Vol.2, No.4, 373-378 (2010)
http://dx.doi.org/10.4236/ns.2010.24045

Natural Science

A nonmonotone adaptive trust-region algorithm for symmetric nonlinear equations*

Gong-Lin Yuan[1], Cui-Ling Chen[2], Zeng-Xin Wei[1]

[1]College of Mathematics and Information Science, Guangxi University, Nanning, China; glyuan@gxu.edu.cn
[2]College of Mathematics Science, Guangxi Normal University, Guilin, China

Received 23 December 2009; revised 1 February 2010; accepted 3 February 2010.

ABSTRACT

In this paper, we propose a nonmonotone adaptive trust-region method for solving symmetric nonlinear equations problems. The convergent result of the presented method will be established under favorable conditions. Numerical results are reported.

Keywords: Trust Region Method; Global Convergence; Symmetric Nonlinear Equations

1. INTRODUCTION

Consider the following system of nonlinear equations:

$$g(x) = 0, x \in R^n \tag{1}$$

where $g: R^n \to R^n$ is continuously differentiable, the Jacobian $\nabla g(x)$ of g is symmetric for all $x \in R^n$. Define a norm function by $\varphi(x) = \frac{1}{2}\|g(x)\|^2$. It is not difficult to see that the nonlinear equations problem **Eq.1** is equivalent to the following global optimization problem

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

Here and throughout this paper, we use the following notations.

- $\|\cdot\|$ denote the Euclidian norm of vectors or its induced matrix norm.
- $\{x_k\}$ is a sequence of points generated by an algorithm, and $g(x_k)$ and $\varphi(x_k)$ are replaced by g_k and φ_k respectively.
- B_k is a symmetric matrix which is an approximation of $\nabla g(x)^T \nabla g(x)$.

It is well known that there are many methods for the unconstrained optimization problem $\min_{x \in R^n} f(x)$ (see [1-7], *etc.*), where the trust-region methods are very successful, *e.g.*, Moré and Sorensen [8]. Other classical references on this topic are [9-12]. Trust- region methods have been applied to equality constrained problems [13-16]. Many authors have studied the trust-region method [2,17-22] too. Zhang [23] combined the trust region subproblem with nonmonotone technique to present a nonmonotone adaptive trust region method and studied its convergence properties.

$$\begin{aligned} &\min \ \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\ &s.t. \ \|d\| \le h_k, \ d \in R^n \end{aligned} \tag{3}$$

where H_k is the Hessian of some function $f: R^n \to R$ at x_k or an approximation to it, $h_k = c_1^{p_1}\|\nabla f(x_k)\| M'_k$, $0 < c_1 < 1$, $M'_k = \|B_k'^{-1}\|$, p_1 is a nonnegative integer, they adjust p_1 instead of adjusting the trust radius, and B'_k is a safely positive definite matrix based on Schnabel and Eskow [24] modified cholesky factorization, $B'_k = H_k + E_k$, where $E_k = 0$ if H_k is safely positive definite, and E_k is a diagonal matrix chosen to make B'_k positive definite otherwise.

For nonlinear equations, Griewank [25] first established a global convergence theorem for quasi-Newton method with a suitable line search. One nonmonotone backtracking inexact quasi-Newton algorithm [26] and the trust region algorithms [27-30] were presented. A Gauss-Newton-based BFGS method is proposed by Li and Fukushima [31] for solving symmetric nonlinear equations. Inspired by their ideas, Wei [32] and Yuan [33-37] made a further study. Recently, Yuan and Lu [38] presented a new backtracking inexact BFGS method for symmetric nonlinear equations.

Inspired by the technique of Zhang [23], we propose a new nonmotone adaptive trust region method for solving

*Foundation item: National Natural Science Foundation of China (10761001), the Scientific Research Foundation of Guangxi University (Grant No. X081082), Guangxi SF grands 0991028, the Scientific Research Foundation of Guangxi Education Department (Grant No. 200911LX53), and the Youth Backbone Teacher Foundation of Guangxi Normal University.

Eq.1. More precisely, we solve **Eq.1** by the method of iteration and the main step at each iteration of the following method is to find the trial step d_k. Let x_k be the current iteration. The trial step d_k is a solution of the following trust region subproblem

$$\min\ q_k(d) = \nabla\varphi(x_k)^T d + \frac{1}{2} d^T B_k d$$
$$s.\,t.\ \|d\| \le \Delta_k,\ d \in R^n \tag{4}$$

where $\nabla\varphi(x_k) = \nabla g(x_k) g(x_k)$, $\Delta_k = c^p \|\nabla\varphi(x_k)\| M_k$, $0 < c < 1$, $M_k = \|B_k^{-1}\|$, p is a nonnegative integer, and matrix B_k is an approximation of $\nabla g(x_k)^T g(x_k)$ which is generated by the following BFGS formula [31]:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{5}$$

where $s_k = x_{k+1} - x_k$, $y_k = g(x_k + \delta_k) - g_k$, $\delta_k = g_{k+1} - g_k$. By $y_k = g(x_k + \delta_k) - g_k$, we have the approximate relations

$$y_k = g(x_k + \delta_k) - g_k \approx \nabla g_{k+1}\delta_k \approx \nabla g_{k+1}\nabla g_{k+1} s_k$$

Since B_{k+1} satisfies the secant equation $B_{k+1}s_k = y_k$ and ∇g_k is symmetric, we have approximately

$$B_{k+1} \approx \nabla g_{k+1}\nabla g_{k+1} s_k = \nabla g_{k+1}^T \nabla g_{k+1} s_k$$

This means that B_{k+1} approximates $\nabla g_{k+1}^T \nabla g_{k+1}$ along direction s_k. We all know that the update **Eq.5** can ensure the matrix B_{k+1} inherits positive property of B_k if the condition $s_k^T y_k > 0$ is satisfied. Then we can use this way to insure the positive property of B_k.

This paper is organized as follows. In the next section, the new algorithm for solving **Eq.1** is represented. In Section 3, we prove the convergence of the given algorithm. The numerical results of the method are reported in Section 4.

2. THE NEW METHOD

In this section, we give our algorithm for solving **Eq.1**. Firstly, one definition is given. Let

$$\varphi_{l(k)} = \max_{0 \le j \le n(k)} \{\varphi_{k-j}\},\ k = 0,1,2,\cdots \tag{6}$$

where $n(k) = \min\{M, k\}$, $M \ge 0$ is an integer constant. Now the algorithm is given as follows.

- **Algorithm 1.**

Initial: Given constants $\rho, c \in (0,1)$, $p = 0$, $\varepsilon > 0$, $M \ge 0$, $x_0 \in R^n$, $B_0 \in R^n \times R^n$. Let $k := 0$;

Step 1: If $\|\nabla\varphi_k\| < \varepsilon$, stop. Otherwise, go to step 2;

Step 2: Solve the problem **Eq.4** with $\Delta = \Delta_k$ to get d_k;

Step 3: Calculate $n(k)$, $\varphi_{l(k)}$ and the following r_k:

$$r_k = \frac{\varphi_{l(k)} - \varphi(x_k + d_k)}{q_k(0) - q_k(d_k)} \tag{7}$$

If $r_k < \rho$, then we let $p = p + 1$, go to step 2. Otherwise, go to step 4;

Step 4: Let $x_{k+1} = x_k + d_k$, $\delta_k = g_{k+1} - g_k$, $y_k = g(x_k + \delta_k) - g_k$. If $d_k^T y_k > 0$, update B_{k+1} by **Eq.5**, otherwise let $B_{k+1} = B_k$.

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

Remark. i) In this algorithm, the procedure of "Step 2-Step 3-Step 2" is named as inner cycle.

ii) The Step 4 in Algorithm 1 ensures that the matrix sequence $\{B_k\}$ is positive definite.

In the following, we give some assumptions.

Assumption A. j) Let Ω be the level set defined by

$$\Omega = \{x \,|\, \|g(x)\| \le \|g(x_0)\|\} \tag{8}$$

is bounded and $g(x)$ is continuously differentiable in Ω for all any given $x_0 \in R^n$.

jj) The matrices $\{B_k\}$ are uniformly bounded on Ω_1, which means that there exists a positive constant M such that

$$\|B_k\| \le M,\ \forall k \tag{9}$$

Based on Assumption A and Remark (ii), we have the following lemma.

Lemma 2.1. Suppose that Assumption A(jj) holds. If d_k is the solution of **Eq.4**, then we have

$$-q_k(d_k) \ge \frac{1}{2}\|\nabla\varphi(x_k)\| \min\{\Delta_k, \frac{\|\nabla\varphi(x_k)\|}{\|B_k\|}\} \tag{10}$$

Proof. Using d_k is the solution of **Eq.4**, for any $\alpha \in [0,1]$, we get

$$-q_k(d_k) \ge -q_k(-\alpha \frac{\Delta_k}{\|\nabla\varphi(x_k)\|}\nabla\varphi(x_k))$$

$$= \alpha\Delta_k \|\nabla\varphi(x_k)\| - \frac{1}{2}\alpha^2\Delta_k^2 (\nabla\varphi(x_k))^T B_k \nabla\varphi(x_k) / \|\nabla\varphi(x_k)\|^2$$

$$\ge \alpha\Delta_k \|\nabla\varphi(x_k)\| - \frac{1}{2}\alpha^2\Delta_k^2 \|B_k\|$$

Then, we have

$$-q_k(d_k) \ge \max_{0 \le \alpha \le 1}[\alpha\Delta_k \|\nabla\varphi(x_k)\| - \frac{1}{2}\alpha^2\Delta_k^2 \|B_k\|]$$

$$\ge \frac{1}{2}\|\nabla\varphi(x_k)\| \min\{\Delta_k, \frac{\|\nabla\varphi(x_k)\|}{\|B_k\|}\}$$

The proof is complete.

In the next section, we will concentrate on the convergence of Algorithm 1.

3. CONVERGENCE ANALYSIS

The following lemma guarantees that Algorithm 1 does not cycle infinitely in the inner cycle.

Lemma 3.1. Let the Assumption A hold. Then Algorithm 1 is well defined, *i.e.*, Algorithm 1 does not cycle in the inner cycle infinitely.

Proof. First, we prove that the following relation holds when p is sufficiently large

$$\frac{\varphi_k - \varphi(x_{k+1})}{-q_k(d_k)} \geq \rho \tag{11}$$

Obviously, $\|\nabla\varphi(x_k)\| \geq \varepsilon$ holds, otherwise, Algorithm 1 stops. Hence

$$\Delta_k = \frac{c^p\|\nabla\varphi(x_k)\|}{\|B_k\|} \to 0,\ p \to \infty \tag{12}$$

By Lemma 2.1, we conclude that

$$-q_k(d_k) \geq \frac{1}{2}\|\nabla\varphi(x_k)\|\min\{\Delta_k, \frac{\|\nabla\varphi(x_k)\|}{\|B_k\|}\} \geq \frac{1}{2}\varepsilon\Delta_k,$$
$$\text{as } p \to \infty \tag{13}$$

Consider

$$|\varphi_k - \varphi(x_{k+1}) + q_k(d_k)| = O(\|d_k\|^2) \tag{14}$$

By **Eqs.12-14**, and $\|d_k\| \leq \Delta_k$, we get

$$\left|\frac{\varphi_k - \varphi(x_{k+1})}{-q_k(d_k)} - 1\right| = \left|\frac{\varphi_k - \varphi(x_{k+1}) + q_k(d_k)}{-q_k(d_k)}\right| \leq \frac{2O(\|d_k\|^2)}{\varepsilon\Delta_k} \to 0$$

Therefore, for p sufficiently large, which implies **Eq.11**.

The definition of the algorithm means that

$$r_k = \frac{\varphi_{l(k)} - \varphi(x_{k+1})}{-q_k(d_k)} \geq \frac{\varphi_k - \varphi(x_{k+1})}{-q_k(d_k)} \geq \rho.$$

This implies that Algorithm 1 does not cycle in the inner cycle infinitely. Then we complete the proof of this lemma.

Lemma 3.2. Let Assumption A hold and $\{x_k\}$ be generated by the Algorithm 1. Then we have $\{x_k\} \subset \Omega$.

Proof. We prove the result by induction. Assume that $\{x_k\} \subset \Omega$, for all $k \geq 0$. By using the definition of the algorithm, we have

$$r_{l(k)} \geq \rho > 0 \tag{15}$$

Then we get

$$\varphi_{l(k)} \geq \varphi_{k+1} - \rho q_k(d_k) > \varphi_{k+1} \tag{16}$$

By $l(k) \leq k$, $\varphi_{l(k)} \leq \varphi_0$, from **Eq.16**, we have

$$\varphi_{k+1} \leq \varphi_0,$$

this implies

$$\|g_{k+1}\| \leq \|g_0\|,$$

i.e.,

$$x_{k+1} \in \Omega$$

which completes the proof.

Lemma 3.3. Let Assumption A hold. Then $\{\varphi_{l(k)}\}$ is not increasing monotonically and is convergent.

Proof. By the definition of the algorithm, we get

$$\varphi_{l(k)} \geq \varphi_{k+1},\ \forall k \tag{17}$$

We proceed the proof in the following two cases.

1) $k \geq M$. In this case, from the definition of $\varphi_{l(k)}$ and **Eq.17**, it holds that

$$\begin{aligned}\varphi_{l(k+1)} &= \max_{0\leq j\leq n(k+1)}\{\varphi_{k+1-j}\} \\ &= \max\{\max_{0\leq j\leq n(k)-1}\{\varphi_{k-j}\}, \varphi_{k+1}\} \\ &\leq \varphi_{l(k)}\end{aligned} \tag{18}$$

2) $k < M$. In this case, using induction, we can prove that

$$\varphi_{l(k)} = \varphi_0$$

Therefore, the sequence $\{\varphi_{l(k)}\}$ is not increasing monotonically. By Assumption A(j) and Lemma 3.2, we know that $\{\varphi_k\}$ is bounded. Then $\{\varphi_{l(k)}\}$ is convergent.

In the following theorem, we establish the convergence of Algorithm 1.

Theorem 3.1. Let the conditions in Assumption A hold. If $\varepsilon = 0$, then the algorithm either stops finitely or generates an infinite sequence $\{x_k\}$ such that

$$\liminf_{k\to\infty}\varphi_k = 0 \tag{19}$$

Proof. We prove the theorem by contradiction. Assume that the theorem is not true. Then here exists a constant $\varepsilon_1 > 0$ satisfying

$$\varphi_k \geq \varepsilon_1,\ \forall k. \tag{20}$$

By Assumption A(jj) and the definition of B_k, there exists a constant $m > 0$ such that

$$\|B_k^{-1}\| \geq m \tag{21}$$

Therefore, according to Assumption A(j), Lemma 2.1, **Eq.20**, and **Eq.21**, there is a constant $b_1 > 0$ such that

$$-q_k(d_k) \geq b_1 c^{p_k} \tag{22}$$

where p_k is the value of p at which the algorithm gets out of the inner cycle at the point x_k. By step 2, step 3, step 4, and **Eq.22**, we know

$$\varphi_{l(k)} \geq \varphi_{k+1} + \rho b_1 c^{p_k} \tag{23}$$

Then

$$\varphi_{l(k+1)} \leq \varphi_{l(l(k))} - \rho b_1 c^{p_{l(k)}}. \tag{24}$$

By Lemma 3.3 and **Eq.24**, we deduce that

$$p_{l(k)} \to \infty \tag{25}$$

The definition of the algorithm implies that $d'_{l(k)}$ which corresponds to the following subproblem is unacceptable:

$$\min_{d\in R^n} \varphi_{l(k)}^T d + \frac{1}{2} d^T B_{l(k)} d = q_{l(k)}(d),$$

$$s.t. \ \| d \| \leq c^{p_{l(k)}-1} M_{l(k)} \varphi_{l(k)} = \frac{\Delta_{l(k)}}{c} \tag{26}$$

i.e.,

$$\frac{\varphi_{l(l(k))} - \varphi(x_{l(k)} + d'_{l(k)})}{-q_{l(k)}(d'_{l(k)})} < \rho \tag{27}$$

By the definition of $\varphi_{l(k)}$, we have

$$\frac{\varphi_{l(l(k))} - \varphi(x_{l(k)} + d'_{l(k)})}{-q_{l(k)}(d'_{l(k)})} \geq \frac{\varphi_{l(k)} - \varphi(x_{l(k)} + d'_{l(k)})}{-q_{l(k)}(d'_{l(k)})} \tag{28}$$

By step 2, step 3, and step 4, we have that when k is sufficiently large, the following formula holds:

$$\frac{\varphi_{l(k)} - \varphi(x_{l(k)} + d'_{l(k)})}{-q_{l(k)}(d'_{l(k)})} \geq \rho \tag{29}$$

This combines with **Eq.28** will contradicts **Eq.27**. The contradiction shows that the theorem is true. The proof is complete.

Remark. Theorem 3.1 shows that the iterative sequence $\{x_k\}$ generated by Algorithm 1 such that $\nabla g(x_k) g(x_k) \to 0$. If x^* is a cluster point of $\{x_k\}$ and $\nabla g(x^*)$ is nonsingular, then we have $\| g(x_k) \| \to 0$. This is a standard convergence result for nonlinear equations. At present, there is no method that can satisfy $\| g(x_k) \| \to 0$ without the assumption that $\nabla g(x^*)$ is nonsingular.

4. NUMERICAL RESULTS

In this section, results of some preliminary numerical experiments are reported to test our given method.

Problem. The discretized two-point boundary value problem is the same to the problem in [39]

$$g(x) \equiv Ax + \frac{1}{(n+1)^2} F(x) = 0$$

where A is the $n \times n$ tridiagonal matrix given by

$$A = \begin{bmatrix} 3 & -1 & & & & \\ -1 & 3 & -1 & & & \\ & -1 & 3 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & -1 \\ & & & & -1 & 3 \end{bmatrix}$$

and $F(x) = (F_1(x), F_2(x), \cdots F_n(x))^T$, with

$$F_i(x) = \sin x_i - 1, \ \text{i}=1,2,\cdots nS$$

In the experiments, the parameters were chosen as $c = 0.01$, $M = 10$, and $\rho = 0.8$, B_0 is the unit matrix. Solving the subproblem **Eq.4** to get d_k by *Dogleg* method. The program was coded in MATLAB 7.0. We stopped the iteration when the condition $\| g_k \| \leq 10^{-5}$ was satisfied. The columns of the tables have the following meaning:

Dim: the dimension of the problem.

NG: the number of the function evaluations.

NI: the total number of iterations.

GG: the norm of the function evaluations.

The numerical results (**Table 1**) indicate that the proposed method performs quite well for the Problem. Moreover, the inverse initial points and the initial points don't influence the performance of Algorithm 1 very much. Especially, the numerical results hardly change with the dimension increasing.

Discussion. In this paper, based on [23], a modified algorithm for solving symmetric nonlinear equations is presented. The convergent result is established and the numerical results are also reported. We hope that the proposed method can be a topic of further research for symmetric nonlinear equations.

Table 1. Test results for problem.

x_0	(2, … ,2)	(10, … ,10)	(50, … ,50)	(-10, … ,-10)	(-2, … ,-2)
Dim	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG	NI/NG/GG
n = 49	191/391/9.557342e-006	196/401/6.091920e-006	253/515/7.487518e-006	286/581/9.484488e-006	206/421/9.047968e-006
n = 100	240/505/9.607401e-006	402/829/9.985273e-006	117/259/8.296290e-006	185/395/9.828274e-006	144/313/9.842536e-006
n = 300	223/463/8.060658e-006	260/537/9.470041e-006	241/499/3.894953e-006	246/509/9.915900e-006	233/483/9.705042e-006
n = 500	157/331/9.236809e-006	171/359/9.814318e-006	177/371/9.567563e-006	170/357/9.852428e-006	155/327/7.401986e-006

REFERENCES

[1] Yuan, G.L. and Lu, X.W. (2009) A modified PRP conjugate gradient method. *Annals of Operations Research*, **166(1)**, 73-90.

[2] Yuan, G.L. and Lu, X.W. (2008) A new line search method with trust region for unconstrained optimization. *Communications on Applied Nonlinear Analysis*, **15(1)**, 35-49.

[3] Yuan, G.L. and Wei, Z.X. (2009) New line search methods for unconstrained optimization. *Journal of the Korean Statistical Society*, **38(1)**, 29-39.

[4] Yuan, G.L. and Wei, Z.X. (2008) Convergence analysis of a modified BFGS method on convex minimizations. *Computational Optimization and Applications*.

[5] Yuan, G.L. and Wei, Z.X. (2008) The superlinear convergence analysis of a nonmonotone BFGS algorithm on convex objective functions. *Acta Mathematica Sinica*, English Series, **24(1)**, 35-42.

[6] Yuan, G.L. and Lu, X.W. and Wei, Z.X. (2007) New two-point stepsize gradient methods for solving unconstrained optimization problems. *Natural Science Journal of Xiangtan University*, **29(1)**, 13-15.

[7] Yuan, G.L. and Wei, Z.X. (2004) A new BFGS trust region method. *Guangxi Science*, **11**, 195-196.

[8] Moré, J.J. and Sorensen, D.C. (1983) Computing a trust-region step. *SIAM Journal on Scientific and Statistical Computing*, **4(3)**, 553-572.

[9] Flecher, R. (1987) Practical methods of optimization. 2nd Edition, John and Sons, Chichester.

[10] Gay, D.M. (1981) Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, **2**, 186-197.

[11] Powell, M.J.D. (1975) Convergence properties of a class of minimization algorithms. Mangasarian, O.L., Meyer, R.R. and Robinson, S.M., Ed., *Nonlinear Programming*, Academic Press, New York, **2**, 1-27.

[12] Schultz, G.A., Schnabel, R.B. and Bryrd, R.H. (1985) A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM Journal on Numerical Analysis*, **22(1)**, 47-67.

[13] Byrd, R.H., Schnabel, R.B. and Schultz G.A. (1987) A trust-region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, **24(5)**, 1152-1170.

[14] Celis, M.R., Dennis, J.E. and Tapia, R.A. (1985) A trust-region strategy for nonlinear equality constrained optimization, in numerical optimization 1984. Boggs, P.R. Byrd, R.H. and Schnabel, R.B., Ed., SIAM, Philadelphia, 71-82.

[15] Liu, X. and Yuan, Y. (1997) A global convergent, locally superlinearly convergent algorithm for equality constrained optimization. Research Report, ICM-97-84, Chinese Academy of Sciences, Beijing.

[16] Vardi, A. (1985) A trust-region algorithm for equality constrained minimization: Convergence properties and implementation. *SIAM Journal of Numerical Analysis*, **22(3)**, 575-579.

[17] Nocedal, J. and Yuan, Y. (1998) Combining trust region and line search techniques. *Advances in Nonlinear Programming*, 153-175.

[18] Sterhaug, T. (1983) The conjugate gradient method and trust regions in large-scale optimization. *SIAM Journal Numerical Analysis*, **20(3)**, 626-637.

[19] Yuan, Y. (2000) A review of trust region algorithms for optimization. *Proceedings of the* 4*th International Congress on Industrial & Applied Mathematics* (*ICIAM* 99), Edinburgh, 271-282.

[20] Yuan, Y. (2000) On the truncated conjugate gradient method. *Mathematical Programming*, **87(3)**, 561-573.

[21] Zhang, X.S., Zhang, J.L. and Liao, L.Z. (2002) An adaptive trust region method and its convergence. *Science in China*, **45**, 620-631.

[22] Yuan, G.L., Meng, S.D. and Wei, Z.X. (2009) A trust-region-based BFGS method with line search technique for symmetric nonlinear equations. *Advances in Operations Research*, **2009**, 1-20.

[23] Zhang, J.L. and Zhang, X.S. (2003) A nonmonotone adaptive trust region method and its convergence. *Computers and Mathematics with Applications*, **45(10-11)**, 1469-1477.

[24] Schnabel, R.B. and Eskow, R. (1990) A new modified cholesky factorization. *SIAM Journal on Scientific and Statistical Computing*, **11(6)**, 1136-1158.

[25] Griewank, A. (1986) The 'global' convergence of Broyden-like methods with a suitable line search. *Journal of the Australian Mathematical Society Series B*, **28**, 75-92.

[26] Zhu, D.T. (2005) Nonmonotone backtracking inexact quasi-Newton algorithms for solving smooth nonlinear equations. *Applied Mathematics and Computation*, **161(3)**, 875-895.

[27] Fan, J.Y. (2003) A modified Levenberg-Marquardt algorithm for singular system of nonlinear equations. *Journal of Computational Mathematics*, **21**, 625-636.

[28] Yuan, Y. (1998) Trust region algorithm for nonlinear equations. *Information*, **1**, 7-21.

[29] Yuan, G.L., Wei, Z.X. and Lu, X.W. (2009) A nonmonotone trust region method for solving symmetric nonlinear equations. *Chinese Quarterly Journal of Mathematics*, **24**, 574-584.

[30] Yuan, G.L. and Lu, X.W. and Wei, Z.X. (2007) A modified trust region method with global convergence for symmetric nonlinear equations. *Mathematica Numerica Sinica*, **11(3)**, 225-234.

[31] Li, D. and Fukushima, M. (1999) A global and superlinear convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations. *SIAM Journal on Numerical Analysis*, **37(1)**, 152-172.

[32] Wei, Z.X., Yuan, G.L. and Lian, Z.G. (2004) An approximate Gauss-Newton-based BFGS method for solving symmetric nonlinear equations. *Guangxi Sciences*, **11(2)**, 91-99.

[33] Yuan, G.L. and Li, X.R. (2004) An approximate Gauss-Newton-based BFGS method with descent directions for solving symmetric nonlinear equations. *OR Transactions*, **8(4)**, 10-26.

[34] Yuan, G.L. and Li, X.R. (2010) A rank-one fitting method for solving symmetric nonlinear equations. *Journal of Applied Functional Analysis*, **5(4)**, 389-407.

[35] Yuan, G.L. and Lu, X.W. and Wei, Z.X. (2009) BFGS trust-region method for symmetric nonlinear equations.

Journal of Computational and Applied Mathematics, **230(1)**, 44-58.

[36] Yuan, G.L., Wei, Z.X. and Lu, X.W. (2006) A modified Gauss-Newton-based BFGS method for symmetric nonlinear equations. *Guangxi Science*, **13(4)**, 288-292.

[37] Yuan, G.L., Wang, Z.X. and Wei, Z.X. (2009) A rank-one fitting method with descent direction for solving symmetric nonlinear equations. *International Journal of Communications, Network and System Sciences*, **2(6)**, 555-561.

[38] Yuan, G.L. and Lu, X.W. (2008) A new backtracking inexact BFGS method for symmetric nonlinear equations. *Computer and Mathematics with Application*, **55(1)**, 116- 129.

[39] Ortega, J.M. and Rheinboldt, W.C. (1970) Iterative solution of nonlinear equations in several variables. Academic Press, New York.

