Institute of Computer Science
Academy of Sciences of the Czech Republic

A shifted Steihaug-Toint method for computing a trust-region step

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček

Technical report No. 914

November 2004

Pod Vodárenskou věží 2, 182 07 Prague 8 phone: +420 2 688 42 44, fax: +420 2 858 57 89,
e-mail:e-mail:ics@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

A shifted Steihaug-Toint method for computing a trust-region step

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček [1]

Technical report No. 914

November 2004

Abstract:

Trust-region methods are globally convergent techniques widely used, for example, in connection with the Newton's method for unconstrained optimization. The most commonly-used iterative approaches for solving the trust-region subproblems are the Moré-Sorensen method that uses complete matrix decompositions and the Steihaug-Toint method based on conjugate gradient iterations. We propose a method which combines both of these approaches. Using the small-size Lanczos matrix, we apply the Moré-Sorensen method to a small-size trust-region subproblem to compute an approximation of the Lagrange multiplier. Then we solve the shifted system by the Steihaug-Toint method. This paper contains a complete theory concerning properties of the Lagrange multipliers and proves that the new method is globally convergent in the preconditioned case. Finally, results of extensive computational experiments are presented, which demonstrate efficiency of the new method in the case when a suitable preconditioning is used.

Keywords:
Unconstrained optimization, large-scale optimization, trust-region methods, trust-region subproblems, conjugate gradients, Krylov subspaces, computational experiments.

[1] This work was supported by the Grant Agency of the Czech Academy of Sciences, project code IAA1030405, and by the Ministry of Education of the Czech Republic, project code MSM 242200002.

Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8 and Technical University of Liberec, Hálkova 6, 461 17 Liberec

1 Introduction

Basic optimization methods can be realized in various ways which differ in direction determination and step-size selection. Line-search and trust-region globalization strategies are most popular. Trust-region methods [11] can be advantageously used when the Hessian matrix of the objective functions (or its approximation) is indefinite, ill conditioned or singular. This situation often arises in connection with the Newton's method for general objective function (indefiniteness) or with the Gauss-Newton's method for nonlinear least-squares problems (near singularity).

Consider the problem

$$\min F(x), \quad x \in \mathcal{R}^n,$$

where $F : \mathcal{R}^n \to \mathcal{R}$ is twice continuously differentiable objective function. Basic optimization methods (trust-region and line-search methods) generate points $x_i \in \mathcal{R}^n, i \in \mathcal{N}$, in such a way that x_1 is arbitrary and

$$x_{i+1} = x_i + \alpha_i d_i, \quad i \in \mathcal{N}, \tag{1}$$

where $d_i \in \mathcal{R}^n$ are direction vectors and $\alpha_i > 0$ are step sizes.

For a description of trust-region methods we define the quadratic function

$$Q_i(d) = \frac{1}{2} d^T B_i d + g_i^T d$$

which locally approximates the difference $F(x_i + d) - F(x_i)$, the vector

$$\omega_i(d) = (B_i d + g_i)/\|g_i\|$$

for the accuracy of computed direction, and the number

$$\rho_i(d) = \frac{F(x_i + d) - F(x_i)}{Q_i(d)}$$

for the ratio of actual and predicted decrease of the objective function. Here $g_i = g(x_i) = \nabla F(x_i)$ and $B_i \approx \nabla^2 F(x_i)$ is an approximation of the Hessian matrix of function at the point $x_i \in \mathcal{R}^n$.

Trust-region methods are based on approximate minimizations of $Q_i(d)$ on the balls $\|d\| \le \Delta_i$ followed by updates of radii $\Delta_i > 0$. Thus direction vectors $d_i \in \mathcal{R}^n$ are chosen to satisfy the conditions

$$\begin{aligned} \|d_i\| &\le \Delta_i, & (2)\\ \|d_i\| &< \Delta_i \quad \Rightarrow \quad \|\omega_i(d_i)\| \le \overline{\omega}, & (3)\\ -Q_i(d_i) &\ge \underline{\sigma} \|g_i\| \min(\|d_i\|, \|g_i\|/\|B_i\|), & (4) \end{aligned}$$

where $0 \le \overline{\omega} < 1$ and $0 < \underline{\sigma} < 1$. Step sizes $\alpha_i \ge 0$ are selected so that

$$\begin{aligned} \rho_i(d_i) \le 0 \quad &\Rightarrow \quad \alpha_i = 0, & (5)\\ \rho_i(d_i) > 0 \quad &\Rightarrow \quad \alpha_i = 1. & (6) \end{aligned}$$

Trust-region radii $0 < \Delta_i \le \overline{\Delta}$ are chosen in such a way that $0 < \Delta_1 \le \overline{\Delta}$ is arbitrary and

$$\begin{aligned} \rho_i(d_i) < \underline{\rho} \quad &\Rightarrow \quad \underline{\beta}\|d_i\| \le \Delta_{i+1} \le \overline{\beta}\|d_i\|, & (7)\\ \rho_i(d_i) \ge \underline{\rho} \quad &\Rightarrow \quad \Delta_i \le \Delta_{i+1} \le \overline{\Delta}, & (8) \end{aligned}$$

where $0 < \underline{\beta} \leq \overline{\beta} < 1$ and $0 < \underline{\rho} < 1$. The following theorem, see [13], establishes the global convergence of trust-region methods.

Theorem 1 *Let the objective function $F : \mathcal{R}^n \to \mathcal{R}$ be bounded from below and have bounded second-order derivatives. Consider the trust-region method (2)-(8) and denote $M_i = \max(\|B_1\|, \ldots, \|B_i\|)$, $i \in \mathcal{N}$. If*

$$\sum_{i \in \mathcal{N}} \frac{1}{M_i} = \infty, \tag{9}$$

then $\liminf_{i \to \infty} \|g_i\| = 0$.

Note that (9) is satisfied if there exist a constant $\overline{B}$ and an infinite set $\mathcal{M} \subset \mathcal{N}$ such that $\|B_i\| \leq \overline{B}\ \forall i \in \mathcal{M}$.

A crucial part of each trust region method is the direction determination. There are various commonly known methods for computing direction vectors satisfying conditions (2)-(4) which we now mention briefly. To simplify the notation, we omit the index i and write $B \succeq 0$ or $B \succ 0$ to indicate that the matrix B is positive semidefinite or positive definite, respectively.

The most sophisticated method is based on a computation of the optimal locally constrained step. In this case, the vector $d \in R^n$ is obtained by solving the subproblem

$$\text{minimize} \quad Q(d) = \frac{1}{2} d^T B d + g^T d \quad \text{subject to} \quad \|d\| \leq \Delta. \tag{10}$$

Necessary and sufficient conditions for this solution are

$$\|d\| \leq \Delta, \quad (B + \lambda I)d = -g, \quad B + \lambda I \succeq 0, \quad \lambda \geq 0, \quad \lambda(\Delta - \|d\|) = 0. \tag{11}$$

The Moré-Sorensen method [10] is based on solving the nonlinear equation $1/\|d(\lambda)\| = 1/\Delta$ with $(B + \lambda I)d(\lambda) + g = 0$ by the Newton's method, possibly the modified Newton's method [17] using the Choleski decomposition of $B + \lambda I$. This method is very robust but requires 2-3 Choleski decompositions for one direction determination on the average.

Simpler methods are based on minimization of $Q(d)$ on the two-dimensional subspace containing the Cauchy step $d_C = -(g^T g / g^T B g) g$ and the Newton step $d_N = -B^{-1} g$. The most popular is the dogleg method [3],[12], where $d = d_N$ if $d_N \leq \Delta$ and $d = (\Delta / \|d_C\|) d_C$ if $\|d_C\| \geq \Delta$. In the remaining case, d is a combination of d_C and d_N such that $\|d\| = \Delta$. This method requires only one Choleski decomposition for one direction determination.

If B is not sufficiently small or sparse, or explicitly available, then it is either too expensive or not possible to compute its Choleski factorization. In this case, methods based on matrix-vector multiplications are more convenient.

Steihaug [18] and Toint [19] proposed a technique for finding an approximate solution of (10) that do not require exact solution of a linear system but still produce an improvement on the Cauchy point. This implementation is based on the conjugate gradient algorithm [11] for solving the linear system $Bd = -g$. We either obtain an unconstrained solution with a sufficient precision or stop on the trust-region boundary. The latter possibility occurs if either a negative curvature is encountered or the constraint is violated. This method is based on the fact that $Q(d_{k+1}) < Q(d_k)$ and $\|d_{k+1}\| > \|d_k\|$ hold in the subsequent CG iterations if the CG coefficients are positive and preconditioning is not used. Note that the inequality $\|d_{k+1}\| > \|d_k\|$ does not hold in general if a general preconditioner C (symmetric and positive definite) is used. In this case, $\|d_{k+1}\|_C > \|d_k\|_C$ (where $\|d_k\|_C^2 = d_k^T C d_k$) holds.

There are two possibilities how the Steihaug-Toint method can be preconditioned. The first way uses norms $\|d_i\|_{C_i}$ (instead of $\|d_i\|$) in (2)–(8), where C_i are preconditioners chosen. This possibility has been tested in [5] and showed that such a way is not always efficient. This is caused by the fact that norms $\|d_i\|_{C_i}$, $i \in \mathcal{N}$, vary considerably in the major iterations and preconditioners C_i, $i \in \mathcal{N}$, can be ill-conditioned. The second way uses Euclidean norms in (2)–(8) even if arbitrary preconditioners C_i, $i \in \mathcal{N}$, are used. In this case the trust region can be leaved prematurely and the direction vector obtained can be farther from the optimal locally-constrained step than that obtained without preconditioning. This shortcoming is usually compensated by the rapid convergence of the preconditioned CG method. Our computational experiments indicated that the second way is more efficient in general. Thus we confine our attention to this technique in the subsequent considerations.

The CG steps can be combined with the Newton step $d_N = -B^{-1}g$ in the multiple dogleg method [18]. Let $k \ll n$ (usually $k = 5$) and d_k be a vector obtained after k CG steps of the Steihaug-Toint method. If $\|d_k\| < \Delta$, we use d_k instead of $d_C = d_1$ in the dogleg method.

Although the Steihaug-Toint method is certainly the most commonly used in trust region methods, the resulting direction vector may be rather far from the optimal solution even in the unpreconditioned case. This drawback can be overcome by using the Lanczos process [5], as we now explain. Initially, the conjugate gradient algorithm is used as in the Steihaug-Toint method. At the same time, the Lanczos tridiagonal matrix is constructed from the CG coefficients. If a negative curvature is encountered or the constraint is violated, we switch to the Lanczos process. In this case, $d = Z\tilde{d}$, where $\tilde{d}$ is obtained by minimizing the quadratic function

$$\frac{1}{2}\tilde{d}^T T \tilde{d} + \|g\| e_1^T \tilde{d} \tag{12}$$

subject to $\|\tilde{d}\| \le \Delta$. Here $T = Z^T B Z$ (with $Z^T Z = I$) is the Lanczos tridiagonal matrix and e_1 is the first column of the unit matrix. Using preconditioner C, the preconditioned Lanczos method generates basis such that $Z^T C Z = I$. Thus we have to use norms $\|d_i\|_{C_i}$ in (2)–(8), i.e., the first way of preconditioning, which can be inefficient when C_i vary considerably in the trust-region iterations or are ill-conditioned.

There are several recently developed techniques for large scale trust region subproblems that are not based on conjugate gradients. Hager [6] developed a method that solves (10) with the additional constraint that d is contained in a low-dimensional subspaces. The subspaces are modified in successive iterations to obtain quadratic convergence to the optimum and they are designed to contain both the prior iterate and the iterate that is generated by applying one step of the sequential quadratic programming algorithm [1] to (10). At first the Lanczos method is used to generate an orthonormal basis for the $k-$dimensional Krylov subspace (usually $k = 10$). Then the problem (10) is reduced to the $k-$dimensional one to obtain an initial iterate. The main loop consists in seeking vectors $d \in \mathcal{S}$ where $\mathcal{S}$ contains the following four vectors:

- The previous iterate. This causes that the value of the cost function can only decrease in consecutive iterations.

- The multiple $Bd + g$ of the cost function gradient. It ensures descent if the current iterate does not satisfy the first-order optimality conditions.
- An estimate for an eigenvector of B associated with the smallest eigenvalue. It will dislodge the iterates from a nonoptimal stationary point.
- The SQP iterate. The convergence is locally quadratic if the subspace $\mathcal{S}$ contains the iterate generated by one step of the sequential quadratic programming algorithm applied to (10).

An orthonormal basis for the subspace $\mathcal{S}$ is constructed, the original problem (10) is reduced to the 4−dimensional one, and a new iterate d is found via this small subproblem. The iteration is finished as soon as $\|(B+\lambda I)d+g\|$ with Lagrange multiplier λ is smaller than some sufficiently small tolerance (usually 10^{-4} or 10^{-6} suffices). The SQP method is equivalent to the Newton's method applied to the nonlinear system

$$(B+\lambda I)d+g=0, \quad \frac{1}{2}d^T d-\frac{1}{2}\Delta^2=0.$$

The Newton iterate can be expressed in the following way:

$$d_{SQP}=d+z, \quad \lambda_{SQP}=\lambda+\nu,$$

where z and ν are solutions of the linear system

$$\begin{aligned} (B+\lambda I)z+d\,\nu &= -\big((B+\lambda I)d+g\big), \\ d^T z &= 0, \end{aligned}$$

which can be solved by preconditioned MINRES or CG methods. The latter case with the incomplete Choleski-type decomposition of matrix $B+\lambda I$ has shown to be more efficient in practice.

Another approach for finding the direction vector d is based on the idea of Sorensen [15],[16]. Consider the bordered matrix

$$B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & B \end{pmatrix}$$

where α is a real number and observe that

$$\frac{\alpha}{2}+Q(d)=\frac{1}{2}(1,d^T)B_\alpha \begin{pmatrix} 1 \\ d \end{pmatrix}.$$

Therefore, there exists a value of the parameter α such that we can rewrite problem (10) as

$$\text{minimize} \quad \frac{1}{2}d_\alpha^T B_\alpha d_\alpha \quad \text{subject to} \quad \|d_\alpha\|^2 \le 1+\Delta^2, \quad e_1^T d_\alpha = 1, \tag{13}$$

where $d_\alpha = (1, d^T)$ and e_1 is the first canonical unit vector in $\mathcal{R}^{n+1}$. This formulation suggests that we can find the desired solution in terms of an eigenpair of B_α. The resulting algorithm is superlinearly convergent.

Several more techniques for computing a trust region step concerning semidefinite programming approach can be found in [4], [14].

In this paper, we apply the Steihaug-Toint method to the subproblem

$$\text{minimize} \quad \tilde{Q}(d) = Q_{\tilde{\lambda}}(d) = \frac{1}{2} d^T (B + \tilde{\lambda} I) d + g^T d \quad \text{s.t.} \quad \|d\| \le \Delta. \tag{14}$$

The number $\tilde{\lambda} \ge 0$, which approximates λ in (11), is found by solving a small-size subproblem of type (12) with the tridiagonal matrix T obtained by using a small number of Lanczos steps. This method, like method [5], combines good properties of the Moré-Sorensen and the Steihaug-Toint methods. Moreover, it can be successfully preconditioned by the second way. The point on the trust-region boundary obtained by this method is usually closer to the optimal solution in comparison with the point obtained by the original Steihaug-Toint method. We restrict our attention to problems with large dimensions.

The paper is organized as follows. Section 2 contains theoretical background concerning this method with global convergence proved in Section 3. Computational results are given in Section 4 and some concluding remarks are reported in Section 5.

2 A shifted Steihaug-Toint method

A shifted Steihaug-Toint method differs from the standard one by using the shifted subproblem (14), where the number $\tilde{\lambda}$ approximates λ in (11). The number $\tilde{\lambda}$ should be chosen in such a way that $\tilde{\lambda} = 0$ if $\|d\| < \Delta$, where d is a solution of (10). This is true if $0 \le \tilde{\lambda} \le \lambda$, since $\lambda = 0$ if $\|d\| < \Delta$. In this section, we prove a theorem, which allows us to obtain a suitable $\tilde{\lambda}$ by a limited number of the Lanczos steps. To make the proof clearer, we first prove four lemmas. The first lemma shows a simple property of the conjugate gradient method, the second one compares Krylov subspaces of the matrices B and $B + \lambda I$. The third lemma relates properties of matrices $B_1 - B_2$ and $B_2^{-1} - B_1^{-1}$ and the last one states a relation between sizes of the Lagrange multipliers and the norms of directions vectors. In this section, we denote by $\mathcal{K}_k = \text{span}\{g, Bg, \dots, B^{k-1}g\}$ the Krylov subspace of dimension k defined by the matrix B and the vector g, and by $Z_k \in \mathcal{R}^{n \times k}$ a matrix whose columns form an orthonormal basis for $\mathcal{K}_k$.

Lemma 1 *Let B be a symmetric and positive definite matrix, let*

$$\mathcal{K}_j = \text{span}\{g, Bg, \dots, B^{j-1}g\}, \quad j \in \{1, \dots, n\},$$

be the j-th Krylov subspace given by the matrix B and the vector g. Let

$$d_j = \arg\min_{d \in \mathcal{K}_j} Q(d), \quad \textit{where} \quad Q(d) = \frac{1}{2} d^T B d + g^T d.$$

If $1 \le k \le l \le n$, then

$$\|d_k\| \le \|d_l\|.$$

Especially

$$\|d_k\| \le \|d_n\|, \quad \textit{where} \quad d_n = \arg\min_{d \in \mathcal{R}^n} Q(d).$$

Proof. The assertion of the lemma holds for vectors d_j, $j \ge 1$, generated by the conjugate gradient method starting from $d_0 = 0$ (see [18]). These vectors are minimizers of $Q(d)$ on Krylov subspaces $\mathcal{K}_j$, $j \ge 1$. □

Corollary 2 *Let B be symmetric and positive definite and let $Z_k \in \mathcal{R}^{n\times k}$ be a matrix whose columns form an orthonormal basis for $\mathcal{K}_k$. Then*

$$g^T Z_k (Z_k^T B Z_k)^{-2} Z_k^T g \le g^T B^{-2} g.$$

Proof. The vector $d_n = -B^{-1}g$ minimizes $Q(d)$ on $\mathcal{R}^n$. Furthermore, if $d = Z_k\tilde{d}$, then

$$Q(d) = Q(Z_k\tilde{d}) = \frac{1}{2}\tilde{d}^T Z_k^T B Z_k \tilde{d} + g^T Z_k \tilde{d}.$$

Thus a minimizer of $Q(d)$ on $\mathcal{K}_k$ has the form

$$d_k = Z_k \tilde{d}_k = -Z_k (Z_k^T B Z_k)^{-1} Z_k^T g \tag{15}$$

and since $Z_k^T Z_k = I$, Lemma 1 implies that

$$\|d_k\|^2 \le \|d_n\|^2 \quad \Rightarrow \quad g^T Z_k (Z_k^T B Z_k)^{-2} Z_k^T g \le g^T B^{-2} g.$$

□

Lemma 2 *Let $\lambda \in \mathcal{R}$ and*

$$\mathcal{K}_k(\lambda) = \mathrm{span}\{g, (B+\lambda I)g, \ldots, (B+\lambda I)^{k-1}g\}, \quad k \in \{1,\ldots,n\},$$

be the k-dimensional Krylov subspace generated by the matrix $B+\lambda I$ and the vector g. Then

$$\mathcal{K}_k(\lambda) = \mathcal{K}_k(0). \tag{16}$$

Proof. Equality (16) immediately follows for $k = 1$ because $\mathcal{K}_1(\lambda) = \mathrm{span}\{g\} = \mathcal{K}_1(0)$. Suppose now that (16) holds for some k. Then

$$(B+\lambda I)^k g = (B+\lambda I)(B+\lambda I)^{k-1} g = (B+\lambda I)v = Bv + \lambda v,$$

where $v \in \mathcal{K}_k(\lambda) = \mathcal{K}_k(0)$. As $\lambda v \in \mathcal{K}_k(0)$ and $Bv \in \mathcal{K}_{k+1}(0)$, we can write $(B+\lambda I)^k g \in \mathcal{K}_{k+1}(0)$. Thus $\mathcal{K}_{k+1}(\lambda) \subset \mathcal{K}_{k+1}(0)$. Similarly

$$B^k g = BB^{k-1}g = [(B+\lambda I) - \lambda I]u = (B+\lambda I)u - \lambda u,$$

where $u \in \mathcal{K}_k(0) = \mathcal{K}_k(\lambda)$. As $\lambda u \in \mathcal{K}_k(\lambda)$ and $(B+\lambda I)u \in \mathcal{K}_{k+1}(\lambda)$, we can write $B^k g \in \mathcal{K}_{k+1}(\lambda)$. Thus $\mathcal{K}_{k+1}(0) \subset \mathcal{K}_{k+1}(\lambda)$. □

Lemma 3 *Let B_1 and B_2 be symmetric and positive definite matrices. Then*

$$\begin{array}{lll} B_1 - B_2 \succeq 0 & \textit{if and only if} & B_2^{-1} - B_1^{-1} \succeq 0, \ \textit{and} \\ B_1 - B_2 \succ 0 & \textit{if and only if} & B_2^{-1} - B_1^{-1} \succ 0. \end{array}$$

Proof. The result follows from the relations

$$B_1 - B_2 = B_2^{\frac{1}{2}}(B_2^{-\frac{1}{2}} B_1 B_2^{-\frac{1}{2}} - I)B_2^{\frac{1}{2}}, \quad B_2^{-1} - B_1^{-1} = B_1^{-\frac{1}{2}}(B_1^{\frac{1}{2}} B_2^{-1} B_1^{\frac{1}{2}} - I)B_1^{-\frac{1}{2}}$$

and from the fact that the matrices $B_2^{-\frac{1}{2}} B_1 B_2^{-\frac{1}{2}}$ and $B_1^{\frac{1}{2}} B_2^{-1} B_1^{\frac{1}{2}}$ have the same eigenvalues because

$$B_2^{-\frac{1}{2}} B_1^{\frac{1}{2}} B_1^{\frac{1}{2}} B_2^{-\frac{1}{2}} x = \lambda x \quad \Leftrightarrow \quad B_1^{\frac{1}{2}} B_2^{-\frac{1}{2}} B_2^{-\frac{1}{2}} B_1^{\frac{1}{2}} y = \lambda y,$$

where $y = B_1^{\frac{1}{2}} B_2^{-\frac{1}{2}} x$. □

Lemma 4 *Let $Z_k^T B Z_k + \lambda_i I$, $\lambda_i \in \mathcal{R}$, $i \in \{1,2\}$, be symmetric and positive definite. Let*

$$d_k(\lambda_i) = \arg\min_{d\in\mathcal{K}_k} Q_{\lambda_i}(d), \quad \textit{where} \quad Q_\lambda(d) = \frac{1}{2}\, d^T(B+\lambda I)d + g^T d.$$

Then

$$\lambda_2 \le \lambda_1 \quad \Leftrightarrow \quad \|d_k(\lambda_2)\| \ge \|d_k(\lambda_1)\|.$$

Proof. It follows from (15) that

$$\|d_k(\lambda_i)\|^2 = g^T Z_k (Z_k^T (B+\lambda_i I) Z_k)^{-2} Z_k^T g = g^T Z_k (Z_k^T B Z_k + \lambda_i I)^{-2} Z_k^T g$$

with $Z_k^T B Z_k + \lambda_i I$ positive definite. Thus

$$\|d_k(\lambda_2)\|^2 - \|d_k(\lambda_1)\|^2 = g^T Z_k \left[(Z_k^T B Z_k + \lambda_2 I)^{-2} - (Z_k^T B Z_k + \lambda_1 I)^{-2} \right] Z_k^T g.$$

Letting $\tilde{B}_2 = Z_k^T B Z_k + \lambda_2 I$ and assuming that $\lambda_2 \le \lambda_1$ we can write

$$\begin{aligned} (Z_k^T B Z_k + \lambda_1 I)^2 - (Z_k^T B Z_k + \lambda_2 I)^2 &= (\tilde{B}_2 + (\lambda_1 - \lambda_2) I)^2 - \tilde{B}_2^2 \\ &= 2(\lambda_1 - \lambda_2)\tilde{B}_2 + (\lambda_1 - \lambda_2)^2 I \succeq 0. \end{aligned}$$

Therefore

$$(Z_k^T B Z_k + \lambda_2 I)^{-2} - (Z_k^T B Z_k + \lambda_1 I)^{-2} \succeq 0$$

by Lemma 3, which gives $\|d_k(\lambda_2)\|^2 - \|d_k(\lambda_1)\|^2 \ge 0$. Using the same procedure and the second assertion of Lemma 3 (with λ_1 and λ_2 changed) one can prove that $\lambda_1 < \lambda_2 \Rightarrow \|d_k(\lambda_1)\|^2 > \|d_k(\lambda_2)\|^2$ or $\|d_k(\lambda_2)\| \ge \|d_k(\lambda_1)\| \Rightarrow \lambda_2 \le \lambda_1$. □

Now we are in a position to prove the main theorem.

Theorem 3 *Let d_j, $j \in \{1,\dots,n\}$, be solutions of the minimization problems*

$$d_j = \arg\min_{d\in\mathcal{K}_j} Q(d) \quad \textit{subject to} \quad \|d\| \le \Delta, \quad \textit{where} \quad Q(d) = \frac{1}{2}\, d^T B d + g^T d,$$

with corresponding Lagrange multipliers λ_j, $j \in \{1,\dots,n\}$. If $1 \le k \le l \le n$, then

$$\lambda_k \le \lambda_l.$$

Proof. The vector d_j is a minimizer of the j-th trust-region subproblem if and only if $\|d_j\| = \|Z_j \tilde{d}_j\| \le \Delta$, where

$$Z_j^T(B+\lambda_j I) Z_j \tilde{d}_j = -Z_j^T g, \quad Z_j^T(B+\lambda_j I)Z_j \succeq 0, \quad \lambda_j \ge 0, \quad \lambda_j(\Delta - \|d_j\|) = 0,$$

see (11). This minimizer is unconstrained (i.e. the same result is obtained without assuming any trust-region constraint) if and only if $\lambda_j = 0$. If $\lambda_l = 0$, which means that d_l is the unconstrained minimizer, Lemma 1 implies that $\|d_k\| \le \|d_l\| \le \Delta$ for the unconstrained minimizer d_k, so $\lambda_k = 0$. If $\lambda_l > 0$ and $\lambda_k = 0$, there is nothing to prove. Let's now suppose that $\lambda_l > 0$ and $\lambda_k > 0$, which means that $\|d_l\| = \|d_k\| = \Delta$. First, assume that $Z_k^T(B+\lambda_k I)Z_k$ is singular and $\lambda_l < \lambda_k$. Then there exists $v \in \mathcal{K}_k$ such that $v^T(B+\lambda_l I)v < 0$ and, since $\mathcal{K}_k \subset \mathcal{K}_l$, $Z_l^T(B+\lambda_l I)Z_l \succeq 0$ cannot hold. This contradiction proves that $\lambda_l \ge \lambda_k$. Assume now that $Z_k^T(B+\lambda_k I)Z_k \succ 0$ and $Z_l^T(B+\lambda_l I)Z_l \succ 0$. Since

$\mathcal{K}_k(\lambda_k) = \mathcal{K}_k$ by Lemma 2, the vector d_k is a solution of the unconstrained minimization problem

$$d_k = \arg\min_{d \in \mathcal{K}_k} Q_{\lambda_k}(d), \quad \text{where} \quad Q_\lambda(d) = \frac{1}{2} d^T(B + \lambda I)d + g^T d.$$

Assume that $\lambda_k > \lambda_l$, which implies that $Z_l^T(B + \lambda_k I)Z_l \succ 0$. Let

$$d_l(\lambda_k) = \arg\min_{d \in \mathcal{K}_l} Q_{\lambda_k}(d).$$

Then $\|d_l(\lambda_k)\| \geq \|d_k\| = \Delta$ follows from Lemma 1. Since

$$d_l = \arg\min_{d \in \mathcal{K}_l} Q_{\lambda_l}(d)$$

and $\|d_l\| = \Delta \leq \|d_l(\lambda_k)\|$, Lemma 4 implies that $\lambda_k \leq \lambda_l$ which is a contradiction. Thus $\lambda_k \leq \lambda_l$ has to hold. Finally, assume that $Z_l^T(B + \lambda_l I)Z_l$ is singular. In this case, we have $\|d_l(\lambda_l + \varepsilon)\| \leq \Delta$ for arbitrary $\varepsilon > 0$. Since $Z_l^T(B + (\lambda_l + \varepsilon)I)Z_l$ is positive definite, also $Z_k^T(B + (\lambda_l + \varepsilon)I)Z_k$ is positive definite and $\|d_k(\lambda_l + \varepsilon)\| \leq \|d_l(\lambda_l + \varepsilon)\| \leq \Delta$ by Lemma 1. Since $\|d_k\| = \Delta$, Lemma 4 implies that $\lambda_k \leq \lambda_l + \varepsilon$ and, since ε is arbitrary, $\lambda_k \leq \lambda_l$. □

Now we return to subproblem (14). If we set $\tilde{\lambda} = \lambda_k$ for some $k \leq n$, then Theorem 3 implies that $0 \leq \tilde{\lambda} = \lambda_k \leq \lambda_n = \lambda$. As a consequence of this inequality, one has that $\lambda = 0$ implies $\tilde{\lambda} = 0$ so that $\|d\| < \Delta$ implies $\tilde{\lambda} = 0$. Thus the shifted Steihaug-Toint method reduces to the standard one in this case. At the same time, if B is positive definite and $0 < \tilde{\lambda} \leq \lambda$, then one has $\Delta = \|(B + \lambda I)^{-1} g\| \leq \|(B + \tilde{\lambda} I)^{-1} g\| < \|B^{-1} g\|$ by Lemma 4. Thus the unconstrained minimizer of (14) is closer to the trust-region boundary than the unconstrained minimizer of (10) and we can expect that $d(\tilde{\lambda})$ is closer to the optimal locally constrained step than d. Finally, if B is positive definite and $\tilde{\lambda} > 0$, then the matrix $B + \tilde{\lambda} I$ is better conditioned than B and we can expect that the shifted Steihaug-Toint method will converge more rapidly than the original one. The shifted Steihaug-Toint method consists of the three major steps.

Algorithm 2.1 ***The preconditioned shifted Steihaug-Toint method.***

Step 1: *Carry out $k \ll n$ steps of the unpreconditioned Lanczos method (described, e.g., in [5]) to obtain the tridiagonal matrix $T = T_k = Z_k^T B Z_k$.*

Step 2: *Solve the subproblem*

$$\text{minimize} \quad (1/2)\tilde{d}^T T \tilde{d} + \|g\| e_1^T \tilde{d} \quad \text{subject to} \quad \|\tilde{d}\| \leq \Delta, \tag{17}$$

using the method of Moré and Sorensen [10], to obtain the Lagrange multiplier $\tilde{\lambda}$.

Step 3: *Apply the (preconditioned) Steihaug-Toint method to subproblem (14) to obtain the direction vector $d = d(\tilde{\lambda})$.*

3 Global convergence

Now we show that the trust region method (2)–(8) with direction vectors d_i determined by the shifted Steihaug-Toint method is globally convergent. Since conditions (2) and (3) are satisfied automatically, it suffices to prove inequality (4) and Theorem 1 can be used (see Corollary 5).

Theorem 4 *Let $d \in \mathcal{R}^n$ be a direction vector obtained by the shifted Steihaug-Toint method with a preconditioner C. Then (4) holds with*

$$\underline{\sigma} = 1/(8\kappa(C)),$$

where $\kappa(C)$ is the spectral condition number of the preconditioner C.

Proof. (a) First, consider the CG method with the preconditioner C (symmetric and positive definite) applied to subproblem (14). This method is equivalent to the (unpreconditioned) CG method applied to a quadratic function $\hat{Q}(\hat{d}) = (1/2)\hat{d}^T\hat{B}\hat{d} + \hat{g}^T\hat{d}$, where $\hat{d} = C^{1/2}d$, $\hat{g} = C^{-1/2}g$ and $\hat{B} = C^{-1/2}(B + \tilde{\lambda}I)C^{-1/2}$. If at least one CG step is performed, then

$$-\tilde{Q}(d) = -\hat{Q}(\hat{d}) \geq \frac{\|\hat{g}\|^2}{2\|\hat{B}\|} = \frac{g^T C^{-1} g}{2\|C^{-1/2}(B + \tilde{\lambda}I)C^{-1/2}\|} \geq \frac{\|g\|^2}{2\kappa(C)\|B + \tilde{\lambda}I\|}$$

(the first inequality is proved in [18]). If the first CG step lies outside the trust-region, then

$$d_1 = C^{-1/2}\hat{d}_1 = -\frac{\hat{g}^T\hat{g}}{\hat{g}^T\hat{B}\hat{g}} C^{-1/2}\hat{g} = -\frac{g^T C^{-1} g}{g^T C^{-1}(B + \tilde{\lambda}I)C^{-1}g} C^{-1}g$$

implies that

$$\frac{g^T C^{-1} g\sqrt{g^T C^{-2} g}}{g^T C^{-1}(B + \tilde{\lambda}I)C^{-1}g} \geq \Delta \quad \Rightarrow \quad \frac{g^T C^{-1}(B + \tilde{\lambda}I)C^{-1}g}{\sqrt{g^T C^{-2} g}}\Delta \leq g^T C^{-1} g.$$

In this case, $d = (\Delta/\|d_1\|)d_1 = -(\Delta/\sqrt{g^T C^{-2} g})C^{-1}g$ and we can write

$$\begin{aligned} -\tilde{Q}(d) &= \frac{g^T C^{-1} g}{\sqrt{g^T C^{-2} g}}\Delta - \frac{1}{2}\frac{g^T C^{-1}(B + \tilde{\lambda}I)C^{-1}g}{g^T C^{-2} g}\Delta^2 \\ &\geq \frac{1}{2}\frac{g^T C^{-1} g}{\sqrt{g^T C^{-2} g}}\Delta \geq \frac{\|g\|}{2\kappa(C)}\Delta. \end{aligned}$$

Using both inequalities above we obtain

$$-\tilde{Q}(d) \geq \frac{\|g\|}{2\kappa(C)} \min\left(\Delta, \frac{\|g\|}{\|B + \tilde{\lambda}I\|}\right).$$

(b) Since $Z_k^T Z_k = I$ implies

$$\max_{\|\tilde{v}\|=1} \tilde{v}^T T \tilde{v} = \max_{\|\tilde{v}\|=1} \tilde{v}^T Z_k^T B Z_k \tilde{v} \leq \max_{\|v\|=1} v^T B v$$

($\tilde{v} \in \mathcal{R}^k$ and $v \in \mathcal{R}^n$), we can write $\|T\| \leq \|B\|$. If $\tilde{\lambda} > 0$, then $\|\tilde{d}(\tilde{\lambda})\| = \Delta$, where $(T + \tilde{\lambda}I)\tilde{d}(\tilde{\lambda}) = -\|g\|e_1$ with $\|e_1\| = 1$ (see (17) and (11)). Thus

$$\|g\|^2 = \tilde{d}(\tilde{\lambda})^T(T + \tilde{\lambda}I)^2\tilde{d}(\tilde{\lambda}) \geq \Delta^2 \min_{\|\tilde{d}\|=1} \tilde{d}^T(T + \tilde{\lambda}I)^2\tilde{d} = \Delta^2(\lambda_1 + \tilde{\lambda})^2,$$

where λ_1 is the smallest eigenvalue of T. Since $\lambda_1 \geq -\|T\|$, we can substitute it into the previous inequality to obtain

$$\tilde{\lambda} \leq \frac{1}{\Delta}\|g\| + \|T\| \leq \frac{1}{\Delta}\|g\| + \|B\|.$$

Thus

$$\frac{\|B+\tilde{\lambda}I\|}{\|g\|} \le \frac{2\|B\|}{\|g\|} + \frac{1}{\Delta} \le 2\max\left(\frac{2\|B\|}{\|g\|}, \frac{1}{\Delta}\right) \quad \Rightarrow$$

$$\frac{\|g\|}{\|B+\tilde{\lambda}I\|} \ge \frac{1}{2}\min\left(\frac{\|g\|}{2\|B\|}, \Delta\right).$$

Using (a) and the inequality $\tilde{Q}(d) = Q(d) + \tilde{\lambda}\Delta^2/2 \ge Q(d)$, we can write

$$\begin{aligned} -Q(d) &\ge -\tilde{Q}(d) \ge \frac{1}{2\kappa(C)}\|g\|\min\left(\Delta, \frac{\|g\|}{\|B+\tilde{\lambda}I\|}\right) \\ &\ge \frac{1}{2\kappa(C)}\|g\|\min\left(\Delta, \frac{1}{2}\min\left(\frac{\|g\|}{2\|B\|}, \Delta\right)\right) \ge \frac{1}{8\kappa(C)}\|g\|\min\left(\Delta, \frac{\|g\|}{\|B\|}\right) \end{aligned}$$

and (4) holds with $\underline{\sigma} = 1/(8\kappa(C))$. □

Corollary 5 *If there exist constants $\overline{B}$ and $\overline{C}$ such that the matrices B_i and the preconditioners C_i satisfy the conditions $\|B_i\| \le \overline{B}$, $\kappa(C_i) \le \overline{C}$ $\forall i \in \mathcal{N}$, then the trust region method (2)–(8) with the direction vectors d_i determined by the shifted Steihaug-Toint method is globally convergent in the sense of Theorem 1.*

4 Computational experiments

Now we present a numerical comparison of nine methods for computing direction vectors satisfying conditions (2)-(4):

- MS - the method of Moré and Sorensen [10] for computing the optimal locally constrained step.
- DL - the dogleg strategy of Powell [12] or Dennis and Mei [3].
- MDL - the multiple dogleg strategy mentioned in [18].
- ST - the basic (unpreconditioned) Steihaug [18] and Toint [19] method.
- SST - the basic (unpreconditioned) shifted Steihaug-Toint method described in this paper.
- GLRT - the method of Gould, Lucidi, Roma and Toint [5] which combines CG method with the Lanczos process to give a good approximation of the optimal locally constrained step.
- PH - the preconditioned Hager method mentioned in [6]. The incomplete Choleski preconditioner is used.
- PST - the preconditioned Steihaug-Toint method. The incomplete Choleski preconditioner is used.
- PSST - the preconditioned shifted Steihaug-Toint method. The incomplete Choleski preconditioner is used.

These methods are implemented in the interactive system for universal functional optimization UFO [9] as subroutines for solving trust-region subproblems. They have been used in trust-region versions of the discrete Newton method. These realizations use the same modules for numerical differentiation, a stepsize selection and a trust-region update. Thus the results are quite comparable. The methods listed above are implemented in the original way in almost all cases (PH is an exception). Methods based on conjugate gradient iterations are terminated whenever $\omega_i(d_i) \le \min(0.9, \sqrt{\|g_i\|}, 1/i)$, see (4). The number of extra CG or Lanczos steps in MDL, SST and PSST methods is equal to 5 and the number of Lanczos vectors in the GLRT method is bounded from above by 100. We devoted a considerable effort to the implementation of the PH method. Our first attempt based on the SSOR-preconditioned MINRES method for a projected system, used in [6], was unsuccessful. Therefore, we have chosen indefinitely preconditioned conjugate gradient method for the full saddle-point system, described in [8], with the incomplete Choleski-type decomposition of the matrix $B + \lambda I$. The tolerance 10^{-4} (see Table 5.1 in [6]) and the maximum dimension 10 of the subspace is used in our implementation of the PH method.

The above methods were tested by using two collections of 22 sparse test problems with 1000 and 5000 variables (subroutines `TEST14` and `TEST15` described in [7], which can be downloaded from `www.cs.cas.cz/~luksan/test.html`). The results are given in Tables 4.1 and 4.2, where `NIT` is the total number of iterations, `NFV` is the total number of function evaluations, `NFG` is the total number of gradient evaluations, `NDC` is the total number of Choleski-type decompositions (complete for methods MS, DL, MDL and incomplete for methods PH, PST, PSST), `NMV` is the total number of matrix-vector multiplications and `Time` is the total computational time in seconds (Table 4.2 concerns only 21 test problems, since Problem 3.11 from [7] has not been solved by any realization of the Newton method). Note that `NFG` is much greater than `NFV` in Table 4.1, since the Hessian matrices are computed by using gradient differences. At the same time, the problems referred in Table 4.2 are the sums of squares having the form $F = (1/2)f^T(x)f(x)$ and `NFV` denotes the total number of vector $f(x)$ evaluations. Since $f(x)$ is used in the expression $g(x) = J^T(x)f(x)$, where $J(x)$ is the Jacobian matrix of $f(x)$, `NFG` is comparable with `NFV` in this case.

Results in Tables 4.1 and 4.2 require several comments. All problems are sparse with a simple sparsity pattern. For this reason, the methods based on complete Choleski-type decompositions (CD) are very efficient, much better than unpreconditioned methods based on matrix-vector multiplications (MV). Since `TEST14` contains reasonably conditioned problems, the preconditioned MV methods are competitive with the CD methods. On the contrary, `TEST15` contains several very ill-conditioned problems (one of them had to be removed) and thus the CD methods work better than the MV ones. Note that the CD methods have also serious limitations, which are mentioned below.

For a better comparison of methods PST, PSST, GLRT, PH and MS, we have performed additional tests with the problems from the widely used CUTE collection [2]. We have selected only large-scale and sufficiently sparse problems. Tables 4.3a and 4.3b contain a list of these problems together with their dimensions and the results obtained. The values `NIT`, `NFV`, `NFG` and `Time` have the same meaning as in the previous tables.

Table 1: Comparison of methods using TEST14.

N	Method	NIT	NFV	NFG	NDC	NMV	Time
1000	MS	1911	1952	8724	3331	1952	3.13
	DL	2272	2409	10653	2195	2347	2.94
	MDL	2132	2232	9998	1721	21670	3.17
	ST	3475	4021	17242	0	63016	5.44
	SST	3149	3430	15607	0	75044	5.97
	GLRT	3283	3688	16250	0	64166	5.40
	PH	1958	2002	8975	3930	57887	5.86
	PST	2608	2806	12802	2609	5608	3.30
	PSST	2007	2077	9239	2055	14440	2.97
5000	MS	8177	8273	34781	13861	8272	49.02
	DL	9666	10146	42283	9398	9936	43.37
	MDL	8913	9244	38846	7587	91784	48.05
	ST	16933	19138	84434	0	376576	134.52
	SST	14470	15875	70444	0	444142	146.34
	GLRT	14917	16664	72972	0	377588	132.00
	PH	8657	8869	37372	19652	277547	127.25
	PST	11056	11786	53057	11057	23574	65.82
	PSST	8320	8454	35629	8432	59100	45.57

Table 2: Comparison of methods using TEST15.

N	Method	NIT	NFV	NFG	NDC	NMV	Time
1000	MS	1946	9094	9038	3669	2023	5.86
	DL	2420	12291	12106	2274	2573	9.00
	MDL	2204	10586	10420	1844	23139	7.86
	ST	2738	13374	13030	0	53717	11.11
	SST	2676	13024	12755	0	69501	11.39
	GLRT	2645	12831	12547	0	61232	11.30
	PH	1987	9491	9444	6861	84563	11.11
	PST	3277	16484	16118	3278	31234	11.69
	PSST	2269	10791	10613	2446	37528	8.41
5000	MS	7915	33607	33495	14099	8047	89.69
	DL	9607	42498	41958	9299	9963	128.92
	MDL	8660	37668	37308	7689	91054	111.89
	ST	11827	54699	53400	0	307328	232.70
	SST	11228	51497	50333	0	366599	231.94
	GLRT	10897	49463	48508	0	300580	214.74
	PH	8455	36434	36236	20538	281736	182.45
	PST	9360	41524	41130	9361	179166	144.40
	PSST	8634	37163	36881	8915	219801	140.44

Method		PST				PSST				GLRT				PH				MS	
Problem	**N**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**
ARWHEAD	5000	6	28	18	0.44	6	7	21	0.38	6	17	21	0.36	6	28	18	0.47	6	28
BDQRTIC	5000	15	39	135	1.59	14	15	135	1.55	18	19	171	1.69	15	39	135	2.02	14	15
BROWNAL	500	8	9	4509	97.49	4	5	2505	86.49	3	4	2004	83.53	5	6	3006	90.77	5	6
BROYDN7D	2000	51	57	468	1.16	50	80	450	1.17	54	84	486	1.22	63	77	576	1.98	36	42
BRYBND	5000	11	13	168	1.22	11	13	168	1.25	20	23	294	1.81	14	17	210	1.75	15	18
CHAINWOO	1000	78	91	395	0.31	80	92	405	0.34	783	999	3920	3.52	497	605	2490	4.56	44	53
COSINE	5000	5	7	24	0.20	6	8	28	0.22	12	13	52	0.31	14	18	60	0.41	15	16
CRAGGLVY	5000	18	19	76	0.59	18	19	76	0.61	17	18	72	0.58	17	20	72	0.86	17	18
CURLY10	1000	21	44	462	0.27	19	21	440	0.30	29	52	638	2.00	18	41	396	1.84	21	23
CURLY20	1000	18	21	798	0.69	18	20	798	0.69	21	43	882	3.33	19	43	798	3.06	15	17
CURLY30	1000	19	22	1240	1.38	16	17	1054	1.22	24	48	1488	4.84	19	21	1240	3.74	18	20
DIXMAANA	3000	6	7	56	0.18	5	6	48	0.20	8	9	72	0.25	5	6	48	0.20	5	6
DIXMAANB	3000	6	7	56	0.19	7	8	64	0.24	8	9	72	0.22	9	11	80	0.34	11	14
DIXMAANC	3000	7	8	64	0.20	7	8	64	0.22	9	10	80	0.25	13	16	112	0.41	9	11
DIXMAAND	3000	8	9	72	0.22	8	9	72	0.25	11	12	96	0.27	11	14	96	0.49	11	13
DIXMAANE	3000	8	9	72	0.20	7	8	64	0.24	11	12	96	0.31	9	11	80	0.39	10	12
DIXMAANF	3000	14	16	120	0.31	15	17	128	0.36	15	18	128	0.37	27	34	224	0.92	24	30
DIXMAANG	3000	14	15	120	0.26	15	17	128	0.27	15	18	128	0.42	24	31	200	0.88	20	25
DIXMAANH	3000	16	19	136	0.34	16	19	136	0.37	15	18	128	0.38	24	30	200	0.99	24	30
DIXMAANI	3000	10	12	88	0.25	9	10	80	0.27	11	12	96	1.47	11	13	96	0.58	10	12
DIXMAANJ	3000	19	24	160	0.74	21	26	176	0.58	19	25	160	0.78	36	45	296	1.64	34	42
DIXMAANK	3000	21	26	176	0.61	23	28	192	0.50	24	29	200	0.94	35	43	288	1.67	22	28
DIXMAANL	3000	25	30	208	0.56	23	28	192	0.74	28	35	232	1.67	29	37	240	1.32	25	31
DQRTIC	5000	33	34	68	0.17	34	35	70	0.24	33	34	68	0.19	34	35	70	2.16	34	35
EDENSCH	5000	12	13	52	0.41	11	12	48	0.41	16	19	68	0.47	12	13	52	0.45	12	13
EG2	1000	3	4	12	0.02	6	7	21	0.05	3	4	12	0.03	13	15	42	0.09	13	36
EIGENALS	506	58	71	29913	96.59	58	70	29913	98.75	53	64	27378	84.28	108	135	55263	527.67	147	185
ENGVAL1	5000	8	9	36	0.20	8	9	36	0.30	11	12	48	0.33	9	37	36	0.47	8	9
EXTROSNB	1000	4745	5002	18984	11.00	4750	5027	19004	12.56	4606	4684	18428	14.72	4593	4654	18376	26.31	4597	4656
FLETCBV2	1000	7	8	32	0.05	7	8	32	0.06	9	10	40	0.15	7	8	32	0.06	7	8

Method		PST				PSST				GLRT				PH				MS	
Problem	**N**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**	**NFG**	**Time**	**NIT**	**NFV**
FLETCHCR	1000	1436	1440	5748	3.21	1433	1435	5736	3.74	1409	1432	5640	3.89	1418	1421	5676	5.54	1396	1402
FMINSRF2	1024	21	28	220	0.17	54	60	550	0.45	151	162	1520	0.95	51	58	520	0.86	45	52
FMINSURF	484	86	104	42195	38.86	46	49	22795	23.14	123	129	60140	23.34	37	41	18430	29.05	48	50
FREUROTH	5000	8	32	32	0.36	8	12	36	0.36	13	40	52	0.45	7	10	32	0.39	7	10
GENHUPPS	1000	8121	8637	32488	36.05	7466	7775	29868	35.94	6977	7225	27912	30.18	23246	23766	92988	153.55	23430	24018
GENROSE	1000	755	892	3024	1.95	722	864	2892	2.18	695	780	2784	1.83	688	845	2756	3.80	709	875
LIARWHD	5000	15	17	48	0.39	15	17	48	0.39	16	17	51	0.39	12	13	39	0.50	13	14
MOREBV	5000	17	18	108	0.47	17	18	108	0.55	5	6	36	0.83	2	3	18	0.33	4	5
MSQRTALS	529	33	41	18020	51.44	30	36	16430	48.30	33	40	18020	52.17	34	48	18550	261.24	53	69
NCB20	1010	52	62	2332	7.37	56	63	2508	8.17	54	65	2420	7.53	57	65	2552	9.75	41	49
NCB20B	1010	24	29	1000	1.67	24	29	1000	1.62	28	33	1160	1.94	15	17	640	1.53	17	21
NONCVXU2	1000	376	396	4524	2.91	263	304	3168	2.38	324	335	3900	2.45	742	782	8916	18.58	298	344
NONCVXUN	1000	1701	1719	20424	121.51	558	589	6708	27.94	1098	1119	13188	78.03		NFG > 80000				NFG > 8000
NONDIA	5000	6	7	21	0.28	7	8	24	0.27	6	7	21	0.27	5	6	18	0.31	5	6
NONDQUAR	5000	26	27	135	0.67	26	30	135	0.81	141	171	710	2.64	22	25	115	1.56	20	22
PENALTY1	500	41	44	21042	23.40	41	44	21042	21.83	40	44	20541	12.86	40	44	20541	22.81	40	44
POWELLSG	5000	17	18	72	0.25	17	18	72	0.28	18	19	76	0.25	17	18	72	0.38	17	18
POWER	500	28	29	14529	18.16	28	29	14529	16.82	29	30	15030	11.45	28	29	14529	46.04	28	29
QUARTC	5000	231	232	464	0.93	224	225	450	1.27	231	232	463	0.95	231	232	464	2.59	231	232
SBRYBND	5000			NFG > 40000				NFG > 40000				NFG > 40000				NFG > 80000		27	29
SCHMVETT	5000	3	4	24	0.33	3	4	24	0.34	9	10	60	0.59	3	4	24	0.36	3	4
SINQUAD	5000	223	242	896	5.34	225	236	904	5.91	226	250	908	5.31	93	112	376	3.47	224	250
SPARSINE	1000	13	15	924	1.34	14	16	990	1.53	19	21	1320	2.09	267	269	17688	37.42	24	33
SPARSQUR	1000	19	20	1320	0.91	19	20	1320	0.95	19	20	1320	0.89	19	20	1320	1.03	19	20
SPMSRTLS	5000	14	20	135	0.92	16	21	153	1.06	17	22	162	1.11	18	23	171	1.75	17	22
SROSENBR	5000	9	11	30	0.16	6	7	21	0.14	8	9	27	0.14	6	7	21	0.20	6	7
TOINTGSS	5000	2	3	18	0.20	9	13	60	0.42	4	5	30	0.42	2	3	18	0.24	1	2
TQUARTIC	5000	20	21	63	0.41	17	18	54	0.42	20	21	63	0.41	13	14	42	0.78	16	17
VAREIGVL	500	14	15	7515	18.06	13	14	7014	16.89	14	15	7515	15.34	14	15	7515	28.11	14	15
WOODS	4000	42	49	172	0.42	41	48	168	0.45	41	48	168	0.49	40	46	164	0.66	40	46

5 Conclusion

We have to stress that the considerations and the results in the previous section concern trust-region versions of the Newton method. In this case, the Hessian matrices are frequently indefinite and the trust region versions are very suitable. The variable metric methods with positive definite approximations of the Hessian matrices can be efficiently implemented in the line-search framework. Our conclusions concern large-scale problems where the sparsity pattern plays a considerable role. First, we would like to point out that it is advantageous to have several different procedures for computing trust-region steps. The CD methods are very efficient for ill-conditioned but reasonably sparse problems, e.g., CHAINWOO and SBRYBND. If the problems do not have sufficiently sparse Hessian matrices, then the CD methods can be much worse than the MV methods as is demonstrated on problems EIGENALS, MSQRTALS, NONCVXU2, NONCVXUN and SPARSINE. An efficiency of the MV methods strongly depends on a suitable preconditioning as is demonstrated in Tables 4.1 and 4.2. There are two possibilities. The first one mentioned in [5] changes the trust-region problem whereas the second one mentioned in Section 1 deforms the trust region path in the original trust-region problem. Note that the GLRT method cannot be preconditioned in the second way, since the preconditioned Lanczos process does not generate an orthonormal basis related to the original trust-region problem. Our preliminary tests have shown that the first preconditioning technique is less efficient because it failed in many cases. Comparing ST and SST methods (Tables 4.1 and 4.2), we can see that SST does not improve efficiency of ST even if it decreases the numbers of iterations and function evaluations. Similarly, we can conclude that PSST is usually slightly worse than PST, measured by the computational time, since it uses additional operations for determining the Lanczos matrix T and computing the parameter $\tilde{\lambda}$. Nevertheless, if the problems are difficult as BROWNAL, CHAINWOO, FMINSURF, MSQRTALS and NONCVXUN, then PSST is much better than PST. Thus the total computational time can be lower for PSST as in Tables 4.1 and 4.2.

To sum up, our computational experiments indicate that the shifted Steihaug-Toint method proposed in this paper works well in the connection with the second way of preconditioning. The trust region step reached in this case is usually close to the optimum step obtained by the MS method. Furthermore, these experiments show that the PH method sometimes uses more matrix-vector multiplications than the GLRT method, which differs from the observation contained in [6]. This is caused by the fact that the results in [6] concern accurate solutions of isolated trust-region problems while our tests are related to unconstrained optimization where the methods based on conjugate gradient iterations use limited accuracy $\omega_i(d_i) \le \min(0.9, \sqrt{\|g_i\|}, 1/i)$. This upper bound can be large enough in many iterations when $\|g_i\|$ is large and i is small.

References

[1] P.T.Boggs, J.W.Tolle: *Sequential Quadratic Programming*, Acta Numerica (1995), pp. 1-51.

[2] I.Bongartz, A.R.Conn, N.Gould, P.L.Toint: *CUTE: constrained and unconstrained testing environment*, ACM Transactions on Mathematical Software 21 (1995), pp. 123-160.

[3] J.E.Dennis, H.H.W.Mei: *An unconstrained optimization algorithm which uses function and gradient values*, Report No. TR 75-246, 1975.

[4] C.Fortin, H.Wolkowicz: *The trust-region subproblem and semidefinite programming*, Optimization Methods and Software 19 (2004), pp. 41-67.

[5] N.I.M.Gould, S.Lucidi, M.Roma, P.L.Toint: *Solving the trust-region subproblem using the Lanczos method*, Report No. RAL-TR-97-028, 1997.

[6] W.W.Hager: *Minimizing a quadratic over a sphere*, SIAM J. Optim., 12(1): 188-208, 2001.

[7] L.Lukšan, J.Vlček: *Sparse and partially separable test problems for unconstrained and equality constrained optimization*, Report No. V767-98, Institute of Computer Science AS CS, 1998.

[8] L.Lukšan, J.Vlček: *Indefinitely Preconditioned Inexact Newton Method for Large Sparse Equality Constrained Nonlinear Programming Problems*, Numerical Linear Algebra with Applications 5 (1998), pp. 219-247.

[9] L.Lukšan, M.Tůma, J.Hartman, J.Vlček, N.Ramešová, M.Šiška, C.Matonoha: *Interactive System for Universal Functional Optimization (UFO). Version 2006* Report No. V977-06, Institute of Computer Science AS CS,Prague, 2006.

[10] J.J.Moré, D.C.Sorensen: *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computations 4 (1983), pp. 553-572.

[11] J.Nocedal, S.J.Wright: *Numerical Optimization*, Springer, New York, 1999.

[12] 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.

[13] M.J.D.Powell: *On the global convergence of trust region algorithms for unconstrained optimization*, Mathematical Programming 29 (1984), pp. 297-303.

[14] F.Rendl, H.Wolkowicz: *A semidefinite framework for trust region subproblems with applications to large-scale minimization*, Mathematical Programming 77 (1997), pp. 273-299.

[15] M.Rojas, S.A.Santos, D.C.Sorensen: *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim., 11(3): 611-646, 2000.

[16] D.C.Sorensen: *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7(1): 141-161, 1997.

[17] D.C.Sorensen: *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19(2): 409-426, 1982.

[18] T.Steihaug: *The conjugate gradient method and trust regions in large-scale optimization*, SIAM Journal on Numerical Analysis 20 (1983), pp. 626-637.

[19] P.L.Toint: *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I.S.Duff, ed., Academic Press, London, 1981, pp. 57-88.