KYBERNETIKA — VOLUME 32 (1996), NUMBER 2, PAGES 121 - 138

COMBINED TRUST REGION METHODS FOR NONLINEAR LEAST SQUARES[1]

LADISLAV LUKŠAN

Trust region realizations of the Gauss–Newton method are commonly used for obtaining solution of nonlinear least squares problems. We propose three efficient algorithms which improve standard trust region techniques: multiple dog-leg strategy for dense problems and two combined conjugate gradient Lanczos strategies for sparse problems. Efficiency of these methods is demonstrated by extensive numerical experiments.

1. INTRODUCTION

Let $f_i : \mathbb{R}^n \to \mathbb{R}$, $1 \leq i \leq r$, be real-valued functions with continuous second order derivatives on the open set $X \subset \mathbb{R}^n$. Let us denote

$$F(x) = \frac{1}{2} \sum_{i=1}^{r} f_i^2(x). \tag{1.1}$$

We are concerned with the finding a local minimum $x^* \in \mathbb{R}^n$ of the function $F : \mathbb{R}^n \to \mathbb{R}$ given by (1.1) on an open set $X \subset \mathbb{R}^n$, i.e. a point $x^* \in \mathbb{R}^n$ that satisfies the inequality $F(x^*) \leq F(x)\ \forall\, x \in B(x^*, \varepsilon)$ for some $\varepsilon > 0$, where $B(x^*, \varepsilon) = \{x \in \mathbb{R}^n : \|x - x^*\| < \varepsilon\} \subset X$ is an open ball contained in $X \subset \mathbb{R}^n$.

If we denote $g_i(x)$ and $G_i(x)$ the gradients and the Hessian matrices of the functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $1 \leq i \leq r$, respectively, and $g(x)$ and $G(x)$ the gradient and the Hessian matrix of the function $F : \mathbb{R}^n \to \mathbb{R}$ respectively then, using (1.1), we obtain

$$g(x) = \sum_{i=1}^{r} f_i(x)\, g_i(x) \tag{1.2}$$

and

$$G(x) = \sum_{i=1}^{r} g_i(x)\, g_i^T(x) + \sum_{i=1}^{r} f_i(x)\, G_i(x) \tag{1.3}$$

[1] This work was supported under the grant No. 23012 given by the Grant Agency of the Academy of Sciences of the Czech Republic.

Numerical methods for local minimization of the objective function $F : \mathbb{R}^n \to \mathbb{R}$ are usually derived from the Newton method. These methods are iterative and their iteration step has the form

$$x^+ = x + \alpha d,$$

where x and x^+ are old and new vectors of variables respectively, α is a stepsize parameter and d is a direction vector which approximately minimizes the quadratic function

$$Q(d) = \frac{1}{2} d^T B d + g^T d \tag{1.4}$$

over some subset of R^n. Here $B = B(x)$ is an approximation of the Hessian matrix $G(x)$ and $g = g(x)$ is the gradient given by (1.2). There are two basic possibilities concerning how the matrix B in (1.4) can be constructed. The first possibility leads to the so-called variable metric methods which use an arbitrary positive definite matrix in the first iteration and which generate subsequent matrices by simple variable metric updates [8]. The main advantage of this approach is its general applicability (the objective function $F : \mathbb{R}^n \to \mathbb{R}$ need not have the special form (1.1)) and the possibility to update matrix factorization which requires only $O(n^2)$ operations in every iteration. Therefore, these methods are very efficient for dense, medium-size, and well conditioned problems.

The second possibility is based on the special form (1.1) of the objective function $F : \mathbb{R}^n \to \mathbb{R}$ and it consists in the substitution

$$B(x) = \sum_{i=1}^{r} g_i(x)\, g_i^T(x). \tag{1.5}$$

One reason for this choice is the fact that often $F(x^*) = 0$ so that the second term of (1.3) is negligible in $B(x^*, \varepsilon)$. Another reason follows from the linearization of (1.1). In this case

$$\begin{aligned} F(x+d) &\approx \frac{1}{2}\sum_{i=1}^{r}(f_i(x) + g_i^T(x)\, d)^2 \\ &= \frac{1}{2}\sum_{i=1}^{r}(f_i^2(x) + 2 f_i(x)\, g_i^T(x)\, d + d^T g_i(x)\, g_i^T(x)\, d) \\ &= F(x) + g^T(x)\, d + \frac{1}{2} d^T B d = F(x) + Q(d) \end{aligned}$$

with B given by (1.5). The methods which use the matrix (1.5) instead of the Hessian matrix $G(x)$ are called Gauss–Newton (or modified Gauss–Newton) methods [6]. The main advantage of the Gauss–Newton methods is their quadratic convergence for zero-residual problems. Convergence of the Guass-Newton methods is usually faster then convergence of the variable metric methods. On the other hand the matrix (1.5) has to be factorized which consume $O(n^3)$ operations in every iteration. Therefore these methods are very efficient for dense, small-size, and zero-residual or ill-conditioned problems. The Gauss–Newton methods are also very efficient for sparse problems since factorizations of sparse matrices are relatively inexpensive

and, moreover, the variable metric methods cannot be efficiently generalized to use sparse matrix factorization.

Besides the above two possibilities there exist their various combinations (see [3], [5] or [1], [9] as an example). We do not concern these hybrid methods here, the detailed investigation of them is given in [15].

All the above methods can be realized in two different forms using either the line search strategy or the trust region strategy. A typical iteration step of the line search strategy has the following form.

(L1) Direction determination:
Choose $d \in \mathbb{R}^n$ so that

$$||Bd + g|| \leq \omega ||g|| \tag{1.6}$$

and

$$-g^T d \geq \bar{\varepsilon}_0 ||g|| \, ||d||, \tag{1.7}$$

where $0 \leq \omega \leq \bar{\omega} < 1$, $\bar{\varepsilon}_0 > 0$ ($\bar{\omega}$ and $\bar{\varepsilon}_0$ do not depend on the iteration step), $g = g(x)$ and $B = B(x)$.

(L2) Stepsize selection:
Choose $\alpha > 0$ so that

$$F(x + \alpha d) - F \leq \bar{\varepsilon}_1 \alpha g^T d \tag{1.8a}$$

and

$$g^T (x + \alpha d) \, d \geq \bar{\varepsilon}_2 g^T d, \tag{1.8b}$$

where $0 \leq \bar{\varepsilon}_1 < 1/2$, $\bar{\varepsilon}_1 < \bar{\varepsilon}_2 < 1$ ($\bar{\varepsilon}_1$ and $\bar{\varepsilon}_2$ do not depend on the iteration step) $F = F(x)$ and $g = g(x)$. Finally set

$$x^+ = x + \alpha s. \tag{1.9}$$

If the conditions (1.6) and (1.7) cannot be satisfied simultaneously, we must change the matrix B (restart).

The line search startegy is very convenient for the variable metric methods that generate matrices which are usually well-conditioned. Another situation appears for the Gauss–Newton methods since the matrix given by (1.5) is very often ill-conditioned even singular. In this case, the direction vector $d \in \mathbb{R}^n$ can have rather large euclidean norm and, moreover, it can be almost orthogonal to the gradient g. Therefore, too many line search steps can appear for satisfying (1.8) and, moreover, frequent restarts can occur due to violation of (1.7).

A typical iteration step of the trust region strategy has the following form.

(T1) Direction determination:
Choose $d \in \mathbb{R}^n$ so that

$$||d|| \leq \Delta \tag{1.10a}$$

$$||d|| < \Delta \Longrightarrow ||Bd + g|| \leq \omega ||g|| \tag{1.10b}$$

and

$$-Q(d) \geq \bar{\varepsilon}_0 ||g|| \, \min(||d||, ||g||/||B||), \tag{1.11}$$

where $0 < \Delta \leq \bar{\Delta}$, $0 \leq \omega \leq \bar{\omega} < 1$, $\bar{\varepsilon}_0 > 0$ (barred constants do not depend on the iteration step), $g = g(x)$ and $B = B(x)$ ($Q(d)$ is given by (1.4)).

(T2) Stepsize selection:

$$x^+ = x + d \quad \text{if} \quad F(x+d) < F(x) \tag{1.12a}$$

$$x^+ = x \quad \text{if} \quad F(x+d) \geq F(x). \tag{1.12b}$$

(T3) Trust region update:

Compute

$$\rho = \frac{F(x+d) - F(x)}{Q(d)}. \tag{1.13}$$

When $\rho < \bar{\rho}_1$, then determine the value

$$\beta = \frac{1}{2\left(1 - \frac{F(x+d)-F(x)}{g^T d}\right)}$$

(quadratic interpolation) and set

$$\Delta^+ = \bar{\beta}_1 ||d|| \quad \text{if} \quad \beta < \bar{\beta}_1 \tag{1.14a}$$

$$\Delta^+ = \beta ||d|| \quad \text{if} \quad \bar{\beta}_1 \leq \beta \leq \bar{\beta}_2 \tag{1.14b}$$

$$\Delta^+ = \bar{\beta}_2 ||d|| \quad \text{if} \quad \bar{\beta}_2 < \beta. \tag{1.14c}$$

When $\bar{\rho}_1 \leq \rho \leq \bar{\rho}_2$ then set

$$\Delta^+ = \min(\Delta, \bar{\gamma}_2 ||d||). \tag{1.15}$$

When $\bar{\rho}_2 < \rho$ then set

$$\Delta^+ = \min(\max(\Delta, \bar{\gamma}_1 ||d||), \bar{\gamma}_2 ||d||, \bar{\Delta}). \tag{1.16}$$

Here $0 < \bar{\beta}_1 \leq \bar{\beta}_2 < 1 < \bar{\gamma}_1 < \bar{\gamma}_2$, $0 < \bar{\rho}_1 < \bar{\rho}_2 < 1$ and $\bar{\Delta} > 0$ (barred constants do not depend on the iteration step).

The trust region strategy with the iteration step (T1)–(T3) has strong global convergence properties (see [20], [21]). Even if it also works well for indefinite matrices $B(x)$, we confine our attention to the positive semidifinite case which appears in connection with the Gauss–Newton methods. In this case the following theorem holds (see [14]).

Theorem 1.1. Let the functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $1 \leq i \leq r$, have continuous second order derivatives and there exist constants $C_1 > 0$, $C_2 > 0$, $C_3 > 0$ such that $|f_i(x)| \leq C_1$, $||g_i(x)|| \leq C_2$, $||G_i(x)|| \leq C_3$, $1 \leq i \leq r$, for all $x \in X \subset \mathbb{R}^n$. Let $x_k \in X \subset \mathbb{R}^n$, $k \in N$, be the sequence generated by the Gauss–Newton method with the trust region strategy (T1)–(T3). Then

$$\liminf_{k \to \infty} ||g(x_k)|| = 0 \tag{1.17}$$

The trust region strategy is very advantageous in connection with the Gauss–Newton method. The matrix (1.5) can be ill-conditioned, even singular, but $||d||$ is always bounded from above according to (1.10). Moreover (1.17) holds without any

restart. Strategies like the trust region strategy (T1)–(T3) were proposed already in [13], [16]. The current realizations were developed in [4], [18], [19], [22].

The most complicated part of the trust region strategy is computation of the vector $d \in \mathbb{R}^n$ satisfying the conditions (1.10)–(1.11). There exists three basic possibilities for a positive semidefinite case. First, the vector $d \in \mathbb{R}^n$ can be obtained as a solution of the subproblem

$$d = \underset{\|d(\lambda)\| \leq \Delta}{\arg\min} \; Q(d(\lambda)) \tag{1.18}$$

which leads to the repeated solution of the equation $(B+\lambda I)d(\lambda)+g = 0$ for selected values of λ [18]. This way gives well-convergent algorithms but for a large number of variables it is time consuming since it uses, on average, 2–3 Choleski decompositions in every iteration. Moreover, an additional matrix has to be used.

The second possibility consists in replacing the complicated subproblem (1.18) by the two-dimensional subproblem

$$d = \underset{\|d(\alpha,\beta)\| \leq \Delta}{\arg\min} \; Q(d(\alpha,\beta)), \tag{1.19}$$

where $d(\alpha,\beta) = \alpha g + \beta B^{-1} g$ [2]. Usually the subproblem (1.19) is solved only approximately, getting (1.10)–(1.11), by the so-called dog-leg methods [4], [19]. In this case we compute the vectors $d_1 \in \mathbb{R}^n$ and $d_n \in \mathbb{R}^n$ such that $g^T B g d_1 + \|g\|^2 g = 0$ and $B d_n + g = 0$. The resulting vector $d \in \mathbb{R}^n$ is obtained as $d = \lambda d_1$ if $\|d_1\| \geq \Delta$, $d = d_1 + \lambda(d_n - d_1)$ if $\|d_1\| < \Delta \leq \|d_n\|$, and $d = d_n$ if $\|d_n\| < \Delta$, where the scaling factor $\lambda > 0$ is chosen so that $\|d\| = \Delta$. This way is more economical since the equation $B d_n + g = 0$ is solved, at most, once in every iteration and no additional matrix is used.

The third possibility is very natural. The equation $Bd + g = 0$ is solved by the conjugate gradient method which generates the vectors $d_i \in \mathbb{R}^n$, $i \in N$, having the following properties (see [22]):

(A) There exists an index $k \leq n$, such that $\|Bd_k + g\| \leq \omega \|g\|$ for a given $0 < \omega < 1$.

(B) The sequence $Q(d_i)$, $1 \leq i \leq k$, is decreasing, i.e. $Q(d_{i+1}) < Q(d_i)$ for $1 \leq i < k$.

(C) The sequence $\|d_i\|$, $1 \leq i \leq k$, is increasing, i.e. $\|d_{i+1}\| > \|d_i\|$ for $1 \leq i < k$.

(D) It holds that $2Q(\lambda d_1) \leq -\|g\| \, \|\lambda d_1\|$ for $0 \leq \lambda \leq 1$, and $2Q(d_i) \leq -\|g\|^2 / \|B\|$ for $1 \leq i \leq k$.

The resulting vector $d \in \mathbb{R}^n$ is then obtained as $d = \lambda d_1$ if $\|d_1\| \geq \Delta$, $d = d_i + \lambda(d_{i+1} - d_i)$ if $\|d_i\| < \Delta \leq \|d_{i+1}\|$ for some $1 \leq i < k$, and $d = d_k$ if $\|d_k\| < \Delta$, where the scaling factor $\lambda > 0$ is chosen so that $\|d\| = \Delta$. Note that (A)–(D) imply (1.10)–(1.11). Note also that no matrix factorization is used in the conjugate gradient method but, for small ω, the index k in (A) can be large. Fortunately the condition $\|d\| \leq \Delta$ also limits the number of iterations.

In the subsequent text we confine our attention to the trust region realizations of the Gauss–Newton method. Our main purpose is to construct new trust region strategies which outperform all the above described ones in both the number of function evaluations and the computational time. Section 2 is devoted to the multiple dog-leg strategies for dense problems. In Section 3 we propose combined conjugate gradient Lanczos methods for sparse problems. Efficiency of these methods is demostrated by extensive numerical experiments.

2. MULTIPLE DOG–LEG STRATEGIES FOR DENSE PROBLEMS

Consider the conjugate gradient method applied to the quadratic function (1.4). This method is represented by the following iterative process

$$d_0 = 0, \quad g_0 = g \tag{2.1a}$$

$$p_0 = 0, \quad \delta_0 = 0 \tag{2.1b}$$

and

$$p_i = -g_{i-1} + \delta_{i-1} p_{i-1} \tag{2.1c}$$

$$q_i = B p_i \tag{2.1d}$$

$$\gamma_i = \|g_{i-1}\|^2 / p_i^T q_i \tag{2.1e}$$

$$d_i = d_{i-1} + \gamma_i p_i \tag{2.1f}$$

$$g_i = g_{i-1} + \gamma_i q_i \tag{2.1g}$$

$$\delta_i = \|g_i\|^2 / \|g_{i-1}\| \tag{2.1h}$$

for $i \in N$. Note that $g_i = B d_i + g$ for $i \in N$.

The matrix B given by (1.5) is always positive semidefinite. First, suppose that it is positive definite. Then the following well-known lemma holds (see [12], [22]).

Lemma 2.1. Consider the conjugate gradient process (2.1) with a symmetric positive definite matrix B. Then there exists an integer $l \leq n$ such that $d_l \in \mathbb{R}^n$ is a minimizer of the quadratic function (1.4) and

$$p_i^T B p_j = d_i^T B p_j = 0 \tag{2.2a}$$

$$p_i^T g_j = d_i^T g_j = 0 \tag{2.2b}$$

$$g_i^T p_j = -g_{j-1}^T g_{j-1} \tag{2.2c}$$

$$g_i^T g_j = 0 \tag{2.2d}$$

$$Q(d_i) > Q(d_j) \tag{2.2e}$$

$$\|d_i\| < \|d_j\| \tag{2.2f}$$

hold for $0 \leq i < j \leq l$. Moreover, if $k \leq l$ then the vectors g_i, $0 \leq i \leq k-1$ form an orthogonal basis in the Krylov subspace

$$\mathcal{K}_k(B, g_0) = span\{B^i g_0, 0 \leq i \leq k-1\}$$

and

$$d_i = \underset{d \in \mathcal{K}_i(B,g_0)}{\arg\min} Q(d)$$

for $0 \leq i \leq k-1$.

If B is only positive semidefinite then the situation when $p_k^T q_k = 0$ can appear for some index $k < n$ so that γ_k in (2.1e) may not be defined (breakdown).

A direction vector satisfying (1.10)–(1.11) can be found using the iterative process (2.1) by the following rules.

(CG1) If $\|d_{k-1}\| < \Delta$ and $p_k^T q_k = 0$ for some $k < m$ (breakdown) then set $d = d_{k-1} + \gamma p_k$ where γ is chosen so that $\|d\| = \Delta$.

(CG2) If $\|d_{k-1}\| < \Delta$ and $\|d_k\| \geq \Delta$ for some $k < m$ then set $d = d_{k-1} + \gamma p_k$ where γ is chosen so that $\|d\| = \Delta$.

(CG3) If either $\|g_k\| \leq \omega \|g\|$ for some $k < m$ or $k = m$ then set $d = d_k$.

Usually $m = n + 3$ since $d_n \in \mathbb{R}^n$ may not be a minimum of the quadratic function (1.4) because of roundoff errors.

The algorithm defined by the iterative process (2.1) and by the rules (CG1)–(CG3) was introduced in [22] and we call it the conjugate gradient trust region (CGTR) method. This algorithm is very natural and simple but it has one disadvantage. Usually $\omega \to 0$ and $\|d\| \to 0 < \Delta$ for $x \to x^*$ (to guarantee superlinear convergence of the Gauss–Newton method) so that the rules (CG1)–(CG3) can require too many CG steps. For dense problems the matrix multiplication (2.1c) consumes $\sim n^2$ operations and if $k \sim n$ then direction determination consume $\sim kn^2 \sim n^3$ operations. On the other hand the Choleski decomposition followed by the solution of the decomposed system consume $\sim (1/3)n^3$ operations which can be considerably less than we had in the previous case. Moreover an exact solution of the equation $Bd + g = 0$ can improve the convergence of the Gauss–Newton method. The simplest method which uses an exact solution of the equation $Bd + g = 0$ is the dog-leg strategy discused in the previous section. But this method can fail applied to ill-conditioned problems. Therefore we recommend a more complicated multiple dog-leg trust region (MDTR) method which uses the iterative process (2.1) together with the following rules.

(MD1) If $\|d_{k-1}\| < \Delta$ and $p_k^T q_k = 0$ for some $k < m$ (breakdown) then set $d = d_{k-1} + \gamma p_k$ where γ is chosen so, that $\|d\| = \Delta$.

(MD2) If $\|d_{k-1}\| < \Delta$ and $\|d_k\| \geq \Delta$ for some $k < m$ then set $d = d_{k-1} + \gamma p_k$ where γ is chosen so that $\|d\| = \Delta$.

(MD3) If either $\|g_k\| \leq \omega \|g\|$ for some $k < m$ or $k = m$ then determine the Gill–Murray decomposition $B + E = LDL^T$ and compute the direction vector d_n such that $LDL^T d_n + g = 0$. If $\|d_n\| \leq \Delta$ then set $d = d_n$. Otherwise set $d = d_k + \gamma(\tau d_n - d_k)$ where $0 < d_k^T g / d_n^T g \leq \tau \leq 1$ and where γ is chosen so that $\|d\| = \Delta$.

Here ω is a small number, $m \leq n$ is a small integer which is usually much less than n and, therefore, than m in (CG1)–(CG3).

The idea of a multiple dog-leg strategy was mentioned in [22], but no proof of efficiency and no implementation details were given there. The multiple dog-leg strategy is based on the following theorem.

Theorem 2.1. Consider the conjugate gradient method applied to the quadratic function (1.4) with the symmetric positive definite matrix B. Let $\|d_i\| < \Delta < \|d_k\|$ for some $0 \leq i < k \leq l$ where l is the integer from Lemma 2.1. Let $0 \leq d_i^T g / d_k^T g \leq \tau \leq 1$. Then the function

$$\varphi(\gamma) = Q(d_i + \gamma(\tau d_k - d_i)) \tag{2.3}$$

is monotonically nonincreasing for $0 \leq \gamma \leq 1$.

Proof. Differentiating (2.3) we obtain

$$\varphi'(\gamma) = (\tau d_k - d_i)^T [B(d_i + \gamma(\tau d_k - d_i)) + g].$$

Let $l \leq n$ be the integer from Lemma 2.1. Then $Bd_l + g = 0$ holds and we can write

$$g = -Bd_l = -Bd_k - \sum_{j=k+1}^{l} Bp_j$$

so that

$$\begin{aligned}
\varphi'(\gamma) &= (\tau d_k - d_i)^T \left(Bd_i + \gamma B(\tau d_k - d_i) - Bd_k - \sum_{j=k+1}^{l} Bp_j \right) \\
&= -(1-\gamma)(\tau d_k - d_i)^T B(\tau d_k - d_i) - (1-\tau)(\tau d_k - d_i)^T Bd_k \\
&\leq (1-\tau)(\tau d_k - d_i)^T \left(g + \sum_{j=k+1}^{l} Bp_j \right) \\
&= (1-\tau)(\tau d_k - d_i)^T g
\end{aligned} \tag{2.4}$$

since $d_i^T Bp_j = d_k^T Bp_j = 0$ for $i < k < j \leq l$ by (2.2a). But

$$(d_k - d_i)^T g = \sum_{j=i+1}^{k} \gamma_j p_j^T g_0 = -\sum_{j=i+1}^{k} \gamma_j g_{j-1}^T g_{j-1} < 0$$

by (2.2c) since $\gamma_j > 0$ for $i < j \leq k$ by (2.1e). Therefore $d_k^T g < d_i^T g \leq d_0^T g = 0$ so that $0 \leq d_i^T g / d_k^T g \leq 1$ and (2.4) imply that $\varphi'(\gamma) \leq 0$ for $0 \leq d_i^T g / d_k^T g \leq \tau \leq 1$.

□

Now we are in the position to give a detailed description of the multiple dog-leg trust region method for nonlinear least squares.

Algorithm 2.1

Data: $0 < \overline{\beta}_1 < \overline{\beta}_2 < 1 < \overline{\gamma}_1 < \overline{\gamma}_2$, $0 < \overline{\rho}_1 < \overline{\rho}_2 < 1$, $0 < \overline{\omega}_1 < \overline{\omega}_2 < 1$, $0 < \overline{\Delta}$, $0 < \overline{\varepsilon}_1 < \overline{\varepsilon}_2$, $\overline{k} \in N$, $\overline{l} \in N$, $\overline{m} \in N$.

Step 1: Choose an initial point $x \in \mathbb{R}^n$. Compute the values $f_i := f_i(x)$ of the functions $f_i : R^n \to \mathbb{R}$, $1 \leq i \leq r$, at the point $x \in \mathbb{R}^n$. Compute the value $F := F(x)$ of the objective function $F : \mathbb{R}^n \to \mathbb{R}$ by (1.1). Set $\Delta = 0$, $m := \min(\overline{m}, n)$ and $k := 1$.

Step 2: Compute the gradients $g_i := g_i(x)$ of the functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $1 \leq i \leq r$, at the point $x \in \mathbb{R}^n$. Determine the matrix $B := B(x)$ by (1.5) and compute the gradient $g := g(x)$ of the objective function $f : R^n \to \mathbb{R}$ by (1.2). If either $F \leq \overline{\varepsilon}_1$ or $\|g\| \leq \overline{\varepsilon}_2$ then stop, otherwise set $l := 1$.

Step 3: If $\Delta = 0$ then set $\Delta := \min(\|g\|^3/g^T Bg,\ 4F/\|g\|,\ \overline{\Delta})$. Compute the vector $d \in \mathbb{R}^n$ by the following subalgorithm.

Step 3.1: Set $d := 0$ $\tilde{g} := g$ and $p := -g$. Set $\rho := \|g\|$ and $i := 1$.

Step 3.2: Set $q := Bp$. If $p^T q \leq 0$ then determine $\gamma > 0$ so that $\|d+\gamma p\| = \Delta$, set $d := d + \gamma p$ and go to Step 4. Otherwise compute $\gamma := \rho/p^T q$. If $\|d + \gamma p\| \geq \Delta$ then determine $\gamma > 0$ so that $\|d + \gamma p\| = \Delta$, set $d := d + \gamma p$ and go to Step 4.

Step 3.3: Set $d := d + \gamma p$ and $g := g + \gamma q$. If either $i = m$ or $\|g\| \leq \overline{\omega}_2\|\tilde{g}\|$ then go to Step 3.4. Otherwise compute $\delta := \|g\|/\rho$, $p := -g + \delta p$, $\rho := \|g\|$, set $i := i + 1$ and go to Step 3.2.

Step 3.4: If $\|g\| \leq \overline{\omega}_1\|\tilde{g}\|$ then go to Step 4. Otherwise compute the Choleski decomposition $B + E = LDL^T$, where E is a small diagonal matrix chosen so that $B + E$ is positive definite and set $s := (LDL^T)^{-1}\tilde{g}$. If $\|s\| \leq \Delta$ then set $d := s$ and go to Step 4. Otherwise compute $\tau := d^T\tilde{g}/s^T\tilde{g}$ and set either $\tau := 1$ (basic dog-leg strategy) or $\tau := \max(\tau, \Delta/\|s\|)$ (modified dog-leg strategy). Set $p := \tau s - d$ and determine $\gamma > 0$ so that $\|d+\gamma p\| = \Delta$. Set $d := d+\gamma p$ and go to Step 4.

Step 4: Set $x^+ := x + d$. Compute the values $f_i^+ := f_i(x^+)$ of the functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $1 \leq i \leq r$, at the point $x^+ \in \mathbb{R}^n$. Compute the value $F^+ := F(x^+)$ of the objective function $F : \mathbb{R}^n \to \mathbb{R}$ by (1.1). Compute the value $Q(d)$ by (1.4) and set $\rho := (F^+ - F)/Q(d)$. When $\rho < \overline{\rho}_1$ then compute $\alpha := (F^+ - F)/d^T\tilde{g}$, $\beta := 1/(2(1-\alpha))$ and set $\Delta := \overline{\beta}_1\|d\|$ if $\beta < \overline{\beta}_1$, $\Delta := \beta\|d\|$ if $\overline{\beta}_1 \leq \beta \leq \overline{\beta}_2$, $\Delta := \overline{\beta}_2\|d\|$ if $\overline{\beta}_2 < \beta$. When $\rho_1 \leq \rho \leq \overline{\rho}_2$ then set $\Delta := \min(\Delta, \overline{\gamma}_2\|d\|)$. When $\overline{\rho}_2 < \rho$ then compute $\Delta := \max(\Delta, \overline{\gamma}_1\|d\|)$ and set $\Delta := \min(\Delta, \overline{\gamma}_2\|d\|, \overline{\Delta})$.

Step 5: If $\rho \leq 0$ and $l \geq \overline{l}$ then stop (too many reductions). If $\rho \leq 0$ and $l < \overline{l}$ then set $l := l + 1$ and go to Step 3. If $\rho > 0$ and $k \geq \overline{k}$ then stop (too many iterations). If $\rho > 0$ and $k < \overline{k}$ then set $x := x^+$, $f_i := f_i^+$, $1 \leq i \leq r$, $F := F^+$, set $k := k + 1$ and go to Step 2.

The maximum number of iterations $\overline{k} \in N$ serves as an alternative termination criterion in the case when the convergence is too slow. The maximum number of reductions $\overline{l} \in N$ serves as a safeguard against a possible infinite cycle. The matrix $B+E$ is used in Step 3.4 to remove the situation when B is singular. The technique for its construction is described in [10]. The matrix E is chosen to keep the diagonal elements of the matrix D no less than $\overline{\omega}_1$.

The global convergence of Algorithm 2.1 is an immediate consequence of Theorem 2.1. Since the CG steps satisfy the conditions (A)-(D) from Section 1 and since $Q(d_i + \gamma(\tau d_n - d_i)) \leq Q(d_i)$ for $0 < \gamma < 1$, we have fulfilled the conditions (1.10)-(1.11). The conditions (1.12)-(1.16) are automatically satisfied for all our algorithms (see Step 4 and Step 5) so that Theorem 1.1 holds.

Now we can present the results of a comparative study of three trust region methods for dense nonlinear least squares problems. The first method, which we call the optimum step trust region (OSTR) method uses the subproblem (1.18) to determine the direction vector $d \in \mathbb{R}^n$ by the procedure given in [18]. The second method is the CGTR method defined by the iterative process (2.1) and by the rules (CG1)-(CG3). More details are given in the next section (see note after Algorithm 3.1). The third method is the MDTR method which is represented by Algorithm 2.1. This algoritm contains several parameters. We have used the values $\overline{\beta}_1 = 0.05$, $\overline{\beta}_2 = 0.75$, $\overline{\gamma}_1 = 2$, $\overline{\gamma}_2 = 10^6$, $\overline{\rho}_1 = 0.1$, $\overline{\rho}_2 = 0.9$, $\overline{\omega}_1 = 10^{-18}$, $\overline{\omega}_2 = 10^{-16}$, $\overline{\Delta} = 10^3$, $\overline{\varepsilon}_1 = 10^{-16}$, $\overline{\varepsilon}_2 = 10^{-8}$, $\overline{k} = 500$, $\overline{l} = 20$, $\overline{m} = 3$.

All test results were obtained by means of the 30 problems given in [17]. Problems 1-19 had the same dimension as in [17] while problems 20-30 were considered with 12 variables.

Table 1 contains a comparative study of various realizations of the MDTR method. The basic realization uses the value $\tau = 1$ in Step 3.4 of Algorithm 2.1 while the modified realization has the value $\tau = \max(d^T\tilde{g}/s^T\tilde{g}, \Delta/\|s\|)$. The standard dog-leg strategy corresponds to the choice $\bar{m} = 1$. Rows of Table 1 correspond to the numbers of CG Steps. The results are presented in the form IT-IF-IG and TIME, where IT is a total number of iterations, IF is a total number of objective values evaluations, IG is a total number of objective gradients evaluations and TIME is a total computational time (over 30 test problems). The asterisk means that 500 iterations did not suffice for problem 18.

Table 1.

$\overline{m}$	basic strategy	modified strategy
1	1274-1598-1304	1302-1526-1332
	0:13.29 (*)	0:13.79 (*)
2	1073-1311-1103	1075-1327-1105
	0:10.82 (*)	0:10.99 (*)
3	596-773-626	576-757-606
	0:06.26	0:06.04
4	622-810-652	619-815-649
	0:06.65	0:06.64
5	649-840-679	650-850-680
	0:06.81	0:06.92

Table 2 contains results for three trust region algorithms (OSTR, CGTR, MDTR). The MDTR algorithm was realized with $\overline{m} = 3$ and with the modified strategy. Rows of Table 2 correspond to individual problems. The results are presented in the form IT-IF-IG where IT is the number of iterations, IF is the number of objective values evaluations and IG is the number of objective gradients evaluations. Also, summary results including total computational time are presented.

Table 2.

Prob.	OSTR	CGTR	MDTR
1	12-15-13	15-20-16	15-20-16
2	30-46-31	36-51-37	36-51-37
3	33-34-34	77-88-78	28-29-29
4	13-16-14	4-5-5	5-7-6
5	6-7-7	7-8-8	7-8-8
6	11-21-12	19-52-20	19-54-20
7	11-14-12	9-11-10	10-12-11
8	5-6-6	6-7-7	4-5-5
9	1-2-2	2-3-3	2-3-3
10	125-141-126	131-138-132	130-153-131
11	39-46-40	36-46-37	31-36-32
12	12-14-13	16-19-17	16-19-17
13	10-11-11	10-11-11	10-11-11
14	69-75-70	56-64-57	38-45-39
15	17-20-18	15-18-16	15-18-16
16	29-66-30	37-73-38	37-73-38
17	21-23-22	24-27-25	18-19-19
18	32-40-33	19-20-20	25-31-26
19	13-15-14	17-20-18	12-14-13
20	7-8-8	8-9-9	10-11-11
21	12-15-13	17-21-18	15-20-16
22	10-11-11	12-14-13	10-11-11
23	20-25-21	26-30-27	21-26-22
24	28-36-29	25-29-26	25-35-26
25	10-11-11	10-11-11	10-11-11
26	9-13-10	16-26-17	8-12-9
27	6-7-7	6-7-7	4-5-5
28	7-8-8	8-9-9	7-8-8
29	2-3-3	4-5-5	3-4-4
30	5-6-6	7-8-8	5-6-6
$\sum$	605-755-635	675-852-705	576-757-606
time	0:06.98	0:06.54	0:06.04

Finally, Table 3 contains similar results for problems 21 – 30 which were considered with 40 variables. The MDTR algorithm was realized with $\overline{m} = 4$ (it was the best choice for 40 variables) and with the modified strategy.

Table 3.

Prob.	OSTR	CGTR	MDTR
21	12-15-13	24-27-25	15-20-16
22	10-11-11	15-17-16	10-11-11
23	16-22-17	28-32-29	18-27-19
24	153-162-154	138-148-139	132-141-133
25	13-14-14	13-14-14	13-14-14
26	12-18-13	17-27-18	23-34-24
27	4-5-5	6-7-7	4-5-5
28	10-11-11	34-35-35	11-12-12
29	2-3-3	4-5-5	**3-4-4**
30	4-5-5	9-10-10	5-6-6
$\sum$	236-266-246	288-322-298	234-274-244
time	1:22.61	1:00.20	0:50.42

3. COMBINED CONJUGATE GRADIENT LANCZOS METHODS FOR SPARSE PROBLEMS

Consider the Lanczos method applied to the quadratic function (1.4). This method is represented by the following process

$$g_0 = g, \quad \beta_0 = \|g_0\|, \quad q_0 = 0 \tag{3.1a}$$

and

$$q_i = g_{i-1}/\beta_{i-1} \tag{3.1b}$$

$$\alpha_i = q_i^T B q_i \tag{3.1c}$$

$$g_i = Bq_i - \alpha_i q_i - \beta_{i-1} q_{i-1} \tag{3.1d}$$

$$\beta_i = \|g_i\| \tag{3.1e}$$

for $i \in N$. If we suppose the matrix B is positive definite then the following well known lemma holds (see [11]).

Lemma 3.1. Consider the Lanczos process (3.1) with a symmetric positive definite matrix B. Let $\beta_k \neq 0$ for some $1 \leq k \leq n$. Then the vectors q_i, $1 \leq i \leq k$, form an orthonormal basis in the Krylov subspace $\mathcal{K}_k(B, g_0) = \text{span}\{B^{i-1}g_0, 1 \leq i \leq k\}$. If we denote $Q_k = [q_1, \ldots, q_k]$, then

$$Q_k^T B Q_k = T_k, \tag{3.2}$$

where

$$T_k = \begin{bmatrix} \alpha_1, & \beta_1, & \ldots, & 0, & 0 \\ \beta_1, & \alpha_2, & \ldots, & 0, & 0 \\ - & - & - & - & - \\ 0, & 0, & \ldots, & \beta_{k-1} & \alpha_k \end{bmatrix} \tag{3.3}$$

is a symmetric tridiagonal matrix.

Consider now a simplification of the subproblem (1.18), namely

$$d = \underset{\|\tilde{d}(\lambda)\| \leq \Delta}{\arg\min}\ Q(\tilde{d}(\lambda)) \tag{3.4}$$

where $\tilde{d}(\lambda) \in \mathcal{K}_m(B, g_0)$ for some $m \leq n$. Since q_i, $1 \leq i \leq m$ form an orthonormal basis in $\mathcal{K}_m(B, g_0)$, we can write $\tilde{d}(\lambda) = Q_m y(\lambda)$ for some $y(\lambda) \in \mathbb{R}^m$, so that

$$\begin{aligned} Q(\tilde{d}(\lambda)) &= \frac{1}{2} y^T(\lambda)\, Q_m^T B Q_m y(\lambda) + g^T Q_m y(\lambda) \\ &= \frac{1}{2} y^T(\lambda)\, T_m y(\lambda) + \beta_0 e_1^T y(\lambda) \\ &\triangleq \tilde{Q}(y(\lambda)) \end{aligned}$$

holds by (3.1a) and (3.2) (e_1 is the first column of the unit matrix). Moreover $\|\tilde{d}(\lambda)\| = \|Q_m y(\lambda)\| = \|y(\lambda)\|$ since the matrix Q_m is orthogonal. Therefore (3.4) can be rewritten in the form $d = Q_m y$ where

$$y = \underset{\|y(\lambda)\| \leq \Delta}{\arg\min}\ \tilde{Q}(y(\lambda)) \tag{3.5}$$

This subproblem can be solved by the standard Newton method which is represented by the following process

$$\lambda_1 = 0 \tag{3.6a}$$

and

$$T_m + \lambda_i I = L_i D_i L_i^T \tag{3.6b}$$

$$L_i D_i L_i^T y_i + \beta_0 e_1 = 0 \tag{3.6c}$$

$$L_i z_i = y_i \tag{3.6d}$$

$$\lambda_{i+1} = \lambda_i + \frac{\|y_i\|^2}{z_i^T D_i^{-1} z_i} \left(\frac{\|y_i\| - \tilde{\delta}\Delta}{\tilde{\delta}\Delta} \right) \tag{3.6e}$$

for $i \in N$. This iterative process is finished if $\|y_i\| \leq \Delta$ for some $i \in N$. Then we set $y = y_i$. The parameter $0 < \tilde{\delta} < 1$ is usually close to 1.

The main advantage of the subproblem (3.5) is the fact that the matrix T_m is symmetric and tridiagonal, so both the Choleski decomposition (3.6b) and solution of the decomposed system (3.6c) consume $O(m)$ operations only.

The simplest method which uses the subproblem (3.5) is the Lanczos conjugate gradient trust region (LCTR) method. This method consists in choosing a fixed (usually small) number m of Lancozs steps. After m steps of the form (3.1) we solve the subproblem (3.5) by the Newton method (3.6) to obtain the parameter $\lambda \geq 0$. Finally we approximately solve the equation $(B + \lambda I)\, d + g = 0$ by the inexact CG method. More details are given in the following algorithm.

Algorithm 3.1 (LCTR)

Data: $0 < \overline{\beta}_1 < \overline{\beta}_2 < 1 < \overline{\gamma}_1 < \overline{\gamma}_2$, $0 < \overline{\rho}_1 < \overline{\rho}_2 < 1$, $0 < \overline{\omega} < 1$, $0 < \overline{\delta} < 1$, $0 < \overline{\Delta}$, $0 < \overline{\lambda}$, $0 < \overline{\varepsilon}_1 < \overline{\varepsilon}_2$, $\overline{k} \in N$, $\overline{l} \in N$, $\overline{m} \in N$.

Step 1: The same as Step 1 of Algorithm 2.1.

Step 2: The same as Step 2 of Algorithm 2.1.

Step 3: If $\Delta = 0$ then set $\Delta := \min(\|g\|^3/g^T B g,\ 4F/\|g\|,\ \overline{\Delta})$. Set $\omega := \min\left(\sqrt{\|g\|},\ 1/k,\ \overline{\omega}\right)$ and compute the vector $d \in \mathbb{R}^n$ by the following subalgorithm.

Step 3.1: Compute a symmetric tridiagonal matrix T of the order m using the m steps of the Lanczos process (3.1). Set $\lambda := 0$ and $i := 1$.

Step 3.2: If $\lambda \geq \overline{\lambda}$ then set $\lambda := \overline{\lambda}$ and go to Step 3.4. Otherwise compute the Choleski decomposition $T + \lambda I = LDL^T$ and solve the equation $LDL^T y + \beta_0 e_1 = 0$. If either $\|y\| \leq \Delta$ or $i \geq 5$ then go to Step 3.4. Otherwise go to Step 3.3.

Step 3.3: Solve the equation $Lz = y$ set $\lambda := \lambda + (\|y\|/z^T D^{-1} z)(\|y\|/(\overline{\delta}\Delta) - 1)$, set $i := i + 1$ and go to Step 3.2.

Step 3.4: Set $d := 0$, $p := -g$, $\rho := \|g\|^2$, $\rho_0 := \rho$ and $i = 1$.

Step 3.5: Compute $q := (B + \lambda I)p$. If $p^T q \leq 0$ then determine $\gamma > 0$ so that $\|d + \gamma p\| = \Delta$, set $d := d + \gamma p$ and go to Step 4. Otherwise compute $\gamma := \rho/p^T q$. If $\|d + \gamma p\| \geq \Delta$ then determine $\gamma > 0$ so that $\|d + \gamma p\| = \Delta$ set $d := d + \gamma p$ and go to Step 4.

Step 3.6: Set $d := d + \gamma p$ and $g := g + \gamma q$. If either $i = n + 3$ or $\|g\|^2 \leq \omega^2 \rho_0$ then go to Step 4. Otherwise compute $\delta := \|g\|/\rho$, $p := -g + \delta p$, $\rho := \|g\|^2$, set $i := i + 1$ and go to Step 3.5.

Step 4: The same as Step 4 in Algorithm 2.1

Step 5: The same as Step 5 in Algorithm 2.1

Note that if we omit Steps 3.1 – 3.3 and set $\lambda := 0$ in Step 3.5 of Algorithm 3.1 we obtain exacly the CGTR method proposed in [22] and tested in the previous section.

Global convergence of Algorithm 3.1 follows from the fact that the direction vector d is an inexact CG solution of the equation $(B + \lambda I)d + g = 0$. Since $\lambda \leq \overline{\lambda}$ (see Step 3.2), the matrix $B + \lambda I$ is bounded from above whenever assumptions of Theorem 1.1 hold. Using the matrix $B + \lambda I$ instead of B in the theory proposed in [20] we get the required result.

The main advantage of the LCTR method is the fact that it does not use aditional vectors. On the other hand, it uses additional matrix-vector multiplications in the Lanczos part of the algorithm. This disadvantage can be removed using the relation between the conjugate gradient and the Lanczos method. This relation is based on the fact that both the set $\{g_{i-1},\ 1 \leq i \leq k\}$ given by (2.1) and the set $\{q_i,\ 1 \leq i \leq k\}$ given by (3.1) form ortogonal bases in the Krylov subspace $\mathcal{K}_k(B, g_0)$. Therefore the vectors g_{i-1}, $1 \leq i \leq k$, have to be collinear with the vectors q_i, $1 \leq i \leq k$. A more

detailed analysis, which is proposed for instance in [11], gives the formulas

$$\alpha_i = \frac{1}{\gamma_i} + \frac{\delta_{i-1}}{\gamma_{i-1}} \tag{3.7a}$$

$$\beta_i = \frac{\sqrt{\delta_i}}{\gamma_i} \tag{3.7b}$$

for $1 \leq i \leq l$, where l is the index from Lemma 2.1. Moreover

$$q_i = (-1)^{i-1} g_{i-1} / \|g_{i-1}\| \tag{3.8}$$

for $1 \leq i \leq l$.

The formulas (3.7) and (3.8) allow us to construct another combined method which we call the conjugate gradient Lanczos trust region (CLTR) method. This method consists in choosing a fixed (usually small) number m of CG-steps. After m steps of the form (2.1), which are followed by the construction of the matrix T and by the computation of the vectors q_i, $1 \leq i \leq m$ using (3.7)–(3.8), we proceed as follows. If $d_m < \Delta$ then we continue in CG steps to fullfill the condition $\|g_k\| \leq \omega \|g\|$. If $d_m \geq \Delta$ then we solve the subproblem (3.5) and set $d = Qy$. More details are given in the following algorithm.

Algorithm 3.2 (CLTR)

Data: $0 < \overline{\beta}_1 < \overline{\beta}_2 < 1 < \overline{\gamma}_1 < \overline{\gamma}_2$, $0 < \overline{\rho}_1 < \overline{\rho}_2 < 1$, $0 < \overline{\omega} < 1$, $0 < \overline{\delta} < 1$, $0 < \overline{\Delta}$, $0 < \overline{\varepsilon}_1 < \overline{\varepsilon}_2$, $\overline{k} \in N$, $\overline{l} \in N$, $\overline{m} \in N$.

Step 1: The same as Step 1 of Algorithm 2.1.

Step 2: The same as Step 1 of Algorithm 2.1.

Step 3: If $\Delta = 0$ then set $\Delta := \min\left(\|g\|^3 / g^T B g,\ 4F/\|g\|,\ \overline{\Delta}\right)$. Set $\omega := \min\left(\sqrt{\|g\|},\ 1/k,\ \overline{\omega}\right)$ and compute the vector $d \in \mathbb{R}^n$ by the following subalgorithm.

Step 3.1: Set $d := 0$, $p := -g$, $\rho := \|g\|^2$, $\rho_0 := \rho$ and $i = 1$. Compute the first Lanczos vector by (3.8).

Step 3.2: Compute $q := Bp$. If $p^T q \leq 0$ then determine $\gamma > 0$ so that $\|d + \gamma p\| = \Delta$, set $d := d + \gamma p$ and go to Step 4. Otherwise compute $\gamma := \rho / p^T q$. If $\|d + \gamma p\| \geq \Delta$ and $i > m$ then determine $\gamma > 0$ so that $\|d + \gamma p\| = \Delta$, set $d := d + \gamma p$ and go to Step 4.

Step 3.3: Set $d := d + \gamma p$ and $g := g + \gamma q$. If $\|d + \gamma p\| \geq \Delta$ and either $i = m$ or $\|g\|^2 \leq \omega^2 \rho_0$ then compute the corresponding column of the matrix T by (3.7) and go to Step 3.5. If $\|d + \gamma p\| < \Delta$ and either $i = n + 3$ or $\|g\| \leq \omega^2 \rho_0$ then go to Step 4.

Step 3.4: Set $\delta := \|g\| / \rho$. If $i < m$ then compute the corresponding column of the matrix T by (3.7) and the corresponding Lanczos vector by (3.8). Set $p := -g + \delta p$, $\rho := \|g\|^2$. Set $i := i + 1$ and go to Step 3.2.

Step 3.5: Set $\lambda := 0$ and $i := 1$.

Step 3.6: Compute the Choleski decomposition $T + \lambda I = LDL^T$ and solve the equation $LDL^T y + \beta_0 e_1 = 0$. If either $\|y\| \leq \Delta$ or $i = 5$ then go to Step 3.8. Otherwise go to Step 3.7.

Step 3.7: Solve the equation $Lz = y$, set $\lambda := \lambda + (\|y\|/z^T D^{-1} z)(\|y\|/(\overline{\delta}\Delta) - 1)$, set $i := i + 1$ and go to Step 3.6.

Step 3.8: Set $d := Qy$ where Q is the matrix whose columns are the Lancozs vectors.

Step 4: The same as Step 4 in Algorithm 2.1.

Step 5: The same as Step 5 in Algorithm 2.1.

Global convergence of Algorithm 3.2 follows from properties of the CG steps (conditions (A)–(D) from Section 1) and from properties of the subproblem (3.4). If we terminate computations before the subproblem (3.4) is solved, then we use the same direction vector as in the CGTR method which satisfies all necessary conditions. In the opposite case, if the subproblem (3.4) is solved, then (1.11) holds since $Q(d)$ cannot be greater than the value obtained in the first CG step.

The main advantage of the CLTR method is the fact that it does not use additional matrix-vector products. On the other hand, it uses additional n-dimensional (Lanczos) vectors.

Now we can present the results of a comparative study of three trust region methods for sparse nonlinear least squares problems. The first method is the CGTR method (Algorithm 3.1 without Steps 3.1–3.3 and with $\lambda = 0$ in Step 3.5) the LCTR method (Algorithm 3.1) and the CLTR method (Algorithm 3.2). We have used the values $\overline{\beta}_1 = 0.05$, $\overline{\beta}_2 = 0.75$, $\overline{\gamma}_1 = 2$, $\overline{\gamma}_2 = 10^6$, $\overline{\rho}_1 = 0.1$, $\overline{\rho}_2 = 0.9$, $\overline{\omega} = 0.4$, $\overline{\delta} = 0.9$, $\overline{\Delta} = 10^3$, $\overline{\lambda} = 10^6$, $\overline{\varepsilon}_1 = 10^{-16}$, $\overline{\varepsilon}_2 = 10^{-8}$, $\overline{k} = 500$, $\overline{l} = 20$, $\overline{m} = 5$, in all algorithms. All test problems were obtained by means of the 10 problems given in [14] which had 100 variables.

1. Chained Rosenbrock function.
2. Chained Wood function.
3. Chained Powell singular function.
4. Chained Cragg and Levy function.
5. Generalized Broyden tridiagonal function.
6. Generalized Broyden banded function.
7. Extended Freudenstein and Roth function.
8. Wright and Holt zero residual problem.
9. Toint quadratic merging problem.
10. Chained exponential system.

Table 4 contains results for three trust region algorithms (CGTR, LCTR, CLTR) in the case where the gradients are given analytically. Rows of Table 4 correspond to individual problems. The results are presented in the same form as in Table 2.

Table 4.

Prob.	CGTR	LCTR	CLTR
1	122-125-123	125-131-126	124-130-125
2	145-175-146	70-78-71	142-170-143
3	19-20-20	19-20-20	19-20-20
4	151-172-152	71-100-72	79-101-80
5	11-12-12	10-11-11	11-12-12
6	11-12-12	11-12-12	11-12-12
7	42-80-43	44-81-45	40-73-41
8	17-18-18	19-20-20	17-18-18
9	56-82-57	56-82-57	56-82-57
10	29-69-30	30-61-31	29-64-30
$\sum$	603-765-613	455-596-465	528-682-538
Time	1:05.20	1:03.80	1:00.80

Table 5 contains similar results, but now for the case where the gradients are computed numerically (numbers of objective gradients evaluations are zero). This case is more favourable for matrix vector products so that the efficiency of the LCTR and the CLTR methods are more clear.

Table 5.

Prob.	CGTR	LCTR	CLTR
1	122-309	125-320	155-418
2	148-427	71-199	141-408
3	19-60	19-60	19-60
4	204-561	66-204	61-195
5	11-47	10-43	11-47
6	11-94	11-94	11-94
7	42-169	39-155	38-149
8	18-57	19-60	18-57
9	57-310	53-290	53-290
10	26-125	29-133	27-129
$\sum$	658-2159	442-1558	534-1846
Time	1:40.95	1:22.82	1:28.32

(Received January 31, 1995.)

REFERENCES

[1] M. Al-Baali and R. Fletcher: Variational methods for nonlinear least squares. J. Optim. Theory Appl. *36* (1985), 405–421.

[2] R. H. Byrd, R. B. Schnabel and G. A. Shultz: Approximate solution of the trust region problem by minimization over two–dimensional subspaces. Math. Programming *40* (1988), 247–263.

[3] J. E. Dennis: Some computational techniques for the nonlinear least squares problem. In: Numerical solution of nonlinear algebraic equations (G. D. Byrne, C. A. Hall, eds.), Academic Press, London 1974.
[4] J. E. Dennis and H. H. W. Mei: An Unconstrained Optimization Algorithm which Uses Function and Gradient Values. Research Report No. TR-75-246, Department of Computer Science, Cornell University 1975.
[5] J. E. Dennis, D. M. Gay and R. E. Welsch: An adaptive nonlinear least-squares algorithm. ACM Trans. Math. Software *7* (1981), 348–368.
[6] J. E. Dennis and R. B. Schnabel: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice–Hall, Englewood Cliffs, New Jersey 1983.
[7] R. Fletcher: A Modified Marquardt Subroutine for Nonlinear Least Squares. Research Report No. R-6799, Theoretical Physics Division, A.E.R.E. Harwell 1971.
[8] R. Fletcher: Practical Methods of Optimization. Second edition. J. Wiley & Sons, Chichester 1987.
[9] R. Fletcher and C. Xu: Hybrid methods for nonlinear least squares. IMA J. Numer. Anal. *7* (1987), 371–389.
[10] P. E. Gill and W. Murray: Newton type methods for unconstrained and linearly constrained optimization. Math. Programming *7* (1974), 311–350.
[11] G. H. Golub and C. F. Van Loan: Matrix Computations. Second edition. Johns Hopkins University Press, Baltimore 1989.
[12] M. R. Hestenes: Conjugate Direction Methods in Optimization. Springer–Verlag, Berlin 1980.
[13] K. Levenberg: A method for the solution of certain nonlinear problems in least squares. Quart. Appl. Math. *2* (1944), 164–168.
[14] L. Lukšan: Inexact trust region method for large sparse nonlinear least squares. Kybernetika *29* (1993), 305–324.
[15] L. Lukšan: Hybrid methods for large sparse nonlinear least squares. J. Optim. Theory Appl. *89* (1996), to appear.
[16] D. W. Marquardt: An algorithm for least squares estimation of non-linear parameters. SIAM J. Appl. Math. *11* (1963), 431–441.
[17] J. J. Moré, B. S. Garbow and K. E. Hillström: Testing unconstrained optimization software. ACM Trans. Math. Software 7 (1981), 17–41.
[18] J. J. Moré and D. C. Sorensen: Computing a trust region step. SIAM J. Sci. Statist. Comput. *4* (1983), 553–572.
[19] M. J. D. Powell: A new algorithm for unconstrained optimization. In: Nonlinear Programming (J. B. Rosen, O. L. Mangasarian, K. Ritter, eds.), Academic Press, London 1970.
[20] M. J. D. Powell: On the global convergence of trust region algorithms for unconstrained minimization. Math. Programming *29* (1984), 297–303.
[21] G. A. Shultz, R. B. Schnabel and R. H. Byrd: A family of trust–region–based algorithms for unconstrained minimization with strong global convergence properties. SIAM J. Numer. Anal. *22* (1985), 47–67.
[22] T. Steihaug: The conjugate gradient method and trust regions in large-scale optimization. SIAM J. Numer. Anal. *20* (1983), 626–637.

Ing. Ladislav Lukšan, DrSc., Ústav informatiky a výpočetní techniky AV ČR (Institute of Computer Science – Academy of Sciences of the Czech Republic), Pod vodárenskou věží 2, 182 07 Praha. Czech Republic.