Thai Journal of Mathematics : 128-152
Special Issue: Annual Meeting in Mathematics 2018

http://thaijmath.in.cmu.ac.th
Online ISSN 1686-0209

A Descent Dai-Liao Projection Method for Convex Constrained Nonlinear Monotone Equations with Applications

Auwal Bala Abubakar[†,§], **Poom Kumam**[†,‡1], **Aliyu Muhammed Awwal** [†,¶],

[†]KMUTTFixed Point Research Laboratory, Department of Mathematics, Room SCL 802
Fixed Point Laboratory, Science Laboratory Building, Faculty of Science,
King Mongkut's University of Technology Thonburi (KMUTT),
126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand.
[‡] KMUTT-Fixed Point Theory and Applications Research Group, Theoretical and
Computational Science Center (TaCS),Science Laboratory Building, Faculty of Science,
King Mongkut's University of Technology Thonburi (KMUTT),
126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand.
e-mail : poom.kumam@mail.kmutt.ac.th (P. Kumam)
[§]Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University,
Kano, Nigeria. e-mail : ababubakar.mth@buk.edu.ng (A.B. Abubakar)
[¶]Department of Mathematics, Faculty of Science, Gombe State University, Gombe, Nigeria.
e-mail : aliyumagsu@gmail.com (A.M. Awwal)

Abstract : In this article, we propose a descent Dai-Liao projection method to solve nonlinear monotone equations with convex constraints. Under some suitable conditions, we prove the global convergence of the method. Numerical experiments presented show that the method is efficient and promising compared to existing method for solving monotone nonlinear equations. Finally, the proposed method was applied to solve the sparse signal reconstruction problem in compressive sensing.

Keywords : Non-linear equations; Monotone equation; Projection method; Compressive sensing
2010 Mathematics Subject Classification : 90C30; 90C06; 90C56

---

[1]Corresponding author email: poom.kumam@mail.kmutt.ac.th (Poom Kumam)

1 Introduction

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

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

where $\mathbf{F} : \Omega \to \mathbb{R}^n$ is continuous, $\Omega \subset \mathbb{R}^n$ 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 [2, 7, 8, 10, 12, 18–20, 22, 26, 28, 29, 33, 38, 39]). With a good initial guess, these algorithms are very attractive as they have fast convergence rate. However, there are relatively few work on constrained problem (1.1).

Constrained problem (1.1) has applications in chemical equilibrium systems and economic equilibrium problems (see [11, 27]). Iterative methods for solving constrained monotone nonlinear equations have recently received relatively high attention. For example, in [43] a multivariate spectral projected gradient method (MSGP) was proposed. The method is an extension of the multivariate gradient method in [17] to bound constrained optimization. Numerical comparison of the method with the classical spectral gradient (SPG) method shows the efficiency of the method. Xiao and Zhu [41] proposed a projected conjugate gradient (CGD) to solve constrained problem of the form (1.1). This method can be viewed as an extension of the CG$-$Descent method for solving convex constrained problems. An extension of the conjugate gradient that belongs to the two unified frameworks to solve the constrained monotone equations was presented in [25]. Numerical experiments were also given to test the efficiency of the methods. Also in [34], three derivative-free projection methods for solving nonlinear equations with convex constraints were presented. These methods can be regarded as a combination of some recently developed conjugate gradient methods and the well-known projection technique. However, Sun and Liu [35] proposed a new hybrid conjugate gradient projection method for convex constrained equations. The method was based on the two famous Hestenes-Stiefel and Dai-Yuan conjugate gradient methods. Two new supermemory gradient methods for solving nonlinear monotone equations with convex constraints were proposed by Ou and Liu in [30]. Furthermore, Liu and Li [24] presented a projection method to solve monotone nonlinear equations with convex constraints. This method is a modification of the method in [41].

Motivated by these methods, we propose a descent Dai-Liao conjugate gradient projection method for constrained nonlinear monotone equations, which is an extension of the method in [1]. The global convergence was established under suitable conditions. Numerical examples were also presented to show the efficiency of the method proposed.

The remaining part of the paper is organized as follows. Section 2 will summarize some basic concepts and related properties which will be used in subsequent sections. Section 3 provides the proposed method and its algorithm. The global convergence of the algorithm is established in Section 4 and the last section will present some numerical results to show its practical performance, and apply it to solve the sparse signal reconstruction in compressive sensing.

2 Preliminaries

Let Ω be a nonempty, closed and convex subset of $\mathbb{R}^n$. Then for any $x \in \mathbb{R}^n$, 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 [42, 44] 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 are helpful 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 $x, y \in \mathbb{R}^n$.

(iii) F is uniformly monotone, that is,

$$\langle F(x) - F(y), x-y \rangle \geq c\|x-y\|^2, \quad c > 0, \tag{2.3}$$

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

3 Proposed Algorithm

Let x_0 be an initial point. An iterative scheme for solving (1.1) has the general form

$$x_{k+1} = x_k + s_k, \quad k = 0, 1, 2, \cdots, \tag{3.1}$$

where $s_k = \alpha_k d_k$, α_k is the step length obtained via some suitable line search and d_k is the search direction.

In this section, we propose an extension of the descent Dai-Liao type method in [1] to solve convex constrained nonlinear equations of the form (1.1). The direction is defined as

$$d_k = \begin{cases} -F(x_k), & \text{if } k = 0, \\ -F(x_k) + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{3.2}$$

where β_k is the descent Dai-Liao conjugate gradient parameter given as

$$\beta_k = \frac{F(x_k)^T y_{k-1}}{y_{k-1}^T d_{k-1}} - t_k \frac{F(x_k)^T s_{k-1}}{y_{k-1}^T d_{k-1}}, \quad t_k = p\frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} - q\frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}, \quad p \geq \frac{1}{4}, \quad q \leq 0,$$

$$y_k = F(x_{k+1}) - F(x_k), \quad s_k = z_k - x_k = \alpha_k d_k.$$

Notice that the CG parameter in (3.2) is a generalization of the two well known CG parameters for solving nonlinear equations. If p=2, q=0, then β_k will reduce to that of Hager and Zhang [15]. Also if p=1, q=0, β_k reduce to that of Dai and Kou [9].

Now all is set to describe our proposed algorithm, which is an extension of the method in [1] to solve convex constrained problems.

Algorithm 3.1. ***Descent Dai-Liao projection method (DLP)***

Step 0. *Given* $x_0 \in \Omega$, ρ, $\sigma \in (0,1)$, *stopping tolerance* $\varepsilon > 0$, *Set* $k = 0$.

Step 1. *Compute* $F(x_k)$. *If* $\|F(x_k)\| \leq \varepsilon$ *stop, else go to* ***Step 2***.

Step 2. *Compute* d_k *by (3.2). Stop if* $d_k = 0$.

Step 3. *Compute* $z_k = x_k + \alpha_k d_k$, *where* $\alpha_k = \rho^{m_k}$ *with* m_k *being the smallest nonnegative integer m such that*

$$-F(x_k + \rho^m d_k)^T d_k \geq \sigma \rho^m \|F(x_k + \rho^m d_k)\| \|d_k\|^2. \tag{3.3}$$

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

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

where

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

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

4 Global convergence analysis

To prove the global convergence of **Algorithm** 3.1, the following preliminaries are needed.

Lemma 4.1. *[3] Let d_k be defined by (3.2), then*

$$F(x_k)^T d_k = -\lambda_k \|F(x_k)\|^2, \quad \lambda_k > 0, \quad \forall k \in \mathbb{N}. \tag{4.1}$$

Lemma 4.2. *Let $\{x_k\}$ be generated by **Algorithm** 3.1, then*

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

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

$$\langle F(z_k), x_k - x_* \rangle \geq \langle F(z_k), x_k - z_k \rangle. \tag{4.2}$$

Also by definition of z_k and the line search (3.3), we have

$$\langle F(z_k), x_k - z_k \rangle \geq \sigma \alpha_k \|F(z_k)\| \|d_k\|^2 \geq 0. \tag{4.3}$$

So, we have

$$\begin{aligned} \|x_{k+1} - x_*\|^2 &= \|P_\Omega[x_k - \zeta_k F(z_k)] - P_\Omega(x_*)\|^2 \\ &\leq \|x_k - \zeta_k F(z_k) - x_*\|^2 \\ &\leq \|x_k - x_*\|^2 - 2\zeta_k \langle F(z_k), x_k - z_k \rangle + \|\zeta_k F(z_k)\|^2 \\ &= \|x_k - x_*\|^2 - \frac{\langle F(z_k), x_k - z_k \rangle^2}{\|F(z_k)\|^2}. \end{aligned} \tag{4.4}$$

Thus the sequence $\{\|x_k - x_*\|\}$ is non increasing and convergent, and hence $\{x_k\}$ is bounded.
From (4.4), we get

$$\|x_{k+1} - x_*\| \leq \|x_k - x_*\|.$$

Using the above inequality recursively, we have

$$\|x_k - x_*\| \leq \|x_0 - x_*\|, \quad \forall k \geq 0.$$

Therefore by (2.2)

$$\|F(x_k)\| = \|F(x_k) - F(x^*)\| \leq L\|x_k - x_*\| \leq L\|x_0 - x_*\|.$$

Letting $\tau = L\|x_0 - x_*\|$, then

$$\|F(x_k)\| \leq \tau \quad \forall k \geq 0. \tag{4.5}$$

Also from the definition of z_k, monotonicity of F and the Cauchy-Schwatz inequality, we obtain

$$\sigma\|x_k - z_k\| = \frac{\sigma\|\alpha_k d_k\|^2}{\|x_k - z_k\|} \leq \frac{\langle F(z_k), x_k - z_k\rangle}{\|x_k - z_k\|} \leq \frac{\langle F(x_k), x_k - z_k\rangle}{\|x_k - z_k\|} \leq \|F(x_k)\|. \tag{4.6}$$

The boundedness of the sequences $\{x_k\}$ and $\{F(x_k)\}$ together with equation (4.6), implies the sequence $\{z_k\}$ is bounded.

From the boundedness of $\{z_k\}$, for any $x^* \in \Omega^{'}$, where $\Omega^{'}$ is the solution set of (1.1), the sequence $\{z_k - x^*\}$ is also bounded, that is, there exists a positive constant $\nu > 0$ such that

$$\|z_k - x^*\| \leq \nu.$$

This together with the Lipschitz continuity of F, we have

$$\|F(z_k)\| = \|F(z_k) - F(\bar{x})\| \leq L\|z_k - \bar{x}\| \leq L\nu.$$

Therefore, using equation (4.4), we have

$$\frac{\sigma^2}{(L\nu)^2}\sum_{k=0}^{\infty}\|x_k - z_k\|^4 \leq \sum_{k=0}^{\infty}(\|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2) \leq \|X_0 - \bar{x}\| < \infty. \tag{4.7}$$

Equation (4.7) implies

$$\lim_{k\to\infty}\|x_k - z_k\| = 0. \tag{4.8}$$

However, equation (2.1), the definition of ζ_k and the Cauchy-Schwatz inequality, we have

$$\begin{aligned} \|x_{k+1} - x_k\| &= \|P_\Omega[x_k - \zeta_k F(z_k)] - x_k\| \\ &\leq \|x_k - \zeta_k F(z_k) - x_k\| \\ &= \|\zeta_k F(z_k)\| \\ &= \|x_k - z_k\|, \end{aligned} \tag{4.9}$$

which yields

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

Thus, by equation (4.8) and definition of z_k, then

$$\lim_{k\to\infty}\alpha_k\|d_k\| = 0. \tag{4.10}$$

□

Lemma 4.3. *[1] Suppose **Assumption A** (ii) hold and the sequence $\{x_k\}$ be generated by **Algorithm** 3.1 is bounded. Then there exist $M > 0$ such that $\|d_k\| \leq M$, $\forall k$.*

The following theorem establish the global convergence of **Algorithm** 3.1.

Theorem 4.4. *Let $\{x_k\}$ and $\{z_k\}$ be sequences generated by **Algorithm** 3.1. Then*

$$\liminf_{k\to\infty} \|F(x_k)\| = 0. \tag{4.11}$$

Proof. The proof will be divided into two cases;

Case I

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 $\tilde{x}$ such that $F(\tilde{x}) = 0$. Since $\{\|x_k - \tilde{x}\|\}$ converges and $\tilde{x}$ is an accumulation point of $\{x_k\}$, it follows that $\{x_k\}$ converges to $\tilde{x}$.

Case II

If $\liminf_{k\to\infty} \|d_k\| > 0$, we have

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

By (4.10), it holds that

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

Using the line search (3.3),

$$-F(x_k + \rho^{m_{k-1}} d_k)^T d_k < \sigma \rho^{m_{k-1}} \|F(x_k + \rho^{m_{k-1}} d_k)\| \|d_k\|^2,$$

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

$$-\langle F(\tilde{x}), \tilde{d}\rangle \leq 0. \tag{4.12}$$

On the other hand, from (4.1) we have

$$-\langle F(\tilde{x}), \tilde{d}\rangle = \lambda_k \|F(\tilde{x})\|^2 > 0. \tag{4.13}$$

(4.12) and (4.13) indicates a contradiction. Therefore, $\liminf_{k\to\infty} \|F(x_k)\| > 0$ does not hold and the proof is complete. □

5 Numerical Experiment

In this section, for convenience sake, we denote **Algorithm** 3.1 by DLP method. We also divided this section into two. First we compare DLP method with PCG method [24] by solving some monotone nonlinear equations with convex constraints using different initial points and several dimensions. Secondly, the DLP 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.

5.1 Experiment on some convex constrained nonlinear monotone equations

DLP and PCG methods have different line search implementation. The specific parameters for each method are as follows:

DLP method: $\rho = 0.6, \quad \sigma = 0.01, \quad p = 0.8 \quad$ and $q = -0.1$

PCG method: $\rho = 0.55, \quad r = 0.1, \quad \sigma = 0.0001.$

All runs were stopped whenever

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

We test problems 1 to 6 with dimensions of $n = 1000, \quad 5000, \quad 10000, \quad 50000, \quad 100000$ and different initial points: $x_1 = (1,1,\cdots,1)^T$, $x_2 = (2,2,\cdots,2)^T$, $x_3 = (3,3,\cdots,3)^T$, $x_4 = (5,5,\cdots,5)^T$, $x_5 = (8,8,\cdots,8)^T$, $x_6 = (0.5,0.5,\cdots,0.5)^T$, $x_7 = (0.1,0.1,\cdots,0.1)^T$, $x_8 = (10,10,\cdots,10)^T$. 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 tested problems $F(x) = (f_1(x), f_2(x), \cdots, f_n(x))^T$, where $x = (x_1, x_2, \cdots, x_n)^T$, are listed as follows:

Problem 1 Logarithmic Function

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

Problem 2 [45]

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

Problem 3 Strictly convex function [37]

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

Problem 4 [23]

$$F_i(x) = \min\left(\min(|x_i|, x_i^2), \max(|x_i|, x_i^3)\right), \text{ for } i = 1,2,\cdots,n \quad \text{and} \quad \Omega = \mathbb{R}^n_+.$$

Problem 5 Linear monotone problem

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

Problem 6 Tridiagonal Exponential Problem [5]

$$F_1(x) = x_1 - e^{\cos(h(x_1+x_2))}$$
$$F_i(x) = x_i - e^{\cos(h(x_{i-1}+x_i+x_{i+1}))} \text{ for } i = 2,3,\cdots,n-1$$
$$F_n(x) = x_n - e^{\cos(h(x_{n-1}+x_n))}$$

where $h = \dfrac{1}{n+1}$,

and $\Omega = \mathbb{R}^n_+$.

The results of the numerical performance in Table 1-6 indicate that the DLP method is more efficient and promising than the PCG method for the given test problems as it solves more problems with less itration than PCG method. It is worth mentioning that the PCG method fails to solve problem 3 completely while DLP was able to solve the problem. Thus, DLP method is an effective tool for solving large-scale nonlinear monotone equations with convex constraints.

Table 1: Numerical Results for DLP and PCG for Problem 1 with given initial points and dimensions

DIMENSION	INITIAL POINT	**DLP**				**PCG**			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	8	17	1.89492	2.97E-06	8	17	0.128661	5.25E-06
	x_2	9	19	0.041418	3.28E-06	9	19	0.007323	6.33E-06
	x_3	10	21	0.016162	2.03E-06	10	21	0.01382	4.22E-06
	x_4	11	23	0.010309	3.87E-06	11	23	0.007542	8.41E-06
	x_5	13	27	0.04376	1.46E-06	13	27	0.009152	3.41E-06
	x_6	7	15	0.008146	4.73E-06	7	15	0.005303	7.63E-06
	x_7	6	13	0.005442	1.62E-06	6	13	0.004304	2.35E-06
	x_8	14	29	0.015187	1.24E-06	14	29	0.01053	2.94E-06
5000	x_1	8	17	0.134513	6.33E-06	9	19	0.040941	1.2E-06
	x_2	9	19	0.027834	6.96E-06	10	21	0.031976	1.45E-06
	x_3	10	21	0.022161	4.29E-06	10	21	0.019165	8.92E-06
	x_4	11	23	0.034782	8.12E-06	12	25	0.020538	1.91E-06
	x_5	13	27	0.03063	3.03E-06	13	27	0.025987	7.1E-06
	x_6	8	17	0.022179	1.01E-06	8	17	0.017705	1.76E-06
	x_7	6	13	0.015895	3.44E-06	6	13	0.011297	5E-06
	x_8	14	29	0.040785	2.55E-06	14	29	0.025084	6.06E-06
10000	x_1	8	17	0.048993	8.89E-06	9	19	0.034986	1.69E-06
	x_2	9	19	0.046191	9.78E-06	10	21	0.041433	2.03E-06
	x_3	10	21	0.090236	6.02E-06	11	23	0.037036	1.35E-06
	x_4	12	25	0.052695	1.14E-06	12	25	0.056032	2.67E-06
	x_5	13	27	0.094989	4.24E-06	13	27	0.043289	9.95E-06
	x_6	8	17	0.037663	1.42E-06	8	17	0.028787	2.47E-06
	x_7	6	13	0.026615	4.83E-06	6	13	0.026157	7.02E-06
	x_8	14	29	0.075396	3.57E-06	14	29	0.056002	8.49E-06
50000	x_1	9	19	0.16338	1.98E-06	9	19	0.138459	3.76E-06
	x_2	10	21	0.164065	2.18E-06	10	21	0.140902	4.52E-06
	x_3	11	23	0.196741	1.34E-06	11	23	0.148143	2.99E-06
	x_4	12	25	0.202617	2.53E-06	12	25	0.161228	5.94E-06
	x_5	13	27	0.245118	9.41E-06	14	29	0.224978	2.37E-06
	x_6	8	17	0.135402	3.16E-06	8	17	0.124476	5.49E-06
	x_7	7	15	0.143212	1.07E-06	7	15	0.094523	1.68E-06
	x_8	14	29	0.236847	7.93E-06	15	31	0.200173	2.02E-06
100000	x_1	9	19	0.325882	2.8E-06	9	19	0.307435	5.32E-06
	x_2	10	21	0.385329	3.07E-06	10	21	0.268732	6.39E-06
	x_3	11	23	0.353255	1.89E-06	11	23	0.287227	4.23E-06
	x_4	12	25	0.402676	3.58E-06	12	25	0.329961	8.39E-06
	x_5	14	29	0.500612	1.33E-06	14	29	0.374643	3.35E-06
	x_6	8	17	0.304638	4.47E-06	8	17	0.220025	7.76E-06
	x_7	7	15	0.225243	1.52E-06	7	15	0.179665	2.37E-06
	x_8	15	31	0.509863	1.12E-06	15	31	0.401959	2.86E-06

Table 2: Numerical Results for DLP and PCG for Problem 2 with given initial points and dimensions

DIMENSION	INITIAL POINT	DLP				PCG			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	7	16	0.168292	2.3E-06	8	18	0.147909	1.77E-06
	x_2	6	15	0.005954	6.94E-06	8	18	0.009771	1.44E-06
	x_3	4	12	0.004452	8.01E-06	9	22	0.009044	1.95E-06
	x_4	7	19	0.012324	9.9E-06	9	22	0.00969	1.84E-06
	x_5	7	18	0.007935	8.47E-06	10	26	0.011794	1.88E-06
	x_6	7	16	0.005343	2.61E-06	8	18	0.006838	1.2E-06
	x_7	6	14	0.00599	6.28E-06	7	16	0.005667	2.51E-06
	x_8	7	20	0.00654	7.09E-06	10	26	0.011761	1.86E-06
5000	x_1	7	16	0.022532	5.13E-06	8	18	0.014723	3.95E-06
	x_2	7	17	0.018983	1.55E-06	8	18	0.024142	3.22E-06
	x_3	5	14	0.035157	1.79E-06	9	22	0.026478	4.37E-06
	x_4	8	21	0.026948	2.21E-06	9	22	0.027124	4.12E-06
	x_5	8	20	0.017493	1.89E-06	10	26	0.021357	4.2E-06
	x_6	7	16	0.016511	5.84E-06	8	18	0.016335	2.69E-06
	x_7	7	16	0.015694	1.4E-06	7	16	0.03198	5.61E-06
	x_8	8	22	0.021744	1.59E-06	10	26	0.03207	4.16E-06
10000	x_1	7	16	0.026521	7.26E-06	8	18	0.023789	5.59E-06
	x_2	7	17	0.033793	2.19E-06	8	18	0.040595	4.55E-06
	x_3	5	14	0.039069	2.53E-06	9	22	0.048168	6.17E-06
	x_4	8	21	0.041747	3.13E-06	9	22	0.041176	5.83E-06
	x_5	8	20	0.03451	2.68E-06	10	26	0.055611	5.95E-06
	x_6	7	16	0.032251	8.27E-06	8	18	0.061043	3.81E-06
	x_7	7	16	0.029218	1.99E-06	7	16	0.048982	7.94E-06
	x_8	8	22	0.038164	2.24E-06	10	26	0.03201	5.89E-06
50000	x_1	8	18	0.132832	1.62E-06	9	20	0.130505	1.34E-06
	x_2	7	17	0.114783	4.9E-06	9	20	0.12844	1.09E-06
	x_3	5	14	0.082067	5.67E-06	10	24	0.146792	1.48E-06
	x_4	8	21	0.144584	7E-06	10	24	0.129886	1.4E-06
	x_5	8	20	0.142008	5.99E-06	11	28	0.146793	1.43E-06
	x_6	8	18	0.111679	1.85E-06	8	18	0.105336	8.52E-06
	x_7	7	16	0.095846	4.44E-06	8	18	0.084007	1.91E-06
	x_8	8	22	0.14324	5.02E-06	11	28	0.145255	1.41E-06
100000	x_1	8	18	0.232209	2.3E-06	9	20	0.19899	1.9E-06
	x_2	7	17	0.223408	6.94E-06	9	20	0.220281	1.55E-06
	x_3	5	14	0.18994	8.01E-06	10	24	0.239587	2.1E-06
	x_4	8	21	0.265192	9.9E-06	10	24	0.247644	1.98E-06
	x_5	8	20	0.275303	8.47E-06	11	28	0.293617	2.02E-06
	x_6	8	18	0.218477	2.61E-06	9	20	0.229052	1.29E-06
	x_7	7	16	0.197266	6.28E-06	8	18	0.199297	2.7E-06
	x_8	8	22	0.28692	7.09E-06	11	28	0.271278	2E-06

Table 3: Numerical Results for DLP and PCG for Problem 3 with given initial points and dimensions

DIMENSION	INITIAL POINT	DLP				PCG			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	1	3	0.169863	0	-	-	-	-
	x_2	1	3	0.001969	0	-	-	-	-
	x_3	1	3	0.003447	0	-	-	-	-
	x_4	1	3	0.002714	0	-	-	-	-
	x_5	1	3	0.002207	0	-	-	-	-
	x_6	-	-	-	-	-	-	-	-
	x_7	-	-	-	-	-	-	-	-
	x_8	1	3	0.001646	0	-	-	-	-
5000	x_1	1	3	0.004619	0	-	-	-	-
	x_2	1	3	0.004941	0	-	-	-	-
	x_3	1	3	0.005255	0	-	-	-	-
	x_4	1	3	0.010039	0	-	-	-	-
	x_5	1	3	0.005467	0	-	-	-	-
	x_6	-	-	-	-	-	-	-	-
	x_7	-	-	-	-	-	-	-	-
	x_8	1	3	0.005478	0	-	-	-	-
10000	x_1	1	3	0.009167	0	-	-	-	-
	x_2	1	3	0.007543	0	-	-	-	-
	x_3	1	3	0.010398	0	-	-	-	-
	x_4	1	3	0.011695	0	-	-	-	-
	x_5	1	3	0.013835	0	-	-	-	-
	x_6	-	-	-	-	-	-	-	-
	x_7	-	-	-	-	-	-	-	-
	x_8	1	3	0.008795	0	-	-	-	-
50000	x_1	1	3	0.028816	0	-	-	-	-
	x_2	1	3	0.044788	0	-	-	-	-
	x_3	1	3	0.040274	0	-	-	-	-
	x_4	1	3	0.058309	0	-	-	-	-
	x_5	1	3	0.041435	0	-	-	-	-
	x_6	-	-	-	-	-	-	-	-
	x_7	-	-	-	-	-	-	-	-
	x_8	1	3	0.035683	0	-	-	-	-
100000	x_1	1	3	0.058245	0	-	-	-	-
	x_2	1	3	0.074857	0	-	-	-	-
	x_3	1	3	0.075574	0	-	-	-	-
	x_4	1	3	0.096167	0	-	-	-	-
	x_5	1	3	0.072731	0	-	-	-	-
	x_6	-	-	-	-	-	-	-	-
	x_7	-	-	-	-	-	-	-	-
	x_8	1	3	0.065024	0	-	-	-	-

Table 4: Numerical Results for DLP and PCG for Problem 4 with given initial points and dimensions

DIMENSION	INITIAL POINT	DLP				PCG			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	7	18	0.011634	1.62E-06	7	16	0.120907	2.05E-06
	x_2	8	24	0.009923	1.17E-06	7	17	0.006656	2.34E-06
	x_3	8	27	0.008514	1.06E-06	8	22	0.004618	1.42E-06
	x_4	1	14	0.005969	0	8	24	0.007953	1.84E-06
	x_5	1	14	0.004946	0	8	28	0.005406	2.73E-06
	x_6	7	17	0.008406	1.59E-06	7	16	0.00529	3.12E-06
	x_7	6	14	0.005229	4.62E-06	7	16	0.008888	1.93E-06
	x_8	1	14	0.003984	0	1	14	0.00308	0
5000	x_1	7	18	0.014825	3.61E-06	7	16	0.01506	4.59E-06
	x_2	8	24	0.016148	2.62E-06	7	17	0.016877	5.22E-06
	x_3	8	27	0.045933	2.37E-06	8	22	0.022512	3.17E-06
	x_4	1	14	0.013368	0	8	24	0.017731	4.12E-06
	x_5	1	14	0.011936	0	8	28	0.019307	6.11E-06
	x_6	7	17	0.014228	3.55E-06	7	16	0.010712	6.99E-06
	x_7	7	16	0.018813	1.03E-06	7	16	0.011305	4.31E-06
	x_8	1	14	0.013579	0	1	14	0.007806	0
10000	x_1	7	18	0.025476	5.11E-06	7	16	0.020382	6.49E-06
	x_2	8	24	0.040854	3.7E-06	7	17	0.024809	7.38E-06
	x_3	8	27	0.040338	3.35E-06	8	22	0.025178	4.48E-06
	x_4	1	14	0.019809	0	8	24	0.028127	5.82E-06
	x_5	1	14	0.020869	0	8	28	0.029626	8.64E-06
	x_6	7	17	0.024477	5.02E-06	7	16	0.019614	9.88E-06
	x_7	7	16	0.031616	1.46E-06	7	16	0.021254	6.1E-06
	x_8	1	14	0.016972	0	1	14	0.011905	0
50000	x_1	8	20	0.118336	1.14E-06	8	18	0.084597	1.56E-06
	x_2	8	24	0.126378	8.28E-06	8	19	0.085577	1.77E-06
	x_3	8	27	0.128701	7.49E-06	9	24	0.105039	1.08E-06
	x_4	1	14	0.086889	0	9	26	0.107482	1.4E-06
	x_5	1	14	0.065903	0	9	30	0.110678	2.08E-06
	x_6	8	19	0.090905	1.12E-06	8	18	0.082901	2.37E-06
	x_7	7	16	0.07588	3.27E-06	8	18	0.073976	1.47E-06
	x_8	1	14	0.058512	0	1	14	0.039872	0
100000	x_1	8	20	0.199397	1.62E-06	8	18	0.150982	2.2E-06
	x_2	9	26	0.249709	1.17E-06	8	19	0.149197	2.51E-06
	x_3	9	29	0.266483	1.06E-06	9	24	0.188938	1.52E-06
	x_4	1	14	0.140324	0	9	26	0.237588	1.98E-06
	x_5	1	14	0.107957	0	9	30	0.205939	2.93E-06
	x_6	8	19	0.192051	1.59E-06	8	18	0.141674	3.36E-06
	x_7	7	16	0.144332	4.62E-06	8	18	0.141241	2.07E-06
	x_8	1	14	0.112528	0	1	14	0.068503	0

Table 5: Numerical Results for DLP and PCG for Problem 5 with given initial points and dimensions

DIMENSION	INITIAL POINT	DLP				PCG			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	14	127	0.105925	5.57E-06	63	290	0.118814	8.48E-06
	x_2	15	136	0.026983	3.82E-06	52	241	0.036904	8.28E-06
	x_3	15	136	0.067848	4.2E-06	67	308	0.040847	8.16E-06
	x_4	15	136	0.054061	6.62E-06	75	344	0.041852	9.58E-06
	x_5	15	136	0.075227	9E-06	82	375	0.046013	9.96E-06
	x_6	15	139	0.073398	8.87E-06	69	316	0.038412	9.78E-06
	x_7	15	133	0.027191	6.25E-06	75	342	0.043601	8.82E-06
	x_8	16	145	0.038284	4.88E-06	86	393	0.045662	8.73E-06
5000	x_1	14	127	0.089164	7.38E-06	62	286	0.113742	8.19E-06
	x_2	14	127	0.105737	8.54E-06	49	228	0.102778	9.94E-06
	x_3	15	136	0.092498	4.77E-06	64	295	0.150502	9.67E-06
	x_4	15	136	0.085904	7.25E-06	74	340	0.133203	9.18E-06
	x_5	16	145	0.085381	4.29E-06	81	371	0.161908	9.59E-06
	x_6	13	118	0.079126	9.9E-06	68	312	0.14433	9.42E-06
	x_7	17	154	0.101356	4.55E-06	74	338	0.152105	8.47E-06
	x_8	16	145	0.146068	5.28E-06	85	389	0.161277	8.4E-06
10000	x_1	14	127	0.180379	7.53E-06	60	277	0.242601	9.78E-06
	x_2	14	127	0.163077	9.11E-06	49	228	0.200631	9.7E-06
	x_3	15	136	0.174075	5.07E-06	64	295	0.252515	9.57E-06
	x_4	15	136	0.170753	7.82E-06	74	340	0.29364	8.93E-06
	x_5	15	136	0.174473	9.91E-06	80	367	0.318742	9.39E-06
	x_6	14	127	0.169452	4.51E-06	68	312	0.273804	9.15E-06
	x_7	17	155	0.201102	7.8E-06	73	334	0.297999	8.45E-06
	x_8	16	145	0.179844	5.5E-06	84	385	0.331882	8.26E-06
50000	x_1	14	127	0.652822	7.22E-06	59	273	0.986398	9.44E-06
	x_2	15	136	0.679862	4.67E-06	48	224	0.797601	9.22E-06
	x_3	15	136	0.689531	6.22E-06	62	287	1.034818	9.23E-06
	x_4	15	136	0.686255	8.28E-06	73	336	1.211206	8.58E-06
	x_5	16	145	0.80502	5.27E-06	79	363	1.308637	9.03E-06
	x_6	14	127	0.668592	5.04E-06	67	308	1.130524	8.82E-06
	x_7	16	143	0.747463	4.23E-06	70	321	1.14032	9.99E-06
	x_8	16	145	0.75258	5.72E-06	81	372	1.393427	9.76E-06
100000	x_1	14	127	1.472278	7.76E-06	59	273	2.176673	9.33E-06
	x_2	15	136	1.514384	5.14E-06	47	220	1.751684	8.99E-06
	x_3	15	136	1.54378	6.18E-06	62	287	2.309171	8.95E-06
	x_4	15	136	1.573957	7.89E-06	72	332	2.732557	8.55E-06
	x_5	16	145	1.649696	4.71E-06	79	363	2.944461	8.9E-06
	x_6	14	127	1.45179	4.76E-06	67	308	2.503094	8.71E-06
	x_7	17	151	1.768302	6.89E-06	70	321	2.65583	9.7E-06
	x_8	16	145	1.680112	5.63E-06	81	372	3.043041	9.59E-06

Table 6: Numerical Results for DLP and PCG for Problem 6 with given initial points and dimensions

DIMENSION	INITIAL POINT	DLP				PCG			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	x_1	12	27	0.140661	7.91E-06	9	21	0.088999	7.74E-06
	x_2	12	27	0.014993	3.76E-06	9	21	0.007196	3.54E-06
	x_3	10	22	0.010194	9.48E-06	9	21	0.009945	1.51E-06
	x_4	13	29	0.01804	1.69E-06	10	23	0.007655	5.3E-06
	x_5	13	29	0.017393	5.4E-06	11	26	0.008004	8.57E-06
	x_6	12	27	0.015558	9.51E-06	9	21	0.010214	9.54E-06
	x_7	13	29	0.017285	1.05E-06	10	23	0.008567	1.79E-06
	x_8	13	29	0.031666	9.1E-06	12	28	0.012799	2.19E-06
5000	x_1	8	18	0.032769	6.16E-06	9	20	0.028147	4.33E-06
	x_2	8	18	0.038725	2.99E-06	8	18	0.029038	4.87E-06
	x_3	7	16	0.028004	4.23E-06	8	18	0.030951	1.95E-06
	x_4	10	22	0.051045	6.07E-06	9	20	0.026379	7.94E-06
	x_5	12	26	0.043347	9.28E-06	10	23	0.022638	9.43E-06
	x_6	8	18	0.034827	7.34E-06	9	20	0.028056	5.33E-06
	x_7	8	18	0.03225	8.08E-06	9	20	0.027591	6.05E-06
	x_8	15	33	0.073159	2.03E-06	11	25	0.024584	2.21E-06
10000	x_1	8	18	0.057972	3.72E-06	9	20	0.047859	1.9E-06
	x_2	8	18	0.055588	1.6E-06	8	18	0.043765	6.18E-06
	x_3	7	16	0.046981	5.65E-06	8	18	0.041857	2.42E-06
	x_4	8	18	0.051349	5.62E-06	9	20	0.047152	2.87E-06
	x_5	9	20	0.064618	8.93E-06	9	20	0.048928	7.34E-06
	x_6	8	18	0.053097	4.74E-06	9	20	0.043221	2.41E-06
	x_7	8	18	0.05044	5.54E-06	9	20	0.043262	2.81E-06
	x_8	10	22	0.060508	9.28E-06	10	22	0.049545	7.42E-06
50000	x_1	8	18	0.209071	7.68E-06	9	20	0.179764	3.54E-06
	x_2	8	18	0.193629	3.21E-06	9	20	0.200668	1.48E-06
	x_3	8	18	0.19544	1.26E-06	8	18	0.152417	5.4E-06
	x_4	9	20	0.217707	1.03E-06	9	20	0.173361	4.7E-06
	x_5	9	20	0.218818	2.39E-06	10	22	0.189286	1.18E-06
	x_6	8	18	0.213434	9.92E-06	9	20	0.178929	4.56E-06
	x_7	9	20	0.221013	1.17E-06	9	20	0.172644	5.39E-06
	x_8	9	20	0.226795	3.31E-06	10	22	0.190464	1.64E-06
100000	x_1	9	20	0.469107	1.09E-06	9	20	0.35486	5E-06
	x_2	8	18	0.400465	4.54E-06	9	20	0.373785	2.09E-06
	x_3	8	18	0.401195	1.78E-06	8	18	0.332346	7.63E-06
	x_4	9	20	0.446933	1.44E-06	9	20	0.360027	6.64E-06
	x_5	9	20	0.456224	3.34E-06	10	22	0.402609	1.65E-06
	x_6	9	20	0.51475	1.4E-06	9	20	0.360589	6.45E-06
	x_7	9	20	0.450645	1.66E-06	9	20	0.357489	7.62E-06
	x_8	9	20	0.467588	4.61E-06	10	22	0.402096	2.28E-06

5.2 Application in compressive sensing

Many problems in signal processing that involves finding sparse solutions to ill-conditioned linear systems of equations. One of the popular approach is to minimize an objective function containing quadratic (ℓ_2) error term and a sparse ℓ_1-regularization term, i.e.,

$$\min_x \omega\|x\|_1 + \frac{1}{2}\|y - Ax\|_2^2, \tag{5.1}$$

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, $\omega > 0$, $\|x\|_2$ denotes the Euclidean norm of x and $\|x\|_1 = \sum_{i=1}^n |x_i|$ is the ℓ_1-norm of x. It is not difficult to observe that problem (5.1) is a convex unconstrained minimization problem. The fact that if the original signal is sparse or approximately sparse in some orthogonal basis, problem (5.1) usually appears in compressive sensing, and hence an exact recovery can be produced by solving (5.1).

Many iterative methods for solving (5.1) have been been proposed in several literatures, (see [4, 6, 13, 14, 16, 21, 31, 36]). gradient based methods happens to be the most popular and the earliest gradient projection method for sparse reconstruction (GPRS) was proposed by Figueiredo et al. [14]. Step one of the GPRS method is to express (5.1) as a quadratic problem using the following approach. Let $x \in \mathbb{R}^n$, by splitting x into its positive and negative parts. Then it 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 $i = 1, 2, \cdots, n$, and $(\cdot)_+ = \max\{0, \cdot\}$. By definition of ℓ_1-norm, we have $\|x\|_1 = e_n^T u + e_n^T v$, where $e_n = (1, 1, \cdots, 1)^T \in \mathbb{R}^n$. Now (5.1) can be written as

$$\min_{u,v} \frac{1}{2}\|y - A(u-v)\|_2^2 + \omega e_n^T u + \omega e_n^T v, \qquad u \geq 0, \quad v \geq 0, \tag{5.2}$$

which is a bound-constrained quadratic program. Furthermore, from [14], equation (5.2) can be written in standard form as

$$\min_z \frac{1}{2} z^T H z + c^T z, \qquad \text{such that} \quad z \geq 0, \tag{5.3}$$

where $z = \begin{pmatrix} u \\ v \end{pmatrix}$, $\quad c = \omega e_{2n} + \begin{pmatrix} -b \\ b \end{pmatrix}$, $\quad b = A^T y$, $\quad H = \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix}$.
Clearly, H is a positive semidefinite matrix, which implies that equation (5.3) is a convex quadratic problem.

Xiao et al. [41] explains that problem (5.3) 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:

$$F(z) = \min\{z, Hz + c\} = 0. \tag{5.4}$$

In [32, 40], it was proved that $F(z)$ is monotone and continuous . Therefore problem (5.1) can be translated into problem (1.1) and thus DLP method can be applied to solve (5.1).

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 $\tilde{x}$,

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

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, $y = A\tilde{x} + \mu$, where μ is the Gaussian noise distributed normally with mean 0 and variance 10^{-4} ($N(0, 10^{-4})$).

The performance of the DLP method in compressive sensing was shown in comparison with the PCG method. The parameters in choosen in DLP method are $\rho = 10$, $\sigma = 10^{-4}$ and that of PCG method are chosen as $\rho = 10$, $\sigma = 10^{-4}$ and $r = 0.5$. 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, DLP and PCG 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 DLP and PCG methods are having same number of iteration but DLP is relatively having less CPU time. This shows that the DLP method is competing with recent methods for solving signal recovery problems.

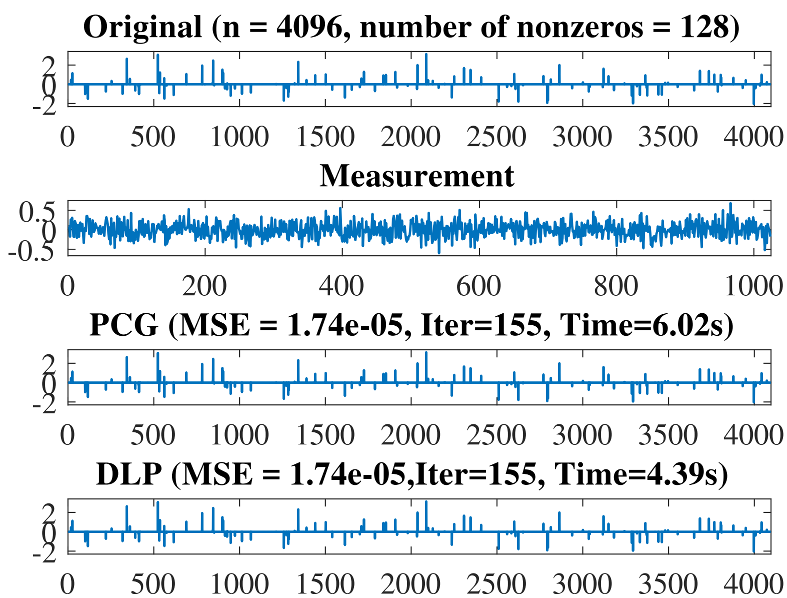

Figure 1: From top to bottom: the original image, the measurement, and the recovered signals by DLP and PCG methods.

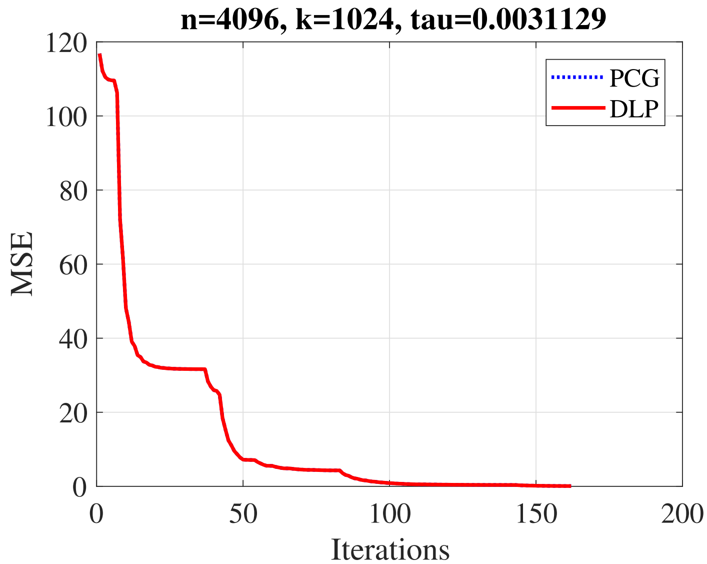

Figure 2: Iterations

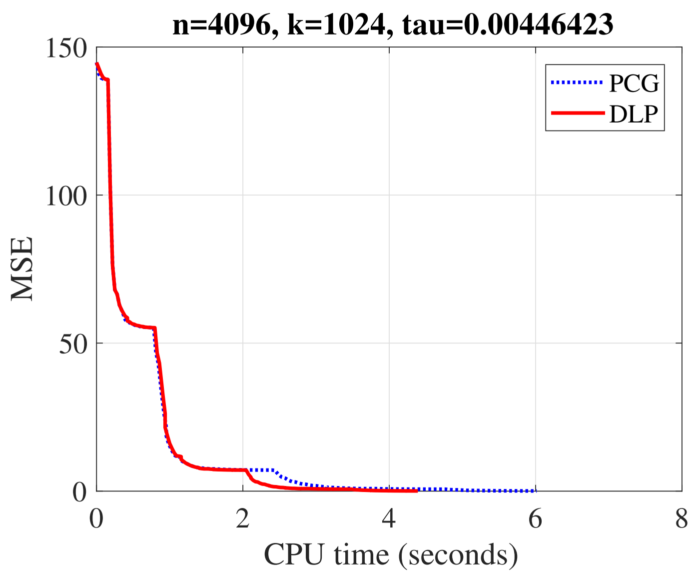

Figure 3: CPU time (seconds)

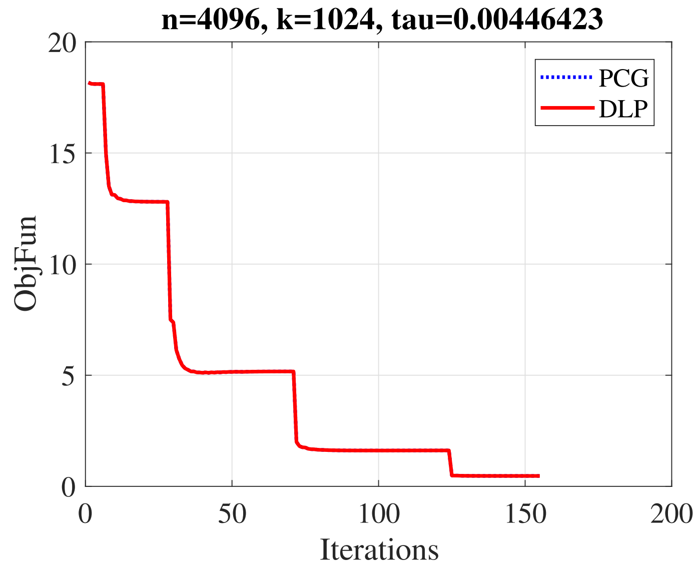

Figure 4: Iterations

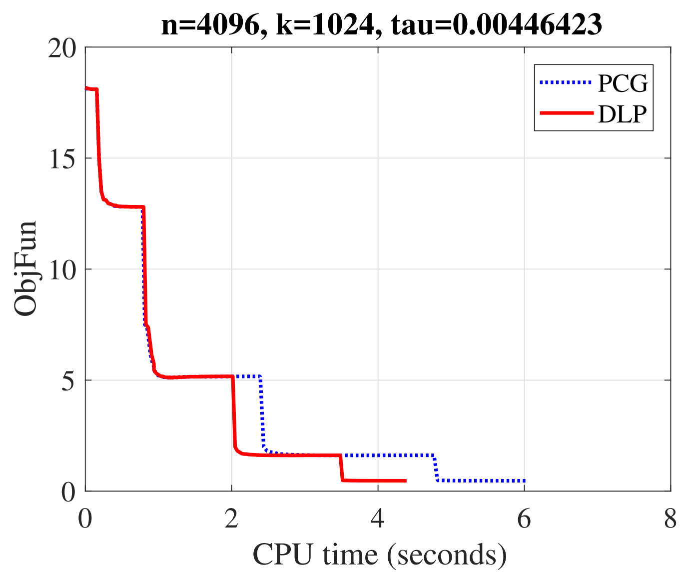

Figure 5: CPU time (seconds)

Table 7: Ten experiment results together with average result of ℓ_1-norm regularization problem for DLP and PCG methods

	DLP			PCG		
	MSE	ITER	CPU(s)	MSE	ITER	CPU(s)
	1.74E-05	155	4.39	1.74E-05	155	6.02
	2.55E-05	115	3.34	2.55E-05	115	3.73
	1.23E-05	143	3.88	1.23E-05	143	3.91
	5.52E-05	128	3.41	5.52E-05	128	3.41
	1.26E-05	125	3.50	1.26E-05	125	3.58
	3.86E-05	130	3.69	3.86E-05	130	3.70
	2.13E-05	161	4.56	2.13E-05	161	4.59
	1.43E-05	115	3.25	1.43E-05	115	3.39
	1.87E-05	139	3.89	1.87E-05	139	4.19
	1.58E-05	136	3.66	1.58E-05	136	3.75
Average	2.32E-05	134.7	3.757	2.32E-05	134.7	4.027

6 Conclusions

In this article, a descent Dai-Liao conjugate gradient method for solving nonlinear convex constraints monotone equations 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 appropriate conditions.

We can view the the proposed method as an extension of the method in [1] to solve convex constrained problems. Numerical results show that the proposed method is more efficient than the PCG method for the given constrained problems. Furthermore, the proposed method can be applied to solve ℓ_1-norm regularization problem in compressive sensing.

Acknowledgements: We would like to thank the referee for his comments and suggestions which provide an improvement on the manuscript. The authors acknowledge the financial support provided by King Mongkut's University of Technology Thonburi through the "KMUTT 55th Anniversary Commemorative Fund". The first author was supported by the Petchra Pra Jom Klao Doctoral Scholarship Academic for Ph.D. Program at KMUTT. This project was supported by the Theoretical and Computational Science (TaCS) Center under Computational and Applied Science for Smart Innovation Research Cluster (CLASSIC), Faculty of Science, KMUTT.

References

[1] Auwal Bala Abubakar and Poom Kumam. A descent Dai-Liao conjugate gradient method for nonlinear equations. *Numerical Algorithms*, pages 1–14, 2018.

[2] Mehiddin Al-Baali, Emilio Spedicato, and Francesca 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):937–954, 2014.

[3] S. babaie Kafaki and R. Gambari. A descent family of dai-liao conjugate gradient methods. *Optimization Methods and Software*, 29(3):583–591, 2013.

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

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

[6] Ernesto G Birgin, José Mario Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

[7] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math Comput*, 19:577–593, 1965.

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

[9] Yu-Hong Dai and Cai-Xia Kou. A nonlinear conjugate gradient algorithm with an optimal property and an improved wolfe line search. *SIAM Journal on Optimization*, 23(1):296–320, 2013.

[10] Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug. Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408, 1982.

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

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

[13] Mário AT Figueiredo and Robert D Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8):906–916, 2003.

[14] Mário AT Figueiredo, Robert D Nowak, and Stephen 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):586–597, 2007.

[15] William W Hager and Hongchao Zhang. Algorithm 851: Cg_descent, a conjugate gradient method with guaranteed descent. *ACM Transactions on Mathematical Software (TOMS)*, 32(1):113–137, 2006.

[16] Elaine T Hale, Wotao Yin, and Yin Zhang. A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, 43:44, 2007.

[17] Le Han, Gaohang Yu, and Lutai Guan. Multivariate spectral gradient method for unconstrained optimization. *Applied Mathematics and Computation*, 201(1-2):621–630, 2008.

[18] Mohammad Hassan and Mohammed Yusuf Waziri. On Broyden-like update via some quadratures for solving nonlinear systems of equations. *Turkish Journal of Mathematics*, 39(3):335–345, 2015.

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

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

[21] Duangkamon Kitkuan, Poom Kumam, Anantachai Padcharoen, Wiyada Kumam, and Phatiphat Thounthong. Algorithms for zeros of two accretive operators for solving convex minimization problems and its application to image restoration problems. *Journal of Computational and Applied Mathematics*, 2018.

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

[23] William La Cruz. A spectral algorithm for large-scale systems of nonlinear monotone equations. *Numerical Algorithms*, 76(4):1109–1130, 2017.

[24] JK Liu and SJ Li. A projection method for convex constrained monotone nonlinear equations with applications. *Computers & Mathematics with Applications*, 70(10):2442–2453, 2015.

[25] San-Yang Liu, Yuan-Yuan Huang, and Hong-Wei Jiao. Sufficient descent conjugate gradient methods for solving convex constrained nonlinear monotone equations. In *Abstract and Applied Analysis*, volume 2014. Hindawi, 2014.

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

[27] Keith Meintjes and Alexander P Morgan. A methodology for solving chemical equilibrium systems. *Applied Mathematics and Computation*, 22(4):333–361, 1987.

[28] J. J. Moré and J. A. Trangenstein. On the global convergence of Broyden's method. *Mathematics of Computation*, 30:523–540, 1976.

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

[30] Yigui Ou and Yuanwen Liu. Supermemory gradient methods for monotone nonlinear equations with convex constraints. *Computational and Applied Mathematics*, 36(1):259–279, 2017.

[31] Anantachai Padcharoen, Poom Kumam, and Juan Martnez-Moreno. Augmented lagrangian method for tv-l1-l2 based colour image restoration. *Journal of Computational and Applied Mathematics*, 2018.

[32] Jong-Shi Pang. Inexact Newton methods for the nonlinear complementarity problem. *Mathematical Programming*, 36(1):54–71, 1986.

[33] 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*, 217:3190–3198, 2010.

[34] Min Sun and Jing Liu. Three derivative-free projection methods for nonlinear equations with convex constraints. *Journal of Applied Mathematics and Computing*, 47(1-2):265–276, 2015.

[35] Min Sun and Jing Liu. New hybrid conjugate gradient projection method for the convex constrained equations. *Calcolo*, 53(3):399–411, 2016.

[36] Ewout Van Den Berg and Michael P Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[37] Chuanwei Wang, Yiju Wang, and Chuanliang Xu. A projection method for a system of nonlinear monotone equations with convex constraints. *Mathematical Methods of Operations Research*, 66(1):33–46, 2007.

[38] MY Waziri, WJ Leong, and MA Hassan. Diagonal Broyden-like method for large-scale systems of nonlinear equations. *Malaysian Journal of Mathematical Sciences*, 6(1):59–73, 2012.

[39] MY Waziri, WJ Leong, MA 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):246–252, 2010.

[40] Yunhai Xiao, Qiuyu Wang, and Qingjie Hu. Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing. *Nonlinear Analysis: Theory, Methods & Applications*, 74(11):3570–3577, 2011.

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

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

[43] Zhensheng Yu, Jing Sun, and Yi Qin. A multivariate spectral projected gradient method for bound constrained optimization. *Journal of computational and applied mathematics*, 235(8):2263–2269, 2011.

[44] Eduardo H Zarantonello. Projections on convex sets in hilbert space and spectral theory: Part i. projections on convex sets: Part ii. spectral theory. In *Contributions to nonlinear functional analysis*, pages 237–424. Elsevier, 1971.

[45] Weijun Zhou and Donghui Li. Limited memory BFGS method for nonlinear monotone equations. *Journal of Computational Mathematics*, 25(1):89–96, 2007.

(Received 11 August 2018)
(Accepted 4 December 2018)