Iranian Journal of Numerical Analysis and Optimization
Vol. 5, No. 2, (2015), pp 95-117

An adaptive nonmonotone trust region method for unconstrained optimization problems based on a simple subproblem

Z. Saeidian and M.R. Peyghami*

Abstract

Using a simple quadratic model in the trust region subproblem, a new adaptive nonmonotone trust region method is proposed for solving unconstrained optimization problems. In our method, based on a slight modification of the proposed approach in (J. Optim. Theory Appl. 158(2):626-635, 2013), a new scalar approximation of the Hessian at the current point is provided. Our new proposed method is equipped with a new adaptive rule for updating the radius and an appropriate nonmonotone technique. Under some suitable and standard assumptions, the local and global convergence properties of the new algorithm as well as its convergence rate are investigated. Finally, the practical performance of the new proposed algorithm is verified on some test problems and compared with some existing algorithms in the literature.

Keywords: Trust region methods; Adaptive radius; Nonmonotone technique; Scalar approximation of the Hessian; Global convergence.

1 Introduction

In this paper, we deal with the following unconstrained optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function. Two popular classes of optimization techniques for solving (1) are line search and trust

*Corresponding author
Received 20 December 2014; revised 18 April 2015; accepted 27 July 2015
Z. Saeidian
Faculty of Mathematics, K.N. Toosi University of Technology, Tehran, Iran. E-mail : z.saeedian@dena.kntu.ac.ir

M.R. Peyghami
Scientific Computations in OPtimization and Systems Engineering (SCOPE), K.N. Toosi University of Technology, Tehran, Iran. E-mail: peyghami@kntu.ac.ir

region methods; see, e.g., [9,17,18]. Line search methods refer to a procedure in which one moves along a (descent) direction as long as a sufficient reduction in the objective is achieved. On the other hand, in the classical trust region methods, a trial step is computed by minimizing a (quadratic) model of the objective function at the current point over a region around this point. Then, using the so-called trust region ratio, the trial step is accepted/rejected and the new point as well as the radius is updated accordingly. It has been shown that trust region methods have appropriate global and local convergence properties. These methods have been widely studied in the literature; see, e.g., [9,12,17,19,24,25].

Here, let us briefly describe one step of the classical trust region method. Given x_k, the trial step d_k is computed by solving the following subproblem:

$$\min q_k(d) = g_k^T d + \frac{1}{2} d^T B_k d \qquad s.t.\ \|d\| \leq \Delta_k, \tag{2}$$

where $g_k = \nabla f(x_k)$, B_k is a $n \times n$ symmetric matrix which is $\nabla^2 f(x_k)$ or its approximation, $\Delta_k > 0$ is the so-called trust region radius, and $\|.\|$ refers to the Euclidean norm. Due to the so-called trust region ratio

$$r_k = \frac{f(x_k) - f(x_k + d_k)}{q_k(0) - q_k(d_k)}, \tag{3}$$

one decides whether the trial step is accepted or rejected; given $\mu \in (0,1)$, if $r_k \geq \mu$, then the trial step is accepted and the new point is introduced by $x_{k+1} = x_k + d_k$. Otherwise, the trial step is rejected and the current point remains unchanged for the next iteration. In both cases, the trust region radius is updated appropriately.

In the monotone trust region methods, the sequence of the objective values is monotonically decreasing. This may cause slow convergence rate in some problems. In order to overcome this disadvantage, the concept of nonmonotone strategies have been introduced in the framework of trust region methods, see, e.g., [13, 14]. A nonmonotone line search method was first proposed by Chamberlain et al. in [8]. Grippo et al. in [13] introduced a nonmonotone technique for Newton's method and developed it for unconstrained optimization in [14]. Nevertheless many advantages of the Grippo's technique, it suffers from some drawbacks [2, 3, 27]. In order to overcome these difficulties, recently, Ahookhosh and Amini in [2] and Ahookhosh et al. in [3] proposed a new nonmonotone term as below:

$$R_k = \epsilon_k f_{\ell(k)} + (1 - \epsilon_k) f_k, \tag{4}$$

where $f_k = f(x_k)$, $\epsilon_k \in [\epsilon_{\min}, \epsilon_{\max}] \subset [0,1]$ and $f_{\ell(k)}$ is the Grippo's nonmonotone term which is defined by

$$f_{\ell(k)} = \max_{0 \leq j \leq M(k)} f_{k-j}, \tag{5}$$

where $M(0) = 0$ and, for $k \geq 1$, $M(k) = \min\{k, M\}$, for given positive integer M. They employed (4) in the trust region ratio (3) and suggested nonmonotone trust region methods which are globally convergent. The reported numerical results on test problems confirm the efficiency and robustness of these methods in practice too.

The radius updating strategy is a crucial point in trust region methods [1, 21, 28]. In the classical trust region methods, this parameter is simply enlarged, shrunk or stayed unchanged based on the magnitude of r_k. Several strategies have been introduced in the literature for radius updating and initial radius choosing; see e.g. [11, 21–23, 29]. Zhang et al. in [29] proposed the radius update according to $\Delta_k = c^p \|g_k\| \|\hat{B}_k^{-1}\|$, where $c \in (0,1)$, p is a nonnegative integer and $\hat{B}_k = B_k + iI$ is a positive definite matrix, for some $i \in \mathbb{N}$. Although, Zhang's method uses more information of the objective function for updating the radius, it requires an estimation of $\|\hat{B}_k^{-1}\|$, which is costly. To reduce the computational cost of Zhang's updating rule, a simple adaptive rule was proposed by Shi and Wang in [23] according to $\Delta_k = c^p \frac{\|g_k\|^3}{g_k^T \hat{B}_k g_k}$, where $c \in (0,1)$, $\hat{B}_k$ is a positive definite matrix and p is a nonnegative integer. Despite Zhang's method that only updates the radius based on the current point information, some updating rules based on the information of the last two iterates have been introduced; see, e.g., [15,29,30]. Among them, Li [15] proposed an adaptive trust region method in which the radius is updated according to $\Delta_k = \frac{\|d_{k-1}\|}{\|y_{k-1}\|} \|g_k\|$, where $y_{k-1} = g_k - g_{k-1}$ and $d_{k-1} = x_k - x_{k-1}$.

The advantages of nonmonotone and adaptive techniques have been simultaneously employed in the framework of trust region methods. Using the adaptive strategy proposed in [15], Sang et al. in [20] introduced a nonmonotone adaptive trust region method based on a simple subproblem for large-scale unconstrained optimization problems which makes full use of information in the last two iterates. The idea of simple subproblem is originated from the fact that solving the subproblem (2) is costly especially when B_k is a large-scale and dense matrix. Therefore, the skills of the quasi-Newton method is used for correcting B_k by a real diagonal matrix ΔB_{k-1} from B_{k-1}. Recently, Zhou et al. in [30] constructed a simple subproblem according to the modification of the secant condition of Wei in [26] and introduced a nonmonotone adaptive trust region method based on the simple subproblem. Later, Biglari and Solimanpur in [7] proposed another simple subproblem with some superior properties to that of [30] in which the approximation of the Hessian at the current point x_k is computed by

$$\hat{\gamma}_k := \gamma(x_k) = \frac{4(f_{k-1} - f_k) + 3g_k^T d_{k-1} + g_{k-1}^T d_{k-1}}{d_{k-1}^T d_{k-1}}. \tag{6}$$

In this paper, we proposed a new nonmonotone adaptive trust region method based on simple subproblem for unconstrained optimization problems. Our approach is equipped with the nonmonotone technique as proposed in [2, 3], and uses a slight modification of the secant condition in [7] for constructing an approximation of the Hessian at the current point. Moreover, a modified version of the adaptive strategy in [20] is employed in the framework of the proposed algorithm. It is worth mentioning that the scalar approximation of the Hessian based on modified secant condition in [6] has superior to the standard Barzilai-Borwein method and its modifications. Under some standard assumptions, the global convergence property, as well as its superlinear convergence rate, is established. Numerical results show the efficiency of the proposed approach in practice comparing with some existing methods in the literature.

The rest of the paper is organized as follows: In Section 2, we present the structure of the new nonmonotone adaptive trust region method in details. The global convergence property, as well as its rate of convergence, is established in Section 3. Preliminary numerical results of applying the proposed algorithm on some test problems are given in Section 4. Finally, we end up the paper by some concluding remarks in Section 5.

2 The new algorithm

In this section, we propose a new adaptive nonmonotone trust region method for solving unconstrained optimization problems. Our algorithm combines the nonmonotone technique as proposed in [2] with an improved scalar approximation of the Hessian according to the modified secant equation as proposed in [6].

Let us describe one step of our new algorithm here: For given x_k, the trial step d_k is computed by (approximately) solving the following simple subproblem:

$$\min q_k(d) = g_k^T d + \frac{1}{2} d^T \gamma(x_k) d \qquad s.t.\ \|d\| \leq \Delta_k, \tag{7}$$

where $\gamma_k := \gamma(x_k)$ is a scalar approximation of the Hessian matrix. Since $\hat{\gamma}_k$, as defined by (6), may become negative in some iterations, we slightly modify (6) and define γ_k as below:

$$\gamma_k = \frac{4(f_{k-1} - f_k) + (3 + \eta_k) g_k^T d_{k-1} + g_{k-1}^T d_{k-1}}{d_{k-1}^T d_{k-1}}, \tag{8}$$

where η_k is computed by:

$$\eta_k = \begin{cases} \frac{4(f_k - f_{k-1}) - 3g_k^T d_{k-1} - g_{k-1}^T d_{k-1} + \delta}{g_k{}^T d_{k-1}}, & \text{if } \hat{\gamma}_k < 0, \\ 0, & \text{Otherwise,} \end{cases}$$

where δ is a small positive number. By this definition, it is obviously seen that $\gamma_k > 0$. Now, using d_k, the nonmonotone ratio is computed by:

$$r_k = \frac{R_k - f(x_k + d_k)}{Pred_k}, \tag{9}$$

where R_k is defined by (4) and $Pred_k = q_k(0) - q_k(d_k)$. For given $\mu \in (0, 1)$, the trial step is accepted whenever $r_k \geq \mu$; otherwise it is rejected. In both cases, the radius is adaptively updated according to $\Delta_k = \min\left\{\nu_k \frac{\|g_k\|}{\gamma_k}, \Delta_{\max}\right\}$, where $\Delta_{\max} > 0$ is a threshold value for the radii and ν_{k+1} is updated by:

$$\nu_{k+1} = \begin{cases} \sigma_0 \nu_k, & r_k < \mu_1, \\ \nu_k, & \mu_1 \leq r_k \leq \mu_2, \\ \min\{\sigma_1 \nu_k, \nu_{\max}\}, & r_k > \mu_2, \end{cases} \tag{10}$$

where $0 < \sigma_0 < 1 < \sigma_1$, $0 < \mu_1 < \mu_2 \leq 1$ and $\nu_{\max} > 0$ are given numbers. By the way, the new point is given by $x_{k+1} = x_k + d_k$ as long as $r_k \geq \mu$; otherwise, we set $x_{k+1} = x_k$.
The procedure of the new proposed nonmonotone trust region algorithm is outlined in Algorithm 1:

Algorithm 1: *A new nonmonotone adaptive trust region algorithm*

Input: $x_0 \in \mathbb{R}^n$, $0 < \mu < \mu_1 < \mu_2 \leq 1$, $0 < \sigma_0 < 1 < \sigma_1$, $0 < \epsilon_{\min} < \epsilon_{\max} < 1$, $\epsilon, \varepsilon, M, \nu_{\max}, \Delta_{\max} > 0$, $0 < \theta_1 < \theta_2$ and $\delta > 0$.

Step 0: Set $k = 0$, $\gamma_0 := \gamma(x_0) = 1$, $g_0 = g(x_0)$, $\nu_0 = 1$ and $\Delta_0 = \min\left\{\nu_0 \frac{\|g_0\|}{\gamma_0}, \Delta_{\max}\right\}$.

Step 1: **If** $\|g_k\| \leq \varepsilon$, **Then** Stop.

Step 2: Determine d_k by solving (7) and compute r_k using (9).

Step 3: **If** $r_k < \mu$, **Then** set $\Delta_k = \sigma_0 \Delta_k$, and goto Step 2.

Step 4: Set $x_{k+1} = x_k + d_k$.

Step 5: Compute γ_{k+1} using (8). **If** $\gamma_{k+1} \leq \epsilon$, **Then** set $\gamma_{k+1} = \theta_1$. **If** $\gamma_{k+1} \geq \frac{1}{\epsilon}$, **Then** set $\gamma_{k+1} = \theta_2$.

Step 6: Update ν_{k+1} using (10) and set $\Delta_{k+1} = \min\left\{\nu_{k+1} \frac{\|g_{k+1}\|}{\gamma_{k+1}}, \Delta_{\max}\right\}$. Set $k =: k + 1$ and goto Step 1.

Remark 1. Step 5 of Algorithm 1 implies that γ_k is a bounded positive number for all k. More precisely, we have $\min\{\epsilon, \theta_1\} \leq \gamma_k \leq \max\left\{\frac{1}{\epsilon}, \theta_2\right\}$.

Remark 2. The subproblem (7) can be easily solved by using the following procedure [20]: Let $\omega_k = \frac{g_k}{\gamma_k}$. If $\|\omega_k\| \leq \Delta_k$, then we set the trial step as $d_k = -\omega_k$. Otherwise, we choose $\alpha \in (0,1)$ so that $\|\alpha\omega_k\| = \Delta_k$. It can be easily verified that $\alpha = \frac{\Delta_k}{\|\omega_k\|}$. In this case, we set $d_k = -\alpha\omega_k = -\frac{\Delta_k}{\|\omega_k\|}\omega_k = -\frac{\Delta_k}{\|g_k\|}g_k$.

Remark 3. From Remark 2, one can easily see that, for all k, there exists a positive constant κ so that $\|d_k\| \leq \kappa\|g_k\|$.

3 Convergence analysis

In this section, our aim is to analyze the local and global convergence properties of Algorithm 1. For this purpose, the following assumption is imposed on the problem:

A1. The set $\Omega = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is a closed and bounded set and $f(x)$ is a twice continuously differentiable function over Ω. Moreover, $\nabla f(x)$ is a Lipschitz continuous function over Ω.

Lemma 1. *Assume that d_k is a solution of the problem (7). Then, one has:*

$$Pred_k := q_k(0) - q_k(d_k) \geq \frac{1}{2}\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\}. \tag{11}$$

Proof. We proceed the proof in the following two possible cases for d_k:

Case I. $\|-\frac{g_k}{\gamma_k}\| \leq \Delta_k$, and therefore, $d_k = -\frac{g_k}{\gamma_k}$: In this case one can easily obtain the following relations:

$$\begin{aligned} q_k(0) - q_k(d_k) &= q_k(0) - q_k\left(-\frac{g_k}{\gamma_k}\right) \\ &= -g_k^T\left(-\frac{g_k}{\gamma_k}\right) - \frac{1}{2}\left(-\frac{g_k}{\gamma_k}\right)^T \gamma_k \left(-\frac{g_k}{\gamma_k}\right) \\ &= \frac{\|g_k\|^2}{\gamma_k} - \frac{1}{2}\frac{\|g_k\|^2}{\gamma_k} = \frac{\|g_k\|^2}{2\gamma_k} \geq \frac{1}{2}\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\}. \end{aligned}$$

Case II. $\|-\frac{g_k}{\gamma_k}\| > \Delta_k$, and therefore, $d_k = -\frac{\Delta_k}{\|g_k\|}g_k$: In this case, we have:

$$\begin{aligned} q_k(0) - q_k(d_k) &= q_k(0) - q_k\left(-\frac{\Delta_k}{\|g_k\|}g_k\right) \\ &= -g_k^T\left(-\frac{\Delta_k}{\|g_k\|}g_k\right) - \frac{1}{2}\left(-\frac{\Delta_k}{\|g_k\|}g_k\right)^T \gamma_k \left(-\frac{\Delta_k}{\|g_k\|}g_k\right) \\ &= \Delta_k\|g_k\| - \frac{1}{2}\gamma_k\Delta_k^2 > \Delta_k\|g_k\| - \frac{1}{2}\Delta_k\|g_k\| \\ &= \frac{1}{2}\Delta_k\|g_k\| \geq \frac{1}{2}\|g_k\|\min\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\}, \end{aligned}$$

where the first inequality is obtained from the fact that $\gamma_k\Delta_k < \|g_k\|$.

Considering the above mentioned cases, the proof is completed. □

Lemma 2. *Let d_k be computed by the procedure as mentioned in Remark 2. Then, for all k, one has:*

$$|f(x_k) - f(x_k + d_k) - Pred_k| \leq O(\|d_k\|^2), \tag{12}$$

where $Pred_k$ is defined by (11).
Proof. Using Taylor's expansion and the fact that γ_k is bounded due to Remark 1, one can easily conclude the result. □
The following lemma states some appealing properties of the sequences $\{f_{\ell(k)}\}$ and $\{R_k\}$, which are defined by (5) and (4), respectively. One can find its proof in [2].

Lemma 3. *Suppose that Assumption A1 holds and the sequence $\{x_k\}$ is generated by Algorithm 1. Then, the following statements hold:*

i) *For all k, we have $f_k \leq R_k \leq f_{\ell(k)}$.*

ii) *The sequence $\{f_{\ell(k)}\}$ is a decreasing and convergent sequence.*

iii) $\lim_{k\to\infty} f_{\ell(k)} = \lim_{k\to\infty} f_k$.

iv) $\lim_{k\to\infty} R_k = \lim_{k\to\infty} f_k$.

Lemma 4. *Let Assumption A1 hold and the sequence $\{x_k\}$ be generated by Algorithm 1. Assume that there exists a constant $\zeta \in (0,1)$ so that $\|g_k\| > \zeta$, for all k. Then, for any k, there exists a nonnegative integer p so that x_{k+p+1} is a successful iteration point, i.e., $r_{k+p+1} > \mu$.*
Proof. Suppose that, on the contrary, there exists an iteration k so that, for all nonnegative integer p, the point x_{k+p+1} is an unsuccessful iteration point, i.e.,

$$r_{k+p} < \mu, \qquad p = 0, 1, 2, \ldots. \tag{13}$$

In this case, from Step 3 of Algorithm 1, we have

$$\Delta_{k+p+1} \leq \sigma_0^{p+1} \Delta_k.$$

This inequality together with the definition of Δ_k imply that:

$$\lim_{p\to\infty} \Delta_{k+p+1} = 0. \tag{14}$$

Therefore, from Lemma 1, Remark 1 and (12), we have

$$\begin{aligned}\left|\frac{f(x_{k+p}) - f(x_{k+p}+d_{k+p})}{Pred_{k+p}} - 1\right| &= \left|\frac{f(x_{k+p}) - f(x_{k+p}+d_{k+p}) - Pred_{k+p}}{Pred_{k+p}}\right| \\ &\leq \frac{O(\|d_{k+p}\|^2)}{\frac{1}{2}\|g_{k+p}\| \min\{\Delta_{k+p}, \frac{\|g_{k+p}\|}{\gamma_{k+p}}\}} \\ &\leq \frac{O(\|\Delta_{k+p}\|^2)}{\frac{1}{2}\zeta \min\left\{\Delta_{k+p}, \frac{\zeta}{\max\left\{\frac{1}{\epsilon}, \theta_2\right\}}\right\}}.\end{aligned}$$

This implies that $\left|\frac{f(x_{k+p})-f(x_{k+p}+d_{k+p})}{Pred_{k+p}} - 1\right| \to 0$, as $p \to \infty$. Thus, for sufficiently large p, using Lemma 3, we have

$$r_{k+p} = \frac{R_{k+p} - f(x_{k+p}+d_{k+p})}{Pred_{k+p}} \geq \frac{f(x_{k+p}) - f(x_{k+p}+d_{k+p})}{Pred_{k+p}} \to 1,$$

which contradicts $r_{k+p} < \mu$. This completes the proof of the lemma. □
Lemma 4 implies that the inner loop in Steps 2–3 of Algorithm 1 will be terminated after finite number of iterations, and therefore, Algorithm 1 is well-defined.
The following theorem provides the global convergence property of Algorithm 1 under some suitable and standard assumptions.

Theorem 1. *Suppose that Assumption A1 holds and $\{x_k\}$ is the sequence generated by Algorithm 1. Then, Algorithm 1 either stops at a stationary point or*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{15}$$

Proof. Suppose that Algorithm 1 does not stop at a stationary point. We show that (15) holds for the infinite sequence $\{x_k\}$. Assume that, on the contrary, there exists a positive constant ζ so that

$$\|g_k\| > \zeta > 0, \qquad \forall k. \tag{16}$$

Using Lemma 4, Algorithm 1 is well-defined and the inner loop in Steps 2–3 is terminated after finite number of iterations. Therefore, we may assume that $r_k \geq \mu$. Now, from (9) and Lemma 1, we have

$$R_k - f_{k+1} \geq \mu Pred_k \geq \frac{1}{2}\mu\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\}$$
$$\geq \frac{1}{2}\mu\zeta \min\left\{\Delta_k, \frac{\zeta}{\max\left\{\frac{1}{\epsilon}, \theta_2\right\}}\right\} \geq 0. \quad (17)$$

By taking limit from both sides of this inequality, as $k \to \infty$, and using Lemma 3, we conclude that

$$\Delta_k = \nu_k \frac{\|g_k\|}{\gamma_k} \to 0. \quad (18)$$

Now, using Remark 1 and (16), (18) implies that

$$\nu_k \longrightarrow 0. \quad (19)$$

Therefore, from (16) and Lemmas 1 and 2, we have

$$\left|\frac{f(x_k) - f(x_k + d_k)}{Pred_k} - 1\right| = \left|\frac{f(x_k) - f(x_k + d_k) - Pred_k}{Pred_k}\right|$$
$$\leq \frac{O\left(\|d_k\|^2\right)}{\frac{1}{2}\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\}}$$
$$\leq \frac{O\left(\Delta_k^2\right)}{\frac{1}{2}\zeta \min\left\{\Delta_k, \frac{\zeta}{\max\left\{\frac{1}{\epsilon}, \theta_2\right\}}\right\}} \overset{k\to\infty}{\longrightarrow} 0,$$

which implies that

$$r_k = \frac{R_k - f(x_k + d_k)}{Pred_k} \geq \frac{f(x_k) - f(x_k + d_k)}{Pred_k} \longrightarrow 1. \quad (20)$$

This shows that, for sufficiently large k, we have successful iterations. Therefore, there exists a positive constant ν^* so that, for sufficiently large k, $\nu_k \geq \nu^*$. This contradicts (19). □

Under some extra assumptions on the problem and using the same proof line of Theorem 3.7 in [30], one can construct the superlinear convergence rate of the sequence $\{x_k\}$, generated by Algorithm 1, to its limit point x^*.

4 Numerical results

In this section, we focus on providing some computational results of applying Algorithm 1, denoted by FATRA, along with the following algorithms on some test problems in order to compare their performances:

- NATRM: Algorithm 2.1 in [30];
- NATRA: Algorithm 2.1 in [30] in which the nonmonotone term in computing the trust region ratio r_k is replaced by R_k, as given by (4);
- FATRM: Algorithm 1 in which the nonmonotone term in computing the trust region ratio r_k is replaced by $f_{\ell(k)}$, as given by (5);

All the algorithms are implemented in MATLAB 7.10.0 (R2010a) environment on a PC with CPU 2.0 GHz and 4GB RAM memory and double precision format. The following parameters are considered in the relevant algorithms:

$$\mu = 0.1, \mu_1 = 0.25, \mu_2 = 0.75, \epsilon_{\min} = 10^{-6}, \epsilon_{\max} = 10^{6}, \Delta_{\max} = 100, M = 10,$$
$$\sigma_0 = c_2 = 0.5, \sigma_1 = c_1 = 4, \nu_{\max} = \sigma_1^4, \nu_0 = 0.25, \varepsilon = \epsilon = 10^{-6}, \delta = 10^{-6}.$$

Moreover, in Step 5 of Algorithm 1, if $\gamma_{k+1} \leq \epsilon$, then we set $\theta_1 = \epsilon$; if $\gamma_{k+1} > \frac{1}{\epsilon}$, then we set $\theta_2 = \frac{1}{\epsilon}$. The simple subproblem at each iteration is solved by the procedure as mentioned in Remark 2. All the algorithms are being stopped either $\|g_k\| \leq 10^{-6}$, or the number of iterations and/or function evaluations exceeds 50000. In the latter case, we declare that the algorithm is failed. The considered test problems are those in [30] as well as some large-scale problems taken from [16] and [4]. We have also utilized the advantages of the performance profile of Dolan and Moré in [10] to compare the performances of considered algorithms.
Numerical results are given in Table 1. In this table, *Prob* stands for the problem name, and n_i, n_f and f_{opt} denote the number of iterations, the number of function evaluations and the optimum value of the objective function, respectively. It should be noted that the number of gradient evaluations are almost the same as n_i.

Figures 1 and 2 show the performance profiles of the results in Table 1 based on the number of iterations and function evaluations, respectively. At a glance to Figure 1, we can find out that, in terms of n_i, FATRA solves all the considered test problems successfully, while the other algorithms have at least one failure in their runs. Moreover, FATRA and FATRM algorithms solve roughly 67% and 61% of the problems at the lowest value of n_i, respectively. This percentage for NATRM and NATRA algorithms are 49% and 47%, respectively. Figure 2 is drawn based on n_f of the results in Table 1. From this figure, it is revealed that FATRA solves all the problems successfully while FATRM has one failure in its run. Moreover, NATRM and NATRA algorithms solve roughly 96% and 98% of the test problems successfully. On the other hand, FATRA and FATRM algorithms solve about 58% and 60% of test problems in the lowest value of n_f while these percentages for NATRM and NATRA algorithms are about 34% and 22%.

Besides the performance profiles of the considered algorithms based on n_i and n_f, we have stored the average CPU time in 20 runs for each algorithms and drew the performance profile of the considered algorithms based on CPU time in Figure 3. The result shows that FATRA works well in this regard too. Based on the above mentioned arguments, one can easily realize that FATRA is competitive with FATRM, NATRM and NATRA algorithms in terms of n_i, n_f and CPU time. Moreover, the performance of FATRM is very close to FATRA.

Table 1: The numerical results

Prob	n	*NATRM*	*NATRA*	*FATRM*	*FATRA*
		$n_i/n_f/f_{opt}$	$n_i/n_f/f_{opt}$	$n_i/n_f/f_{opt}$	$n_i/n_f/f_{opt}$
Almost Perturbed Quadratic [4]	1000	655/900/1.93e-013	637/892/9.27e-014	493/973/2.03e-013	503/1046/7.99e-014
	5000	1595/2286/2.09e-013	1299/1895/1.85e-014	1272/2605/2.33e-013	1141/2331/5.99e-014
	10000	3192/4675/2.07e-013	2710/4156/6.38e-014	2627/5713/4.47e-014	1993/4350/5.43e-022
BIGGSB1(CUTE) [4]	100	933/1286/1.28e-010	666/996/2.05e-010	930/1801/2.48e-010	908/1824/2.67e-012
	500	11877/17751/6.31e-009	4979/7945/6.15e-010	10565/23394/3.19e-010	7179/16265/3.39e-012
	1000	Failed	7920/13527/4.08e-011	Failed	18672/43134/9.51e-009
Diagonal 4 [4]	1000	9/12/6.96e-018	9/12/6.96e-018	8/8/8.94e-022	8/8/8.94e-022
	5000	9/12/3.48e-018	9/12/3.48e-018	8/9/1.60e-022	8/9/1.60e-022
	10000	9/12/6.96e-0187	9/12/6.96e-018	8/8/8.94e-022	8/8/8.94e-022
Diagonal 5 [4]	1000	6/6/6.93e+002	6/6/6.93e+002	6/6/6.93e+002	6/6/6.93e+002
	5000	6/6/3.46e+003	6/6/3.46e+003	6/6/3.46e+003	6/6/3.46e+003
	10000	6/6/6.93e+003	6/6/6.93e+003	6/6/6.93e+003	6/6/6.93e+003

Table 1: The numerical results (continued)

$Prob$	n	$NATRM$ $n_i/n_f/f_{opt}$	$NATRA$ $n_i/n_f/f_{opt}$	$FATRM$ $n_i/n_f/f_{opt}$	$FATRA$ $n_i/n_f/f_{opt}$
Diagonal 7 [4]	1000	8/9/-8.16e+003	8/9/-8.16e+003	7/7/-8.16e+003	7/7/-8.16e+003
	5000	8/9/-4.08e+003	8/9/-4.08e+003	7/7/-4.08e+003	7/7/-4.08e+003
	10000	8/9/-8.16e+003	8/9/-8.16e+003	7/7/-8.16e+003	7/7/-8.16e+003
Diagonal 8 [4]	1000	6/7/-4.80e+003	6/7/-4.80e+003	6/6/-4.80e+003	6/6/-4.80e+003
	5000	6/7/-2.40e+003	6/7/-2.40e+003	6/6/-2.40e+003	6/6/-2.40e+003
	10000	6/7/-4.80e+003	6/7/-4.80e+003	6/6/-4.80e+003	6/6/-4.80e+003
DIXON3DQ(CUTE) [4]	100	1352/1912/2.92e-011	1107/1641/5.46e-011	1081/2123/2.46e-010	889/1782/4.24e-015
	500	13032/19407/5.56e-009	5386/8471/5.65e-011	10856/24068/5.36e-009	4994/11222/1.83e-009
	1000	30764/46139/4.49e-010	15248/25248/3.78e-014	Failed	15193/34094/1.73e-008
DQDRTIC(CUTE) [4]	1000	33/37/1.05e-016	33/37/1.05e-016	50/60/4.51e-019	47/58/1.88e-015
	5000	30/34/2.13e-015	30/34/2.13e-015	40/40/7.99e-017	40/40/7.99e-017
	10000	31/35/1.36e-016	31/35/1.36e-016	39/39/1.89e-016	36/41/6.72e-016

Table 1: The numerical results (continued)

Prob	n	$NATRM$ $n_i/n_f/f_{opt}$	$NATRA$ $n_i/n_f/f_{opt}$	$FATRM$ $n_i/n_f/f_{opt}$	$FATRA$ $n_i/n_f/f_{opt}$
Extended DENSCHNB [4]	1000	4/5/0	4/5/0	4/4/0	4/4/0
	5000	4/5/1.10e-027	4/5/1.10e-027	4/4/1.97e-027	4/4/1.97e-027
	10000	4/5/3.54e-026	4/5/3.54e-026	9/9/4.85e-022	9/9/4.85e-022
Extended Himmelblau [4]	1000	15/19/5.76e-018	15/19/5.76e-018	15/17/1.22e-024	15/17/1.22e-024
	5000	15/19/2.88e-018	15/19/2.88e-018	15/15/5.88e-016	15/15/5.88e-016
	10000	15/19/5.76e-018	15/19/5.76e-018	14/14/5.76e-018	14/14/5.76e-018
Extended PSC1 [4]	100	14/17/38.65	14/17/38.65	15/16/38.65	15/16/ 38.65
	500	14/17/1.93e+002	14/17/1.93e+002	14/15/1.93e+002	14/15/1.93e+002
	1000	14/17/3.86e+002	14/17/3.86e+002	15/16/3.86e+002	15/16/3.86e+002
Extended Tridiagonal 1 [4, 30]	1000	27/28/5.91e-009	27/28/5.91e-009	29/34/2.08e-009	29/34/2.08e-009
	5000	22/23/1.07e-008	22/23/1.07e-008	29/34/3.31e-009	28/34/3.39e-009
	10000	33/37/5.95e-009	33/37/5.95e-009	35/38/9.31e-009	35/38/9.31e-009

Table 1: The numerical results (continued)

Prob	n	*NATRM* $n_i/n_f/f_{opt}$	*NATRA* $n_i/n_f/f_{opt}$	*FATRM* $n_i/n_f/f_{opt}$	*FATRA* $n_i/n_f/f_{opt}$
Extended White and Holst [4, 30]	1000	81/126/9.10e-013	81/126/9.10e-013	51/74/1.29e-016	51/74/1.29e-016
	5000	82/127/2.27e-012	82/127/2.27e-012	51/64/1.77e-013	51/64/1.77e-013
	10000	84/129/4.09e-018	84/129/4.09e-018	71/98/1.41e-012	71/98/1.41e-012
Extended Wood [4]	1000	362/535/1.16e-014	362/535/1.16e-014	727/1486/2.71e-015	729/1477/1.14e-013
	5000	340/482/1.71e-014	340/482/1.71e-014	738/1347/5.33e-016	775/1477/9.67e-014
	10000	154/234/4.16e-013	163/247/2.71e-016	810/1494/1.21e-013	819/1667/1.26e-014
FLETCHCR [4]	100	1628/2261/5.15e-013	1342/1998/5.49e-013	1470/2969/3.70e-013	1349/2816/4.92e-013
	500	17074/25117/1.48e-011	10312/16396/9.85e-012	11449/24915/1.37e-011	9827/23269/1.001e-011
	1000	Failed	21632/36672/5.86e-011	Failed	Failed
Full Hessian FH2 [4]	100	958/1365/6.90e-013	1806/3034/5.98e-014	1185/2513/1.55e-013	1386/3161/9.65e-013
	500	10604/15314/9.06e-013	7234/11124/8.75e-014	8535/18335/4.49e-013	11448/26746/7.49e-013
	1000	31551/45869/7.05e-013	Failed	Failed	Failed

Table 1: The numerical results (continued)

Prob	n	*NATRM* $n_i/n_f/f_{opt}$	*NATRA* $n_i/n_f/f_{opt}$	*FATRM* $n_i/n_f/f_{opt}$	*FATRA* $n_i/n_f/f_{opt}$
Full Hessian FH3 [4]	1000	5/11/-0.24	5/11/-0.24	5/6/-0.24	5/6/-0.24
	5000	5/12/-0.24	5/12/-0.24	5/5/-0.24	5/5/-0.24
	10000	5/12/-0.24	5/12/-0.24	4/4/-0.24	4/4/-0.24
Generalized Quartic [4]	1000	12/14/2.46e-019	12/14/2.46e-019	12/13/4.31e-018	12/13/4.31e-018
	5000	11/13/6.55e-015	11/13/6.55e-015	14/14/8.49e-021	14/14/8.49e-021
	10000	10/12/9.61e-014	10/12/9.61e-014	10/10/5.74e-021	10/10/5.74e-021
Generalized Rosenbrock [4]	100	3866/5551/6.41e-013	3653/5351/5.45e-014	3675/7761/6.30e-013	3574/7681/9.96e-013
	500	13137/18512/9.78e-013	12942/18263/9.15e-013	12098/24912/2.10e-013	12031/24890/7.25e-013
	1000	24527/34507/9.87e-013	24550/34658/2.27e-013	23410/48550/2.62e-013	23352/48494/5.17e-014
Generalized Tridiagonal 1 [4]	100	29/31/97.21	29/31/97.21	29/30/97.21	29/30/97.21
	500	29/31/4.97e+002	29/31/4.97e+002	29/30/4.97e+002	29/30/4.97e+002
	1000	29/31/9.97e+002	29/31/9.97e+002	29/30/9.97e+002	29/30/9.97e+002

Table 1: The numerical results (continued)

Prob	n	$NATRM$ $n_i/n_f/f_{opt}$	$NATRA$ $n_i/n_f/f_{opt}$	$FATRM$ $n_i/n_f/f_{opt}$	$FATRA$ $n_i/n_f/f_{opt}$
IE [16]	1000	9/10/ 1.54e-011	10/11/4.50e-15	9/9/ 1.26e-11	10/10/3.51e-15
	5000	10/11/2.25e-014	10/11/2.25e-014	10/10/1.75e-014	10/10/1.75e-014
	10000	10/11/ 9.00e-015	10/11/ 9.00e-015	10/10/7.02e-15	10/10/7.02e-15
LIARWHD [4]	1000	53/84/2.89e-018	53/84/2.89e-018	64/102/2.83e-018	64/102/2.83e-018
	5000	49/82/2.48e-021	49/82/2.48e-021	46/69/1.36e-014	46/69/1.36e-014
	10000	58/105/2.36e-019	58/105/2.36e-019	58/88/4.02e-017	58/88/4.02e-017
NONDIA [4]	100	22/30/1.59e-023	22/30/1.59e-023	19/22/4.53e-016	19/22/4.53e-016
	500	24/33/3.75e-024	24/33/3.75e-024	17/23/5.34e-021	17/23/5.34e-021
	1000	18/28/1.69e-013	18/28/1.69e-013	17/23/1.09e-018	17/23/1.09e-018
PEN1 [16]	100	309/555/9.02e-004	162/290/ 9.02e-04	35/39/ 9.02e-004	42/46/9.02e-04
	500	252/426/0.004	290/526/0.004	107/121/0.004	107/121/0.004
	1000	87/165/9.68e-3	204 /338/ 9.68e-3	229/249/0.9.68e-3	230/250/9.68e-3

Table 1: The numerical results (continued)

Prob	n	$NATRM$ $n_i/n_f/f_{opt}$	$NATRA$ $n_i/n_f/f_{opt}$	$FATRM$ $n_i/n_f/f_{opt}$	$FATRA$ $n_i/n_f/f_{opt}$
Perturbed Quadratic [4]	1000	420/555/2.23e-015	637/919/2.33e-013	513/1020/2.28e-013	538/1100/1.61e-013
	5000	1928/2826/2.42e-013	1675/2582/2.33e-013	1138/2379/1.77e-013	1046/2277/1.84e-013
	10000	2007/2927/4.4263714e-014	2112/3291/1.32e-013	1816/3947/2.32e-013	1671/3790/2.45e-013
Perturbed quadratic diagonal [4]	1000	470/735/1.20e-011	1324/2497/2.03e-011	522/1238/1.68e-011	1103/2921/1.50e-011
	5000	1219/2003/2.17e-011	2789/5432/1.92e-011	1407/3528/1.98e-011	1152/3067/1.32e-011
	10000	1867/3122/5.17e-012	5785/11286/2.21e-011	1289/3250/8.13e-012	2687/7408/1.61e-011
Quadratic QF1 [4]	1000	451/587/-4.99e-004	515/723/-4.99e-004	576/1194/-4.99e-004	694/1470/-4.99e-004
	5000	2034/2931/-9.99e-005	1598/2446/-9.99e-005	2147/4663/-9.99e-005	1889/4172/-9.99e-005
	10000	2467/3604/-4.99e-005	2789/4392/-4.99e-005	2445/5195/-4.99e-005	1651/3483/-4.99e-005
QUARTC [4]	1000	2/3/0	2/3/0	2/2/0	2/2/0
	5000	2/3/0	2/3/0	2/2/0	2/2/0
	10000	2/3/0	2/3/0	2/2/0	2/2/0

Table 1: The numerical results

Prob	n	*NATRM* $n_i/n_f/f_{opt}$	*NATRA* $n_i/n_f/f_{opt}$	*FATRM* $n_i/n_f/f_{opt}$	*FATRA* $n_i/n_f/f_{opt}$
Raydan 1 [4,30]	1000	8/8/10000	8/8/10000	7/7/10000	7/7/10000
	5000	8/8/5000	8/8/5000	7/7/5000	7/7/5000
	10000	8/8/10000	8/8/10000	7/7/10000	7/7/10000
ROSEX [16]	1000	61/122 / 4.50e-021	64/100/5.18e-17	68/109/2.05e-014	68/109/2.05e-14
	5000	60/118 /2.73e-020	67/105 /2.93e-17	32/38/3.41e-016	32/38/3.41e-16
	10000	56/115/5.95e-016	71/151/4.66e-16	44/58/ 7.75e-016	44/58/3.23e-16
SINGX [16]	100	517/775/ 2.64e-009	7724 / 14030 / 4.01e-09	373/769/9.74e-008	491/1003/ 1.84e-09
	500	384/595/2.91e-009	7474/13674/2.91e-009	575/1178/5.88e-009	575/1178/5.88e-009
	1000	28640/46632/2.66e-006	3421 /6120/ 5.70e-09	350/479/1.42e-007	1045/2440/4.90e-09
TRIDIA [4]	1000	1984/2896/3.30e-013	2510/3880/3.42e-013	2657/5835/3.41e-013	2185/4915/4.29e-014
	5000	13693/20572/3.39e-013	11464/17956/6.38e-014	10928/24370/3.46e-013	9142/21255/3.40e-013
	10000	20721/31085/3.39e-013	13317/21063/1.53e-014	22037/48796/2.59e-013	18893/42973/4.81e-017

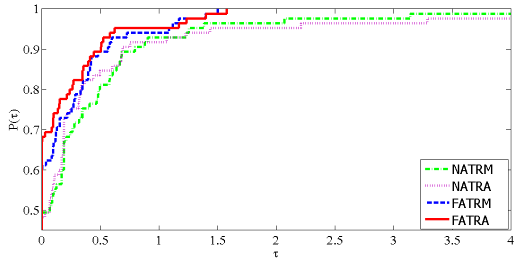

Figure 1: Performance profile of considered algorithms based on n_i

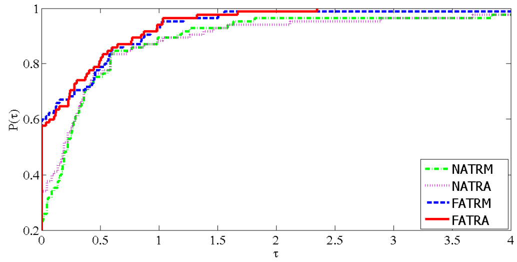

Figure 2: Performance profile of considered algorithms based on n_f

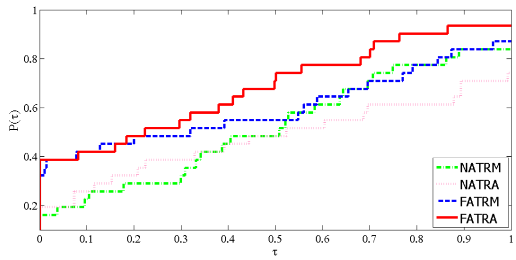

Figure 3: Performance profile of considered algorithms based on CPU time

5 Conclusion

In this paper, a new nonmonotone adaptive trust region method for solving unconstrained optimization problems based on a simple subproblem is presented. The new proposed algorithm uses the advantage of the adaptive trust region method, as proposed in [5], with the nonmonotone term, as suggested in [2]. The global convergence property of the new proposed method is established under some standard assumptions. Numerical results on some large-scale test problems confirm the efficiency and effectiveness of the new proposed algorithm in comparison with some other existing algorithms in the literature.

Acknowledgment: The authors would like to thank the Research Council of K. N. Toosi University of Technology and the SCOPE research center for supporting this work. The authors also would like to appreciate D. Ataee Tarzanagh for his helpful and constructive comments.

References

1. Ahookhosh, M. and Amini, K. *A nonmonotone trust region method with adaptive radius for unconstrained optimization*, Comput. Math. Appl. 60(2010) 411–422.

2. Ahookhosh, M. and Amini, K. *An efficient nonmonotone trust-region method for unconstrained optimization*, Numer. Algorithms 59(2011) 523–540.

3. Ahookhosh, M., Amini, K. and Peyghami, M.R. *A nonmonotone trust-region line search method for large- scale unconstrained optimization*, Appl. Math. Model. 36(2012) 478–487.

4. Andrei, N. *An unconstrained optimization test functions collection*, Adv. Model. Optim. 10(1)(2008) 147–161.

5. Ataee Tarzanagh, D., Peyghami, M. R. and Mesgarani, H. *A new nonmonotone trust region method for unconstrained optimization equipped by an efficient adaptive radius*, Optim. Methods Softw. 29(4)(2014) 819–836.

6. Biglari, F., Hassan, M. and Leong, W. J. *New quasi-Newton methods via higher order tensor models*, J. Comput. Appl. Math. 235(2011) 2412–2422.

7. Biglari, F. and Solimanpur, M. *Scaling on the Spectral Gradient Method*, J. Optim. Theory Appl. 158(2)(2013) 626–635.

8. Chamberlain, R. M., Powell, M. J. D., Lemarechal, C. and Pedersen, H. C. *The watchdog technique for forcing convergence in algorithm for constrained optimization*, Math. Program. Stud. 16(1982) 1–17.

9. Conn, A., Gould, N. and Toint, Ph. L. *Trust Region Methods*, SIAM, Philadelphia, 2000.

10. Dolan, E. and Moré, J. J. *Benchmarking optimization software with performance profiles*, Math. Program. 91(2002) 201–213.

11. Fan, J. Y. and Yuan, Y. X. *A new trust region algorithm with trust region radius converging to zero*, Proceedings of the 5th International Conference on Optimization, Techniques and Applications, 2001.

12. Gertz, E. M. *A quasi-Newton trust-region method*, Math. Program. Ser. A. 100(3)(2004) 447–470.

13. Grippo, L., Lampariello, F. and Lucidi, S. *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal. 23(1986) 707–716.

14. Grippo, L., Lampariello, F. and Lucidi, S. *A truncated Newton method with nonmonotone line-search for unconstrained optimization*, J. Optim. Theory Appl. 60(1989) 401–419.

15. Li, G. D. *A trust region method with automatic determination of the trust region radius*, Chinese J. Engry. Math. 23(2006) 843–848.

16. Moré, J. J., Garbow, B. S. and Hilstron, K. E. *Testing unconstrained optimization software*, ACM Tran. Math. Software 7(1981) 17–41.

17. Nocedal, J. and Wright, S. J. *Numerical Optimization*, Springer, New York, 2006.

18. Nocedal, J. and Yuan, Y. *Combining trust region and line search techniques*, In: Y. Yuan (ed.), Advanced in Nonlinear Programming, Kluwer Academic, Dordrecht, 153–175, 1996.

19. Powell, M. J. D. *On the global convergence of trust region algorithms for unconstrained optimization*, Math. Program. 29(1984) 297–303.

20. Sang, Z. and Sun, Q. *A self-adaptive trust region method with line search based on a simple subproblem model*, J. Appl. Math. Comput. 232(2009) 514–522.

21. Sartenaer, A. *Automatic determination of an initial trust region in nonlinear programming*, SIAM J. Sci. Comput. 18(6)(1997) 1788–1803.

22. Shi, Z. and Guo, J. *A new trust region method for unconstrained optimization*, J. Comput. Appl. Math. 213(2008) 509–520.

23. Shi, Z. J. and Wang, H. Q. *A new self-adaptive trust region method for unconstrained optimization*, Technical Report, College of Operations Research and Management, Qufu Normal University, 2004.

24. Shultz, G. A., Schnabel, R. B. and Byrd, A. R. H. *A family of trust-region-based algorithm for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal. 22(1)(1985) 47–67.

25. Toint, Ph. L. *Non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints*, Math. Program. 77(1997) 69–94.

26. Wei, Z., Li, G. and Qi, L. *New quasi-Newton methods for unconstrained optimization problems*, Appl. Math. Comput. 175(2006) 1156–1188.

27. Zhang, H. C. and Hager, W. W. *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim. 14(2004) 1043–1056.

28. Zhang, X. S., Zhang, J. L. and Liao, L. Z. *A nonmonotone adaptive trust region method and its convergence*, Int. J. Comput. Math. Appl. 45(2003) 1469–1477.

29. Zhang, X. S., Zhang, J. L. and Liao, L. Z. *An adaptive trust region method and its convergence*, Sci. China 45(2002) 620–631.

30. Zhou, Q. and Zhang, C. *A new nonmonotone adaptive trust region method based on simple quadratic models*, J. Appl. Math. Comput. 40(2012) 111–123.

یک روش ناحیه اعتماد وفقی نایکنوا برای مسایل بهینه سازی نامقید بر اساس یک زیرمساله ساده

زینب سعیدیان طربی[۱] و *محمدرضا پیغامی*[۱،۲]

[۱] دانشگاه صنعتی خواجه نصیر الدین طوسی، دانشکده ریاضی
[۲] دانشگاه صنعتی خواجه نصیر الدین طوسی، مرکز پژوهشی محاسبات علمی در بهینه سازی و مهندسی سامانه

چکیده : با بکارگیری یک مدل درجه دوم ساده در زیرمساله ناحیه اعتماد، یک روش ناحیه اعتماد وفقی نایکنوا برای حل مسایل بهینه‌-سازی نامقید پیشنهاد می‌-شود. در این روش، با اعمال اصلاح جزیی روی روش پیشنهادی در (J. Optim. Theory Appl. ۱۵۸ (۲) ۶۲۶-۶۳۵ ، ۲۰۱۳) یک تقریب اسکالر جدید از ماتریس هسین در نقطه فعلی ارایه می‌-‌-گردد. روش پیشنهادی به یک روش وفقی جدید برای بهنگام شعاع ناحیه اعتماد و یک تکنیک نایکنوا مجهز شده است. تحت برخی فرضیات استاندارد و مناسب، خواص همگرایی سراسری و موضعی الگوریتم پیشنهادی به همراه نرخ همگرایی آن بررسی می‌-شوند. در پایان، عملکرد عملی الگوریتم پیشنهادی روی برخی مسایل آزمونی مورد بررسی قرار گرفته و نتایج حاصل با برخی الگوریتم‌-های موجود در ادبیات موضوع مورد مقایسه قرار می‌-گیرند.

کلمات کلیدی : روش‌-های ناحیه اعتماد؛ شعاع وفقی؛ تکنیک نایکنوا؛ تقریب اسکالر ماتریس هسین؛ همگرایی سراسری.