Comput Optim Appl (2009) 43: 119–131
DOI 10.1007/s10589-007-9128-7

Truncated regularized Newton method for convex minimizations

Ying-Jie Li · Dong-Hui Li

Received: 13 October 2006 / Revised: 16 June 2007 / Published online: 9 November 2007

Abstract Recently, Li et al. (Comput. Optim. Appl. 26:131–147, 2004) proposed a regularized Newton method for convex minimization problems. The method retains local quadratic convergence property without requirement of the singularity of the Hessian. In this paper, we develop a truncated regularized Newton method and show its global convergence. We also establish a local quadratic convergence theorem for the truncated method under the same conditions as those in Li et al. (Comput. Optim. Appl. 26:131–147, 2004). At last, we test the proposed method through numerical experiments and compare its performance with the regularized Newton method. The results show that the truncated method outperforms the regularized Newton method.

Keywords Convex minimization · Regularized Newton method · Truncated conjugate gradient strategy

1 Introduction

Let $f : R^n :\to R$. Consider the unconstrained optimization problem

$$\min f(x), \quad x \in R^n. \tag{1.1}$$

Throughout the paper, we assume that the following conditions hold.

The work was supported by the 973 project granted 2004CB719402 and the NSF project of China granted 10471036.

Y.-J. Li · D.-H. Li (✉)
College of Mathematics and Econometrics, Hunan University, Changsha 410082, China
e-mail: madhli@sina.com

D.-H. Li
e-mail: dhli@hnu.cn

Y.-J. Li
e-mail: lyjat2001@yahoo.com.cn

Assumption A

(i) The solution set X of (1.1) is not empty.
(ii) The function f is convex and twice continuously differentiable. We use $g(x)$ and $G(x)$ to denote the gradient and Hessian of f at x, respectively.
(iii) The function $G(x)$ is Lipschitz continuous, i.e., there is a constant $L > 0$ such that

$$\|G(x) - G(y)\| \le L\|x - y\|, \quad \forall x, y \in R^n. \tag{1.2}$$

It is well-known that under the conditions of Assumption A, the solution set X of (1.1) coincides with the solution set of the nonlinear equation $g(x) = 0$. Moreover, $G(x)$ is positive semidefinite for all $x \in R^n$.

Newton's method is very efficient for solving (1.1) and nonlinear equations. An attractive property of Newton's method is its local superlinear/quadratic convergence property. However, if at some solution x^*, $G(x^*)$ is singular, the convergence rate of Newton's method may reduce to be linear. We call a solution x^* of problem (1.1) a singular solution if $G(x^*)$ is singular. Recently, there has been a growing interest in the study of Newton's method for solving (1.1) and nonlinear equations with singular solutions [1, 3, 5–7, 13–16]. These methods retain local quadratic convergence even if the problem has singular solutions.

In this paper, we further study Newton's method for solving (1.1) with singular solutions. When the method is applied to solve large-scale problems, the cost of the method is mainly on the solution of the subproblem, a system of linear equations. For large scale problems, the solution of the system of linear equations may be very costly. In order to reduce the computation cost, a popular technique is to find an inexact solution of the system. Truncated conjugate gradient method is regarded to be very efficient. Truncated Newton methods were introduced in the early 1980s [2] and have received much attention since then, see for instance, [2, 9–12] and references therein. Truncated Newton methods are based on the idea that an exact solution of the Newton equation is unnecessary and can be computationally wasteful in the framework of a basic descent method. In a truncated Newton method, the Newton equation is solved inexactly by some inner iterative algorithm within a few inner iterations. In this way, it is desirable to enhance the effectiveness of the method.

In this paper, we introduce a truncated technique in the regularized Newton method [7] to develop a truncated regularized Newton method. We use truncated conjugate gradient method as the inner iterative algorithm. Under the same conditions as those in [7], we obtain the global convergence and local quadratic convergence of the method.

The paper is organized as follows. In the next section, we proposed a truncated regularized Newton method and establish its global convergence property. We prove the superlinear convergence of the method in Sect. 3. In Sect. 4, we present some preliminary numerical results and compare its performance with the regularized Newton method.

2 The algorithm

Let us recall the regularized Newton method in [7] first. Let x_k be the current iterate. For simplicity, we abbreviate $g(x_k)$ and $G(x_k)$ as g_k and G_k respectively. The regularized Newton method generates the next iterate x_{k+1} by letting

$$x_{k+1} = x_k + \lambda_k d_k,$$

where $\lambda_k > 0$ is the steplength and is determined by Armijo type line search strategy, d_k is the unique solution of the system of linear equations

$$(G_k + \mu_k I)d + g_k = 0, \tag{2.1}$$

where I denotes the identity matrix, and $\mu_k > 0$ is a suitable parameter. In our methods, we always take $\mu_k = C_1\|g_k\|$, where $C_1 > 0$ is a constant. Since G_k is positive semidefinite, it is clear that the solution of (2.1) is a descent direction of f at x_k, i.e., ${g_k}^T d_k < 0$.

We are going to introduce a truncate strategy in the above regularized Newton method in a way similar to the method in [4] to develop a truncated regularized Newton method. The steps of the method are stated as follows.

Algorithm (Truncated Regularized Newton (TRN) Method)
Initialization Chosen constants $\alpha, \sigma, \rho \in (0, 1)$ and $C > 0$. Given initial point $x_0 \in R^n$. Let $k = 0$.
Step 0 Set $i = 0$ and

$$p_0 = 0, \qquad q_0 = s_0 = -g_k, \qquad B_k = (G_k + \mu_k I).$$

Step 1 Compute

$$\begin{aligned} \delta_i &= \frac{s_i^T q_i}{s_i^T B_k s_i}, \\ q_{i+1} &= q_i - \delta_i B_k s_i, \\ p_{i+1} &= p_i + \delta_i s_i. \end{aligned}$$

Step 2 If $\frac{\|q_{i+1}\|}{\|g_k\|} \le \min(C, \frac{1}{2\|g_k\|})\|g_k\|$, we let $d_k = p_{i+1}$ and $r_k = -q_{i+1}$, and go to Step 3. Else compute

$$\begin{aligned} \beta_i &= \frac{\|q_{i+1}\|^2}{\|q_i\|^2}, \\ s_{i+1} &= q_{i+1} + \beta_i s_i. \end{aligned}$$

Set $i := i + 1$. Go to Step 1.
Step 3 Find the smallest nonnegative integer i such that

$$f(x_k + \rho^i d_k) \le f(x_k) + \sigma\rho^i g_k^T d_k. \tag{2.2}$$

Let $\lambda_k = \rho^i$.
Step 4 Let $x_{k+1} = x_k + \lambda_k d_k$ and $k := k + 1$. Go to Step 0.

The Steps 1 and 2 in the TRN method are called the truncated steps. They are used to find an inexact solution of the system of linear equations (2.1) by linear conjugate gradient method. Vector r_k in Step 2 represents the error in computing d_k. This means that

$$B_k d_k = -g_k + r_k. \tag{2.3}$$

We conclude this section by giving the following lemma. It can be proved in a way similar to Proposition 1 in [4].

Lemma 2.1 *Let $\{x_k\}$ be generated by the TRN method. Then the following statements are true.*

(i) *For any $i \geq 1$, equality $s_i^T B_k s_i = 0$ holds if and only if $q_i = 0$.*
(ii) *There exists an integer $h \in [0, n-1]$ such that*

$$\frac{\|q_{h+1}\|}{\|g_k\|} \leq \eta_k.$$

(iii) *It holds that*

$$g_k^T p_i < 0, \quad \forall i \geq 1.$$

3 Convergence of the TRN method

To prove the global convergence of the TRN method, we first derive some useful properties for the method.

Lemma 3.1 *We have*

$$r_k^T d_k = 0.$$

Proof Let $m(k)$ be the number of inner iterations to obtain d_k in Step 2. By direct computation, we get

$$r_k = -q_{m(k)+1} = -\left(q_0 - \sum_{i=0}^{m(k)} \delta_i B_k s_i\right)$$

and

$$d_k = p_{m(k)+1} = p_0 + \sum_{i=0}^{m(k)} \delta_i s_i = \sum_{i=0}^{m(k)} \delta_i s_i.$$

So, we derive

$$r_k^T d_k = -\left(q_0 - \sum_{i=0}^{m(k)} \delta_i B_k s_i\right)^T \sum_{i=0}^{m(k)} \delta_i s_i = -\sum_{i=0}^{m(k)} \delta_i q_0^T s_i + \left(\sum_{i=0}^{m(k)} \delta_i B_k s_i\right)^T \sum_{i=0}^{m(k)} \delta_i s_i$$

$$= -\sum_{i=0}^{m(k)} \delta_i q_0^T s_i + \sum_{i=0}^{m(k)} \delta_i{}^2 s_i^T B_k s_i = \sum_{i=0}^{m(k)} \delta_i (-q_0^T s_i + \delta_i s_i^T B_k s_i)$$

$$= \sum_{i=0}^{m(k)} \delta_i \left(-q_0^T s_i + \frac{s_i^T q_i}{s_i^T B_k s_i} s_i^T B_k s_i \right) = \sum_{i=0}^{m(k)} \delta_i (-q_0^T s_i + s_i^T q_i)$$

$$= \sum_{i=0}^{m(k)} \delta_i (-q_i^T s_i + s_i^T q_i) = 0,$$

where the seventh equation follows from the fact that $s_i^T q_i = s_i^T q_0$ for all $i = 0, 1, \ldots, n-1$. This fact can be derived from Step 1 of the TRN method. $\square$

Lemma 3.2 *The direction* $\{d_k\}$ *generated by the TRN method satisfies*

$$g_k^T d_k \le -\mu_k \|d_k\|^2. \tag{3.1}$$

Proof By direct computing, we have:

$$g_k^T d_k = -d_k^T (G_k + \mu_k I) d_k + d_k^T r_k = -d_k^T (G_k + \mu_k I) d_k \le -\mu_k \|d_k\|^2. \qquad \square$$

The following lemma reveals another good property of the TRN method.

Lemma 3.3 *The sequence* $\{d_k\}$ *generated by the TRN method is bounded.*

Proof It follows from the Cauchy-Schwarz inequality that

$$\|g_k\| \|d_k\| \ge |g_k^T d_k| = |-d_k^T (G_k + \mu_k I) d_k + d_k^T r_k| = \|d_k^T (G_k + \mu_k I) d_k\| \ge \mu_k \|d_k\|^2.$$

Since $\mu_k = C_1 \|g_k\|$, the last inequality implies $\|d_k\| \le C_1^{-1}$. $\square$

The following theorem establishes the global convergence of the TRN method.

Theorem 3.4 *Let* $\{x_k\}$ *be generated by the TRN method. Suppose that* f *is bounded from below. Then every accumulation point of* $\{x_k\}$, *if exists, is a solution of* (1.1).

Proof It is easy to deduce from (2.2) that

$$\lim_{k\to\infty} \lambda_k g_k^T d_k = 0. \tag{3.2}$$

Let $\{x_k\}_{k\in K}$ converge to some point $\tilde{x}$. Since $\{d_k\}_{k\in K}$ is bounded, without loss of generality, we suppose $\{d_k\}_{k\in K}$ converges to some vector $\tilde{d}$. Let $\tilde{\lambda} = \limsup_{k\to\infty,\, k\in K} \lambda_k$. It is clear that $\tilde{\lambda} \ge 0$. Denote $\tilde{g} = g(\tilde{x})$.

If $\tilde{\lambda} > 0$, we get from (3.2)

$$\tilde{g}^T \tilde{d} = 0. \tag{3.3}$$

If $\tilde{\lambda} = 0$, we see from the line search process that when $k \in K$ is sufficiently large, the following inequality holds

$$f(x_k + \lambda'_k d_k) > f(x_k) + \sigma \lambda'_k g_k^T d_k,$$

where $\lambda'_k = \frac{\lambda_k}{\rho}$. Dividing both side of the inequality by λ'_k and then taking limits as $k \to \infty, k \in K$, we get

$$\lim_{k\in K,\ k\to\infty} \frac{f(x_k + \lambda'_k d_k) - f(x_k)}{\lambda'_k} \geq \lim_{k\in K,\ k\to\infty} \sigma g_k^T d_k = \sigma \tilde{g}^T \tilde{d}.$$

This implies $\tilde{g}^T \tilde{d} \geq \sigma \tilde{g}^T \tilde{d}$. Since $\sigma \in (0, 1)$ and $\tilde{g}^T \tilde{d} \leq 0$, we also get (3.3).

Let $\tilde{\mu} = C_1 \|\tilde{g}\|$. It follows from (2.3) and Lemma 3.1 that

$$d_k^T (G_k + \mu_k I) d_k + d_k^T g_k = 0.$$

Taking limits in both side as $k \to \infty$ with $k \in K$, taking into account that $G(\tilde{x})$ is positive semidefinite, we get

$$\tilde{\mu} \|\tilde{d}\|^2 = 0.$$

If $\tilde{\mu} = 0$, then we have $\|\tilde{g}\| = 0$. Otherwise, we have $\tilde{d} = 0$. It follows from (2.3) and Step 2 that

$$\|g_k\| \leq \|B_k d_k\| + \|r_k\| \leq \|B_k d_k\| + \min\left(C, \frac{1}{2\|g_k\|}\right) \|g_k\|^2.$$

Observe that the sequence $\{B_k\}_{k\in K} = \{G_k + \mu_k I\}_{k\in K}$ is bounded. Taking limits as $k \to \infty$ with $k \in K$ in the last inequality, we get

$$\|\tilde{g}\| \leq \min\left(C, \frac{1}{2\|\tilde{g}\|}\right) \|\tilde{g}\|^2 \leq \frac{1}{2} \|\tilde{g}\|.$$

The last inequality also implies $\tilde{g} = 0$. The proof is complete. □

Theorem 3.4 has established the global convergence of the TRN method. Moreover, as $\{x_k\}_{k\in K}$ tends to $\tilde{x}$ with $g(\tilde{x}) = 0$, the truncated rule in Step 2 of TRN method essentially reduces to find d_k satisfying

$$\|B_k d_k + g_k\| \leq C \|g_k\|^2.$$

Consequently, the TRN method locally reduces to the inexact regularized Newton method I in [7]. The local quadratic convergence of the method then follows from Theorems 3.2. We state the theorem as follows but omit the proof.

Theorem 3.5 *Suppose that $\|g(x)\|$ provides a local error bound near $\bar{x} \in X$, and x_{k+1} is determined by $x_{k+1} := x_k + d_k$. If the initial point x_0 is sufficient close to $\bar{x}$, then the entire sequence $\{x_k\}$ convergent to a solution of (1.1). Moreover, the rate of convergence is quadratic.*

In a way similar to the proof of Theorem 4.4 in [7], it is not difficult to establish the following theorem, which shows that the unit steplength is essentially accepted.

Theorem 3.6 *Let $\{x_k\}$ be generated by the TRN method with $\sigma \in (0, \frac{1}{2})$. Let $\{x_k\}_{x\in K}$ be an arbitrary subsequence whose limit is $\tilde{x} \in X$. If $\|g(x)\|$ provides a local error bound near $\tilde{x}$, i.e., there is a constant $M > 0$ such that the inequality*

$$\|g(x)\| \geq M\mathrm{dist}(x, X)$$

holds for all x in a neighborhood of $\tilde{x}$, then we have $\lambda_k = 1$ for all $k \in K$ sufficiently large. Moreover, $\{x_k\}_{x\in K}$ is quadratically convergent.

4 Numerical experiments

In this section, we test the TRN method and compare its performance with that of the regularized Newton method in [7]. We test the following problems:

Problem 1

$$f = \frac{1}{2}\sum_{i=1}^{n-1}(x_i - x_{i+1})^2 + \frac{1}{12}\sum_{i=1}^{n-1}(x_i - x_{i+1})^4;$$

Problem 2

$$f = \frac{1}{2}\sum_{i=1}^{n-1}(x_i - x_{i+1})^2;$$

Problem 3

$$f = \sum_{i=2}^{n}[\exp(x_i - x_{i-1})^2 + (x_i - x_{i-1})^2 + 2{x_i}^4 + 4{x_{i-1}}^4];$$

Problem 4

$$f = \sum_{i=1}^{n}\left[\frac{1}{2}(x_i - x_{i-1})^2 + \sin(x_i - x_{i-1}) + 2(2x_i + 3x_{i-1} - 15)^4\right];$$

Problem 5

$$f = \sum_{j=1}^{k}[(x_{i-1} + 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+1})^4],$$

$$i = 2j, k = (n-2)/2;$$

Problem 6

$$f = \sum_{j=1}^{k} [(x_{i-1} + 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4],$$
$$i = 2j, k = (n-2)/2;$$

Problem 7

$$f = \sum_{i=2}^{n} [(x_{i-1} - 3)^2 + (x_{i-1} - x_i)^2 + \exp(20(x_{i-1} - x_i))].$$

Problems 1 and 2 come from [7] and Problems 6 and 7 come from [8] that are called Chained Powell Singular Function and Generalization of Brown Function 1, respectively. Problem 5 is a little different from Problem 6 in the last term. It is easy to see that all the problems are convex. In addition, Problems 1–6 are singular while Problem 7 is nonsingular. It is not difficult to show that the function $\|g(x)\|$ in Problems 1–6 provide a local error bound for the solution set.

The algorithms were coded in Fortran 95 and run on a PC with 1.6 GHZ CPU processor. We first test the global convergence of the TRN method. We test it on Problems 1–7 with different dimensions and different initial points. The parameters in the TRN method are specified as follows. We choose $C = 10^{-5}$, $C_1 = 10^{-5}$, $\sigma = 0.2$ and $\rho = 0.5$. We stop the iterative process if $\|g_k\| \le 10^{-6}$. The results are listed in Table 1. The meaning of each column in Table 1 is stated below.

Prob:	the number of the test problem;
n:	the dimension of the problem;
md:	the number of iterations;
mg:	the evaluations of the gradients;
x_i:	the i-th element of initial point;
normg:	the final value of $\|g_k\|$;
mf:	the function evaluations;
Time:	the CPU time used (in second).

We see from Table 1 that starting from any initial point, the TRN method terminated at a solution of the problem in a few iterations. We then compare the performance of the TRN method with that of the regularized Newton method [7] which we denoted by RN. The parameters in the regularized Newton method are set to be the same as those in the TRN method. The subproblems of RN method are solved by linear conjugate gradient method.

Table 2 lists the performance of the methods TRN and RN. For simplicity, in Table 2, we simply use "1" and "2" to denote TRN method and RN method respectively. We define a ratio $Ratio \triangleq \frac{Time_1}{Time_2}$ to compare the efficiency of the two methods, where $Time_1$ and $Time_2$ denote the cpu time used by TRN method and RN method. The value of $Ratio$ greater than 1 shows that the TRN method performs better than the RN method does.

Table 1 Results for TRN

Prob	dim	x_i	md	mg	normg	dim	x_i	md	mg	normg
1	100	i	4	208	0.59914E−12		i	4	1011	0.77770E−08
		$n-i$	4	204	0.59973E−12	500	$n-i$	4	1011	0.77770E−08
		$\frac{1}{i}$	3	295	0.75795E−11		$\frac{1}{i}$	3	1495	0.82621E−11
	1000	i	5	3042	0.32073E−10		i	6	7174	0.13727E−10
		$n-i$	5	3040	0.32071E−10	2000	$n-i$	6	7180	0.13741E−10
		$\frac{1}{i}$	3	2995	0.26710E−10		$\frac{1}{i}$	3	5214	0.12674E−09
2	100	i	2	99	0.16297E−06		i	3	748	0.20294E−07
		$n-i$	2	99	0.16297E−06	500	$n-i$	3	748	0.20294E−07
		$\frac{1}{i}$	2	197	0.22365E−12		$\frac{1}{i}$	2	997	0.33609E−11
	1000	i	4	2488	0.59443E−10		i	4	3990	0.79725E−06
		$n-i$	4	2482	0.59442E−10	2000	$n-i$	4	3990	0.79725E−06
		$\frac{1}{i}$	2	1997	0.78897E−11		$\frac{1}{i}$	2	3313	0.15058E−10
3	100	0.5	15	992	0.35472E−06		0.5	15	3247	0.79789E−06
		1	16	928	0.84089E−06	500	1	17	3392	0.56049E−06
		$\frac{1}{i}$	13	1189	0.46691E−06		$\frac{1}{i}$	13	5271	0.31373E−06
	1000	0.5	16	5948	0.33460E−06		0.5	16	8837	0.47342E−06
		1	17	5168	0.79337E−06	2000	1	18	9337	0.33271E−06
		$\frac{1}{i}$	13	10046	0.32727E−06		$\frac{1}{i}$	13	18865	0.33290E−06
4	100	i	31	773	0.56567E−06		i	37	2305	0.58221E−06
		$n-i$	30	779	0.20186E−06	500	$n-i$	35	2137	0.70141E−06
		$\frac{1}{i}$	23	839	0.12285E−06		$\frac{1}{i}$	24	2620	0.53932E−06
	1000	i	37	2674	0.99928E−06		i	41	3218	0.88771E−06
		$n-i$	37	2646	0.92274E−06	2000	$n-i$	40	3229	0.75977E−06
		$\frac{1}{i}$	24	3421	0.62350E−06		$\frac{1}{i}$	24	3581	0.76882E−06
5	100	i	25	167	0.75172E−06		i	32	207	0.41250E−06
		$n-i$	26	179	0.30504E−06	500	$n-i$	31	197	0.33036E−06
		$\frac{1}{i}$	14	114	0.67386E−06		$\frac{1}{i}$	14	106	0.67386E−06
	1000	i	35	219	0.66167E−06		i	45	243	0.45296E−06
		$n-i$	35	234	0.37234E−06	2000	$n-i$	41	246	0.65094E−06
		$\frac{1}{i}$	14	102	0.67386E−06		$\frac{1}{i}$	14	98	0.67386E−06
6	100	i	25	204	0.65413E−06		i	32	233	0.40582E−06
		$n-i$	27	217	0.30695E−06	500	$n-i$	37	236	0.50521E−06
		$\frac{1}{i}$	14	112	0.95939E−06		$\frac{1}{i}$	14	104	0.95939E−06
	1000	i	36	235	0.89223E−06		i	41	263	0.63125E−06
		$n-i$	42	296	0.92445E−06	2000	$n-i$	46	332	0.79274E−06
		$\frac{1}{i}$	14	103	0.95939E−06		$\frac{1}{i}$	14	101	0.95939E−06
7	100	0.5	7	726	0.35130E−06		0	7	1161	0.35208E−06
		1	7	735	0.35130E−06	500	1	7	1165	0.35174E−06
		$\frac{1}{i}$	16	732	0.20750E−07		$\frac{1}{i}$	16	1005	0.18033E−07
	1000	0.5	7	1151	0.35289E−06		0.5	7	1143	0.35464E−06
		1	7	1157	0.35228E−06	2000	1	4	1147	0.35324E−06
		$\frac{1}{i}$	16	1007	0.17582E−07		$\frac{1}{i}$	16	1005	0.20111E−07

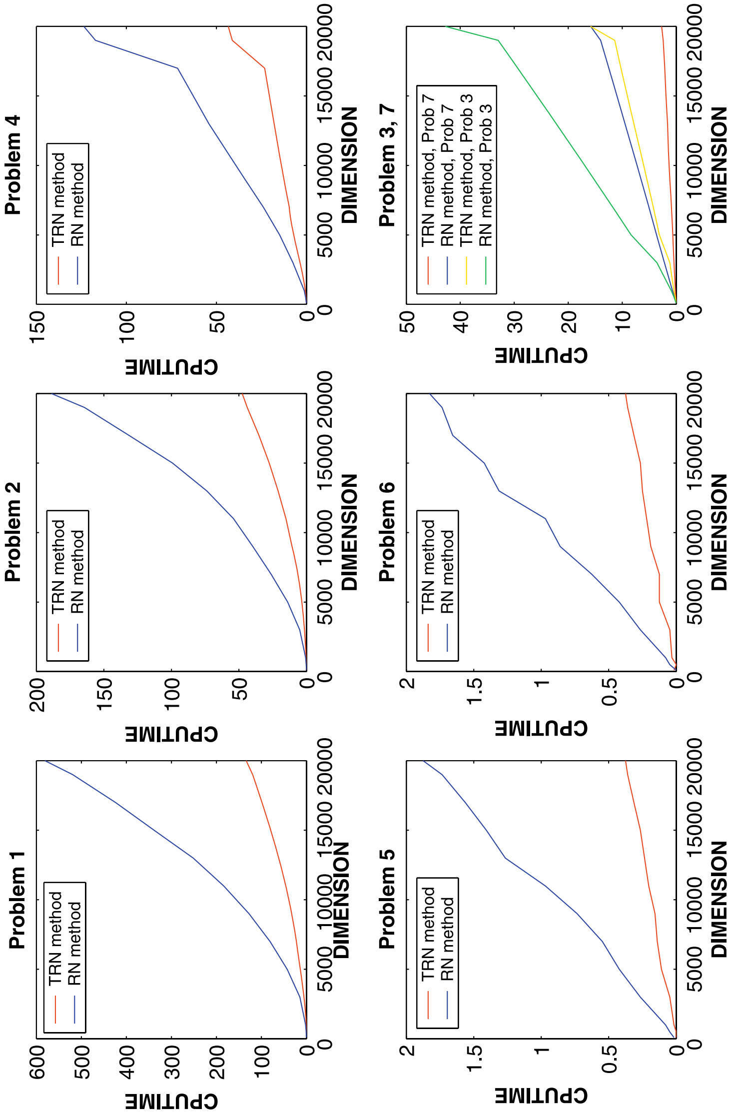

Fig. 1 Comparison of RN and TRN methods

Table 2 Comparison of TRN and RN methods

Prob	x_i	n	method	md (mf)	Time	mg	normg	Ratio
1	$\frac{1}{i}$	100	1	3	0.00000E+00	295	0.75795E−11	–
			2	3	0.15625E−01	885	0.75794E−11	
		1000	1	3	0.75000E+00	2995	0.26710E−10	2.3749
			2	3	0.17812E+01	7430	0.26710E−10	
		10000	1	3	0.40484E+02	16062	0.38280E−09	3.8302
			2	3	0.15506E+03	62372	0.37296E−09	
		20000	1	3	0.13358E+03	26060	0.41148E−09	4.3484
			2	3	0.58086E+03	112183	0.40239E−09	
2	$\frac{1}{i}$	100	1	2	0.00000E+00	197	0.22365E−12	–
			2	2	0.00000E+00	650	0.22365E−12	
		1000	1	2	0.25000E+00	1997	0.78897E−11	2.3750
			2	2	0.59375E+00	5260	0.78897E−11	
		10000	1	2	0.13016E+02	11310	0.19895E−10	3.6541
			2	2	0.47562E+02	42579	0.17177E−10	
		20000	1	2	0.49594E+02	21310	0.20600E−10	3.8077
			2	2	0.18884E+03	80001	0.17834E−10	
3	1	100	1	16	0.15625E−01	928	0.84089E−06	2.0000
			2	16	0.31250E−01	2171	0.84088E−06	
		1000	1	17	0.57812E+00	5168	0.79337E−06	2.1622
			2	17	0.12500E+01	11585	0.79332E−06	
		10000	1	18	0.13969E+02	12820	0.74463E−06	2.7706
			2	18	0.38703E+02	36323	0.74457E−06	
		20000	1	19	0.43422E+02	19702	0.31221E−06	2.8467
			2	19	0.12361E+03	56650	0.31220E−06	
4	1	100	1	24	0.15625E−01	915	0.38815E−07	2.0000
			2	23	0.31250E−01	3049	0.91771E−07	
		1000	1	24	0.40625E+00	3300	0.70051E−06	2.4999
			2	24	0.10156E+01	9314	0.38296E−06	
		10000	1	24	0.47812E+01	4244	0.86894E−06	3.0490
			2	24	0.14578E+02	13108	0.86908E−06	
		20000	1	25	0.15875E+02	6692	0.36093E−06	2.6929
			2	25	0.42750E+02	18676	0.36093E−06	
5	1	100	1	15	0.15625E−01	130	0.58834E−06	1.0000
			2	20	0.15625E−01	586	0.34742E−06	
		1000	1	15	0.15625E−01	127	0.58905E−06	6.0000
			2	21	0.93750E−01	655	0.52062E−06	
		10000	1	15	0.18750E+00	121	0.56746E−06	5.0000
			2	22	0.93750E+00	730	0.58836E−06	
		20000	1	17	0.37500E+00	120	0.53492E−06	5.0000
			2	22	0.18750E+01	716	0.32470E−06	

Table 2 (*Continued*)

Prob	x_i	n	method	md (mf)	Time	mg	normg	Ratio
6	1	100	1	16	0.00000E+00	134	0.51681E−06	–
			2	20	0.00000E+00	603	0.34113E−06	
		1000	1	16	0.15625E−01	131	0.51813E−06	6.0000
			2	21	0.93750E−01	663	0.48967E−06	
		10000	1	16	0.20312E+00	128	0.51754E−06	4.5386
			2	22	0.92188E+00	725	0.50989E−06	
		20000	1	17	0.37500E+00	108	0.99132E−06	4.8749
			2	21	0.18281E+01	704	0.92198E−06	
7	1	100	1	7	0.15625E−01	735	0.35130E−06	1.0000
			2	7	0.15625E−01	1586	0.35126E−06	
		1000	1	7	0.12500E+00	1157	0.35228E−06	5.0000
			2	7	0.75000E+00	6684	0.35168E−06	
		10000	1	7	0.13125E+01	1123	0.35721E−06	5.8810
			2	7	0.77188E+01	6758	0.35308E−06	
		20000	1	7	0.26875E+01	1114	0.35742E−06	5.8954
			2	7	0.15844E+02	6759	0.35394E−06	

We see from Table 2 that for each initial point, both methods terminated at a solution of the problem. However, in all cases, TRN method used less CPU time. Figure 1 further shows the comparison of the two methods in CPU time. As it is shown in Fig. 1, TRN method performs better than RN method. Moreover, as the size of the problem goes to larger, the advantage becomes more obvious. This shows that TRN method has the potential advantage for solving large scale problems with singular solutions.

References

1. Dan, H., Yamashita, N., Fukushima, M.: Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions. Optim. Methods Softw. **17**, 605–626 (2002)
2. Dembo, R.S., Steihaug, T.: Truncated-Newton algorithms for large-scale unconstrained optimization. Math. Program. **26**, 190–212 (1983)
3. Fan, J., Yuan, Y.: On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption. Computing **74**, 23–29 (2005)
4. Fasano, G., Lampariello, F., Sciandrone, M.: Truncated nonmonotone Gauss-Newton method for large-scale nonlinear least-squares problems. Comput. Optim. Appl. **34**, 343–358 (2006)
5. Fischer, A.: Local behavior of an iterative framework for generalized equations with nonisolated solutions. Math. Program. **94**, 91–124 (2002)
6. Kanzow, C., Yamashita, N., Fukushima, M.: Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties. J. Comput. Appl. Math. **173**, 321–343 (2005)
7. Li, D.H., Fukushima, M., Qi, L., Yamashita, N.: Regularized Newton method for convex minimization problems with singular solutions. Comput. Optim. Appl. **26**, 131–147 (2004)
8. Lukšan, L., Vičck, J.: Sparse and partially separable test problems for unconstrained and equality constrained optimization. Academy of the Czech Republic, Institute of Computer Science, Technical Report no. 767 (January 1999)

9. Nash, S.G.: Solving nonlinear programming problems using truncated-Newton techniques. In: Boggs, P.T., Byrd, R.H., Schnabel, R.B. (eds.) Numerical Optimization, pp. 119–136. SIAM, Philadelphia (1985)
10. Nash, S.G.: A survey of truncated-Newton methods. J. Comput. Appl. Math. **124**, 45–59 (2000)
11. Nash, S.G., Nocedal, J.: A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. SIAM J. Optim. **1**, 358–372 (1991)
12. Nash, S.G., Sofer, A.: Assessing a search direction within a truncated-Newton method. Oper. Res. Lett. **9**, 219–221 (1990)
13. Solodov, M.V., Svaiter, B.F.: A globally convergent inexact Newton method for system of monotone equations. In: Fukushima, M., Qi, L. (eds.) Reformulation: Nonsmooth, Piecewise Smooth, Semismooth, and Smoothing Methods, pp. 355–269. Kluwer Academic, Dordrecht (1999)
14. Yamashita, N., Fukushima, M.: On the rate of convergence of the Levenberg-Marquardt method. Computing **15**(Suppl.), 239–249 (2001)
15. Zhou, G., Qi, L.: On the convergence of an inexact Newton-type method. Oper. Res. Lett. **34**, 647–652 (2006)
16. Zhou, G., Yoh, K.C., Sun, J.: Superlinear convergence of a Newton-type algorithm for monotone equations. J. Optim. Theory Appl. **125**, 205–221 (2005)