

Article

# Least-Square-Based Three-Term Conjugate Gradient Projection Method for $\ell_1$ -Norm Problems with Application to Compressed Sensing

Abdulkarim Hassan Ibrahim <sup>1,2</sup> , Poom Kumam <sup>1,2,3,\*</sup> , Auwal Bala Abubakar <sup>1,4</sup> ,  
Jamilu Abubakar <sup>1,5</sup>  and Abubakar Bakoji Muhammad <sup>6</sup> 

<sup>1</sup> KMUTTFixed Point Research Laboratory, Room SCL 802 Fixed Point Laboratory, Science Laboratory Building, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand; ibrahimkarym@gmail.com (A.H.I.); ababubakar.mth@buk.edu.ng (A.B.A.); abubakar.jamilu@udusok.edu.ng (J.A.)

<sup>2</sup> Center of Excellence in Theoretical and Computational Science (TaCS-CoE), 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

<sup>3</sup> Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan

<sup>4</sup> Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano 700241, Nigeria

<sup>5</sup> Department of Mathematics, Usmanu Danfodiyo University, Sokoto 840004, Nigeria

<sup>6</sup> Faculty of Natural Sciences II, Institute of Mathematics, Martin Luther University Halle-Wittenberg, 06099 Halle, Germany; abubakar.muhammad@mathematik.uni-halle.de

\* Correspondence: poom.kum@kmutt.ac.th

Received: 27 February 2020; Accepted: 9 April 2020; Published: 15 April 2020



**Abstract:** In this paper, we propose, analyze, and test an alternative method for solving the  $\ell_1$ -norm regularization problem for recovering sparse signals and blurred images in compressive sensing. The method is motivated by the recent proposed nonlinear conjugate gradient method of Tang, Li and Cui [Journal of Inequalities and Applications, 2020(1), 27] designed based on the least-squares technique. The proposed method aims to minimize a non-smooth minimization problem consisting of a least-squares data fitting term and an  $\ell_1$ -norm regularization term. The search directions generated by the proposed method are descent directions. In addition, under the monotonicity and Lipschitz continuity assumption, we establish the global convergence of the method. Preliminary numerical results are reported to show the efficiency of the proposed method in practical computation.

**Keywords:** image processing; compressed sensing;  $\ell_1$ -norm regularization; nonlinear equations; conjugate gradient method; projection method; global convergence

**MSC:** 65L09; 65K05; 90C30

## 1. Introduction

Discrete ill-posed problems are systems of linear equations arising from the discretization of ill-posed problems. Consider the linear system

$$b = At, \quad (1)$$

where  $t \in R^n$  is an original signal,  $A \in R^{m \times n}$  ( $m < n$ ) is a linear map, and  $b \in R^m$  is an observed data. In particular, the original signal  $t$  is assumed to be sparse, that is, it has very few non-zero coefficients.

Since  $m < n$ , the linear system (1) is usually referred to as ill-conditioned or under-determined problems. In compressive sensing (CS), it is possible to regain the sparse signal  $t$  from the linear system (1), by finding the solution of the  $\ell_0$ -regularized problem

$$\min_t \{\|t\|_0 : At = b\}, \quad (2)$$

where  $\|t\|_0$  denotes the nonzero components in  $t$ . Unfortunately, the  $\ell_0$ -norm is **NP**-hard in general. Based on this, researchers have developed alternative model by replacing the  $\ell_0$ -norm with  $\ell_1$ -norm. Thus, they solved the following problem:

$$\min_t \{\|t\|_1 : At = b\}, \quad (3)$$

Under some mild assumptions, Donoho [1] proved that solution(s) of problem (2) also solves (3). In most applications, the observed value  $b$  usually contains some noise, thus problem (3) can be relaxed to the penalized least squares problem

$$\min_t \tau \|t\|_1 + \frac{1}{2} \|At - b\|_2^2, \quad (4)$$

where  $\tau > 0$ , balancing the trade-off between sparsity and residual error. Problems of the form (4) have become familiar over the past three decades, particularly in *compressive sensing* contexts. Interested readers may refer to the recent papers (Refs. [2,3]) for more details.

In order to address problem (4), several numerical algorithms have been proposed, for instance, Daubechies, Defrise, and Demol [4] proposed the iterative shrinkage thresholding (IST) algorithm; thereafter, Beck and Teboulle [5] proposed the fast iterative shrinkage thresholding algorithm (FISTA). These algorithms are well known due to their simplicity and efficiency. Likewise, Hale, Yin, and Zhang [6] proposed the fixed-point continuous search method [6], and an acceleration technique was introduced by nonmonotone line search with the Barzilai–Borwein stepsize [7]. He, Chang, and Osher [8] introduced the unconstrained formulation of  $\ell_1$ - regularization problem, where the Bregman iterative approach was used to obtain the solutions to problem (4). The proximal forward backward splitting technique is another efficient technique for solving (4). This technique is based on the proximal operator introduced by Moreau [9]. Another type of method for solving the problem (4) is by using the gradient descent method. Quite recently, Figueiredo, Nowak, and Wright [10] first developed a gradient projection method to solve the sparse reconstruction problem. Thereafter, Xiao and Zhu [11,12] proposed a conjugate gradient projection method and a spectral gradient method to solve problem (4), respectively. Unlike IST and FISTA, in order to solve problem (4), the problem was first transformed into a monotone operator equation; see Section 2. Thereafter, an algorithm is developed to solve these systems of equations.

With the approximate equivalence between problem (4) and a system of monotone operator equations, one of the methods for solving the systems of nonlinear monotone operator equations is the conjugate gradient method. Considering the importance of the method, several extensions of this method have been proposed. The three-term conjugate gradient method happens to be one such extension. The first two three-term conjugate gradient method was introduced by Beale [13] and Nazareth [14] to weaken the condition of global convergence of the two-term conjugate gradient method. It is clear that, due to the existence of an additional parameter in the three-term conjugate gradient schemes, establishing the sufficient descent property is more accessible than the two-term conjugate gradient ones. To this end, the three-term conjugate gradient methods are presented, analyzed, and extensively studied in several references because of their advantages in the descent property and the computational performance. The references [15–17] have proposed different three-term conjugate gradient methods, and shown their specific properties, global convergence, and numerical performance. Tang, Li and Cui [18] presented a new three-term conjugate gradient approach based on the technique of the least-squares. Their approach incorporates the advantage of

two existing efficient conjugate gradient approaches and also generates sufficient descent direction without the aid of a line search procedure. Preliminary numerical tests indicate that their method is efficient. Due to the simplicity and low storage requirements of the conjugate gradient method, numerous researchers have recently extended a number of conjugate gradient algorithms designed to solve unconstrained optimization problem to solve large-scale nonlinear equations. Using the popular CG\_DESCENT method [19], Xiao and Zhu [12] recently constructed a conjugate gradient method (CGD) based on the projection scheme of Solodov and Svaiter [20] to solve monotone nonlinear operator equations with convex constraints. The method was also successfully used to solve the sparse signal in compressive sensing. Interested readers may refer to the following articles [21–27] for an overview of algorithms used for solving monotone operator equations.

Inspired by the work of Xiao and Zhu [12], the least-squares-based three-term conjugate gradient method (LSTT) for solving unconstrained optimization problems by Tang, Li, and Cui [18] and the projection technique of Solodov and Svaiter [20], we further study, analyze, and construct a derivative-free least-square-based three-term conjugate gradient method to solve the  $\ell_1$ -norm problem arising from the reconstruction of sparse signal and image in compressive sensing. The method can be viewed as an extension of the LSTT method for solving unconstrained optimization problem and a projection technique. Under the monotonicity and Lipchitz continuity assumption, the global convergence of the proposed method is established using the backtracking line search. Computational experiments are carried out to reconstruct sparse signal and image in compressive sensing. The numerical results indicate that the proposed method is more efficient and robust.

The rest of the paper is organized as follows. In Section 2, we give a review of the reformulation of problem (4) into a convex quadratic program problem by Figueiredo et al. [10]. In Section 3, we present the motivation and general algorithm of the proposed method. The global convergence of the proposed algorithm is presented in Section 4. In Section 5, numerical experiments are presented to illustrate the efficiency of our algorithm. Unless otherwise stated, throughout this paper, the symbol  $\|\cdot\|$  denotes the Euclidean norm. Furthermore, the projection map denoted as  $P_S$ , which is a mapping from  $R^n$  onto the non-empty, closed and convex subset  $S \subseteq R^n$ , that is,

$$P_S(t) := \arg \min\{\|t - y\| \mid y \in S\},$$

which has the well known nonexpansive property, that is,

$$\|P_S(h) - P_S(g)\| \leq \|h - g\|, \forall h, g \in R^n. \tag{5}$$

## 2. Reformulation of the Model

Figueiredo, Nowak, and Wright [10] gave the reformulation of the minimization problem (4) into a quadratic programming problem as follows. Consider any vector  $t \in R^n$ ,  $t$  can be rewritten as follows:

$$t = u - v, \quad u \geq 0, \quad v \geq 0,$$

where  $u \in R^n, v \in R^n$  and  $u_i = (t_i)_+, v_i = (-t_i)_+$  for all  $i \in [1, n]$  with  $(\cdot)_+ = \max\{0, \cdot\}$ . Therefore, the  $\ell_1$ -norm could be represented as  $\|t\|_1 = e_n^T u + e_n^T v$ , where  $e_n$  is an  $n$ -dimensional vector with all elements one. Thus, (4) was rewritten as

$$\min_{u,v} \left\{ \frac{1}{2} \|b - A(u - v)\|^2 + \tau e_n^T u + \tau e_n^T v : u, v \geq 0 \right\}. \tag{6}$$

Moreover, from [10], with no difficulty, (6) can be rewritten as the quadratic program problem with box constraints. That is,

$$\min_z \frac{1}{2} z^T H z + c^T z, \quad z \geq 0, \tag{7}$$

where  $z = \begin{bmatrix} u \\ v \end{bmatrix}$ ,  $c = \tau e_{2n} + \begin{bmatrix} -y \\ y \end{bmatrix}$ ,  $b = A^T y$ ,  $H = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}$ .

Simple calculation shows that  $H$  is a semi-definite positive matrix. Hence, (7) is a convex quadratic program problem, and it is equivalent to

$$F(z) = \min\{z, Hz + c : z \in S\} = 0. \tag{8}$$

The function  $F$  is a vector-valued and the “min” interpreted as componentwise minimum. From ([28], Lemma 3) and ([11], Lemma 2.2), we know that the mapping  $F$  is Lipschitz continuous and monotone. Hence, the algorithm for solving problem (8) can be equally and effectively used to solve problem (4).

### 3. Algorithm

Recently, Chunning, Shuangyu, and Zengru [18] proposed a new three-term conjugate gradient method based on the least-squares technique for solving the following unconstrained optimization problem:

$$\min\{f(t) : t \in R^n\}$$

where the function  $f$  is continuously differentiable from  $R^n$  into  $R$  and the gradient  $\nabla f(t)$  is available. Similar to most conjugate gradient methods, the iterative scheme of the conjugate gradient method developed in [18] generates a sequence of iterates by letting

$$t_{k+1} = t_k + \alpha_k d_k, \quad k \geq 0$$

where  $\alpha_k$  is the steplength and the search direction  $d_k$  is updated by

$$d_k := \begin{cases} -\nabla f(t_k) + \tilde{\beta}_k d_{k-1} - \theta_k \bar{y}_{k-1} & \text{if } k > 0, \\ -\nabla f(t_k) & \text{if } k = 0, \end{cases} \tag{9}$$

where  $\bar{y}_{k-1} = \nabla f(t_k) - \nabla f(t_{k-1})$ ,  $\tilde{\beta}_k$  and  $\theta_k$  are scalar computed as

$$\tilde{\beta}_k := \beta_k^{MHS}(\tau_k^*) := \beta_k^{HS} - \frac{\nabla f(t_k)^T d_{k-1}}{\|d_{k-1}\|^2} := \frac{\nabla f(t_k)^T \bar{y}_{k-1}}{\bar{y}_{k-1}^T d_{k-1}} - \frac{\nabla f(t_k)^T d_{k-1}}{\|d_{k-1}\|^2},$$

$$\theta_k := \frac{\nabla f(t_k)^T d_{k-1}}{\bar{y}_{k-1}^T d_{k-1}}.$$

A valid approach for solving (8) is by using a derivative-free line search to ensure the step-size  $\alpha_k$  [29]. To this end, we present the following derivative-free least-square three term conjugate gradient projection algorithm (Algorithm 1).

---

**Algorithm 1** DF-LSTT

---

**Input.** Choose any arbitrary initial point  $t_0 \in S$ , the positive constants:  $Tol \in (0, 1)$ ,  $\zeta \in (0, 2)$ ,  $\rho \in (0, 1)$ ,  $\beta > 0$ ,  $\varsigma > 0$ .

**Step 0.** Let  $d_0 = -F(t_0)$  and  $k := 0$ .

**Step 1.** Determine the step-size  $\alpha_k = \beta \rho^i$ , where  $i$  is the smallest non-negative integer such that the following line search is satisfied:

$$-F(t_k + \alpha_k d_k)^T d_k \geq \varsigma \alpha_k \|d_k\|^2. \tag{10}$$

**Step 2.** Compute

$$u_k := t_k + \alpha_k d_k. \tag{11}$$

**Step 3.** If  $u_k \in S$  and  $F(u_k) = 0$ , stop. Otherwise, compute the next iterate by

$$t_{k+1} := P_S[t_k - \delta_k \zeta F(u_k)], \tag{12}$$

where

$$\delta_k := \frac{F(u_k)^T (t_k - u_k)}{\|F(u_k)\|^2} \tag{13}$$

**Step 4.** If the stopping criterion is satisfied, that is, if  $\|F(t_k)\| \leq Tol$ , stop. Otherwise, compute the next search direction  $d_k$  by

$$d_k := -F(t_k) + \beta_k (N_k^*) d_{k-1} - v_k y_{k-1} \tag{14}$$

where

$$y_{k-1} := F(t_k) - F(t_{k-1}), \quad v_k := \frac{F(t_k)^T d_{k-1}}{\tilde{y}_{k-1}^T d_{k-1}}, \tag{15}$$

$$\beta_k (N_k^*) := \beta_k^{MHS} - \frac{F(t_k)^T d_{k-1}}{\|d_{k-1}\|^2} := \frac{y_{k-1}^T F(t_k)}{\tilde{y}_{k-1}^T d_{k-1}} - \frac{F(t_k)^T d_{k-1}}{\|d_{k-1}\|^2}$$

$$\tilde{y}_{k-1} := y_{k-1} + j_{k-1} d_{k-1}, \quad j_{k-1} := 1 + \max \left\{ 0, -\frac{y_{k-1}^T d_{k-1}}{\|d_{k-1}\|^2} \right\} \tag{16}$$

**Step 5.** Finally, we set  $k := k + 1$  and return to step 1.

---

**Remark 1.** In order to ensure that the parameters  $\tilde{\beta}_k$  and  $\theta_k$  are well defined when extending them to solve (4), we re-modify their denominator using the scalar  $\tilde{y}_{k-1}^T d_{k-1}$ . Furthermore, to guarantee the boundness of the search direction, we assume the operator under consideration to be monotone rather than uniformly monotone as assumed in [18].

**Lemma 1.** The search direction  $d_k$  generated by DF-LSTT algorithm is a descent direction. That is,

$$F(t_k)^T d_k = -\|F(t_k)\|^2, \quad \forall k \geq 0. \tag{17}$$

**Proof.** Now, by direct computation, we have

$$\begin{aligned} F(t_k)^T d_k &= -\|F(t_k)\|^2 + \frac{y_{k-1}^T F(t_k)}{\tilde{y}_{k-1}^T d_{k-1}} F(t_k)^T d_{k-1} - \frac{F(t_k)^T d_{k-1}}{\|d_{k-1}\|^2} F(t_k)^T d_{k-1} + \frac{F(t_k)^T d_{k-1}}{\tilde{y}_{k-1}^T d_{k-1}} F(t_k)^T y_{k-1} \\ &= -\|F(t_k)\|^2 - \frac{(F(t_k)^T d_{k-1})^2}{\|d_{k-1}\|^2} \\ &\leq -\|F(t_k)\|^2. \end{aligned}$$

This completes the proof.  $\square$

### 4. Global Convergence

In this section, we investigate the global convergence property of DF-LSTT algorithm for solving (8). For this purpose, we make the following assumptions.

**Assumption 1.**

A1. The mapping  $F$  is Lipschitz continuous, that is, there exists a constant  $L > 0$  such that

$$\|F(t) - F(y)\| \leq L\|t - y\|, \quad \forall t, y \in R^n \tag{18}$$

A2. Xiao et al. [11] proved that, for the problem (8),  $F$  is monotone. That is,

$$(F(t) - F(y))^T(t - y) \geq 0, \quad \forall t, y \in R^n. \tag{19}$$

**Lemma 2.** Suppose that Assumption 1 holds. Let  $\{u_k\}$  and  $\{t_k\}$  be sequences generated by (11) and (12) in the DF-LSTT algorithm. Then, the following statements hold:

1.  $\{t_k\}$  and  $\{u_k\}$  are bounded.
2.  $\lim_{k \rightarrow \infty} \|u_k - t_k\| = 0$ .
3.  $\lim_{k \rightarrow \infty} \|t_{k+1} - t_k\| = 0$ .

**Proof.** Since  $F$  is monotone, then, for any solution  $t_*$  of problem (8), we have

$$\begin{aligned} F(u_k)^T(t_k - t_*) &= F(u_k)^T(t_k - u_k) + F(u_k)^T(u_k - t_*) \\ &\geq F(u_k)^T(t_k - u_k) + F(t_*)^T(u_k - t_*) \\ &= F(u_k)^T(t_k - u_k) \\ &\geq \zeta \alpha_k^2 \cdot \|d_k\|^2 \end{aligned} \tag{20}$$

$$= \zeta \|t_k - u_k\|^2. \tag{21}$$

Note that Equation (20) is obtained from the line search. From (5), we get

$$\begin{aligned} \|t_{k+1} - t_*\|^2 &= \|P_S[t_k - \delta_k \zeta F(u_k)] - t_*\|^2 \\ &\leq \|t_k - \delta_k \zeta F(u_k) - t_*\|^2 \\ &= \|t_k - t_*\|^2 - 2\delta_k \zeta F(u_k)^T(t_k - t_*) + \delta_k^2 \zeta^2 \|F(u_k)\|^2 \\ &\leq \|t_k - t_*\|^2 - 2\delta_k \zeta F(u_k)^T(t_k - u_k) + \delta_k^2 \zeta^2 \|F(u_k)\|^2 \end{aligned} \tag{22}$$

$$\begin{aligned} &= \|t_k - t_*\|^2 - \zeta(2 - \zeta) \frac{(F(u_k)^T(t_k - u_k))^2}{\|F(u_k)\|^2} \\ &\leq \|t_k - t_*\|^2 - \zeta(2 - \zeta) \frac{\zeta^2 \|t_k - u_k\|^4}{\|F(u_k)\|^2} \end{aligned} \tag{23}$$

$$\leq \|t_k - t_*\|^2 \tag{24}$$

where (22) and (23) follow from (21).

From (24), it is easy to see that the sequence  $\{t_k\}$  is bounded. That is,

$$\|t_k\| \leq \gamma, \quad \forall k \geq 0. \tag{25}$$

Furthermore, by (18), we have

$$\|F(t_k)\| = \|F(t_k) - F(t_*)\| \leq L\|t_k - t_*\| \leq L\|t_0 - t_*\|. \tag{26}$$

Letting  $M = L\|t_0 - t_*\|$ , then

$$\|F(t_k)\| \leq M, \quad \forall k \geq 0 \tag{27}$$

By the monotonicity of property given in (19), we know that

$$(F(t_k) - F(u_k))^T(t_k - u_k) \geq 0.$$

Therefore, by Cauchy–Schwarz inequality, we have

$$\|F(t_k)\| \|t_k - u_k\| \geq F(t_k)^T(t_k - u_k) \geq F(u_k)^T(t_k - u_k) \geq \varsigma \|t_k - u_k\|^2,$$

where the last inequality can be implied from (21). Thus, it is easy to obtain that

$$\varsigma \|t_k - u_k\| \leq \|F(t_k)\| \leq M, \forall k \geq 0$$

which implies that  $\{u_k\}$  is bounded. Using the continuity of  $F$ , we know that there exists a constant  $M_1 > 0$ , such that

$$\|F(u_k)\| \leq M_1 \forall k \geq 0. \tag{28}$$

It follows from (23) that

$$\bar{\zeta}(2 - \zeta) \frac{\varsigma^2 \|t_k - u_k\|^4}{\|F(u_k)\|^2} \leq \|t_k - t_*\|^2 - \|t_{k+1} - t_*\|^2. \tag{29}$$

Adding (29) for  $k \geq 0$ , we obtain

$$\bar{\zeta}(2 - \zeta) \frac{\varsigma^2}{M_1^2} \sum_{k=0}^{\infty} \|t_k - u_k\|^4 \leq \sum_{k=0}^{\infty} (\|t_k - t_*\|^2 - \|t_{k+1} - t_*\|^2) \leq \|t_0 - t_*\|^2 < \infty \tag{30}$$

Equation (30) implies that

$$\lim_{k \rightarrow \infty} \|t_k - u_k\| = 0. \tag{31}$$

Hence, the second assertion holds.

From (5), we have

$$\|t_{k+1} - t_k\| = \|P_S[t_k - \delta_k F(u_k)] - t_k\| \leq \|\delta_k \zeta F(u_k)\|,$$

then by (13) and Cauchy–Schwarz inequality, we obtain

$$\|t_{k+1} - t_k\| \leq \zeta \|u_k - t_k\|,$$

which shows that the third assertion holds.  $\square$

**Theorem 1.** Suppose that Assumption 1 holds. Let the sequence  $u_k$  and  $t_k$  be generated by (11) and (12) in the DF-LSTT algorithm. Then,

$$\liminf_{k \rightarrow \infty} \|F(t_k)\| = 0. \tag{32}$$

**Proof.** Suppose that conclusion (32) does not hold, that is, there exists  $\kappa_0 \geq 0$  such that

$$\|F(t_k)\| \geq \kappa_0, \forall k \geq 0. \tag{33}$$

Equation (33) together with (17) imply that

$$\|d_k\| \geq \varphi, \forall k \geq 0.$$

We note that the sequences  $\{t_k\}$ ,  $\{u_k\}$ ,  $\{F(t_k)\}$ , and  $\{F(u_k)\}$  are bounded from (25), (31), (27), and (28), respectively. In addition, from the Lipchitz continuity and (25), we have

$$\begin{aligned} \|y_{k-1}\| &= \|F(t_k) - F(t_{k-1})\| \\ &\leq L\|t_k - t_{k-1}\| \\ &= L(\|t_k\| + \|t_{k-1}\|) \\ &\leq 2L\gamma. \end{aligned}$$

On the other hand, from the definition of  $\tilde{y}_{k-1}$ , it holds that

$$\tilde{y}_{k-1}^T d_{k-1} \geq y_{k-1}^T d_{k-1} + \|d_{k-1}\|^2 - y_k^T d_{k-1} = \|d_{k-1}\|^2. \tag{34}$$

Thus, by Cauchy–Schwartz inequality, we obtain

$$\begin{aligned} |\beta_k(N_k^*)| &\leq \frac{|y_{k-1}^T F(t_k)|}{\tilde{y}_{k-1}^T d_{k-1}} + \frac{|F(t_k)^T d_{k-1}|}{\|d_{k-1}\|^2} \leq \frac{\|y_{k-1}\| \|F(t_k)\|}{\|d_{k-1}\|^2} + \frac{\|F(t_k)\|}{\|d_{k-1}\|} \\ &\leq \frac{2LM\gamma}{\varphi^2} + \frac{M}{\varphi} \\ &= \frac{M}{\varphi} \left( \frac{2L\gamma}{\varphi} + 1 \right) \end{aligned} \tag{35}$$

Note that the last inequality in (35) can be easily obtained using (34). Therefore, from (14), it follows that

$$\begin{aligned} \|d_k\| &\leq \|F(t_k)\| + |\beta_k(N_k^*)| \|d_{k-1}\| + \frac{\|F(t_k)\| \|d_{k-1}\|}{\|d_{k-1}\|^2} \|y_{k-1}\| \\ &\leq M + \left( \frac{2ML\gamma}{\varphi} + M \right) + \frac{2ML\gamma}{\varphi} \\ &= M + \frac{\varphi M + 4ML\gamma}{\varphi} \triangleq \chi \end{aligned}$$

From (22),  $\alpha_k \varrho^{-1}$  does not satisfy (10). Thus, we have

$$-F(t_k + \alpha_k \varrho^{-1} d_k)^T d_k < \varsigma \alpha_k \varrho^{-1} \|d_k\|^2.$$

It follows from Lemma 1 that

$$\begin{aligned} \|F(t_k)\|^2 &\leq -F(t_k)^T d_k \\ &\leq (F(t_k + \alpha_k \varrho^{-1} d_k) - F(t_k))^T d_k - F(t_k + \alpha_k \varrho^{-1} d_k)^T d_k \\ &\leq L\alpha_k \varrho^{-1} \|d_k\|^2 + \varsigma \alpha_k \varrho^{-1} \|d_k\|^2 \\ &\leq \alpha_k \varrho^{-1} (L + \varsigma) \|d_k\|^2. \end{aligned} \tag{36}$$

Equation (36) is obtained by using Cauchy–Schwartz inequality and Lipchitz continuity (18). Therefore, it holds that

$$\alpha_k \|d_k\|^2 \geq \frac{\varrho \|F(t_k)\|^2}{(L + \varsigma)} \geq \frac{\varrho \kappa_0^2}{(L + \varsigma)}.$$

Note, from (33), the last inequality is obtained which implies that

$$\lim_{k \rightarrow \infty} \|t_k - u_k\| > 0$$

holds, which contradicts (31). Thus, (32) holds.  $\square$



## 5. Numerical Experiment

We present numerical experiments to show the efficiency of the DF-LSTT method. The experiments presented here are of two types. The first experiment involves applying the DF-LSTT method to solve the  $\ell_1$ -norm regularization problem arising in compressive sensing. The second involves testing DF-LSTT in solving some given convex constrained nonlinear equations with different initial points and various dimensions. The implementations were performed using Matlab R2019b Update 1 (9.7.0.1216025, Mathwork Inc, Massachusetts, USA) on a HP PC (Hewlett-Packard, California, USA) with CPU 2.4 GHz, 8.0 GB RAM with the Windows 10 operating system.

### 5.1. Experiments on the $\ell_1$ -Norm Regularization Problem in Compressive Sensing

We begin by considering a typical compressive sensing scenario where the ultimate goal is to reconstruct a length- $n$  sparse signal from  $m$  observations ( $m \ll n$ ) with Gaussian noise, where the number of samples is dramatically smaller than the size of the original signal. We compare DF-LSTT with the CGD conjugate gradient method [12] and the PCG projection method [30] designed to solve nonlinear equations with convex constrained and signal recovery problems.

As a consequence of the limited memory of our PC, in this test, we considered a small size of signal with  $n = 2^{11}$ ,  $m = 2^9$  and the original  $t$  contains  $2^6$  randomly non-zero elements. Similar to [11,12,31], the quality of the restored signal is assessed by the mean of squared error (MSE) to the original signal  $t$ , that is,

$$MSE := \frac{1}{n} \|t - t^*\|^2$$

where  $t^*$  is the restored signal. In this test, we generate the random matrix  $A$  using the Matlab command `rand(n,k)`. In addition, noise is appropriately added to the observed data computed by

$$b = At + \eta \tag{37}$$

where  $\eta$  is the Gaussian noise with  $N(0, 10^{-4})$ . The DF-LSTT algorithm is implemented with the following parameters:  $\beta = 10$ ,  $\rho = 0.55$ ,  $\zeta = 10^{-4}$ .

For the compared methods, their parameters are set as reported in their respective papers. In line with [32], we chose the parameter  $\tau$  in the merit function  $f(t) = \tau \|t\| + \frac{1}{2} \|b - At\|^2$  as  $\tau = 0.008 \|A^T b\|_\infty$  and the initial point for all algorithms also starts at  $t_0 = A^T b$ . The process terminates when

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

where  $f_k$  denotes the function value at  $t_k$ . Note, for this test, we only observe the convergence behavior of each method to obtain a similar accuracy solution.

In view of the plots depicted in Figure 1, DF-LSTT wins in decoding sparse signals in compressive sensing in terms of number of iterations for around 116 and around 0.69s. Table 1 contains the number of iterations, mean of Squared error (MSE), and time of the decoding of sparse signal for over 10 runs for the algorithms tested. The reported results in Figure 2 shows that, in decoding sparse signal in compressive sensing, DF-LSTT is faster than CGD and PCG with a clear lowest number of iterations.

Table 1. Results for sparse signal recovery.

DF-LSTT			CGD			PCG			
ITER	MSE	TIME	ITER	MSE	Time	ITER	MSE	Time	
137	$7.39 \times 10^{-6}$	1.31	259	$4.51 \times 10^{-5}$	1.7	184	$1.50 \times 10^{-5}$	0.88	
88	$9.18 \times 10^{-6}$	0.73	187	$7.04 \times 10^{-5}$	1.38	144	$1.63 \times 10^{-5}$	1.42	
90	$1.40 \times 10^{-5}$	0.66	231	$3.78 \times 10^{-5}$	1.73	144	$1.99 \times 10^{-5}$	1.61	
86	$1.26 \times 10^{-5}$	0.78	228	$2.15 \times 10^{-5}$	1.41	82	$5.32 \times 10^{-4}$	0.78	
97	$1.22 \times 10^{-5}$	0.7	245	$6.75 \times 10^{-5}$	1.55	118	$5.99 \times 10^{-5}$	0.78	
87	$9.72 \times 10^{-6}$	0.64	199	$5.13 \times 10^{-5}$	1.33	82	$4.50 \times 10^{-4}$	0.83	
115	$5.39 \times 10^{-6}$	0.84	211	$3.67 \times 10^{-5}$	1.64	152	$9.68 \times 10^{-6}$	0.97	
89	$1.31 \times 10^{-5}$	1.27	158	$1.56 \times 10^{-4}$	3.14	150	$1.16 \times 10^{-5}$	1.59	
105	$1.35 \times 10^{-5}$	0.63	280	$4.82 \times 10^{-5}$	1.89	152	$2.23 \times 10^{-5}$	0.86	
97	$5.22 \times 10^{-6}$	0.63	228	$3.89 \times 10^{-5}$	1.45	154	$8.00 \times 10^{-6}$	2.3	
Average	$99.1$	$1.02 \times 10^{-5}$	$0.819$	$222.6$	$5.73 \times 10^{-5}$	$1.722$	$136.2$	$1.14 \times 10^{-4}$	$1.202$

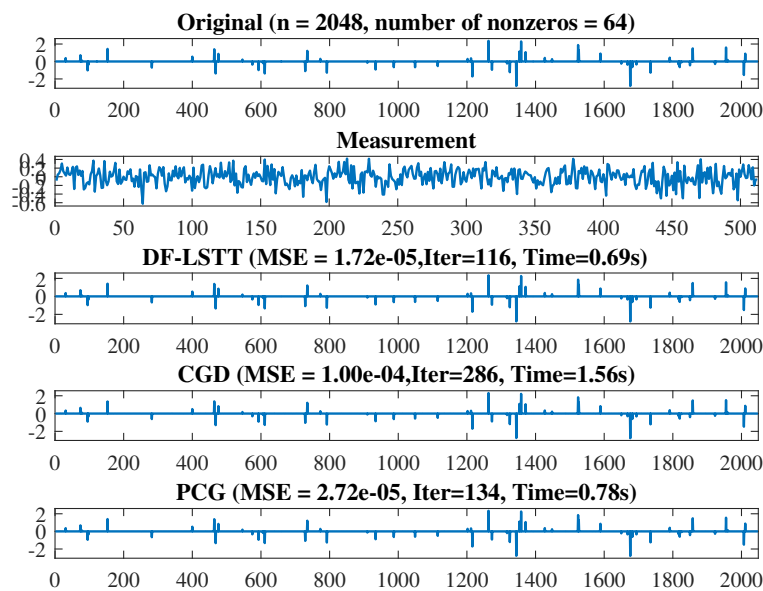
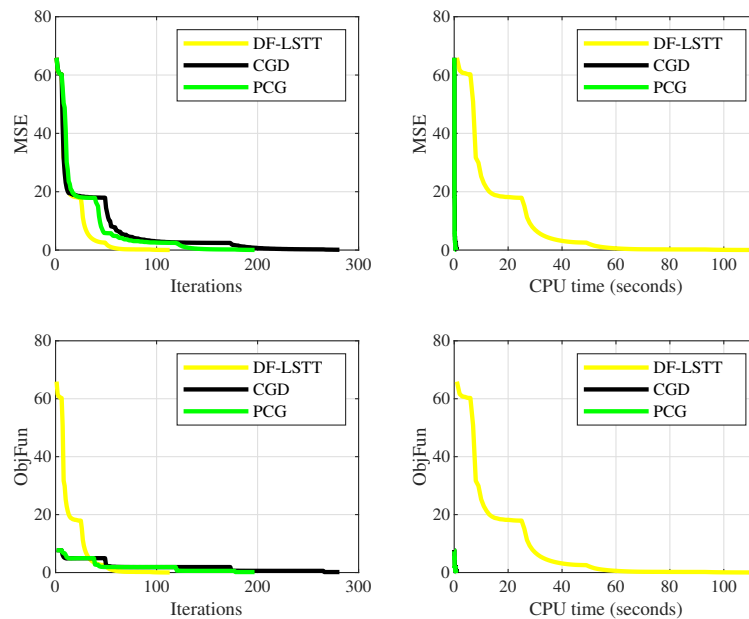


Figure 1. Illustration of the sparse signal recovery. From the top to the bottom is the original signal (First plot), the measurement (Second plot), and the reconstructed signals by DF-LSTT (Third plot), CGD (Fourth plot), and PCG (Fifth plot).



**Figure 2.** Comparison results of the DF-LSTT, CGD, and PCG algorithms with the signal parameter chosen as  $n = 2048$ ,  $m = 512$ ,  $\tau = 0.00396321$ . The  $x$ -axes represent the number of iterations (**top left** and **bottom left**) and the CPU time in seconds (**top right** and **bottom right**). The  $y$ -axes represent the MSE (**top left** and **top right**) and the function values (**bottom left** and **right**).

Next, the effectiveness and robustness of DF-LSTT algorithm is illustrated in an image de-blurring problem. We carried out our experiment using some widely used test images obtained from <http://hlevkin.com/06testimages.htm>. DF-LSTT is compared with the state-of-art methods CGD proposed by Xiaoh and Zhu [12], SGCS [11], and MFR [33].

The performance evaluation criteria to measure the quality of restoration by the methods are measured by the signal-to-noise ratio (SNR) defined as

$$SNR := 20 \times \log_{10} \left( \frac{\|t\|}{\|t^* - t\|} \right),$$

the peak signal-to-noise ratio (PSNR) [34], and the structural similarity index (SSIM) [35]. For fairness in comparing the algorithms, the iteration process of all algorithms started at  $t_0 = A^T b$  and terminates when  $Tol < 10^{-5}$ . For the image de-blurring experiment, the following parameters were used in our implementation:  $\beta = 0.001$ ,  $\rho = 0.6$ ,  $\zeta = 10^{-4}$ . We tested several images including Tiffany ( $512 \times 512$ ), Lena ( $512 \times 512$ ), Barbara ( $720 \times 576$ ), degraded by Gaussian blur and 10% Gaussian noise.

In Table 2, we report the performance for SNR, PSNR, and SSIM for DF-LSTT, CGD, SGCS, and the MFRM method in recovery blurred and noisy images. We can see that the SNR, PSNR, and SSIM of the test images calculated by the DF-LSTT algorithm are a bit higher than CGD, SGCS, and MFRM. The higher value of SNR, PSNR, and SSIM reflects better quality of restoration.

**Table 2.** Numerical results of DF-LSTT, CGD, SGCS, and MFRM methods in image restoration

Image	DF-LSTT			CGD			SGCS			MFRM		
	SNR	PSNR	SSIM	SNR	PSNR	SSIM	SNR	PSNR	SSIM	SNR	PSNR	SSIM
Tiffany	21.25	23.08	0.9204	21.20	23.04	0.9193	21.24	23.07	0.9202	20.87	22.70	0.9128
Lenna	16.98	22.31	0.9176	16.93	22.26	0.9166	16.96	22.29	0.9173	16.60	21.94	0.9104
Barbara	13.81	20.23	0.6377	13.77	20.19	0.6355	13.80	20.22	0.6373	13.57	19.99	0.6231
Average	17.35	21.87	0.8252	17.30	21.83	0.8238	17.33	21.86	0.8249	17.01	21.54	0.8154

Based on the performance reported in Table 2, the DF-LSTT algorithm restores blurred and noisy images quite well and obtain better quality reconstructed images in an efficient manner. Figures 3 and 4 show the original and blurred images while Figure 5 shows the restored images by each method.



Figure 3. The original test images: From (left), Tiffany, Lenna (middle), and Barbara (right)



Figure 4. Blurred and noisy Barbara and Lenna test images.



Figure 5. Restored images by DF-LSTT (left column), CGD (left middle column), SGCS (right middle column), and MFRM (right column).

### 5.2. Experiments on Some Large-Scaled Monotone Nonlinear Equations

In this subsection, we evaluate the performance of the proposed conjugate gradient method in solving nonlinear equations with convex constraints. We compare the proposed method with CGD [12] and PCG [30]. For each test problem, the stopping condition employed is

$$\|F(t_k)\| \leq 10^{-6}. \tag{38}$$

We also stop the algorithms when the iterations exceed 1000 without achieving convergence. The algorithms are tested using seven different initial points, one of which is randomly generated in  $R^n$ . We ran the algorithms for several dimensions ranging from  $n = 1000$  to  $100,000$ . The parameters chosen for the proposed algorithm are;  $\rho = 0.75$ ,  $\beta = 1$ ,  $\zeta = 10^{-4}$ ,  $\xi = 1.2$ . For the CGD and PCG algorithm, all parameters are chosen as in [12,30], respectively. We use the following well-known benchmark test functions where the mapping  $F$  is taken as

$$F(t) = (f_1(t), f_2(t), \dots, f_n(t))^T,$$

where the associated initial points for these problems are

$$t_1 = (0.1, 0.1, \dots, 0.1)^T, t_2 = (0.2, 0.2, \dots, 0.2)^T, t_3 = (0.5, 0.5, \dots, 0.5)^T, t_4 = (1.2, 1.2, \dots, 1.2)^T, \\ t_5 = (1.5, 1.5, \dots, 1.5)^T, t_6 = (2, 2, \dots, 2)^T, t_7 = rand(n, 1).$$

**Problem 1.** This problem is the Exponential function [36] with constraint set  $S = R_+^n$ , that is,

$$f_1(t) = e^{t_1} - 1, \\ f_i(t) = e^{t_i} + t_i - 1, \text{ for } i = 2, 3, \dots, n.$$

**Problem 2.** Modified Logarithmic function [36] with constraint set  $S = \{t \in R^n : \sum_{i=1}^n t_i \leq n, t_i > -1, i = 1, 2, \dots, n\}$ , that is,

$$f_i(t) = \ln(t_i + 1) - \frac{t_i}{n}, i = 2, 3, \dots, n.$$

**Problem 3.** The function  $f_i(x)$  [37] with  $S = R_+^n$  defined by

$$f_i(t) = \min \left( \min(|t_i|, t_i^2), \max(|t_i|, t_i^3) \right) \text{ for } i = 2, 3, \dots, n.$$

**Problem 4.** The Strictly convex function [38], with constraint set  $S = R_+^n$ , that is,

$$f_i(t) = e^{t_i} - 1, i = 2, 3, \dots, n.$$

**Problem 5.** Strictly convex function II [38], with constraint set  $S = R_+^n$ , that is,

$$f_i(x) = \frac{1}{n} e^{t_i} - 1, i = 2, 3, \dots, n.$$

**Problem 6.** Tridiagonal Exponential function [39] with constraint set  $S = R_+^n$ , that is,

$$f_1(t) = t_1 - e^{\cos(h(t_1+t_2))}, \\ f_i(t) = t_i - e^{\cos(h(t_{i-1}+t_i+t_{i+1}))}, \text{ for } 2 \leq i \leq n-1, \\ f_n(t) = t_n - e^{\cos(h(t_{n-1}+t_n))}, \text{ where } h = \frac{1}{n+1}$$

**Problem 7.** Nonsmooth function [40] with constraint set  $S = \{t \in R^n : \sum_{i=1}^n t_i \leq n, t_i \geq -1, 1 \leq i \leq n\}$  :

$$f_i(t) = t_i - \sin |t_i - 1|, i = 2, 3, \dots, n.$$

**Problem 8.** The function  $f_i(t)$  with  $S = R_+^n$  defined by

$$f_i(t) = 8^{\frac{1}{2}} t_i - 1 \quad i = 1, 2, 3, \dots, n.$$

In Tables 3–10, we report the computation results obtained from the implementation of DF-LSTT, CGD, and PCG algorithm. We report the number of iterations (ITER), the number of function evaluations (FVAL), and the CPU time in seconds (TIME).

We employ the widely used performance profile metric of Dolan and Moré [41] to compare the performance of the methods. The profile metric measures the ratio of each method by its computational outcome versus the computational outcome of the best presented method. The performance profile metric operates in the following manner. Let  $S_m$  and  $P_m$  denote the set of methods and test problems,

respectively. In this section, we see a problem with different dimensions or different initial points as another new problem. For  $n_s$  methods and  $n_p$  problems, the performance profile  $\rho : R \rightarrow [0, 1]$  is defined as follows: for each problem  $p \in P_m$  and for each method  $s \in S_m$ , they define

$$t_{p,s} := (\text{computing time required to solve problem } p \text{ by method } s)$$

The performance ratio is given by  $r_{p,s} := t_{p,s} / \min_{s \in S_m} t_{p,s}$ . Then, the performance profile is defined by

$$\rho(\tau) := n_p^{-1} \text{size}\{p \in P_m \mid \log_2(r_{p,s}) \leq \tau\}, \forall \tau \in R^+, \text{ where } 1 \leq s \leq n_s.$$

We note that  $\rho(\tau)$  is the probability for a method  $s \in S_m$  that  $\log_2(r_{p,s})$  is within a factor  $\tau \in R^+$  of the best possible ratio. Obviously, when  $\tau$  takes certain value, a method with high value of  $\rho(\tau)$  is preferable or represents the best method. As usual, we obtain the number of iterations, number of function evaluations, and the computing time (CPU time) from the performance profile metric. Figures 6 and 7 are the performance profile obtained based on the Dolan and Moré profile. Both figures indicate that the performances of the proposed method is competitive with the other methods. However, from both Figures 6 and 7, DF-LSTT is more efficient because it was able to solve a higher percentage of the test problems with less number of iterations and function evaluations compared to CGD and PCG methods. However, with respect to CPU time, our algorithm did not perform well. This could be as a result of several computations involved in the method.

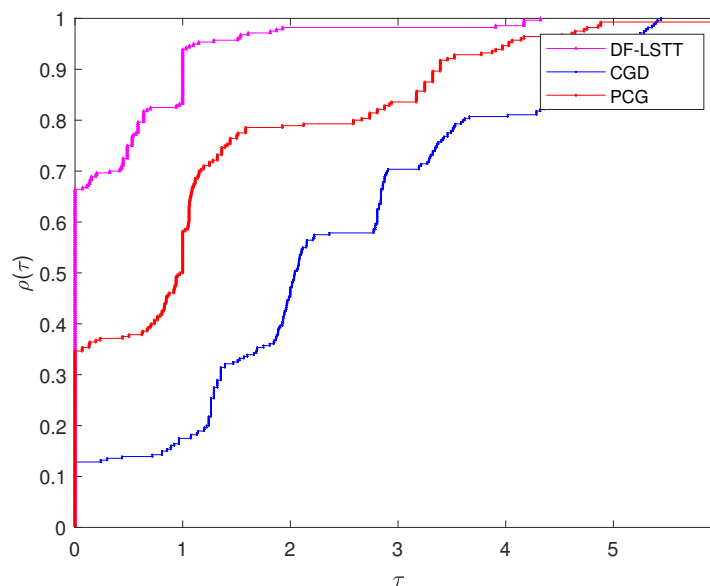


Figure 6. Performance of compared methods relative to the number of iterations.

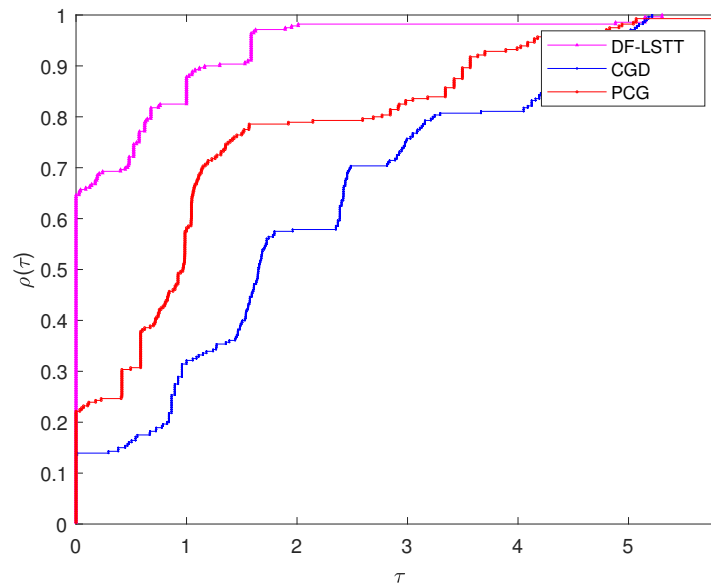


Figure 7. Performance of compared methods relative to the number of function evaluations.

Table 3. Numerical test reports for the three tested methods for Problem 1.

DIM	DF-LSTT					CGD				PCG			
	INP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	$t_1$	2	7	0.011683	$0.00 \times 10^0$	42	125	0.032606	$9.97 \times 10^{-7}$	18	71	0.020567	$5.72 \times 10^{-6}$
	$t_2$	2	7	0.007157	$0.00 \times 10^0$	45	134	0.020408	$9.45 \times 10^{-7}$	18	71	0.017477	$9.82 \times 10^{-6}$
	$t_3$	2	7	0.059349	$0.00 \times 10^0$	48	143	0.025053	$9.82 \times 10^{-7}$	19	75	0.010401	$7.10 \times 10^{-6}$
	$t_4$	2	7	0.010097	$0.00 \times 10^0$	50	149	0.020462	$9.70 \times 10^{-7}$	18	71	0.014602	$8.27 \times 10^{-6}$
	$t_5$	31	124	0.075653	$7.60 \times 10^{-7}$	51	152	0.022198	$8.17 \times 10^{-7}$	63	251	0.035587	$9.58 \times 10^{-6}$
	$t_6$	14	55	0.041615	$0.00 \times 10^0$	51	152	0.017499	$8.56 \times 10^{-7}$	61	243	0.070899	$9.15 \times 10^{-6}$
	$t_7$	44	175	0.052359	$1.83 \times 10^{-12}$	38	113	0.013579	$9.14 \times 10^{-7}$	18	71	0.009228	$9.24 \times 10^{-6}$
5000	$t_1$	2	7	0.023049	$0.00 \times 10^0$	41	122	0.045814	$8.34 \times 10^{-7}$	18	71	0.037294	$7.42 \times 10^{-6}$
	$t_2$	2	7	0.013178	$0.00 \times 10^0$	43	128	0.051819	$9.81 \times 10^{-7}$	19	75	0.030883	$6.53 \times 10^{-6}$
	$t_3$	2	7	0.008287	$0.00 \times 10^0$	47	140	0.056948	$8.05 \times 10^{-7}$	20	79	0.040342	$5.20 \times 10^{-6}$
	$t_4$	2	7	0.008819	$0.00 \times 10^0$	48	143	0.060879	$9.93 \times 10^{-7}$	19	75	0.03691	$8.10 \times 10^{-6}$
	$t_5$	28	112	0.14831	$6.87 \times 10^{-7}$	49	146	0.053006	$8.36 \times 10^{-7}$	62	247	0.0775	$9.53 \times 10^{-6}$
	$t_6$	28	112	0.17661	$3.38 \times 10^{-7}$	49	146	0.061534	$8.76 \times 10^{-7}$	60	239	0.081647	$9.10 \times 10^{-6}$
	$t_7$	19	75	0.10719	$0.00 \times 10^0$	40	119	0.052358	$7.70 \times 10^{-7}$	19	75	0.042343	$9.16 \times 10^{-6}$
10,000	$t_1$	2	7	0.017941	$0.00 \times 10^0$	40	119	0.082014	$8.97 \times 10^{-7}$	18	71	0.055224	$9.50 \times 10^{-6}$
	$t_2$	2	7	0.025597	$0.00 \times 10^0$	43	128	0.074301	$8.26 \times 10^{-7}$	19	75	0.050823	$8.15 \times 10^{-6}$
	$t_3$	2	7	0.015978	$0.00 \times 10^0$	46	137	0.096062	$8.46 \times 10^{-7}$	20	79	0.046486	$6.74 \times 10^{-6}$
	$t_4$	2	7	0.24659	$0.00 \times 10^0$	48	143	0.097356	$8.30 \times 10^{-7}$	20	79	0.05297	$5.11 \times 10^{-6}$
	$t_5$	39	156	0.86102	$4.62 \times 10^{-7}$	48	143	0.11071	$8.75 \times 10^{-7}$	62	247	0.19236	$8.87 \times 10^{-6}$
	$t_6$	2	7	0.023214	$0.00 \times 10^0$	48	143	0.087449	$9.17 \times 10^{-7}$	59	235	0.16649	$9.96 \times 10^{-6}$
	$t_7$	11	43	0.089553	$0.00 \times 10^0$	37	110	0.072561	$7.58 \times 10^{-7}$	20	79	0.055156	$5.82 \times 10^{-6}$
50,000	$t_1$	2	7	0.046111	$0.00 \times 10^0$	39	116	0.29127	$8.43 \times 10^{-7}$	19	75	0.21501	$8.80 \times 10^{-6}$
	$t_2$	2	7	0.088396	$0.00 \times 10^0$	41	122	0.32975	$9.37 \times 10^{-7}$	20	79	0.27646	$7.39 \times 10^{-6}$
	$t_3$	2	7	0.050768	$0.00 \times 10^0$	44	131	0.36349	$9.16 \times 10^{-7}$	21	83	0.28492	$6.31 \times 10^{-6}$
	$t_4$	2	7	0.052953	$0.00 \times 10^0$	46	137	0.34443	$8.84 \times 10^{-7}$	21	83	0.21931	$5.10 \times 10^{-6}$
	$t_5$	34	136	1.336	$2.22 \times 10^{-7}$	46	137	0.41225	$9.34 \times 10^{-7}$	61	243	0.5975	$8.85 \times 10^{-6}$
	$t_6$	2	7	0.10667	$0.00 \times 10^0$	46	137	0.44581	$9.78 \times 10^{-7}$	59	235	0.59777	$8.50 \times 10^{-6}$
	$t_7$	64	256	2.2443	$4.32 \times 10^{-13}$	46	137	0.44419	$8.86 \times 10^{-7}$	21	83	0.2052	$5.79 \times 10^{-6}$
100,000	$t_1$	2	7	0.092134	$0.00 \times 10^0$	39	116	0.55718	$7.72 \times 10^{-7}$	20	79	0.37816	$5.52 \times 10^{-6}$
	$t_2$	2	7	0.14977	$0.00 \times 10^0$	41	122	0.57738	$8.33 \times 10^{-7}$	21	83	0.45721	$4.62 \times 10^{-6}$
	$t_3$	2	7	0.093678	$0.00 \times 10^0$	44	131	0.62707	$7.92 \times 10^{-7}$	21	83	0.54042	$8.78 \times 10^{-6}$
	$t_4$	2	7	0.10506	$0.00 \times 10^0$	45	134	0.63408	$9.66 \times 10^{-7}$	21	83	0.53141	$7.21 \times 10^{-6}$
	$t_5$	39	156	9.4738	$9.14 \times 10^{-7}$	46	137	0.66002	$7.99 \times 10^{-7}$	60	239	1.1271	$9.73 \times 10^{-6}$
	$t_6$	2	7	0.32882	$0.00 \times 10^0$	46	137	0.68921	$8.38 \times 10^{-7}$	58	231	1.1519	$9.42 \times 10^{-6}$
	$t_7$	61	244	13.7075	$7.48 \times 10^{-7}$	41	122	0.58348	$9.17 \times 10^{-7}$	21	83	0.38529	$8.20 \times 10^{-6}$

**Table 4.** Numerical test reports for the three tested methods for Problem 2.

DIM	INP	DF-LSTT				CGD				PCG			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	$t_1$	6	23	0.018184	$1.73 \times 10^{-7}$	55	163	0.032169	$8.99 \times 10^{-7}$	15	58	0.011849	$8.59 \times 10^{-6}$
	$t_2$	6	23	0.014572	$2.51 \times 10^{-7}$	61	181	0.0357	$8.88 \times 10^{-7}$	11	41	0.010006	$9.07 \times 10^{-6}$
	$t_3$	6	23	0.019846	$5.60 \times 10^{-7}$	69	205	0.032634	$8.33 \times 10^{-7}$	17	65	0.012605	$6.44 \times 10^{-6}$
	$t_4$	6	23	0.057022	$1.19 \times 10^{-7}$	76	226	0.028091	$9.17 \times 10^{-7}$	18	68	0.014979	$6.00 \times 10^{-6}$
	$t_5$	8	31	0.030773	$1.98 \times 10^{-7}$	78	232	0.039636	$9.18 \times 10^{-7}$	13	47	0.009625	$7.58 \times 10^{-6}$
	$t_6$	7	27	0.014646	$2.53 \times 10^{-7}$	81	241	0.03899	$8.64 \times 10^{-7}$	18	67	0.011159	$5.40 \times 10^{-6}$
	$t_7$	14	55	0.026256	$6.83 \times 10^{-7}$	72	214	0.032568	$8.68 \times 10^{-7}$	19	73	0.016356	$6.13 \times 10^{-6}$
5000	$t_1$	6	23	0.046357	$4.66 \times 10^{-7}$	59	175	0.08852	$8.02 \times 10^{-7}$	16	62	0.037321	$9.35 \times 10^{-6}$
	$t_2$	6	23	0.07737	$6.75 \times 10^{-7}$	64	190	0.10686	$9.97 \times 10^{-7}$	12	45	0.030551	$8.80 \times 10^{-6}$
	$t_3$	7	27	0.068471	$2.49 \times 10^{-7}$	72	214	0.12192	$9.37 \times 10^{-7}$	18	69	0.047498	$6.98 \times 10^{-6}$
	$t_4$	7	27	0.1053	$7.27 \times 10^{-7}$	80	238	0.12654	$8.26 \times 10^{-7}$	19	72	0.039088	$6.45 \times 10^{-6}$
	$t_5$	8	31	0.065362	$5.25 \times 10^{-7}$	82	244	0.11871	$8.26 \times 10^{-7}$	14	51	0.03502	$6.71 \times 10^{-6}$
	$t_6$	7	27	0.062662	$7.28 \times 10^{-7}$	84	250	0.13106	$9.71 \times 10^{-7}$	19	71	0.040604	$5.71 \times 10^{-6}$
	$t_7$	23	91	0.45265	$1.77 \times 10^{-7}$	75	223	0.16438	$9.73 \times 10^{-7}$	20	77	0.059424	$6.86 \times 10^{-6}$
10,000	$t_1$	6	23	0.081017	$6.73 \times 10^{-7}$	60	178	0.16239	$9.04 \times 10^{-7}$	17	66	0.075512	$6.60 \times 10^{-6}$
	$t_2$	6	23	0.15334	$9.76 \times 10^{-7}$	66	196	0.16906	$9.00 \times 10^{-7}$	13	49	0.041324	$6.11 \times 10^{-6}$
	$t_3$	7	27	0.11084	$3.61 \times 10^{-7}$	74	220	0.19362	$8.46 \times 10^{-7}$	18	69	0.076734	$9.83 \times 10^{-6}$
	$t_4$	5	19	0.076488	$6.03 \times 10^{-7}$	81	241	0.23663	$9.32 \times 10^{-7}$	19	72	0.064377	$9.07 \times 10^{-6}$
	$t_5$	8	31	0.25185	$7.57 \times 10^{-7}$	83	247	0.21784	$9.33 \times 10^{-7}$	14	51	0.050394	$9.18 \times 10^{-6}$
	$t_6$	8	31	0.15676	$1.66 \times 10^{-7}$	86	256	0.23307	$8.77 \times 10^{-7}$	19	71	0.075462	$8.93 \times 10^{-6}$
	$t_7$	27	107	1.1061	$4.66 \times 10^{-7}$	77	229	0.27924	$8.81 \times 10^{-7}$	20	77	0.088795	$9.69 \times 10^{-6}$
50,000	$t_1$	7	27	0.5849	$2.40 \times 10^{-7}$	64	190	0.70165	$8.26 \times 10^{-7}$	18	70	0.25185	$7.37 \times 10^{-6}$
	$t_2$	7	27	0.48516	$3.47 \times 10^{-7}$	70	208	0.75749	$8.23 \times 10^{-7}$	14	53	0.27213	$6.74 \times 10^{-6}$
	$t_3$	7	27	1.0522	$8.23 \times 10^{-7}$	77	229	0.81368	$9.67 \times 10^{-7}$	20	77	0.31128	$5.50 \times 10^{-6}$
	$t_4$	7	27	0.34532	$2.05 \times 10^{-7}$	85	253	0.93153	$8.52 \times 10^{-7}$	21	80	0.28943	$5.07 \times 10^{-6}$
	$t_5$	9	35	4.3347	$2.69 \times 10^{-7}$	87	259	0.94105	$8.53 \times 10^{-7}$	16	59	0.22366	$5.02 \times 10^{-6}$
	$t_6$	8	31	0.92387	$3.80 \times 10^{-7}$	90	268	1.0832	$8.02 \times 10^{-7}$	20	75	0.27706	$8.93 \times 10^{-6}$
	$t_7$	20	79	1.8206	$8.21 \times 10^{-7}$	81	241	1.4008	$8.03 \times 10^{-7}$	22	85	0.39411	$5.41 \times 10^{-6}$
100,000	$t_1$	7	27	0.73406	$3.39 \times 10^{-7}$	65	193	1.3721	$9.34 \times 10^{-7}$	19	74	0.52829	$5.22 \times 10^{-6}$
	$t_2$	7	27	0.49843	$4.92 \times 10^{-7}$	71	211	1.5287	$9.30 \times 10^{-7}$	14	53	0.37191	$9.52 \times 10^{-6}$
	$t_3$	8	31	0.75063	$1.82 \times 10^{-7}$	79	235	1.6725	$8.75 \times 10^{-7}$	20	77	0.54637	$7.78 \times 10^{-6}$
	$t_4$	7	27	0.50287	$3.21 \times 10^{-7}$	86	256	1.8409	$9.64 \times 10^{-7}$	21	80	0.57391	$7.17 \times 10^{-6}$
	$t_5$	9	35	0.76034	$3.81 \times 10^{-7}$	88	262	1.9085	$9.64 \times 10^{-7}$	16	59	0.52223	$7.07 \times 10^{-6}$
	$t_6$	8	31	0.48187	$5.39 \times 10^{-7}$	91	271	1.9642	$9.07 \times 10^{-7}$	21	79	0.72579	$6.32 \times 10^{-6}$
	$t_7$	21	83	1.7606	$1.06 \times 10^{-7}$	82	244	2.5148	$9.07 \times 10^{-7}$	22	85	0.85567	$7.66 \times 10^{-6}$

**Table 5.** Numerical test reports for the three tested methods for Problem 3.

DIM	INP	DF-LSTT				CGD				PCG			
		ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	$t_1$	2	6	0.004359	0	1	2	0.029302	0	1	3	0.006887	0
	$t_2$	2	6	0.004097	0	1	2	0.003218	0	1	3	0.002975	0
	$t_3$	2	6	0.002719	0	1	2	0.002851	0	1	3	0.007665	0
	$t_4$	2	6	0.002236	0	1	3	0.003144	0	1	4	0.007096	0
	$t_5$	2	6	0.002477	0	1	3	0.004593	0	1	4	0.005078	0
	$t_6$	2	6	0.002744	0	1	3	0.003093	0	1	4	0.002427	0
	$t_7$	15	59	0.056123	$7.18 \times 10^{-7}$	1	2	0.003046	0	1	3	0.006539	0
5000	$t_1$	2	6	0.008889	0	1	2	0.008198	0	1	3	0.008733	0
	$t_2$	2	6	0.012555	0	1	2	0.007947	0	1	3	0.008462	0
	$t_3$	2	6	0.008658	0	1	2	0.007763	0	1	3	0.008092	0
	$t_4$	2	6	0.00612	0	1	3	0.008812	0	1	4	0.009649	0
	$t_5$	2	6	0.005853	0	1	3	0.009794	0	1	4	0.00889	0
	$t_6$	2	6	0.006947	0	1	3	0.007755	0	1	4	0.007802	0
	$t_7$	18	71	0.31812	$7.95 \times 10^{-7}$	1	2	0.010464	0	1	3	0.008169	0
10,000	$t_1$	2	6	0.015963	0	1	2	0.012685	0	1	3	0.012387	0
	$t_2$	2	6	0.016001	0	1	2	0.011649	0	1	3	0.01044	0
	$t_3$	2	6	0.015956	0	1	2	0.010229	0	1	3	0.010306	0
	$t_4$	2	6	0.018197	0	1	3	0.011267	0	1	4	0.01704	0
	$t_5$	2	6	0.011355	0	1	3	0.01182	0	1	4	0.010001	0
	$t_6$	2	6	0.018842	0	1	3	0.012149	0	1	4	0.012029	0
	$t_7$	18	71	0.38333	$8.40 \times 10^{-7}$	1	2	0.010945	0	1	3	0.011036	0
50,000	$t_1$	2	6	0.069633	0	1	2	0.038998	0	1	3	0.038391	0
	$t_2$	2	6	0.058314	0	1	2	0.040189	0	1	3	0.04888	0
	$t_3$	2	6	0.095231	0	1	2	0.038922	0	1	3	0.036443	0
	$t_4$	2	6	0.046998	0	1	3	0.040942	0	1	4	0.043418	0
	$t_5$	2	6	0.044827	0	1	3	0.053045	0	1	4	0.040382	0
	$t_6$	2	6	0.091957	0	1	3	0.042239	0	1	4	0.041188	0
	$t_7$	18	71	1.2966	$5.49 \times 10^{-7}$	1	2	0.046987	0	1	3	0.040932	0
100,000	$t_1$	2	6	0.11838	0	1	2	0.090727	0	1	3	0.077927	0
	$t_2$	2	6	0.11658	0	1	2	0.077024	0	1	3	0.074788	0
	$t_3$	2	6	0.12338	0	1	2	0.10601	0	1	3	0.082006	0
	$t_4$	2	6	0.16886	0	1	3	0.085545	0	1	4	0.08095	0
	$t_5$	2	6	0.086942	0	1	3	0.090827	0	1	4	0.12206	0
	$t_6$	2	6	0.11021	0	1	3	0.080531	0	1	4	0.080564	0
	$t_7$	20	79	3.0379	$3.93 \times 10^{-7}$	1	2	0.078351	0	1	3	0.07851	0







**Table 10.** Numerical test reports for the three tested methods for Problem 8.

DIM	DF-LSTT				CGD				PCG				
	INP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
1000	$t_1$	13	52	0.011502	$4.53 \times 10^{-7}$	21	62	0.007103	$9.28 \times 10^{-7}$	9	35	0.007659	$2.15 \times 10^{-6}$
	$t_2$	13	52	0.010497	$2.74 \times 10^{-7}$	21	62	0.007684	$5.62 \times 10^{-7}$	8	31	0.005796	$7.72 \times 10^{-6}$
	$t_3$	12	48	0.018922	$7.51 \times 10^{-7}$	21	62	0.006208	$5.36 \times 10^{-7}$	8	31	0.004682	$7.36 \times 10^{-6}$
	$t_4$	14	56	0.011298	$2.60 \times 10^{-7}$	23	68	0.00871	$5.84 \times 10^{-7}$	9	35	0.005209	$7.18 \times 10^{-6}$
	$t_5$	14	56	0.0119	$3.53 \times 10^{-7}$	23	68	0.006241	$7.91 \times 10^{-7}$	9	35	0.004898	$9.73 \times 10^{-6}$
	$t_6$	14	56	0.020175	$5.07 \times 10^{-7}$	24	71	0.006771	$4.93 \times 10^{-7}$	10	39	0.005248	$2.36 \times 10^{-6}$
	$t_7$	9	36	0.007601	$9.21 \times 10^{-7}$	22	65	0.005934	$5.11 \times 10^{-7}$	9	35	0.004781	$2.82 \times 10^{-6}$
5000	$t_1$	14	56	0.044747	$2.48 \times 10^{-7}$	22	65	0.030281	$9.01 \times 10^{-7}$	9	35	0.014961	$4.81 \times 10^{-6}$
	$t_2$	13	52	0.039991	$6.13 \times 10^{-7}$	22	65	0.037367	$5.46 \times 10^{-7}$	9	35	0.019975	$2.91 \times 10^{-6}$
	$t_3$	13	52	0.061786	$4.11 \times 10^{-7}$	22	65	0.024256	$5.20 \times 10^{-7}$	9	35	0.012236	$2.78 \times 10^{-6}$
	$t_4$	14	56	0.037006	$5.82 \times 10^{-7}$	24	71	0.030963	$5.67 \times 10^{-7}$	10	39	0.012613	$2.71 \times 10^{-6}$
	$t_5$	14	56	0.042827	$7.89 \times 10^{-7}$	24	71	0.027167	$7.68 \times 10^{-7}$	10	39	0.012167	$3.67 \times 10^{-6}$
	$t_6$	15	60	0.040744	$2.77 \times 10^{-7}$	25	74	0.032369	$4.79 \times 10^{-7}$	10	39	0.014305	$5.27 \times 10^{-6}$
	$t_7$	10	40	0.031693	$5.07 \times 10^{-7}$	23	68	0.017202	$5.02 \times 10^{-7}$	9	35	0.011532	$6.18 \times 10^{-6}$
10,000	$t_1$	14	56	0.1304	$3.51 \times 10^{-7}$	23	68	0.031436	$5.53 \times 10^{-7}$	9	35	0.015852	$6.80 \times 10^{-6}$
	$t_2$	13	52	0.082982	$8.67 \times 10^{-7}$	22	65	0.032629	$7.71 \times 10^{-7}$	9	35	0.01903	$4.12 \times 10^{-6}$
	$t_3$	13	52	0.10124	$5.82 \times 10^{-7}$	22	65	0.031087	$7.36 \times 10^{-7}$	9	35	0.025673	$3.93 \times 10^{-6}$
	$t_4$	14	56	0.084077	$8.24 \times 10^{-7}$	24	71	0.058121	$8.02 \times 10^{-7}$	10	39	0.015808	$3.83 \times 10^{-6}$
	$t_5$	15	60	0.13836	$2.73 \times 10^{-7}$	25	74	0.034695	$4.72 \times 10^{-7}$	10	39	0.021954	$5.19 \times 10^{-6}$
	$t_6$	15	60	0.09151	$3.92 \times 10^{-7}$	25	74	0.033872	$6.78 \times 10^{-7}$	10	39	0.017664	$7.45 \times 10^{-6}$
	$t_7$	10	40	0.074314	$7.22 \times 10^{-7}$	23	68	0.04517	$7.08 \times 10^{-7}$	9	35	0.0245	$8.64 \times 10^{-6}$
50,000	$t_1$	14	56	0.48989	$7.84 \times 10^{-7}$	24	71	0.15209	$5.37 \times 10^{-7}$	10	39	0.070466	$2.57 \times 10^{-6}$
	$t_2$	14	56	0.39664	$4.75 \times 10^{-7}$	23	68	0.12867	$7.49 \times 10^{-7}$	9	35	0.059796	$9.21 \times 10^{-6}$
	$t_3$	14	56	1.1025	$3.19 \times 10^{-7}$	23	68	0.26774	$7.15 \times 10^{-7}$	9	35	0.055552	$8.79 \times 10^{-6}$
	$t_4$	15	60	0.37756	$4.51 \times 10^{-7}$	25	74	0.18066	$7.79 \times 10^{-7}$	10	39	0.063615	$8.57 \times 10^{-6}$
	$t_5$	15	60	0.48364	$6.11 \times 10^{-7}$	26	77	0.20521	$4.58 \times 10^{-7}$	11	43	0.066811	$1.96 \times 10^{-6}$
	$t_6$	15	60	0.38215	$8.77 \times 10^{-7}$	26	77	0.1468	$6.58 \times 10^{-7}$	11	43	0.070802	$2.81 \times 10^{-6}$
	$t_7$	11	44	0.29749	$3.97 \times 10^{-7}$	24	71	0.15389	$6.86 \times 10^{-7}$	10	39	0.11456	$3.27 \times 10^{-6}$
100,000	$t_1$	15	60	0.76533	$2.72 \times 10^{-7}$	24	71	0.28556	$7.60 \times 10^{-7}$	10	39	0.15136	$3.63 \times 10^{-6}$
	$t_2$	14	56	0.96545	$6.72 \times 10^{-7}$	24	71	0.30656	$4.60 \times 10^{-7}$	10	39	0.15471	$2.20 \times 10^{-6}$
	$t_3$	14	56	0.71691	$4.51 \times 10^{-7}$	24	71	0.27951	$4.39 \times 10^{-7}$	10	39	0.2006	$2.10 \times 10^{-6}$
	$t_4$	15	60	0.76153	$6.38 \times 10^{-7}$	26	77	0.39044	$4.78 \times 10^{-7}$	11	43	0.18627	$2.04 \times 10^{-6}$
	$t_5$	15	60	0.79539	$8.64 \times 10^{-7}$	26	77	0.57875	$6.48 \times 10^{-7}$	11	43	0.13307	$2.77 \times 10^{-6}$
	$t_6$	16	64	0.82574	$3.04 \times 10^{-7}$	26	77	0.4141	$9.31 \times 10^{-7}$	11	43	0.142	$3.98 \times 10^{-6}$
	$t_7$	11	44	0.52429	$5.60 \times 10^{-7}$	24	71	0.47424	$9.71 \times 10^{-7}$	10	39	0.15694	$4.64 \times 10^{-6}$

### 6. Conclusions

In this paper, we have presented a derivative-free conjugate gradient method for solving the  $\ell_1$ -norm regularization problem by combining the projection technique with the proposed direction in [18]. Unlike the uniform convexity assumption used in establishing the convergence of the method in [18], our convergence was established under the monotonicity and Lipchitz continuity assumption. Our numerical experiments in recovering sparse signals and blurred images indicate efficient and robust behavior of the proposed algorithm. For instance, in recovering sparse signals, the proposed algorithm is faster than the compared algorithms. In addition, it exhibits less iterations and least mean squared error. Moreover, our algorithm is able to restore blurred and noisy images with better quality. This is reflected in the values of the SNR, PSNR, and SSIM. Furthermore, numerical experiments on a set of monotone problems with different initial points and dimensions were reported. Although, applying the proposed method in solving monotone operator equations, the proposed method does not perform well in terms of CPU time. This could be as a result of several computations involved in the method.

**Author Contributions:** Conceptualization, A.H.I.; Formal analysis, A.H.I.; Funding acquisition, P.K.; Methodology, J.A.; Project administration, P.K.; Resources, A.B.M.; Software, A.B.A. and J.A.; Supervision, P.K.; Writing—original draft, A.H.I.; Writing—review and editing, A.B.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Petchra Pra Jom Klao Scholarship, grant number 16/2561.

**Acknowledgments:** The authors are very grateful to Jinkui Liu of the University of Chongqing Three Gorges, Chongqing, China, for his kind offer of the source codes for the signal reconstruction problem. We would also like to thank the anonymous referees for their valuable comments. The authors acknowledge the support provided by the Theoretical and Computational Science (TaCS) Center under Computational and Applied Science for Smart

research Innovation Cluster (CLASSIC), Faculty of Science, KMUTT. The first author was supported by the Petchra Pra Jom Klao Doctoral Scholarship, Academic for Ph.D. Program at KMUTT (Grant No.16/2561).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Donoho, D.L. For most large underdetermined systems of linear equations the minimal  $\ell_1$ -norm solution is also the sparsest solution. *Commun. Pure Appl. Math.* **2006**, *59*, 797–829. [[CrossRef](#)]
2. Lustig, M.; Donoho, D.L.; Santos, J.M.; Pauly, J.M. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306.
3. Candes, E.; Romberg, J. Sparsity and incoherence in compressive sampling. *Inverse Probl.* **2007**, *23*, 969. [[CrossRef](#)]
4. Daubechies, I.; Defrise, M.; De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math. J. Issued Courant Inst. Math. Sci.* **2004**, *57*, 1413–1457. [[CrossRef](#)]
5. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [[CrossRef](#)]
6. Hale, E.T.; Yin, W.; Zhang, Y. A fixed-point continuation method for  $\ell_1$ -regularized minimization with applications to compressed sensing. *CAAM TR07-07 Rice Univ.* **2007**, *43*, 44.
7. Huang, S.; Wan, Z. A new nonmonotone spectral residual method for nonsmooth nonlinear equations. *J. Comput. Appl. Math.* **2017**, *313*, 82–101. [[CrossRef](#)]
8. He, L.; Chang, T.C.; Osher, S. MR image reconstruction from sparse radial samples by using iterative refinement procedures. In Proceedings of the 13th Annual Meeting of ISMRM, Seattle, WA, USA, 6–12 May 2006; Volume 696.
9. Moreau, J.J. Fonctions Convexes Duales et Points Proximaux dans un Espace Hilbertien. 1962. Available online: [http://www.numdam.org/article/BSMF\\_1965\\_\\_93\\_\\_273\\_0.pdf](http://www.numdam.org/article/BSMF_1965__93__273_0.pdf) (accessed on 26 February 2020).
10. Figueiredo, M.A.; Nowak, R.D.; Wright, S.J. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Top. Signal Process.* **2007**, *1*, 586–597. [[CrossRef](#)]
11. Xiao, Y.; Wang, Q.; Hu, Q. Non-smooth equations based method for  $\ell_1$ -norm problems with applications to compressed sensing. *Nonlinear Anal. Theory Methods Appl.* **2011**, *74*, 3570–3577. [[CrossRef](#)]
12. Xiao, Y.; Zhu, H. A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing. *J. Math. Anal. Appl.* **2013**, *405*, 310–319. [[CrossRef](#)]
13. Beale, E.M.L. A derivation of conjugate gradients. In *Numerical Methods for Nonlinear Optimization*; Lootsma, F.A., Ed.; Academic Press: London, UK, 1972.
14. Nazareth, L. A conjugate direction algorithm without line searches. *J. Optim. Theory Appl.* **1977**, *23*, 373–387. [[CrossRef](#)]
15. Zhang, L.; Zhou, W.; Li, D.H. A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **2006**, *26*, 629–640. [[CrossRef](#)]
16. Andrei, N. On three-term conjugate gradient algorithms for unconstrained optimization. *Appl. Math. Comput.* **2013**, *219*, 6316–6327. [[CrossRef](#)]
17. Liu, J.; Li, S. New three-term conjugate gradient method with guaranteed global convergence. *Int. J. Comput. Math.* **2014**, *91*, 1744–1754. [[CrossRef](#)]
18. Tang, C.; Li, S.; Cui, Z. Least-squares-based three-term conjugate gradient methods. *J. Inequalities Appl.* **2020**, *2020*, 27. [[CrossRef](#)]
19. Fletcher, R.; Reeves, C.M. Function minimization by conjugate gradients. *Comput. J.* **1964**, *7*, 149–154. [[CrossRef](#)]
20. Solodov, M.V.; Svaiter, B.F. A new projection method for variational inequality problems. *SIAM J. Control. Optim.* **1999**, *37*, 765–776. [[CrossRef](#)]
21. Liu, J.; Feng, Y. A derivative-free iterative method for nonlinear monotone equations with convex constraints. *Numer. Algorithms* **2018**, *82*, 245–262. [[CrossRef](#)]
22. Liu, J.; Xu, J.; Zhang, L. Partially symmetrical derivative-free Liu–Storey projection method for convex constrained equations. *Int. J. Comput. Math.* **2019**, *96*, 1787–1798. [[CrossRef](#)]

23. Ibrahim, A.H.; Garba, A.I.; Usman, H.; Abubakar, J.; Abubakar, A.B. Derivative-free RMIL conjugate gradient algorithm for convex constrained equations. *Thai J. Math.* **2019**, *18*, 212–232.
24. Abubakar, A.B.; Rilwan, J.; Yimer, S.E.; Ibrahim, A.H.; Ahmed, I. Spectral three-term conjugate descent method for solving nonlinear monotone equations with convex constraints. *Thai J. Math.* **2020**, *18*, 501–517.
25. Ibrahim, A.H.; Kumam, P.; Abubakar, A.B.; Jirakitpuwapat, W.; Abubakar, J. A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing. *Heliyon* **2020**, *6*, e03466. [[CrossRef](#)] [[PubMed](#)]
26. Abubakar, A.B.; Kumam, P.; Awwal, A.M. Global convergence via descent modified three-term conjugate gradient projection algorithm with applications to signal recovery. *Results Appl. Math.* **2019**, *4*, 100069. [[CrossRef](#)]
27. Abubakar, A.B.; Kumam, P.; Awwal, A.M. An inexact conjugate gradient method for symmetric nonlinear equations. *Comput. Math. Methods* **2019**, *1*, e1065. [[CrossRef](#)]
28. Pang, J.S. Inexact Newton methods for the nonlinear complementarity problem. *Math. Program.* **1986**, *36*, 54–71. [[CrossRef](#)]
29. Zhou, W.; Li, D. Limited memory BFGS method for nonlinear monotone equations. *J. Comput. Math.* **2007**, *25*, 89–96.
30. Liu, J.; Li, S. A projection method for convex constrained monotone nonlinear equations with applications. *Comput. Math. Appl.* **2015**, *70*, 2442–2453. [[CrossRef](#)]
31. Wan, Z.; Guo, J.; Liu, J.; Liu, W. A modified spectral conjugate gradient projection method for signal recovery. *Signal Image Video Process.* **2018**, *12*, 1455–1462. [[CrossRef](#)]
32. Kim, S.; Koh, K.; Lustig, M.; Boyd, S.; Gorinevsky, D. A method for large-scale  $\ell_1$ -regularized least squares. *IEEE J. Sel. Top. Signal Process.* **2007**, *1*, 606–617. [[CrossRef](#)]
33. Abubakar, A.B.; Kumam, P.; Mohammad, H.; Awwal, A.M.; Sitthithakerngkiet, K. A Modified Fletcher–Reeves Conjugate Gradient Method for Monotone Nonlinear Equations with Some Applications. *Mathematics* **2019**, *7*, 745. [[CrossRef](#)]
34. Bovik, A.C. *Handbook of Image and Video Processing*; Academic Press: Cambridge, MA, USA, 2010.
35. Lajevardi, S.M. Structural similarity classifier for facial expression recognition. *Signal Image Video Process.* **2014**, *8*, 1103–1110. [[CrossRef](#)]
36. La Cruz, W.; Martínez, J.; Raydan, M. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Math. Comput.* **2006**, *75*, 1429–1448. [[CrossRef](#)]
37. La Cruz, W. A spectral algorithm for large-scale systems of nonlinear monotone equations. *Numer. Algorithms* **2017**, *76*, 1109–1130. [[CrossRef](#)]
38. Wang, C.; Wang, Y.; Xu, C. A projection method for a system of nonlinear monotone equations with convex constraints. *Math. Methods Oper. Res.* **2007**, *66*, 33–46. [[CrossRef](#)]
39. Bing, Y.; Lin, G. An efficient implementation of Merrill’s method for sparse or partially separable systems of nonlinear equations. *SIAM J. Optim.* **1991**, *1*, 206–221. [[CrossRef](#)]
40. Yu, G.; Niu, S.; Ma, J. Multivariate spectral gradient projection method for nonlinear monotone equations with convex constraints. *J. Ind. Manag. Optim.* **2013**, *9*, 117–129. [[CrossRef](#)]
41. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–213. [[CrossRef](#)]

