mathematics

MDPI

Article

A Modified Hestenes-Stiefel-Type Derivative-Free Method for Large-Scale Nonlinear Monotone Equations

Zhifeng Dai * and Huan Zhu

College of Mathematics and Computational Science, Changsha University of Science and Technology, Changsha 410114, China; huanzhu1995@163.com

* Correspondence: zhifengdai823@163.com

Received: 26 December 2019; Accepted: 21 January 2020; Published: 30 January 2020

Abstract: The goal of this paper is to extend the modified Hestenes-Stiefel method to solve large-scale nonlinear monotone equations. The method is presented by combining the hyperplane projection method (Solodov, M.V.; Svaiter, B.F. A globally convergent inexact Newton method for systems of monotone equations, in: M. Fukushima, L. Qi (Eds.) Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Kluwer Academic Publishers. 1998, 355–369) and the modified Hestenes-Stiefel method in Dai and Wen (Dai, Z.; Wen, F. Global convergence of a modified Hestenes-Stiefel nonlinear conjugate gradient method with Armijo line search. *Numer Algor.* 2012, 59, 79–93). In addition, we propose a new line search for the derivative-free method. Global convergence of the proposed method is established if the system of nonlinear equations are Lipschitz continuous and monotone. Preliminary numerical results are given to test the effectiveness of the proposed method.

Keywords: nonlinear equations; monotonicity property; projection method; global convergence

1. Introduction

In this paper, we consider the problem of finding numerical solutions of the following large-scale nonlinear equations

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

where the function $F : R^n \longrightarrow R^n$ is monotone and continuous. If $F(x)$ is monotone, it implies that the following inequality holds

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

Nonlinear monotone equations can be applied in different fields, for example, they are used as subproblems in the generalized proximal algorithms with Bregman distances [1]. Some monotone variational inequality problems can be converted into nonlinear monotone equations [2]. Monotone systems of equations can also be applied in L_1-norm regularization sparse optimization problems (see [3,4]) and discrete mathematics such as graph theory (see [5,6]).

Being aware of the important applications of nonlinear monotone equations, in recent years, many scholars have paid attention to propose efficient algorithms for solving problem (1). These algorithms are mainly divided into the following categories.

Each of the Newton-type method, Levenberg-Marquardt method, and quasi-Newton method enjoy fast local convergence property, and are attractive (see [7–13]). However, for large-scale problems, a drawback of these methods is that at each iteration these algorithms require computing a large-scale linear system of equations by using approximate systems of equations or a Jacobian matrix. The large demand of storage for matrix results in improper handling of large-scale nonlinear monotone systems.

In recent years, gradient-type algorithms have attracted the attention of many scholars. The main reasons are low storage requirements, easy implementation, and global convergence under mild conditions. For example, the spectral gradient method [14] only needs to use gradient information which is easy but effective for an optimization problem. From a different perspective, the spectral gradient method [14] is extended to solve nonlinear monotone equations by combining it with the projection method (see [15,16]).

In addition, for large-scale unconstrained optimization problems, the conjugate gradient method (CG) is another easy but effective method, due to the two attractive features: one is the low memory requirement, the other is strong global convergence properties. In recent years, the conjugate gradient method has achieved rich results (see [17–36]) from the perspective of the sufficient descent property, qausi-Newton direction and conjugacy condition. Inspired by the extension of spectral gradient method to nonlinear monotone equations, CG methods have been applied in solving the nonlinear monotone equations problem. For example, PRP-type method ([37–43]); Perry conjugate gradient method ([44]); Liu-Storey type method [45], among others.

In this paper, we will focus on extending the Hestenes-Stiefel (HS) CG method to solve large-scale nonlinear monotone equations. To the best of our knowledge, the HS CG method [46] is generally considered as the most efficient CG method for computing performance. However, the HS CG method does not enjoy sufficient descent property. Based on the modified secant equation [10], Dai and Wen [47] propose a modified HS conjugate gradient method that can generate sufficient descent directions (i.e., $c > 0$ exists such that $g_k^T d_k < -c\|g_k\|^2$). Global convergence results under the Armijo line search are obtained in Dai and Wen [47]. Hence, we aim to present a derivative-free method to solve the nonlinear monotone Equations (1). The proposed method can be seem as a further study of the modified HS CG method in Dai and Wen [47] for unconstrained optimization problems.

Our paper makes two contributions to large-scale nonlinear monotone equations. Firstly, a new line search is proposed for the derivative-free method. A significant advantage of this line search is that it is easier to obtain the search stepsize. Secondly, we propose a derivative-free method for solving large-scale nonlinear monotone equations which combines the modified Hestenes-Stiefel method in Dai and Wen [47] for unconstrained optimization problems and the hyperplane projection method [13]. A good property of the proposed method is that it is suitable to solve large-scale nonlinear monotone equations due to its lower storage requirement.

The rest of the article is organized as follows. In Section 2, we give the algorithm and prove the sufficient descent property. The global convergence is proved in Section 3. We report the numerical results In Section 4. The last Section gives the conclusion.

2. Algorithm and the Sufficient Descent Property

In this section, we will present the derivative-free method for solving problem (1), that is a combination of the modified Hestenes-Stiefel method [47] and the hyperplane projection method [13]. Different from the traditional conjugate gradient method, the iteration sequence $\{x_{k+1}\}$ is obtained in two steps at each iteration.

In the first step, the algorithm produces an iterative sequence $\{z_k = x_k + \alpha_k d_k\}$, where d_k is the search direction, and $\alpha_k > 0$ is the steplength obtained by a suitable line search. For most iterative algorithms of optimization problems, the line search plays an important role in convergence analysis and numerical calculation. Zhang and Zhou [15] obtained the steplength $\alpha_k > 0$ by the following Armijio-type line search: calculating the search steplength $\alpha_k = max\{\beta\rho^i : i = 0, 1, \ldots, \}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \|d_k\|^2, \tag{3}$$

where β is some initial attempt for α_k, $\beta > 0$ and $\rho \in (0, 1)$.

In addition, Li and Li [39] introduced an alternative line search, that is, computing the search steplength $\alpha_k = max\{\beta\rho^i : i = 0, 1, \ldots, \}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \|F(x_k + \alpha_k d_k)\| \cdot \|d_k\|^2, \tag{4}$$

From the above introduction, we can see that the steplength α_k is obtained by calculating $\alpha_k = max\{\beta\rho^i : i = 0, 1, \ldots, \}$ such that (3) or (4) is satisfied. If the point x_k is far from the solution, the obtained steplength α_k may be very small. Taking this into account, we present the following line search rule where the steplength α_k is obtained by computing $\alpha_k = max\{\beta\rho^i : i = 0, 1, \ldots, \}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \min\{\|d_k\|^2, \|F(x_k + \alpha_k d_k)\| \|d_k\|^2, -F(x_k)^T d_k\}. \tag{5}$$

In the second step, $\{x_{k+1}\}$ can be determined by using $x_k, z_k, F(z_k)$ via the hyperplane projection method [13]. Now, let's introduce how to generate $\{x_{k+1}\}$ via the hyperplane projection method [13]. Along the search direction $d_k > 0$, we can generate a point $z_k = x_k + \alpha_k d_k$ by a suitable line search such that

$$F(z_k)^T (x_k - z_k) > 0. \tag{6}$$

On the other hand, the monotonicity of F implies that for any solution $x^*(F(x^*) = 0)$, the following inequality holds

$$F(z_k)^T (x^* - z_k) = -(F(z_k) - F(x^*))^T (z_k - x^*) \leq 0. \tag{7}$$

From (6) and (7), we can see that $\{F(z_k)^T (x_k - z_k) > 0\}$ holds for any x_k and $\{F(z_k)^T (x^* - z_k) \leq 0\}$ holds for the solution x^*. Therefore, from (6) and (7), there is a hyperplane

$$H_k = \{x \in R^n | F(z_k)^T (x - z_k) = 0\}, \tag{8}$$

which can strictly separate the current point x_k from the x^* (zero point) of equation in (1).

Following Solodov and Svaiter [13] and Zhang and Zhou [15], taking the projection of x_k onto the hyperplane (8) as the next iterate x_{k+1} is a reasonable choice. In detail, the next iterate point x_{k+1} can be computed by

$$x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k)}{\|F(z_k)\|^2} F(z_k). \tag{9}$$

In what following, we pay our attention to the search direction which plays a crucial role in an iterative algorithm. Our main starting point is to extend the search direction of Dai and Wen [47] to nonlinear monotone equations problem (1). Similar to Dai and Wen [47] for unconstrained optimization, we give the search direction as

$$d_k = \begin{cases} -F_0, & \text{if} \quad k = 0, \\ -F_k + \beta_k^{\text{NHZ}} d_{k-1}, & \text{if} \quad k \geq 1, \end{cases} \tag{10}$$

where

$$\beta_k^{\text{NHZ}} = \frac{F_k^T y_{k-1}}{d_{k-1}^T w_{k-1}} - \mu \frac{\|y_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2} F_k^T d_{k-1}, \quad \mu > \frac{1}{4}, \tag{11}$$

$$w_{k-1} = y_{k-1} + \gamma \bar{s}_{k-1}, \gamma > 0, \quad y_{k-1} = F(x_k) - F(x_{k-1}), \quad \bar{s}_{k-1} = z_{k-1} - x_{k-1} = \alpha_{k-1} d_{k-1}. \tag{12}$$

For simplicity, we refer to (10) and (11) as NHZ method hereafter.

Further, in this paper, the function F is assumed to satisfy the following assumptions witch are often utilized in convergence analysis for nonlinear monotone equations (see, [37–45,48]).

Assumption 1. *(A_1) The F is a monotone function:*

$$(F(x) - F(y))^T (x - y)\rangle \geq 0, \quad \forall x, y \in R^n. \tag{13}$$

(A_2) *The F is Lipschitz continuous function, namely, there exists a $L > 0$ such that*

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in R^n. \tag{14}$$

In what following, we will describe the proposed Algorithm 1.

Algorithm 1: NHZ derivative-free method.

Step 0: Given $x_0 \in R^n$ as an initial point, and the constants $\varepsilon > 0, \beta > 0, \sigma > 0, \rho \in (0, 1)$. Then set k := 0.
Step 1: Calculate $F(x_k)$. If $\|F(x_k)\| \leq \varepsilon$, stop the algorithm. Otherwise, go to **step 2**.
Step 2: Determine the search direction d_k by (10), (11) and (12).
Step 3: Calculate the search steplength α_k by (5). Let $z_k = x_k + \alpha_k d_k$.
Step 4: Calculate $F(z_k)$. If $\|F(z_k)\| \leq \varepsilon$, stop the algorithm. Otherwise, calculate x_{k+1} by using the projection (9). Set k := k + 1 and go to **Step 1**.

In what follow, we will show that the proposed NHZ derivative-free method enjoys the sufficient descent property which plays an important role in proof of convergence. From now on, we use F_k to denote $F(x_k)$.

Theorem 1. *The search direction d_k generated by (10), (11) and (12) is sufficient descent direction. That is, if $d_k^T w_k \neq 0$, then we have*

$$F_k^T d_k \leq -(1 - \frac{1}{4\mu})\|F_k\|^2, \qquad \mu > \frac{1}{4}. \tag{15}$$

Proof. When $k = 0$, we have

$$F_0^T d_0 = -\|F_0\|^2 \leq -(1 - \frac{1}{4\mu})\|F_0\|^2.$$

It is obvious that (15) is satisfied for $k = 0$. □

Now we will show that the sufficient descent condition (15) holds for $k \geq 1$. We can obtain from (10) and (11) that

$$\begin{aligned} F_k^T d_k &= -\|F_k\|^2 + \beta_k F_k^T d_{k-1} \\ &= -\|F_k\|^2 + \left\{ \frac{F_k^T y_{k-1}}{d_{k-1}^T w_{k-1}} - \mu \frac{\|y_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2} F_k^T d_{k-1} \right\} F_k^T d_{k-1} \\ &= \frac{F_k^T y_{k-1}(d_{k-1}^T w_{k-1})(F_k^T d_{k-1}) - \|F_k\|^2 (d_{k-1}^T w_{k-1})^2 - \mu \|y_{k-1}\|^2 (F_k^T d_{k-1})^2}{(d_{k-1}^T w_{k-1})^2}. \end{aligned}$$

Define

$$u_k = \frac{1}{\sqrt{2\mu}}(d_{k-1}^T w_{k-1})F_k, \qquad v_k = \sqrt{2\mu}(F_k^T d_{k-1})y_{k-1}. \tag{16}$$

By using the Equation (16) and the inequality $u_k^T v_k \leq 1/2(\|u_k\|^2 + \|v_k\|^2)$, we have

$$\begin{aligned} F_k^T d_k &= \frac{u_k^T v_k - 1/2(\|u_k\|^2 + \|v_k\|^2)}{(d_{k-1}^T w_{k-1})^2} - (1 - \frac{1}{4\mu})\frac{(d_{k-1}^T w_{k-1})^2}{(d_{k-1}^T w_{k-1})^2}\|F_k\|^2 \\ &\leq -(1 - \frac{1}{4\mu})\|F_k\|^2. \end{aligned}$$

Thus (15) holds for $k \geq 1$.

3. Global Convergence Analysis

Now, we will investigate the global convergence of Algorithm 1. Firstly, we give the following Lemma which shows the line search strategy (5) is well-defined if the search directions $\{d_k\}$ satisfy the sufficient descent property.

Lemma 1. *If the iterative sequences $\{x_k\}$ and $\{z_k\}$ are generated by the Algorithm 1. Then, there always exists a steplength α_k satisfying the line search (5).*

Proof. Assume that, for any nonnegative integer i for $\beta\rho^i$, the line search strategy (5) does not hold in the k_0-th iterate, then we have

$$-F(x_{k_0}+\beta\rho^i d_{k_0})^T d_{k_0} < \sigma\beta\rho^i \min\{\|d_{k_0}\|^2, \|F(x_{k_0}+\beta\rho^i d_{k_0})\|\|d_{k_0}\|^2, -F(x_{k_0})^T d_{k_0}\}. \tag{17}$$

Letting $i \mapsto \infty$, we have from the continuity of F and $\rho \in (0,1)$ that

$$-F(x_{k_0}+\beta\rho^i d_{k_0})^T d_{k_0} < 0, \tag{18}$$

which contradicts (15). The proof is completed. □

The next Lemma indicates that the line search strategy (5) provides a lower bound for steplength α_k.

Lemma 2. *If the iterative sequences $\{x_k\}$ and $\{z_k\}$ are generated by the Algorithm 1. Then, we can obtain that*

$$\alpha_k \geq min\Big\{\beta, \frac{\delta\rho}{(L+\sigma)\|}\frac{\|F_k\|^2}{\|d_k\|^2}\Big\}, \tag{19}$$

where $\delta = 1 - \frac{1}{4\mu}$.

Proof. Tf $\alpha_k = \beta$, it is obviously that (19) holds. Suppose that $\alpha_k \neq \beta$, then we can obtain that $\alpha_k' = \rho^{-1}\alpha_k$ does not satisfy the line search process (5). That is,

$$-F(z_k')^T d_k < \sigma\alpha_k' \min\{\|d_k\|^2, \|F(z_k')\|\|d_k\|^2, -F(x_k')^T d_k\} \leq \sigma\alpha_k'\|d_k\|^2, \tag{20}$$

where $z_k' = x_k + \alpha_k' d_k$. □

From the sufficient descent condition (15), we have that

$$(1-\frac{1}{4\mu})\|F_k\|^2 \doteq \delta\|F_k\|^2 \leq -F_k^T d_k. \tag{21}$$

From the Lipschitz continuity of F (14), (20) and (21), we can obtain that

$$\begin{aligned} -F_k^T d_k &= (F(z_k') - F(x_k))^T d_k - F(z_k')^T d_k \\ &\leq \|F(z_k') - F(x_k)\|\|d_k\| + \sigma\alpha_k'\|d_k\|^2 \\ &\leq L\|z_k' - x_k\|\|d_k\| + \sigma\alpha_k'\|d_k\|^2 \\ &= L\alpha_k'\|d_k\|^2 + \sigma\alpha_k'\|d_k\|^2 \\ &= \rho^{-1}\alpha_k(L+\sigma)\|d_k\|^2. \end{aligned}$$

Therefore, the above inequalities and (21) imply

$$\alpha_k \geq \frac{\delta\rho}{(L+\sigma)\|}\frac{\|F_k\|^2}{\|d_k\|^2}.$$

This shows that Lemma about the search steplength α_k holds.

The next Lemma is proved by Solodov and Svaiter (see Lemma 2.1 in [13]), which can also hold for Algorithm 1. Now, we give this lemma without proof, because its proof is similar in Solodov & Svaiter [13].

Lemma 3. *Assume the function F is monotone and the Lipschitz continuous condition (14) holds. If the iterative sequences $\{x_k\}$ is generated by the Algorithm 1, then for any x^*, such that $F(x^*) = 0$, we can obtain that*

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

In particular, the iterative sequence $\{x_k\}$ is bounded and

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 < \infty. \tag{22}$$

Remark 1. *The above Lemma 2 confirms that the sequence $\{\|x_k - x^*\|\}$ decreases with k. In addition, (22) implies that*

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{23}$$

Theorem 2. *If the iterative sequences $\{x_k\}$ is generated by the Algorithm 1, then, we can have that*

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0. \tag{24}$$

Proof. We can obtain from (5) and (9) that, for any k,

$$\|x_{k+1} - x_k\| = \frac{|F(z_k)^T(x_k - z_k)|}{\|F(z_k)\|} = \frac{-\alpha_k F(z_k)^T d_k}{\|F(z_k)\|} \geq \sigma \alpha_k^2 \|d_k\|^2. \tag{25}$$

In particular, it follows from (23) and (25) that

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0. \tag{26}$$

□

Lemma 4. *If the iterative sequences $\{x_k\}$ is generated by the Algorithm 1, and x^* satisfies $F(x^*) = 0, z_k^{'} = x_k + \alpha_k^{'} d_k, \alpha_k^{'} = \rho^{-1}\alpha_k$. Then, $\{\|F(z_k^{'})\|\}$ and $\{\|F_k\|\}$ are bounded, i.e, there is a constant $M \geq 0$, such that*

$$\|F(z_k^{'})\| \leq M, \quad \|F_k\| \leq M. \tag{27}$$

Proof. By the Lemma 2, we have

$$\|x_k - x^*\| \leq \|x_0 - x^*\|.$$

From (26), we have that there is a constant $M_1 > 0$ such that $\alpha_k \|d_k\| \leq M_1$. Hence

$$\|z_k^{'} - x^*\| \leq \|x_k - x^*\| + \alpha_k^{'} \|d_k\| \leq \|x_0 - x^*\| + \rho^{-1}\alpha_k \|d_k\| \leq \|x_0 - x^*\| + M_1. \tag{28}$$

Since the function $F(x)$ is Lipschitz continuous, we can easily obtain the following two inequalities

$$\|F(z_k^{'})\| \leq \|F(z_k^{'}) - F(x^*)\| \leq L\|z_k^{'} - x^*\| \leq L(\|x_0 - x^*\| + \rho^{-1} M_1,$$

and

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

Let $M = max\{L\|x_0 - x^*\|, L(\|x_0 - x^*\| + \rho^{-1} M_1\}$. We can obtain (27). □

In what following, we will give the global convergence theorem for our proposed method.

Theorem 3. *If the iterative sequences $\{x_k\}$ is generated by the Algorithm 1. We can obtain that*

$$\liminf_{k\to\infty} \|F_k\| = 0. \tag{29}$$

Proof. We will prove this Theorem by contradiction. Assume that (29) is not true. Then it implies there is a constant $\varepsilon > 0$, s.t. $\|F_k\| > \varepsilon$.

Since $F_k \neq 0$, we have from (15) that $d_k \neq 0$. Hence, the monotonicity of F together with (10) implies

$$\begin{aligned} \bar{s}_{k-1}^T w_{k-1} &= \langle F(z_{k-1}) - F(x_{k-1}), z_{k-1} - x_{k-1} \rangle + \gamma \bar{s}_{k-1}^T \bar{s}_{k-1} \\ &\geq \gamma \bar{s}_{k-1}^T \bar{s}_{k-1}. \end{aligned}$$

This together with the definition of $\bar{s}_{k-1}$ implies

$$d_{k-1}^T w_{k-1} \geq \gamma \alpha_{k-1} \|d_{k-1}\|^2. \tag{30}$$

We have from (11), (12) and (30) that

$$\begin{aligned} |\beta_k^{\text{NHZ}}| &= \left| \frac{F_k^T y_{k-1}}{d_{k-1}^T w_{k-1}} - \mu \frac{\|y_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2} F_k^T d_{k-1} \right| \\ &\leq \frac{L\alpha_{k-1}\|d_{k-1}\|\|F_k\|}{\gamma \alpha_{k-1}\|d_{k-1}\|^2} + \mu \frac{L^2 \alpha_{k-1}^2 \|d_{k-1}\|^2}{\gamma^2 \alpha_{k-1}^2 \|d_{k-1}\|^4} \|F_k\| \|d_{k-1}\| \\ &\leq \left(\frac{L}{\gamma} + \mu \frac{L^2}{\gamma^2}\right) \frac{\|F_k\|}{\|d_{k-1}\|}. \end{aligned}$$

Therefore, from (10) and (30), we can obtain

$$\begin{aligned} \|d_k\| &\leq \|F_k\| + |\beta_k^{\text{NHZ}}| \|d_{k-1}\| \\ &\leq \|F_k\| + \left(\frac{L}{\gamma} + \mu \frac{L^2}{\gamma}\right) \|F_k\| \\ &\leq \left(1 + \frac{L}{\gamma} + \mu \frac{L^2}{\gamma^2}\right) M. \end{aligned}$$

Define $C = (1 + \frac{L}{\gamma} + \mu \frac{L^2}{\gamma^2}) M$. Then, we can have $\|d_k\| \leq C$. □

It follows from Lemmas 2, Lemmas 3, $\|F_k\| \geq \varepsilon$ and $\|d_k\| \geq \varepsilon$ that for all k sufficiently large,

$$\begin{aligned} \alpha_k \|d_k\| &\geq \min\left\{\beta, \frac{\delta \rho}{(L+\sigma)\|} \frac{\|F_k\|^2}{\|d_k\|^2}\right\} \|d_k\| \\ &\geq \min\left\{\beta\varepsilon, \frac{\delta \rho \varepsilon^2}{(L+\sigma)\|C}\right\} > 0. \end{aligned}$$

It is obvious that the above inequality contradicts with (24). That is, (29) holds. And we complete this proof.

4. Numerical Experiments

Now, we will give some numerical experiments to test the numerical performance of our proposed method. We try to test the NHZ Algorithm 1 and compare it's performance with the spectral gradient

(SG) method [15] and the MPRP method in [39]. In the testing experiments, all codes were written in Matlab R2018a, and run on a Lenovo PC with 4 GB RAM memory.

To obtain better numerical performance, we select the following initial steplength as in [39] and [43]

$$\beta = \left|\frac{F(x_k)^T d_k}{d_k^T (F(x_k + \epsilon d_k) - F(x_k))/\epsilon}\right| \approx \left|\frac{F(x_k)^T d_k}{d_k^T \nabla F(x_k) d_k}\right|, \quad \epsilon = 10^{-8}. \tag{31}$$

We set $\rho = 0.5, \sigma = 2$. Further, let $\beta = 1$ if $\beta < 10^{-4}$.

Following the MPRP method in [39], we terminate the iterative process if the following condition

$$\min\{\|F(x_k)\|, \|F(z_k)\|\} \le atol + rtol\|F(x_0)\|$$

is satisfied, and $rtol = atol = 10^{-4}$.

The numerical performance of SG, MPRP, and NHZ methods are tested by using the following five nonlinear monotone equations problem with different various sizes and initial points.

Problem 1 ([49]). *The specific expression of the function $F(x)$ is defined as*

$$F_i(x) = 2x_i - sin(|x_i|), \quad i = 1, \dots, n.$$

Problem 2 ([49]). *The specific expression of the function $F(x)$ is defined as*

$$\begin{aligned} F_1(x) &= 2x_1 + sin(x_1) - 1, \\ F_i(x) &= -2x_{i-1} + 2x_i + sin(x_i) - 1, \quad i = 2, \dots, n-1, \\ F_n(x) &= 2x_n + sin(x_n) - 1. \end{aligned}$$

Problem 3 ([50]). *The specific expression of the function $F(x)$ is defined as*

$$F_i(x) = x_i - sin(x_i), \quad i = 1, \dots, n.$$

Problem 4 ([50]). *The specific expression of the function $F(x)$ is defined as*

$$F(x) = Ax + g(x)$$

where $g(x) = (e^{x_1} - 1, e^{x_2} - 1, \dots, e^{x_n} - 1)^T$

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}$$

Problem 5. *The specific expression of the function $F(x)$ is defined as*

$$F(x) = Ax + |X| - B$$

where $|X| = (|x_1|, |x_2|, \ldots, |x_n|)^T, B = (1, 1, \ldots, 1)^T$, *and*

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

The numerical results for 5 tested problems are reported in Tables 1–5 respectively, where the given initial points are $x_1 = (0.1, \ldots, 0.1)^T, x_2 = (1, \ldots, 1)^T, x_3 = (1, \frac{1}{2}, \ldots, \frac{1}{n})^T, x_4 = (-10, \ldots, -10)^T$, $x_5 = (-0.1, \ldots, -0.1)^T, x_6 = (-1, \ldots, -1)^T$. Meanwhile, the numerical results are listed in the Tables 1–5 where (**Time**) is the CPU time (in seconds), (**Iter**) is the number of iterations and (**Feval**) is the number of function evaluations.

Tables 1–5 report the numerical results of the proposed algorithm, the spectral gradient (SG) method [15] and the MPRP method in [39] with five tested problems where the test indicators of **Time**, **Iter** and **Feval** are used to evaluate numerical performance.

Comparing the CPU time within the tested algorithms, we note that our proposed algorithm needs lower time consuming than both the spectral gradient (SG) method [15] and the MPRP method in [39] for all tested problems, and the difference is substantial and significant especially for large-scale problems. In addition, the MPRP method in [39] needs lower CPU time than the spectral gradient (SG) method [15]. Assessing the number of iterations within the tested algorithms, we find that the NHZ method requires fewer iterations than the spectral gradient (SG) method [15] and the MPRP method in [39] for all tested problems. We also note that our proposed algorithm requires fewer number of function evaluations for all tested problems, and the difference is substantial and significant.

In sum, from the numerical results in Tables 1–5, it isn't difficult to see that the proposed algorithm performs better than the spectral gradient (SG) method [15] and the MPRP method in [39] for three test indicators which implies that the modified Hestenes-Stiefel-based derivative-free method is computationally efficient for nonlinear monotone equations.

Table 1. Numerical results for the tested Problem 1 with various sizes and given initial points.

Initial	Dim.	SG			MPRP			NHZ		
		Time	**Iter**	**Feval**	**Time**	**Iter**	**Feval**	**Time**	**Iter**	**Feval**
x_1	1000	1.16	16	37	0.81	12	39	0.65	10	29
x_2	1000	1.16	16	37	0.81	12	39	0.65	10	29
x_3	1000	0.77	13	26	0.55	8	27	0.48	7	22
x_4	1000	1.24	15	43	0.85	9	42	0.78	7	35
x_5	1000	1.15	14	44	0.53	6	28	0.48	5	24
x_6	1000	1.15	14	44	0.53	6	28	0.48	5	24
x_1	5000	6.96	18	37	5.32	13	42	4.55	11	32
x_2	5000	6.96	17	37	5.01	12	39	3.85	11	29
x_3	5000	4.16	11	22	3.01	7	24	2.43	6	20
x_4	5000	10.45	18	62	7.38	12	60	5.99	11	48
x_5	5000	6.88	14	44	3.40	6	28	2.84	5	24
x_6	5000	6.96	15	45	3.40	6	28	2.84	5	24
x_1	10,000	27.77	18	38	21.15	13	42	15.72	11	32
x_2	10,000	27.62	17	36	19.65	12	39	14.36	11	27
x_3	10,000	14.90	10	20	11.92	7	24	8.96	6	20
x_4	10,000	44.65	20	69	35.15	14	72	30.98	12	68
x_5	10,000	29.55	15	45	13.86	6	28	11.22	5	24
x_6	10,000	29.55	15	45	13.86	6	28	11.22	5	24

Table 2. Numerical results for the tested Problem 2 with various sizes and given initial points.

Initial	Dim.	SG			MPRP			NHZ		
		Time	Iter	Feval	Time	Iter	Feval	Time	Iter	Feval
x_1	1000	0.67	326	646	0.17	196	591	0.12	155	568
x_2	1000	0.64	364	714	0.11	203	612	0.10	180	584
x_3	1000	0.64	343	691	0.19	195	588	0.17	174	555
x_4	1000	1.18	468	941	0.15	243	741	0.14	220	709
x_5	1000	0.67	479	988	0.17	194	585	0.15	162	549
x_6	1000	0.65	431	857	0.16	189	571	0.14	165	529
x_1	5000	10.38	723	2047	8.19	487	1465	8.10	468	1448
x_2	5000	16.87	753	2622	12.85	769	2310	12.63	744	2205
x_3	5000	9.01	824	2849	7.98	469	1411	7.65	442	1324
x_4	5000	18.74	1023	3261	15.96	929	2840	15.51	901	2781
x_5	5000	10.05	1226	3053	7.59	453	1363	7.50	442	1320
x_6	5000	11.32	836	2476	6.32	371	1122	6.20	338	1103
x_1	10,000	47.90	960	2002	34.59	539	1622	10.01	460	1508
x_2	10,000	82.62	1334	4066	65.06	1023	3072	60.79	1001	3003
x_3	10,000	50.99	833	2469	34.30	516	1553	32.32	501	1502
x_4	10,000	56.82	2042	6668	39.65	1668	5075	36.31	1602	5003
x_5	10,000	49.63	832	2268	31.57	497	1497	30.25	436	1405
x_6	10,000	45.70	850	1706	25.36	396	1202	22.24	375	1106

Table 3. Numerical results for the tested Problem 3 with various sizes and given initial points.

Initial	Dim.	SG			MPRP			NHZ		
		Time	Iter	Feval	Time	Iter	Feval	Time	Iter	Feval
x_1	1000	1.16	16	37	0.81	12	39	0.62	10	28
x_2	1000	1.17	17	36	0.83	12	39	0.71	11	28
x_3	1000	0.77	11	24	0.57	8	27	0.49	7	28
x_4	1000	1.25	14	44	0.88	9	42	0.75	7	32
x_5	1000	1.16	13	42	0.56	6	28	0.48	5	22
x_6	1000	1.16	13	42	0.57	6	28	0.48	5	22
x_1	5000	6.98	17	36	5.42	13	42	4.63	11	32
x_2	5000	6.98	17	36	5.11	12	39	3.95	11	30
x_3	5000	4.29	10	22	3.12	7	24	2.34	6	20
x_4	5000	10.57	19	64	7.46	12	60	6.25	11	52
x_5	5000	6.99	13	42	3.52	6	28	3.92	5	24
x_6	5000	6.99	13	42	3.52	6	28	3.92	5	24
x_1	10,000	27.78	17	36	21.35	13	42	15.97	11	32
x_2	10,000	27.79	17	36	19.75	12	39	15.86	11	30
x_3	10,000	15.65	9	26	11.99	7	24	9.98	6	19
x_4	10,000	44.85	20	69	35.36	14	72	29.98	12	60
x_5	10,000	29.89	14	45	13.98	6	28	12.56	5	24
x_6	10,000	29.89	14	45	13.98	6	28	13.59	6	24

Table 4. Numerical results for the tested Problem 4 with various sizes and given initial points.

Initial	Dim.	SG			MPRP			NHZ		
		Time	Iter	Feval	Time	Iter	Feval	Time	Iter	Feval
x_1	1000	0.20	219	431	0.06	50	216	0.05	38	168
x_2	1000	0.28	261	463	0.06	56	252	0.05	46	185
x_3	1000	0.28	224	329	0.05	34	152	0.03	32	137
x_4	1000	0.22	263	529	0.07	100	421	0.06	96	399
x_5	1000	0.28	183	403	0.06	42	187	0.05	40	177
x_6	1000	0.28	212	424	0.06	60	261	0.05	48	218
x_1	5000	2.15	263	456	1.11	48	209	1.05	47	183
x_2	5000	2.45	225	378	1.19	46	224	0.92	38	169
x_3	5000	1.65	122	267	0.62	27	117	0.65	29	128
x_4	5000	3.41	265	558	2.59	109	483	2.49	104	455
x_5	5000	2.86	290	467	1.21	53	231	1.07	44	189
x_6	5000	2.97	231	477	1.20	54	234	1.16	48	213
x_1	10,000	5.30	278	502	3.96	45	195	3.83	42	185
x_2	10,000	6.26	237	574	4.27	41	210	3.84	38	158
x_3	10,000	5.62	275	585	1.93	42	96	2.62	35	142
x_4	10,000	18.15	333	596	10.86	117	533	9.45	109	498
x_5	10,000	13.52	341	595	4.34	49	212	3.85	44	186
x_6	10,000	13.55	336	553	4.89	56	246	3.78	48	195

Table 5. Numerical results for the tested Problem 5 with various sizes and given initial points.

Initial	Dim.	SG			MPRP			NHZ		
		Time	Iter	Feval	Time	Iter	Feval	Time	Iter	Feval
x_1	1000	0.89	119	289	0.66	47	199	0.44	38	168
x_2	1000	0.78	122	263	0.45	22	105	0.44	24	98
x_3	1000	0.69	130	235	0.35	48	209	0.28	38	120
x_4	1000	0.85	190	249	0.47	37	165	0.34	35	98
x_5	1000	0.75	194	248	0.55	94	237	0.45	66	192
x_6	1000	1.22	225	462	0.79	174	396	0.75	142	372
x_1	5000	2.32	113	260	1.22	51	221	0.98	42	172
x_2	5000	2.92	128	270	0.56	22	105	0.58	28	96
x_3	5000	3.80	228	412	1.11	47	200	0.79	44	144
x_4	5000	3.50	216	424	1.20	48	206	0.79	44	142
x_5	5000	3.00	226	443	1.17	47	206	0.81	44	122
x_6	5000	6.57	461	881	5.06	308	707	4.25	262	628
x_1	10,000	5.92	66	209	3.90	44	191	3.42	38	184
x_2	10,000	6.86	68	218	51.1	22	105	49.1	21	98
x_3	10,000	5.76	60	181	4.24	47	204	3.23	38	132
x_4	10,000	11.84	69	227	10.5	48	209	8.52	44	148
x_5	10,000	10.55	68	221	4.02	45	196	3.82	42	168
x_6	10,000	12.46	89	326	10.1	74	262	7.83	68	232

5. Conclusions

This paper aims to present a modified Hestenes-Stiefel method to solve the nonlinear monotone equations which combines the hyperplane projection method [13] and the modified Hestenes-Stiefel method in Dai and Wen [47]. In the proposed method, the search direction satisfies sufficient descent conditions. A new line search is proposed for the derivative-free method. Under appropriate conditions, the proposed method converges globally. The given numerical results show the presented method is more efficient compared to the methods proposed by the spectral gradient method Zhang and Zhou [15] and the MPRP method in Li and Li [39].

In addition, we also expect that our proposed method and its further modifications could produce new applications for problems in relevant areas of symmetric equations [51], image processing [52], and finance [53–55].

Author Contributions: Conceptualization, Z.D.; methodology, Z.D. and H.Z.; software, Z.D. and H.Z.; validation, Z.D.; formal analysis, Z.D.; investigation, Z.D.; resources, Z.D. and H.Z.; data curation, Z.D.; writingoriginal draft preparation, Z.D. and H.Z.; writing-review and editing Z.D. and H.Z.; visualization, Z.D. and H.Z.; supervision, Z.D. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the NSF of China granted 71771030, 11301041, and fund of Hunan Provincial Education Department granted 19A007.

Acknowledgments: This work was supported by the National Natural Science Foundation of China, grants 71771030, and 11301041; and the Scientific Research Fund of Hunan Provincial Education Department, grant number 19A007.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Iusem, A.N.; Solodov, M.V. Newton-type methods with generalized distances for constrained optimization. *Optimization* **1997**, *41*, 257–278. [CrossRef]
2. Zhao, Y.B.; Li, D.H. Monotonicity of fixed point and normal mapping associated with variational inequality and its application. *SIAM J. Optim.* **2001**, *4*, 962–973. [CrossRef]
3. Figueiredo, M.; Nowak, R.; Wright, S.J. Gradient projection for sparse reconstruction, application to compressed sensing and other inverse problems. *IEEE J-STSP* **2007**, *1*, 586–597. [CrossRef]
4. 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]
5. Shang, Y. Vulnerability of networks: Fractional percolation on random graphs. *Phys. Rev. E* **2014**, *89*, 012813. [CrossRef]
6. Shang, Y. Super Connectivity of Erdos-Renyi Graphs. *Mathematics* **2019**, *7*, 267. [CrossRef]
7. Brown, P.N.; Saad, Y. Convergence theory of nonlinear Newton-Krylov algorithms, *SIAM J. Optim.* **1994**, *4*, 297–330.
8. Gasparo, M. A nonmonotone hybrid method for nonlinear systems. *Optim. Methods Softw.* **2000**, *13*, 79–94. [CrossRef]
9. Griewank, A. The global convergence of Broyden-like methods with suitable line search. *J. Austral. Math. Soc. Ser. B* **1996**, *28*, 75–92. [CrossRef]
10. Li, D.H.; Fukushima, M. A modified BFGS method and its global convergence in nonconvex minimization. *J. Comput. Appl. Math.* **2001**, *129*, 15–35. [CrossRef]
11. Li, D.H.; Fukushima, M. A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. *Optim. Methods Softw.* **2000**, *13*, 583–599. [CrossRef]
12. Martínez, J.M. A family of quasi-Newton methods for nonlinear equations with direct secant updates of matrix factorizations. *SIAM J. Numer. Anal.* **1990**, *27*, 1034–1049. [CrossRef]
13. Solodov, M.V.; Svaiter, B.F. A globally convergent inexact Newton method for systems of monotone equations. In *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*; Fukushima, M., Qi, L., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1998; pp. 355–369.
14. Barzilai, J.; Borwein, J.M. Two-point step size gradient methods. *IMA J. Numer. Anal.* **1998**, *8*, 141–148. [CrossRef]
15. Zhang, L.; Zhou, W.J. Spectral gradient projection method for solving nonlinear monotone equations. *J. Comput. Appl. Math.* **2006**, *196*, 478–484. [CrossRef]
16. Yu, Z.S.; Ji, L.N.; Sun, J.; Xiao, Y.H. Spectral gradient projection method for monotone nonlinear equations with convex constraints. *Appl. Numer. Math.* **2009**, *59*, 2416–2423. [CrossRef]
17. Hager, W.W.; Zhang, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **2005**, *16*, 170–192. [CrossRef]
18. Zhang, L.; Zhou, W.J.; 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]

19. Cheng, W.Y. A two-term PRP-based descent method. *Numer. Funct. Anal. Optim.* **2007**, *28*, 1217–1230. [CrossRef]
20. Zhang, L.; Zhou, W.J.; Li, D.H. Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search. *Numer. Math.* **2006**, *104*, 561–572. [CrossRef]
21. Dai, Z.; Zhu, H. Stock return predictability from a mixed model perspective. *Pac-Basin. Finac. J.* **2020**, *60*, 101267. [CrossRef]
22. Narushima, Y.; Yabe, H.; Ford, J.A. A three-term conjugate gradient method with sufficient descent property for unconstrained optimization. *SIAM J. Optim.* **2011**, *21*, 212–230. [CrossRef]
23. Andrei, N. A simple three-term conjugate gradient algorithm for unconstrained optimization. *J. Comp. Appl. Math.* **2013**, *241*, 19–29. [CrossRef]
24. Andrei, N. On three-term conjugate gradient algorithms for unconstrained optimization. *Appl. Math. Comput.* **2013**, *219*, 6316–6327. [CrossRef]
25. Liu, Z.X.; Liu, H.W.; Dong, X.L. A new adaptive Barzilai and Borwein method for unconstrained optimization. *Optim. Lett.* **2018**, *12*, 845–873. [CrossRef]
26. Babaie-Kafaki, S.; Reza, G. The Dai-Liao nonlinear conjugate gradient method with optimal parameter choices. *Eur. J. Oper. Res.* **2014**, *234*, 625–630. [CrossRef]
27. Babaie-Kafaki, S. On optimality of the parameters of self-scaling memoryless quasi-Newton updating formulae. *J. Optim. Theory Appl.* **2015**, *167*, 91–101 [CrossRef]
28. Yuan, G. Zhang, M. A three-terms Polak-Ribiére-Polyak conjugate gradient algorithm for large-scale nonlinear equations. *J. Comput. Appl. Math.* **2015**, *286*, 186–195. [CrossRef]
29. Yuan, G.; Meng, Z.H.; Li, Y. A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations. *J. Optim. Theory. Appl.* **2016**, *168*, 129–152. [CrossRef]
30. Dong, X.; Han, D.; Reza, G.; Li, X.; Dai, Z. Some new three-term Hestenes-Stiefel conjugate gradient methods with affine combination. *Optimization* **2017**, *66*, 759–776. [CrossRef]
31. Dong, X.; Han, D.; Dai, Z.; Li, L.; Zhu, J. An accelerated three-term conjugate gradient method with sufficient descent condition and conjugacy condition. *J. Optim. Theory Appl.* **2018**, *179*, 944–961. [CrossRef]
32. Li, M.; Feng, H. A sufficient descent Liu-Storey conjugate gradient method for unconstrained optimization problems. *Appl Math Comput.* **2011**, *218*, 1577–1586.
33. Dai, Z.; Wen, F. Another improved Wei-Yao-Liu nonlinear conjugate gradient method with sufficient descent property. *Appl Math Comput.* **2012**, *218*, 4721–4730. [CrossRef]
34. Dai, Z.; Tian, B. Global convergence of some modified PRP nonlinear conjugate gradient methods. *Optim. Lett.* **2011**, *5*, 615–630. [CrossRef]
35. Dai, Z. Comments on a new class of nonlinear conjugate gradient coefficients with global convergence properties. *Appl. Math. Computat.* **2016**, *276*, 297–300. [CrossRef]
36. Dai, Z.; Zhou, H.; Wen, F.; He, S. Efficient predictability of stock return volatility: The role of stock market implied volatility. *N. Am. J. Econ. Finance.* **2020**, forthcoming. [CrossRef]
37. Cheng, W.Y. A PRP type method for systems of monotone equations. *Math. Comput. Model.* **2009**, *50*, 15–20. [CrossRef]
38. Yu, G. A derivative-free method for solving large-scale nonlinear systems of equations. *J. Ind. Manag. Optim.* **2010**, *6*, 149–160. [CrossRef]
39. Li, Q.; Li, D.H. A class of derivative-free methods for large-scale nonlinear monotone equations. *IMA J. Numer. Anal.* **2011**, *31*, 1625–1635. [CrossRef]
40. Yu, G. Nonmonotone spectral gradient-type methods for large-scale unconstrained optimization and nonlinear systems of equations. *Pac. J. Optim.* **2011**, *7*, 387–404 .
41. Zhou, W.; Shen, D. An inexact PRP conjugate gradient method for symmetric nonlinear equations. *Numer. Funct. Anal. Optim.* **2014**, *35*, 370–388. [CrossRef]
42. Sun, M.; Wang, X.; Feng, D. A family of conjugate gradient methods for large-scale nonlinear equations. *J. Inequal. Appl.* **2017**, *236*, 1–8.
43. Zhou, W.; Wang, F. A PRP-based residual method for large-scale monotone nonlinear equations. *Appl. Math. Comput.* **2015**, *261*, 1–7. [CrossRef]
44. Dai, Z.; Chen, X.; Wen, F. A modified Perry's conjugate gradient method-based derivative-free method for solving large-scale nonlinear monotone equation. *Appl. Math. Comput.* **2015**, *270*, 378–386. [CrossRef]

45. Li, M. A Liu-Storey type method for solving large-scale nonlinear monotone equations. *Numer. Funct. Anal. Optim.* **2014**, *35*, 310–322. [CrossRef]
46. Hestenes, M.R.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409–436. [CrossRef]
47. Dai, Z.; Wen, F. Global convergence of a modified Hestenes-Stiefel nonlinear conjugate gradient method with Armijo line search. *Numer Algor.* **2012**, *59*, 79–93. [CrossRef]
48. Yan, Q.R.; Peng, X.Z.; Li, D.H. A globally convergent derivative-free method for solving large-scale nonlinear monotone equations. *J. Comput. Appl. Math.* **2010**, *234*, 649–657. [CrossRef]
49. Zhou, W.J.; Li, D.H. Limited memory BFGS method for nonlinear monotone equations. *J. Comput. Math.* **2007**, *25*, 89–96.
50. Zhou, W.J.; Li, D.H. A globally convergent BFGS method for nonlinear monotone equations without any merit functions. *Math. Comput.* **2008**, *77*, 2231–2240. [CrossRef]
51. Zhou, W.J. A modified BFGS type quasi-Newton method with line search for symmetric nonlinear equations problems. *J. Comput. Appl. Math.* **2020**, *357*, 454. [CrossRef]
52. Gao, P.T.; He, C.; Liu, Y. An adaptive family of projection methods for constrained monotone nonlinear equations with applications. *Appl. Math. Comput.* **2019**, *359*, 1–16. [CrossRef]
53. Dai, Z.; Zhu, H. Forecasting stock market returns by combining sum-of-the-parts and ensemble empirical mode decomposition. *Appl. Econ.* **2019**. [CrossRef]
54. Dai, Z.; Zhu, H.; Wen, F. Two nonparametric approaches to mean absolute deviation portfolio selection model, *J. Ind. Manag. Optim.* **2019**. [CrossRef]
55. Dai, Z.; Zhou, H. Prediction of stock returns: Sum-of-the-parts method and economic constraint method. *Sustainability* **2020**, *12*, 541. [CrossRef]