10^{-4} ($N(0, 10^{-4})$).

To show the performance of the FCG method in compressive sensing, we compare it with the CGD method. The parameters in both FCG and CGD methods are chosen as $\rho = 10$, $\sigma = 10^{-4}$ and $r = 0.5$, which came from [30]. After series of experiments, we observe that for FCG method, the parameter η has a great impact on the restoration of signal. Finally, we choose $\eta = 0.2$ in our experiment and the merit function used is $f(x) = \frac{1}{2}\|y - Ax\|_2^2 + \omega\|x\|_1$. To achieve fairness in comparison, each code was run from same initial point, same continuation technique on the parameter ω, and observed only the behaviour of the convergence of each method to have a similar accurate solution. The experiment is initialized by $x_0 = A^T y$ and terminates when

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

where f_k is the function evaluation at x_k.

In Fig. 1, FCG and CGD methods recovered the disturbed signal almost exactly. In order to show visually the performance of both methods, four figures were plotted to demonstrate their convergence behaviour based on MSE, objective function values, number of iterations and CPU time, see Fig. 2 – 5. Furthermore, the experiment was repeated for 10 different noise samples and the average was also computed, see Table 7. From the Table, it can be observed that the FCG is more efficient as it has fewer iterations and CPU time than CGD method in most cases.

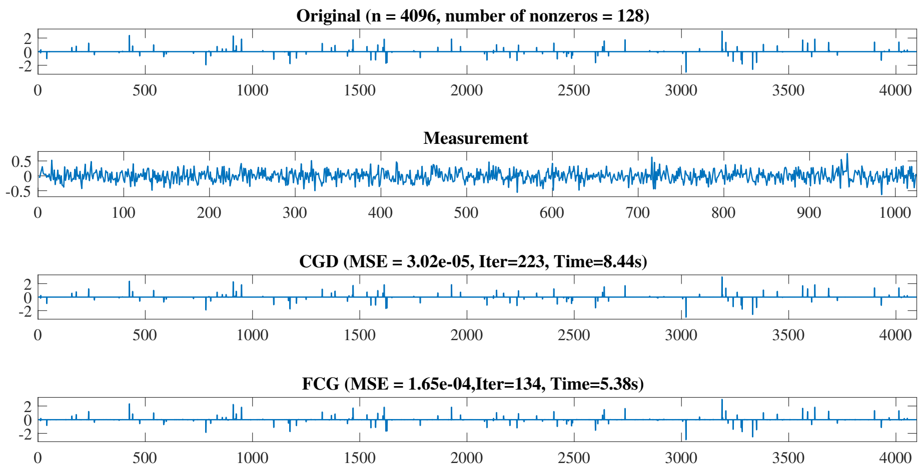

Fig. 1. From top to bottom: the original image, the measurement, and the recovered signals by CGD and FCG methods.

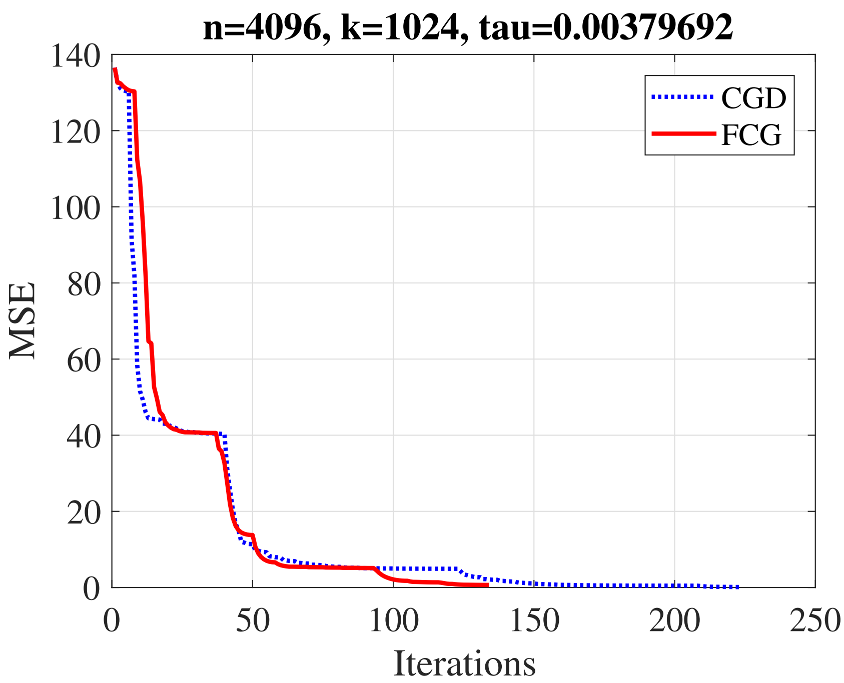

Fig. 2. Iterations

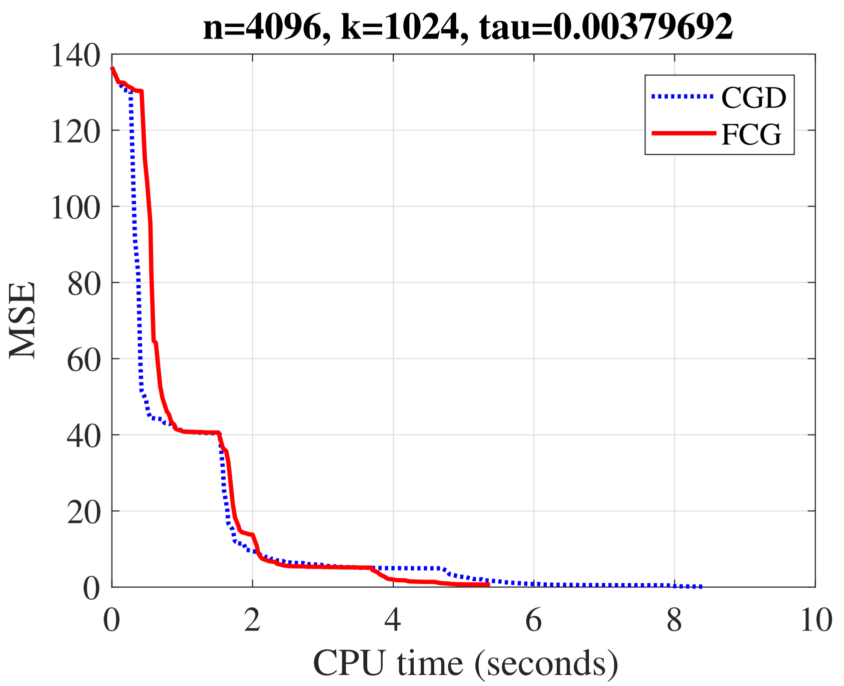

Fig. 3. CPU time (seconds)

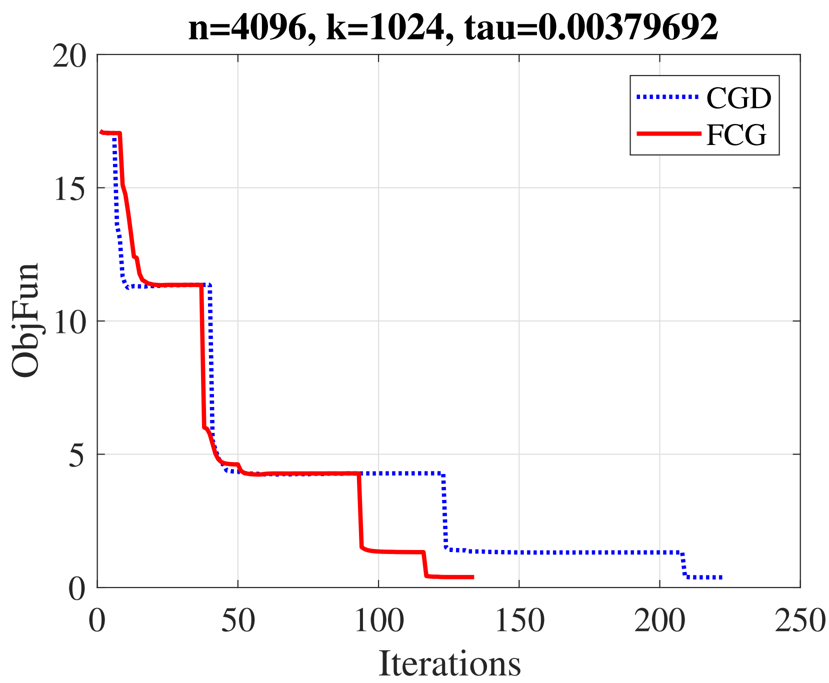

Fig. 4. Iterations

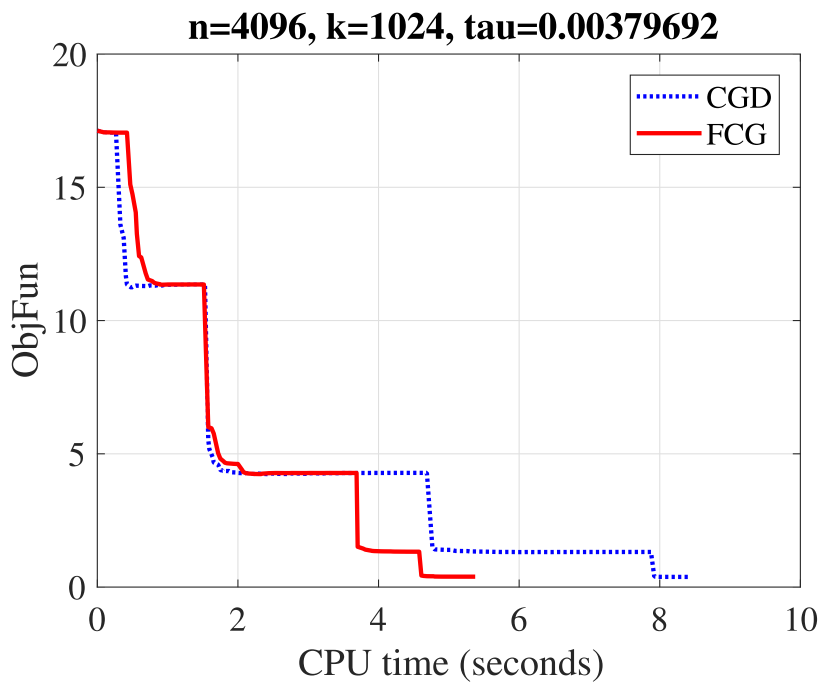

Fig. 5. CPU time (seconds)

Table 7. Ten experiment results together with average result of ℓ_1-norm regularization problem for FCG and CGD methods

	FCG			CGD		
	MSE	ITER	CPU(s)	MSE	ITER	CPU(s)
	2.31E-04	100	3.98	3.40E-05	196	7.31
	1.65E-04	134	5.38	3.02E-05	223	8.44
	1.40E-04	130	5.14	5.21E-05	164	6.3
	1.65E-04	134	5.59	3.02E-05	223	8.69
	1.75E-04	127	4.83	4.48E-05	218	8.14
$\eta = 0.2$	6.78E-04	169	6.38	1.85E-05	215	8.44
	1.47E-04	137	5.28	4.94E-05	191	8.66
	2.72E-04	94	4.53	4.33E-05	224	8.83
	1.67E-04	117	4.89	1.26E-05	135	5.55
	1.07E-04	119	4.64	2.78E-05	181	6.91
Average	2.25E-04	126.1	5.064	3.43E-05	197	7.727

5. Conclusions

In this article, a family of conjugate gradient projection method for solving nonlinear monotone equation with convex constraints was proposed. The proposed method is suitable for for solving nonsmooth equations as it does not require Jacobian information of the nonlinear equations. The global convergence of the proposed method was established under suitable conditions.

We can view the the proposed method as an extension of the method in [10] to solve convex constrained problems. Numerical results show that the proposed method is more efficient than the CGD method for the given constrained problems. Furthermore, the proposed method can be applied to solve ℓ_1-norm regularization problem in compressive sensing.

Acknowledgements

The authors are grateful to the anonymous reviewers for helpful suggestions that improved the paper.

References

[1] M. Al-Baali, E. Spedicato, and F. Maggioni, Broyden's quasi-Newton methods for a nonlinear system of equations and unconstrained optimization: a review and open problems, Optimization Methods and Software, 29 (5) (2014) 937–954.

[2] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM journal on imaging sciences, 2 (1) (2009) 183–202.

[3] Y. Bing and G. Lin, An efficient implementation of merrill's method for sparse or partially separable systems of nonlinear equations, SIAM Journal on Optimization, 1 (2) (1991) 206–221.

[4] E.G. Birgin, J.M. Martínez, and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, SIAM Journal on Optimization, 10 (4) (2000) 1196–1211.

[5] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, Math Comput, 19 (1965) 577–593.

[6] A. Cordero and J.R. Torregrosa, Variants of Newton's method for functions of several variables, Appl Math Comput, 183 (2006) 199–208.

[7] R.S. Dembo, S.C. Eisenstat, and T. Steihaug, Inexact Newton methods, SIAM Journal on Numerical analysis, 19 (2) (1982) 400–408.

[8] S.P. Dirkse and M.C. Ferris, MCPLIB: A collection of nonlinear mixed complementarity problems, Optimization Methods and Software, 5 (4) (1995) 319–345.

[9] M.T. Dirvashi and A. Barati, A third-order Newton-type method to solve systems of nonlinear equations, Appl Math Comput, 187 (2007) 630–635.

[10] D. Feng, M. Sun, and X. Wang, A family of conjugate gradient methods for large-scale nonlinear equations, Journal of Inequalities and Applications, Article number: 236 (2017) 8 pages.

[11] M.A. Figueiredo and R.D. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Transactions on Image Processing, 12 (8) (2003) 906–916.

[12] M.A. Figueiredo, R.D. Nowak, and S.J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, IEEE Journal of selected topics in signal processing, 1 (4) (2007) 586–597.

[13] E.T. Hale, W. Yin, and Y. Zhang, A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing, CAAM TR07-07, Rice University, 2007.

[14] M. Hassan and M.Y. Wazir, On Broyden-like update via some quadratures for solving nonlinear systems of equations, Turkish Journal of Mathematics, 39 (3) (2015) 335–345.

[15] C.T. Kelly, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.

[16] C.T. Kelly. Solving Nonliner Equations with Newton's Method, SIAM, Philadelphia, 2003.

[17] D. Knoll and D. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, J. Comput. Phy., 193 (2004) 357–397.

[18] F. Ma and C. Wang, Modified projection method for solving a system of monotone equations with convex constraints, Journal of Applied Mathematics and Computing, 34 (1-2) (2010) 47–56.

[19] J. M. Martinéz, Practical quasi-newton methods for solving nonlinear systems, J Comput App Math, 124 (2000) 97–121.

[20] K. Meintjes and A.P. Morgan, A methodology for solving chemical equilibrium systems, Applied Mathematics and Computation, 22 (4) (1987) 333–361.

[21] J.J. More and J.A. Trangenstein, On the global convergence of Broyden's method, Mathematics of Computation, 30 (1976) 523–540.

[22] K. Natasa and L. Zorana, Newton-like method with modification of the right-hand-side vector, Math Comput, 71 (2002) 237–250.

[23] J.-S. Pang, Inexact Newton methods for the nonlinear complementarity problem, Mathematical Programming, 36 (1) (1986) 54–71.

[24] B.C. Shin, M.T. Dirvashi, and C.H. Kim, A comaprison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems, Appl Math Comput, 217 (2010) 3190–3198.

[25] E. Van Den Berg and M.P. Friedlander, Probing the pareto frontier for basis pursuit solutions, SIAM Journal on Scientific Computing, 31 (2) (2008) 890–912.

[26] C. Wang, Y. Wang, and C. Xu, A projection method for a system of nonlinear monotone equations with convex constraints, Mathematical Methods of Operations Research, 66 (1) (2007) 33–46.

[27] M. Waziri, W. Leong, and M. Hassan, Diagonal Broyden-like method for large-scale systems of nonlinear equations, Malaysian Journal of Mathematical Sciences, 6 (1) (2012) 59–73.

[28] M. Waziri, W. Leong, M. Hassan, and M. Monsi, A new Newton's method with diagonal jacobian approximation for systems of nonlinear equations, Journal of Mathematics and Statistics, 6 (3) (2010) 246–252.

[29] 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, 74 (11) (2011) 3570–3577.

[30] Y. Xiao and H. Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing, Journal of Mathematical Analysis and Applications, 405 (1) (2013) 310–319.

[31] N. Xiu, C. Wang, and J. Zhang, Convergence properties of projection and contraction methods for variational inequality problems, Applied Mathematics and Optimization, 43 (2) (2001) 147–168.

[32] 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, 9 (1) (2013) 117–129.

[33] Z. Yu, J. Lin, J. Sun, Y. Xiao, L. Liu, and Z. Li, Spectral gradient projection method for monotone nonlinear equations with convex constraints, Applied numerical mathematics, 59 (10) (2009) 2416–2423, 2009.

[34] X. Yunhai, W. Qiuyu and H. Qingjie, Non-smooth equations based methods for l_1-norm problems with applications to compressed sensing, Nonlinear Analysis, 74 (2011) 3570–3577.

[35] E.H. Zarantonello, Projections on convex sets in hilbert space and spectral theory: Part i. projections on convex sets: Part ii. spectral theory, Contributions to Nonlinear Functional Analysis, Elsevier (1971), 237–424.

[36] L. Zheng, A new projection algorithm for solving a system of nonlinear equations with convex constraints, Bulletin of the Korean Mathematical Society, 50 (3) (2013) 823–832.

[37] W. Zhou and D. Li, Limited memory BFGS method for nonlinear monotone equations, Journal of Computational Mathematics, 25 (1) (2007) 89–96.