IMA Journal of Numerical Analysis (1984) **4**, 327–335

An Example of Only Linear Convergence of Trust Region Algorithms for Non-smooth Optimization†

Y. YUAN

Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB3 9EW

[Received 14 July 1983]

Most superlinear convergence results about trust region algorithms for non-smooth optimization are dependent on the inactivity of trust region restrictions. An example is constructed to show that it is possible that at every iteration the trust region bound is active and the rate of convergence is only linear, though strict complementarity and second order sufficiency conditions are satisfied.

1. Introduction

IT IS SHOWN in this paper that some trust region algorithms for non-smooth optimization may converge only linearly, though many strong conditions are satisfied.

Trust region algorithms are iterative. The problem we want to solve has the following form

$$\min h(f(x)), \tag{1.1}$$

where $f(x)$ from $\mathbb{R}^n$ to $\mathbb{R}^m$ is a continuously differentiable function, and $h(.)$ from $\mathbb{R}^m$ to $\mathbb{R}$ is a convex function. This form of objective function occurs frequently in discrete approximation and data fitting calculations.

At the beginning of the kth iteration, an estimate x_k of the solution of (1.1), a trust region bound $\rho_k > 0$, and a $n \times n$ symmetric matrix B_k are available. We calculate d_k by solving the following subproblem

$$\min \phi_k(d) \equiv h(f(x_k) + \nabla^T f(x_k)d) + \tfrac{1}{2} d^T B_k d \tag{1.2}$$

subject to

$$\|d\| \leqslant \rho_k, \tag{1.3}$$

where $\|.\|$ is a given norm. Then the "predicted reduction" $\phi_k(0) - \phi_k(d_k)$ and the "actual reduction" $h(f(x_k)) - h(f(x_k + d_k))$ are compared, to decide whether to set $x_{k+1} = x_k + d_k$ or $x_{k+1} = x_k$. The choice of ρ_{k+1} and B_{k+1} also depend on this comparison, and the calculation of B_{k+1} usually requires some first derivatives at x_k and x_{k+1}. Many strategies are known, and each leads to a specific algorithm. Here we give a general algorithm, and it turns out that many known methods are special cases of our algorithm.

† Presented at the 1983 Dundee Conference on Numerical Analysis.

0272–4979/84/030327+09 $03.00/0

Denote the ratio between the actual reduction and the predicted reduction by r_k, that is,

$$r_k = \frac{h(f(x_k)) - h(f(x_k + d_k))}{\phi_k(0) - \phi_k(d_k)}. \tag{1.4}$$

Let c_i $(i = 1, 2, 3)$ be any constants from the open interval $(0, 1)$ satisfying $c_1 \leqslant c_2 \leqslant c_3$. Then x_{k+1} is defined as follows

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k \geqslant c_1 \\ x_k & \text{if } r_k < c_1 \end{cases} \tag{1.5}$$

and ρ_{k+1} is chosen in some way that satisfies the following conditions.

(A) If $r_k < c_2$ then ρ_{k+1} is bounded by $c_4 \rho_k$ where c_4 is a constant from $(0, 1)$;
(B) if $r_k \geqslant c_3$ then ρ_{k+1} is not less than $\|d_k\|$;
(C) ρ_{k+1} is not greater than $M\rho_k$ for some constant M.

We permit the replacement of "$\geqslant$" by "$>$" and "$<$" by "$\leqslant$" in (1.5), and similar changes in (A) and (B).

The following more restrictive conditions for the revision of the trust region bound, pointed out by the referee, are usual:

(D) $\rho_{k+1} = \rho_k$ if $c_2 \leqslant r_k < c_3$, and
(E) $\rho_{k+1} = M\rho_k (M > 1)$ if $r_k \geqslant c_3$ and if $\rho_k = \|d_k\|$.

Furthermore, in addition to (A), in practice it is also usual to set

(F) $\rho_{k+1} =$ either $c_4 \rho_k$ or $c_4 \|d_k\|$ if $r_k < c_2$.

The class of algorithms stated above is very general and many known trust region methods are special cases of it, for example, see Fletcher (1980), Moré (1982), Powell (1983) and Yuan (1983). All these algorithms require that ρ_{k+1} does not exceed a fraction of ρ_k or a fraction of $\|d_k\|$, when $r_k < c_2$. We stress this property because it can cause slow convergence, and any other trust region methods sharing this property may also suffer slow convergence.

By constructing a minimax problem, we show that for non-smooth optimization some trust region algorithms may converge only linearly, though superlinear convergence would occur if the objective function (1.1) were continuously differentiable.

2. The Example

In this section, an example is constructed to show that the algorithm given in Section 1 may converge only linearly.

The problem we consider is a minimax calculation. In (1.1), we let $h(f) = \max\{f_1, f_2\}$, $n = m = 2$, and $f = (f_1, f_2)^T$ is defined by

$$\begin{aligned} f_1 &= 1 + v - u^2, \\ f_2 &= 1 - v + (1 + \varepsilon)u^2, \end{aligned} \tag{2.1}$$

where $\varepsilon > 0$ is a parameter to be defined, and where u and v are the components of x. The solution of (1.1) is $x^* = (u^*, v^*)^T = (0, 0)^T$. By expressing the minimax problem as a constrained optimization calculation (Overton, 1982), we find that the Lagrange multipliers of our problem at the solution are $\lambda_1^* = \lambda_2^* = \frac{1}{2}$. Strict complementarity and second order sufficiency conditions (Fletcher, 1981) hold, and we have

$$G_\infty^* = \sum_{i=1}^{2} \lambda_i^* \nabla^2 f_i(u^*, v^*) = \begin{bmatrix} \varepsilon & 0 \\ 0 & 0 \end{bmatrix}. \tag{2.2}$$

We let $B_k \equiv G_\infty^*$ for all k, and we let the norm in (1.3) be the infinity norm. The motivation of setting $B_k \equiv G_k^*$ is that, if $x_k \to x^*$, if $x_{k+1} = x_k + d_k$ for all sufficiently large k, where $d = d_k$ is a local minimum of (1.2), and if $\{B_k\}$ $(k = 1, 2, \ldots)$ are uniformly bounded, then the superlinear convergence of $\{x_k\}$ is equivalent to the limit

$$\lim_{k \to \infty} \frac{\|P_\infty^*(B_k - G_\infty^*)d_k\|}{\|d_k\|} = 0, \tag{2.3}$$

where P_∞^* is the orthogonal projection operator from $\mathbb{R}^n$ to $\{d;\ \nabla^T f_i(x^*)d = 0,\ i = 1, 2\}$ (see Powell, 1983, and Powell & Yuan, 1983).

What we will show is that the parameter ε and the initial information may be such that for all k $(u_k, v_k)^T$ lies on the curve $v = \theta u^2$, and $u_{k+1} = \alpha u_k$ for some constant $\alpha \in (0, 1)$, where θ is a constant satisfying $\theta < 1 + \varepsilon/2$. We consider the kth iteration, in the case when $v_k = \theta u_k^2$ and $\rho_k = (1-\alpha)u_k$.

Let us consider the subproblem (1.2), which becomes

$$\min \phi_k(d) \equiv \max\{1 - (1-\theta)u_k^2 - 2u_k d_1 + d_2,\ 1 + (1+\varepsilon-\theta)u_k^2 + 2(1+\varepsilon)u_k d_1 - d_2\} + \tfrac{1}{2}\varepsilon d_1^2, \tag{2.4}$$

subject to

$$\|d\|_\infty \leqslant \rho_k. \tag{2.5}$$

To solve (2.4)–(2.5), we consider the cases in which the first term or the second term achieves the max in (2.4) separately.

In the case that the second term in the bracket achieves the max, (2.4) reduces to the following problem:

$$\min \phi_k(d) = 1 + (1+\varepsilon-\theta)u_k^2 + 2(1+\varepsilon)u_k d_1 - d_2 + \tfrac{1}{2}\varepsilon d_1^2, \tag{2.6}$$

subject to $d \in H_1 \cap B(0, \rho_k)$, where

$$H_1 = \{(d_1, d_2)^T;\ (2+\varepsilon-2\theta)u_k^2 + 2(2+\varepsilon)u_k d_1 - 2d_2 \geqslant 0\}, \tag{2.7}$$
$$B(0, \rho_k) = \{d;\ \|d\|_\infty \leqslant \rho_k\}.$$

Further we assume $|u_k|$ is so small that

$$H_1 \cap B(0, \rho_k) = \left\{(d_1, d_2);\ |d_1| \leqslant \rho_k,\ -\rho_k \leqslant d_2 \leqslant \left(1 + \frac{\varepsilon}{2} - \theta\right)u_k^2 + (2+\varepsilon)u_k d_1\right\}. \tag{2.8}$$

From (2.6) ϕ_k decreases as d_2 increases for all d in $H_1 \cap B(0, \rho_k)$. By (2.8) it follows that $\phi_k(d)$ attains its minimum over $H_1 \cap B(0, \rho_k)$ only when

$$d_2 = \left(1 + \frac{\varepsilon}{2} - \theta\right) u_k^2 + (2+\varepsilon)u_k d_1.$$

In the other case, a similar analysis holds. Hence the solution of (2.4) and (2.5) lies on the line $H_1 \cap H_2$, where

$$H_2 = \{(d_1, d_2)^T; (2+\varepsilon-2\theta)u_k^2 + 2(2+\varepsilon)u_k d_1 - 2d_2 \leq 0\}. \tag{2.9}$$

Noticing that on $H_1 \cap H_2 \cap B(0, \rho_k)$ (2.6) still holds, we have that the solution of (2.4) and (2.5) is

$$(d_1, d_2)^T = (-(1-\alpha)u_k, (2+\varepsilon)u_k(\alpha u_k - \tfrac{1}{2}u_k) - \theta u_k^2)^T, \tag{2.10}$$

which is the intersection point of $H_1 \cap H_2$ and $\{d; d_1 = -\rho_k\}$. Therefore

$$x_k + d_k = (\alpha u_k, (2+\varepsilon)(\alpha - \tfrac{1}{2})u_k^2)^T. \tag{2.11}$$

Since we require to retain $x_k + d_k$ on the curve $v = \theta u^2$, we need the relation

$$(2+\varepsilon)(\alpha - \tfrac{1}{2}) = \theta\alpha^2, \tag{2.12}$$

which is independent of k.

The actual reduction and predicted reduction are as follows

$$h(f(x_k)) - h(f(x_k + d_k)) = (1+\varepsilon-\theta)(1-\alpha^2)u_k^2, \tag{2.13}$$

$$\phi_k(0) - \phi_k(d_k) = 2(1+\varepsilon)(1-\alpha)u_k^2 - \theta(1-\alpha^2)u_k^2 - \tfrac{1}{2}\varepsilon(1-\alpha)^2 u_k^2,$$

where the last line depends on Equation (2.12). Thus we have

$$r_k = \frac{(1+\varepsilon-\theta)(1+\alpha)}{2(1+\varepsilon) - \theta(1+\alpha) - \frac{1}{2}\varepsilon(1-\alpha)}. \tag{2.14}$$

From (2.12) and (2.14), elementary calculations show that

$$r_k = R(\varepsilon, \alpha) = \frac{2(1-\alpha)^2(1+\alpha) + \varepsilon(1-\alpha+2\alpha^3)}{2(1-\alpha) + \varepsilon(1-\alpha+\alpha^2+\alpha^3)}. \tag{2.15}$$

Since $R(\varepsilon, \alpha)$ is continuous for $\varepsilon > 0$, $\alpha \in (0, 1)$, and since

$$\lim_{\varepsilon, \alpha \to 0+} R(\varepsilon, \alpha) = 1, \qquad \lim_{\alpha \to 1-} R((1-\alpha)^2, \alpha) = 0, \tag{2.16}$$

for any given $0 \leq c_1 < c_2$ we can choose $\varepsilon > 0$, $\alpha \in (0, 1)$ properly such that $r_k \in (c_1, c_2)$. Thus $x_{k+1} = x_k + d_k$ and ρ_{k+1} satisfies (A). We let θ have the value $(2+\varepsilon)(\alpha - \frac{1}{2})/\alpha^2$ so that (2.12) holds. Therefore if $c_4 \geq \alpha$ ($c_4 = \alpha$ when the algorithm requires (F)), the algorithm sets

$$\rho_{k+1} = \alpha\rho_k = \alpha\|d_k\| = \alpha(1-\alpha)u_k = (1-\alpha)u_{k+1}, \tag{2.17}$$

and we have that

$$x_{k+1} = x_k + d_k = (u_{k+1}, \theta u_{k+1}^2)^T. \tag{2.18}$$

Comparing the above two equations with the conditions at the beginning of the kth iteration, it follows by induction that $x_1 = (u_1, v_1)^T$ and $\rho_1 > 0$ may be such that x_k is on the curve $v = \theta u^2$ and $u_{k+1} = \alpha u_k$ for all k. Thus for all large k $\|x_{k+1} - x^*\|_\infty = \alpha\|x_k - x^*\|_\infty$, which shows that linear convergence occurs. Thus we have the following result.

Result

For any algorithm which belongs to the class described in Section 1, if the algorithm only requires that $c_4 \in (0, 1)$ instead of giving a specific c_4, then for any given $0 \leqslant c_1 < c_2 < c_3 \leqslant 1$, there exists $c_4 \in (0, 1)$ such that by suitable choices of initial point and initial trust region bound, the algorithm may converge only linearly, though $B_k \equiv G^*_\infty$, and strict complementarity and second order sufficiency conditions are satisfied.

It is noted that if $c_4 < 1$ is given in advance, and if $\rho_{k+1} = c_4 \rho_k$ on every iteration for which $r_k < c_2$, the above example is not valid for all choices of c_1, c_2 and c_3. To ensure the validity of the example we require that $\alpha = c_4$. If

$$1 - c_4^2 = R(0, c_4) < c_2,$$

$$\frac{1 - c_4 + 2c_4^3}{1 - c_4 + c_4^2 + c_4^3} = R(\infty, c_4) > c_1, \tag{2.19}$$

then we can choose $\varepsilon > 0$ such that $R(\varepsilon, c_4) \in (c_1, c_2)$. Therefore if (2.19) is satisfied, we can choose ε so that linear convergence occurs. The second equation of (2.19) is usually satisfied since c_1 is usually less than a half and since the left side is greater than 0·851 for all $c_4 \in (0, 1)$. Thus for many algorithms if $1 - c_4^2 < c_2$ then linear convergence may occur.

However, (2.19) is not a necessary condition for linear convergence. For $c_2 = c_3 = 0.25$, $c_1 = 0$, $c_3 = 0{\cdot}75$ (Fletcher, 1981, p. 208), (2.19) is not satisfied, yet the following example shows linear convergence. Let f be defined by (2.1) as before, and let

$$\hat{x}_k = \left(\hat{u}_k, \frac{5}{9}\left(1 + \frac{\varepsilon}{2}\right)\hat{u}_k^2\right)^T, \qquad \hat{\rho}_k = \tfrac{1}{6}\hat{u}_k, \tag{2.20}$$

where $\hat{u}_k > 0$ is very small and $\varepsilon > 0$ is a parameter. Then we have

$$\hat{x}_k + \hat{d}_k = (\tfrac{5}{6}\hat{u}_k, \tfrac{1}{3}(2+\varepsilon)\hat{u}_k^2)^T, \qquad \hat{r}_k = \frac{30 + 26\varepsilon}{32 + 27\varepsilon}, \tag{2.21}$$

which gives $\hat{r}_k > \frac{15}{16} > c_3$, since $\varepsilon > 0$. Therefore,

$$\hat{x}_{k+1} = \hat{x}_k + \hat{d}_k, \qquad \hat{\rho}_{k+1} = 2\hat{\rho}_k = \tfrac{1}{3}\hat{u}_k, \tag{2.22}$$

since $M = 2$ (see Fletcher, 1981). Direct calculations show that

$$\hat{x}_{k+1} + \hat{d}_{k+1} = \left(\tfrac{1}{2}\hat{u}_k, \frac{5}{9}\left(1 + \frac{\varepsilon}{2}\right)\left(\frac{\hat{u}_k}{2}\right)^2\right)^T,$$
$$\hat{r}_{k+1} = \frac{-6+13\varepsilon}{2+17\varepsilon}. \tag{2.23}$$

Let $\varepsilon > \frac{6}{13}$ and sufficiently close to $\frac{6}{13}$ in order that $0 < \hat{r}_{k+1} < c_2$. Then the algorithm sets

$$\hat{x}_{k+2} = \left(\hat{u}_{k+2}, \frac{5}{9}\left(1 + \frac{\varepsilon}{2}\right)\hat{u}_{k+2}^2\right)^T,$$
$$\hat{\rho}_{k+2} = \tfrac{1}{4}\hat{\rho}_{k+1} = \tfrac{1}{12}\hat{u}_k = \tfrac{1}{6}\hat{u}_{k+2} \tag{2.24}$$

where $\hat{u}_{k+2} = \frac{1}{2}u_k$. By induction, $\{x_k\}$ converges only linearly, the step-bound is reduced once every two iterations, and without the assumption (2.19) linear convergence still occurs. We believe that for any given $0 < c_1 < c_2 < c_3 < 1$, $c_4 \in (0, 1)$, by modifying the given example, it is possible to demonstrate that an algorithm may converge only linearly.

3. A Range of Initial Points for Linear Convergence

In this section it is shown that for some algorithms there is a range of starting points which may cause linear convergence.

The problem we want to solve is that in the previous section. Assume $0 < c_1 < c_2 < \frac{2}{3}$, $c_3 > \frac{3}{4}$, and $c_4 \geq \frac{1}{2}$ [$c_4 = \frac{1}{2}$ if the algorithm requires (F)]. Let $\varepsilon > 0$ be very small so that

$$\frac{\frac{\varepsilon}{2} - \theta}{1+\varepsilon-\theta} < c_1,$$
$$\frac{\varepsilon}{2(1+\varepsilon)} < c_1, \tag{3.1}$$
$$\frac{3}{4}\left(\frac{1+\varepsilon-\frac{4}{3}\theta}{1+\frac{7}{8}\varepsilon-\theta}\right) \in (c_2, c_3),$$

hold for all $\theta \in [0, \frac{1}{4}]$. We consider the initial point $\bar{x}_1 = (\bar{u}_1, \theta\bar{u}_1^2)^T$, where $\theta \in [0, \frac{1}{4}]$, and where $\bar{u}_1 < \bar{\rho}_1$ is a small positive number. First it is noted that the solution of (2.4) is

$$d_k^* = \left(-u_k, -\left(1 + \frac{\varepsilon}{2} + \theta\right)u_k^2\right)^T \tag{3.2}$$

when the trust region bound is inactive. Hence from (3.2) and the fact that $\bar{u}_1 < \bar{\rho}_1$, we have that

$$\bar{d}_1 = \left(-\bar{u}_1,\ -\left(1+\frac{\varepsilon}{2}+\theta\right)\bar{u}_1^2\right)^T,$$
$$\bar{r}_1 = \frac{\frac{\varepsilon}{2}-\theta}{1+\varepsilon-\theta}. \tag{3.3}$$

Because it follows from (3.1) and (3.3) that $\bar{r}_1 < c_1$, after the first iteration we have that

$$\bar{x}_2 = \bar{x}_1 = (\bar{u}_1,\ \theta\bar{u}_1^2)^T,$$
$$\bar{\rho}_2 = \tfrac{1}{2}\|\bar{d}_1\| = \tfrac{1}{2}\bar{u}_1. \tag{3.4}$$

At the second iteration, the subproblem (1.2)–(1.3) gives

$$\bar{d}_2 = (-\tfrac{1}{2}\bar{u}_1,\ -\theta\bar{u}_1^2)^T,$$
$$\bar{r}_2 = \frac{3}{4}\left(\frac{1+\varepsilon-\frac{4}{3}\theta}{1+\frac{7}{8}\varepsilon-\theta}\right). \tag{3.5}$$

From (3.1) and (3.5), we find that $\bar{r}_2 \in (c_2, c_3)$, which implies

$$\bar{x}_3 = \bar{x}_2 + \bar{d}_2 = (\tfrac{1}{2}\bar{u}_1, 0)^T,$$
$$\bar{\rho} = \bar{\rho}_2. \tag{3.6}$$

Hence it follows that

$$\bar{d}_3 = \left(-\tfrac{1}{2}\bar{u}_1,\ -\left(1+\frac{\varepsilon}{2}\right)(\tfrac{1}{2}\bar{u}_1)^2\right)^T,$$
$$\bar{r}_3 = \frac{\varepsilon}{2(1+\varepsilon)}. \tag{3.7}$$

Consequently, since $\bar{r}_3 < c_1$, the algorithm sets

$$\bar{x}_4 = \bar{x}_3,$$
$$\bar{\rho}_4 = \tfrac{1}{2}\bar{\rho}_3 = \tfrac{1}{4}\bar{u}_1. \tag{3.8}$$

Thus we obtain

$$\bar{d}_4 = (-\tfrac{1}{4}\bar{u}_1, 0)^T,$$
$$\bar{x}_5 = \bar{x}_4 + \bar{d}_4 = (\tfrac{1}{4}\bar{u}_1, 0)^T. \tag{3.9}$$

It follows by induction that

$$\bar{x}_{2k+2} = \bar{x}_{2k+1} = ((\tfrac{1}{2})^k\bar{u}_1, 0)^T, \tag{3.10}$$

for all $k \geqslant 1$. Therefore linear convergence occurs, and we have the following result.

Result

For any algorithm which belongs to the class described in Section 1, if $0 < c_1 < c_2 < \frac{2}{3}$, $c_3 > \frac{3}{4}$, and $c_4 \geqslant \frac{1}{2}$ [$c_4 = \frac{1}{2}$ if the algorithm requires (F)], then for any given initial trust region bound, there exists a region such that any starting point in the region causes a linear rate of convergence, though $B_k \equiv G^*_\infty$, and strict complementarity and second order sufficiency conditions are satisfied.

As in the previous section, we also believe that for any given $0 < c_1 < c_2 < c_3 < 1$, $c_4 \in (0, 1)$, by modifying the example given above, it is possible to verify that the above result is valid.

4. Discussion

Though for some trust region algorithms the values of the parameters c_i $(i = 1, 2, 3, 4)$ are given (for example, Fletcher, 1980), and for some algorithms typical parameter values are suggested (Powell, 1983), many algorithms do not provide specific parameter values, and they allow many choices of these constants. Usually the conditions for the choices of the parameters are not more restrictive than $0 < c_1 < c_2 < c_3 < 1$, $c_4 \in (0, 1)$ (see Moré, 1982; Powell, 1975, 1983; and Yuan, 1983). Thus, for many trust region algorithms, suitable choices of the initial point and the initial step-bound may cause linear convergence for non-smooth problems.

From the example discussed in this paper, we see that the inactivity of trust region bounds plays an important role in the rate of convergence. In fact, for most trust region algorithms, the inactivity of trust region restrictions is a necessary condition for superlinear convergence, while for many specific algorithms, under certain other conditions, this condition is also sufficient for superlinear convergence (for example, Fletcher, 1980; Powell, 1983; Powell & Yuan, 1983). Specifically, for minimax problems, if $\{x_k\} \to x^*$, if at x^* strict complementarity and second order sufficiency conditions are satisfied, if $B_k \equiv G^*_k$, and if the step-bounds are inactive for all large k, then the convergence is superlinear (see Powell & Yuan, 1983). Our example satisfies all these conditions except the inactivity of the step-bounds, and the lack of this condition obviates the superlinear convergence.

It is noted that our example is not the same as the "Maratos effect", though both are caused by derivative discontinuities. In fact for the example in Section 2, if we let $\varepsilon = 1$, $\alpha = \frac{1}{2}$, $\theta = 0$, if we choose the initial point as $(u_1, 0)^T$ for small $u_1 > 0$, and if we remove the trust region constraint, then the solution is achieved within two iterations [the first iteration gives $x_2 = (0, -\frac{3}{2}u_1^2)^T$].

The given examples depend on the derivative discontinuities of the objective function, because in the smooth case we may have

$$f(x_k+d_k) = f(x_k) + \nabla^T f(x_k) d_k + \tfrac{1}{2} d_k^T B_k d_k + o(\|d_k\|^2) \tag{4.1}$$

when B_k is an approximation to the second derivative matrix. Thus r_k tends to 1 and the superlinear convergence follows (Powell, 1975). However, in the non-smooth case the relation

$$h(f(x_k+d_k)) = h(f(x_k) + \nabla^T f(x_k) d_k) + \tfrac{1}{2} d_k^T B_k d_k + o(\|d_k\|^2) \tag{4.2}$$

does not generally hold, even though $B_k \equiv G^*_\infty$. Therefore when $\{x_k\} \to x^*$ along a path such that $[h(f(x_k)) - h(f(x^*))]$ is $O(\|x_k - x^*\|^2)$, the error in (4.2) may cause the ratio r_k not to indicate that x_k is converging, which leads to a reduction in the step-bound, and consequently linear convergence may follow.

A possible way to force superlinear convergence is to consider second derivative information. Fletcher (1982) conjectures that his "second order correction" method can ensure superlinear convergence if some mild conditions are satisfied. Yet superlinear convergence still depends on the inactivity of the trust region bound, so it is desirable to investigate whether this assumption holds.

I am greatly indebted to my supervisor Professor M. J. D. Powell for his constant help and encouragement and for his studying the early versions of this paper. He made many important corrections and helpful suggestions which make the paper possible. I also wish to thank the referee for many good suggestions which led to improvements.

References

FLETCHER, R. 1980 A model algorithm for composite NDO problem. *Mathematical Programming Studies* 17 (1982), pp. 67–76. Amsterdam: North-Holland.

FLETCHER, R. 1981 *Practical Methods of Optimization*, Vol. 2, *Constrained Optimization*. New York: John Wiley.

FLETCHER, R. 1982 Second order corrections for nondifferentiable optimization. In *Numerical Analysis* (Proc. Dundee 1981), (Watson, G. A., Ed.). Springer-Verlag.

MORÉ, J. J. 1982 Recent developments in algorithms and software for trust region methods. Report ANL/MCS-TM2, Argonne National Laboratory, Illinois.

OVERTON, M. L. 1982 Algorithms for nonlinear l_1 and l_∞ fitting. In *Nonlinear Optimization* 1981, (Powell, M. J. D., Ed.). London: Academic Press.

POWELL, M. J. D. 1975 Convergence properties of a class of minimization algorithms. In *Nonlinear Programming* 2, (Mangasarian, O. L., Meyer, R. R. & Robinson, S. M., Eds). New York: Academic Press.

POWELL, M. J. D. 1983 General algorithms for discrete nonlinear approximation calculation. Report DAMTP 1983/NA2, University of Cambridge.

POWELL, M. J. D. & YUAN, Y. 1983 Conditions for superlinear convergence in l_1 and l_∞ solution of overdetermined nonlinear equations. Report DAMTP 1983/NA8, University of Cambridge.

YUAN, Y. 1983 Some properties of trust region algorithms for nonsmooth optimization. Report DAMTP 1983/NA4, University of Cambridge.