Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6391321, 13 pages
https://doi.org/10.1155/2020/6391321

Research Article

A New Hybrid PRPFR Conjugate Gradient Method for Solving Nonlinear Monotone Equations and Image Restoration Problems

Yingjie Zhou, Yulun Wu, and Xiangrong Li

College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, China

Correspondence should be addressed to Xiangrong Li; xrli68@163.com

Received 24 July 2020; Accepted 25 August 2020; Published 25 September 2020

Guest Editor: Wenjie Liu

A new hybrid PRPFR conjugate gradient method is presented in this paper, which is designed such that it owns sufficient descent property and trust region property. This method can be considered as a convex combination of the PRP method and the FR method while using the hyperplane projection technique. Under accelerated step length, the global convergence property is gained with some appropriate assumptions. Comparing with other methods, the numerical experiments show that the PRPFR method is more competitive for solving nonlinear equations and image restoration problems.

1. Introduction

Consider the following nonlinear equation:

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

where S is a closed and convex subset. The function $F\colon \mathfrak{R}^n \longrightarrow \mathfrak{R}^n$ is monotone and continuously differentiable which satisfies

$$\langle F(x_1) - F(x_2), x_1 - x_2 \rangle \geq 0, \quad \forall x_1, x_2 \in \mathfrak{R}^n. \tag{2}$$

This problem of monotone equations has all kinds of applications, such as compressive sensing [1] and chemical equilibrium problems [2]. The conjugate gradient algorithm generates the iteration point via

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots, \tag{3}$$

where α_k is the step size generated from a proper line search and d_k is a search direction. There are many methods to find d_k, such as the Newton method [3], quasi-Newton method [4], and conjugate gradient method [5–13]. As we all know, the Newton method, the quasi-Newton method, and their related methods are very popular due to their local superlinear convergence property. But, it is expensive for them to compute the Jacobian matrix or the approximate Jacobian matrix in per iteration while the dimensions are very large.

Due to the simplicity, less storage, efficiency, and nice convergence property, the conjugate gradient method becomes more and more popular for solving nonlinear equations [14–18]. The search direction of the conjugate gradient method is usually defined as

$$d_k = \begin{cases} -F_k, & \text{if } k = 0, \\ -F_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{4}$$

where $F_k = F(x_k)$ and β_k is a parameter. Diverse β_k means a diverse CG method. There are some famous CG methods such as the FR method [7], the PRP method [10], the DY method [5], the LS method [9], the HS method [8], and the CD method [6]. The parameter we mentioned is as follows:

$$\beta_k^{FR} = \frac{\|F_k\|^2}{\|F_{k-1}\|^2}, \tag{5}$$

$$\beta_k^{\mathrm{PRP}} = \frac{F_K^T y_{k-1}}{\|F_{k-1}\|^2}, \tag{6}$$

$$\beta_k^{DY} = \frac{\|F_k\|^2}{d_{k-1}^T y_{k-1}}, \quad (7)$$

$$\beta_k^{\mathrm{LS}} = \frac{F_k^T y_{k-1}}{-F_{k-1}^T d_{k-1}}, \quad (8)$$

$$\beta_k^{\mathrm{HS}} = \frac{F_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad (9)$$

$$\beta_k^{\mathrm{CD}} = \frac{\|F_k\|^2}{-g_{k-1}^T y_{k-1}}, \quad (10)$$

where $y_{k-1} = F_k - F_{k-1}$ and $\|\cdot\|$ means the Euclidian norm.

In 1990, the first hybrid conjugate gradient method was proposed by Touati-Ahmed and Storey [19], and this method has global convergence while using the strong Wolfe line search. Dai and Yuan [20] proposed a CG method for unconstrained optimization and proved the global convergence of these hybrid computational schemes. Dong and Jiao [21] proposed a convex combination of the PRP and DY methods for solving nonlinear equations. Furthermore, the projection technique proposed by Solodov and Svaiter [22] motivated many scholars to further develop methods for solving (1) [23, 24].

Inspired by the convex combination proposed in [25], we proposed a convex combination of a modified PRP and a modified FR method with trust region property which [25] was not present before. We also try to make the algorithm inherit the convergence of the PRP method and excellent performance of the FR method.

The main contributions of the hybrid CG method are as follows:

(i) The given direction is designed as a convex combination of PRP and FR methods
(ii) The given direction has the sufficient descent property
(iii) The given direction has the trust region property
(iv) The global convergence of the presented algorithm is proved
(v) Numerical experiments show that the algorithm is more competitive for nonlinear equations and image restoration problems

In Section 2, we introduce the motivation and the PRPFR algorithm. In Section 3, we prove the sufficient property and the trust region property and give the convergence analysis. Section 4 shows the numerical experiment results. The conclusion is reported in the last section.

2. Algorithm

Motivated by the convex combination proposed in [25], we proposed a method which is a convex combination of a modified PRP method and a modified FR method for solving problem (1) and image restoration problems. We have recalled the classic PRP and classic FR conjugate gradient parameters in (5) and (6). For global convergence and good numerical performance, we defined the modified parameters as

$$\beta_k^{\mathrm{MPRP}} = \frac{F_k^T y_{k-1}}{\max\left\{t\|d_{k-1}\|\|y_{k-1}\|, \|F_{k-1}\|^2\right\}}, \quad (11)$$

$$\beta_k^{MFR} = \frac{\|F_k\|^2}{\max\left\{t\|d_{k-1}\|\|F_k\|, \|F_{k-1}\|^2\right\}}, \quad (12)$$

where $t > 0$ is a constant.

Next, we utilized the global convergence of β_k^{MPRP} and the excellent numerical behavior of β_k^{MFR} by defining a new parameter called β_k^{PRPFR}. The new parameter is defined as

$$\beta_k^{\mathrm{PRPFR}} = (1-\gamma_k)\beta_k^{\mathrm{MPRP}} + \gamma_k \beta_k^{\mathrm{MFR}}, \quad (13)$$

where $\gamma_k = \|y_{k-1}\|^2 / y_{k-1}^T \widehat{s}_{k-1}$, $\widehat{s}_{k-1} = s_{k-1} + (\max\{0, -s_{k-1}^T y_{k-1}/\|y_{k-1}\|^2\} + 1) y_{k-1}$, $s_{k-1} = x_k - x_{k-1}$.

The definition of $\widehat{s}_{k-1}$ is by Li and Fukushima [4], and γ_k is proposed in [26]. We now proposed a hybrid gradient search direction as

$$d_k = \begin{cases} -F_k, & \text{if } k = 0, \\ -\left(1 + \beta_k^{\mathrm{PRPFR}} \dfrac{F_k^T d_{k-1}}{\|F_k\|^2}\right) F_k + \beta_k^{\mathrm{PRPFR}} d_{k-1}, & \text{if } k \geq 1, \end{cases} \quad (14)$$

where β_k^{PRPFR} is defined in (13).

Remark 1. By the definition of $\widehat{s}_{k-1}$ and y_{k-1}, we obtain that

$$\begin{aligned} y_{k-1}^T \widehat{s}_{k-1} &\geq y_{k-1}\left\{ s_{k-1} + \left(-\frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2} + 1 \right) y_{k-1} \right\} \\ &= y_{k-1}^T s_{k-1} - s_{k-1}^T y_{k-1} + \|y_{k-1}\|^2 \\ &= \|y_{k-1}\|^2 \\ &> 0. \end{aligned} \quad (15)$$

Therefore, we have

$$0 < \frac{\|y_{k-1}\|^2}{y_{k-1}^T \widehat{s}_{k-1}} \leq 1. \quad (16)$$

Remark 2. By the definition of β_k^{PRPFR} and γ_k, we have

$$\left|\beta_k^{\mathrm{PRPFR}}\right| \leq \left|\beta_k^{\mathrm{MPRP}}\right| + \left|\beta_k^{\mathrm{MFR}}\right|. \quad (17)$$

Now, we introduce the hyperplane projection method by Solodov and Svaiter [22]. We first give the definition of the projection operator $P_S(\cdot)$:

$$P_S(x) = \arg\min\{\|x-y\| \mid y \in S\}, \quad \forall x \in \mathfrak{R}^n. \tag{18}$$

There is a fundamental property, that is,

$$\|P_S(x_1) - P_S(x_2)\| \le \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathfrak{R}^n. \tag{19}$$

The hyperplane projection method is as follows: let x_k be the current iteration point and $z_k = x_k + \alpha_k d_k$ be obtained by line search direction d_k such that $F(z_k)^T(x_k - z_k) > 0$. According to the monotonicity of $F(x)$, we have

$$F(z_k)^T(x^* - z_k) \le 0, \tag{20}$$

if x^* is a solution. Then, the hyperplane

$$H_k = \left\{x \in \mathfrak{R}^n \,\middle|\, F(z_k)^T(x_k - z_k) = 0\right\} \tag{21}$$

strictly separates the current iteration x_k from the solution of problem (1). The next iterate can be computed by projecting on it. That is,

$$x_{k+1} = x_k - \frac{F(z_k)^T(x_k - z_k)F(z_k)}{\|F(z_k)\|^2}. \tag{22}$$

As for the step size, we will use an appropriate line search to make performance better. Andrei [27] presented an acceleration scheme that generates the step size α_k in a multiplicative manner to improve the reduction of the function values along the iterations. The step size is defined as follows:

$$\widehat{\alpha}_k = \xi_k \alpha_k, \tag{23}$$

$$\begin{aligned} \xi_k &= -\frac{\varphi_k}{\phi_k}, \\ \varphi_k &= \alpha_k F_k^T d_k, \\ \phi_k &= -\alpha_k (F_k - F(z_k))^T d_k. \end{aligned} \tag{24}$$

If $\phi_k > 0$, let $\alpha_k = \widehat{\alpha}_k$.

Motivated by the above discussions, we proposed a hybrid conjugate gradient algorithm in Algorithm 1.

Algorithm 1. PRPFR conjugate gradient algorithm.

Step 0: given an initial point $x_0 \in \mathbb{R}^n$ and constant $0 < \varepsilon < 1$, $0 < \rho < 1$, $\kappa > 0$, $\sigma > 0$, and $t > 0$, let $k{:} = 0$.

Step 1: if $\|F_k\| \le \varepsilon$, stop. Otherwise, compute d_k by (11)–(14).

Step 2: choosing $\alpha_k = \max\{\kappa\rho^i : i = 0, 1, 2, \ldots,\}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \ge \sigma \alpha_k \|d_k\|^2. \tag{25}$$

Step 3: computing ϕ_k through (24), if $\phi_k > 0$, then $\alpha_k = \widehat{\alpha}_k$ by (23).

Step 4: setting $z_k = x_k + \alpha_k d_k$, if $\|F(z_k)\| \le \varepsilon$, stop. Otherwise, compute the next iterate x_{k+1} using (22).

Step 5: let $k{:} = k + 1$, go to Step 2.

3. Convergence Analysis

We will establish the convergence of Algorithm 1 in this section. First, we give the following assumptions.

Assumption 1. The function $F(x)$ is Lipschitz continuous; that is, there exists a positive constant L such that

$$\|F(x_1) - F(x_2)\| \le L\|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathbb{R}^n. \tag{26}$$

Assumption 2. The solution set of problem (1) is nonempty.

From Assumption 1, there exists a positive constant ω such that

$$\|F(x_k)\| \le \omega. \tag{27}$$

Lemma 1. *Let d_k be defined by (11)–(14), and then d_k satisfies the sufficient descent condition. That is,*

$$F_k^T d_k = -\|F_k\|^2. \tag{28}$$

Proof. If $k = 0$, we have $F_0^T d_0 = -\|F_0\|^2$. If $k \ge 1$, by (14), we have

$$\begin{aligned} F_k^T d_k &= -\left(1 + \beta_k^{\mathrm{PRPFR}} \frac{F_k^T d_{k-1}}{\|F_k\|^2}\right) F_k^T F_k + \beta_k^{\mathrm{PRPFR}} F_k^T d_{k-1} \\ &= -\|F_k\|^2 - \beta_k^{\mathrm{PRPFR}} \frac{F_k^T d_{k-1}}{\|F_k\|^2} \|F_k\|^2 + \beta_k^{\mathrm{PRPFR}} F_k^T d_{k-1} \\ &= -\|F_k\|^2 - \beta_k^{\mathrm{PRPFR}} F_k^T d_{k-1} + \beta_k^{\mathrm{PRPFR}} F_k^T d_{k-1} \\ &= -\|F_k\|^2. \end{aligned} \tag{29}$$

□

Lemma 2. *From d_k defined in (11)–(14), d_k satisfies the trust region property independent of the line search. That is,*

$$\|F_k\| \le \|d_k\| \le \left(1 + \frac{2}{t}\right)\|F_k\|. \tag{30}$$

Proof. According to equality (28) and Cauchy–Schwartz inequality, we have

$$\|F(x_k)\| \le \|d_k\|. \tag{31}$$

Furthermore, since $\max\left\{t\|d_{k-1}\|\|y_{k-1}\|, \|F_{k-1}\|^2\right\} \ge t\|d_{k-1}\|\|y_{k-1}\|$ and $\max\left\{t\|d_{k-1}\|\|F_k\|, \|F_{k-1}\|^2\right\} \ge t\|d_{k-1}\|\|F_k\|$, then

$$\begin{aligned}\|d_k\| &= \left\| -\left(1+\beta_k^{\mathrm{PRPFR}}\frac{F_k^T d_{k-1}}{\|F_k\|^2}\right)F_k + \beta_k^{\mathrm{PRPFR}} d_{k-1}\right\| \\ &\le \|F_k\| + \left|\beta_k^{\mathrm{PRPFR}}\right| \frac{\|F_k\|^2\|d_{k-1}\|}{\|F_k\|^2} + \left|\beta_k^{\mathrm{PRPFR}}\right|\|d_{k-1}\| \\ &\le \|F_k\| + 2\left(\left|\beta_k^{MPRP}\right| + \left|\beta_k^{MFR}\right|\right)\|d_{k-1}\| \\ &\le \|F_k\| + 2\left(\frac{\|F_k\|\|y_{k-1}\|}{\max\left\{t\|d_{k-1}\|\|y_{k-1}\|, \|F_{k-1}\|^2\right\}}\right. \\ &\quad \left. + \frac{\|F_k\|^2}{\max\left\{t\|d_{k-1}\|\|F_k\|, \|F_{k-1}\|^2\right\}}\right)\|d_{k-1}\| \\ &\le \|F_k\| + 2\left(\frac{\|F_k\|\|y_{k-1}\|}{t\|d_{k-1}\|\|y_{k-1}\|} + \frac{\|F_k\|^2}{t\|d_{k-1}\|\|F_k\|}\right)\|d_{k-1}\| \\ &= \|F_k\| + \frac{2\|F_k\|}{t} \\ &= \left(1+\frac{2}{t}\right)\|F_k\|. \end{aligned} \tag{32}$$

Then, the proof is complete. □

Lemma 3. *If $\{x_k\}$ and $\{z_k\}$ can be generated by the PRPRFR algorithm, then the step size α_k satisfies*

$$\alpha_k \ge \min\left\{\kappa, \frac{\rho\|F_k\|^2}{(L+\sigma)\|d_k\|^2}\right\}. \tag{33}$$

Proof. From the line search (25), supposing $\alpha_k \ne \kappa$, then $\alpha_k' = \alpha_k\rho^{-1}$ does not satisfy the line search (25). That is,

$$-F\left(x_k+\alpha_k' d_k\right)^T d_k < \sigma\alpha_k'\|d_k\|^2. \tag{34}$$

Using the Lipschitz continuous of $F(x)$ and (28), we have

$$\begin{aligned}\|F_k\|^2 &= -F_k^T d_k \\ &= \left(F\left(x_k+\alpha_k' d_k\right) - F_k\right)^T d_k - F\left(x_k+\alpha_k' d_k\right)^T d_k \\ &\le L\alpha_k'\|d_k\|^2 + \sigma\alpha_k'\|d_k\|^2 \\ &= \alpha_k'(L+\sigma)\|d_k\|^2. \end{aligned} \tag{35}$$

Therefore,

$$\alpha_k \ge \frac{\rho\|F_k\|^2}{(L+\sigma)\|d_k\|^2}. \tag{36}$$

The proof is complete. □

Lemma 4 (see [22]). *Suppose that $x^* \in \mathbb{R}^n$ satisfies $F(x^*) = 0$. Let $\{x_k\}$ be generated by the PRPFR algorithm. Then,*

$$\|x_{k+1}-x^*\|^2 \le \|x_k - x^*\|^2 - \|x_{k+1}-x_k\|^2. \tag{37}$$

Lemma 5. *Let $\{x_k\}$ be generated by the PRPFR algorithm, and then*

$$\lim_{k\longrightarrow\infty} \alpha_k\|d_k\| = 0. \tag{38}$$

Proof. Lemma 4 indicates that $\{x_k\}$ is bounded and

$$\lim_{k\longrightarrow\infty}\|x_{k+1}-x_k\| = 0. \tag{39}$$

From (22) and (25), we have that

$$\begin{aligned}\|x_{k+1}-x_k\| &= \frac{\left|F(z_k)^T(x_k-z_k)\right|\|F(z_k)\|}{\|F(z_k)\|^2} \\ &= \frac{-\alpha_k F(z_k)^T d_k}{\|F(z_k)\|} \\ &\ge \sigma\alpha_k^2\|d_k\|^2 \ge 0. \end{aligned} \tag{40}$$

From the abovementioned, the proof is complete.

Now, we establish the global convergence of the PRPFR algorithm. □

Theorem 1. *Let $\{x_k\}$ be generated by the PRPFR algorithm. Then,*

$$\lim\inf_{k\longrightarrow\infty}\|F_k\| = 0. \tag{41}$$

Proof. We assume (41) does not hold, and then there exists $\delta > 0$ such that $\forall k \ge 0$, $\|F_k\| \ge \delta$. This together with (30) yields

$$\|d_k\| \ge \|F_k\| \ge \delta, \quad \forall k \ge 0. \tag{42}$$

According to (21)–(26) and (42), we get

$$\begin{aligned}\|d_k\| &\le \|F_k\| + \left|\beta_k^{\mathrm{PRPFR}}\right| \frac{\|F_k\|^2\|d_{k-1}\|}{\|F_k\|^2} + \left|\beta_k^{\mathrm{PRPFR}}\right|\|d_{k-1}\| \\ &\le \|F_k\| + 2\left(\left|\beta_k^{\mathrm{MPRP}}\right| + \left|\beta_k^{\mathrm{MFR}}\right|\right)\|d_{k-1}\| \\ &\le \|F_k\| + 2\left(\frac{\|F_k\|\|y_{k-1}\|}{\max\left\{t\|d_{k-1}\|\|y_{k-1}\|, \|F_{k-1}\|^2\right\}}\right. \\ &\quad \left. + \frac{\|F_k\|^2}{\max\left\{t\|d_{k-1}\|\|F_k\|, \|F_{k-1}\|^2\right\}}\right)\|d_{k-1}\| \\ &\le \|F_k\| + 2\left(\frac{\|F_k\|\|y_{k-1}\|}{\|F_{k-1}\|^2} + \frac{\|F_k\|^2}{\|F_{k-1}\|^2}\right)\|d_{k-1}\| \\ &\le \omega + 2\left(\omega\delta^{-2}L\|x_k - x_{k-1}\| + \omega^2\delta^{-2}\right)\|d_{k-1}\|.\end{aligned} \tag{43}$$

The fourth inequality holds since $\max\left\{t\|d_{k-1}\|\|y_{k-1}\|, \|F_{k-1}\|^2\right\} \ge \|F_{k-1}\|^2$ and $\max\left\{t\|d_{k-1}\|\|F_k\|, \|F_{k-1}\|^2\right\} \ge \|F_{k-1}\|^2$. The last inequality implies that $\|d_k\|$ is bounded. By (30), we have

$$\|d_k\| \le M, \tag{44}$$

where $M = (1 + (2/t))\omega$. Multiplying both sides of (33) with $\|d_k\|$, we obtain that

$$\begin{aligned}\alpha_k\|d_k\| &\ge \max\left\{\kappa, \frac{\rho\|F_k\|^2}{(L+\sigma)\|d_k\|^2}\right\}\|d_k\| \\ &\ge \max\left\{\kappa\delta, \frac{\rho\delta^2}{(L+\sigma)M}\right\} \\ &> 0.\end{aligned} \tag{45}$$

This inequality mentioned above contradicts (38); therefore, (41) holds. □

4. Numerical Experiments

In this section, we report some numerical experiments to investigate the computational efficiency of the proposed hybrid conjugate gradient algorithm. The numerical experiments will be divided into two sections including normal nonlinear equations and image restoration problems. All tests in this section are written in MATLAB 2018a, run on a PC with AMD Ryzen 7 4800 U with Radeon Graphics 1.80 Hz, 16 GB of SDRAM memory, and Windows 10 system.

4.1. Normal Nonlinear Equations. In this section, we test some normal nonlinear equations. The concrete test problems come from [28] and are listed in Table 1. To compare the numerical performance of the PRPFR algorithm, we also do the experiments with the modified PRP algorithm [29] and the FR algorithm [7]. The columns of Tables 2–4 have the following meaning.

Table 1: Test problems.

No.	Test problems
1	Exponential function 2
2	Trigonometric function
3	Singular function
4	Logarithmic function
5	Broyden tridiagonal function
6	Trigexp function
7	Strictly convex function 1
8	Variable dimensioned function
9	Tridiagonal system
10	Five-diagonal system
11	Extended Freudenstein and Roth function (n is even)
12	Discrete boundary value problem

Initialization: the parameters are chosen as $\rho = 0.5$, $\kappa = 1$, $\sigma = 0.5$, $t = 0.85$, and $\varepsilon = 10^{-5}$.

Stop rule: when $\|F(x)\| \le 10^{-5}$ or $\mathrm{NI} \ge 20000$ is satisfied, we stop the process.

According to Tables 2–4, it is evident that the three methods can solve most of the test problems with $\mathrm{NI} \ge 20000$. However, the FR method cannot handle the function 3 with 9000 dimensions, 30000 dimensions, and 90000 dimensions, while the PRPFR can solve the problem with $\mathrm{NI} \ge 20000$. For more directly knowing the performance of this method, Dolan and Moré [30] introduced a technique to compare the performance of different algorithms. In Figure 1, when $\tau > 1.2$, the PRPFR algorithm is obviously better than the FR algorithm and the modified PRP algorithm. In Figure 2, the PRPFR algorithm solves all the problems at approximately $\tau = 4.4$, while the FR algorithm solves 90% of the test problems at approximately $\tau = 4.6$, and the modified PRP algorithm solves 75% of the test problems at approximately $\tau = 5$. In Figure 3, the PRPFR algorithm solves 94% of the problems at approximately $\tau = 3.3$. The FR algorithm and the modified PRP algorithm solve 77% and 58% of the test problems at approximately $\tau = 3.3$ and $\tau = 4$, respectively. From Tables 2–4 and Figures 1–3, it is obvious that the performance of the PRPFR algorithm is better than that of the FR algorithm and the modified PRP algorithm for most problems. Therefore, we conclude that the PRPFR algorithm is competitive to the FR algorithm and the modified PRP algorithm.

4.2. Image Restoration Problems. Image restoration aims to recover the original image from an image damaged by impulse noise. These problems are significant in optimization fields. The stop rule is $((\|F_{k+1}\| - \|F_k\|)/\|F_k\|) < 10^{-3}$ or $(\|(x_{k+1} - x_k)\|/\|x_k\|) < 10^{-3}$. The experiments choose Lena (512×512), Barbara (512×512), and Man (1024×1024) as the test images. Meanwhile, we compare the PRPFR algorithm experiments' performance with that of the modified PRP algorithm, where the step size α_k is generated by Step 2

Table 2: Test results of the PRPFR algorithm.

No.	Dim	PRPFR algorithm NI/NF	CPU	GN
1	3000	13/135	0.03125	$8.02E-06$
	9000	17/205	1.71875	$9.48E-06$
	30000	61/802	13.046875	$9.73E-06$
	90000	6/91	2.421875	$7.17E-06$
2	3000	14/39	0.0625	$9.54E-06$
	9000	14/39	0.828125	$5.50E-06$
	30000	13/36	0.984375	$6.01E-06$
	90000	12/33	1.21875	$6.18E-06$
3	3000	19999/20253	33.25	$1.46E-05$
	9000	19501/19813	425.3125	$9.99E-06$
	30000	17291/17984	783.796875	$1.00E-05$
	90000	19518/21490	1765.453125	$1.00E-05$
4	3000	24/49	0.0625	$7.65E-06$
	9000	25/51	0.515625	$6.59E-06$
	30000	26/53	1.0625	$6.00E-06$
	90000	27/55	2.734375	$5.20E-06$
5	3000	49/192	0.078125	$6.04E-06$
	9000	60/229	1.140625	$9.16E-06$
	30000	49/190	3.5	$4.52E-06$
	90000	46/179	5.25	$5.04E-06$
6	3000	28/147	0.109375	$7.39E-06$
	9000	29/151	1.1875	$7.66E-06$
	30000	29/154	3.09375	$7.03E-06$
	90000	33/170	5.171875	$3.56E-06$
7	3000	22/46	0.046875	$5.78E-06$
	9000	23/48	0.234375	$5.00E-06$
	30000	23/48	1.203125	$9.14E-06$
	90000	24/50	2.234375	$7.91E-06$
8	3000	1/3	0.0625	$3.19E-08$
	9000	1/3	0	$0.00E+00$
	30000	1/3	0.03125	$0.00E+00$
	90000	1/3	0.078125	$0.00E+00$
9	3000	626/3337	1.96875	$9.89E-06$
	9000	1696/10439	71.9375	$9.91E-06$
	30000	1410/8756	150.765625	$9.98E-06$
	90000	1374/8779	236.90625	$9.99E-06$
10	3000	917/5266	1.53125	$9.96E-06$
	9000	664/3474	24.890625	$9.92E-06$
	30000	480/3047	49.15625	$9.87E-06$
	90000	759/4313	119.015625	$9.87E-06$
11	3000	305/1898	0.625	$9.96E-06$
	9000	276/1714	11.078125	$9.69E-06$
	30000	333/2069	35.9375	$9.91E-06$
	90000	283/1759	44.671875	$8.69E-06$
12	3000	12/37	0.046875	$5.29E-06$
	9000	11/34	0.953125	$5.99E-06$
	30000	10/31	0.75	$6.52E-06$
	90000	9/28	1.5	$7.52E-06$

No : the serial number of the problem. Dim: the dimensions of x. NF: the function evaluation numbers. NI: the iteration numbers. CPU: the calculation time in seconds. GN: the final function norm evaluations when the program is stopped.

Table 3: Test results of the modified PRP algorithm.

No.	Dim	Modified PRP algorithm		
		NI/NF	CPU	GN
1	3000	38/389	0.234375	$9.84E-06$
	9000	21/250	1.6875	$9.08E-06$
	30000	20/271	4.203125	$9.70E-06$
	90000	10/155	3.78125	$8.09E-06$
2	3000	18/47	0.0625	$9.87E-06$
	9000	18/47	0.6875	$5.67E-06$
	30000	17/44	0.90625	$6.32E-06$
	90000	15/39	1.703125	$7.76E-06$
3	3000	9999/10404	10.984375	$2.83E-05$
	9000	9999/10873	190.140625	$5.01E-05$
	30000	9999/11895	435.296875	$3.15E-05$
	90000	9999/13913	839.734375	$4.85E-05$
4	3000	42/194	0.109375	$5.69E-09$
	9000	66/352	3	$3.37E-06$
	30000	121/748	13.28125	$4.61E-07$
	90000	218/1538	41.4375	$7.05E-08$
5	3000	67/280	0.1875	$5.90E-06$
	9000	90/393	3.171875	$8.05E-06$
	30000	111/541	9.453125	$7.88E-06$
	90000	139/787	24.28125	$6.16E-06$
6	3000	75/517	0.265625	$6.67E-06$
	9000	107/840	7.203125	$3.88E-06$
	30000	165/1487	26.96875	$7.27E-06$
	90000	252/2537	77.25	$8.93E-06$
7	3000	39/138	0.046875	$6.65E-06$
	9000	56/245	2.84375	$6.87E-06$
	30000	89/477	7.53125	$5.70E-06$
	90000	139/869	25.09375	$6.00E-06$
8	3000	1/2	0.03125	$0.00E+00$
	9000	1/2	0	$0.00E+00$
	30000	1/2	0.0625	$0.00E+00$
	90000	1/2	0.15625	$0.00E+00$
9	3000	5544/43870	12.765625	$9.85E-06$
	9000	5832/49309	347.203125	$9.84E-06$
	30000	6424/60507	960.546875	$9.68E-06$
	90000	7445/80662	1983.578125	$9.83E-06$
10	3000	768/4997	1.515625	$9.84E-06$
	9000	881/6108	44.8125	$9.28E-06$
	30000	1043/8118	121.359375	$9.89E-06$
	90000	1343/11835	301.625	$8.92E-06$
11	3000	512/3345	1.84375	$9.85E-06$
	9000	545/3646	23.953125	$9.95E-06$
	30000	1122/7498	119.3125	$9.90E-06$
	90000	1229/8722	223.140625	$9.54E-06$
12	3000	16/48	0.078125	$7.22E-06$
	9000	13/39	0.5625	$8.96E-06$
	30000	11/33	1.390625	$8.41E-06$
	90000	10/30	1.671875	$7.92E-06$

Table 4: Test results of the FR algorithm.

No.	Dim	FR algorithm NI/NF	CPU	GN
1	3000	29/371	0.203125	$9.84E-06$
	9000	21/294	2.25	$9.38E-06$
	30000	12/183	3.984375	$9.11E-06$
	90000	5/80	2.09375	$6.44E-06$
2	3000	13/50	0.078125	$6.48E-06$
	9000	12/46	0.484375	$7.47E-06$
	30000	11/42	0.984375	$8.14E-06$
	90000	10/38	1.484375	$9.72E-06$
3	3000	19999/266078	194	$1.12E-03$
	9000	19999/265865	2918.28125	$1.20E-03$
	30000	19999/265318	6049.65625	$7.84E-04$
	90000	19999/266012	8251.859375	$3.84E-04$
4	3000	24/72	0.046875	$5.59E-06$
	9000	24/72	0.59375	$9.65E-06$
	30000	25/75	1.28125	$8.80E-06$
	90000	26/78	3.203125	$7.62E-06$
5	3000	62/381	0.21875	$7.73E-06$
	9000	69/462	3	$9.20E-06$
	30000	101/767	12.609375	$9.75E-06$
	90000	128/1215	26.90625	$8.22E-06$
6	3000	26/165	0.109375	$9.06E-06$
	9000	21/127	0.90625	$6.10E-06$
	30000	22/134	1.9375	$7.51E-06$
	90000	22/134	3.359375	$5.31E-06$
7	3000	23/70	0.0625	$5.30E-06$
	9000	23/70	0.40625	$9.17E-06$
	30000	24/73	1.578125	$8.37E-06$
	90000	25/76	1.90625	7.25E − 06
8	3000	1/3	0	$3.19E-08$
	9000	1/3	0.078125	$0.00E+00$
	30000	1/3	0.140625	0.00E + 00
	90000	1/3	0.03125	$0.00E+00$
9	3000	1248/15399	4.46875	$9.91E-06$
	9000	1544/19579	125.3125	$9.87E-06$
	30000	1175/14881	203.828125	$9.96E-06$
	90000	1027/13000	263.96875	$9.91E-06$
10	3000	763/8892	2.640625	$9.90E-06$
	9000	765/8912	53.84375	$9.92E-06$
	30000	806/9498	127.421875	$9.65E-06$
	90000	1572/20089	423.4375	$9.97E-06$
11	3000	373/4072	1.078125	$9.98E-06$
	9000	387/4226	24.71875	$9.49E-06$
	30000	402/4393	57.828125	$9.77E-06$
	90000	415/4537	97.296875	$9.44E-06$
12	3000	10/36	0.0625	$7.27E-06$
	9000	9/31	0.21875	$5.83E-06$
	30000	6/19	0.234375	$7.66E-06$
	90000	6/19	0.765625	$3.95E-06$

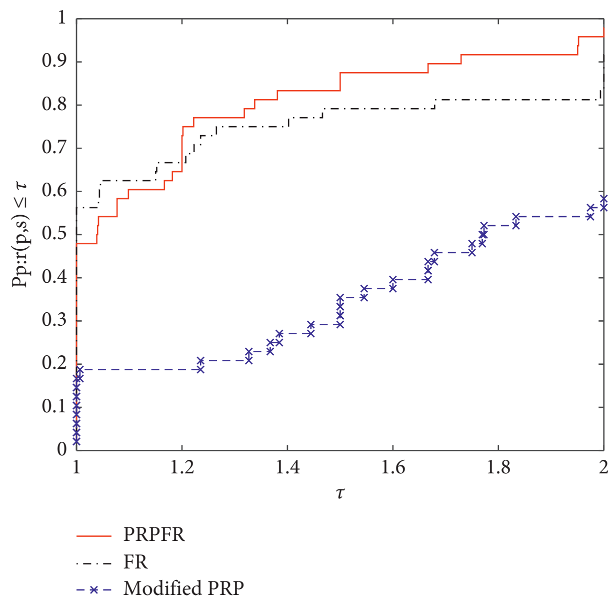

FIGURE 1: Performance profiles of the methods (NI).

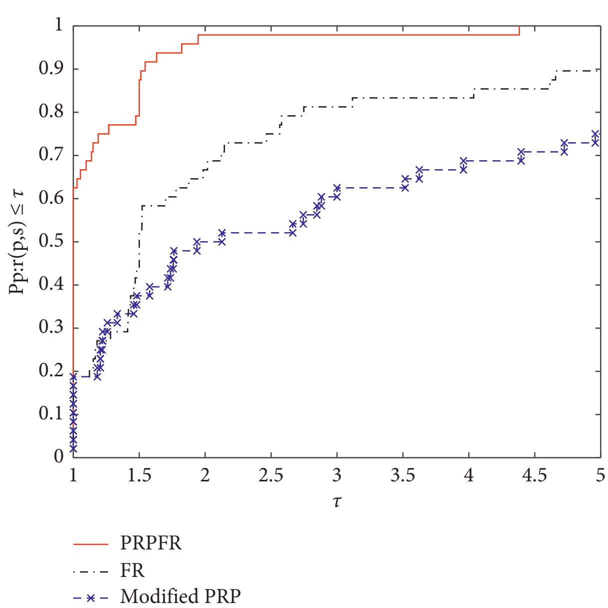

FIGURE 2: Performance profiles of the methods (NF).

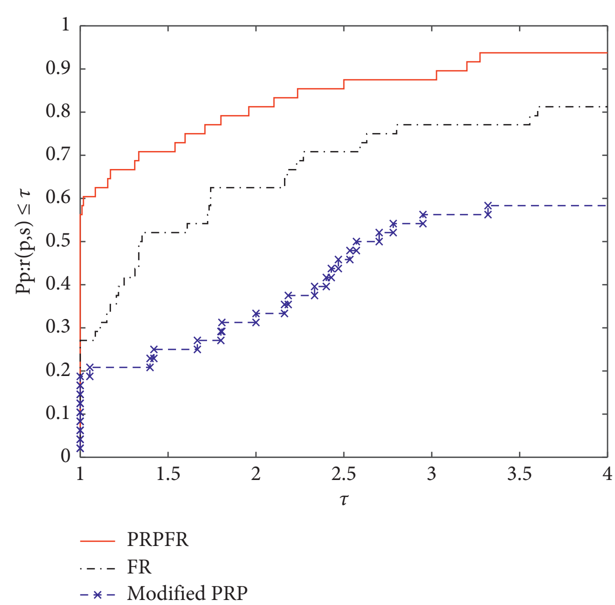

FIGURE 3: Performance profiles of the methods (CPU).

and Step 3 in the PRPFR algorithm. The detailed performance results are shown in Figures 4–6. It can be observed that both the PRPFR algorithm and the modified PRP algorithm are able to restore the blur image of these three images.

From Figures 4–6, we can easily notice that both algorithms are successful for restoring these noisy images with 20%, 45%, and 70% noise. According to the results in Table 5, we can draw the conclusion that the PRPFR algorithm is more effective than the modified PRP algorithm for 20% noise problems, 45% noise problems, and 70% noise problems.

TABLE 5: CPU times of the PRPFR algorithm and the modified PRP algorithm in seconds.

20% noise	Lena	Barbara	Man	Total
PRPFR algorithm	2.719	3.578	12.844	19.141
Modified PRP algorithm	3.281	4.563	13.125	20.969
45% noise	Lena	Barbara	Man	Total
PRPFR algorithm	5.672	7.172	29.969	42.813
Modified PRP algorithm	6.156	7.797	31.609	45.563
70% noise	Lena	Barbara	Man	Total
PRPFR algorithm	14.063	17.391	61.906	93.359
Modified PRP algorithm	17.078	18.797	62.453	98.328

Figure 4: From left to right: a noisy image with 20% salt-and-pepper noise and the restorations obtained with the PRPFR algorithm and the modified PRP algorithm.

Figure 5: From left to right: a noisy image with 45% salt-and-pepper noise and the restorations obtained with the PRPFR algorithm and the modified PRP algorithm.

5. Conclusion

In this paper, a new hybrid conjugate gradient algorithm that combines the PRP and FR methods is proposed, while using the projection technique. The direction d_k has the sufficient descent and trust region properties automatically. Global convergence of the proposed algorithm is established under appropriate assumptions. The numerical experiments show that the proposed algorithm is competitive and efficient for solving nonlinear equations and image restoration problems.

For further research, we have some thinking as follows: (i) If the convex combination is applied to the quasi-Newton method, can it have better properties? (ii) Under other line search techniques, can this conjugate gradient method have global convergence? (iii) Can the proposed algorithm be applied to compressive sensing?

Figure 6: From left to right: a noisy image with 70% salt-and-pepper noise and the restorations obtained with the PRPFR algorithm and the modified PRP algorithm.

Data Availability

All data are included in the paper.

Conflicts of Interest

There are no potential conflicts of interest.

Acknowledgments

The authors want to thank the support of the funds. This work was supported by the High Level Innovation Teams and Excellent Scholars Program in Guangxi Institutions of Higher Education (Grant no. [2019]32), the National Natural Science Foundation of China (Grant no. 11661009), and the Guangxi Natural Science Key Foundation (no. 2017GXNSFDA198046).

References

[1] 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.

[2] K. Meintjes and A. P. Morgan, "Chemical equilibrium systems as numerical test problems," *ACM Transactions on Mathematical Software*, vol. 16, no. 2, pp. 143–151, 1990.

[3] P. N. Brown and Y. Saad, "Convergence theory of nonlinear Newton-Krylov algorithms," *SIAM Journal on Optimization*, vol. 4, no. 2, pp. 297–330, 1994.

[4] D.-H. Li and M. Fukushima, "A modified BFGS method and its global convergence in nonconvex minimization," *Journal of Computational and Applied Mathematics*, vol. 129, no. 1-2, pp. 15–35, 2001.

[5] Y. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, pp. 177–182, 2000.

[6] R. Fletcher, "Unconstrained optimization," *Pratical Method of Optimization*, vol. 1, 1987.

[7] R. Fletcher and C. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 2, pp. 149–154, 1964.

[8] 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. 409–432, 1952.

[9] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 129–137, 1991.

[10] E. Polak and G. Ribière, "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.

[11] G. Yuan, J. Lu, and Z. Wang, "The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems," *Applied Numerical Mathematics*, vol. 152, pp. 1–11, 2020.

[12] G. Yuan, X. Wang, and Z. Sheng, "Family weak conjugate gradient algorithms and their convergence analysis for nonconvex functions," *Numerical Algorithms*, vol. 84, no. 3, pp. 935–956, 2020.

[13] G. Yuan, Z. Wei, and Y. Yang, "The global convergence of the Polak-Ribière-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions," *Journal of Computational and Applied Mathematics*, vol. 362, pp. 262–275, 2019.

[14] A. B. Abubakar and P. Kumam, "A descent Dai-Liao conjugate gradient method for nonlinear equations," *Numerical Algorithms*, vol. 81, no. 1, pp. 197–210, 2019.

[15] A. B. Abubakar and P. Kumam, "An improved three-term derivative-free method for solving nonlinear equations," *Computational and Applied Mathematics*, vol. 37, no. 5, pp. 6760–6773, 2018.

[16] Z. Dai and H. Zhu, "A modified Hestenes-Stiefel-type derivative-free method for large-scale nonlinear monotone equations," *Mathematics*, vol. 8, no. 2, p. 168, 2020.

[17] G. Yuan, T. Li, and W. Hu, "A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems," *Applied Numerical Mathematics*, vol. 147, pp. 129–141, 2020.

[18] G. Yuan and M. Zhang, "A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 286, pp. 186–195, 2015.

[19] D. Touati-Ahmed and C. Storey, "Efficient hybrid conjugate gradient techniques," *Journal of Optimization Theory and Applications*, vol. 64, no. 2, pp. 379–397, 1990.

[20] Y. Dai and Y. Yuan, "An efficient hybrid conjugate gradient method for unconstrained optimization," *Annals of Operations Research*, vol. 103, pp. 33–47, 2001.

[21] J. Dong and B. Jiao, "A new hybrid PRP-DY conjugate gradient method," *2010 Third International Joint Conference on Computational Science and Optimization*, vol. 2, pp. 70–74, 2010.

[22] M. V. Solodov and B. F. Svaiter, "A globally convergent inexact Newton method for systems of monotone equations," *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, vol. 22, no. 1998, pp. 355–369, 1969.

[23] X. Y. Wang, S. J. Li, and X. P. Kou, "A self-adaptive three-term conjugate gradient method for monotone nonlinear equations with convex constraints," *Calcolo*, vol. 53, no. 2, pp. 133–145, 2016.

[24] 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.

[25] K. Zahra and A. Ali, "A new hybrid conjugate gradient method for large-scale unconstrained optimization problem with non-convex objective function," *Computational and Applied Mathematics*, vol. 38, p. 186, 2019.

[26] E. G. Birgin and J. Martnez, "A spectral conjugate gradient method for unconstrained optimization," *Applied Mathematics and Optimization*, vol. 43, no. 2, pp. 117–128, 2001.

[27] N. Andrei, "Another conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 159, no. 1, pp. 159–182, 2013.

[28] G. Yuan, Z. Wei, and S. Lu, "Limited memory BFGS method with backtracking for symmetric nonlinear equations," *Mathematical and Computer Modelling*, vol. 54, no. 1-2, pp. 367–377, 2011.

[29] L. Zhang, W. Zhou, and D.-H. Li, "A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence," *IMA Journal of Numerical Analysis*, vol. 26, no. 4, pp. 629–640, 2006.

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