Scientific Research Publishing

**Journal of Applied Mathematics and Physics, 2021, 9, 3102-3114**
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

# A New Nonmonotone Adaptive Trust Region Method

**Yang Zhang**[1], **Quanming Ji**[1], **Qinghua Zhou**[1,2*]

[1]College of Mathematics & Information Science, Hebei University, Baoding, China
[2]School of Applied Mathematics, Beijing Normal University, Zhuhai, China
Email: zybaby97@163.com, jqm19970917@163.com, *qinghua.zhou@gmail.com

**How to cite this paper:** Zhang, Y., Ji, Q.M. and Zhou, Q.H. (2021) A New Nonmonotone Adaptive Trust Region Method. *Journal of Applied Mathematics and Physics*, **9**, 3102-3114.
https://doi.org/10.4236/jamp.2021.912202

**Received:** October 30, 2021
**Accepted:** December 20, 2021
**Published:** December 23, 2021

## Abstract

The trust region method plays an important role in solving optimization problems. In this paper, we propose a new nonmonotone adaptive trust region method for solving unconstrained optimization problems. Actually, we combine a popular nonmonotone technique with an adaptive trust region algorithm. The new ratio to adjusting the next trust region radius is different from the ratio in the traditional trust region methods. Under some appropriate conditions, we show that the new algorithm has good global convergence and superlinear convergence.

## Keywords

Unconstrained Optimization, Trust Region Method, Nonmonotone Technique, Global Convergence, Superlinear Convergence

## 1. Introduction

In this paper, we consider the following unconstrained optimization problem:

$$\min_{x \in R^n} f(x), \tag{1.1}$$

where $f(x): R^n \to R$ is a real-valued twice continuously differentiable function.

The trust region method is an effective iterative method for solving problem (1.1). In 1944, Levenberg [1] firstly proposed a trust region method, where a modified Gauss-Newton method is given for solving nonlinear least square problems. Pioneer researches on trust region methods were given by Powell [2] [3] [4] [5], Fletcher [6], Hebden [7], Madsen [8], Osborne [9], Moré [10], Toint [11] [12] [13] [14] [15], Dennis and Mei [16], Sorensen [17] [18], and Steihaug [19]. Moré [20] gave good research of early works on trust region methods, which promoted trust region methods and standardized the term "trust region". The modern version of trust region methods can be traced back to Powell [4]. Later, Conn [21] applied the modern trust region methods to solve ill-conditioned problems and proved that it has strong global convergence. In order to minimize $f(x)$, the trust region method constructs trust region subproblem to compute a trial step $d_k$ through the following quadratic approximation:

$$\begin{aligned} &\min_{d\in R^n} \ q_k(d) := f_k + g_k^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} B_k d \\ &\text{s.t.} \quad \|d\| \le \Delta_k \end{aligned} \tag{1.2}$$

where $f_k = f(x_k)$, $g_k = g(x_k) = \nabla f(x_k)$ is the gradient of the objective function at the current iteration point, the approximation of the Hessian matrix $B_k = \nabla^2 f(x_k) \in R^{n\times n}$ is an $n\times n$ symmetric matrix, $\|\cdot\|$ denotes the Euclidean norm and $\Delta_k$ is a positive parameter which is called trust region radius.

At each iteration, the strategy of choosing a trust region radius $\Delta_k$ is very crucial. In the standard trust region method, the following ratio to make a comparison between the objective function and the model is defined:

$$r_k = \frac{Ared_k}{Pred_k} = \frac{f_k - f(x_k + d_k)}{q_k(0) - q_k(d_k)}. \tag{1.3}$$

In the case, if the ratio $r_k$ is close to 1, it is concluded that there is a good agreement between the objective function and the model over this step, so it is safe to increase the trust region radius for the next iteration. Otherwise, if $r_k$ is close to 0 or even negative, we must shrink the trust region radius.

The strategies of determining and updating trust region radius affect the number of computational cost and convergence of the algorithm. There are many researchers who pay much attention to determining and updating the trust region radius [22]-[27]. In 1997, Sartenaer [22] proposed a new approach to determine a radius by monitoring an agreement between the model and the objective function along the direction $-g_k$ computed at the starting point. But the parameters of this procedure may be dependent on the problem that should be solved. In 2005, Gould *et al.* [28] examine the sensitivity of trust-region algorithms on the parameters related to the step acceptance and update of the trust region, although, they did not discuss an initial trust-region radius. Motivated by a problem in neural network, in 2002, Zhang *et al.* [26] proposed a strategy to determine the trust region radius. Specifically, they solved the subproblem (1.2) with

$$\Delta_k = c^p \|g_k\| \left\|\hat{B}_k^{-1}\right\|,$$

where $c \in (0,1)$, $p$ is a nonnegative integer and $\hat{B}_k = B_k + iE$ is a positive definite matrix based on a modified Cholesky factorization from Schnabel and Eskow [29]. It needs to estimate $\left\|\hat{B}_k^{-1}\right\|$ at each iteration which can not use this radius for large-scale optimization problems. Inspired by Zhang's method, Cui and Wu [30] proposed a new adaptive trust region method, which can automatically update the trust region radius during calculation. The adaptive radius is given by

$$\Delta_k = \left\|d_k\right\|^2 \Big/ d_k^{\mathrm{T}} B_{k+1} d_k \ ,$$

avoiding the repeated solution of $\left\|\hat{B}_k^{-1}\right\|$, and the trust region is no longer dependent on the current iteration information $g_k$. In 2004, Shi and Wang [27] proposed an adaptive radius given by

$$\Delta_k = c^p \left\|g_k\right\|^3 \Big/ g_k^{\mathrm{T}} \hat{B}_k g_k \ ,$$

where $c \in (0,1)$, $p$ is a non-negative integer and $\hat{B}_k = B_k + iE$ is a positive definite matrix. More recently, Shi and Guo [31] proposed an adaptive trust region:

$$\Delta_k = -c^p \frac{g_k^{\mathrm{T}} q_k}{q_k^{\mathrm{T}} \hat{B}_k q_k} \left\|q_k\right\|, \tag{1.4}$$

the vector $q_k$ satisfies the angle consdition:

$$-\frac{g_k^{\mathrm{T}} q_k}{\left\|g_k\right\| \cdot \left\|q_k\right\|} \geq \theta . \tag{1.5}$$

where $\theta \in (0,1)$. Theoretical analysis shows that the proposed trust region method has global convergence for first-order critical points, and preliminary numerical results show that the proposed method is effective for solving medium-scale unconstrained optimization problems. Kamandi *et al.* [32] give an improved version of the trust-region radius (1.4). They proposed a modification of $q_k$:

$$q_k = \begin{cases} -g_k, & \text{if } k = 0 \text{ or } \dfrac{-\left(g_k^{\mathrm{T}} d_{k-1}\right)}{\left\|g_k\right\| \left\|d_{k-1}\right\|} \leq \theta, \\ d_{k-1}, & \text{otherwise,} \end{cases} \tag{1.6}$$

where $d_{k-1}$ is a solution of the subproblem (1.2) which can be accessed and $\theta \in (0,1)$. It is straightforward that $q_k$ satisfies the condition (1.5). To avoid a very small trust region radius, the formula is defined:

$$s_k = \begin{cases} -\dfrac{g_k^{\mathrm{T}} q_k}{q_k^{\mathrm{T}} B_k q_k} \left\|q_k\right\|, & \text{if } k = 0, \\ \max\left\{ -\dfrac{g_k^{\mathrm{T}} q_k}{q_k^{\mathrm{T}} B_k q_k} \left\|q_k\right\|, \lambda \Delta_{k-1} \right\}, & \text{if } k \geq 1. \end{cases} \tag{1.7}$$

where $\lambda > 1$, and $q_k$ is determined by (1.5). Then, the trust region radius is updated by

$$\Delta_k = h^p \min\left\{s_k, \tilde{\Delta}\right\}, \tag{1.8}$$

where $\tilde{\Delta} > 0$ is a real-valued constant, $h \in (0,1)$ and $p$ is a nonnegative integer.

Due to the high efficiency of nonmonotone techniques, many researchers use the nonmonotone technique in the trust region algorithm framework. In 1986, Grippo *et al.* [33] put forward the nonmonotone line search technology for the first time. The stepsize $\alpha_k$ satisfies condition

$$f\left(x_k + \alpha_k d_k\right) \leq f_{l(k)} + \beta \alpha_k g_k^{\mathrm{T}} d_k , \tag{1.9}$$

where $\beta \in \left(0, \frac{1}{2}\right)$, the nonmonotone term $f_{l(k)}$ is defined by

$$f_{l(k)} \le \max_{0 \le j \le m(k)} \left\{ f\left(x - j\right) \right\}, \tag{1.10}$$

where $m(0) = 0$, $0 \le m(k) \le \min\left\{ m(k-1)+1, \tilde{M} \right\}$, for $k \ge 1$, and $\tilde{M}$ is a given non-negative integer. This technique leads to a breakthrough in nonmonotonic algorithms for nonlinear optimization. Based on the proposed nonmonotone technique by Grippo et al, Ke and Han [34], Sun [35], Fu and Sun [36] presented various nonmonotone trust region methods. In 2004, Zhang and Hager [37] found that nonmonotone techniques (1.10) have some drawbacks. For example, the numerical performances are seriously dependent on the choice of parameter $\tilde{M}$; A good function value generated at any iteration may not be useful; For any given bound $\tilde{M}$ on the memory, even an iterative method is generating R-linearly convergence for a strongly convex function, the iterates may not satisfy the condition (1.3) for $k$ sufficiently large [38]. In order to cope with these disadvantages, Zhang and Hager [37] propose another nonmonotonic technique where the stepsize $\alpha_k$ satisfies the following condition:

$$f\left(x_k + \alpha_k d_k\right) \le C_k + \beta \alpha_k g_k^{\mathrm{T}} d_k,$$

where

$$C_k = \begin{cases} f\left(x_k\right), & k \ge 0 \\ \dfrac{\eta_{k-1} Q_{k-1} C_{k-1} + f\left(x_k\right)}{Q_k}, & k \ge 1 \end{cases} \tag{2.1}$$

$$Q_k = \begin{cases} 1, & k \ge 0 \\ \eta_{k-1} Q_{k-1} + 1, & k \ge 1 \end{cases} \tag{2.2}$$

where $\eta_{k-1} \in \left[\eta_{\min}, \eta_{\max}\right]$; $\eta_{\min} \in [0,1)$ and $\eta_{\max} \in \left[\eta_{\min}, 1\right)$ are two prefixed constants. Inspired by this nonmonotone technique, in 2006, Mo *et al.* [39] introduced it into trust region method and developed a nonmonotone algorithm. The numerical results indicate that the algorithm is robust and encouraging. In 2019, Xue *et al.* [40] propose a new improved nonmonotone adaptive trust region method for solving unconstrained optimization problems. From the perspective of theoretical analysis, it is shown that algorithm possesses global convergence and superlinear convergence under classical assumptions.

Among the existing nonmonotone strategies, Gu and Mo [41] propose a simpler nonmonotone technique, and its computational complexity is greatly reduced. Therefore, based on the method in [40] and [41], we propose a new improved nonmonotone adaptive trust region method for solving unconstrained optimization problems. Under appropriate conditions, we analyze the global convergence and superlinear convergence of the algorithm.

## 2. The Structure of the New Algorithm

In this section, we talk about our algorithm for solving unconstrained optimization problems in detail. As we see, The nonmonotone technique proposed by

Zhang and Hager [37] implies that each $C_k$ is a convex combination of the previous $C_{k-1}$ and $f_k$, including the complex $\eta_k$ and $Q_k$. In practice, it becomes an encumbrance to update $\eta_k$ and $Q_k$ at each iteration. Therefore, Gu and Mo [41] proposed another nonmonotone technique where the nonmonotone term is revised by:

$$D_k = \begin{cases} f_k, & k = 0, \\ \eta_k D_{k-1} + (1-\eta_k) f_k, & k \geq 1, \end{cases} \tag{2.1}$$

where $\eta_k \in [\eta_{\min}, \eta_{\max}]$; $\eta_{\min} \in (0,1)$ and $\eta_{\max} \in [\eta_{\min}, 1)$ are two prefixed constants.

Then, the actual reduction of the objective function value is given by:

$$Ared_k = D_k - f(x_k + d_k), \tag{2.2}$$

the predicted reduction of the objective function value is given by:

$$Pred_k = q_k(0) - q_k(d_k). \tag{2.3}$$

In order to determine whether the trial step is feasible and how to update the new trust region radius, we compute the modified ratio that is given by:

$$r_k = \frac{Ared_k}{Pred_k} = \frac{D_k - f(x_k + d_k)}{q_k(0) - q_k(d_k)}. \tag{2.4}$$

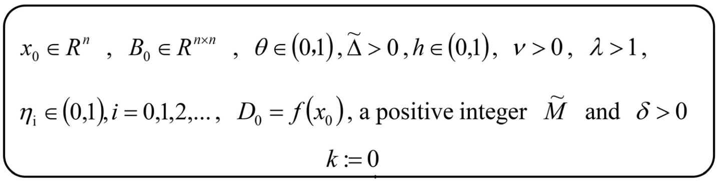
We describe the new trust region region algorithm with adjustable radius as below:

**Algorithm 2.1.** ANNATR (A new nonmonotone adaptive trust region method)

```
Step 0. Let $x_0 \in R^n$, a positive definite matrix $B_0 \in R^{n \times n}$, $\theta \in (0,1)$, $\tilde{\Delta} > 0$, $h \in (0,1)$,
        $v > 0$, $\lambda > 1$, $\eta_i \in (0,1)$, $i = 0,1,2,\cdots$, $D_0 = f(x_0)$, a positive integer $\tilde{M}$ and
        $\delta > 0$. Set $k := 0$.
Step 1. If $g_k \leq \delta$, then stop.
Step 2. Compute $q_k$ according to expression (1.6), $s_k$ by (1.7) and set $p = 0$.
Step 3. Compute $\Delta_k$ by (1.8), solve subproblem (1.2) to find the trial step $d_k$ and
    compute $r_k$ by (2.4).
Step 4. If $r_k < \upsilon$, then $p = p + 1$. Set $x_{k+1} = x_k$, $k := k + 1$. Go to Step 3.
Step 5. Set $x_{k+1} = x_k + d_k$.
Step 6. Update $B_k$ by BFGS update formula [42].
    Set $k := k + 1$ and go to Step 1.
```

In Algorithm 2.1, if $r_k \geq v$, it is called a successful iteration. The loop started from Step 3 to Step 4 is called the inner cycle.

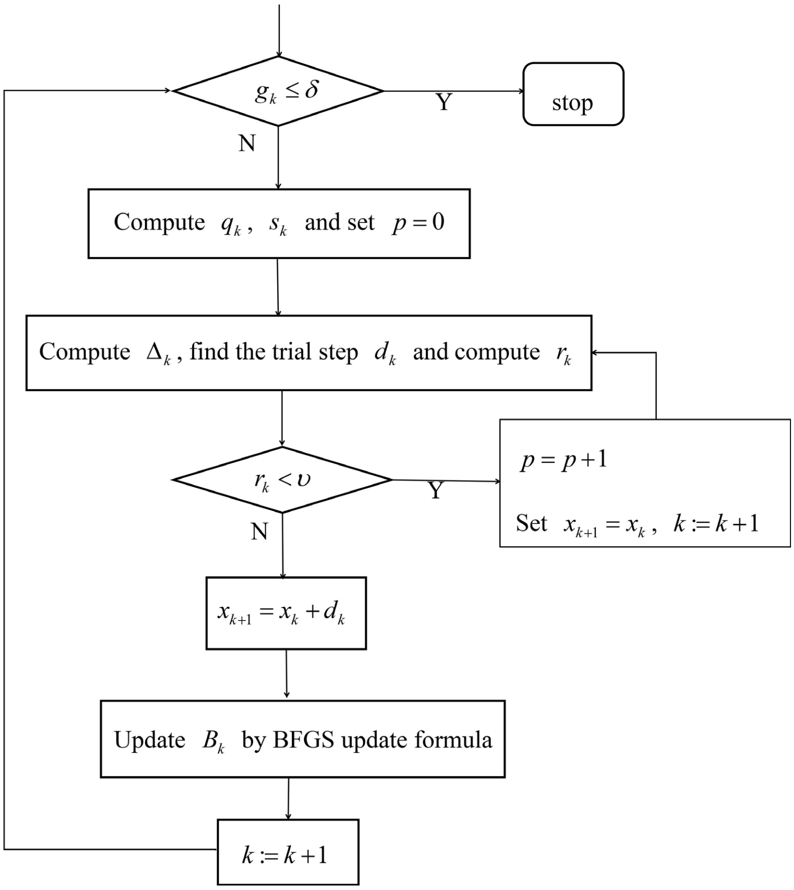
The flowchart of our algorithm is provided here:



$x_0 \in R^n$, $B_0 \in R^{n \times n}$, $\theta \in (0,1), \tilde{\Delta} > 0, h \in (0,1)$, $v > 0$, $\lambda > 1$,

$\eta_i \in (0,1), i = 0,1,2,\ldots$, $D_0 = f(x_0)$, a positive integer $\tilde{M}$ and $\delta > 0$

$k := 0$


$g_k \le \delta$
Y
stop
N
Compute $q_k$, $s_k$ and set $p=0$
Compute $\Delta_k$, find the trial step $d_k$ and compute $r_k$
$r_k < \upsilon$
Y
$p=p+1$
Set $x_{k+1}=x_k$, $k:=k+1$
N
$x_{k+1}=x_k+d_k$
Update $B_k$ by BFGS update formula
$k:=k+1$

## 3. Convergence Analysis

In this paper, we consider the following assumptions that will be used to analyze the convergence properties and the superlinear convergence rate of the below new algorithm:

(H1) The level set $W_0 = \left\{ x \in R^n : f(x) \le f(x_0) \right\} \subset \Omega$, where $\Omega \in R^n$ is a closed and bounded set and the objective function *f* is a twice continuously differentiable over $W_0$;

(H2) The approximation matrix $B_k$ is uniformly bounded, *i.e.* there exists a positive constant *M* such that $\|B_k\| \le M$ for all $k \in N$;

(H3) Matrix $B_k$ is invertible and the step $d_k = -B_k^{-1} g_k$ computed from Algorithm 2.1 for all $k \in N$.

The subsequent results are essential results to establish the global convergence of Algorithm 2.1.

**Lemma 3.1.** Suppose that the sequence $\{x_k\}$ is generated by Algorithm 2.1. Then we have

$$\left| f_k - f(x_k + d_k) - Pred_k \right| \le O\left(\|d_k\|^2\right). \tag{3.1}$$

Proof. See [43] for reference. □

**Lemma 3.2.** If (H2) holds, the sequence $\{x_k\}$ is generated by Algorithm 2.1, and $d_k$ is a solution of (1.2) with $\Delta_k$ given by (1.7), then we get

$$q_k(0)-q_k(d_k)\geq\frac{1}{2}h^{p_k}\min\left\{\frac{1}{M}\left(\frac{-g_k^{\mathrm{T}}q_k}{\|q_k\|}\right)^2,\tilde{\Delta}\left(\frac{-g_k^{\mathrm{T}}q_k}{\|q_k\|}\right)\right\},\tag{3.2}$$

for all $k\in N$.

Proof. See [32] for reference. □

**Lemma 3.3.** Let $\{x_k\}$ be the sequence generated by Algorithm 2.1. Then we have

$$f_{k+1}\leq D_{k+1}\leq D_k,\quad\forall k\in N.\tag{3.3}$$

Proof. From the definition of $D_k$, we have

$$D_{k+1}-D_k=(1-\eta_{k+1})(f_{k+1}-D_k)\tag{3.4}$$

and

$$D_{k+1}-f_{k+1}=\eta_{k+1}(D_k-f_{k+1}),\tag{3.5}$$

Now we Let $I=\{k|r_k\geq\nu\}$ and $J=\{k|r_k<\nu\}$. We consider two cases:

Case 1. $k\in I$. From (2.4) and (3.2), we have

$$D_k-f_{k+1}\geq\nu\left[q_k(0)-q_k(d_k)\right]\geq\frac{1}{2}\nu h^{p_k}\min\left\{\frac{1}{M}\left(\frac{-g_k^{\mathrm{T}}q_k}{\|q_k\|}\right)^2,\tilde{\Delta}\left(\frac{-g_k^{\mathrm{T}}q_k}{\|q_k\|}\right)\right\}\geq0.\tag{3.6}$$

From (3.4), (3.5) and (3.6), we have

$$f_{k+1}\leq D_{k+1}\leq D_k,\quad\forall k\in I.\tag{3.7}$$

Case 2. $k\in J$. If $k-1\in I$, from Case 1, we have $f_k\leq D_k$, and from Step 4 of Algorithm 2.1 we get $x_{k+1}=x_k$, $f_{k+1}=f_k$, $\forall k\in J$. Then we get

$$D_{k+1}=\eta_{k+1}D_k+(1-\eta_{k+1})f_{k+1}\geq\eta_{k+1}f_{k+1}+(1-\eta_{k+1})f_{k+1}=f_{k+1}.\tag{3.8}$$

From (3.4), (3.5) and (3.8), we have

$$f_{k+1}\leq D_{k+1}\leq D_k,\quad\forall k-1\in I.\tag{3.9}$$

If $k-1\in J$, let $K=\{i|1<i<k,k-i\in I\}$. If $K=\varnothing$, then from Step 4 of Algorithm 2.1, we have $f_0=f_{k-j}=f_{k+1}$, $j=0,1,\cdots,k-1$. Therefore, from the definition of $D_k$, $D_{k+1}=D_k=f_{k+1}$. On the other hand, if $K\neq\varnothing$, let $m=\min\{i|i\in K\}$, then we have $f_{k-j}=f_k=f_{k+1}$, $j=0,1,\cdots,m-1$. Obviously, $k-m\in I$, then we get $f_{k-m+1}\leq D_{k-m+1}\leq D_{k-m}$ from Case 1. Then we have

$$\begin{aligned}D_{k-m+2}&=\eta_{k-m+2}D_{k-m+1}+(1-\eta_{k-m+2})f_{k-m+2}\\&\geq\eta_{k-m+2}f_{k-m+2}+(1-\eta_{k-m+2})f_{k-m+2}=f_{k-m+2}\end{aligned}$$

Then from (3.4) and (3.5), we get $D_{k-m+2}\leq D_{k-m+1}$. Therefore, from the definition of $D_k$, we have $D_{k+1}\leq D_k$. From (3.4) and (3.5), we get $f_{k+1}\leq D_{k+1}\leq D_k$.

Both Case 1 and Case 2 imply that $f_{k+1}\leq D_{k+1}\leq D_k$, $\forall k\in N$. So the proof is completed. □

**Lemma 3.4.** Step 3 and Step 4 of Algorithm 2.1 are well-defined in the sense that at each iteration they terminate finitely.

Proof. The proof this lemma is quite as the same as [40], for the completeness of this work, we prove it again here. By contradiction, assume that the inner loop from Step 3 to Step 4 of algorithm 2.1 is infinite. Now, let $d_k^i$ be the solution of subproblem (1.2) corresponding to $i \in N$ at $x_k$. Then we have

$$r_k^i < \nu, \quad i = 1, 2, \cdots \tag{3.10}$$

Since $x_k$ is not optimum, then we have

$$\|g_k\| \ge \delta, \quad k \in N, \tag{3.11}$$

using (3.11) and (1.5), we get

$$-\frac{g_k q_k}{\|q_k\|} \ge \theta\delta. \tag{3.12}$$

It follows from Lemma 3.1-3.2 and (3.12) that

$$\begin{aligned}\left|\frac{f(x_k)-f(x_k+d_k^i)}{q_k(0)-q_k(d_k^i)}-1\right| &= \left|\frac{f(x_k)-f(x_k+d_k^i)-Pred_k^i}{q_k(0)-q_k(d_k^i)}\right| \\ &\le \frac{O\left(\|d_k^i\|^2\right)}{q_k(0)-q_k(d_k^i)} \\ &\le \frac{O\left(\|d_k^i\|^2\right)}{\frac{1}{2}h^{p_{k_i}}\min\left\{\frac{1}{M}\left(\frac{-g_k^{\mathrm{T}}q_k}{\|q_k\|}\right)^2, \tilde{\Delta}\left(\frac{-g_k^{\mathrm{T}}q_k}{\|q_k\|}\right)\right\}} \\ &\le \frac{O\left(\|d_k^i\|^2\right)}{\frac{1}{2}h^{p_{k_i}}\min\left\{\frac{(\theta\delta)^2}{M}, \tilde{\Delta}(\theta\delta)\right\}}\end{aligned} \tag{3.13}$$

By the assumption that the inner cycle cycles infinity and (1.8), we obtain that $\Delta_k^i \to 0$ with $i \to \infty$. $\|d_k^i\| \le \Delta_k^i \le h_k^{i\,p} s_k$ implies that the right-hand side of the above equation (3.13) tends to zero. This means that for sufficiently large *i*, we get

$$\lim_{i\to\infty}\frac{f(x_k)-f(x_k+d_k^i)}{q_k(0)-q_k(d_k^i)} = 1, \tag{3.14}$$

combining (2.4) and Lemma 3.3, we get

$$r_k^i = \frac{D_k - f(x_k+d_k^i)}{q_k(0)-q_k(d_k^i)} \ge \frac{f(x_k)-f(x_k+d_k^i)}{q_k(0)-q_k(d_k^i)}, \tag{3.15}$$

this means for $i \to \infty$, $r_k^i \ge \nu \in (0,1)$ which is contradictory to (3.10), so the proof is completed. □

**Theorem 3.1.** If (H1) holds and the sequence $\{x_k\}$ is generated by Algorithm 2.1, then we have

$$\liminf_{k\to\infty}\|g_k\| = 0. \tag{3.16}$$

Proof. By contradiction, suppose that there exists a constant $\delta > 0$ such that

$$\|g_k\| \ge \delta, \quad k \in N. \tag{3.17}$$

Using (2.4), Lemma 3.2 and $r_k \ge \nu$, we conclude that

$$f\left(x_k+d_k\right)\le D_k-\nu Pred_k\le D_k-\frac{1}{2}\nu h^{p_k}\min\left\{\frac{1}{M}\left(\frac{-g_k^{\mathrm{T}}q_k}{\left\|q_k\right\|}\right)^2,\tilde{\Delta}\left(\frac{-g_k^{\mathrm{T}}q_k}{\left\|q_k\right\|}\right)\right\}. \tag{3.18}$$

From the definitions of $D_k$ and (3.18), we can get

$$\begin{aligned} D_{k+1}&=\eta_{k+1}D_k+\left(1-\eta_{k+1}\right)f_{k+1}\\ &\le\eta_{k+1}D_k+\left(1-\eta_{k+1}\right)\left[D_k-\frac{1}{2}\nu h^{p_k}\min\left\{\frac{1}{M}\left(\frac{-g_k^{\mathrm{T}}q_k}{\left\|q_k\right\|}\right)^2,\tilde{\Delta}\left(\frac{-g_k^{\mathrm{T}}q_k}{\left\|q_k\right\|}\right)\right\}\right]\\ &=D_k-\frac{1-\eta_{k+1}}{2}\nu h^{p_k}\min\left\{\frac{1}{M}\left(\frac{-g_k^{\mathrm{T}}q_k}{\left\|q_k\right\|}\right)^2,\tilde{\Delta}\left(\frac{-g_k^{\mathrm{T}}q_k}{\left\|q_k\right\|}\right)\right\} \end{aligned} \tag{3.19}$$

Using (3.12) and (3.19), then we get

$$D_k-D_{k+1}\ge\frac{1-\eta_{\max}}{2}\nu h^{p_k}\min\left\{\frac{\left(\theta\delta\right)^2}{M},\tilde{\Delta}\left(\theta\delta\right)\right\}. \tag{3.20}$$

We can conclude from Lemma 3.3 that the sequence $D_k$ is monotonically nonincreasing and $f_{k+1}\le D_{k+1}$. According to assumption (H1) that $f$ has a lower bound, then we deduce that $D_k$ is convergent. From (3.20) we have

$$\sum_{k=0}^{\infty}\frac{1-\eta_{\max}}{2}\nu h^{p_k}\min\left\{\frac{\left(\theta\delta\right)^2}{M},\tilde{\Delta}\left(\theta\delta\right)\right\}<\infty, \tag{3.21}$$

we define $\frac{1-\eta_{\max}}{2}\nu\min\left\{\frac{\left(\theta\delta\right)^2}{M},\tilde{\Delta}\left(\theta\delta\right)\right\}=\gamma$, then

$$\sum_{k=0}^{\infty}\gamma h^{p_k}<\infty. \tag{3.22}$$

From (3.17) and (3.22), we get there exists an index set H such that

$$\lim_{k\to\infty,k\in\mathrm{H}}\frac{-g_k^{\mathrm{T}}q_k}{\left\|q_k\right\|}\ne 0, \tag{3.23}$$

Therefore, $$\lim_{k\to\infty,k\in\mathrm{H}}h^{p_k}=0. \tag{3.24}$$

From (1.8), $\Delta_k\to 0$ as $k\to\infty$ and $k\in\mathrm{H}$. Now, assume that there are more than one inner cycles in the loop from Steps 3 to 4 at the *k*th iterate for all $k\in N$. Therefore, the solution $\tilde{d}_k$ of the subproblem

$$\begin{aligned} &\min\ m_k\left(d\right)=f_k+g_k^{\mathrm{T}}d+\frac{1}{2}d^{\mathrm{T}}B_kd\\ &\text{s.t.}\ \ \left\|d\right\|\le\Delta_k/h,\ \ k\in\mathrm{H} \end{aligned} \tag{3.25}$$

is not accepted at the *k*th iteration, then we denote

$$r_k=\frac{D_k-f\left(x_k+\tilde{d}_k\right)}{m_k\left(0\right)-m_k\left(\tilde{d}_k\right)}<\nu,\ \ k\in N, \tag{3.26}$$

but we have $r_k > \nu$ for $k \to \infty$ from Lemma 3.4, which is a contradiction with (3.26). This implies the result is valid. □

**Theorem 3.2.** If (H1)-(H3) holds, the sequence $\{x_k\}$ is generated by Algorithm2.1 converges to $x^*$, suppose that $\nabla^2 f(x)$ is a Lipschitz continuous matrix in a neighborhood $N(x^*, \varepsilon)$ of $x^*$, also suppose that $H(x)$ and $B_k$ are uniformly positive definite matrices such that

$$\lim_{k \to \infty} \frac{\left\| \left[ B_k - H(x^*) \right] d_k \right\|}{\left\| d_k \right\|} = 0 . \tag{3.27}$$

Proof. See [44] for reference. □

## 4. Conclusion

In this paper, we introduce the algorithm of a new nonmonotone adaptive trust region method for solving unconstrained optimization problems based on (1.8) and (2.1). The nonmonotone strategy is introduced into a new adaptive trust region. Maratos effects are avoided and the amount of calculation is reduced. Furthermore, it is obvious that the current objective function value $f_k$ is fully employed. With the help of nonmonotone technique and adaptive trust region radius, our algorithm can reduce the number of ineffective iterations so that we enhance the effectiveness of the algorithm. Under some standard and suitable assumptions, the global convergence and superlinear convergence of the new algorithm are analyzed theoretically. However, our algorithm still has some continuation and expansion, we can consider the following aspect: although this paper gives the theoretical proof of the proposed method as detailed as possible, it does not fully demonstrate the superiority of the new algorithm through numerical experiments, which will be the focus of further work.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

[1] Levenberg, K. (1944) A Method for Solution of Certain Problems in Least Squares. *Quarterly of Applied Mathematics*, **2**, 164-168. https://doi.org/10.1090/qam/10666

[2] Powell, M.J.D. (1970) A Hybrid Method for Nonlinear Equations. In: Rabinowitz, P., Ed., *Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach, London, 87-114.

[3] Powell, M.J.D. (1970) A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations. In: Rabinowitz, P., Ed., *Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach, London, 115-161.

[4] Powell, M.J.D. (1970) A New Algorithm for Unconstrained Optimization. *Nonlinear Programming: Proceedings of a Symposium Conducted by the Mathematics Research Center*, New York, 4-6 May 1970, 31-65. https://doi.org/10.1016/B978-0-12-597050-1.50006-3

[5] Powell, M.J.D. (1975) Convergence Properties of a Class of Minimization Algo-

rithms. *Nonlinear Programming* 2: *Proceedings of the Special Interest Group on Mathematical Programming Symposium Conducted by the Computer Sciences Department*, New York, 15-17 April 1974, 1-27. https://doi.org/10.1016/B978-0-12-468650-2.50005-5

[6] Fletcher, R. (1970) An Efficient, Global Convergent Algorithm for Unconstrained and Linearly Constrained Optimization Problems. Technical Report TP 431, AERE Harwell Laboratory, Oxfordshire.

[7] Hebden, M.D. (1973) An Algorithm for Minimization Using Exact Second Order Derivatives. Technical Report TP 515, AERE Harwell Laboratory, Oxfordshire.

[8] Madsen, K. (1975) An Algorithm for the Minimax Solution of Overdetermined Systems of Non-Linear Equations. *IMA Journal of Applied Mathematics*, **16**, 321-328. https://doi.org/10.1093/imamat/16.3.321

[9] Osborne, M.R. (1976) Nonlinear Least Squares—The Levenberg-Marquardt Algorithm Revisited. *The ANZIAM Journal*, **19**, 343-357. https://doi.org/10.1017/S033427000000120X

[10] Moré, J. (1978) The Levenberg-Marquardt Algorithm: Implementation and Theory. In: Watson, G.A., Ed., *Numerical Analysis*, Springer, Berlin, 105-116. https://doi.org/10.1007/BFb0067700

[11] Toint, P.L. (1978) Some Numerical Result Using a Sparse Matrix Updating Formula in Unconstrainded Optimization. *Mathematics of Computation*, **32**, 839-851. https://doi.org/10.1090/S0025-5718-1978-0483452-7

[12] Toint, P.L. (1979) On the Superlinear Convergence of an Algorithm for Solving a Sparse Minimization Problem. *SIAM Journal on Numerical Analysis*, **16**, 103-1045. https://doi.org/10.1137/0716076

[13] Toint, P.L. (1980) Sparsity Exploiting Quasi-Newton Methods for Unconstrained Optimization. In: Dixon, L.C.W., Spedicato, E. and Szego, G.P., Eds., *Nonlinear Optimization*: *Theory and Algorithms*, Birkhauser, Belgium, 65-90.

[14] Toint, P.L. (1981) Convergence Properties of a Class of Minimization Algorithms that Use a Possibly Unbounded Sequences of Quadratic Approximation. Technical Report 81/1, Department of Mathematics, University of Namur, Belgium.

[15] Toint, P.L. (1978) Towards an Efficient Sparsity Exploiting Newton Method for Minimization. In: Duff, I.S., Ed., *Sparse Matrices and Their Uses*, Academic Press, London, 57-88.

[16] Dennis, J.E. and Mei, H.H. (1979) Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values. *Journal of Optimization Theory and Applications*, **28**, 453-482. https://doi.org/10.1007/BF00932218

[17] Sorensen, D.C. (1982) Newton's Method with a Model Trust Region Modification. *SIAM Journal on Numerical Analysis*, **19**, 409-426. https://doi.org/10.1137/0719026

[18] Sorensen, D.C. (1982) Trust Region Methods for Unconstrained Optimization. In: Powell, M.J.D., Ed., *Non-Linear Optimization* 1981, Academic Press, London, 29-39.

[19] Steihaug, T. (1983) The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. *SIAM Journal on Numerical Analysis*, **20**, 626-637. https://doi.org/10.1137/0720042

[20] Moré, J.J. (1983) Recent Developments in Algorithms and Software for Trust Region Methods. In: Bachem, A., Grötschel, M. and Korte, B., Eds., *Mathematical Programming*: *The State of the Art*, Springer, Berlin, 258-287. https://doi.org/10.1007/978-3-642-68874-4_11

[21] Conn, A.R., Gould, N.I.M. and Toint, P.L. (2000) Trust Region Methods Trust Re-

gion Methods (MOS-SIAM Series on Optimization). Society for Industrial and Applied Mathematics, Philadelphia. https://doi.org/10.1137/1.9780898719857

[22] Sartenaer, A. (1997) Automatic Determination of an Initial Trust Region in Nonlinear Programming. *SIAM Journal on Scientific Computing*, **18**, 1788-1803. https://doi.org/10.1137/S1064827595286955

[23] Lin, C.J. and More, J.J. (1999) Newton's Method for Large Bound-Constrained Optimization Problems. *SIAM Journal on Optimization*, **9**, 1100-1127. https://doi.org/10.1137/S1052623498345075

[24] Zhang, X.S. (2000) NN Models for General Nonlinear Programming. *Neural Networks in Optimization*, **46**, 273-288. https://doi.org/10.1007/978-1-4757-3167-5_11

[25] Fan, J.Y. and Yuan, Y.X. (2001) A New Trust Region Algorithm with Trust Region Radius Converging to Zero. *Proceedings of the* 5*th International Conference on Optimization*: *Techniques and Applications*, Hong Kong, 15-17 December 2001, 786-794.

[26] Zhang, X.S., Zhang, J.L. and Liao, L.Z. (2002) An Adaptive Trust Region Method and Its Convergence. *Science in China: Series A*: *Mathematics*, *Physics*, *Astronomy*, **45**, 620-631.

[27] Shi, Z.J. and Wang, H.Q. (2004) A New Self-Adaptive Trust Region Method for Unconstrained Optimization. Technical Report, College of Operations Research and Management, Qufu Normal University.

[28] Gould, N.I.M., Orban, D., Sartenaer, A. and Toint, P.L. (2005) Sensitivity of Trust Region Algorithms to Their Parameters. 4*OR-A Quarterly Journal of Operations Research*, **3**, 227-241. https://doi.org/10.1007/s10288-005-0065-y

[29] Schnabel, R.B. and Eskow, E. (1990) A New Modified Cholesky Factorization. *SIAM Journal on Scientific and Statistical Computing*, **11**, 1136-1158. https://doi.org/10.1137/0911064

[30] Cui, Z.C. and Wu, B.Y. (2011) A New Self-Adaptive Trust Region Method for Unconstrained Optimization. *Journal of Vibration and Control*, **18**, 1303-1309. https://doi.org/10.1177/1077546311408473

[31] Shi, Z.J. and Guo, J.H. (2008) A New Trust Region Method for Unconstrained Optimization. *Journal of Computational and Applied Mathematics*, **213**, 509-520. https://doi.org/10.1016/j.cam.2007.01.027

[32] Kamandi, A., Amini, K. and Ahookhosh, M. (2017) An Improved Adaptive Trust Region Algorithm. *Optimization Letters*, **11**, 555-569. https://doi.org/10.1007/s11590-016-1018-4

[33] Grippo, L., Lampariello, F. and Lucidi, S. (1986) A Nonmonotone Line Search Technique for Newton's Method. *SIAM Journal on Numerical Analysis*, **23**, 707-716. https://doi.org/10.1137/0723046

[34] Ke, X.W. and Han, J.Y. (1998) A Class of Nonmonotone Trust Region Algorithms for Unconstrained Optimization. *Science in China Series A*: *Mathematics*, **41**, 927-932. https://doi.org/10.1007/BF02880001

[35] Sun, W.Y. (2004) Nonmonotone Trust Region Method for Solving Optimization Problems. *Applied Mathematics and Computation*, **156**, 159-174. https://doi.org/10.1016/j.amc.2003.07.008

[36] Fu, J. and Sun, W. (2005) Nonmonotone Adaptive Trust-Region Method for Unconstrained Optimization Problems. *Applied Mathematics and Computation*, **163**, 489-504. https://doi.org/10.1016/j.amc.2004.02.011

[37] Zhang, H. and Hager, W.W. (2004) A Nonmonotone Line Search Technique and Its

Application to Unconstrained Optimization. *SIAM Journal on Optimization*, **14**, 1043-1056. https://doi.org/10.1137/S1052623403428208

[38] Dai, Y.H. (2002) On the Nonmonotone Line Search. *Journal of Optimization Theory and Applications*, **112**, 315-330. https://doi.org/10.1023/A:1013653923062

[39] Mo, J., Liu, C. and Yan, S. (2006) A Nonmonotone Trust Region Method Based on Nonincreasing Technique of Weighted Average of the Successive Function Values. *Journal of Computational and Applied Mathematics*, **209**, 97-108. https://doi.org/10.1016/j.cam.2006.10.070

[40] Xue, Y., Liu, H.W. and Liu, Z.H. (2019) An Improved Nonmonotone Adaptive Trust Region Method. *Applications of Mathematics*, **64**, 335-350. https://doi.org/10.21136/AM.2019.0138-18

[41] Gu, N.Z. and Mo, J.T. (2008) Incorporating Nonmonotone Strategies into the Trust Region Method for Unconstrained Optimization. *Computers & Mathematics with Application*, **55**, 2158-2172. https://doi.org/10.1016/j.camwa.2007.08.038

[42] Yuan, Y.Y. and Sun, W.Y. (1997) Optimization Theory and Method. Science Press, Beijing.

[43] Nocedal, J. and Wright, S. (2006) Numerical Optimization. Springer Science & Business Media, Berlin.

[44] Ahookhosh, M. and Amini, K. (2010) A Nonmonotone Trust Region Method with Adaptive Radius for Unconstrained Optimization Problems. *Computers & Mathematics with Applications*, **60**, 411-422. https://doi.org/10.1016/j.camwa.2010.04.034