The Tenth International Symposium on Operations Research and Its Applications (ISORA 2011)
Dunhuang, China, August 28–31, 2011

A Non-monotone Adaptive Trust Region Algorithm For Nonlinear Equations

Hong-Wei Liu*

School of Information, Beijing Wuzi University, Beijing 101149, China

Abstract In this paper, a non-monotone adaptive trust region method for the system of non-linear equations is proposed, in part, which is based on the technique in [9]. The local and global convergence properties of non-monotone adaptive trust region method are proved under favorable conditions. Some numerical experiments show that the method is effective.

Keywords Trust region method; Global convergence; Nonlinear equations; Non-monotone line search technique

1 Introduction

We consider the nonlinear equation system

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

where $F : R^n \to R^m$ is a set of continuously differentiable functions. Throughout the paper, we assume that the solution set of (1) is nonempty and denoted by X^*. In all cases, $\|\cdot\|$ denotes the Euclidian norm of vectors or its induced matrix norm. Let $F'(x)$ denote the transpose of the Jacobian of $F(x)$, i.e., $F'(x) = (F_1'(x), F_2'(x), \cdots, F_m'(x))^T$.

A problem which is closely related to (1) is the following minimization problem

$$\min_{x \in R^n} \varphi(x) = \frac{1}{2}\|F(x)\|^2. \tag{2}$$

This problem is called the least square problem. Obviously, x^* solves (1) iff x^* solves (2) when X^* is nonempty.

(1) and (2) have many applications in engineering, such as nonlinear fitting, parameter estimating and function approximating. At present, a lot of algorithms have been proposed for solving these two problems, for examples, Gauss-Newton method, Levenberg-Marquardt method, trust region method, etc., see [1, 2, 3, 4, 5]. These algorithms are super-linearly convergent if $F'(x^*)$ is non-degenerate. Here we are interested in trust region method since it has strong convergence and robustness. For the traditional trust region methods, at each iterative point x_k (non-stationary point), the trial step is usually

*E-mail address: ryuhowell@163.com.

obtained by solving the following trust region subproblem

$$\begin{cases} \min\limits_{d\in R^n} & \phi_k(d)=\frac{1}{2}\|F(x_k)+F'(x_k)d\|^2 \\ s.t. & \|d\|\leq \triangle_k. \end{cases} \tag{3}$$

It is well known that the trust region methods are globally convergent under suitable conditions and super-linearly convergent under the condition that $F'(x^*)$ (x^* is a solution of (1.1)) is non-degenerate. For simplicity, we omit the subscript and drop the term $\frac{1}{2}\|F(x)\|^2$ in (3). Then (3) is equivalent to the following problem

$$\begin{cases} \min\limits_{d\in R^n} & g(x_k)^T d+\frac{1}{2}d^T H_k d \\ s.t. & \|d\|\leq \triangle_k, \end{cases} \tag{4}$$

where $g(x_k)$ denotes $F'(x_k)^T F(x_k)$, H_k denotes $F'(x_k)^T F'(x_k)$ and $\triangle_k$ is the trust radius. A merit function is normally used to test whether the trial step is accepted or the trust radius needs to be adjusted. Comparing with quasi-Newton methods, trust region methods converge to a point which not only is a stationary point, but also satisfies second-order necessary conditions. Because of its strong convergence and robustness, trust region methods have been studied by many authors[6, 7, 8, 5]. J.Zhang and X.Zhang [9] combine the upper trust region subproblem with non-monotone technique to present a non-monotone adaptive trust region method and study its convergence properties. Based on the technique of some of theirs, we propose a new non-monotone adaptive trust method for solving (1). For nonlinear equations, to the authors' knowledge, the global convergence is due to Griewank [10] for Broyden's rank one method.

In this paper, we solve (1) by the method of iteration and the main step at each iteration of the following method is finding the trial step d_k. Let x_k be the current iteration. The trial step d_k is a solution of the following problem

$$\begin{cases} \min\limits_{d\in R^n} & g(x_k)^T d+\frac{1}{2}d^T B_k d \\ s.t. & \|d\|\leq \triangle_k, \end{cases} \tag{5}$$

where B_k is a safely positive definite matrix based on Schnabel and Eskow [12] 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 H_k positive definite otherwise, and $\triangle_k = c^p\|g_k\|^\beta M_k$, $0<c<1$, $M_k = \|B_k^{-1}\|$, and p is a nonnegative integer.

The remainder of the paper is arranged as follows. In Section 2, the algorithm model is presented. In Section 3, the global convergence properties are studied. Numerical result in Section 4 indicate that the algorithm is very efficient. Finally some concluding remarks are addressed in Section 5.

2 Algorithm model

In this section, we give our algorithm for solving (1). Firstly, some definitions are given. At point x_k, let

$$\|F_{l(k)}\| = \max_{0\leq j\leq m(k)}\{\|F_{k-j}\|\}, \quad k=1,\ 2,\ \ldots,$$

where $m(k) = \min\{M,\ k\},\ M \geq 0$ is an integer constant and denote $\varphi_{l(k)} = \frac{1}{2}\|F_{l(k)}\|^2$. Then we define the actual reduction as

$$Ared_k = \varphi(x_k + d_k) - \varphi_{l(k)},$$

the predict reduction as

$$Pred_k = \phi_k(d_k) - \varphi(x_k) = g(x_k)^T d_k + \frac{1}{2} d_k^T B_k d_k,$$

where d_k is the solution of (5), and the ratio of actual reduction over predict reduction as

$$\gamma_k = \frac{Ared_k}{Pred_k}.$$

Now the algorithm is given as follows.

ALGORITHM MODEL

Initial: choose constants $\rho,\ \tau,\ c \in (0,1),\ p = 0,\ \varepsilon > 0,\ M \geq 0,\ x_0 \in R_n$. Let $k := 0$;

Step1: If $\|g_k\| < \varepsilon$, stop.

Step2: Compute d_k by solving (5) and calculate $m(k),\ F_{l(k)},\ pred_k$ *and* γ_k. If $\gamma_k < \rho$, then $p := p+1$, go to Step2. Otherwise, go to Step3.

Step3: $x_{k+1} = x_k + d_k$, generate B_{k+1}, set $p = 0, k := k+1$, go to Step1.

Remark 1. (i)B_k can be obtained by quasi-Newton iterate formula.

(ii)In this algorithm, the procedure of "Step2" is named as inner cycle.

(iii)If $M = 0$, this algorithm reduces to ......

3 Global convergence

In this section, we discuss the convergence properties of the algorithm. Before we address some theoretical issues, we would like to make the following assumptions.

ASSUMPTION 3.1.

(i)$F(x)$ is twice continuously differentiable;

(ii)$\{x_k\}$ is a bounded sequence.

By Assumption 3.1 there exists $M > 0$ such that

$$\|B_k\| \leq M, \quad \forall\ k.$$

Based on Assumption 3.1, we have the following lemma.

Lemma 1. *[11] Suppose that Assumption 3.1 holds. Then*

$$-Pred_k \geq \frac{c^p}{2\|B_k\|}\|g_k\|^2, \quad p = 0,\ 1,\ \ldots.$$

Lemma 2.

$$Pred_k \leq -\frac{1}{2}\|g_k\| \min\{\alpha_k, \|g_k\|/\|B_k\|\}.$$

Proof. By the definition of d_k, we know that for any $\alpha \in (0,1)$

$$\begin{aligned} Pred_k = & \ \phi_k(d_k) - \varphi(x_k) \\ \leq & \ \phi_k(-\alpha \frac{g_k}{\|g_k\|}) - \varphi(x_k) \\ = & \ -\alpha \|g_k\| + \frac{\alpha^2 g_k^T B_k g_k}{2\|g_k\|^2} \\ \leq & \ -\alpha \|g_k\| + \frac{1}{2}\alpha^2 M. \end{aligned}$$

Thus, together with Lemma 1, we have

$$\begin{aligned} Pred_k \leq & \ \min_{0 \leq \alpha \leq 1} \{-\alpha \|g_k\| + \frac{1}{2}\alpha^2 M\} \\ \leq & \ -\frac{1}{2}\|g_k\| \min\{\alpha_k, \|g_k\|/\|B_k\|\} \end{aligned}$$

□

The following lemma guarantees that the non-monotone adaptive trust region algorithm does not cycle infinitely in the inner cycle.

Lemma 3. *Suppose that Assumption 3.1 holds. The algorithm is well defined, i.e., the algorithm does not cycle in the inner cycle infinitely.*

Proof. First, we prove that when p is sufficiently large, it holds that

$$\frac{\varphi(x_k + d_k) - \varphi(x_k)}{Pred_k} \geq \rho. \tag{6}$$

Let d_k^i be the solution of (5) corresponding to $p = i$ at x_k and $\text{Pred}_{k(i)}$ be the predict reduction corresponding to $p = i$ at x_k. It follows from Lemma 1 that

$$-Pred_{k(i)} \geq \frac{c^i}{2\|B_k\|}\|g_k\|^2.$$

So,

$$|\frac{\varphi(x_k + d_k) - \varphi(x_k)}{Pred_k} - 1| \leq \frac{O(\|d_k^i\|^2)}{(c^i/2\|B_k\|)\|g_k\|^2} \leq \frac{O(\|\alpha_k\|^2)}{(c^i/2\|B_k\|)\|g_k\|^2} \to 0, \ i \to \infty,$$

which implies that (6) holds for p sufficiently large.

The definition of the algorithm implies that

$$\gamma_k = \frac{\varphi(x_k + d_k) - \varphi(x_{l(k)})}{Pred_k} \geq \frac{\varphi(x_k + d_k) - \varphi(x_k)}{Pred_k}.$$

Therefore, when p is sufficiently large, $\gamma_k \geq \rho$. This implies that the algorithm does not cycle in the inner cycle infinitely. □

Lemma 4. *Suppose that Assumption holds and $\{x_k\}$ is generated by the algorithm. Then we have $\{x_k\} \subset L(x_0)$.*

Proof. We prove the result by induction. The result evidently holds for $k = 0$. Assume that $x_k \in L(x_0)$, for $k \geq 0$. By the definition of the algorithm, we get

$$\gamma_{l(k)} \geq \rho > 0.$$

Then we get

$$\varphi_{l(k)} \geq \varphi(x_k + d_k) + \rho Pred_k \geq \varphi(x_k + d_k). \tag{7}$$

By $l(k) \leq k$, $\|F_{l(k)}\| \leq \|F_0\|$, then it follows from (7) that

$$\|F_{k+1}\| \leq \|F_0\|, \forall\, k.$$

i.e.,

$$x_{k+1} \in L(x_0),$$

which completes the proof. □

Lemma 5. *Suppose that Assumption holds. The $\{\|F_{l(k)}\|\}$ is not increasing monotonically and is convergent.*

Proof. From the definition of the algorithm, we have that

$$\|F_{l(k)}\| \geq \|F_{k+1}\|, \ \forall\, k. \tag{8}$$

Now we proceed the proof in the following two cases.

(i) $k \geq M$. In this case, from the definition of $f_{l(k)}$ and (8), it holds that

$$\|F_{l(k+1)}\| = \max_{0 \leq j \leq n(k+1)} \|F(x_{k+1-j})\| \leq \|F_{l(k)}\|.$$

(ii) $k < M$. In this case, by induction, we can prove that

$$\|F_{l(k)}\| = \|F_0\|.$$

So the sequence $\{\|F_{l(k)}\|\}$ is not increasing monotonically. From Assumption 3.1(i) and Lemma 4, we know that $\{\|F_k\|\}$ is bounded. Hence, $\{\|F_{l(k)}\|\}$ is convergence. □

Theorem 6. *Suppose that Assumption 3.1 holds. If $\varepsilon = 0$, then the algorithm either stops finitely or generates an infinite sequence $\{x_k\}$ such that*

$$\lim_{k \to \infty} \|g_k\| = 0.$$

Proof. We prove the theorem by contradiction. Assume that the theorem is not true. Then there exists a constant ε_0 such that

$$\|g(x_k)\| \geq \varepsilon_0, \ \forall\, k.$$

By Assumption 3.1 and the definition of B_k imply that there exists $M > 0$ such that

$$\|B_k^{-1}\| \geq M \tag{9}$$

Therefore, by Assumption 3.1, Lemma 1 and (9), there exists a constant $a > 0$ such that

$$pred_k \geq ac^{p_k}, \tag{10}$$

where p_k is the value of p at which the algorithm gets out of the inner cycle at the point x_k.

From Step2, Step3 and (10), we know that

$$\varphi_{l(k)} \geq \varphi(x_k + d_k) + \rho ac^{p_k}.$$

So

$$\varphi_{l(k+1)} \leq \varphi_{l(l(k))} - \rho ac^{p_{l(k)}}. \tag{11}$$

By Lemma 5 and (11), we deduce that

$$p_{l(k)} \to \infty.$$

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

$$\begin{cases} \min\limits_{d \in R^n} & pred_{l(k)} = g_{l(k)}^T d + \dfrac{1}{2} d^T B_{l(k)} d \\ s.t. & \|d\| \leq c^{p_{l(k)}-1} M_{l(k)} \|g_{l(k)}\| = \dfrac{\triangle_{l(k)}}{c} \end{cases}$$

i.e.,

$$\frac{\varphi(x_{l(k)} + d'_{l(k)}) - \varphi_{l(l(k))}}{pred_{l(k)}} < \rho. \tag{12}$$

It follows from the definition of $\|F_{l(k)}\|$ that

$$\frac{\varphi(x_{l(k)} + d'_{l(k)}) - \varphi_{l(l(k))}}{pred_{l(k)}} \geq \frac{\varphi(x_{l(k)} + d'_{l(k)}) - \varphi_{l(k)}}{pred_{l(k)}}.$$

we have that when k is sufficiently large, the following formula holds:

$$\frac{\varphi(x_{l(k)} + d'_{l(k)}) - \varphi_{l(k)}}{pred_{l(k)}} > \rho.$$

This contradicts (12). The contradiction shows that the theorem is true. □

Remark 2. Theorem 6 says that the iterative sequence $\{x_k\}$ generated by our algorithm satisfies $\|g(x_k)\| \to 0$. If x^* is a cluster point of $\{x_k\}$ and $F'(x^*)$ is non-degenerate, then we have $\|F(x_k)\| \to 0$. This is a standard convergence result for nonlinear equations. At present, there is no algorithm which has the property that the iterative sequence generated by the algorithm satisfies $\|F(x_k)\| \to 0$ without the assumption that $F'(x^*)$ is non-degenerate.

Table 1: Numerical results of some test problems.

Numerical experiments		Trad				NMAdapt			
Prob	Dim	I	F	G	$\frac{1}{2}\|F(x^*)\|^2$	I	F	G	$\frac{1}{2}\|F(x^*)\|^2$
Rosenbrock	2	23	34	27	0.44-16	15	28	21	0.22-17
Powell singular	4	11	19	16	0.31-7	10	17	14	0.65-7
Powell badly scaled	2	104	163	123	0.46-9	163	213	186	0.71-8
Wood	4	45	76	59	0.74-9	31	59	44	0.47-12
Helical valley	3	19	30	23	0.65-14	12	21	17	0.44-14
Waston	12	126	179	143	0.23-6	62	117	87	0.26-7
Brown almost linear	30	10	17	14	0.73-13	24	36	29	0.15-13
Discrete boundary value	10	12	21	15	0.59-8	16	29	22	0.63-10
Discrete integral equation	10	5	9	7	0.98-16	5	8	6	0.98-16
Trigonometric	30	149	243	183	0.89-5	85	142	113	0.93-5
Variably dimensioned	10	121	174	152	0.17-10	98	166	129	0.26-9
Broyden tridiagonal	10	73	125	102	0.11-10	73	125	102	0.68-16
Broyden banded	30	21	32	25	0.15-9	14	26	19	0.11-13

4 Numerical Experiments and Conclusions

In this section, in order to see the efficiency of our method, numerical results are reported on some classical problems. The algorithm is implemented in Fortran 90, and in Compaq Visual Fortran 6.5 environment in PC. The test problems are created by modifying the problems given in [13] and have the same form as in [14]. The subroutine solving trust region subproblem was provided by Jorge J. Moré. x_0 is suggested by Moré, Garbow and Hillstrom in [13]. The stopping criterion used is $\|g_k\| < \varepsilon$, where $\varepsilon = 10^{-8}$. For comparison, the quadratic subproblems are solved precisely and all of the algorithms use the same subroutine to solve the quadratic subproblems. The traditional trust region method used here is the method described in [8] and B_k is obtained by the BFGS update. The radius of the trust region in [8] is determined as follows:

$$\triangle_{k+1} = \begin{cases} \dfrac{c_3\|s_k\| + c_4\triangle_k}{2}, & if \;\; r < c_2, \\[2ex] \dfrac{(1+c_1)\triangle_k}{2}, & otherwise. \end{cases}$$

where $\eta = 0.1, c_1 = 2, c_2 = 0.25, c_3 = 0.25$ and $c_4 = 0.5$. For the non-monotone adaptive trust region method, B_k is also obtained by the BFGS update. In the computation, we chose $\eta = 0.1, c = 0.5, \beta = 0.6$. However, we found that the choice of c has little impact on the computational efficiency. The detailed results are summarized in the following Table 1. The columns of the table have the following meaning:

Prob. : the name of the test problem in Fortran;

Dim. : the dimension of the problem;

Trad. : the traditional trust region method in [8];

NMAdapt. : the non-monotone adaptive trust region method;

In columns 3-7, I, F and G represent the numbers of iteration, function evaluations and gradient evaluations.

From numerical result of the table 1., we easily know that the proposed method is robust in most occasion. However, in some case (such as Powell badly scaled, Brown almost linear, Discrete boundary value,etc), the classical method is more efficient. The result of the table show that the proposed method is rather efficient.

In this paper, Based on [9], we give a modified algorithm for solving nonlinear equations. Theoretical analysis shows that the method possesses global convergence and the numerical results show that the method is very efficient.

Acknowledgements

This work is partly supported by Chinese National Science Foundations (grant No. 10701080), Beijing Natural Science Foundations (grant No. 1092011), Foundation of Beijing Education Commission (grant No. SM200910037005), Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (grant No. PHR200906210), Funding Project for Base Construction of Scientific Research of Beijing Municipal Commission of Education (grant No. WYJD200902).

References

[1] Fletcher R (1987) Practical Methods of Optimizations. New York: John Wiley & Sons

[2] Levenberg K (1944) A method for the solution of certain nonlinear problem in least squares. Quart. Appl. Math. 2: 164-166

[3] Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear inequalities. SIAM J. Appl. Math. 11: 431-441

[4] Nocedal J, Wright SJ (1999) Numerical Optimization. New York: Springer

[5] Yuan Y, Sun W (1997) Optimization Theory and Algorithm. Scienti.c Publisher House, China

[6] R.Fletcher, Practical Method of Optimization, Unconstrained Optimization,Volume 1, John Wiley, New York, 1980

[7] Moré JJ, Recent developments in algorithms and software for trust region methods, In Mathematical Programming: The State of the Art,(Edited by A.Bachem, M.Grötchel and B.Korte) Springer-Verlag, Berlin, 1983, 258-287

[8] Yuan Y, On the convergence of trust region algorithms, *Mathematica Numerica Sinica*, 16(1996), 333-346

[9] J.L.Zhang and X.S.Zhang (2003) A Nonmonotone Adaptice Trust Region method and its convergence. Computers and Mathematics with Applications 45: 1469-1477

[10] Griewank.A. The global convergence of Broyden-like methods with a suitable line search. *J. Austral. Math. Soc. Ser.B*, 28(1986),75-92

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

[12] R.B.Schnabel and E.Eskow, A new modified Cholesky factorization, *SIAM J. Sci. Stat. Comput.*, 11(1990), 1136-1158

[13] Mor'e JJ, Garbow BS, Hillstrom KH (1981) Testing unconstrained optimization software. ACM Transaction on Mathematical Software 7: 17-41

[14] R.B.Schnabel and P.D.Frank,Tensor method for nonlinear equation. *SIAM J. Numerical Analysis*, 21, 815-843