Improved Convergence Results of a BFGS Trust Region Quasi-Newton Method for Nonlinear Equations

Hongwei Fu*

Abstract—**Quasi-Newton method is one of the most effective methods for solving nonlinear equations. In this paper, we improve convergence results of a BFGS trust region quasi-Newton method for nonlinear equations. The global and superlinear convergence are proved under the local error bound and the Hölderian continuity conditions, which are weaker than the nonsingularity and the Lipschitz continuity, respectively. Numerical results show that the algorithm is efficient and promising.**

Index Terms—**nonlinear equations, BFGS, quasi-Newton method, local error bound, Hölderian continuity**

I. Introduction

CONSIDER the numerical solution of systems of nonlinear equations

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

where $F : R^n \to R^n$ is continuously differentiable, $F(x) = (f_1(x), \cdots, f_n(x))^T$. Let $J(x)$ denote the Jacobian matrix of F at x, and $J(x) = (\nabla f_1(x), \cdots, \nabla f_n(x))^T$. Throughout the paper, we assume that the solution set of (1), denoted by X^*, is nonempty. In all cases, $\|\cdot\|$ denotes 2-norm. Let $f(x) = \frac{1}{2}\|F(x)\|^2$. The nonlinear equation problem (1) is equivalent to the global optimization problem

$$\min f(x), x \in R^n. \tag{2}$$

Many algorithms have been presented for solving the problem (1), for examples, Newton method [4], [22], quasi-Newton method [3], [9], [10], [13], [14], [23], Gauss-Newton method [12], [16], Levenberg-Marquardt method [5], [6], [20], tensor method [1], [8], [17], etc. Quasi-Newton method is one of the most effective methods among them. An attractive feature of quasi-Newton method is its local superlinear convergence property without computing the Jacobians.

Conventional quasi-Newton methods [2], [3] for solving (1) generate a sequence of iterates $\{x_k\}$ by letting $x_{k+1} = x_k + d_k$, where d_k is a solution of the following system of linear equations:

$$F_k + B_k d = 0, \tag{3}$$

where $F_k = F(x_k), B_k$ is an approximation of $J_k = J(x_k)$. Since

$$\nabla f(x_k)^T d_k = -F_k^T J_k B_k^{-1} F_k, \tag{4}$$

Manuscript received December 11, 2019. This work was supported by the University Development Reform and Research Project of Hubei Engineering University (HGFGZX201803) and University-Industry Collaborative Education Program (201901104009, 201901032002).

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

then d_k is not necessarily a descent direction of f at x_k. One way to globalize such quasi-Newton methods is to exploit the line search rule given by Griewank [9] and Li [11]. In [11], λ_k satisfies the line search condition

$$\|F(x_k + \lambda_k d_k)\| \le (1 + \eta_k)\|F(x_k)\| - \sigma_1\|\lambda_k d_k\|^2, \tag{5}$$

where $\sigma_1 > 0, \eta > 0, \Sigma_k \eta_k \le \eta < \infty$. The global and superlinear convergence are obtained under the condition that $J(x^*)$ (x^* is the solution of (1)) is nonsingular. Another way is to employ the trust region strategy [13], [18].

Yuan et al. [21] proposed a BFGS trust region method for solving nonlinear equations:

$$\min q_k(d) \quad \text{such that} \quad \|d\| \le \Delta_k, \tag{6}$$

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, B_{k+1} is generated by the BFGS formula

$$B_{k+1} = B_k - \frac{B_k d_k d_k^T B_k}{d_k^T B_k d_k} + \frac{y_k y_k^T}{y_k^T d_k}, \tag{7}$$

where $d_k = x_{k+1} - x_k, y_k = F_{k+1} - F_k$, B_k is nonsingular. B_{k+1} satisfies the quasi-Newton equation $B_{k+1} d_k = y_k$. The global and superlinear convergence are obtained without the nonsingularity assumption.

In real applications, some nonlinear equations may not satisfy the nonsingularity condition, but they satisfy the local error bound condition defined as follows.

Definition 1.1: We say $\|F(x)\|$ provides a local error bound in some neighbourhood of $x^* \in X^*$, if there exists a constant $c_0 > 0$ such that

$$c_0 \cdot dist(x, X^*) \le \|F(x)\|, \forall x \in N(x^*), \tag{8}$$

where $dist(x, X^*)$ is the distance from x to X^* and $N(x^*)$ is some neighbourhood of $x^* \in X^*$ (cf. [7], [22]).

For example, the nonlinear function given by $F(x_1, x_2) = (e^{x_1} - 1, 0)^T$, the solution set of $F(x_1, x_2) = 0$ is $X^* = \{x \in R^2 | x_1 = 0\}$, $dist(x, X^*) = |x_1|$. $F(x_1, x_2)$ has the local error bound in the $I = \{x \in R^2 | -\epsilon < x_1 < \epsilon\}$, where $\epsilon > 0$ is small enough. However, $J(x^*)$ is singulary for all $x^* \in X^*$.

Convergence results of the BFGS quasi-Newton method are improved in this paper. Global and superlinear convergence are proved under the conditions of local error bound and Hölderian continuity. The local error bound is weaker than the nonsingularity, and the Hölderian continuity is more general than the Lipschitz continuity.

The paper is organized as follows. Section 2 gives the BFGS trust region quasi-Newton method and proves its global convergence. Local convergence rate is presented in

Section 3. Section 4 demonstrates the numerical results of test problems. Conclusions are drawn in Section 5.

II. BFGS TRUST REGION QUASI-NEWTON METHOD AND ITS GLOBAL CONVERGENCE

In this section, we give the BFGS trust region quasi-Newton method for solving nonlinear equations, and prove its global convergence.

Let d_k^p be the solution of the trust region subproblem (6) corresponding to p. We define the actual reduction as

$$Ared_k(d_k^p) = f(x_k) - f(x_k + d_k^p), \tag{9}$$

the predict reduction as

$$Pred_k(d_k^p) = q_k(0) - q_k(d_k^p). \tag{10}$$

and the radio of actual reduction over predict reduction as

$$r_k^p = \frac{Ared_k(d_k^p)}{Pred_k(d_k^p)}. \tag{11}$$

The algorithm is given as follows.

Algorithm 2.1.

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 $\|F_k\| \leq \varepsilon$, terminate.

Step 2. Solve the subproblem (6) to obtain d_k^p.

Step 3. Compute

$$r_k^p = \frac{Ared_k(d_k^p)}{Pred_k(d_k^p)}. \tag{12}$$

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

Step 4. If $y_k^T d_k^p > 0$, update B_{k+1} by (7). Otherwise, let $B_{k+1} = B_k$.

Step 5. Set $k := k + 1$ and $p = 0$. Go to Step 1.

In order to prove the global convergence of Algorithm 2.1, we make the following assumptions.

Assumption 2.1 (1) The level set $\Omega = \{x \in R^n | f(x) \leq f(x_0)\}$ is bounded.

(2) $J(x)$ is bounded, i.e., there exists a positive constant L such that

$$\|J(x)\| \leq L, \forall x \in R^n. \tag{13}$$

(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 there exist positive constants $0 < M_0 \leq M$ such that

$$M_0 \leq \|B_k\| \leq M, \ \forall k. \tag{14}$$

By (13), we have

$$\|F(y) - F(x)\| \leq L\|y - x\|, \forall x, y \in R^n. \tag{15}$$

Lemma 2.1: $|Ared_k(d_k^p) - Pred_k(d_k^p)| = O(\|d_k^p\|^2) = O(\Delta_k^2)$.

Proof: By (9) and (10), we have

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

This completes the proof. ■

Lemma 2.2: If d_k^p is a solution of (6), then

$$Pred_k(d_k^p) \geq \frac{1}{2}\|B_k F_k\| \min\left\{\Delta_k, \frac{\|B_k F_k\|}{\|B_k\|^2}\right\}. \tag{16}$$

Proof: Since d_k^p is a solution of (6), for any $t \in [0, 1]$, it follows

$$\begin{aligned} Pred_k(d_k^p) &= \frac{1}{2}\left(\|F_k\|^2 - \|F_k + B_k d_k^p\|^2\right) \\ &\geq \frac{1}{2}\left(\|F_k\|^2 - \|F_k - B_k \frac{t\Delta_k}{\|B_k F_k\|} B_k F_k\|^2\right) \\ &\geq t\Delta_k\|B_k F_k\| - \frac{1}{2}t^2\Delta_k^2\|B_k\|^2. \end{aligned} \tag{17}$$

Therefore,

$$\begin{aligned} Pred_k(d_k^p) &\geq \max_{0 \leq t \leq 1}\left[t\Delta_k\|B_k F_k\| - \frac{1}{2}t^2\Delta_k^2\|B_k\|^2\right] \\ &\geq \frac{1}{2}\|B_k F_k\| \min\left\{\Delta_k, \frac{\|B_k F_k\|}{\|B_k\|^2}\right\}. \end{aligned} \tag{18}$$

This completes the proof. ■

Lemma 2.3: Algorithm 2.1 does not circle between Steps 2 and 3 infinitely.

Proof: If Algorithm 2.1 circles between Steps 2 and 3 infinitely, i.e., $p \to \infty, r_k^p < \rho$ and $c^p \to 0$. Obviously, $\|F_k\| > \varepsilon$, otherwise the algorithm stops. Thus, $\|d_k^p\| \leq \Delta_k = c^p\|F_k\| \to 0$.

From Lemmas 2.1 and 2.2, it follows

$$|r_k^p - 1| = \frac{|Ared_k(d_k^p) - Pred_k(d_k^p)|}{|Pred_k(d_k^p)|} \leq \frac{2O(\Delta_k^2)}{\Delta_k\|B_k F_k\|} \to 0. \tag{19}$$

Therefore, for all k big enough,

$$r_k^p \geq \rho, \tag{20}$$

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

Theorem 2.1: Let Assumption 3.1 hold and $\{x_k\}$ be generated by Algorithm 2.1. Then the algorithm either stops finitely or

$$\lim_{k\to\infty} \inf \|F_k\| = 0. \tag{21}$$

Proof: Assume that Algorithm 2.1 does not stop after finite steps, we prove that the following relation

$$\lim_{k\to\infty} \inf \|B_k F_k\| = 0 \tag{22}$$

is true. By (14), we can obtain (21). Therefore, in order to get this theorem, we must show (22).

Assume that (22) is not true, i.e., there exists a positive constant ε and an infinite subsequence $\{k_j\}$ such that

$$\|B_{k_j} F_{k_j}\| \geq \varepsilon.$$

Let $K = \{k | \|B_k F_k\| \geq \varepsilon\}$. By (14), we can assume $\|F_k\| \geq \varepsilon, \forall k \in K$.

Using Algorithm 2.1 and Lemma 2.2, we have

$$\begin{aligned}\sum_{k\in K}[f(x_k)-f(x_{k+1})] &\geq \sum_{k\in K}\rho\cdot Pred_k(d_k^p)\\ &\geq \sum_{k\in K}\rho\cdot\frac{\varepsilon}{2}\min\left\{\Delta_k,\frac{\varepsilon}{M^2}\right\},\end{aligned}$$

Because $\{f(x_k)\}$ 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 $k\to+\infty$ and $k\in K$. Therefore, we can assume $p_k\geq 1$ for all $k\in K$.

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

$$\begin{aligned}\min\quad q_k(d) &= \frac{1}{2}\|F_k+B_kd\|^2 \\ s.t.\quad \|d\| &\leq c^{p_k-1}\|F_k\|\end{aligned}\tag{23}$$

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

$$\frac{f(x_k)-f(\tilde{x}_{k+1})}{Pred_k(\tilde{d}_k)}<\rho.\tag{24}$$

From Lemma 2.2, it follows

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

By Lemma 2.1, we get

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

Therefore,

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

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 contradicts (24). This completes the proof. ∎

Remark Theorem 2.1 shows that the iterative sequence $\{x_k\}$ generated by Algorithm 2.1 satisfies $\|F_k\|\to 0$ without the assumption that $J(x^*)$ is nonsingular, where x^* is a cluster point of $\{x_k\}$.

III. Local convergence rate

In this section, we study some convergence properties of Algorithm 2.1 under the Hölderian continuity and the local error bound conditions.

In order to establish the local superlinear convergence of Algorithm 2.1, we assume that the sequence $\{x_k\}$ generated by Algorithm 2.1 lies in some neighborhood of $x^*\in X^*$ and converges to the solution set X^* of (1). The following assumption is further needed.

Assumption 3.1 (1) There exists a constant $c_1\in[1,+\infty)$ and $0<b<1$ such that

$$c_1\cdot dist(x,X^*)\leq\|F(x)\|,\forall x\in N(x^*,b),\tag{25}$$

where $N(x^*,b)=\{x\in R^n|\|x-x^*\|\leq b\}$.

(2) $J(x)$ is the Hölderian continuous of order $\upsilon\in(0,1]$, i.e., there exists a positive constant c_2 such that

$$\|J(x)-J(y)\|\leq c_2\|x-y\|^\upsilon,\forall x,y\in N(x^*,b).\tag{26}$$

(3) B_k is a good approximation to J_k, i.e., $\|B_k-J_k\|=O(dist(x_k,X^*))$.

When $\upsilon=1$ in (26), $J(x)$ is Lipschitz continuous, which implies that the Hölderian continuity is more general than the Lipschitz continuity. By (26), for $\forall x,y\in N(x^*,b)$, we have

$$\begin{aligned}&\|F(y)-F(x)-J(x)(y-x)\|\\ =&\left\|\int_0^1 J(x+t(y-x))(y-x)dt-J(x)(y-x)\right\|\\ \leq&\|y-x\|\int_0^1\|J(x+t(y-x))-J(x)\|dt\\ \leq&c_2\|y-x\|^{1+\upsilon}\int_0^1 t^\upsilon dt\\ =&\frac{c_2}{1+\upsilon}\|y-x\|^{1+\upsilon}.\end{aligned}$$

Theorem 3.1: Let Assumptions 2.1 and 3.1 hold, $\{x_k\}$ be generated by Algorithm 2.1. Then, for k large enough, the iteration formula is as follows

$$x_{k+1}=x_k+d_k^0,$$

where d_k^0 is the solution of (6) corresponding to $p=0$, and

$$dist(x_{k+1},X^*)=O((dist(x_k,X^*))^{1+\upsilon}),\tag{27}$$

where $\upsilon\in(0,1]$. i.e., Algorithm 2.1 is superlinearly convergent.

Proof: In what follows, let $\bar{x}_k\in X^*$ such that

$$dist(x_k,X^*)=\|x_k-\bar{x}_k\|.\tag{28}$$

Without loss of generality, we assume that x_k lies in $N(x^*,\frac{b}{2})$,

$$\|x_k-x^*\|\leq\frac{b}{2}.$$

Then

$$\|x_k-\bar{x}_k\|\leq\|x_k-x^*\|\leq\frac{b}{2}.$$

Thus,

$$\|\bar{x}_k-x^*\|\leq\|\bar{x}_k-x_k\|+\|x_k-x^*\|\leq b.$$

So $\bar{x}_k\in N(x^*,b)$ and

$$\|x_k-\bar{x}_k\|\leq c_1\|x_k-\bar{x}_k\|\leq\|F(x_k)\|.$$

Since p starts from 0 at each iterative point x_k, $\bar{x}_k-x_k$ is a feasible point of (6) corresponding to $p=0$ for k large enough. Hence it follows from Assumption 3.2 (2) and (3) that

$$\begin{aligned}&q_k(d_k^0)\\ \leq&q_k(\bar{x}_k-x_k)=\frac{1}{2}\|F_k+B_k(\bar{x}_k-x_k)\|^2\\ =&\frac{1}{2}\|F_k-F(\bar{x}_k)+J_k(\bar{x}_k-x_k)+(B_k-J_k)(\bar{x}_k-x_k)\|^2\\ \leq&\frac{1}{2}\|F(\bar{x}_k)-F_k-J_k(\bar{x}_k-x_k)\|^2\\ &+\frac{1}{2}\|B_k-J_k\|^2\|\bar{x}_k-x_k\|^2\\ =&O(\|x_k-\bar{x}_k\|^{2(1+\upsilon)})\\ =&O((dist(x_k,X^*))^{2(1+\upsilon)}).\end{aligned}$$

Therefore,

$$q_k(d_k^0) = \frac{1}{2}\|F_k + B_k d_k^0\|^2 = O((dist(x_k, X^*))^{2(1+v)}). \quad (29)$$

By (6) and (15), we have

$$\|d_k^0\| \le \|F_k\| = \|F_k - F(\bar{x}_k)\| = O(\|x_k - \bar{x}_k\|).$$

Therefore, for k large enough, we obtain $x_k + d_k^0 \in N(x^*, b)$ and

$$\begin{aligned} & c_1 \cdot dist(x_k + d_k^0, X^*) \\ \le\ & \|F(x_k + d_k^0)\| \\ \le\ & \|F_k + J_k d_k^0\| + O(\|d_k^0\|^2) \\ =\ & \|F_k + B_k d_k^0 + (J_k - B_k) d_k^0\| + O(\|d_k^0\|^2) \\ \le\ & \|F_k + B_k d_k^0\| + \|B_k - J_k\|\|d_k^0\| + O(\|d_k^0\|^2) \\ =\ & O(\|x_k - \bar{x}_k\|^{1+v}) \\ =\ & O((dist(x_k, X^*))^{1+v}). \end{aligned}$$

So we obtain

$$\|F(x_k + d_k^0)\| = O((dist(x_k, X^*))^{1+v}) \quad (30)$$

and

$$dist(x_k + d_k^0, X^*) = O((dist(x_k, X^*))^{1+v}). \quad (31)$$

In the following we prove that for k sufficiently large the iteration formula is as follows

$$x_{k+1} = x_k + d_k^0. \quad (32)$$

For k large enough, (29) and (30) imply

$$\begin{aligned} & |Ared_k(d_k^0) - Pred_k(d_k^0)| \\ =\ & \left|\frac{1}{2}\|F(x_k + d_k^0)\|^2 - q_k(d_k^0)\right| \\ \le\ & \frac{1}{2}\|F(x_k + d_k^0)\|^2 + q_k(d_k^0) \\ =\ & O(\|x_k - \bar{x}_k\|^{2(1+v)}) \\ =\ & O((dist(x_k, X^*))^{2(1+v)}). \quad (33) \end{aligned}$$

By Assumption 3.1 (2) and (29), we obtain

$$\begin{aligned} |Pred_k(d_k^0)| &= \left|\frac{1}{2}\|F_k\|^2 - q_k(d_k^0)\right| \\ &\ge \frac{1}{2}\|F_k\|^2 - q_k(d_k^0) \\ &\ge \frac{1}{2}c_1^2\|x_k - \bar{x}_k\|^2 + O(\|x_k - \bar{x}_k\|^{2(1+v)}) \\ &= O(\|x_k - \bar{x}_k\|^2) \\ &= O((dist(x_k, X^*))^2). \quad (34) \end{aligned}$$

Combining (33) and (34), we have

$$\lim_{k\to\infty} |r_k^0 - 1| = \lim_{k\to\infty} \frac{|Ared_k(d_k^0) - Pred_k(d_k^0)|}{|Pred_k(d_k^0)|} = 0.$$

Therefore, for sufficiently large k, $r_k^0 \ge \rho$, then the iteration formula is (32). By (31) and (32), we get (27). This completes the proof. ■

IV. Numerical experiments

In this section, we report some numerical experiments to show that Algorithm 2.1 is an effective algorithm for solving nonlinear equations.

All codes are written in MATLAB R2016 programming environment on a personal PC with 2.5 GHz and 2.7 GHz, 8.0 GB RAM, using Windows 10 operation system. The algorithms are terminated when the number of iterations exceeds 3000 or $\|F(x_k)\| \le 10^{-5}$.

In the experiments, we choose the parameters $c = 0.5, \rho = 0.001$. The initial quasi-Newton matrix is set to be $B_0 = I$. For Step 4 of Algorithm 2.1, we update B_k by (7) if $y_k^T s_k > 10^{-5}$, otherwise, we set $B_{k+1} = B_k$. d_k in (6) is determined by Dogleg method in [19]. The results are summarized in Table I, where Dim is the dimension of the functions [15], Iters is the total number of iterations, Time is the average time of iterations and measured in seconds.

TABLE I
NUMERICAL RESULTS OF ALGORITHM 2.1

Function	Dim	Iters	Time(s)	$\|F(x_k)\|$
convex	50	6	0.0386	5.8511e-07
disbound	50	45	0.0187	9.332e-06
logarithmic	50	5	0.0518	6.5924e-06
exponential	50	349	0.0398	9.9839e-06
singular	50	278	0.0243	2.0516e-06
band	100	19	0.0578	3.0534e-06
logarithmic	100	5	0.0030	1.5081e-06
vardim	100	10	0.0048	1.1206e-07
convex	100	6	0.0027	7.6163e-07
disbound	100	96	0.0463	9.3941e-06
band	1000	20	0.8242	3.3348e-06
logarithmic	1000	34	0.2776	3.8839e-08
vardim	1000	65	1.0216	2.2513e-06
convex	1000	21	0.1524	6.0492e-09
disbound	1000	173	1.2143	9.9815e-06

V. Conclusions

In this paper, we improve convergence results of the BFGS quai-Newton method for nonlinear equations in [21]. The global and superlinear convergence are obtained under the local error bound and the Hölderian continuity conditions. The local error bound is weaker than the nonsingularity of Jacobian, and the Hölderian continuity is more general than the Lipschitz continuity. Numerical experiments demonstrate that the algorithm is efficient and promising.

References

[1] A. Bouaricha, R.B. Schnabel, "Tensor methods for large sparse systems of nonlinear equations," *Math. Program.*, vol.82, pp.377-400, 1998.

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

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

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

[5] J.Y. Fan, "A modified Levenberg-Marquardt algorithm for singular system of nonlinear equations," *J. Comput. Math.*, vol.21, no.5, pp.625-636, 2003.

[6] J.Y. Fan, J.C. Huang, J.Y. Pan, "An adaptive multi-step Levenberg-Marquardt method," *Journal of Scientific Computing*, vol.78, no.1, pp.531-548, 2019.

[7] J.Y. Fan, J.Y. Pan, "A modified trust region algorithm for nonlinear equations with new updating rule of trust region radius," *International Journal of Computer Mathematics*, vol.87, no.14, pp.3186-3195, 2010.

[8] D. Feng, P.D. Frank, R.B. Schnabel, "Local convergence analysis of tensor methods for nonlinear equations," *Math. Program.*, vol.62, no.1, pp.427-459, 1993.
[9] A. Griewank, "The 'global' convergence of Broyden-like methods with a suitable line search," *J. Austral. Math. Soc. Ser. B.*, vol.28, no.1, pp.75-92, 1986.
[10] G.Z. Gu, D.H. Li, L.Q. Qi and S.Z. Zhou, "Descent directions of quasi-Newton methods for symmetric nonlinear equations," *SIAM J. Numer. Anal.*, vol.40, no.5, pp.1763-1774, 2002.
[11] 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.
[12] 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.
[13] X.P. Lu, Q. Ni, "A quasi-Newton trust region method with a new conic model for the unconstrained optimization," *Applied Mathematics and Computation*, vol.204, no.1, pp.373-384, 2008.
[14] J.M. Martinez, "Practical quasi-Newton methods for solving nonlinear systems," *J. Comput. Appl. Math.*, vol.124, no.1, pp.97-121, 2000.
[15] J.J. Moré, B.S. Garbow, K.E. Hillstrom, "Testing unconstained optimization software," *ACM Transactions on Mathematical software*, vol.7, no.1, pp.17-41, 1981.
[16] J. Nocedal, S.J. Wright, "Numerical optimization," Science Press, Beijing, 2006.
[17] R.B. Schnabel, P.D. Frank, "Tensor methods for nonlinear equations," *SIAM J. Numer. Anal.*, vol.21, no.5, pp.815-843, 1984.
[18] W. Sun, Y. Yuan, "Optimization Theory and Method," Nonlinear Programming, Springer, New York, 2006.
[19] Y.J. Wang, N.H. Xiu, "Theory and algorithm for nonlinear programming," Shanxi Science and Technology Press, Xi'an, 2004.
[20] N. Yamashita, M. Fukushima, "On the rate of convergence of the Levenberg-Marquardt method," *Computing*, vol.15, pp.239-249, 2001.
[21] 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.
[22] J.L. Zhang, Y. Wang, "A new trust region method for nonlinear equations," *Mathematical Methods of Operations Research*, vol.58, no.2, pp.283-298, 2003.
[23] M.L. Zeng, H.W. Fu, "A Broyden trust region quasi-Newton method for nonlinear equations," *IAENG International Journal of Computer Science*, vol.46, no.3, pp.454-458, 2019.