A Broyden Trust Region Quasi-Newton Method for Nonlinear Equations

Meilan Zeng, Hongwei Fu*

Abstract—**Systems of nonlinear equations have been widely applied in many aspects, such as computational science and engineering, etc. In this paper, we propose a Broyden trust region quasi-Newton method for solving general nonlinear equations. The method is based on a new trust region radius, and possesses the global and superlinear convergence under appropriate conditions. Numerical experiments show that the proposed method is more competitive than the Broyden linear search and BFGS trust region quasi-Newton methods.**

Index Terms—**Broyden quasi-Newton method, trust region, nonlinear equations.**

I. Introduction

Consider the numerical solution of nonlinear equations

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

where $F:R^n \to R^n$ is continuously differentiable. Let $J(x)$ denote the Jacobian matrix of F at x point. Throughout the paper, we assume that the solution set of (1.1) is nonempty. In all cases, $\|\cdot\|$ denotes the Euclidian norm of vectors or its induced matrix norm. Let $f(x)=\frac{1}{2}\|F(x)\|^2$, the nonlinear equation problem (1.1) is equivalent to the global optimization problem [12]

$$f(x)=0. \tag{1.2}$$

Conventional quasi-Newton methods [1, 2] for solving (1.1) generate a sequence of iterates $\{x_k\}$ by $x_{k+1}=x_k+d_k$, where d_k is a solution of the following system of linear equations

$$F_k+B_k d=0, \tag{1.3}$$

where $F_k=F(x_k), B_k$ is an approximation of $J_k=J(x_k)$. An attractive feature of quasi-Newton methods is its local superlinear convergence without computing the Jacobian matrix. Since

$$\nabla f(x_k)^T d_k=-F_k^T J_k B_k^{-1} F_k, \tag{1.4}$$

then d_k is not necessarily a descent direction of f at x_k. One way to globalize such quasi-Newton methods is to employ the line search rule given by Griewank [5] and Li et al. [7]. In [7], λ_k satisfies the line search condition

$$\|F(x_k+\lambda_k d_k)\| \le (1+\eta_k)\|F(x_k)\|-\sigma_1\|\lambda_k d_k\|^2, \tag{1.5}$$

where $\sigma_1>0, \eta>0, \sum_k \eta_k \le \eta<\infty$. Another way is to exploit the trust region strategy. In this paper, we use the latter technique.

Yuan et al. [15] propose a trust region BFGS quasi-method for solving nonlinear equations

$$\min q_k(d) \text{ such that } \|d\| \le \Delta_k, \tag{1.6}$$

where $q_k(d)=\frac{1}{2}\|F_k+B_k d\|^2$, $\Delta_k=c^p\|F_k\|, 0<c<1$, and p is a nonnegative integer, the trust region radius Δ_k has an important relation with $\|F_k\|$. B_k is an approximation of J_k, B_k is generated by the BFGS formula

$$B_{k+1}=B_k-\frac{B_k d_k d_k^T B_k}{d_k^T B_k d_k}+\frac{y_k y_k^T}{y_k^T d_k}, \tag{1.7}$$

where $d_k=x_{k+1}-x_k$, $y_k=F_{k+1}-F_k$. Under appropriate conditions, the global and superlinear convergence is obtained. The BFGS quasi-Newton method is not necessarily suitable for nonsymmetric nonlinear equations. Some derivative-free methods (see [6, 8, 9, 16]) are proposed for symmetric nonlinear equations, whereas derivative-free methods are few for general nonlinear equations.

The purpose of this paper is to propose a Broyden quasi-Newton method for solving general nonlinear equations with a new trust region radius. At each iterative point x_k, the trial step d_k is obtained by solving the following subproblem

$$q_k(d)=\frac{1}{2}\|F_k+B_k d\|^2 \text{ such that } \|d\| \le \Delta_k, \tag{1.8}$$

$\Delta_k=c^p$ is a new trust region radius, $0<c<1$, and p is a nonnegative integer. B_k is generated by the Broyden updating formula

$$B_{k+1}=B_k+\frac{(y_k-B_k d_k)d_k^T}{d_k^T d_k}, \tag{1.9}$$

B_k is nonsingular. B_{k+1} satisfies the quasi-Newton equation $B_{k+1}d_k=y_k$. The global and superlinear convergence is established under suitable conditions. Numerical results show

Manuscript received February 28, 2019. This work was supported by the National Natural Science Foundation of China (11601138), the Teaching Research Project of Hubei Engineering University (2018C18) and the University Development Reform and Research Project of Hubei Engineering University (HGFGZX201803).

M.L. Zeng is with the *School of Mathematics and Statistics, Hubei Engineering University, Xiaogan, Hubei, 432000, China* (e-mail: zml571@126.com).

H.W. Fu is the corresponding author with the *School of Mathematics and Statistics, Hubei Engineering University, Xiaogan, Hubei, 432000, China* (e-mail: gcxyfhw@163.com).

that the proposed algorithm performs better than the two quasi-Newton algorithms determined by the Broyden line search quasi-Newton and BFGS trust region quasi-Newton methods.

II. New quasi-newton method

In this section, we give the Broyden trust region quasi-Newton method for solving nonlinear equations.

Let d_k^p be the solution of the trust region subproblem (1.8) corresponding to p. We define the actual reduction as

$$Ared_k(d_k^p) = f(x_k) - f(x_k + d_k^p), \tag{2.1}$$

the predict reduction as

$$Pred_k(d_k^p) = q_k(0) - q_k(d_k^p), \tag{2.2}$$

and the ratio of actual reduction over predict reduction as

$$r_k^p = \frac{Ared_k(d_k^p)}{Pred_k(d_k^p)}.$$

Algorithm 1

Step 0 Choose $\rho, c \in (0,1),\ p = 0,\ \varepsilon > 0$. Initialize x_0, B_0. Set $k := 0$.

Step 1 Evaluate F_k, if $\|F_k\| \le \varepsilon$, terminate.

Step 2 Solve the subproblem (1.8) to obtain d_k^p.

Step 3 Compute

$$r_k^p = \frac{Ared_k(d_k^p)}{Pred_k(d_k^p)}.$$

If $r_k^p \ge \rho$, then $x_{k+1} = x_k + d_k^p$, go to step 4. Otherwise, set $p := p+1$ go to step 2.

Step 4 Update B_k by (1.9). Set $k := k+1$ and $p = 0$. Go to Step 1.

III. Convergence analysis

In this section, we prove the global and superlinear convergence of Algorithm 1.

A. Global convergence

In order to prove the global convergence of Algorithm 1, we make the following assumptions.

Assumption 3.1 (1) The level set $\Omega = \{x \in R^n \mid f(x) \le f(x_0)\}$ is bounded.

(2) $F(x)$ is twice differentiable in an open convex set Ω_1 containing Ω.

(3) The following relation

$$\|[J_k - B_k]^T F_k\| = O(\|d_k^p\|)$$

holds.

(4) The matrices $\{B_k\}$ are uniformly bounded in Ω_1, which means that there exist positive constants $0 < M_0 \le M$ such that

$$M_0 \le \|B_k\| \le M, \forall k. \tag{3.1}$$

Similar to Yuan [14], we get the following lemmas from Algorithm 1 and Assumption 3.1.

Lemma 3.1

$$|Ared_k(d_k^p) - Pred_k(d_k^p)| = O(\|d_k^p\|^2) = O(\Delta_k^2).$$

Proof By (2.1) and (2.2), we have

$$\begin{aligned} &|Ared_k(d_k^p) - Pred_k(d_k^p)| = |q_k(d_k^p) - f(x_k + d_k^p)| \\ &= \frac{1}{2}\left|\|F_k + B_k d_k^p\|^2 - \|F_k + J_k d_k^p + O(\|d_k^p\|^2)\|^2\right| \\ &= F_k^T(B_k - J_k)d_k^p + O(\|d_k^p\|^2) \\ &\le \|[B_k - J_k]^T F_k\|\|d_k^p\| + O(\|d_k^p\|^2) \\ &= O(\|d_k^p\|^2) = O(\Delta_k^2). \end{aligned}$$

Lemma 3.2 If d_k^p is a solution of (1.8), then

$$Pred_k(d_k^p) \ge \frac{1}{2}\|B_k^T F_k\| \min\left\{\Delta_k, \frac{\|B_k^T F_k\|}{\|B_k\|^2}\right\}. \tag{3.2}$$

Proof Since d_k^p is a solution of (1.8), for any $\alpha \in [0,1]$, it follows

$$\begin{aligned} Pred_k(d_k^p) &= \frac{1}{2}\left(\|F_k\|^2 - \|F_k + B_k d_k^p\|^2\right) \\ &\ge \frac{1}{2}\left(\|F_k\|^2 - \left\|F_k - B_k \frac{\alpha\Delta_k}{\|B_k^T F_k\|} B_k^T F_k\right\|^2\right) \qquad (3.3) \\ &= \alpha\Delta_k \|B_k^T F_k\| - \frac{1}{2}\alpha^2\Delta_k^2\|B_k\|^2. \end{aligned}$$

Therefore,

$$Pred_k(d_k^p) \ge \max_{0\le\alpha\le1}\left[\alpha\Delta_k\|B_k^T F_k\| - \frac{1}{2}\alpha^2\Delta_k^2\|B_k\|^2\right]$$

$$\ge \frac{1}{2}\|B_k^T F_k\|\min\left\{\Delta_k, \frac{\|B_k^T F_k\|}{\|B_k\|^2}\right\}. \tag{3.4}$$

This completes the proof.

Lemma 3.3 Algorithm 1 does not circle between Steps 2-3 infinitely.

Proof If Algorithm 1 circles between Steps 2 and 3 infinitely, i.e., $p \to \infty, r_k^p < \rho$ and $c^p \to 0$. Obviously $\|B_k^T F_k\| > \varepsilon$, otherwise the algorithm stops. Thus $\|d_k^p\| \le \Delta_k = c^p \to 0$.

From Lemmas 3.1 and 3.2, we have

$$|r_k^p - 1| = \frac{|Ared_k(d_k^p) - Pred_k(d_k^p)|}{|Pred_k(d_k^p)|} \le \frac{2O(\Delta_k^2)}{\Delta_k\|B_k^T F_k\|} \to 0. \tag{3.5}$$

Hence, for k large enough,

$$r_k^p \ge \rho, \tag{3.6}$$

which contradicts the fact that $r_k^p < \rho$.

Theorem 3.1 Let Assumption 3.1 hold and $\{x_k\}$ be generated by Algorithm 1. Then either there exists some finite k_0 such that $B_{k_0}^T F_{k_0} = 0$ or

$$\liminf_{k\to\infty} \left\| B_k^T F_k \right\| = 0. \quad (3.7)$$

Proof Assume that Algorithm 1 does not stop after finitely many steps, then there exists a positive constant ε and an infinite subsequence $\{k_j\}$ such that $\left\| B_{k_j}^T F_{k_j} \right\| \ge \varepsilon.$ Let $K = \{k \mid \left\| B_k^T F_k \right\| \ge \varepsilon\}$.

Using Algorithm 1 and Lemma 3.2, we have

$$\sum_{k\in K}[f(x_k) - f(x_{k+1})] \ge \sum_{k\in K} \rho \cdot Pred_k(d_k^p)$$

$$\ge \sum_{k\in K} \rho \cdot \frac{\varepsilon}{2}\min\left\{\Delta_k, \frac{\varepsilon}{M^2}\right\}.$$

By the definition of Algorithm 1, it follows

$$r_k^p \ge \rho > 0. \quad (3.8)$$

This implies

$$0 \le f(x_{k+1}) \le f(x_k) \le \cdots \le f(x_0).$$

Therefore, $\{f(x_k)\}$ is convergent, then

$$\sum_{k\in K} \rho \cdot \frac{\varepsilon}{2}\min\left\{\Delta_k, \frac{\varepsilon}{M^2}\right\} < \infty.$$

Thus $\Delta_k \to 0, k \to +\infty, k \in K$, which implies that $p_k \to +\infty$ as $k \to +\infty$ ($k \in K$). Therefore, we can assume $p_k \ge 1$ for all $k \in K$.

From the determination of $p_k (k \in K)$ in the inner circle, the solution $\tilde{d}_k$ corresponding to the following subproblem

$$\min q_k(d) = \frac{1}{2}\left\| F_k + B_k d \right\|^2 \ s.t. \left\| d \right\| \le c^{p_k - 1} \quad (3.9)$$

is unacceptable. Let $\tilde{x}_{k+1} = x_k + \tilde{d}_k$, we have

$$\frac{f(x_k) - f(\tilde{x}_{k+1})}{Pred_k(\tilde{d}_k)} < \rho. \quad (3.10)$$

From Lemma 3.2, it follows

$$Pred_k(\tilde{d}_k) \ge \frac{\varepsilon}{2}\min\left\{c^{p_k-1}, \frac{\varepsilon}{M^2}\right\}.$$

By Lemma 3.1, we get

$$f(x_k) - f(\tilde{x}_{k+1}) - Pred_k(\tilde{d}_k) = O(\left\| \tilde{d}_k \right\|^2) = O(c^{2(p_k-1)}).$$

Therefore,

$$\left| \frac{f(x_k) - f(\tilde{x}_{k+1})}{Pred_k(\tilde{d}_k)} - 1 \right| \le \frac{O(c^{2(p_k-1)})}{\frac{\varepsilon}{2}\min\left\{c^{p_k-1}, \frac{\varepsilon}{M^2}\right\}}.$$

Since $p_k \to +\infty$ as $k \to +\infty$, we obtain

$$\frac{f(x_k) - f(\tilde{x}_{k+1})}{Pred_k(\tilde{d}_k)} \to 1, k \in K.$$

This is in contradiction with (3.10). Thus, the conclusion follows.

Remark Theorem 3.1 shows that the iterative sequence $\{x_k\}$ generated by Algorithm 1 satisfies $\left\| B_k^T F_k \right\| \to 0$. If The matrices $\{B_k^{-1}\}$ are uniformly bounded in Ω_1, then we have $\left\| F_k \right\| \to 0$.

The following lemma means that the solution of the subproblem in Step 2 is close to quasi-Newton direction when the number of iterations is sufficiently large.

Lemma 3.4 Let the conditions of Theorem 3.1 hold, d_k^p be generated by Algorithm 1. If the iterative sequence $\{x_k\}$ converges to x^*, then there exists a positive integer K such that

$$d_k^p = -B_k^{-1} F_k \quad (3.11)$$

for all $k \ge K$.

Proof By Assumption 3.1(4), we have

$$\Delta q_k = q_k(0) - q_k(d_k^p)$$

$$= -F_k^T B_k d_k^p - \frac{1}{2}(d_k^p)^T B_k^T B_k d_k^p$$

$$= O(\left\| d_k^p \right\|).$$

Because $\{x_k\}$ converges to x^* , $J(x^*)^T J(x^*)$ is positive definite. By the continuity of $J(x)^T J(x)$, $J(x)^T J(x)$ is bounded. From Assumption 3.1(3),

$$\Delta f_k - \Delta q_k = -F_k^T J_k d_k^p - \frac{1}{2}(d_k^p)^T J_k^T J_k d_k^p + o(\left\| d_k^p \right\|^2)$$

$$+ F_k^T B_k d_k^p + \frac{1}{2}(d_k^p)^T B_k^T B_k d_k^p$$

$$= F_k^T (B_k - J_k) d_k^p + \frac{1}{2}(d_k^p)^T (B_k^T B_k - J_k^T J_k) d_k^p$$

$$+ o(\left\| d_k^p \right\|^2)$$

$$\le \left\| (B_k - J_k)^T F_k \right\| \cdot \left\| d_k^p \right\| + \frac{1}{2}(d_k^p)^T (B_k^T B_k - J_k^T J_k) d_k^p$$

$$+ o(\left\| d_k^p \right\|^2) = O(\left\| d_k^p \right\|^2).$$

Therefore, we obtain

$$\left| \frac{\Delta f_k}{\Delta q_k} - 1 \right| \le \frac{O(\left\| d_k^p \right\|^2)}{O(\left\| d_k^p \right\|)},$$

which implies that $\frac{\Delta f_k}{\Delta q_k} \to 1$ as $k \to \infty$ .Then there exists a positive integer K and a lower bound of the trust region radius $\tilde{\Delta}$, such that

$$\left\| (B_k^T B_k)^{-1} B_k^T F_k \right\| \le \tilde{\Delta} \le \Delta_k$$

holds for all $k \ge K$. Because $\{B_k\}$ are nonsingular, we have

$$d_k^p = -(B_k^T B_k)^{-1} B_k^T F_k = -B_k^{-1} F_k.$$

B. Superlinear convergence

In order to establish the superlinear convergence of Algorithm 1, the following assumption is further needed.

Assumption 3.2 (1) $x_k \to x^*$, where x^* is the solution of (1.1).

(2) Let $D \subseteq R^n$ be an open convex set, $J(x) \in \mathrm{Lip}_\gamma(D),\ x \in D,$ i.e., $J(x)$ is Lipschitz continuous.

(3) $\lim\limits_{k\to\infty} \dfrac{\|(B_k - J(x^*))(x_{k+1}-x_k)\|}{\|x_{k+1}-x_k\|} = 0.$

Lemma 3.5 ([3]) Let $F: R^n \to R^n$ be continuously differentiable in the open convex set $D \subseteq R^n$, $J(x) \in \mathrm{Lip}_\gamma(D),\ x \in D$. For any $v, u \in D$, it follows

$$\|F(v)-F(u)-J(x)(v-u)\| \le \frac{\gamma}{2}\left[\|v-x\|+\|u-x\|\right]\|v-u\|. \tag{3.12}$$

Lemma 3.6 ([3]) Let F, J satisfy the conditions of Lemma 3.5 and assume that $J(x)^{-1}$ exists. Then there exists $\alpha > 0$, such that

$$\|F(v)-F(u)\| \ge \alpha\|v-u\|, \tag{3.13}$$

for all $v, u \in D$.

Theorem 3.2 Let Assumptions 3.1 and 3.2 hold, $\{x_k\}$ be generated by Algorithm 1. Then $\{x_k\}$ converges superlinearly to x^* and $F(x^*) = 0$.

Proof Define $e_k = x_k - x^*$. Now we show that the sequence $\{x_k\}$ converges superlinearly to x^*. From Lemma 3.4, there exists a positive integer K, it follows

$$0 = B_k d_k^p + F_k = (B_k - J(x^*))d_k^p + F_k + J(x^*)d_k^p$$

for any $k \ge K$. Therefore, we have

$$-F_{k+1} = (B_k - J(x^*))d_k^p + [-F_{k+1} + F_k + J(x^*)d_k^p],$$

where $d_k^p = x_{k+1} - x_k$.

From Assumption 3.2(2) and Lemma 3.5,

$$\frac{\|F_{k+1}\|}{\|d_k^p\|} \le \frac{\|(B_k - J_*)d_k^p\|}{\|d_k^p\|} + \frac{\|-F_{k+1}+F_k+J(x^*)d_k^p\|}{\|d_k^p\|} \le \frac{\|(B_k - J_*)(x_{k+1}-x_k)\|}{\|x_{k+1}-x_k\|} + \frac{\gamma}{2}(\|e_{k+1}\|+\|e_k\|).$$

Using Assumption 3.2(3), we have $\lim\limits_{k\to\infty}\dfrac{\|F_{k+1}\|}{\|d_k^p\|} = 0.$

Since $\lim\limits_{k\to\infty}\|d_k^p\| = 0$, this implies

$$F(x^*) = \lim_{k\to\infty} F(x_k) = 0. \tag{3.14}$$

From Lemma 3.6, there exist $\alpha \ge 0, k_0 \ge 0$, such that

$$\|F_{k+1}\| = \|F_{k+1} - F(x^*)\| \ge \alpha\|e_{k+1}\| \tag{3.15}$$

for all $k \ge k_0$. According to (3.14) and (3.15), we get

$$0 = \lim_{k\to\infty}\frac{\|F_{k+1}\|}{\|d_k^p\|} \ge \lim_{k\to\infty}\alpha\frac{\|e_{k+1}\|}{\|d_k^p\|} \ge \lim_{k\to\infty}\frac{\alpha\|e_{k+1}\|}{\|e_{k+1}\|+\|e_k\|} = \lim_{k\to\infty}\frac{\alpha r_k}{r_k+1},$$

where $r_k = \dfrac{\|e_{k+1}\|}{\|e_k\|}$. This implies

$$\lim_{k\to\infty} r_k = 0,$$

which completes the proof of superlinear convergence.

TABLE I
FUNCTIONS OF EXPERIMENT

No.	Function
1	Extended Rosebrock
2	Logarithmic
3	Brown almost linear
4	Penalty
5	Trigonometric
6	Broyden tridiagonal
7	Broyden banded
8	Discrete boundary value
9	Extended Freudentein and Roth

IV. NUMERICAL EXPERIMENTS

In this section, we report the results of some numerical experiments with the proposed algorithm. We also give another two quasi-Newton algorithms determined by the line search condition (1.5) and the trust region (1.6), and we call them Algorithms 2 and 3, respectively.

We choose 9 test functions [10, 11, 14] listed in Table 1.

In the experiments, the parameters are chosen as $\varepsilon = 10^{-5}, \rho = 0.0001, c = 0.5.$ The initial quasi-Newton matrix $B_0 = A$ (see [8]). For Algorithms 1 and 3, we obtain d_k from Dogleg method (see [13]). For Algorithm 3, we update B_{k+1} by (1.7) if $y_k^T d_k > 10^{-5}$, otherwise, we set $B_{k+1} = B_k$. We stop the iteration process if $\|F_k\| \le 10^{-5}$, and stop the program if the iteration number is larger than 5000. The program is coded in MATLAB 9.0.

To compare the efficiency of the three algorithms, we use the performance profile proposed by Dolan and More [4]. The dimensions of test functions 1-9 are 50. According to the numerical results, we plot two figures based on the total number of iterations and the CPU time, respectively.

From Figure 1, we find that Algorithm 2 is slightly better than Algorithm 3, Algorithm 1 obviously performs better than Algorithms 2 and 3 on the total number of iterations. From Figure 2, we can see that Algorithms 2 and 3 have similar performances. There are no large discrepancies on the CPU time because a curve crosses another curve. It is clear that our algorithm needs fewer CPU time than Algorithms 2 and 3 in the test problems.

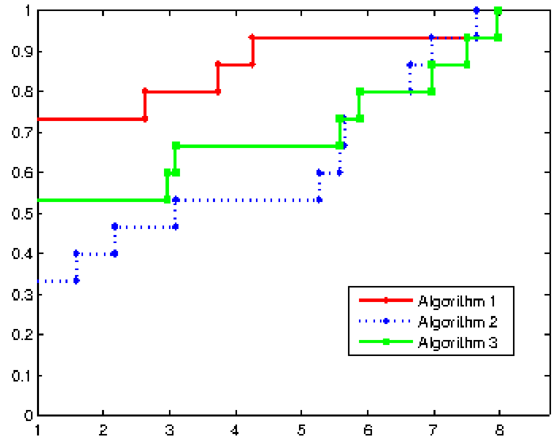

Figure 1: Performance profile of the number of iterations

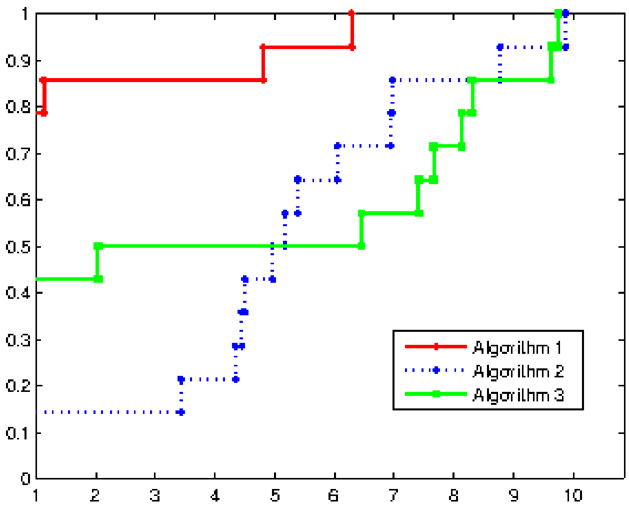

Figure 2: Performance profile of the CPU time

V. Conclusions

In this paper, we present a Broyden quasi-Newton method to solve nonlinear equations with a new trust region radius. The global and superlinear convergence is established under suitable assumptions. Preliminary numerical results show that our algorithm is promising.

References

[1] J.E. Dennis, J.J. More, "A characterization of superlinear convergence and its application to quasi-Newton methods," *Mathematics of computation*, vol. 28, no. 126, pp. 549-560, 1974.

[2] J.E. Dennis, J.J. More, "Quasi-Newton methods, motivation and theory," *SIAM review*, vol. 19, no. 1, pp. 46-89, 1977.

[3] J.E. Dennis, R.B. Schnabel, "Numerical methods for unconstrained optimization and nonlinear equation," Science Press, Beijing, 2009.

[4] E.D. Dolan, J.J. More, "Benchmarking optimization software with performance profiles," *Math. Program.*, vol. 91, pp. 201-213, 2002.

[5] A. Griewank, "The `global' convergence of Broyden-like methods with a suitable line search," *J. Austral. Math. Soc. Ser. B*, vol. 28, no. 01, pp. 75-92, 1986.

[6] G.Z. Gu, L. Qi, "Descent directions of quasi-Newton methods for symmetric nonlinear equations," *SIAM J. Numer. Anal.*, vol. 40, no. 5, pp. 1763-1774, 2002.

[7] D.H. Li, M. Fukushima, "A derivative-free line search and global convergence of Broyden-like method for nonlinear equations," *Optimization Methods and Software*, vol. 13, no. 3, pp. 181-201, 2000.

[8] D.H. Li, M. Fukushima, "A global and superlinear convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations," *SIAM J. Numer. Anal.*, vol. 37, no. 1, pp. 152-172,1999.

[9] D.H. Li, X.L. Wang, "A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations," *Numerical Algebra Control Optimization*, vol. 1, no. 1, pp. 71-82, 2011.

[10] J.J. More, B.S. Garbow, K.E. Hillstrom, "Testing unconstained optimization software," *ACM Transaction Mathematical Software*, vol. 7, no. 1, pp. 17-41, 1981.

[11] J. Nocedal, S.J. Wright, "Numerical optimization," Science Press, Beijing, 2006.

[12] H. Wang, M. Qin, "A modified regularized newton method for unconstrained convex optimization," *IAENG International Journal of Applied Mathematics*, vol.46, no.2, pp. 130-134, 2016.

[13] Y.J. Wang, N.H. Xiu, "Theory and algorithm for nonlinear programming," Shanxi Science and Technology Press, Xi'an, 2004.

[14] G.L. Yuan, Z.X. Wei, X. W. Liu, "A BFGS trust-region method for nonlinear equations," *Computing* , vol. 92, no. 4, pp. 317-333, 2011.

[15] M.L. Zeng, Q. Ni, "A new trust region method for nonlinear equations involving fractional model," *Pacific J. Optim.*, vol.15, no.2, pp. 317-329, 2019.

[16] W.J. Zhou, D.M. Shen, "An inexact PRP conjugate gradient method for symmetric nonlinear equations," *Numerical Functional Analysis and Optimization*, vol. 35, no. 3, pp. 370-388, 2014.