Research Article

A Modified Conjugate Gradient Method for Solving Large-Scale Nonlinear Equations

Hongbo Guan and Sheng Wang

School of Mathematics, Physics and Energy Engineering, Hunan Institute of Technology, Hengyang 421002, China

Correspondence should be addressed to Sheng Wang; wangshengmath@163.com

Received 26 March 2021; Revised 24 May 2021; Accepted 5 June 2021; Published 24 June 2021

Academic Editor: Zhifeng Dai

Copyright © 2021 Hongbo Guan and Sheng Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose a modified Polak–Ribière–Polyak (PRP) conjugate gradient method for solving large-scale nonlinear equations. Under weaker conditions, we show that the proposed method is globally convergent. We also carry out some numerical experiments to test the proposed method. The results show that the proposed method is efficient and stable.

1. Introduction

Solving nonlinear equations is an important problem which appears in various models of science and engineering such as computer vision, computational geometry, signal processing, computational chemistry, and robotics. More specifically, the subproblems in the generalized proximal algorithms with Bergman distances is a monotone nonlinear equations [1], and $l_1$-norm regularized optimization problems can be reformulated as monotone nonlinear equations [2]. Due to its wide applications, the studies in the numerical methods for solving the monotone nonlinear equations have received much attention [3–10]. In this paper, we are interested in the numerical methods for solving monotone nonlinear equations with convex constraints:

$$F(x) = 0, \quad x \in S, \tag{1}$$

where $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuous function and $S$ is a nonempty, closed, and convex set. The monotonicity of the mapping $F$ means that

$$(F(x) - F(y))^T (x - y) \geq 0, \quad \forall x, y \in \mathbb{R}^n. \tag{2}$$

The methods for solving monotone nonlinear equations (1) are closely relevant to the methods for solving the following optimization problems:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{3}$$

Notice that if $f(x)$ is strictly convex, then $\nabla f(x)$ is strictly monotone which means $(\nabla f(x) - \nabla f(y))^T (x - y) \geq 0$. It is well known that the strictly convex function exists a unique solution $x^*$, satisfying $\nabla f(x^*) = 0$. To sum up, if there is a convex function $f(x)$ satisfying $\nabla f(x) = F(x)$, then solving the optimization problems (3) is equivalent to solving monotone nonlinear equations (1). So, a natural idea to solve monotone nonlinear equations (1) is to use the existing efficient methods for solving optimization problems (3). There are many methods for solving optimization problems (3), such as the Newton method, quasi-Newton method, trust region method, and conjugate gradient method. Among these methods, the conjugate gradient method is a very effective method for solving optimization problems (3) due to their simplicity and low storage. A conjugate gradient method generates a sequence of iterates:

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{4}$$

where $\alpha_k$ is the step length and direction $d_k$ is defined by

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{5}$$

where $\beta_k$ is a parameter and $g_k$ is the gradient of the objective function $f(x)$. The choice of $\beta_k$ determines different
conjugate gradient methods [11–17]. We are interested in the PRP conjugate gradient method in which the parameter \( \beta_k \) is defined by
\[
\beta_k^{\text{PRP}} = \frac{\nabla f(x_k)^T \nabla f(x_{k-1})}{\|
abla f(x_{k-1})\|^2}.
\]
(6)
where \( y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}) \). Based on the idea of [18, 19], Zhang et al. [20] proposed a new modified nonlinear PRP method in which \( \beta_k \) is defined by
\[
\beta_k^{\text{NPRP}} = \frac{y_{k-1}^T y_{k-1}}{z_{k-1}^2} + \eta \frac{\|
abla f(x_{k-1})\|^2}{z_{k-1}^2} g_k^T d_{k-1},
\]
(7)
where \( z_{k-1} = \max\{t\|d_{k-1}\|, \|
abla f(x_{k-1})\|^2\}, t > 0 \) and \( \eta > (1/4) \) are two constants. There is a mistake about the definition of \( \beta_k^{\text{NPRP}} \). By this definition, we will not be able to prove Lemma 1 in [20]. It should be
\[
\beta_k^{\text{NPRP}} = \frac{y_{k-1}^T y_{k-1}}{z_{k-1}^2} + \eta \frac{\|
abla f(x_{k-1})\|^2}{z_{k-1}^2} g_k^T d_{k-1}.
\]
(8)

There are many conjugate gradient methods for solving nonlinear equations (1). Zhang and Zhou [4] proposed a spectral gradient method by combining the modified spectral gradient and projection method, which can be applied to solve nonsmooth equations. Xiao and Zhou [10] extended the CG-DESCENT method to solve large-scale nonlinear monotone equations and extended this method to decode a sparse signal in compressive sensing. Dai and Zhu [21] proposed a derivative-free method for solving large-scale nonlinear monotone equations and proposed a new line search for the derivative-free method. Other related works can be found [3, 5–8, 10, 22–30]. In this paper, we combined the projection method [3], the modified nonlinear PRP conjugate gradient method for unconstrained optimization [20] and the iterative method [10] and proposed a modified nonlinear conjugate gradient method for solving large-scale nonlinear monotone equations with convex constrains.

This paper is organized as follows. In Section 2, we propose a modified nonlinear PRP method for solving monotone nonlinear equations with convex constrains. Under reasonable conditions, we prove its global convergence. In Section 3, we make some improvement to the proposed method and give the convergence theorem of the improved method. In Section 4, we do some numerical experiments to test the proposed methods. The results show that our methods are efficient and promising. Furthermore, we use the proposed methods to solve practical problems in compressed sensing.

2. A Modified Nonlinear PRP Method

In this section, we develop a modified nonlinear PRP method for solving the nonlinear equations with convex constrains. Based on the modified nonlinear PRP method [20], we now introduce our method for solving (1). Inspired by (8), we define \( d_k \) as
\[
d_k = \begin{cases} -F_k, & \text{if } k = 0, \\ -F_k + \beta_k^{\text{NPRP}} d_{k-1}, & \text{if } k \geq 1, \end{cases}
\]
(9)
where \( y_{k-1} = F_k - F_{k-1} \). The parameter \( \beta_k^{\text{NPRP}} \) is computed as
\[
\beta_k^{\text{NPRP}} = \frac{F_k^T y_{k-1}}{z_{k-1}} - \eta \frac{\|
abla f(x_{k-1})\|^2}{z_{k-1}^2} F_k^T d_{k-1},
\]
(10)
where \( z_{k-1} = \max\{t\|d_{k-1}\|, \|
abla f(x_{k-1})\|^2\}, t > 0 \) and \( \eta > (1/4) \) are two constants.

The lemma below shows a good property of \( d_k \). The steps of the method are given in Algorithm 1.

**Lemma 1.** Let \( \{d_k\} \) be generated by Algorithm 1. If \( z_{k-1} \neq 0 \), then there exists a constant \( c > 0 \) such that
\[
F_k^T d_k \leq -c \|F_k\|^2.
\]
(11)

**Proof.** For \( k = 0 \), we have
\[
F_0^T d_0 \leq -\|F_0\|^2.
\]
(12)

For \( k \geq 1 \), we obtain
\[
F_k^T d_k = F_k^T (-F_k + \beta_k^{\text{NPRP}} d_{k-1}),
\]
\[
= -\|F_k\|^2 + \left( \frac{F_k^T y_{k-1}}{z_{k-1}} - \eta \frac{\|
abla f(x_{k-1})\|^2}{z_{k-1}} F_k^T d_{k-1} \right) F_k^T d_{k-1},
\]
\[
= -\|F_k\|^2 + \frac{z_{k-1} F_k^T d_{k-1} F_k^T y_{k-1} - \eta \|
abla f(x_{k-1})\|^2 (F_k^T d_{k-1})^2}{z_{k-1}}
\]
(13)

Denote
\[
u_k = \frac{1}{\sqrt{2\eta}} z_{k-1},
\]
(14)

\[
u_k = \sqrt{2\eta} F_k^T d_{k-1} y_{k-1}.
\]

Then, we obtain
\[
F_k^T d_k = -\|F_k\|^2 + \frac{\nu_k^T y_k - (1/2) \|y_k\|^2}{z_{k-1}}
\]
\[
= -\left( 1 - \frac{1}{4\eta} \right) \|F_k\|^2 + \frac{\nu_k^T y_k - (1/2) \left( \|\nu_k\|^2 + \|y_k\|^2 \right)}{z_{k-1}}
\]
\[
\leq -\left( 1 - \frac{1}{4\eta} \right) \|F_k\|^2.
\]
(15)

Let \( c = -(1 - (1/4\eta)) \); then, inequality (11) is satisfied.
Next, we establish the global convergence of the proposed method. Without specification, we always suppose
that the solution set of equation (1) is nonempty and the following assumption holds.

**Assumption 1**

(i) The mapping $F$ is Lipchitz continuous, and it means that the mapping $F$ satisfies

$$
\|F(x) - F(y)\| \leq L \|x - y\|, \quad \forall x, y \in S. \tag{16}
$$

(ii) The projection operator $P_S[\cdot]$ is nonexpansive, i.e.,

$$
\|P_S[x] - P_S[y]\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{17}
$$

**Lemma 2.** Suppose that Assumption 1 holds and $x^*$ is a solution of (1), and the sequence \{\(x_k\)\} and \{\(x_{k+1}\)\} are generated by Algorithm 1. Then, the sequence \{\(x_k\)\}, \{\(x_{k+1}\)\}, and \{\(F_k\)\} are bounded.

**Proof.** We first show that \{\(x_k\)\} is bounded. From the monotonicity of function $F$, we have

$$
F(\bar{x}_{k+1})^T (x_k - x^*) = F(\bar{x}_{k+1})^T (x_k - \bar{x}_{k+1}) + (F(\bar{x}_{k+1}) - F(x^*))^T (\bar{x}_{k+1} - x^*),
$$

$$
\geq F(\bar{x}_{k+1})^T (x_k - \bar{x}_{k+1}). \tag{18}
$$

It is easy to see that

$$
\|x_{k+1} - x^*\|^2 = \|P_S[x_k - \alpha_k F(\bar{x}_{k+1})] - P_S[x^*]\|^2
\leq \|x_k - \alpha_k F(\bar{x}_{k+1}) - x^*\|^2
\leq \|x_k - x^*\|^2 - 2\alpha_k F(\bar{x}_{k+1})^T (x_k - \bar{x}_{k+1}) + \alpha_k^2 \|F(\bar{x}_{k+1})\|^2
= \|x_k - x^*\|^2 - \frac{[F(\bar{x}_{k+1})^T (x_k - \bar{x}_{k+1})]^2}{\|F(\bar{x}_{k+1})\|^2}
\leq \|x_k - x^*\|^2. \tag{19}
$$

The last inequality implies

$$
\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 \leq \cdots \|x_0 - x^*\|^2. \tag{20}
$$

It obviously that the sequences \{\(x_k\)\} is bounded, i.e., there is a constant $M > 0$ such that

$$
\|x_k\| \leq M, \quad \forall k > 0. \tag{21}
$$

Next, we show that \{\(F_k\)\} is bounded. Since $F(x)$ is Lipchitz continuous, we obtain

$$
F(\bar{x}_{k+1})^T (x_k - \bar{x}_{k+1}) = -t_k F(\bar{x}_{k+1})^T d_k \geq \sigma t_k^2 \|d_k\|^2 = \sigma \|x_k - \bar{x}_{k+1}\|^2,
$$

$$
F(\bar{x}_{k+1})^T (x_k - \bar{x}_{k+1}) = (F(\bar{x}_{k+1}) - F(x_k))^T (x_k - \bar{x}_{k+1}) + F(x_k)^T (x_k - \bar{x}_{k+1}), \tag{24}
$$

$$
\leq \|F_k\| \|x_k - \bar{x}_{k+1}\| \leq A \|x_k - \bar{x}_{k+1}\|. \tag{25}
$$

So, the following inequality holds:
We first prove the right side of inequality (26). For \( k = 0 \), from (9), we have
\[
\|d_k\| = \|F_k\|.
\]
(27)

For \( k \geq 1 \), by the definition of \( \beta_k^{\text{NPRP}} \), we obtain
\[
\|\beta_k^{\text{NPRP}}\| = \frac{\langle y_{k-1}, y_{k-1}\rangle}{\eta \|F_{k-1}\|^2} = \frac{\|F_k\|^{2}}{\|F_{k-1}\|^2} \|y_{k-1}\| \|F_{k-1}\| \|d_{k-1}\|
\]
(28)

By the definition of \( d_k \) (9) and the last inequality, we obtain
\[
\|d_k\| \leq \|F_k\| + \beta_k^{\text{NPRP}} \|d_{k-1}\| \leq \left(1 + \frac{2LM(t + 2\eta LM)}{t^2}\right) \|F_k\|.
\]
(29)

Let \( \gamma = 1 + (2LM(t + 2\eta LM)/t^2) \); then, we have
\[
\|d_k\| \leq \gamma \|F_k\|.
\]

Now, we turn to prove the left side of the inequality. It follows from (11) that
\[
c \|F_k\|^2 - F_k^T d_k \leq \|F_k\| \|d_k\|.
\]
(30)

Therefore, we have
\[
\|d_k\| \geq c \|F_k\|. \tag{31}
\]

**Lemma 4.** Suppose Assumption 1 holds; then, the step length \( t_k \) satisfies
\[
t_k \geq \min \left\{ \xi, \frac{pc}{(L + \sigma)\gamma^2} \right\}. \tag{32}
\]

**Proof.** If the Algorithm 1 terminates in a finite number of steps, then there is a \( k \in \mathbb{R} \) such that \( x_k \) is a solution of equation (1) and \( \|F_k\| = 0 \). From now on, we assume that \( \|F_k\| \neq 0 \), for any \( k \). It is easy to see that \( d_k \neq 0 \) from (11). If \( t_k \neq \xi \), by the line search process, we know that \( \rho^{-1}t_k \) does not satisfy Algorithm 1, that is,
\[
-F(x_k + \rho^{-1}t_k d_k) = \lambda_k d_k \leq \sigma \rho^{-1}t_k \|d_k\|^2. \tag{33}
\]

It follows from (11) and Assumption 1 that
\[
c \|F_k\|^2 - F_k^T d_k \leq -F_k^T d_k \leq \sigma \rho^{-1}t_k \|d_k\|^2.
\]
Let \( \gamma = 1 + (2LM(t + 2\eta LM)/t^2) \); then, we have
\[
\|d_k\| \leq \gamma \|F_k\|.
\]

Now, we turn to prove the left side of the inequality. It follows from (11) that
\[
c \|F_k\|^2 - F_k^T d_k \leq \|F_k\| \|d_k\|. \tag{30}
\]

Therefore, we have
\[
\|d_k\| \geq c \|F_k\|. \tag{31}
\]

**Algorithm 1:** Modified NPRP method.

**Lemma 3.** Suppose that Assumption 1 holds, and the sequence \( \{x_k\} \) and \( \{F_k\} \) is generated by Algorithm 1. Then, there exists a constant \( \gamma > 1 \) such that
\[
c \|F_k\| \leq \|d_k\| \leq \gamma \|F_k\|, \quad \forall k \geq 0. \tag{26}
\]

**Proof.** We first prove the right side of inequality (26). For \( k = 0 \), from (9), we have
\[
\|d_k\| = \|F_k\|. \tag{27}
\]

For \( k \geq 1 \), by the definition of \( \beta_k^{\text{NPRP}} \) and (21), we obtain
\[
\|\beta_k^{\text{NPRP}}\| = \frac{\|F_k\|^{2}}{\|F_{k-1}\|^2} \|y_{k-1}\| \|F_{k-1}\| \|d_{k-1}\| \|F_k\| \|d_{k-1}\| \leq \frac{2LM\|F_k\|^{2}}{\|F_{k-1}\|^2} \|y_{k-1}\| \|F_{k-1}\| \|d_{k-1}\| \|F_k\| \|d_{k-1}\|
\]
(28)

By the definition of \( d_k \) (9) and the last inequality, we obtain
\[
\|d_k\| \leq \|F_k\| + \beta_k^{\text{NPRP}} \|d_{k-1}\| \leq \left(1 + \frac{2LM(t + 2\eta LM)}{t^2}\right) \|F_k\|.
\]
(29)

Let \( \gamma = 1 + (2LM(t + 2\eta LM)/t^2) \); then, we have
\[
\|d_k\| \leq \gamma \|F_k\|.
\]

Now, we turn to prove the left side of the inequality. It follows from (11) that
\[
c \|F_k\|^2 - F_k^T d_k \leq \|F_k\| \|d_k\|. \tag{30}
\]

Therefore, we have
\[
\|d_k\| \geq c \|F_k\|. \tag{31}
\]
\[ \| x_{k+1} - x^* \|^2 \leq \| x_k - x^* \|^2 - \frac{\| F(x_{k+1}) \|^2}{\| F(x_{k+1}) \|^2} \]

\[ \leq \| x_k - x^* \|^2 - \frac{\sigma^2 \| x_k - x_{k+1} \|^4}{\| F(x_{k+1}) \|^2} \]

(38)

Since the function \( F(x) \) is continuous, and the sequence \( \{ x_k \} \) is bounded, so the sequence \( \{ \| F(x_k) \| \} \) is bounded. That is, for all \( k \geq 0 \), there exists a positive constant \( B > 0 \), such that \( \| F(x_k) \| \leq B \). Then, we obtain

\[ \sum_{k=0}^{\infty} \| x_k - x_{k+1} \|^4 \leq \sum_{k=0}^{\infty} B^2 \sigma^2 \left( \| x_k - x^* \|^2 - \| x_{k+1} - x^* \|^2 \right) < +\infty. \]

(39)

So, we have

\[ \lim_{k \to \infty} t_k \| d_k \| = \lim_{k \to \infty} \| x_k - x_{k+1} \| = 0. \]

(40)

**Theorem 1.** Suppose that Assumption 1 holds. The sequence \( \{ x_k \} \) is generated by Algorithm 1. Then, we have

\[ \lim_{k \to \infty} \inf \| F_k \| = 0. \]

(41)

**Proof.** Suppose that (41) does not hold; then, there exists \( \varepsilon > 0 \) such that, for any \( k \geq 0 \),

\[ \| F_k \| \geq \varepsilon. \]

(42)

From (26) and the last inequality, it is easy to see

\[ c \varepsilon \leq \| d_k \| \leq \gamma A. \]

(43)

From (41) and (42), we obtain

\[ t_k \| d_k \| \geq \min \left\{ \frac{\xi}{\gamma (L + \sigma)} \right\} \| d_k \| \geq \min \left\{ \frac{\rho c^2 \varepsilon}{\gamma^2 (L + \sigma)} \right\}. \]

(44)

The last inequality yields a contradiction with (37), so (41) is satisfied. \( \square \)

### 3. An Improvement

In this section, we make some improvement to the modified nonlinear PRP method proposed in Section 2. In Algorithm 1, we take the step length \( \alpha_k = (F(x_{k+1})^T (x_k - x_{k+1}))/\| F(x_{k+1}) \|^2 \). Is there a better choice for \( \alpha_k \)? This is our purpose to improve Algorithm 1. Under the condition of ensuring the convergence of the algorithm and the related good properties and results, we improve Algorithm 1 in order to get better numerical results.

From Algorithm 1, to make the inequality \( \| x_{k+1} - x^* \| \leq \| x_k - x^* \| \) hold, we only need to satisfy

\[ \phi(\alpha) = -2\alpha_k F(x_{k+1})^T (x_k - x_{k+1}) + \alpha_k^2 \| F(x_{k+1}) \|^2 \leq 0. \]

(45)

By solving the last inequality, we have

\[ 0 \leq \alpha_k \leq 2 \frac{F(x_{k+1})^T (x_k - x_{k+1})}{\| F(x_{k+1}) \|^2}. \]

(46)

It is easy to see that \( \alpha_k = (F(x_{k+1})^T (x_k - x_{k+1}))/\| F(x_{k+1}) \|^2 \) is the minimum point of the function \( \phi(\alpha) \). This is the reason why Algorithm 1 takes \( \alpha_k = (F(x_{k+1})^T (x_k - x_{k+1}))/\| F(x_{k+1}) \|^2 \). Under reasonable conditions, we hope to get a large step length than Algorithm 1. So, we obtain

\[ \frac{F(x_{k+1})^T (x_k - x_{k+1})}{\| F(x_{k+1}) \|^2} \leq \alpha_k \leq 2 \frac{F(x_{k+1})^T (x_k - x_{k+1})}{\| F(x_{k+1}) \|^2}. \]

(47)

Based on the above arguments, we propose an improved algorithm of Algorithm 1. In the improved algorithm, we make the step length:

\[ \alpha_k = \theta \frac{F(x_{k+1})^T (x_k - x_{k+1})}{\| F(x_{k+1}) \|^2}, \quad \theta \in [1,2]. \]

(48)

Similar to the proof of Theorem 1, we have the following results.

**Theorem 2.** Suppose that Assumption 1 holds. The sequence \( \{ x_k \} \) is generated by Algorithm 2; then, we have

\[ \lim_{k \to \infty} \inf \| F_k \| = 0. \]

(49)

The iterative process of the improved method is stated as follows.

### 4. Numerical Results

In this section, we do some numerical experiments to test the performance of the proposed methods. We implemented our methods in MATLAB R2020b and run the codes on a personal computer with 2.3 GHz CPU and 16 GB RAM.

We first solve Problems 1 and 2.

**Problem 1** (see [8]). The mapping \( F \) is taken as \( F(x) = (f_1(x), f_2(x), \ldots, f_n(x))^T \), where

\[ f_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n, \]

\[ S = \mathbb{R}^n_+ = \{ x \in \mathbb{R}^n | x_i \geq 0, i = 1, 2, \ldots, n \}. \]

(50)

**Problem 2** (see [4]). The mapping \( F \) is taken as \( F(x) = (f_1(x), f_2(x), \ldots, f_n(x))^T \), where

\[ f_i(x) = 2x_i - \sin|x_i|, \quad i = 1, 2, \ldots, n, \]

and \( S = \mathbb{R}^n_+ \).
We set $\bar{t} = 1, t = 1, \rho = 0.95, \eta = 1, \sigma = 0.0001$, and $\theta = 2$. The stopping criterion of the algorithm is set to $\|F(x_k)\| < 10^{-5}$ or the number of iteration reach to 500. The latter case means that the method is a failure for the test problems. We test both problems with the dimensions of variables $n = 1000, 10000, 100000, 1000000,$ and $1000000$. Start from different initial points, and we list all results in Tables 1 and 2. We compared the performance of the proposed methods with the classical Newton method and an efficient algorithm CGD [10] in the total number of iterations as well as the computational time. The meaning of each column is given below.

- ‘Init’: the initial point
- ‘n’: the dimension of the problem
- ‘Iter’: the total number of iterations
- ‘Time’: the CPU time (in seconds) used for the method
- ‘NM’: the Newton method
- ‘CGD’: the conjugate gradient method in [10]
- ‘MNPRP’: the modified nonlinear PRP method
- ‘IMNPRP’: the improved nonlinear PRP method

The results in Tables 1 and 2 show that our methods performs very well both in the number of iterations and CPU time. The IMNPRP performs best among these methods. It is worth noting that the number of iterations does not increase significantly as $n$ increases. Hence, the proposed method is very suitable for solving large-scale problems. Because of the lack of memory, the dimension of the problems solved by the Newton method is no more than 100,000.
The following example is a signal reconstruction problem from compressed sensing.

Problem 3 (see [10]). Consider a typical compressive sensing scenario, where we aim to reconstruct a length-$n$ sparse signal form $m$ observations ($m \ll n$). In this test, the measurement $b$ contains noise:

$$b = Ax + \omega,$$  \hspace{1cm} (52)

where $\omega$ is the Gaussian noise distributed as $N(0, \sigma^2 I)$ with $\sigma^2 = 10^{-4}$. The random $A$ is the Gaussian matrix which is

---

**Table 2: Comparison on Problem 2.**

| Init | $n$ | NM | CGD | MNPRP | IMNPRP |
|------|-----|----|-----|-------|--------|
| $(1,1,\ldots,1)^T$ | 1000 | 3  | 0.0187 | 11 | 0.0006 | 4 | 0.0002 | 1 | 0.0001 |
|       | 10000 | 3  | 1.1675 | 12 | 0.0033 | 4 | 0.0009 | 1 | 0.0004 |
|       | 100000 | Out of memory | 12 | 0.0245 | 4 | 0.0058 | 1 | 0.0028 |
|       | 1000000 | Out of memory | 13 | 0.4156 | 5 | 0.1036 | 1 | 0.0350 |
|       | 10000000 | Out of memory | 14 | 4.3595 | 5 | 1.1379 | 1 | 0.3650 |
| $(2,2,\ldots,2)^T$ | 1000 | 4  | 0.0207 | 11 | 0.0005 | 5 | 0.0003 | 1 | 0.0002 |
|       | 10000 | 4  | 1.6028 | 12 | 0.0036 | 5 | 0.0016 | 1 | 0.0009 |
|       | 100000 | Out of memory | 12 | 0.0312 | 6 | 0.0124 | 1 | 0.0061 |
|       | 1000000 | Out of memory | 13 | 0.4153 | 6 | 0.1628 | 1 | 0.0771 |
|       | 10000000 | Out of memory | 14 | 4.5275 | 6 | 1.7940 | 1 | 0.8300 |
| $((1/2),(1/2),\ldots,(1/2))^T$ | 1000 | 3  | 0.0226 | 11 | 0.0009 | 4 | 0.0002 | 1 | 0.0001 |
|       | 10000 | 3  | 1.1711 | 11 | 0.0030 | 4 | 0.0007 | 1 | 0.0002 |
|       | 100000 | Out of memory | 12 | 0.0248 | 4 | 0.0049 | 1 | 0.0015 |
|       | 1000000 | Out of memory | 13 | 0.3998 | 5 | 0.0936 | 1 | 0.0236 |
|       | 10000000 | Out of memory | 14 | 4.3801 | 5 | 0.9911 | 1 | 0.2697 |
| $(2,2,\ldots,2)^T$ | 1000 | 3  | 0.0188 | 11 | 0.0005 | 4 | 0.0002 | 1 | 0.0001 |
|       | 10000 | 3  | 1.1775 | 11 | 0.0029 | 4 | 0.0008 | 1 | 0.0003 |
|       | 100000 | Out of memory | 12 | 0.0234 | 4 | 0.0044 | 1 | 0.0023 |
|       | 1000000 | Out of memory | 13 | 0.3964 | 4 | 0.0828 | 1 | 0.0320 |
|       | 10000000 | Out of memory | 14 | 4.2691 | 5 | 1.1491 | 1 | 0.3430 |

Figure 1: The original signal (top), the measurement (bottom), and the reconstructed signals by SGCS, CGD, MNPRP, and IMNPRP.
generated by command `randn(m,n)` in Matlab. The merit function

\[
f(x) = \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2,
\]

where the value \(\tau\) is forced to decrease as the measure in [31]. The iterative process starts at the measurement image, i.e., \(x_0 = A^Tb\), and terminates when the relative change between successive iterates falls below \(10^{-4}\), i.e.,

\[
\text{Tol} = \frac{\|f_k - f_{k-1}\|}{\|f_{k-1}\|} < 10^{-4},
\]

where \(f_k\) denotes the function value at iteration \(x_k\). By the discussion in [10], we know that the \(l_1\)-norm problem can transformed a monotone nonlinear equation. Hence, it can be solved by Algorithms 1 and 2.

Due to the storage limitations of the PC, we test a small size signal with \(n = 2^{11}\) and \(m = 2^n\), and the original contains \(2^6\) randomly nonzero elements. The quality of restoration is measured by the mean of squared error (MSE) to the original signal \(\bar{x}\), that is,

\[
\text{MSE} = \frac{1}{n} \|\bar{x} - x\|^2,
\]

where \(x\) is the restored signal. We take the parameters \(\xi = 10, \sigma = 10^{-4}\), and \(\rho = 0.5\) in CGD, MNPRP, and IMNPRP.

In order to test the effectiveness of the proposed methods, we compare the proposed methods with the CGD method [10] and the solver SGCS which is specially designed to solve monotone equations for recovering a large sparse signal in compressive sensing. The results are listed in Figures 1 and 2.

It can be seen from Figures 1 and 2 that all methods have recovered the original sparse signal almost exactly. Among these methods, the IMNPRP method performs best.

5. Conclusions

In this paper, a modified conjugate gradient method and its improved method are proposed for solving the large-scale nonlinear equations. Under some assumptions, global convergence of the proposed methods are established.
Numerical results show that the proposed methods are very efficient and competitive.

Data Availability
All data generated or analysed during this study are included within the article.

Conflicts of Interest
The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments
This research was funded by the Education Department of Hunan Province (Grant no. 20C0559).

References
[1] N. A. Iusem and V. M. Solodov, “Newton-type methods with generalized distances for constrained optimization,” Optimization, vol. 41, no. 3, pp. 257–278, 1997.
[2] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems,” IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 4, pp. 586–597, 2007.
[3] M. V. Solodov and B. F. Svaiter, “A globally convergent inexact Newton method for systems of monotone equations,” in Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing-Methods, pp. 355–369, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
[4] L. Zhang and W. Zhou, “Spectral gradient projection method for solving nonlinear monotone equations,” Journal of Computational and Applied Mathematics, vol. 196, no. 2, pp. 478–484, 2006.
[5] G. Zhou and K. C. Toh, “Superlinear convergence of a Newton-type Algorithm for monotone equations,” Journal of Optimization Theory and Applications, vol. 125, no. 1, pp. 205–221, 2005.
[6] W.-J. Zhou and D.-H. Li, “A globally convergent BFGS method for nonlinear monotone equations without any merit functions,” Mathematics of Computation, vol. 77, no. 264, pp. 2231–2240, 2008.
[7] W. Zhou and D. Li, “Limited memory BFGS method for nonlinear monotone equations,” Journal of Computational Mathematics, vol. 25, pp. 89–96, 2007.
[8] C. Wang, Y. Wang, and C. Xu, “A projection method for a system of nonlinear monotone equations with convex constraints,” Mathematical Methods of Operations Research, vol. 66, no. 1, pp. 33–46, 2007.
[9] Z. Yu, J. Lin, J. Sun, Y. Xiao, L. Liu, and Z. Li, “Spectral gradient projection method for monotone nonlinear equations with convex constraints,” Applied Numerical Mathematics, vol. 59, no. 10, pp. 2416–2423, 2009.
[10] Y. Xiao and H. Zhu, “A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing,” Journal of Mathematical Analysis and Applications, vol. 405, no. 1, pp. 310–319, 2013.
[11] E. Polak and G. Ribiere, “Note sur la convergence de méthodes de directions conjuguées,” Revue française d’informatique et de recherche opérationnelle. Série rouge, vol. 3, no. 16, pp. 35–43, 1969.
[12] B. T. Polyak, “The conjugate gradient method in extremal problems,” USSR Computational Mathematics and Mathematical Physics, vol. 9, no. 4, pp. 94–112, 1969.
[13] M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” Journal of Research of the National Bureau of Standards, vol. 49, no. 6, pp. 409–436, 1952.
[14] Y. Liu and C. Storey, “Efficient generalized conjugate gradient algorithms, part I: theory,” Journal of Optimization Theory and Applications, vol. 69, no. 1, pp. 129–137, 1991.
[15] Y. H. Dai and Y. Yuan, “A nonlinear conjugate gradient method with a strong global convergence property,” SIAM Journal on Optimization, vol. 10, no. 1, pp. 177–182, 1999.
[16] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” The Computer Journal, vol. 7, no. 2, pp. 149–154, 1964.
[17] R. Fletcher, “Practical methods of optimization,” in Unconstrained Optimization Wiley, New York, NJ, USA, 1980.
[18] G. Yu, L. Guan, and W. Chen, “Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization,” Optimization Methods and Software, vol. 23, no. 2, pp. 275–293, 2008.
[19] M. Li and A. Qi, “Some sufficient descent conjugate gradient methods and their global convergence,” Computational and Applied Mathematics, vol. 33, no. 2, pp. 333–347, 2014.
[20] M. Zhang, Y. Zhou, and S. Wang, “A modified nonlinear conjugate gradient method with the armijo line search and its application,” Mathematical Problems in Engineering, vol. 2020, Article ID 6210965, 14 pages, 2020.
[21] Z. Dai and H. Zhu, “A modified hestenes-stiefel-type derivative-free method for large-scale nonlinear monotone equations,” Mathematics, vol. 8, no. 168, 2020.
[22] N. Andrei, “A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization,” Applied Mathematics Letters, vol. 20, no. 6, pp. 645–650, 2007.
[23] L. Zheng, L. Yang, and Y. Liang, “A conjugate gradient projection method for solving equations with convex constraints,” Journal of Computational and Applied Mathematics, vol. 375, Article ID 112781, 2020.
[24] A. H. Ibrahim, P. Kumam, A. B. Abubakar, W. Jirakitpuwapat, and J. Abubakar, “A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing,” Heliyon, vol. 6, no. 3, Article ID e03466, 2020.
[25] A. M. Awwal, P. Abubakar, and A. B. Abubakara, “A modified conjugate gradient method for monotone nonlinear equations with convex constraints,” Applied Numerical Mathematics, vol. 145, pp. 507–520, 2019.
[26] W. La Cruz, J. M. Martinez, and M. Raydan, “Spectral residual method without gradient information for solving large-scale nonlinear systems,” Mathematics of Computation, vol. 75, pp. 1449–1466, 2006.
[27] W. L. Cruz, “A spectral algorithm for large-scale systems of nonlinear monotone equations,” Numerical Algorithms, vol. 76, no. 4, pp. 1109–1130, 2017.
[28] Z. Dai, H. Zhou, J. Kang et al., “The skewness of oil price returns and equity premium predictability,” Energy Economics, vol. 94, Article ID 105069, 2021.
[29] Z. Dai and J. Kang, “Some new efficient mean-variance portfolio selection models,” International Journal of Finance and Economics, vol. 7, pp. 1–13, 2021.
[30] Z. F. Dai and H. Zhu, "Stock return predictability from a mixed model perspective," *Pacific-Basin Finance Journal*, vol. 60, Article ID 101267, 2020.

[31] M. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 586–597, 2008.