Computing 48, 303–317 (1992)

Computing

Nonmonotone Trust Region Methods with Curvilinear Path in Unconstrained Optimization*

Y. Xiao and **F. Zhou**, Beijing

Received December 23, 1991

Abstract — Zusammenfassung

Nonmonotone Trust Region Methods with Curvilinear Path in Unconstrained Optimization. A general nonmonotone trust region method with curvilinear path for unconstrained optimization problem is presented. Although this method allows the sequence of the objective function values to be nonmonotone, convergence properties similar to those for the usual trust region methods with curvilinear path are proved under certain conditions. Some numerical results are reported which show the superiority of the nonmonotone trust region method with respect to the numbers of gradient evaluations and function evaluations.

AMS(MOS) Subject Classifications: 65K05, 65K10

Key words: Unconstrained optimization, trust region, curvilinear path, nonmonotone, convergence.

Ein nichtmonotones Konfidenzbereichs-Verfahren mit krummlinigem Pfad zur unrestringierten Optimierung. Es wird ein allgemeines nichtmonotones Konfidenzbereichs-Verfahren mit krummlinigem Pfad für die unrestringierte Optimierung vorgeschlagen. Obwohl bei diesem Verfahren die Folge der Werte der Objektfunktion nicht monoton zu sein braucht, werden Konvergenzeigenschaften bewiesen, die denen der gängigen Verfahren dieser Art entsprechen. An Hand einiger numerischer Beispiele wird die Überlegenheit des nichtmonotonen Verfahrens bezüglich der Zahl der Gradienten- und der Funktionsauswertungen gezeigt.

1. Introduction

In this paper we consider the unconstrained minimization problem

$$(P) \qquad \min f(x), \qquad x \in R^n$$

where $f: R^n \to R^1$ is a twice continuously differentiable function.

Trust region method is an algorithm which can ensure global convergence and locally fast iterative processes for optimization. It has proven to be effective and robust for solving unconstrained minimization problem. The idea of combining curvilinear paths and trust regions is originally due to Powell [13], Sorensen [16, 17], Moré and Sorensen [11], and Bulteau and Vial [1]. The basic step of these

* This work, in part, was supported by the Natural Science Foundation of Tsinghua University.

algorithms is to find a trial point which is a minimizer of the local quadratic model of the function within the trust region along a curve. All their methods always constructed an iteration series $\{x_k\}$ such that the series $\{f_k\} = \{f(x_k)\}$ is monotonically decreasing, but it appears that an ideal trust region method should allow increase in the function value at some steps, while retaining global convergence. This paper is the first to propose a trust region method which combines the curvilinear path and allows the sequence $\{f_k\}$ to be nonmonotone.

The paper is organized as follows. In Section 2, a nonmonotone trust region method with curvilinear path(NTRCP) is described. We discuss its convergence properties in Section 3. Under certain conditions, we prove the convergence results, which are similar to those of usual trust region method with curvilinear path(UTRCP), though the algorithm allows the sequence $\{f_k\}$ to be nonmonotone. In Section 4, the NTRCP method and the UTRCP method are compared by the numerical experiments. The numerical results show that the nonmonotone trust region method with curvilinear path is superior to the usual trust region method with curvilinear path according to both the numbers of gradient evaluations and function evaluations. The vector norm used in this paper is an arbitrary norm of R^n, and the norm of a matrix is one which is consistent with the vector norm.

2. General NTRCP Algorithm

Let $\nabla f(x)$ and $\nabla^2 f(x)$ represent the gradient and the Hessian matrix of f, $g_k \in R^n$ be an approximation to $\nabla f(x_k)$, and $B_k \in R^{n,n}$ be some symmetric matrix. Define a curve $\gamma_{g_k B_k}$: $[0, \infty) \to R^n$, $\gamma_{g_k B_k}(0) = 0$, $\Gamma(g_k, B_k)$ be the closure of the image set of $\gamma_{g_k B_k}$. For the sake of simplicity, Γ^k and γ^k will denote respectively the set $\Gamma(g_k, B_k)$ and the function $\gamma_{g_k B_k}$.

For easy reference, let us recall UTRCP methods first. The UTRCP methods are based on a local quadratic model of f about the k-th iteration x_k.

$$\phi_k(w) = f_k + g_k^T w + w^T B_k w/2 \tag{2.1}$$

and a curve $\gamma_{g_k B_k}$, where $w = x - x_k$.

The following is a specific algorithm of the UTRCP methods.

Algorithm 2.1 (UTRCP)

Data: $x_0 \in R^n$, $0 < \bar{\Delta} \in R^1$, $\mu \in (0, 1)$, $\eta \in (0, \mu)$, $0 < \gamma_1 < 1 < \gamma_2$, $0 < \Delta^0 \leq \bar{\Delta}$, $\varepsilon > 0$.

Step 1. Set $k = 0$, compute $f_0 = f(x_0)$.

Step 2. Compute $g_k = g(x_k)$. If $\|g_k\| < \varepsilon$, stop and set $x^* = x_k$; otherwise, compute B_k, and let $\Delta_{\max}^k = \sup\{\|\gamma^k(t)\|\}$, $\delta^k = \min\{\Delta^k, \Delta_{\max}^k\}$.

Step 3. Find an approximate solution $s_k \in \Gamma^k$ to the problem

$$(SP) \qquad \min\{\phi_k(w): w \in \Gamma^k, \|w\| \leq \delta^k\} \tag{2.2}$$

where $\phi_k(w)$ is defined by (2.1).

Step 4. Compute $f_{k+1} = f(x_k + s_k)$ and

$$ared_k = f_k - f_{k+1}, \qquad pred_k = f_k - \phi_k(s_k) \tag{2.3}$$

$$\rho_k = \frac{ared_k}{pred_k} \tag{2.4}$$

Step 5. If $\rho_k \geq \mu$, then set $x_{k+1} = x_k + s_k$, go to Step 6; else if $0 < \mu - \rho_k < \eta$, then set $\delta^k \in [\gamma_1 \delta^k, \delta^k)$, go to Step 3; else if $\mu - \rho_k \geq \eta$, then set $\delta^k \in (0, \gamma_1 \delta^k)$, go to Step 3.

Step 6. Update Δ^{k+1} such that

$$\delta^k \leq \Delta^{k+1} \leq \min\{\gamma_2 \delta^k, \bar{\Delta}\} \tag{2.5}$$

set $k = k + 1$, go to Step 2.

Remark 2.1. In the above algorithm the approximate solution s_k to the subproblem (2.2) is acceptable if and only if $\rho_k \geq \mu$. Therefore the algorithm guarantees a certain decrease of the objective function value after every iteration. So the sequence $\{f_k\}$ is monotonically decreasing.

Relaxing the acceptability condition on s_k, a nonmonotone trust region method with curvilinear path(NTRCP) is obtained.

Algorithm 2.2(NTRCP)

Data: $x_0 \in R^n$, $0 < \bar{\Delta} \in R^1$, $\mu \in (0,1)$, $\eta \in (0, \mu)$, $0 < \gamma_1 < 1 < \gamma_2$, $0 < \Delta^0 \leq \bar{\Delta}$, $\varepsilon > 0$, $\gamma > 0$, and a nonnegative integer M.

Step 1. Set $k = 0$, $m(k) = 0$, compute $f_0 = f(x_0)$.

Step 2. Compute $g_k = g(x_k)$. If $\|g_k\| < \varepsilon$, stop and set $x^* = x_k$; otherwise, compute B_k,

$$f_{l(k)} = f(x_{l(k)}) = \max_{0 \leq j \leq m(k)} \{f_{k-j}\} \tag{2.6}$$

and let $\Delta_{\max}^k = \sup\{\|\gamma^k(t)\|\}$, $\delta^k = \min\{\Delta^k, \Delta_{\max}^k\}$.

Step 3. Find an approximate solution $s_k \in \Gamma^k$ to the problem

$$(SP) \qquad \min\{\phi_k(w) \colon w \in \Gamma^k, \|w\| \leq \delta^k\} \tag{2.7}$$

where $\phi_k(w)$ is defined by (2.1).

Step 4. Compute $f_{k+1} = f(x_k + s_k)$ and

$$ared_k = f_k - f_{k+1}, \qquad pred_k = f_k - \phi_k(s_k) \tag{2.8}$$

and

$$\mu_k = \begin{cases} [(f_{l(k)} - f_k) - \gamma \delta^k \|g_k\|]/(-pred_k), & \text{if } M > 0; \\ \mu, & \text{if } M = 0. \end{cases} \tag{2.9}$$

$$\bar{\mu}_k = \min\{\mu, \mu_k\} \tag{2.10}$$

$$\rho_k = \frac{ared_k}{pred_k} \tag{2.11}$$

Step 5. If $\rho_k \geq \bar{\mu}_k$, then set $x_{k+1} = x_k + s_k$, go to Step 6; else if $0 < \bar{\mu}_k - \rho_k < \eta$, then set $\delta^k \in [\gamma_1 \delta^k, \delta^k)$, go to Step 3; else if $\bar{\mu}_k - \rho_k \geq \eta$, then set $\delta^k \in (0, \gamma_1 \delta^k)$, go to Step 3.

Step 6. Update Δ^{k+1} such that

$$\delta^k \leq \Delta^{k+1} \leq \min\{\gamma_2 \delta^k, \bar{\Delta}\} \tag{2.12}$$

set $k = k + 1$ and $m(k) = \min\{m(k-1) + 1, M\}$, go to Step 2.

Remark 2.2. Comparing Algorithm 2.1 with Algorithm 2.2, it is easy to see that they are identical when $M = 0$ in Algorithm 2.2. So Algorithm 2.1 can be viewed as a special case of Algorithm 2.2.

Remark 2.3. In Algorithm 2.2, $\{f_k\}$ may be not monotonically decreasing if $M \neq 0$.

In order to guarantee that Algorithm 2.2 is well defined, we give a set of assumptions that $\Gamma(g_k, B_k)$ satisfy

A1. $\gamma_{g_k B_k}(t)$ is continuous on $(0, \infty)$ and differentiable at $t = 0$. Moreover $g_k^T \dfrac{d\gamma_{g_k B_k}}{dt}(0) \neq 0$ if $g_k \neq 0$.

A2. $\|\gamma_{g_k B_k}(t)\|$ is strictly increasing with t. Either $\lim_{t\to\infty} \gamma_{g_k B_k}(t)$ exists or $\|\gamma_{g_k B_k}(t)\| \to \infty$ as $t \to \infty$.

A3. $\phi_k(\gamma_{g_k B_k}(t))$ is strictly decreasing with t.

Theorem 2.1. *Under the assumptions* A1, A2, *and* A3, *Algorithm 2.2 is well defined.*

Proof. This proof is similar to that of Theorem 2.3 in [1]. The detail is omitted. Q.E.D.

3. Convergence Analysis

We first give some other assumptions that the arc $\Gamma(g_k, B_k)$ and s_k must partially or completely satisfy in order to achieve convergence properties when it is implemented with Algorithm 2.2. These conditions are similar to those used for the UTRCP method.

A4. Let $l = \sup\{\|w\| : w \in \Gamma(g_k, B_k)\}$. Moreover, if B_k is positive definite, $l \leq \|g_k\| \, \|B_k^{-1}\|$.

A5. There exist two constants $c_1, c_2 > 0$, independent of g_k and B_k, such that, for all $\delta^k \geq 0$.

$$f_k - \min\{\phi_k(w) : w \in \Gamma^k, \|w\| \leq \delta^k\}$$
$$\geq \begin{cases} c_1 \|g_k\| \min\{\min\{\delta^k, c_2\|g_k\|\}, \|g_k\|/\|B_k\|\}, & \text{if } B_k \neq 0; \\ c_1 \|g_k\| \min\{\delta^k, c_2 \|g_k\|\}, & \text{if } B_k = 0. \end{cases} \tag{3.1}$$

A6. There exists a positive scalar c_3, such that

$$\|s_k\| \leq c_3 \|g_k\|, \qquad k = 1, 2, \ldots \tag{3.2}$$

A7. There exists a positive scalar σ_1, such that

$$\|B_k\| \leq \sigma_1, \qquad k = 1, 2, \dots \tag{3.3}$$

A8. There exists a positive scalar c_4, such that

$$f_k - \phi_k(s_k) \geq c_4(-\lambda_1(B_k))(\delta^k)^2 \tag{3.4}$$

Assumptions A4, A5, A6 and A7 yield global convergence to critical points. Assumption A5 is a suffucient decrease condition. Assumption A6 is a particular condition for NTRCP method. We think that it is reasonable since it holds for the UTRCP method. Assumption A8 ensure convergence to critical points satisfying the second order necessary condition for a minimum.

First we will give some Lemmas. Following [1], we get the following lemma.

Lemma 3.1. *Let Assumptions* A2 *and* A3 *hold, and* δ^k *be determined by Algorithm* 2.2, *then the problem*

$$\min\{\phi_k(w): w \in \Gamma^k, \|w\| \leq \delta^k\}$$

has a unique solution $s_k \in \Gamma^k$, *which is defined by*

$$\|s_k\| = \sup\{\|\gamma_{g_k B_k}(t)\|: \|\gamma_{g_k B_k}(t)\| \leq \delta^k\} \tag{3.5}$$

Moreover, $\|s_k\| = \delta^k$.

Furthermore, if A5 *holds, then*

$$f_k - \phi_k(s_k) \geq \begin{cases} c_1 \|g_k\| \min\{\min\{\delta^k, c_2\|g_k\|\}, \|g_k\|/\|B_k\|\}, & \text{if } B_k \neq 0; \\ c_1 \|g_k\| \min\{\delta^k, c_2\|g_k\|\}, & \text{if } B_k = 0. \end{cases} \tag{3.6}$$

Lemma 3.2. *Let* $\Omega_0 = \{x: f(x) \leq f(x_0)\}$, *and assume that* A5 *holds,* $\{x_k\}$ *is generated by Algorithm* 2.2, *then* $\{x_k\} \subset \Omega_0$.

Proof. We prove the lemma by induction. Assume that $x_k \in \Omega_0$ when $k \leq m$, let $k = m + 1$,

(1) if $M \neq 0$, then we have

$$f_{m+1} \leq f_{l(m)} - \gamma\delta^m\|g_m\| \tag{3.7}$$

(2) else if $M = 0$, then we have either

$$f_m - f_{m+1} \geq \mu c_1 \|g_m\| \min\{\min\{\delta^m, c_2\|g_m\|\}, \|g_m\|/\|B_m\|\} \tag{3.8}$$

or

$$f_m - f_{m+1} \geq \mu c_1 \|g_m\| \min\{\delta^m, c_2 \|g_m\|\} \tag{3.9}$$

thus $x_{m+1} \in \Omega_0$ since $x_m \in \Omega_0$ and $l(m) \leq m$. Q.E.D.

Lemma 3.3. *Assume that* A5 *holds and* Ω_0 *is compact, then* $\{f_{l(k)}\}$ *is nonincreasing and convergent, where* $\{f_{l(k)}\}$ *is determined by* (2.6.).

Proof. First we prove that $\{f_{l(k)}\}$ *is nonincreasing.*

$$f(x_{l(k+1)}) = \max_{0\le j\le m(k+1)} [f_{k+1-j}] \le \max_{0\le j\le m(k)+1} [f_{k+1-j}] = \max\{f(x_{l(k)}), f_{k+1}\}$$

from (3.7), (3.8) and (3.9), we have

$$f(x_{k+1}) \le f(x_{l(k)})$$

thus

$$f(x_{l(k+1)}) \le f(x_{l(k)})$$

From Lemma 3.2, since Ω_0 is compact, so that $\{f_{l(k)}\}$ is convergent. Q.E.D.

Lemma 3.4. *Let $f(x)$ be twice continuously differentiable and $\Omega_0 = \{x: f(x) \le f(x_0)\}$ be compact. Assume that* A1 *through* A7 *hold. Then*

$$\lim_{k\to\infty} \|s_k\| \, \|g_k\| = 0 \tag{3.10}$$

and

$$\lim_{k\to\infty} \|s_k\| = 0 \tag{3.11}$$

Proof. According to (3.7), (3.8) and (3.9) respectively, we have

$$f(x_{l(k)}) \le f(x_{l(k)-1}) - \mu c_1 \|g_{l(k)-1}\| \min\{\min\{\delta^{l(k)-1}, c_2\|g_{l(k)-1}\|\}, \|g_{l(k)-1}\|/\|B_{l(k)-1}\|\} \tag{3.12}$$

or

$$f(x_{l(k)}) \le f(x_{l(k)-1}) - \mu c_1 \|g_{l(k)-1}\| \min\{\delta^{l(k)-1}, c_2\|g_{l(k)-1}\|\} \tag{3.13}$$

or

$$f(x_{l(k)}) \le f(x_{l(k)-1}) - \gamma\delta^{l(k)-1}\|g_{l(k)-1}\| \tag{3.14}$$

for all $k > M$. It follows from (3.12), (3.13) and (3.14) by Lemma 3.3 that

$$\lim_{k\to\infty} \delta^{l(k)-1}\|g_{l(k)-1}\| = 0 \tag{3.15}$$

thus

$$\lim_{k\to\infty} \|s_{l(k)-1}\| \, \|g_{l(k)-1}\| = 0 \tag{3.16}$$

and

$$\lim_{k\to\infty} \|s_{l(k)-1}\| = 0 \tag{3.17}$$

since $\|s_{l(k)-1}\| = \delta^{l(k)-1}$ and A6 holds.

Now we prove that $\lim_{k\to\infty} \|s_k\| \, \|g_k\| = 0$, let

$$\hat{l}(k) = l(k + M + 2) \ge k + 2$$

First we show, by induction, that for any given $j \ge 1$,

$$\lim_{k\to\infty} \delta^{\hat{l}(k)-j}\|g_{\hat{l}(k)-j}\| = 0 \tag{3.18}$$

and

$$\lim_{k\to\infty} f(x_{\hat{l}(k)-j}) = \lim_{k\to\infty} f(x_{l(k)}) \tag{3.19}$$

If $j \geq 1$, from (3.15), it is obvious that (3.18) is true. From (3.17), we know that $\|x_{\hat{l}(k)} - x_{\hat{l}(k)-1}\| \to 0$ is true. So that (3.19) holds, since $f(x)$ is uniformly continuous on Ω_0.

Assume that (3.18) and (3.19) hold for a given j. Then we get

$$f(x_{\hat{l}(k)-j}) \leq f(x_{l(\hat{l}(k)-(j+1))}) - \mu c_1 \|g_{\hat{l}(k)-(j+1)}\| \min\{\min\{\delta^{\hat{l}(k)-(j+1)}, c_2\|g_{\hat{l}(k)-(j+1)}\|\}, \|g_{\hat{l}(k)-(j+1)}\|/\|B_{\hat{l}(k)-(j+1)}\|\} \tag{3.20}$$

or

$$f(x_{\hat{l}(k)-j}) \leq f(x_{l(\hat{l}(k)-(j+1))}) - \mu c_1 \|g_{\hat{l}(k)-(j+1)}\| \min\{\delta^{\hat{l}(k)-(j+1)}, c_2\|g_{\hat{l}(k)-(j+1)}\|\} \tag{3.21}$$

or

$$f(x_{\hat{l}(k)-j}) \leq f(x_{l(\hat{l}(k)-(j+1))}) - \gamma\delta^{\hat{l}(k)-(j+1)}\|g_{\hat{l}(k)-(j+1)}\| \tag{3.22}$$

By (3.19), we have

$$\lim_{k\to\infty} \delta^{\hat{l}(k)-(j+1)}\|g_{\hat{l}(k)-(j+1)}\| = 0 \tag{3.23}$$

thus

$$\lim_{k\to\infty} \|s_{\hat{l}(k)-(j+1)}\| \, \|g_{\hat{l}(k)-(j+1)}\| = 0 \tag{3.24}$$

and

$$\lim_{k\to\infty} \|s_{\hat{l}(k)-(j+1)}\| = 0 \tag{3.25}$$

Moreover this implies that

$$\|x_{\hat{l}(k)-j} - x_{\hat{l}(k)-(j+1)}\| \to 0 \tag{3.26}$$

So that

$$\lim_{k\to\infty} f(x_{\hat{l}(k)-(j+1)}) = \lim_{k\to\infty} f(x_{\hat{l}(k)-j}) = \lim_{k\to\infty} f(x_{l(k)}) \tag{3.27}$$

thus we conclude that (3.18) and (3.19) hold for any given $j \geq 0$.

Now for any k,

$$x_{k+1} = x_{\hat{l}(k)} - \sum_{j=1}^{\hat{l}(k)-(k+1)} s_{\hat{l}(k)-j} \tag{3.28}$$

From (3.18) and (3.28), since $\hat{l}(k) - (k+1) = l(k+M+2) - (k+1) \leq M+1$, (3.27) implies

$$\lim_{k\to\infty} \|x_{k+1} - x_{\hat{l}(k)}\| = 0 \tag{3.29}$$

By continuity of f, we get

$$\lim_{k\to\infty} f(x_k) = \lim_{k\to\infty} f(x_{\hat{l}(k)}) = \lim_{k\to\infty} f(x_{l(k)}) \tag{3.30}$$

From (3.12), (3.13) and (3.14) and using (3.30), we obtain

$$\lim_{k\to\infty} \delta^k \|g_k\| = 0 \tag{3.31}$$

So that (3.10) and (3.11) are true. Q.E.D.

Lemma 3.5. *Let the hypotheses of Lemma* 3.4 *hold, and assume that*

$$\lim_{k\to\infty} \|s_k\| = 0 \qquad \text{implies } \lim_{k\to\infty} \|\nabla f(x_k) - g_k\| = 0 \tag{3.32}$$

then

$$\liminf_{k\to\infty} \|g_k\| = 0 \tag{3.33}$$

Proof. We prove that $\{\|g_k\|\}$ is not bounded away from zero by contradiction. Assume that there is an $\varepsilon_0 > 0$ and a $K > M$, such that

$$\|g_k\| \geq \varepsilon_0 \tag{3.34}$$

for all $k > K$. From the proof of Lemma 3.4, we get

$$\lim_{k\to\infty} \delta^k = 0 \tag{3.35}$$

Thus when k is large enough, from Lemma 3.1, we have

$$pred_k = f_k - \phi_k(s_k) \geq c_1 \varepsilon_0 \delta^k \tag{3.36}$$

On the otherhand,

$$\begin{aligned} |ared_k - pred_k| &= |-f_{k+1} + \phi_k(s_k)| \\ &= |-(\nabla f(x_k + \xi_k s_k))^T s_k + g_k^T s_k + \tfrac{1}{2} s_k^T B_k s_k| \\ &= |-[\nabla f(x_k) - g_k]^T s_k + [\nabla f(x_k) - \nabla f(x_k + \xi_k s_k)]^T s_k + \tfrac{1}{2} s_k^T B_k s_k| \\ &\leq \|\nabla f(x_k) - g_k\| \|s_k\| + \|\nabla f(x_k) - \nabla f(x_k + \xi_k s_k)\| \|s_k\| + \tfrac{1}{2}\sigma_1 \|s_k\|^2 \end{aligned} \tag{3.37}$$

where $\xi_k \in (0, 1)$. From the assumptions of lemma, we also have

$$\lim_{k\to\infty} \|\nabla f(x_k) - g_k\| = 0 \tag{3.38}$$

and

$$\lim_{k\to\infty} \|\nabla f(x_k) - \nabla f(x_k + \xi_k s_k)\| = 0 \tag{3.39}$$

Thus when k is large enough, we have

$$|\rho_k - 1| \leq [\|\nabla f(x_k) - g_k\| + \|\nabla f(x_k) - \nabla f(x_k + \xi_k s_k)\| + \sigma_1 \|s_k\|/2]/(c_1 \varepsilon_0) \tag{3.40}$$

Therefore

$$\lim_{k\to\infty} \rho_k = 1 \tag{3.41}$$

So according to the structure of Algorithm 2.2, there exists a $k_0 > 0$, such that

$$\Delta^{k+1} \geq \min\{\Delta^{k_0}, \bar{\Delta}\} \tag{3.42}$$

for all $k \geq k_0$. On the otherhand,

$$\delta^k = \min\{\Delta^k_{\max}, \Delta^k\}$$

By A4 either $\Delta^k_{\max} = +\infty$ if $B_k = 0$ or $\Delta^k_{\max} \geq \|g_k\| \|B_k\|^{-1} \geq \varepsilon_0/\sigma_1$, for all $k > K$. Let $k_1 = \max\{k_0, K\}$, for all $k > k_1$, we have

$$\delta^k = \min\{\Delta^k_{\max}, \Delta^k\} \geq \min\{\varepsilon_0/\sigma_1, \Delta^k\} \geq \min\{\varepsilon_0/\sigma_1, \Delta^{k_0}, \bar{\Delta}\} \tag{3.43}$$

This contradicts to (3.35). So that

$$\liminf_{k\to\infty} \|g_k\| = 0 \qquad \text{Q.E.D.}$$

We will discuss the first and second order necessary conditions and the convergence of the point sequence $\{x_k\}$ by the following theorems.

Theorem 3.1. *Let the hypotheses of Lemma* 3.5 *hold, then*

$$\lim_{k\to\infty} \|g_k\| = 0 \tag{3.44}$$

Theorem 3.2. *Let the hypotheses of Lemma* 3.4 *hold,* $g_k = \nabla f(x_k)$, $B_k = \nabla^2 f(x_k)$, *and* $\nabla^2 f(x)$ *be uniformly continuous on* Ω_0. *The sequence* $\{x_k\}$ *is generated by Algorithm* 2.2.

(a) *Let* x^* *be a limit point of the sequence* $\{x_k\}$, *then* $\nabla f(x^*) = 0$. *If* A8 *holds and* $\nabla^2 f(x^*)$ *is nonsingular, then* $\nabla^2 f(x^*)$ *is positive definite and the whole sequence converges to* x^*.

(b) *If* x^* *is an isolated limit point of the sequence* $\{x^*\}$ *and* A8 *holds, then* $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ *is positive semi-definite.*

Remark 3.1. Theorems 3.1 and 3.2 show that any limit point of the sequence $\{x_k\}$, generated by Algorithm 2.2, satisfies the first and second order necessary conditions for the problem (P).

Theorem 3.3. *Let the hypotheses of Lemma* 3.4 *be satisfied, and* $g_k = \nabla f(x_k)$. *The sequence* $\{x_k\}$ *is generated by Algorithm* 2.2. *If the number of the stationary points of* $f(x)$ *in* Ω_0 *is finite, then the sequence* $\{x_k\}$ *converges.*

Theorem 3.4. *Let the hypotheses of Theorem* 3.1 *hold, then no any limit point of* $\{x_k\}$ *is a local maximum of* $f(x)$.

The proofs of Theorems 3.1, 3.2, 3.3 and 3.4 are standard and thus omitted.

4. Numerical Experiments

In this section, we give the numerical results of NTRCP algorithm, and compare with that of UTRCP algorithm. We first give a special curvilinear path: the path of conjugate gradients. By applying a truncated conjugate direction algorithm to the quadratic model (2.1), we generate a sequence of conjugate directions $p_k^1, \ldots, p_k^{m+1}$ ($p^1 = -g_k$) which satisfy

$$p_k^{i+1} = -d_k^{i+1} + \beta_k^i p_k^i, \qquad i = 1, \ldots, m \tag{4.1}$$

$$r_k^{i+1} = r_k^i + \alpha_k^i p_k^i, \qquad i = 1, \ldots, m \tag{4.2}$$

$$(p_k^i)^T B_k p_k^i > 0, \qquad i = 1, \ldots, m \tag{4.3}$$

where

$$d_k^1 = g_k \tag{4.4}$$

$$d_k^{i+1} = B_k r_k^{i+1} + g_k \tag{4.5}$$

$$\beta_k^i = (d_k^{i+1})^T B_k p_k^i / (p_k^i)^T B_k p_k^i > 0 \tag{4.6}$$

and

$$\alpha_k^i = -(d_k^i)^T p_k^i / (p_k^i)^T B_k p_k^i > 0 \tag{4.7}$$

The procedure stops either because $d_k^{i+1} = 0$ or $(p_k^{i+1})^T B_k p_k^{i+1} \leq 0$ [18].

Define the path of conjugate gradients by the following formula

$$\gamma_{g_k B_k}(t) = \sum_{i=1}^{m} a_k^i(t) p_k^i - a_k^{m+1}(t) p_k^{m+1} \tag{4.8}$$

and

$$a_k^i(t) = \min\left\{\alpha_k^i, \max\left\{0, t - \sum_{j=1}^{i-1} \alpha_k^j\right\}\right\} \tag{4.9}$$

where $\sum_{j=1}^{0} \alpha_k^j = 0$.

We can prove that the path of conjugate gradients defined by (4.8) satisfies Assumptions A1 to A6.

Given a special path, we are in the position to compare the UTRCP method (Algorithm 2.1) with the NTRCP method (Algorithm 2.2). We only need to compare the results for the two cases: $M = 0$ or $M > 0$ in Algorithm 2.2 according to Remark 2.2. In addition, we assume that

$$g_k = \nabla f(x_k), \qquad B_k = \nabla^2 f(x_k)$$

and the parameters are chosen as follows

$$\mu = 0.7, \qquad \eta = 0.1, \qquad \gamma_1 = 0.75, \qquad \gamma_2 = 2, \qquad \gamma = 10^{-3}, \qquad \varepsilon = 10^{-5}$$

All tested problems are quoted from the related literature. The numerical performance of the algorithm for different M value can be compared by the following numbers:

n_g—the number of gradient evaluations.
n_f—the number of function evaluations.

The computations have been performed in double precision arithmetic on SGI 4D25.

Problem 4.1. Maratos Function

$$f(x) = x_1 + \tau(x_1^2 + x_2^2 - 1)^2$$
$$x_0 = (1., 0.1)^T$$
$$x^* = (-1, 0)^T$$
$$f(x^*) = -1$$

The Maratos effect can very seriously affect the performance of UTRCP algorithm on this problem, but NTRCP algorithm is also very good for this problem. The results are reported in Table 4.1.

Table 4.1. Results for problem 4.1

τ	$\bar{\Delta}$	$M = 0$(UTRCP)	$M = 10$(NTRCP)	$M = 20$(NTRCP)
10^2	1	$n_g = 41, n_f = 74$	$n_g = 31, n_f = 48$	$n_g = 29, n_f = 41$
10^3	1	$n_g = 80, n_f = 151$	$n_g = 47, n_f = 58$	$n_g = 47, n_f = 52$
10^4	1	$n_g = 159, n_f = 318$	$n_g = 47, n_f = 70$	$n_g = 47, n_f = 58$
10^5	1	$n_g = 317, n_f = 636$	$n_g = 102, n_f = 175$	$n_g = 101, n_f = 119$
10^6	1	$n_g = 705, n_f = 1432$	$n_g = 151, n_f = 263$	$n_g = 86, n_f = 100$

Problem 4.2. Scaled Sine-valley Function

$$f(x) = c[x_2 - \sin(x_1)]^2 + 0.25x_1^2$$
$$x_0 = (\tfrac{3}{2}\pi, -1)^T$$
$$x^* = (0, 0)^T$$
$$f(x^*) = 0$$

The scaled sine-valley function has a valley along the curve $x_2 = \sin(x_1)$. The results are reported in Table 4.2.

Table 4.2. Results for problem 4.2

c	$\bar{\Delta}$	$M = 0$(UTRCP)	$M = 10$(NTRCP)	$M = 20$(NTRCP)
10^2	1	$n_g = 22, n_f = 35$	$n_g = 20, n_f = 30$	$n_g = 15, n_f = 20$
10^3	1	$n_g = 44, n_f = 84$	$n_g = 31, n_f = 53$	$n_g = 28, n_f = 47$
10^4	1	$n_g = 90, n_f = 162$	$n_g = 60, n_f = 113$	$n_g = 51, n_f = 97$
10^5	1	$n_g = 176, n_f = 350$	$n_g = 128, n_f = 252$	$n_g = 114, n_f = 227$
10^6	1	$n_g = 375, n_f = 750$	$n_g = 265, n_f = 529$	$n_g = 273, n_f = 546$

Problem 4.3. Scaled Exponential-valley Function

$$f(x) = c[x_2 - \exp(x_1)]^2 + 0.25x_1^2$$

$$x_0 = (1, -1)^T$$

$$x^* = (0, 1)^T$$

$$f(x^*) = 0$$

The scaled exponential-valley function has a valley along the curve $x_2 = \exp(x_1)$. The results are reported in Table 4.3.

Table 4.3. Results for problem 4.3

c	$\bar{\Delta}$	$M = 0$(UTRCP)	$M = 10$(NTRCP)
10^2	1	$n_g = 11, n_f = 15$	$n_g = 6, n_f = 7$
10^3	1	$n_g = 16, n_f = 29$	$n_g = 5, n_f = 6$
10^4	1	$n_g = 27, n_f = 54$	$n_g = 5, n_f = 6$
10^5	1	$n_g = 51, n_f = 105$	$n_g = 5, n_f = 6$
10^6	1	$n_g = 100, n_f = 210$	$n_g = 5, n_f = 6$
10^7	1	$n_g = 207, n_f = 425$	$n_g = 5, n_f = 6$
10^8	1	$n_g = 504, n_f = 1024$	$n_g = 5, n_f = 6$
10^2	2	$n_g = 15, n_f = 27$	$n_g = 6, n_f = 7$
10^3	2	$n_g = 26, n_f = 50$	$n_g = 5, n_f = 6$
10^4	2	$n_g = 54, n_f = 105$	$n_g = 4, n_f = 5$
10^5	2	$n_g = 107, n_f = 220$	$n_g = 4, n_f = 5$
10^6	2	$n_g = 209, n_f = 431$	$n_g = 4, n_f = 5$
10^7	2	$n_g = 503, n_f = 1036$	$n_g = 4, n_f = 5$
10^8	2	$n_g > 1000, n_f > 2000$	$n_g = 4, n_f = 5$

Problem 4.4. Scaled Rosenbrock Function

$$f(x) = c(x_2 - x_1^2)^2 + (1 - x_1)^2$$

$$x_0 = (-1.2, 1)^T$$

$$x^* = (1, 1)^T$$

$$f(x^*) = 0$$

Table 4.4 gives the results. They show that the usefulness of the NTRCP algorithm for the ill-conditioned problem.

Table 4.4. Results for problem 4.4

c	$\bar{\Delta}$	$M = 0$(UTRCP)	$M = 10$(NTRCP)
10^2	1	$n_g = 22, n_f = 38$	$n_g = 11, n_f = 13$
10^3	1	$n_g = 40, n_f = 74$	$n_g = 15, n_f = 16$
10^4	1	$n_g = 84, n_f = 162$	$n_g = 13, n_f = 16$
10^5	1	$n_g = 173, n_f = 348$	$n_g = 13, n_f = 16$
10^6	1	$n_g = 347, n_f = 703$	$n_g = 11, n_f = 14$

Problem 4.5. Scaled Power-valley Function

$$f(x) = c(x_2 - x_1^p)^2 + (1 - x_1)^2, \qquad p \geq 2$$

$$x_0 = (-1.2, 1)^T$$

$$x^* = (1, 1)^T$$

$$f(x^*) = 0$$

The scaled power-valley function has a valley along the curve $x_2 = x_1^p$. The results are reported in Table 4.5.

Table 4.5. Results for problem 4.5

c	p	$\bar{\Delta}$	$M = 0$(UTRCP)	$M = 10$(NTRCP)
10^2	2	1	$n_g = 22, n_f = 38$	$n_g = 11, n_f = 13$
10^2	3	1	$n_g = 25, n_f = 39$	$n_g = 11, n_f = 12$
10^2	4	1	$n_g = 36, n_f = 66$	$n_g = 13, n_f = 14$
10^2	5	1	$n_g = 37, n_f = 65$	$n_g = 13, n_f = 15$
10^2	6	1	$n_g = 45, n_f = 82$	$n_g = 17, n_f = 18$
10^4	2	1	$n_g = 84, n_f = 162$	$n_g = 13, n_f = 16$
10^4	3	1	$n_g = 101, n_f = 204$	$n_g = 10, n_f = 11$
10^4	4	1	$n_g = 144, n_f = 295$	$n_g = 14, n_f = 15$
10^4	5	1	$n_g = 157, n_f = 320$	$n_g = 16, n_f = 17$
10^4	6	1	$n_g = 185, n_f = 380$	$n_g = 14, n_f = 15$

5. Concluding Remarks

The general nonmonotone trust region algorithm described in this paper is able to handled a great number of curvilinear paths. Its convergence properties similar to those for usual trust region method with curvilinear path have been proved. It provides a unified framework of studying convergence properties. The numerical results reported in Section 4 show that the NTRCP method can allow a considerable computational saving, especially for the problem with narrow curving valleys, and can prevent occurrence of the Maratos effect.

Acknowledgement

The authors are grateful to Professor Deng Naiyang for useful discussions on this topic.

References

[1] Bulteau, J. P., Vial, J. P.: Curvilinear path and trust region in unconstrained optimization: a convergence analysis. Math. Programming Study *30*, 82–101 (1987).
[2] Chamberlain, R. M., Powell, M. J. D., Lemarechal, C., Pedersen, H. C.: The watchdog technique for forcing convergence in algorithms for constrained optimization. Math. Programming Study *26*, 1–17 (1986).
[3] Dennis, J. E., Jr., Mei, H. H. W.: Two new unconstrained optimization algorithms which use function and gradient values. JOTA *28*, 453–482 (1979).
[4] Dennis, J. E., Jr., Schnabel, R. B.: Numerical methods for unconstrained optimization and nonlinear equations. Englewood Cliffs, New York: Prentice-Hall 1983.
[5] Gan, Z., Zhou, F., Xiao, Y.: Study on the algorithm for multidimensional scaling. J. of Tsinghua Univ. *31*, 6, 20–27 (1991).
[6] Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's methods. SIAM J. Numer. Anal. *23*, 707–716 (1986).
[7] Grippo, L., Lampariello, F., Lucidi, S.: A truncated Newton method with nonmonotone line search for unconstrained optimization. JOTA *60*, 401–419 (1989).
[8] Grippo, L., Lampariello, F., Lucidi, S.: A quasi-discrete Newton algorithm with a nonmonotone stabilization technique. JOTA *64*, 495–510 (1990).
[9] Grippo, L., Lampariello, F., Lucidi S.: A class of nonmonotone stabilization methods in unconstrained optimization, IASI-CNR, Report No. 290, 1990.
[10] Moré, J. J.: Recent development in algorithms and software for trust region methods, pp. 258–287. In: Bachem, A., Grotschel, M., Kortz, B. (eds.) Mathematical Programming: the state of the art. Berlin, Heidelberg, New York, Tokyo: Springer 1983.
[11] Moré, J. J., Sorensen, D. C.: Computing a trust region step. SIAM J. Sci. Statist. Comput. *4*, 553–572 (1983).
[12] Panier, E. R., Tits, A. L.: Avoiding the Maratos effect by means of a nonmonotone linear search: I. General constrained problems, System Research Center, University of Maryland, Technical Report SRC-TR-89-41, 1989.
[13] Powell, M. J. D.: A hybrid method for nonlinear equations, pp. 87–114. Babinowitz, P., (ed.) Numerical methods for nonlinear algebraic equations. London: Gordon and Breach Press 1970.
[14] Powell, M. J. D.: Convergence properties of a class of minimization algorithms, pp. 1–29. In: Mangasarian, O. L., Meyer, R. R., Robinson, S. M. (eds.) Nonlinear programming 2. New York: Academic Press 1975.
[15] Shultz, G. A., Schnabel, R. B., Byrd, R. H.: A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. SIAM J. Numer. Anal. *22*, 47–67 (1985).
[16] Sorensen, D. C.: Trust region methods for unconstrained minimization, pp. 29–38. In: Powell, M. J. D. (ed.) Nonlinear optimization 1981. London: Academic Press 1982.

[17] Sorensen, D. C.: Newton's method with a model trust region modification. SIAM J. Numer. Anal. *19*, 409–426 (1982).
[18] Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal. *20*, 626–637 (1983).

Yi Xiao
Department of Applied Mathematics
Beijing Polytechnic University
Beijing 100022, P.R. China

Fangjun Zhou
Department of Applied Mathematics
Tsinghua University
Beijing 100084, P.R. China