

Accepted Manuscript

A norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations

J.K. Liu, Y.M. Feng

PII: S0377-0427(18)30276-0
DOI: <https://doi.org/10.1016/j.cam.2018.05.006>
Reference: CAM 11670

To appear in: *Journal of Computational and Applied Mathematics*

Received date : 20 October 2017
Revised date : 19 April 2018

Please cite this article as: J.K. Liu, Y.M. Feng, A norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations, *Journal of Computational and Applied Mathematics* (2018), <https://doi.org/10.1016/j.cam.2018.05.006>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



A norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations

J. K. LIU^{a,b}, Y. M. FENG^b

^a School of Mathematics and Statistics,
Chongqing Three Gorges University, Chongqing, 404100, China

^b School of Mathematics and Statistics,
Chongqing Three Gorges University, Chongqing, 404100, China.

Abstract

In this paper, we propose a norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations without involving any information of the gradient or Jacobian matrix by using some approximate substitutions. The proposed algorithm is extended from an efficient three-term conjugate gradient method for solving unconstrained optimization problems, and inherits some nice properties such as simple structure, low storage requirements and symmetric property. Under some appropriate conditions, the global convergence is proved. Finally, the numerical experiments and comparisons show that the proposed algorithm is very effective for large-scale problems.

Keywords: Nonlinear symmetric equations, Derivative-free method, Conjugate gradient method, Global convergence.

1. Introduction

In this paper, we mainly consider finding the solutions of the following nonlinear symmetric equations

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

where $F : R^n \rightarrow R^n$ is a continuously differentiable mapping, and its Jacobian $J(x) = \nabla F(x)$ is symmetric, i.e., $J(x) = J(x)^T$. This equations originates from many practical problems such as the KKT systems of equality constrained optimization, the discredited two-point boundary value problem, the discredited elliptic boundary value problem, the saddle points problems, and finding a stationary point for unconstrained optimization problem where F is the corresponding gradient of the objective function, etc.

¹This research was partially supported by Chongqing Research Program of Basic Research and Frontier Technology (Grant number: cstc2017jcyjAX0318), Chongqing Three Gorges University (Grant number: 14ZD-14), Program for Innovation Team Building at Institutions of Higher Education in Chongqing (Grant number: CXTDX201601035) and Project Supported by Chongqing Municipal Key Laboratory of Institutions of Higher Education (Grant No. [2017]3).

* Corresponding author. Email address: liujinkui2006@126.com (J.K.Liu)

Nonlinear symmetric equations is a special type of nonlinear equations. In the last decade, numerical methods for solving nonlinear equations, especially nonlinear monotone equations, have been developed, analyzed, tested and established. Among these methods, Newton method, quasi-Newton method, Gauss-Newton method, Levenberg-Marquart method and their various variants [1]-[7] are very popular because they converge rapidly from efficiently initial points. However, to obtain the next search direction, they need to solve a linear equations using the Jacobian matrix or the approximation of the Jacobian matrix at each iteration. This means that these methods are hard to be designed for solving large-scale problems. Recent years, some conjugate gradient methods and spectral gradient methods have been extended to solve large-scale nonlinear equations, see Refs. [8]-[17].

Most Recently, some different methods have been proposed for solving nonlinear symmetric equations (1.1). By using the symmetry of $J(x)$, based on BFGS method, Li and Fukushima [18] proposed a Gauss-Newton method, and established its global convergence and local superlinear convergence. However, this method is only approximate norm descent. Gu et al.[19] extended the method proposed in [18] and established a norm descent BFGS method. It is a pity that this quasi-Newton method is not suitable to solve large-scale problems. Li and Wang [20] proposed a modified Fletcher-Reeves derivative-free method for large-scale symmetric nonlinear equations, which is an extension of the modified Fletcher-Reeves conjugate gradient method [21]. This method has good global convergence, but its numerical efficiency is not satisfactory. Xiao et al.[22] developed, analyzed and validated a family of derivative-free methods for symmetric equations with the small size of the variable x , which are based on the state-of-art descent conjugate gradient methods proposed in [23] for unconstrained optimization. Xiao et al. proved the global convergence of proposed methods under some appropriate conditions, and showed its effectiveness by a lot of numerical results. Zhou and Shen [26] presented an effective iterative method for solving relatively large-scale symmetric equations without computing Jacobian or gradient of the underlying function. This method can be viewed as an extension of the three-term PRP conjugate gradient method [27] for solving unconstrained optimization problems.

In [28], Liu and Li proposed an efficient three-term conjugate gradient method for solving unconstrained optimization problems, in which the search direction d_k can be written as

$$d_k = -\left(I - \frac{d_{k-1}y_{k-1}^T}{d_{k-1}^T y_{k-1}}\right)\left(I - \frac{y_{k-1}d_{k-1}^T}{d_{k-1}^T y_{k-1}}\right)g_k = -A_k g_k.$$

The conjugate iterative matrix A_k is obviously symmetric. Thus, based on the reference [26], in this paper we try to use this symmetric property to establish a norm descent derivative-free algorithm for solving large-scale symmetric equations. The main feature distinguished from other algorithms is that the iterative matrix of the proposed algorithm is symmetric. Moreover, the gradient or Jacobian information of the underlying function is needless in the full iteration process. By using some derivative-free backtracking line search technique, the generated sequence of function values is decreasing. Under some appropriate conditions, the proposed algorithm is globally convergent. Finally, we give some numerical results and performance comparisons to show that the proposed algorithm is effective and encouraging.

In the next section, we propose the norm descent derivative-free algorithm. Section 3 is devoted to proving its global convergence. In Section 4, we report some numerical results and performance comparisons. Throughout this paper, all vectors are column vectors, and $\|\cdot\|$ denotes the Euclidean norm of a vector. For convenience, we abbreviate $F(x_k)$ and $J(x_k)$ as F_k and J_k , respectively.

2. Algorithm

In this section, we first simply recall one three-term conjugate gradient method [28] for solving the following unconstrained optimization problem

$$\min f(x), x \in R^n, \quad (2.1)$$

where $f : R^n \rightarrow R$ is smooth, and its gradient ∇f is available. This method generates a sequence $\{x_k\}$ satisfying

$$x_{k+1} = x_k + \alpha_k d_k, k \geq 0, \quad (2.2)$$

where the step-size α_k is positive and obtained by some line searches, and the search direction d_k is defined as

$$d_k = \begin{cases} -\nabla f(x_k), & \text{if } k = 0, \\ -\nabla f(x_k) + \beta_k d_{k-1} + \theta_k \hat{y}_{k-1}, & \text{otherwise,} \end{cases} \quad (2.3)$$

where $\hat{y}_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$, β_k and θ_k are computed as:

$$\beta_k = \frac{\nabla f(x_k)^T \hat{y}_{k-1}}{d_{k-1}^T \hat{y}_{k-1}} - \frac{\|\hat{y}_{k-1}\|^2 \nabla f(x_k)^T d_{k-1}}{(d_{k-1}^T \hat{y}_{k-1})^2}, \theta_k = \frac{\nabla f(x_k)^T d_{k-1}}{d_{k-1}^T \hat{y}_{k-1}}.$$

An attractive property of this method is that its search direction d_k satisfies

$$d_k^T \nabla f(x_k) \leq -\mu \|\nabla f(x_k)\|^2, \mu \in (0, 1),$$

which is independent of any line search. This method is globally convergent for non-convex optimization if the strong Wolfe line search is used. Moreover, the performance comparisons with some three-term conjugate gradient methods [24]-[25] is encouraging.

In the following, we turn our attention to solving the symmetric nonlinear equation (1.1). Assuming that f in (2.1) is specific as

$$f(x) = \frac{1}{2} \|F(x)\|^2, \quad (2.4)$$

which means problem (1.1) is equivalent to the global optimization problem (2.1). Noting that $F(x)$ is derivative and $J(x)$ is symmetric, it holds that

$$\nabla f(x) = J(x)^T F(x) = J(x)F(x).$$

Thus the calculation methods related with the information of gradient or Jacobian matrix are not suitable for such large-scale problems whose gradient or Jacobian matrix is not available or very difficult to compute. Li and Fukushima [18] suggested Newton method generates a sequence $\{x_k\}$ by the iterative process (2.2), where d_k is computed by the following linear equations:

$$J_k d + F_k = 0. \quad (2.5)$$

If J_k is nonsingular, it follows from the symmetry of J that $\nabla f(x_k)^T d_k = -F_k^T J_k^T J_k^{-1} F_k = -\|F_k\|^2$, which means that d_k is a descent direction of f at x_k . The conventional quasi-Newton methods use a matrix B_k instead of J_k in (2.5), i.e.,

$$B_k d + F_k = 0.$$

Although the conventional quasi-Newton methods preserve some nice properties without requiring exact Jacobian matrix, it is a pity that d_k is not necessarily a descent direction even though B_k is nonsingular. For this reason, Li and Fukushima [18] proposed a novel strategy to determine the search direction. Notice that (2.5) is equivalent to the following linear equations

$$J_k J_k^T d + J_k F_k = 0, \quad (2.6)$$

if J_k is nonsingular. This implies that Newton direction coincides with Gauss-Newton direction, i.e.,

$$d_k = -(J_k J_k^T)^{-1} J_k F_k,$$

which as mentioned above is a descent direction of f at x_k . Specially, if $J_k J_k^T$ in (2.6) is replaced by a positive definite matrix B_k , then the search direction determined by (2.6) satisfies $\nabla f(x_k)^T d_k = -F_k^T B_k^{-1} J_k F_k < 0$, which shows that d_k is also a descent direction of f at x_k . However, (2.6) still contains J_k which should be avoided. To cope with this problem, they used the term g_k as the approximate substitutions of $\nabla f(x_k)$, i.e.

$$g_k = \frac{F(x_k + \alpha_{k-1} F_k) - F(x_k)}{\alpha_{k-1}}, \quad (2.7)$$

where α_{k-1} is the step-size at the previous iteration. By using the differential mean value theorem, we have

$$g_k = J(x_k + \alpha_{k-1} t F_k) F(x_k), t \in (0, 1).$$

Then from Assumption H(ii) in the next section, we have

$$\|g_k - \nabla f(x_k)\| \leq \|J(x_k + \alpha_{k-1} \theta F_k) - J(x_k)\| \cdot \|F(x_k)\| \leq 2M \|F(x_k)\|,$$

which means that the error between g_k and $\nabla f(x_k)$ is getting smaller and smaller when the sequence $\{x_k\}$ gradually approaches to the solution of the problem (1.1). Thus, $g_k \approx \nabla f(x_k)$ is reasonable, especially when α_{k-1} is sufficiently small. To some extent, the linear equations (2.6) reduces to

$$B_k^T d + \frac{F(x_k + \alpha_{k-1} F_k) - F_k}{\alpha_{k-1}} = 0,$$

where $B_k = J_k J_k^T$. The search direction determined above is without any information of gradient or Jacobian matrix. So it belongs to the derivative-free framework. From [18], letting $\gamma_k = F_{k+1} - F_k$, it holds that

$$\gamma_k = \nabla F(x_k + \theta s_k) s_k \approx J_{k+1} s_k, \theta \in (0, 1), \quad (2.8)$$

where $s_k = x_{k+1} - x_k$. Furthermore, let

$$y_k = F(x_k + \gamma_k) - F_k, \quad (2.9)$$

it gets that

$$y_k = \nabla F(x_k + \vartheta \gamma_k) \gamma_k \approx J_{k+1} \gamma_k \approx J_{k+1} J_{k+1}^T s_k = B_{k+1} s_k, \vartheta \in (0, 1). \quad (2.10)$$

In optimization literatures, B_{k+1} in the conventional quasi-Newton methods usually satisfies the following secant equations, i.e.,

$$B_{k+1} s_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

In addition, from the previous equality we have

$$\begin{aligned}
 \|y_k - \nabla f(x_{k+1}) + \nabla f(x_k)\| &= \|J(x_k + \vartheta\gamma_k)\gamma_k - J_{k+1}F_{k+1} + J_kF_k\| \\
 &= \|(J(x_k + \vartheta\gamma_k) - J_{k+1})F_{k+1} - (J(x_k + \vartheta\gamma_k) - J_k)F_k\| \\
 &\leq \|J(x_k + \vartheta\gamma_k) - J_{k+1}\| \cdot \|F_{k+1}\| + \|J(x_k + \vartheta\gamma_k) - J_k\| \cdot \|F_k\| \\
 &\leq 2M(\|F_{k+1}\| + \|F_k\|),
 \end{aligned}$$

where the second inequality follows from Assumption H(ii) in the next section. It is not difficult to find that the error between y_k and $\nabla f(x_{k+1}) - \nabla f(x_k)$ is upper bounded, and gradually becomes smaller when x_k approaches to the solution of the problem (1.1). Therefore, y_k defined above is the best approximate substitution for the term $\nabla f(x_{k+1}) - \nabla f(x_k)$ at the case of $f(x) = \frac{1}{2}\|F(x)\|^2$.

Based on these observation and analysis, we construct an efficient three-term derivative-free algorithm for solving problem (1.1), which avoids any information of gradient or Jacobian matrix. The search direction d_k is defined as

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} + \theta_k y_{k-1}, & \text{otherwise,} \end{cases} \quad (2.11)$$

where g_k is given in (2.7) and

$$\beta_k = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}, \theta_k = \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}, \quad (2.12)$$

where $y_{k-1} = F(x_{k-1} + \gamma_{k-1}) - F_{k-1}$. Notice that the structures of β_k and θ_k are the same as those in [28], but the parameters y_{k-1} and g_k are obviously different from those in [28].

For the proposed algorithm, we use a non-monotone line search to determine the step-size α_k . Let $\rho, \sigma \in (0, 1)$ and $\eta > 0$ be constants and $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{+\infty} \eta_k \leq \eta < +\infty, \quad (2.13)$$

then $\alpha_k = \max\{\rho^i | i = 0, 1, 2, \dots\}$ satisfies

$$f(x_k + \alpha d_k) \leq f(x_k) - \sigma \|\alpha_k d_k\|^2 + \eta_k. \quad (2.14)$$

On the basis of the previous analysis, we now formally complete the steps of the approximate algorithm for solving problem (1.1) as follows.

Algorithm 2.1:

- Step 0: Choose $\sigma, \rho \in (0, 1)$, the sequence $\{\eta_k\}$, and the initial point $x_0 \in R^n$. Set $k = 0$.
- Step 1: If $\|F_k\| = 0$, stop. Otherwise, compute d_k by (2.11).
- Step 2: Generate the step-size α_k by (2.14).
- Step 3: Set $x_{k+1} = x_k + \alpha_k d_k$.
- Step 4: Set $k := k + 1$, go to step 1.

Remark 2.1

- (i) The direction search d_k generated by Algorithm 2.1 satisfies

$$g_k^T d_k \leq -u \|g(x_k)\|^2, \forall k \geq 0, \quad (2.15)$$

where $u \in (0, 1)$. Its proof is similar to Theorem 2.1 in [28], here we omit its proof. This implies that d_k is a descent direction of f at x_k , if $f(x) = \frac{1}{2}\|F(x)\|^2$. Furthermore, using the Cauchy-Schwartz inequality, it yields that

$$\|d_k\| \geq u\|g_k\|. \quad (2.16)$$

(ii) Suppose that for the index k_0 , the line search (2.14) is not satisfied for $\forall i \geq 0$, i.e.,

$$f(x_{k_0} + \rho^i d_{k_0}) \geq f(x_{k_0}) - \sigma \|\rho^i d_{k_0}\|^2 + \eta_{k_0},$$

which can be written as

$$\frac{f(x_{k_0} + \rho^i d_{k_0}) - f(x_{k_0})}{\rho^i} \geq -\sigma \rho^i \|d_{k_0}\| + \frac{\eta_{k_0}}{\rho^i}.$$

Let $i \rightarrow +\infty$, from $\rho \in (0, 1)$ and $\eta_k > 0$ we have

$$g_{k_0}^T d_{k_0} > 0,$$

which contradicts with (2.15). Thus, the line search (2.14) is defined well.

3. Global Convergence

In this section, let us define the following level set

$$\Omega = \{x \mid f(x) \leq f(x_0) + \eta\}. \quad (3.1)$$

To prove the convergence of Algorithm 2.1, we need the following assumptions, which have been used in [18] and [20].

Assumption H

(i) The level set Ω is bounded.

(ii) F is continuous differentiable on an open and convex set V containing the level set Ω , and its Jacobian matrix is symmetric and bounded on V , i.e., there exists a positive constant M such that

$$\|J(x)\| \leq M, \forall x \in V. \quad (3.2)$$

(iii) $J(x)$ is uniformly non-singular on V , i.e., there exists a positive constant m such that

$$m\|p\| \leq \|J(x)p\|, \forall x \in V, p \in R^n. \quad (3.3)$$

It follows from Assumption (H) that

$$m\|p\| \leq \|J(x)p\| \leq M\|p\|, \forall x \in V, p \in R^n, \quad (3.4)$$

$$m\|x - y\| \leq \|F(x) - F(y)\| \leq M\|x - y\|, \forall x, y \in V, \quad (3.5)$$

$$m\|F_k\| \leq \|g_k\| = \left\| \frac{\int_0^1 J(x_k + t\alpha_k F_k) \alpha_k F_k dt}{\alpha_k} \right\| \leq M\|F_k\|, \forall x_k \in V. \quad (3.6)$$

Lemma 3.1. *Let Assumption H holds. Then we have*

$$\lim_{k \rightarrow +\infty} \alpha_k \|d_k\| = 0. \quad (3.7)$$

Proof. It follows from (2.14) that

$$\sigma \|\alpha_k d_k\|^2 \leq f(x_k) - f(x_k + \alpha d_k) + \eta_k.$$

Plugging both sides of above inequality yields

$$\sigma \sum_{k=0}^{+\infty} \|\alpha_k d_k\|^2 \leq \sum_{k=0}^{+\infty} (f(x_k) - f(x_k + \alpha d_k)) + \sum_{k=0}^{+\infty} \eta_k < f(x_0) + \eta,$$

which shows that

$$\sigma \sum_{k=0}^{+\infty} \|\alpha_k d_k\|^2 < +\infty.$$

This implies the result (3.7) holds.

Lemma 3.2. *Let Assumption H holds. Then we have*

$$\|y_k\| \leq M^2 \|s_k\|, k \geq 0, \quad (3.8)$$

where $s_k = x_{k+1} - x_k$.

Proof. From (2.9), Assumption H(ii) and (3.4), we have

$$\begin{aligned} \|y_k\| &= \|F(x_k + \gamma_k) - F(x_k)\| = \left\| \int_0^1 J(x_k + t\gamma_k) \gamma_k dt \right\| \\ &\leq M \|\gamma_k\| = M \|F(x_{k+1}) - F(x_k)\| \\ &= M \left\| \int_0^1 J(x_k + ts_k) s_k dt \right\| \leq M^2 \|s_k\|. \end{aligned}$$

Lemma 3.3. *Let Assumption H holds. If there exists a positive constant r such that*

$$\|F_k\| \geq r, \forall k \geq 0, \quad (3.9)$$

then we have

$$\hat{m} \leq \|d_k\| \leq \hat{M}, \forall k \geq 0, \quad (3.10)$$

where $\hat{m} = um$, and \hat{M} is a positive constant.

Proof. From (2.16), (3.6) and (3.9), we have

$$\|d_k\| \geq u \|g_k\| \geq um \|F_k\| \geq umr.$$

Thus the left inequality of (3.10) holds.

By (2.8) and (2.10), it is easy to obtain that

$$d_{k-1}^T y_{k-1} = d_{k-1}^T \nabla F(x_{k-1} + \vartheta \gamma_{k-1}) \gamma_{k-1} = d_{k-1}^T \nabla F(x_{k-1} + \vartheta \gamma_{k-1}) \cdot \nabla F(x_{k-1} + \theta s_{k-1}) s_{k-1},$$

where $\theta, \vartheta \in (0, 1)$, and $s_{k-1} = \alpha_{k-1}d_{k-1}$. Then we have

$$|d_{k-1}^T y_{k-1}| \geq \alpha_{k-1} \left(\frac{2}{\frac{1}{|d_{k-1}^T \nabla F(x_{k-1} + \vartheta \gamma_{k-1})|} + \frac{1}{|\nabla F(x_{k-1} + \theta s_{k-1})d_{k-1}|}} \right)^2.$$

It follows from (3.3) that

$$|d_{k-1}^T y_{k-1}| \geq \alpha_{k-1} m^2 \|d_{k-1}\|^2.$$

This inequality together with β_k in (2.12) yield

$$\begin{aligned} |\beta_k| &\leq \frac{\|g_k\| \cdot \|y_{k-1}\|}{|d_{k-1}^T y_{k-1}|} + \frac{\|y_{k-1}\|^2 \cdot \|g_k\| \cdot \|d_{k-1}\|}{|d_{k-1}^T y_{k-1}|^2} \\ &\leq \frac{\|g_k\| \cdot M^2 \|s_{k-1}\|}{\alpha_{k-1} m^2 \|d_{k-1}\|^2} + \frac{M^4 \|s_{k-1}\|^2 \cdot \|g_k\| \cdot \|d_{k-1}\|}{\alpha_{k-1}^2 m^4 \|d_{k-1}\|^4} \\ &= \frac{M^2 \|g_k\|}{m^2 \|d_{k-1}\|} + \frac{M^4 \cdot \|g_k\|}{m^4 \|d_{k-1}\|}, \end{aligned}$$

where the first inequality follows from the Cauchy-Schwartz inequality, the second inequality follows from (3.8). By using the same technique, it is not difficult to obtain that

$$|\theta_k| = \frac{\|g_k\| \cdot \|d_{k-1}\|}{|d_{k-1}^T y_{k-1}|} \leq \frac{\|g_k\| \cdot \|d_{k-1}\|}{\alpha_{k-1} m^2 \|d_{k-1}\|^2} = \frac{\|g_k\|}{\alpha_{k-1} m^2 \|d_{k-1}\|}.$$

From Assumption H(i-ii), the sequence $\{F_k\}$ is bounded. Therefore, it follows from (3.6) that the sequence $\{g_k\}$ is also bounded, i.e., there exists a positive constant \hat{r} such that

$$\|g_k\| \leq \hat{r}. \quad (3.11)$$

Then from (2.11), (3.8) and (3.11), it is easy to obtain that

$$\|d_k\| \leq \|g_k\| + |\beta_k| \cdot \|d_{k-1}\| + |\theta_k| \cdot \|y_{k-1}\| \leq \left(1 + \frac{M^2}{m^2} + \frac{M^4}{m^4} + \frac{M^2}{m^2} \right) \hat{r} \triangleq \hat{M},$$

where $\hat{M} = \left(1 + \frac{2M^2}{m^2} + \frac{M^4}{m^4} \right) \hat{r}$. The proof is completed.

Lemma 3.4. *Let Assumption H holds. Then we have*

$$\alpha_k \geq \min \left\{ 1, \frac{2\rho(u\|g_k\|^2 - \delta_k \|F_k\| \cdot \|d_k\|)}{(M^2 + 2\sigma)\|d_k\|^2} \right\} \quad (3.12)$$

where

$$\delta_k = \int_0^1 \|J(x_k + t\rho^{-1}\alpha_k d_k) - J(x_k + t\alpha_k F_k)\| dt. \quad (3.13)$$

Proof. If $\alpha_k = 1$, then (3.12) holds by the line search (2.14). If $\alpha_k \neq 1$, then $\alpha'_k = \rho^{-1}\alpha_k$ does not satisfies the line search (2.14), i.e.,

$$f(x_k + \alpha'_k d_k) - f(x_k) > -\sigma \|\alpha'_k d_k\|^2 + \eta_k > -\sigma \|\alpha'_k d_k\|^2. \quad (3.14)$$

On the other hand, from (3.5) we have

$$\begin{aligned} f(x_k + \alpha'_k d_k) - f(x_k) &= \frac{1}{2} \|F(x_k + \alpha'_k d_k)\|^2 - \frac{1}{2} \|F(x_k)\|^2 \\ &= F(x_k)^T (F(x_k + \alpha'_k d_k) - F(x_k)) + \frac{1}{2} \|F(x_k + \alpha'_k d_k) - F(x_k)\|^2 \\ &\leq F(x_k)^T \int_0^1 J(x_k + t\alpha'_k d_k) \alpha'_k d_k dt + \frac{1}{2} M^2 \|\alpha'_k d_k\|^2, \end{aligned}$$

it together with (3.14) can obtain that

$$\begin{aligned} \alpha'_k &\geq \frac{-2F(x_k)^T \int_0^1 J(x_k + t\alpha'_k d_k) d_k dt}{(M^2 + 2\sigma)\|d_k\|^2} \\ &= \frac{-2F(x_k)^T \int_0^1 J(x_k + t\alpha'_k d_k) d_k dt + 2g_k^T d_k - 2g_k^T d_k}{(M^2 + 2\sigma)\|d_k\|^2} \\ &\geq \frac{2u\|g_k\|^2 - 2F_k^T \int_0^1 J(x_k + t\alpha'_k d_k) d_k dt + 2g_k^T d_k}{(M^2 + 2\sigma)\|d_k\|^2} \\ &= \frac{2u\|g_k\|^2 - 2F_k^T \int_0^1 J(x_k + t\alpha'_k d_k) d_k dt + 2(\int_0^1 J(x_k + t\alpha_k F_k) F_k dt)^T d_k}{(M^2 + 2\sigma)\|d_k\|^2} \\ &= \frac{2u\|g_k\|^2 - 2F_k^T d_k \int_0^1 (J(x_k + t\alpha'_k d_k) - J(x_k + t\alpha_k F_k)) dt}{(M^2 + 2\sigma)\|d_k\|^2} \\ &\geq \frac{2u\|g_k\|^2 - 2\delta_k \|F_k\| \cdot \|d_k\|}{(M^2 + 2\sigma)\|d_k\|^2}, \end{aligned}$$

where the second inequality follows from (2.15), the second equality follows from (2.7) and the differentiability of F , and the third inequality follows from the Cauchy-Schwartz inequality, and δ_k is given in (3.13). By $\alpha'_k = \rho^{-1} \alpha_k$, the desired result is obtained.

Theorem 3.1. *Let Assumption H holds and the sequence $\{x_k\}$ be generated by Algorithm 2.1. Then we have*

$$\lim_{k \rightarrow +\infty} \|F_k\| = 0. \quad (3.15)$$

Proof. Assume that (3.15) is not true, then (3.9) holds. From (3.10) and (3.7), it is not difficult to obtain that

$$\lim_{k \rightarrow +\infty} \alpha_k = 0, \quad (3.16)$$

it together with (3.13) gets that

$$\lim_{k \rightarrow +\infty} \delta_k = 0. \quad (3.17)$$

From (3.6), (3.9) and (3.10), we have

$$\alpha_k \geq \min \left\{ 1, \frac{2\rho(um^2 r^2 - \delta_k \|F_k\| \cdot \|d_k\|)}{(M^2 + 2\sigma)\hat{M}^2} \right\},$$

it together with (3.17) obtained that

$$\lim_{k \rightarrow +\infty} \alpha_k \geq \min \left\{ 1, \frac{2\rho um^2 r^2}{(M^2 + 2\sigma)\hat{M}^2} \right\} > 0,$$

which contradicts with (3.16). The proof is completed.

4. Numerical Experiments

To give some insight into the behavior of Algorithm 2.1, we implemented it in Matlab to solve large-scale nonlinear symmetric equations, and compared its performance with the derivative-free MPRP method (DF-MPRP) in [26]. The parameters of Algorithm 2.1 are chosen as $\sigma = 10^{-4}$, $\rho = 0.4$ and $\eta_k = \frac{1}{(10^4+k)^2}$. The parameters of the DF-MPRP method come from [26]. All codes were written in Matlab 7.0 and they were run on a HP personal computer with Intel Core (TM) CPU 3.30GHZ and 8G memory. The iteration is stopped if

$$\|F(x_k)\| \leq 10^{-4}.$$

We also stop the iteration when the number of iterations exceeds 10000 without achieving convergence. In the following we list the test problems

$$F(x) = (F_1(x), F_2(x), \dots, F_n(x))^T$$

and the associated initial point x_0 .

Problem 4.1. This problem is the strictly convex function 1 [9]. $F(x)$ is the gradient of $h(x) = \sum_{i=1}^n (e^{x(i)} - x(i))$, i.e.,

$$F_i(x) = e^{x_i} - 1, i = 1, 2, \dots, n.$$

The initial point x_0 is randomly selected from $(0, 0.1)$ by the function "rand()".

Problem 4.2. This problem comes from Zhou et al. [3].

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

The initial point x_0 is randomly selected from $(0, 1)$ by the function "rand()".

Problem 4.3. This problem is defined as

$$\begin{aligned} F_1(x) &= -2x_1^2 + 3x_1 - x_2 + 1, \\ F_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - x_{i+1} + 1, i = 2, 3, \dots, n-1, \\ F_n(x) &= -2x_n + 3x_n - x_{n-1} + 1. \end{aligned}$$

The initial point x_0 is randomly selected from $(-0.1, 0)$ by the function "rand()".

Problem 4.4. This problem can be viewed as a modification of one problem in [26].

$$\begin{aligned} F_1(x) &= 2x_1 - x_2 + e^{x_1} - 2, \\ F_i(x) &= -x_{i-1} + 2x_i - x_{i+1} + e^{x_i} - 2, i = 2, 3, \dots, n-1, \\ F_n(x) &= -x_{n-1} + 2x_n + e^{x_n} - 2, \\ x_0 &= (0.7, 0.7, \dots, 0.7)^T. \end{aligned}$$

Problem 4.5. This problem comes from the reference [17].

$$\begin{aligned} F_1(x) &= 2.5x_1 + x_2 - 1, \\ F_i(x) &= x_{i-1} + 2.5x_i + x_{i+1} - 1, i = 2, 3, \dots, n-1, \\ F_n(x) &= x_{n-1} + 2.5x_n - 1, \\ x_0 &= (0.1, 0.1, \dots, 0.1)^T. \end{aligned}$$

Problem 4.6. This problem is the gradient of Extended Himmelblau function in CUTer [29].

$$\begin{aligned} F_{2i-1}(x) &= 4x_{2i-1}(x_{2i-1}^2 + x_{2i} - 11) + 2(x_{2i-1} + x_{2i}^2 - 7), \\ F_{2i}(x) &= 2(x_{2i-1}^2 + x_{2i} - 11) + 4x_{2i}(x_{2i-1} + x_{2i}^2 - 7), \\ x_0 &= (-1, -1, \dots, -1)^T. \end{aligned}$$

Problem 4.7. This problem is the gradient of ENGVAl function in CUTer [29].

$$\begin{aligned} F_1(x) &= 4x_1(x_1^2 + x_2^2) - 4, \\ F_i(x) &= 4x_i(x_{i-1}^2 + x_i^2) + 4x_i(x_i^2 + x_{i+1}^2) - 4, i = 2, 3, \dots, n-1, \\ F_n(x) &= 4x_n(x_{n-1}^2 + x_n^2), \\ x_0 &= (-0.1, -0.1, \dots, -0.1)^T. \end{aligned}$$

Table 1: The experiment results of Problem 4.1 via Algorithm 2.1 and DF-MPRP method

| Dim | Algorithm 2.1 | | | DF-MPRP | | |
|-------|---------------|------|--------------|---------|------|--------------|
| | Niter | time | norm(F) | Niter | time | Norm(F) |
| 3000 | 6 | 0.05 | 8.42755e-005 | 15 | 0.05 | 4.88934e-005 |
| 3000 | 6 | 0.02 | 8.77537e-005 | 15 | 0.07 | 4.92026e-005 |
| 3000 | 6 | 0.01 | 8.42474e-005 | 15 | 0.07 | 4.86115e-005 |
| 5000 | 7 | 0.05 | 1.22603e-006 | 15 | 0.07 | 6.31425e-005 |
| 5000 | 7 | 0.02 | 1.20878e-006 | 15 | 0.07 | 6.36967e-005 |
| 5000 | 7 | 0.02 | 1.19396e-006 | 15 | 0.07 | 6.39106e-005 |
| 10000 | 7 | 0.03 | 1.76086e-006 | 15 | 0.11 | 8.95949e-005 |
| 10000 | 7 | 0.03 | 1.73847e-006 | 15 | 0.12 | 8.91266e-005 |
| 10000 | 7 | 0.02 | 1.67444e-006 | 15 | 0.12 | 8.96641e-005 |
| 15000 | 7 | 0.05 | 2.14243e-006 | 17 | 0.17 | 2.73511e-005 |
| 15000 | 7 | 0.04 | 2.11205e-006 | 17 | 0.18 | 2.73155e-005 |
| 15000 | 7 | 0.01 | 2.12148e-006 | 17 | 0.17 | 2.72951e-005 |
| 20000 | 7 | 0.04 | 2.44954e-006 | 17 | 0.21 | 3.15998e-005 |
| 20000 | 7 | 0.06 | 2.50460e-006 | 17 | 0.23 | 3.15306e-005 |
| 20000 | 7 | 0.04 | 2.44354e-006 | 17 | 0.22 | 3.17312e-005 |
| 25000 | 7 | 0.05 | 2.71887e-006 | 17 | 0.27 | 3.52324e-005 |
| 25000 | 7 | 0.06 | 2.70065e-006 | 17 | 0.24 | 3.51942e-005 |
| 25000 | 7 | 0.05 | 2.76221e-006 | 17 | 0.23 | 3.54252e-005 |
| 30000 | 7 | 0.05 | 2.93391e-006 | 17 | 0.31 | 3.88033e-005 |
| 30000 | 7 | 0.06 | 2.97723e-006 | 17 | 0.28 | 3.85890e-005 |
| 30000 | 7 | 0.05 | 3.04345e-006 | 17 | 0.32 | 3.86010e-005 |

We test Algorithm 2.1 and DF-MPRP method on the above seven problems with different sizes. The numerical results are listed in Tables 1-4, where "Dim" denotes the size of the problem, i.e., the size of the variable x , "Niter" denotes the total number of iterations, "time" stands for the CPU time in seconds, and "Norm(F)" denotes the final norm of F when the algorithm terminates. Tables 1-4 shows that the proposed algorithm can solve the given problems successfully. Through an intuitive comparison in the CPU time and the number of iterations, it is clear to find that the proposed algorithm performs better than the competitor for most cases.

Table 2: The experiment results of Problem 4.2 via Algorithm 2.1 and DF-MPRP method

| Dim | Algorithm 2.1 | | | DF-MPRP | | |
|-------|---------------|------|--------------|---------|------|--------------|
| | Niter | time | norm(F) | Niter | time | Norm(F) |
| 3000 | 9 | 0.03 | 2.27820e-006 | 19 | 0.08 | 7.30060e-005 |
| 3000 | 8 | 0.03 | 5.49510e-005 | 19 | 0.08 | 7.24213e-005 |
| 3000 | 8 | 0.03 | 3.89106e-005 | 19 | 0.08 | 7.21331e-005 |
| 5000 | 8 | 0.04 | 3.76338e-005 | 19 | 0.10 | 2.67296e-005 |
| 5000 | 8 | 0.04 | 9.98721e-005 | 19 | 0.16 | 2.68681e-005 |
| 5000 | 8 | 0.03 | 6.46677e-005 | 19 | 0.10 | 2.66384e-005 |
| 10000 | 8 | 0.05 | 4.47492e-005 | 19 | 0.16 | 3.76539e-005 |
| 10000 | 9 | 0.09 | 3.85702e-006 | 21 | 0.17 | 3.28637e-005 |
| 10000 | 9 | 0.06 | 2.58068e-006 | 19 | 0.14 | 3.77101e-005 |
| 15000 | 9 | 0.07 | 4.16468e-006 | 21 | 0.22 | 4.05386e-005 |
| 15000 | 8 | 0.07 | 2.78003e-005 | 19 | 0.21 | 4.62622e-005 |
| 15000 | 8 | 0.09 | 8.60771e-005 | 19 | 0.22 | 4.61363e-005 |
| 20000 | 9 | 0.08 | 2.62703e-006 | 21 | 0.28 | 4.68991e-005 |
| 20000 | 9 | 0.11 | 2.81289e-006 | 21 | 0.29 | 4.66947e-005 |
| 20000 | 9 | 0.08 | 3.80369e-006 | 21 | 0.29 | 4.71957e-005 |
| 25000 | 9 | 0.12 | 5.45376e-006 | 19 | 0.30 | 5.99489e-005 |
| 25000 | 9 | 0.09 | 6.47549e-006 | 21 | 0.36 | 5.25745e-005 |
| 25000 | 9 | 0.09 | 5.99922e-006 | 21 | 0.34 | 5.22860e-005 |
| 30000 | 9 | 0.12 | 6.91320e-006 | 21 | 0.43 | 5.70499e-005 |
| 30000 | 8 | 0.11 | 8.78074e-005 | 19 | 0.37 | 6.52845e-005 |
| 30000 | 9 | 0.13 | 7.53149e-006 | 21 | 0.41 | 5.78082e-005 |

Table 3: The experiment results of Problem 4.3 via Algorithm 2.1 and DF-MPRP method

| Dim | Algorithm 2.1 | | | DF-MPRP | | |
|-------|---------------|------|--------------|---------|------|--------------|
| | Niter | time | norm(F) | Niter | time | Norm(F) |
| 3000 | 43 | 0.08 | 8.53567e-005 | 53 | 0.15 | 8.37550e-005 |
| 3000 | 43 | 0.07 | 8.45330e-005 | 53 | 0.17 | 8.34647e-005 |
| 3000 | 44 | 0.08 | 4.81906e-005 | 53 | 0.16 | 8.35553e-005 |
| 5000 | 42 | 0.10 | 7.48374e-005 | 55 | 0.27 | 7.61069e-005 |
| 5000 | 56 | 0.11 | 4.32998e-005 | 55 | 0.26 | 7.84752e-005 |
| 5000 | 52 | 0.10 | 9.68929e-005 | 55 | 0.25 | 7.63853e-005 |
| 10000 | 49 | 0.14 | 2.02359e-005 | 57 | 0.45 | 8.14832e-005 |
| 10000 | 37 | 0.14 | 8.78056e-005 | 57 | 0.46 | 8.03619e-005 |
| 10000 | 51 | 0.18 | 4.72819e-005 | 57 | 0.47 | 8.07561e-005 |
| 15000 | 56 | 0.24 | 3.76223e-005 | 57 | 0.67 | 9.72658e-005 |
| 15000 | 65 | 0.28 | 4.11716e-005 | 57 | 0.65 | 9.97803e-005 |
| 15000 | 49 | 0.19 | 5.64680e-005 | 57 | 0.68 | 9.89181e-005 |
| 20000 | 58 | 0.32 | 7.62819e-005 | 59 | 0.91 | 8.73651e-005 |
| 20000 | 49 | 0.30 | 4.10693e-005 | 59 | 0.90 | 8.85322e-005 |
| 20000 | 47 | 0.27 | 4.99308e-005 | 59 | 0.91 | 8.69460e-005 |
| 25000 | 48 | 0.28 | 9.09809e-005 | 59 | 1.14 | 9.80684e-005 |
| 25000 | 45 | 0.31 | 8.94214e-005 | 59 | 1.10 | 9.80414e-005 |
| 25000 | 55 | 0.35 | 8.99718e-005 | 59 | 1.15 | 9.77665e-005 |
| 30000 | 48 | 0.42 | 9.84083e-005 | 61 | 1.39 | 8.58837e-005 |
| 30000 | 48 | 0.34 | 6.51353e-005 | 61 | 1.42 | 8.43327e-005 |
| 30000 | 51 | 0.44 | 9.73236e-005 | 61 | 1.39 | 8.55057e-005 |

Table 4: The experiment results via Algorithm 2.1 and DF-MPRP method

| | Algorithm 2.1 | | | | DF-MPRP | | |
|-------------|---------------|-------|------|--------------|---------|-------|--------------|
| | Dim | Niter | time | norm(F) | Niter | time | Norm(F) |
| Problem 4.4 | 3000 | 16 | 0.04 | 8.12332e-005 | 67 | 0.25 | 9.11789e-005 |
| | 5000 | 21 | 0.06 | 9.76878e-005 | 69 | 0.42 | 8.93827e-005 |
| | 10000 | 22 | 0.10 | 9.60944e-005 | 73 | 0.87 | 6.99000e-005 |
| | 15000 | 17 | 0.12 | 9.64037e-005 | 73 | 1.27 | 8.29428e-005 |
| | 20000 | 107 | 0.72 | 7.05200e-005 | 71 | 1.64 | 8.21958e-005 |
| | 25000 | 50 | 0.46 | 6.23862e-005 | 71 | 2.06 | 8.82361e-005 |
| | 30000 | 22 | 0.23 | 8.83250e-005 | 71 | 2.49 | 9.49405e-005 |
| Problem 4.5 | 3000 | 52 | 0.05 | 8.71393e-005 | 353 | 0.77 | 9.82623e-005 |
| | 5000 | 59 | 0.07 | 9.45272e-005 | 353 | 1.23 | 9.93081e-005 |
| | 10000 | 58 | 0.11 | 5.69105e-005 | 353 | 2.37 | 9.94476e-005 |
| | 15000 | 53 | 0.14 | 7.22079e-005 | 353 | 3.48 | 9.95185e-005 |
| | 20000 | 50 | 0.17 | 9.80776e-005 | 353 | 4.65 | 9.96506e-005 |
| | 25000 | 58 | 0.25 | 7.53797e-005 | 355 | 5.90 | 9.72970e-005 |
| | 30000 | 52 | 0.27 | 9.01675e-005 | 353 | 6.94 | 9.99679e-005 |
| Problem 4.6 | 3000 | 19 | 0.06 | 1.24405e-005 | 75 | 0.24 | 8.33826e-005 |
| | 5000 | 17 | 0.06 | 5.66862e-005 | 73 | 0.36 | 9.93397e-005 |
| | 10000 | 17 | 0.10 | 8.01123e-005 | 79 | 0.76 | 8.49809e-005 |
| | 15000 | 17 | 0.12 | 9.81849e-005 | 81 | 1.15 | 8.01166e-005 |
| | 20000 | 26 | 0.24 | 1.38078e-005 | 79 | 1.48 | 7.87608e-005 |
| | 25000 | 18 | 0.23 | 2.67189e-005 | 83 | 1.94 | 7.56907e-005 |
| | 30000 | 18 | 0.24 | 2.92661e-005 | 79 | 2.23 | 9.63206e-005 |
| Problem 4.7 | 3000 | 53 | 0.07 | 8.48670e-005 | 657 | 2.02 | 9.93122e-005 |
| | 5000 | 74 | 0.14 | 9.27647e-005 | 687 | 3.41 | 9.85825e-005 |
| | 10000 | 51 | 0.18 | 6.69782e-005 | 669 | 6.43 | 9.83548e-005 |
| | 15000 | 87 | 0.39 | 8.60435e-005 | 629 | 9.04 | 9.96839e-005 |
| | 20000 | 58 | 0.37 | 7.67571e-005 | 657 | 12.44 | 9.81281e-005 |
| | 25000 | 82 | 0.62 | 9.89245e-005 | 675 | 16.00 | 9.98608e-005 |
| | 30000 | 85 | 0.77 | 9.50747e-005 | 617 | 17.70 | 9.88761e-005 |

5. Conclusions

In this paper, a new norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations were proposed and analyzed, in which the search direction is symmetric. Under some suitable assumptions, it was established that the iterations converge to a solution of the problem via some properties of Jacobian matrix of F . There is not any information about Jacobian matrix of F to be used in the whole implementation process of the proposed algorithm. So the proposed algorithm is suitable to solve large-scale problems. Preliminary computational results also show that the proposed algorithm worked well and performs better than the DF-PRP method for the given symmetric equations. However, in this paper we only considered solving nonlinear symmetric equations without any constrained conditions. How to solve nonlinear symmetric equations with convex constraints will be further studied.

References

- [1] S. Bellavia, B. Morini, "A globally convergent Newton-GMRES subspace method for systems of nonlinear equations", *SIAM J.Sci.Comput.*, Vol.23, pp. 940-960, 2001.
- [2] G. Zhou, K.C. Toh, "Superline convergence of a Newton-type algorithm for monotone equations", *J.Optim.Theory Appl.*, Vol.125, pp.205-221, 2005.
- [3] W.J. Zhou, D.H. Li, "Limited memory BFGS method for nonlinear monotone equations", *J.Comput.Math.*, Vol.25, pp.89-96, 2007.
- [4] E.G. Birgin, N.K. Krejic, J.M. Martínez, "Globally convergent inexact quasi-Newton methods for solving nonlinear systems", *Numer. Algorithms*, Vol.32, pp.249-260, 2003.
- [5] P.N. Brown, Y. Saad, "Convergence theory of nonlinear Newton-Krylov algorithms", *SIAM J.Optim.*, Vol.4, pp.297-330, 1994.
- [6] M.V. Solodov, B.F. Svaiter, "A globally convergent inexact Newton method for systems of monotone equations", In: Fukushima, M., Qi,L.(eds.)*R eformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*,pp.355-369. Kluwer Academic, 1998.
- [7] C.G. Broyden, "A class of methods for solving nonlinear simultaneous equations", *Math.Comput.*, Vol.19, pp.577-593, 1965.
- [8] W. La Cruz, M. Raydan, "Nonmonotone spectral methods for large-scale nonlinear systems", *Optim.Mehtods Softw.*, Vol.18, pp. 583-599, 2003.
- [9] W. La Cruz, J.M. Martínez, M. Raydan, "Spectral residual method without gradient minformation for solving large-scale nonlinear systems of equations", *Math.Comput.*, Vol.75, pp. 1429-1448, 2006.
- [10] W.Y. Cheng, "A PRP type method for systems of monotone equations", *Math.Comput.Model.*, Vol.50, pp. 15-20, 2009.
- [11] Q.N. Li, D.H. Li. "A class of derivative-free methods for large-scale nonlinear monotone equations", *IMA J.Numer.Anal.*, Vol.31, pp.1625-1635, 2011.
- [12] M. Ahookhosh, K. Amini, S. Bahrami, "Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations", *Numer.Algorithms*, Vol.64, pp.21-42, 2013.
- [13] G.L. Yuan, M.J Zhang. "A three-terms Polak-Ribire-Polyak conjugate gradient algorithm for large-scale nonlinear equations", *J. Comput. Appl. Math.*, Vol.286, pp.186-195, 2015.
- [14] J.K. Liu, S.J. Li. "Spectral DY-type projection method for nonlinear monotone system of equations", *J.Comput.Math.*, Vol.33, pp.341-355, 2015.
- [15] G.L. Yuan, Z.H. Meng, Y. Li. " A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations", *J.Optim.Theory Appl.*, Vol.168, pp.129-152, 2016.
- [16] Z.F. Dai, X.H. Chen, F.H. Wen. "A modified Perry's conjugate gradient method-based derivative-free method for solving large-scale nonlinear monotone equations", *Appl.Math.Comput.*, Vol.270, pp.378-386, 2015.
- [17] J.K. Liu, S.J. Li, "Multivariate spectral DY-type projection method for convex constrained nonlinear monotone equations", *J. Ind. Manag. Optim.*, Vol.13, pp.283-295, 2017.
- [18] D. Li, M. Fukushima, "A global and superlinear convergencet Gauss-Newton-based BFGS method for symmetric nonlinear equations", *SIAM J.Numer.Anal.*, Vol.37, pp.152-172, 1999.
- [19] G.Z. Gu, D.H. Li, L.Q. Qi, S.Z. Zhou, "Descent direction of quasi-Newton methods for symmetic nonlinear equations", *SIAM J.Numer.Anal.*, Vol. 40, pp.1763-1774, 2002
- [20] D.H. Li, X.L. Wang, "A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations", *Numer. Alge. Ctrl.and Optim.*, Vol.1, pp.71-82,2011.
- [21] L. Zhang, W.J. Zhou, D.H. Li, "Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search", *Numer.Math.*, Vol.104, pp.561-572, 2006.
- [22] Y.H. Xiao, C.J. Wu, S.Y. Wu, "Norm descent conjugate gradient methods for solving symmetric nonlinear equations", *J.Glob.Optim.*, Vol.62, pp.751-762, 2015.
- [23] Y.H. Dai, C.X. Kou, "A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search", *SIAM J.Optim.*, Vol.23, pp.296-320, 2013
- [24] N. Andrei, "A simple three-term conjugate gradient algorithm for unconstrained optimization", *J.Comput.Appl.Math.*, Vol.241, pp.19-29, 2013.
- [25] N. Andrei, "On three-term conjugate gradient algorithms for unconstrained optimization", *Appl.Math.Comput.*, Vol.219, pp.6316-6327, 2013.
- [26] W.J. Zhou, D.M. Shen, "Convergence properties of an iterative method for solving symmetric non-linear equations", *J.Optim.Theory Appl.*, Vol.164, pp.277-289, 2015.
- [27] L. Zhang, W.Zhou, D. Li, "A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence", *IMA J.Numer.Anal.*, Vol.26, pp.629-640, 2006.
- [28] J.K. Liu, S.J. Li, "New three-term conjugate gradient method with guaranteed global convergence", *Int.J.Comput.Math.*, Vol.91, pp.1744-1754, 2014.

- [29] A.R. Conn, N.I.M. Gould, Ph.L. Toint, "CUTEr: constrained and unconstrained testing environment", ACM Trans. Math. Softw., Vol.21, pp.123-160, 1995.