SIAM J. OPTIM.
Vol. 29, No. 1, pp. 571–594

A BLOCK LANCZOS METHOD FOR THE EXTENDED TRUST-REGION SUBPROBLEM*

LIQIANG SONG† AND WEI HONG YANG†

Abstract. We present a block Lanczos method for solving the large-scale extended trust-region (ETR) subproblem. During the algorithm, the original ETR subproblem is projected to a small-scale one, and then the active-set method is employed to solve this small-scale ETR subproblem to get a solution that can be used to derive an approximate solution of the original ETR subproblem. Theoretical analysis of error bounds for the optimal value, the optimal solution, and the multipliers is also proposed. We compare the block Lanczos method with the TOMLAB solver. Numerical experiments demonstrate that the block Lanczos method is effective and can achieve high accuracy for large-scale ETR subproblems.

Key words. extended trust-region subproblem, block Lanczos method, Krylov subspace, active-set method, alternating direction method of multipliers (ADMM)

AMS subject classifications. 90C20, 90C26, 90C30, 65K05

DOI. 10.1137/17M1156617

1. Introduction.

In this paper, we consider the following extended trust-region (ETR) subproblem:

$$\min\ f(p) := \frac{1}{2}p^T Mp + q^T p \tag{1}$$

$$\text{s.t.}\quad Ap = \beta, \tag{2}$$

$$Bp \le \gamma, \tag{3}$$

$$\|p\| \le \Delta, \tag{4}$$

where $M \in \mathbb{R}^{n\times n}$ is symmetric, $A \in \mathbb{R}^{m_1\times n}$, $B \in \mathbb{R}^{m_2\times n}$, $q \in \mathbb{R}^n$, $\beta \in \mathbb{R}^{m_1}$, and $\gamma \in \mathbb{R}^{m_2}$. Such a problem arises from the trust-region methods for solving nonlinear programming problems with inequality constraints [17, 29] and robust optimization problems under matrix norm or polyhedral uncertainty [7, 9, 25]. In the special case of ETR where $m_1 = m_2 = 0$, it is the classic trust-region (TR) problem, which has been extensively studied in the literature (see [17, 40] and the references therein).

Many interesting theoretical works have been established on the ETR subproblem (see [2, 10, 12, 13, 23, 26, 30, 35] and the references therein). It is proved in [10] that the ETR subproblem is polynomial-time solvable under the condition that the number of faces of the linear constraints intersecting with the unit ball is polynomially bounded. Thus, proposing an efficient algorithm to solve the ETR subproblem is of great concern to optimization researchers.

A powerful tool for the ETR subproblem is the semidefinite programming (SDP) relaxation method (see [2, 6, 26, 31]). But, even for the simple case of the ETR subproblem with $(m_1, m_2) = (0, 1)$, its SDP relaxation may have a positive duality (see [2, 26]). It has been shown that exact SDP-relaxation of the ETR subproblem

*Received by the editors November 13, 2017; accepted for publication (in revised form) December 10, 2018; published electronically February 21, 2019.
http://www.siam.org/journals/siopt/29-1/M115661.html
Funding: The work of the second author was supported by the National Natural Science Foundation of China NSFC-91730304.
†School of Mathematical Sciences, Fudan University, Shanghai 200433, People's Republic of China (lqsong14@fudan.edu.cn, whyang@fudan.edu.cn).

and strong duality hold under some conditions (see [6, 26] and the references therein). However, these conditions are rather restrictive. For example, the dimension condition in [26] requires that the number of inequality constraints must be strictly less than the multiplicity of the minimum eigenvalue of the M.

A variant of the SDP relaxation method is the SDP relaxation with second-order-cone reformulation (SDPR-SOCR) (see, e.g., [12, 13, 23, 35, 37, 38]). In particular, it is proved that exact SDPR-SOCR holds if the linear constraints are nonintersecting in the trust-region ball [13], which is still a strong condition.

Recently, an interesting approach has been developed for solving the ETR subproblem, based on solving a single generalized eigenvalue problem (see [1, 34]). But, until now, this method has only been used to treat the case when $m_2 \leq 1$.

In this paper, we propose a subspace method for large-scale ETR subproblems. For the classic TR method, subspace algorithms have shown advantages for large-scale problems (see, e.g., [15, 19, 20, 36, 41]). Subspace techniques also play an important role in the development of numerical methods for large-scale nonlinear optimization. For more detailed discussions on subspace selection, see Yuan [39].

Our study is inspired by the work of Gould et al. [20], in which a generalized Lanczos trust-region method is presented for solving the TR problem. In the generalized Lanczos method, a Krylov subspace is added to the constraints so that a small-scale TR subproblem is formed. An approximate solution can be obtained by solving the smaller TR subproblem. For the ETR subproblem, the situation is a little different because the ETR subproblem has linear constraints. Our approach instead adds a block Krylov subspace to the constraints of (1)–(4) to form a small-scale ETR subproblem, and then uses the active-set method or the TOMLAB solver to solve the small-scale ETR subproblem. Numerical experiments demonstrate that our approach is effective, and can achieve high accuracy for large-scale ETR subproblems.

This paper is organized as follows. In section 2, we introduce some notation, definitions, and preliminary results used in this paper. In section 3, we prove that the pair of two sets—the block Krylov subspace and the feasible region—is linear regular. Error bounds for the optimal value, the optimal solution, and the multipliers are presented in section 4. The block Lanczos method for the ETR subproblem is given in detail in section 5. Some numerical experiments comparing the block Lanczos method with the TOMLAB solver are proposed in section 6. The paper ends with some conclusions and a short discussion on possible future works.

2. Notions and preliminary results. In this section, we introduce the notation, definitions, and preliminary results that will be used throughout the paper. As usual, I_n denotes the $n \times n$ identity matrix and e_i denotes the ith column of I_n. We use E^n_{m+} to denote the first m columns of I_n, and E^n_{m-} to denote the last m columns of I_n. The notation $\mathrm{span}\{v_1, v_2, \ldots, v_k\}$ denotes the linear space spanned by vectors $v_1, v_2, \ldots, v_k$. To simplify our presentation, we shall also adopt a MATLAB-like convention to access the entries of vectors and matrices. For example, $(i : j)$ stands for the set of integers from i to j inclusive, and $A(k : l, i : j)$ is the submatrix of A that consists of the intersections of rows k to l and columns i to j.

For $p \in \mathbb{R}^n$ and a positive definite symmetric matrix $H \in \mathbb{R}^{n\times n}$, $\|p\|_H$ denotes the H-norm of p, i.e., $\|p\|_H = \sqrt{p^T H p}$. If $H = I_n$, we omit the subscript and denote by $\|p\|$ the Euclidean norm of p. The 2-norm of H is denoted by $\|H\|$. Let $0 < \theta_1 \leq \cdots \leq \theta_n$ be the eigenvalues of H. Then we have

$$\sqrt{\theta_1}\|p\| \leq \|p\|_H \leq \sqrt{\theta_n}\|p\|. \tag{5}$$

Let S be a closed, convex, and nonempty set in $\mathbb{R}^n$. As usual, $\mathrm{bd}(S)$ and $\mathrm{int}(S)$ denote the boundary and interior of S, respectively. We denote by $d_H(p,S) := \inf_{p' \in S} \|p - p'\|_H$ the distance function from p to S in the H-norm. If $H = I_n$, we omit the subscript and only use $d(p,S)$ to denote $d_H(p,S)$. Let $P_S(p)$ be the projection operator from p to S in the Euclidean norm. For $z \in S$, let $N_S(z)$ denote the normal cone (see [5, Page 137] or [28, equation (2.4)]) of S at z, defined by

$$N_S(z) := \{v \in \mathbb{R}^n : v^T(y - z) \le 0 \ \forall y \in S\}. \tag{6}$$

It is proved in [5, Proposition 12] that

$$p - P_S(p) \in N_S(P_S(p)). \tag{7}$$

For the ETR subproblem (1)–(4), we write A and B as

$$A = [a_1, \ldots, a_{m_1}]^T \quad \text{and} \quad B = [b_1, \ldots, b_{m_2}]^T. \tag{8}$$

The index set $I(p)$ at any feasible p is defined as follows:

$$I(p) = \{i : 1 \le i \le m_2,\ b_i^T p = \gamma_i\}. \tag{9}$$

Let A, B, β, and γ be as defined in the extended trust-region problem (1)–(4). For simplicity of notation, we will use the following throughout the paper:

$$\begin{cases} \Omega := \{p : Ap = \beta,\ Bp \le \gamma,\ \|p\| \le \Delta\}, \\ \Theta := \{p : Ap = \beta,\ Bp \le \gamma\}, \\ \Phi := \{p : \|p\| \le \Delta\}. \end{cases} \tag{10}$$

Let

$$\Delta_{\min} := \min_{x \in \Theta} \|p\| \quad \text{and} \quad \hat{p} = \arg\min_{x \in \Theta} \|p\|. \tag{11}$$

We can see that if $\Delta < \Delta_{\min}$, then problem (1)–(4) is infeasible; if $\Delta = \Delta_{\min}$, the only feasible solution is $\hat{p}$. Thus, $\Delta > \Delta_{\min}$ is the only interesting case. In the remainder of the paper, we assume that $\Delta > \Delta_{\min}$. Equivalently, we make the following assumption.

Assumption 2.1. Assume that $\|\hat{p}\| < \Delta$, that is, $\hat{p} \in \mathrm{int}(\Phi)$.

Under this assumption, the following KKT condition of the ETR subproblem holds.

THEOREM 2.1. *Suppose $\Delta > \Delta_{\min}$. If p^* is a global solution of* (1)–(4)*, then there exist $\sigma^* \ge 0$, λ^* and $\mu^* \ge \mathbf{0}$, such that the following conditions are satisfied:*

$$(M + \sigma^* I_n)p^* + q + A^T\lambda^* + B^T\mu^* = \mathbf{0}_n, \tag{12}$$

$$p^* \in \Omega, \tag{13}$$

$$\mu^{*T}(Bp^* - \gamma) = 0, \tag{14}$$

$$\sigma^*(\|p^*\| - \Delta) = 0. \tag{15}$$

2.1. Block Lanczos method. In this subsection, we give a brief introduction to the block Lanczos method. For a detailed description of this method, refer to [33, Chapter 6]. Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix, $q \in \mathbb{R}^n$, and $\mathcal{C} \in \mathbb{R}^{n \times m_0}$. The kth

Krylov subspace $\mathcal{K}_k(M,q)$ is defined by $\mathcal{K}_k(M,q) := \text{span}\{q, Mq, \ldots, M^{k-1}q\}$, and the kth block Krylov subspace $\mathcal{K}_k(M,\mathcal{C})$ is defined by

$$\mathcal{K}_k(M,\mathcal{C}) := \text{span}\{\mathcal{C}, M\mathcal{C}, \ldots, M^{k-1}\mathcal{C}\}.$$

We can see that

$$\text{if } q \text{ belongs to the range of } \mathcal{C}, \text{ then } \mathcal{K}_k(M,q) \subset \mathcal{K}_k(M,\mathcal{C}), \tag{16}$$

and therefore the Lanczos method for $Mx = q$ and the block Lanczos method for $MX = \mathcal{C}$ are closely related.

2.1.1. Convergence bound for the Lanczos method. We consider solving $Mx = q$ by the Lanczos method. For this method, various error estimates have been obtained (see [33, section 6.11.3] or [29, section 5.1]), and we now introduce some of them. To give the error bound, we need a powerful tool named the Chebyshev polynomial. Let $\mathbb{P}_k$ denote all polynomials with degree no higher than k. The Chebyshev polynomial $\mathscr{T}_k(t) \in \mathbb{P}_k$ of the first kind of degree k is defined by

$$\mathscr{T}_k(t) := \begin{cases} \cos(k \arccos\ t) & \text{for } |t| \leq 1, \\ \frac{1}{2}\left[(t+\sqrt{t^2-1})^k + (t+\sqrt{t^2-1})^{-k}\right] & \text{for } |t| \geq 1. \end{cases}$$

If $t > 1$, then $\mathscr{T}_k(t) \geq (t+\sqrt{t^2-1})^k/2$, and therefore it grows extremely fast for $t > 1$.

Let p^* be the solution of $Mp = q$. If M is positive definite, we have (see [33, Theorem 6.29])

$$d_M(p^*, \mathcal{K}_k(M,q)) \leq \frac{\|p^*\|_M}{\mathscr{T}_k(\frac{\kappa+1}{\kappa-1})}, \tag{17}$$

where κ is the condition number of M. If M is indefinite, assume that all eigenvalues of M lie in the two intervals $[\alpha_1, \alpha_2]$ and $[\alpha_3, \alpha_4]$, where $\alpha_1 < \alpha_2 < 0 < \alpha_3 < \alpha_4$ and $\alpha_2 - \alpha_1 = \alpha_4 - \alpha_3 > 0$. Then we have (see [21])

$$d(p^*, \mathcal{K}_k(M,q)) \leq 2\left(\frac{\sqrt{|\alpha_1\alpha_4|} - \sqrt{|\alpha_2\alpha_3|}}{\sqrt{|\alpha_1\alpha_4|} + \sqrt{|\alpha_2\alpha_3|}}\right)^{k/2} \|p^*\|. \tag{18}$$

2.1.2. Description of the block Lanczos method. Consider solving the linear systems $MX = \mathcal{C}$ by the block Lanczos method. In [32], a variant of this method was developed by Ruhe for constructing an orthonormal basis of $\mathcal{K}_k(M,\mathcal{C})$. In Ruhe's implementation, the vectors in the Krylov subspace bases are generated and orthogonalized sequentially. This implementation is attractive, since the breakdown can be easily identified (see [3]). The algorithm is labeled as Algorithm 4 and is given in the appendix for simplicity.

An application of k steps of Algorithm 4 yields

$$MQ_k = Q_kT_k + W_{k+1}(E^{km_0}_{m_0-})^T,$$

with

$$Q_k^TQ_k = I_{km_0}, \qquad Q_k^TW_{k+1} = \mathbf{0}_{km_0\times m_0},$$

and

$$T_k = \begin{pmatrix} D_1 & Z_1^T & & & \\ Z_1 & D_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & D_{k-1} & Z_{k-1}^T \\ & & & Z_{k-1} & D_k \end{pmatrix}, \tag{19}$$

where $Z_j \in \mathbb{R}^{m_0 \times m_0}$ are upper triangular blocks, $D_j \in \mathbb{R}^{m_0 \times m_0}$, and

$$Q_k = [V_1, V_2, \ldots, V_k]. \tag{20}$$

2.1.3. Dealing with breakdowns. The values

$$t_{j+1\ r} = \|w_1\|, \quad j = m_0, m_0 + 1, \ldots, km_0 - 1, \quad r = 1, 2, \ldots, km_0 - m_0$$

(step 16 in Algorithm 4), correspond to the diagonal elements in the off-diagonal blocks $Z_1, \ldots, Z_{k-1}$ in (19). Therefore, an off-diagonal block is singular if and only if $t_{j+1\ r} = \|w_1\| = 0$ for some j and r. Let ϵ be the prescribed tolerance. We say a breakdown occurs when

$$\|w_1\| < \epsilon.$$

When breakdown occurs, Algorithm 4 needs to be modified to handle this case. The following algorithm is obtained by replacing steps 15–19 of Algorithm 4 by new steps (2–13 of Algorithm 1). The other steps of Algorithm 4 remain the same and are represented by the vertical ellipses.

Algorithm 1 Block Lanczos algorithm (Ruhe's variant) can deal with breakdowns.

$\vdots$

1:
2: **if** $j < km_0$ **then**
3: $\quad t_{j+1\ r} = \|w_1\|$;
4: $\quad$**if** $t_{j+1\ r} < \epsilon$ **then**
5: $\qquad$set $t_{r\ j+1} = t_{j+1\ r} = 0$;
6: $\qquad$let w_1 be a random vector;
7: $\qquad$**for** $i = 1 : j$ **do**
8: $\qquad\quad w_1 = w_1 - \langle w_1, v_i \rangle v_i$;
9: $\qquad$**end for**
10: $\quad$**end if**
11: $\quad v_{j+1} = w_1 / \|w_1\|$;
12: $\quad t_{r\ j+1} = t_{j+1\ r}$;
13: **end if**

$\vdots$

3. Linear regularity. In our method, the constraint $p \in \mathcal{K}_k(M, [q, A^T, B^T])$ is added to the original problem (1)–(4) to form the following problem:

$$\min \quad f(p) := \frac{1}{2} p^T M p + q^T p \tag{21}$$

$$\text{s.t.} \quad Ap = \beta, \ Bp \leq \gamma, \ \|p\| \leq \Delta, \tag{22}$$

$$p \in \mathcal{K}_k(M, [q, A^T, B^T]). \tag{23}$$

Now we show that constraints (22) and (23) are consistent. We need the following preliminary result, which has several applications, such as studying the structure of the normal fans of polyhedral convex sets (see [27]) or deriving the KKT conditions for linear constraint optimization problems (see [29, section 12.6]).

LEMMA 3.1 (see [29, Lemma 12.9] or [27, equation (6)]). *Write A and B as in* (8). *For $\Theta = \{p : Ap = \beta,\ Bp \leq \gamma\}$ and $q \in \Theta$, it holds that*

$$N_\Theta(q) = \left\{ \sum_{i=1}^{m_1} \lambda_i a_i + \sum_{j \in \mathcal{A}(q)} \mu_j b_j : \mu_j \geq 0 \text{ for } j \in \mathcal{A}(q) \right\},$$

where $\mathcal{A}(q) = \{j : 1 \leq j \leq m_2,\ b_j^T q = \gamma_j\}$.

Let $\hat{p}$ be as in (11). By (7) and Lemma 3.1, there exist μ_1 and $\mu_2 \geq \mathbf{0}$ such that $-\hat{p} = A^T\mu_1 + B^T\mu_2$, which implies

$$\hat{p} \in \mathcal{K}_k(M, [q, A^T, B^T]), \tag{24}$$

and therefore $\hat{p}$ is feasible for the problem (21)–(23).

Let Θ and Φ be as defined in (10). We use $\mathcal{K}_k$ to denote the Krylov subspace $\mathcal{K}_k(M, [q, A^T, B^T])$. Then (21)–(23) can be written as $\min_{p \in \mathcal{K}_k \cap \Theta \cap \Phi} f(p)$, the optimal solution of which is denoted by p_k. We need to know how well p_k approximates the original optimal solution p^*. In fact, this problem depends on the relative position of the three sets $\{\mathcal{K}_k, \Theta, \Phi\}$. To this end, to establish the linear convergence of the projection algorithm, we need the concept of linear regularity, which is introduced in [4, Definition 5.6].

DEFINITION 3.2. *Let $\{C_i\}_{i=1}^m$ be a collection of closed sets in $\mathbb{R}^n$ such that $C := \cap_{i=1}^m C_i \neq \emptyset$. We say that $\{C_i\}_{i=1}^m$ is linearly regular if there exists $\varrho > 0$ such that*

$$d(x, C) \leq \varrho \max_{1 \leq i \leq m} d(x, C_i) \quad \forall x \in \mathbb{R}^n.$$

In this case, we also say that $\{C_i\}_{i=1}^m$ is linearly regular with modulus ϱ.

The following result characterizes the concept of linear regularity in terms of normal cones, which has been generalized in [14, 42] to disclose the connection between linear regularity and the important concepts of metric regularity and error bounds for convex inequalities.

THEOREM 3.3 (see [28, Theorem 4.2]). *Suppose that $C_1, \ldots, C_k$ are closed convex sets such that $C := \cap_{i=1}^k C_i \neq \emptyset$. Then the following statements are equivalent:*

(i) *for any $x \in C$, we have $N_C(x) = \sum_{i=1}^k N_{C_i}(x)$ and*

$$\min \left\{ \sum_{i=1}^k \|z_i\|^2 : z_i \in N_{C_i}(x),\ \sum_{i=1}^k z_i = z \right\} \leq \rho \|z\|^2 \quad \forall z \in N_C(x);$$

(ii) $d^2(y, C) \leq \rho \sum_{i=1}^k d^2(y, C_i)$ $\forall y \in \mathbb{R}^n$.

Next we use the above result to prove the linear regularity of $\{\mathcal{K}_k, \Theta, \Phi\}$. First, we give a preliminary result. Define

$$\tau := \frac{\|\hat{p}\|}{\Delta}. \tag{25}$$

It is obvious that $\tau < 1$.

LEMMA 3.4. *Suppose that Assumption* 2.1 *holds. Let* $p \in \Theta \cap \mathrm{bd}(\Phi)$. *For any nonzero vector* $w \in N_\Theta(p)$, *we have*

$$\cos\angle(p, w) \geq -\tau, \tag{26}$$

where τ *is defined by* (25).

Proof. Since $w \in N_\Theta(p)$ and $\hat{p} \in \Theta$, by (6), we have $w^T(\hat{p} - p) \leq 0$, which, together with $\|p\| = \Delta$, implies that

$$\cos\angle(p, w) = \frac{w^T p}{\|p\| \cdot \|w\|} \geq \frac{w^T \hat{p}}{\Delta\|w\|} \geq \frac{-\|\hat{p}\|}{\Delta}.$$

This completes the proof. ∎

Now we give the main result of this subsection, which tells us that $\{\mathcal{K}_k, \Theta, \Phi\}$ is linearly regular and the modulus is independent on k.

THEOREM 3.5. *Suppose that Assumption* 2.1 *holds. Let* $C := \mathcal{K}_k \cap \Theta \cap \Phi$. *Then*

$$d^2(y, C) \leq \frac{1}{1-\tau}(d^2(y, \mathcal{K}_k) + d^2(y, \Theta) + d^2(y, \Phi)) \quad \forall y \in \mathbb{R}^n, \tag{27}$$

where τ *is defined by* (25).

Proof. By (24) and Assumption 2.1, we have $\hat{p} \in \mathrm{int}(\Phi) \cap \mathcal{K}_k \cap \Theta$, which, together with [5, Theorem 2], implies that

$$N_C(x) = N_{\mathcal{K}_k}(x) + N_\Theta(x) + N_\Phi(x) \quad \forall x \in C. \tag{28}$$

To prove (27), by Theorem 3.3, we only need to show that

$$\begin{aligned} &\min\left\{\sum_{i=1}^{3}\|z_i\|^2 : z_1 \in N_{\mathcal{K}_k}(x),\ z_2 \in N_\Theta(x),\ z_3 \in N_\Phi(x),\ \sum_{i=1}^{3} z_i = z\right\} \\ &\leq \frac{1}{1-\tau}\|z\|^2 \quad \forall z \in N_C(x), \quad \forall x \in C. \end{aligned} \tag{29}$$

Pick any $x \in C$. For any $z \in N_C(x)$, by (28), there exist $z_1 \in N_{\mathcal{K}_k}(x)$, $z_2 \in N_\Theta(x)$ and $z_3 \in N_\Phi(x)$ such that $z = z_1 + z_2 + z_3$. Since $z_1 \in N_{\mathcal{K}_k}(x)$ and $\mathcal{K}_k$ is a linear space, by (6), it holds that $z_1 \perp w$ for any $w \in \mathcal{K}_k$. Write A and B as in (8). Let $I := \{i : b_i^T x = \gamma_i\}$. By Lemma 3.1, there exist $\lambda_i \in \mathbb{R}$ $(1 \leq i \leq m_1)$ and $\mu_j \geq 0$ $(j \in I)$ such that

$$z_2 = \sum_{i=1}^{m_1} \lambda_i a_i + \sum_{j \in I} \mu_j b_j.$$

It is obvious that $z_2 \in \mathcal{K}_k$, and therefore $z_1 \perp z_2$.

If $\|x\| < \Delta$, by $z_3 \in N_\Phi(x)$ and (6), we have $z_3 = \mathbf{0}_n$. Thus, $\|z\|^2 = \|z_1\|^2 + \|z_2\|^2$, and therefore (27) holds.

Now assume that $\|x\| = \Delta$. From (6), it follows that $N_\Phi(x) = \{tx : t \geq 0\}$. Then $z_3 = sx$ for some $s \geq 0$. Since $x \in C \subset \mathcal{K}_k$ and $z_1 \in \mathcal{K}_k^\perp$, we have $z_1 \perp z_3$. Thus, it holds that

$$\|z\|^2 = \|z_1\|^2 + \|z_2 + z_3\|^2. \tag{30}$$

We only need to treat the case when z_2 and z_3 are nonzero vectors. It is obvious that $x \in \Theta \cap \mathrm{bd}(\Phi)$. By (26), we have $\cos\angle(z_2, z_3) = \cos\angle(z_2, x) \geq -\tau$. Thus,

$$\|z_2 + z_3\|^2 = \|z_2\|^2 + \|z_3\|^2 + 2\|z_2\| \cdot \|z_3\| \cos\angle(z_2, z_3) \geq (1-\tau)(\|z_2\|^2 + \|z_3\|^2),$$

which, together with (30), implies

$$\|z\|^2 \geq (1-\tau)(\|z_1\|^2 + \|z_2\|^2 + \|z_3\|^2).$$

Since $x \in C$ and $z \in N_C(x)$ are arbitrary, (29) holds. The proof is complete. $\square$

By $\Omega = \Theta \cap \Phi$ and (27), we have the following result.

COROLLARY 3.6. *Suppose that the assumptions of Theorem* 3.5 *are satisfied. We have*

$$d(p, \Omega \cap \mathcal{K}_k) \leq \frac{1}{\sqrt{1-\tau}}\, d(p, \mathcal{K}_k) \quad \forall p \in \Omega. \tag{31}$$

4. Error bounds. In this section, we give some theoretical results on the error estimates of our method. First, we introduce some notation. Recall that p^* is the optimal solution of (1)–(4), σ^*, λ^*, and μ^* are multipliers satisfying (12)–(15), p_k is the optimal solution of (21)–(23), and $\mathcal{K}_k$ denotes $\mathcal{K}_k(M, [q, A^T, B^T])$. For any index set $I \subseteq \{1, \ldots, m_2\}$, B_I denotes the submatrix that consists of the rows of B indexed by I. In the rest of the paper, we use the notation $M^* := M + \sigma^* I_n$ and $q^* := q + A^T\lambda^* + B^T\mu^*$.

4.1. Error bounds for optimal value and optimal solution. In this subsection, we give the results on error bounds for $f(p_k) - f(p^*)$ and $\|p_k - p^*\|$. We show that $f(p_k) - f(p^*)$ and $\|p_k - p^*\|$ are bounded by quadratic and linear functions of $d(p^*, \mathcal{K}_k)$, respectively, which generalizes [41, Theorem 4.3]. The reason the error bounds involve the term $d(p^*, \mathcal{K}_k)$ can be explained as follows.

Suppose that M^* is positive definite and we use the Lanczos method to solve $M^*p^* = -q^*$. Let $\widetilde{K}_k = \mathcal{K}_k(M^*, q^*)$ and $\mathcal{C} := [q, A^T, B^T]$. By $M^* = M + \sigma^* I_n$ and (16), we have

$$\widetilde{K}_k = \mathrm{span}\{q^*, \ldots, (M^*)^{k-1}q^*\} = \mathrm{span}\{q^*, \ldots, M^{k-1}q^*\} \subset \mathcal{K}_k(M, \mathcal{C}) = \mathcal{K}_k.$$

Thus, by (5) and (17), it holds that

$$d(p^*, K_k) \leq d(p^*, \widetilde{K}_k) \leq \frac{d_{M^*}(p^*, \widetilde{K}_k)}{\sqrt{\theta_1^*}} \leq \frac{\sqrt{\theta_n^*}\|p^*\|}{\sqrt{\theta_1^*}\,\mathscr{T}_k(\frac{\kappa^*+1}{\kappa^*-1})} = \frac{\sqrt{\kappa^*}\|p^*\|}{\mathscr{T}_k(\frac{\kappa^*+1}{\kappa^*-1})}, \tag{32}$$

where $\theta_1^* \leq \cdots \leq \theta_n^*$ are eigenvalues of M^*, and κ^* is the condition number of M^*. By (32), we know that $d(p^*, K_k)$ converges to zero rapidly as k increases. If M^* is indefinite, then by (18) we can draw the same conclusion for $d(p^*, K_k)$.

Let $\bar{p} := P_{\Omega\cap\mathcal{K}_k}(p^*)$. Then $\|p^* - \bar{p}\| = d(p^*, \Omega \cap \mathcal{K}_k)$. The following result is related to some properties of $\bar{p}$.

LEMMA 4.1. *Suppose that Assumption* 2.1 *holds, and $p^* \notin \mathcal{K}_k$. Then $\|\bar{p}\| < \Delta$, $p^* - \bar{p} \perp \mathcal{K}_k$, and $Bp^* = B\bar{p}$.*

Proof. By (7), we have $p^* - \bar{p} \in N_{\Omega\cap\mathcal{K}_k}(\bar{p})$. Assume that $\|\bar{p}\| = \Delta$. Write A and B as in (8), and let $I(p)$ be defined by (9). By (28) and Lemma 3.1, there exist $\delta \geq 0$, $\lambda_i \in \mathbb{R}$ $(1 \leq i \leq m_1)$, $\mu_j \geq 0$ $(j \in I(\bar{p}))$, and $h \in \mathcal{K}_k^{\perp}$ such that

$$p^* - \bar{p} = \delta\bar{p} + \sum_{i=1}^{m_1} \lambda_i a_i + \sum_{j \in I(\bar{p})} \mu_j b_j + h. \tag{33}$$

Let $w := \delta\bar{p} + \sum_{i=1}^{m_1} \lambda_i a_i + \sum_{j\in I(\bar{p})} \mu_j b_j$. By $\bar{p} \in \mathcal{K}_k$ and the definition of $\mathcal{K}_k$, we know that $w \in \mathcal{K}_k$, and therefore $h \perp w$, and hence $\|p^*-\bar{p}\|^2 = \|w\|^2+\|h\|^2$. Since $\|p^*\| \le \Delta$ and $\|\bar{p}\| = \Delta$, we have $(p^*-\bar{p})^T\bar{p} \le 0$. From $Ap^* = A\bar{p}$ and $B_{I(\bar{p})}p^* \le B_{I(\bar{p})}\bar{p}$, it follows that $(p^*-\bar{p})^T a_i = 0$ $(1 \le i \le m_1)$ and $(p^*-\bar{p})^T b_j \le 0$ $(j \in I(\bar{p}))$. Thus, taking inner products with $p^*-\bar{p}$ on both sides of (33) yields $\|p^*-\bar{p}\|^2 \le (p^*-\bar{p})^T h \le \|p^*-\bar{p}\|\cdot\|h\|$, which, together with $\|p^*-\bar{p}\|^2 = \|w\|^2+\|h\|^2$, implies that $w = \mathbf{0}_n$ and $p^*-\bar{p} = h$. Since $p^* \notin \mathcal{K}_k$, we have $h \ne \mathbf{0}_n$. From $\bar{p} \in \mathcal{K}_k$ and $\mathbf{0}_n \ne h \in \mathcal{K}_k^{\perp}$, it follows that $\|p^*\| > \|\bar{p}\| = \Delta$, which is a contradiction. Thus, $\|\bar{p}\| < \Delta$.

Since $\|\bar{p}\| < \Delta$, we have $N_\Phi(\bar{p}) = \mathbf{0}_n$. Now we have $p^*-\bar{p} = \sum_{i=1}^{m_1} \lambda_i a_i + \sum_{j\in I(\bar{p})} \mu_j b_j + h$. Similarly to the proof above, we can prove that $p^*-\bar{p} = h$, which implies $p^*-\bar{p} \perp \mathcal{K}_k$. From $b_j \in \mathcal{K}_k$ for all $j = 1,\dots,m_2$, it follows that $Bp^* = B\bar{p}$. ◻

Under the second-order sufficient condition (for the definition, see [29, Theorem 12.6]), we can give the main result of this subsection, which is a generalization of [41, Theorem 4.3]. If $\|p^*\| = \Delta$, the second-order sufficient condition holding at p^* means that

$$M + \sigma^* I_n \text{ is positive definite on the kernel of the matrix } [p^* \;\; A^T \;\; B_{I(p^*)}^T].$$

By [8, Lemma 1.25], there exists $\varpi > 0$ such that

$$M^{**} := M^* + \varpi(p^*(p^*)^T + A^TA + B_{I(p^*)}^T B_{I(p^*)}) \tag{34}$$

is positive definite. For the case when $\|p^*\| < \Delta$, we can obtain (34) analogously.

THEOREM 4.2. *Consider problem* (1)–(4). *Suppose that Assumption* 2.1 *holds. The following assertions hold.*

(i) *Let $f(p)$ be defined by* (1). *Then*

$$0 \le f(p_k) - f(p^*) \le \frac{\|M\| + 2\sigma^*}{2(1-\tau)} d^2(p^*, \mathcal{K}_k).$$

(ii) *Suppose the second-order sufficient condition holds at p^*. Let M^{**}, ϖ be as in* (34), *and θ_1^{**} be the smallest eigenvalue of M^{**}. Then*

$$\|p_k - p^*\| \le \sqrt{\frac{\|M\| + 2\sigma^* + \varpi(\|B\|^2 + \Delta^2)}{\theta_1^{**}(1-\tau)}} \cdot d(p^*, \mathcal{K}_k).$$

Proof. (i) We only need to prove the second inequality. Let $\bar{p} := P_{\Omega\cap\mathcal{K}_k}(p^*)$. By (31), it holds that

$$\|p^* - \bar{p}\| = d(p^*, \Omega\cap\mathcal{K}_k) \le \frac{1}{\sqrt{1-\tau}}\, d(p^*, \mathcal{K}_k). \tag{35}$$

Let

$$L(p) := \frac{1}{2}p^TMp + q^Tp + \frac{1}{2}\sigma^*(\|p\|^2 - \Delta^2) + (\lambda^*)^T(Ap-\beta) + (\mu^*)^T(Bp-\gamma).$$

Then, we have $L(p^*) = f(p^*)$ and $\nabla L(p^*) = \mathbf{0}_n$. By Lemma 4.1, we have $Bp^* = B\bar{p}$ and $p^*-\bar{p} \perp \bar{p}$. Thus, it holds that

$$\begin{aligned} & f(\bar{p}) - f(p^*) \\ &= L(\bar{p}) - L(p^*) + \frac{1}{2}\sigma^*(\|p^*\|^2 - \|\bar{p}\|^2) + (\lambda^*)^T A(p^*-\bar{p}) + (\mu^*)^T B(p^*-\bar{p}) \\ &= \nabla L(p^*)^T(\bar{p}-p^*) + \frac{1}{2}(\bar{p}-p^*)^T(M+\sigma^* I)(\bar{p}-p^*) + \frac{1}{2}\sigma^*\|p^*-\bar{p}\|^2 \\ &\le \frac{\|M\| + 2\sigma^*}{2(1-\tau)} d^2(p^*, \mathcal{K}_k). \end{aligned} \tag{36}$$

Since p_k is the optimal solution of (21)–(23), it holds that $f(\bar{p}) \geq f(p_k)$, which, together with (36), implies the assertion.

(ii) Since $M^* p^* = -q^*$ and $Ap_k = Ap^*$, we have

$$\begin{aligned}
&\frac{1}{2}\|p_k - p^*\|^2_{M^{**}} \\
&= \frac{1}{2}p_k^T M^* p_k + p_k^T q^* + \frac{1}{2}p^{*T} M^* p^* + \frac{\varpi}{2}\|B_{I(p^*)}(p_k - p^*)\|^2 + \frac{\varpi}{2}((p_k - p^*)^T p^*)^2 \\
&\leq f(p_k) + \frac{1}{2}\sigma^* \Delta^2 + \beta^T \lambda^* + \gamma^T \mu^* + \frac{1}{2}(p^*)^T M^* p^* + \frac{\varpi}{2}(\|B\|^2 + \Delta^2)\|p_k - p^*\|^2 \\
&= f(p^*) + \frac{1}{2}\sigma^* \|p^*\|^2 + (p^*)^T (A^T \lambda^* + B^T \mu^*) + \frac{1}{2}(p^*)^T M^* p^* \\
&\quad + \frac{\|M\| + 2\sigma^* + \varpi(\|B\|^2 + \Delta^2)}{2(1-\tau)} d^2(p^*, \mathcal{K}_k) \qquad \text{(by (13)–(15), (35), and (36))} \\
&= \frac{\|M\| + 2\sigma^* + \varpi(\|B\|^2 + \Delta^2)}{2(1-\tau)} d^2(p^*, \mathcal{K}_k) \qquad (\text{by } M^* p^* = -q^*),
\end{aligned}$$

which, together with (5), implies the assertion. ∎

4.2. Error bounds for multipliers. In this subsection, we show that error bounds for the multipliers are of the same order as those for the optimal solution. Since p_k is the solution of (21)–(23), by [29, Theorem 12.8] it holds that $-\nabla f(p_k) \in N_C(p_k)$, where $C = \mathcal{K}_k \cap \Theta \cap \Phi$. This, together with (28), implies that there exist $\widetilde{\sigma}_k \geq 0$, $\widetilde{\lambda}_k$, $\widetilde{\mu}_k \geq \mathbf{0}$, and $h_k \in \mathcal{K}_k^\perp$ such that

$$(M + \widetilde{\sigma}_k I_n) p_k = -(q + A^T \widetilde{\lambda}_k + B^T \widetilde{\mu}_k) + h_k, \tag{37}$$

$$p_k \in \Omega, \tag{38}$$

$$\widetilde{\mu}_k^T (Bp_k - \gamma) = 0, \tag{39}$$

$$\widetilde{\sigma}_k (\Delta - \|p_k\|) = 0. \tag{40}$$

LEMMA 4.3. *We have*

$$\|h_k\| \leq (\|M\| + \sigma^*)\|p_k - p^*\|. \tag{41}$$

Proof. By (12) and (37), we have

$$M(p_k - p^*) + \widetilde{\sigma}_k p_k - \sigma^* p^* = A^T(\lambda^* - \widetilde{\lambda}_k) + B^T(\mu^* - \widetilde{\mu}_k) + h_k. \tag{42}$$

By $h_k \in \mathcal{K}_k^\perp$ and the definition of $\mathcal{K}_k$, it holds that $Ah_k = \mathbf{0}_{m_1}$ and $Bh_k = \mathbf{0}_{m_2}$. As $p_k \in \mathcal{K}_k$, we also have $p_k^T h_k = 0$. Thus, from (42), it follows that

$$h_k^T M(p_k - p^*) + \sigma^* h_k^T (p_k - p^*) = \|h_k\|^2,$$

which implies $\|h_k\|^2 \leq (\|M\| + \sigma^*)\|p_k - p^*\| \cdot \|h_k\|$. Then (41) follows. ∎

To give error bounds for the multipliers, we need the following result, which proves the boundedness of $\widetilde{\sigma}_k$.

LEMMA 4.4.

$$|\widetilde{\sigma}_k| \leq \frac{(3\|M\| + 2\sigma^*)\Delta + \|q\|}{\Delta\sqrt{1-\tau}}. \tag{43}$$

Proof. If $\|p_k\| < \Delta$, then $\widetilde{\sigma}_k = 0$. Thus, we only need to prove the case when $\|p_k\| = \Delta$. From (41), it follows that $\|h_k\| \le 2(\|M\| + \sigma^*)\Delta$, which, together with (37), implies

$$\|\widetilde{\sigma}_k p_k + A^T\widetilde{\lambda}_k + B^T\widetilde{\mu}_k\| \le \|M\|\Delta + \|q\| + 2(\|M\| + \sigma^*)\Delta. \tag{44}$$

Let $w := A^T\widetilde{\lambda}_k + B^T\widetilde{\mu}_k$. By $\widetilde{\mu}_k \ge \mathbf{0}_{m_2}$, (39), and Lemma 3.1, it holds that $w \in N_\Theta(p_k)$. Thus, by (26), we have $\angle(p_k, w) \ge -\tau$. Then

$$\|\widetilde{\sigma}_k p_k + w\|^2 = \|\widetilde{\sigma}_k p_k\|^2 + 2\|\widetilde{\sigma}_k p_k\| \cdot \|w\|\angle(p_k, w) + \|w\|^2 \ge (1-\tau)(\|\widetilde{\sigma}_k p_k\|^2 + \|w\|^2),$$

which, together with (44), implies (43). ∎

Remark 4.1. By (12)–(15), similarly to the proof above, we have

$$|\sigma^*| \le \frac{\|M\|\Delta + \|q\|}{\Delta\sqrt{1-\tau}}.$$

Hence, by (43), an upper bound of $\widetilde{\sigma}_k$ can be known in advance.

Let $I(p)$ be defined by (9). To obtain the main results, we need the assumption that the linear independence constraint qualification (LICQ) (see [29, Definition 12.4]) holds at p^*. Under the LICQ condition, the multipliers $(\sigma^*, \lambda^*, \mu^*)$ are unique. If $\|p^*\| = \Delta$, the LICQ means that $[p^*, A^T, B^T_{I(p^*)}]$ has full column rank.

In the following, we use the notation $U := [p^*, A^T, B^T_{I(p^*)}]$ and $\mathfrak{U} := [U^TU]^{-1}U^T$.

THEOREM 4.5. *Suppose that Assumption* 2.1 *holds,* $\|p^*\| = \Delta$*, and the LICQ holds at* p^**. If* $\|p_k - p^*\|$ *is small enough such that* $I(p_k) \subseteq I(p^*)$*, then*

$$\left\| \begin{array}{c} \widetilde{\sigma}_k - \sigma^* \\ \widetilde{\lambda}_k - \lambda^* \\ \widetilde{\mu}_k - \mu^* \end{array} \right\| \le \rho\ \|p_k - p^*\|, \tag{45}$$

where

$$\rho := \|\mathfrak{U}\| \cdot \left[(2\|M\| + \sigma^*) + \frac{(3\|M\| + 2\sigma^*)\Delta + \|q\|}{\Delta\sqrt{1-\tau}} \right].$$

Proof. Let $w := A^T(\lambda^* - \widetilde{\lambda}_k) + B^T(\mu^* - \widetilde{\mu}_k)$. Write (42) as

$$M(p_k - p^*) + \widetilde{\sigma}_k(p_k - p^*) - h_k = w + (\sigma^* - \widetilde{\sigma}_k)p^*,$$

which, together with (41) and (43), implies that

$$\begin{aligned} \|(\sigma^* - \widetilde{\sigma}_k)p^* + w\| &\le \left(2\|M\| + \sigma^* + \frac{(3\|M\| + 2\sigma^*)\Delta + \|q\|}{\Delta\sqrt{1-\tau}} \right) \|p_k - p^*\| \\ &= (\rho/\|\mathfrak{U}\|) \cdot \|p_k - p^*\|. \end{aligned} \tag{46}$$

Since $I(p_k) \subseteq I(p^*)$, we have $B^T(\mu^* - \widetilde{\mu}_k) = B^T_{I(p^*)}[\mu^*_{I(p^*)} - (\widetilde{\mu}_k)_{I(p^*)}]$. By the definitions of $\mathfrak{U}$, it holds that

$$\left\| \begin{array}{c} \widetilde{\sigma}_k - \sigma^* \\ \widetilde{\lambda}_k - \lambda^* \\ \widetilde{\mu}_k - \mu^* \end{array} \right\| = \left\| \mathfrak{U} \cdot U \begin{pmatrix} \sigma^* - \widetilde{\sigma}_k \\ \lambda^* - \widetilde{\lambda}_k \\ \mu^*_{I(p^*)} - (\widetilde{\mu}_k)_{I(p^*)} \end{pmatrix} \right\| \le \|\mathfrak{U}\| \cdot \|(\sigma^* - \widetilde{\sigma}_k)p^* + w\|,$$

which, together with (46), implies (45). ∎

Remark 4.2. For the case when $\|p^*\| < \Delta$, let $U := [A^T\ B^T_{I(p^*)}]$ and $\mathfrak{U} := [U^TU]^{-1}U^T$. Using the same argument as in Theorem 4.5, we can derive an inequality similar to (45).

5. Algorithm descriptions. In this section, we present the block Lanczos method for the problem (1)–(4). Let $\mathcal{C} := [q, A^T, B^T]$ and $m = m_1 + m_2$. We make the following assumption.

Assumption 5.1. The matrix $[q, A^T, B^T]$ has full column rank.

If the above condition is not satisfied, then, by replacing $\mathcal{K}_k(M, \mathcal{C})$ with $\mathcal{K}_k(M, \mathcal{D})$, we can derive similar results, where $\mathcal{D}$ is the full rank submatrix of $\mathcal{C}$ such that $\mathcal{R}(\mathcal{D}) = \mathcal{R}(\mathcal{C})$ (here $\mathcal{R}$ represents the range of the matrix).

The first step of our method is to apply Algorithm 1 to the matrices M and $\mathcal{C}$. Then we obtain the Krylov subspace $\mathcal{K}_k(M, \mathcal{C})$ and the sequences $\{Q_k, V_k, W_k, T_k\}$, which satisfy (19), (20), and

$$MQ_k = Q_k T_k + W_{k+1}(E^{k(m+1)}_{(m+1)-})^T. \tag{47}$$

Note that $V_k \in \mathbb{R}^{n\times(m+1)}$ for all k, $Q_k \in \mathbb{R}^{n\times(m+1)}$, and $T_k \in \mathbb{R}^{(m+1)\times(m+1)}$.

From Algorithm 1, we have

$$[q, A^T, B^T] = V_1 R = V_1[u, R_1, R_2] = [V_1 u, V_1 R_1, V_1 R_2], \tag{48}$$

where $u \in \mathbb{R}^{m+1}$, $R_1 \in \mathbb{R}^{(m+1)\times m_1}$, and $R_2 \in \mathbb{R}^{(m+1)\times m_2}$. It then follows directly that

$$Q_k^T M Q_k = T_k \in \mathbb{R}^{k(m+1)\times k(m+1)}, \tag{49}$$

$$Q_k^T q = \begin{bmatrix} u \\ \mathbf{0}_{(k-1)(m+1)} \end{bmatrix} := \widetilde{q} \in \mathbb{R}^{k(m+1)}, \tag{50}$$

$$Q_k^T A^T = \begin{bmatrix} R_1 \\ \mathbf{0}_{[(k-1)(m+1)]\times m_1} \end{bmatrix} := \widetilde{A}^T \in \mathbb{R}^{k(m+1)\times m_1}, \tag{51}$$

$$Q_k^T B^T = \begin{bmatrix} R_2 \\ \mathbf{0}_{[(k-1)(m+1)]\times m_2} \end{bmatrix} := \widetilde{B}^T \in \mathbb{R}^{k(m+1)\times m_2}, \tag{52}$$

$$q = Q_k \widetilde{q}, \tag{53}$$

$$A^T = Q_k \widetilde{A}^T, \tag{54}$$

$$B^T = Q_k \widetilde{B}^T, \tag{55}$$

where (53)–(55) follow from (20) and (48).

Substitute p by $Q_k x$ in (21)–(23). By (49)–(55), we obtain the following problem:

$$\min_{x\in\mathbb{R}^{k(m+1)}} \quad \widetilde{f}(x) = \frac{1}{2}x^T T_k x + \widetilde{q}^T x \tag{56}$$

$$\text{s.t.} \quad \widetilde{A}x = \beta, \tag{57}$$

$$\widetilde{B}x \le \gamma, \tag{58}$$

$$\|x\|_2 \le \Delta. \tag{59}$$

Let $\widetilde{\Delta}_{\min} := \min_{x\in\widetilde{\Theta}} \|x\|_2$, where $\widetilde{\Theta} = \{x \mid \widetilde{A}x = \beta,\ \widetilde{B}x \le \gamma\}$. By (24), we have $\widetilde{\Delta}_{\min} = \Delta_{\min}$, and therefore (56)–(59) are also in the interesting case.

Let x^* be the optimal solution of (56)–(59). Then $p_k = Q_k x^*$ is an optimal solution of (21)–(23). Let $\{\widetilde{\sigma}_k, \widetilde{\lambda}_k, \widetilde{\mu}_k, h_k\}$ be the Lagrange multipliers of (21)–(23) corresponding to p_k. From (49)–(55), we can see that $\{\widetilde{\sigma}_k, \widetilde{\lambda}_k, \widetilde{\mu}_k\}$ are the Lagrange multipliers of (56)–(59) corresponding to x^*.

In our implementation, we choose a k satisfying $k > \lceil \frac{n}{3(m+1)} \rceil$ as the initial value, and solve the projected ETR subproblem (56)–(59) to get the optimal solution x^*, and then set $p_k = Q_k x^*$ as the approximate solution. If the algorithm does not reach the desired precision, then we increase k by 5 and recalculate p_k. We will terminate our algorithm if either the stop criterion is satisfied or the dimension of the Krylov subspace exceeds $\frac{n}{2}$. The final algorithm is summarized by Algorithm 2.

Algorithm 2 The block Lanczos method for the ETR subproblem.

Input:

M, q, A, β, B, γ, Δ, k.

Output:

the approximate solution d_k to the original problem (1)–(4).

1: Apply Algorithm 1 to the matrices M and $\mathcal{C}$ to get Q_k, T_k, and W_{k+1}.
2: $\widetilde{q}(1) = \|q\|_2$; $\widetilde{q}(2 : k(m+1)) = \mathbf{0}_{k(m+1)-1}$.
3: $\widetilde{A}^T(1 : m+1, :) = R(:, 2 : m_1 + 1)$; $\widetilde{A}^T(m+1 : k(m+1), :) = \mathbf{0}_{[(k-1)(m+1)] \times m_1}$.
4: $\widetilde{B}^T(1 : m+1, :) = R(:, m_1+2 : m+1)$; $\widetilde{B}^T(m+1 : k(m+1), :) = \mathbf{0}_{[(k-1)(m+1)] \times m_2}$.
5: Solve (56)–(59) by the active-set method or the TOMLAB solver to get the solution x^*.
6: Set $p_k = Q_k x^*$.
7: If the stop criterion is satisfied or $k(m+1) > \frac{n}{2}$, then stop.
8: $k_1 = k$; $k = k + 5$.
9: Continue block Lanczos algorithm (Ruhe's variant).
10: $\widetilde{q}((k_1(m+1)+1) : k(m+1)) = \mathbf{0}_{5(m+1)}$.
11: $\widetilde{A}^T((k_1(m+1)+1) : k(m+1), :) = \mathbf{0}_{5(m+1) \times m_1}$.
12: $\widetilde{B}^T((k_1(m+1)+1) : k(m+1), :) = \mathbf{0}_{5(m+1) \times m_2}$.
13: Go to 5.

The following result, which is useful in establishing the stop criterion, is a generalization of [20, Theorem 5.1].

THEOREM 5.1. *Let h_k be as in* (37). *Then we have*

$$h_k = W_{k+1} (E_{(m+1)-}^{k(m+1)})^T x^*, \tag{60}$$

and

$$\varrho = \|h_k\|_2 = \|W_{k+1} x^* \big([(k-1)(m+1)+1] : k(m+1)\big)\|_2. \tag{61}$$

Proof. We only need to prove (60). By

$$\begin{aligned}
&(M + \widetilde{\sigma}_k I_n) p_k \\
&= M Q_k x^* + \widetilde{\sigma}_k Q_k x^* \\
&= (Q_k T_k + W_{k+1} (E_{(m+1)-}^{k(m+1)})^T) x^* + \widetilde{\sigma}_k Q_k x^* && \text{(by (47))} \\
&= Q_k (T_k + \widetilde{\sigma}_k I_n) x^* + W_{k+1} (E_{(m+1)-}^{k(m+1)})^T x^* \\
&= -Q_k (\widetilde{q} + \widetilde{A}^T \widetilde{\lambda}_k + \widetilde{B}^T \widetilde{\mu}_k) + W_{k+1} (E_{(m+1)-}^{k(m+1)})^T x^* && \text{(by (37) and } h_k \perp \mathcal{K}_k(M, \mathcal{C})) \\
&= -(q + A^T \widetilde{\lambda}_k + B^T \widetilde{\mu}_k) + W_{k+1} (E_{(m+1)-}^{k(m+1)})^T x^* && \text{(by (53)–(55))}
\end{aligned}$$

and (37), we have that (60) holds. ∎

5.1. Active-set method. We now describe the active-set method for the projected ETR subproblem (56)–(59). In fact, (56)–(59) is just a small-scale ETR subproblem. Thus, we will propose an active-set method for the original ETR subproblem (1)–(4) instead.

For the convenience of notation, let $\mathcal{E} = \{1, \dots, m_1\}$, $\mathcal{I} = \{m_1+1, \dots, m_1+m_2\}$. Let $d_i = a_i$, $h_i = \beta_i$ for all $i \in \mathcal{E}$, and $d_j = b_{j-m_1}$, $h_j = \gamma_{j-m_1}$ for all $j \in \mathcal{I}$. In the active-set method, all iterates are feasible points of the ETR subproblem. At current iterate p_k, we need to solve the problem

$$\begin{aligned} \text{(62)} \qquad & \min \quad f(p_k + s) = \frac{1}{2}(p_k + s)^T M(p + s) + q^T(p_k + s) \\ & \text{s.t.} \quad d_i^T(p_k + s) = h_i, \ i \in \mathcal{W}_k, \ \|p_k + s\| \le \Delta, \end{aligned}$$

where $\mathcal{W}_k$ is the current working set, which is the active set of p_k and serves as a guess of active set of the optimal solution. Let $m_k = Mp_k + q$. Then (62) can be written as

$$\text{(63)} \qquad \min \quad \frac{1}{2}s^T M s + m_k^T s$$

$$\text{(64)} \qquad \text{s.t.} \quad d_i^T s = 0, \quad i \in \mathcal{W}_k,$$

$$\text{(65)} \qquad \|s + p_k\| \le \Delta.$$

Denote the solution of this subproblem by s_k. If $s_k \neq \mathbf{0}_n$, the next iterate point is $p_{k+1} = p_k + \alpha_k s_k$, where

$$\text{(66)} \qquad \alpha_k := \min\left(1, \min_{i \notin \mathcal{W}_k, d_i^T s_k > 0} \frac{h_i - d_i^T p_k}{d_i^T s_k}\right).$$

We call the constraints i for which the minimum in (66) is achieved the blocking constraints. If $\alpha_k < 1$, a new working set $\mathcal{W}_{k+1}$ is constructed by adding one of the blocking constraints to $\mathcal{W}_k$, i.e.,

$$\text{(67)} \qquad \mathcal{W}_{k+1} = \mathcal{W}_k \cup \{i\}, \quad i = \arg\min\left\{1, \frac{h_j - d_j^T p_k}{d_j^T s_k} \;\middle|\; j \notin \mathcal{W}_k, \ d_j^T s_k > 0\right\}.$$

If $s_k = \mathbf{0}_n$, there exist $\sigma_k \ge 0$, and μ_i^k, $i \in \mathcal{W}_k$, such that

$$\text{(68)} \qquad (M + \sigma_k I)p_k + q + \sum_{i \in \mathcal{W}_k} \mu_i^k d_i = \mathbf{0}_n.$$

If μ_i^k is nonnegative for all $i \in \mathcal{W}_k \cap \mathcal{I}$, p_k is the solution to the original problem. If, on the other hand, one or more of the multipliers μ_j^k, $j \in \mathcal{W}_k \cap \mathcal{I}$, is negative, then we remove the index j corresponding to one of the negative multipliers from the working set and solve the new subproblem

$$\begin{aligned} \text{(69)} \qquad & \min \quad \frac{1}{2}s^T M s + m_k^T s \\ & \text{s.t.} \quad d_i^T s = 0, \quad i \in \mathcal{W}_k \backslash \{j\}, \\ & \qquad \|s + p_k\|_2 \le \Delta \end{aligned}$$

for the new step.

We now present the formal specification of the algorithm (see Algorithm 3).

Algorithm 3 Active-set method.

1: Choose an initial point p_0 and the initial working set $\mathcal{W}_0$;
2: **for** $k = 0, 1, 2, \ldots$ **do**
3:   solve (63)–(65) to find s_k and the corresponding Lagrange multipliers μ_i^k and σ_k;
4:   **if** $s_k = \mathbf{0}_n$ **then**
5:     **if** $\mu_i^k \geq 0$ for all $i \in \mathcal{W}_k \cap \mathcal{I}$ **then**
6:       stop with the solution $p^* = p_k$;
7:     **else**
8:       $j = \arg\min_{j \in \mathcal{W}_k \cap \mathcal{I}} \mu_j^k$;
9:       $p_{k+1} = p_k$;
10:       $\mathcal{W}_{k+1} = \mathcal{W}_k \backslash \{j\}$;
11:     **end if**
12:   **else**
13:     compute α_k from (66);
14:     $p_{k+1} = p_k + \alpha_k s_k$;
15:     **if** there are blocking constraints **then**
16:       obtain $\mathcal{W}_{k+1}$ from (67);
17:     **else**
18:       $\mathcal{W}_{k+1} = \mathcal{W}_k$;
19:     **end if**
20:   **end if**
21: **end for**

Under the condition that M is positive definite, we can prove that the active-set method is convergent. First, we show that, after deleting one constraint in step 10 of Algorithm 3, the direction s_k at the next iteration is feasible with respect to the dropped constraint.

THEOREM 5.2. *Suppose that M is positive definite, and that* (68) *holds and there is an index $j \in \mathcal{W}_k \cap \mathcal{I}$ such that $\mu_j^k < 0$. Let s be the nonzero solution of problem* (69). *Then $d_j^T s < 0$ and $m_k^T s < 0$, where $m_k = Mp_k + q$.*

Proof. Since M is positive definite, s must be the unique optimal solution of (69). Thus, we have $\frac{1}{2} s^T Mx + m_k^T s < 0$, which implies that $m_k^T s < 0$. Taking inner products with s on both sides of (68), and using the fact that $d_i^T s = 0$ for all $i \in \mathcal{W}_k \backslash \{j\}$, we have

$$m_k^T s + \sigma_k s^T p_k + \mu_j^k d_j^T s = 0.$$

If $\|p_k\|_2 = \Delta$, then we get from $\|s + p_k\|_2 \leq \Delta$ that $s^T p_k < 0$, which, together with $\sigma_k \geq 0$ and $m_k^T s < 0$, implies $\mu_j^k d_j^T s > 0$. If $\|p_k\|_2 < \Delta$, then we have $\sigma_k = 0$, and we also have $\mu_j^k d_j^T s > 0$. Since $\mu_j^k < 0$, it follows that $d_j^T s < 0$. ∎

The proof of the following result is almost the same as that in [29, Page 477]. For completeness, we give it below.

THEOREM 5.3. *Suppose that M is positive definite, and the step length α_k in step* 13 *of Algorithm* 3 *is always greater than zero. Then Algorithm* 3 *identifies the solution p^* in a finite number of iterations.*

Proof. By theorem 5.2, s_k in step 13 is a strict descent direction for $f(\cdot)$. Thus, by $0 < \alpha_k \le 1$, the value of f is less than $f(p_k)$ at all subsequent iterations. It follows that the algorithm can never return to the working set $\mathcal{W}_k$.

The algorithm encounters an iterate k for which $s_k = \mathbf{0}_n$ solves (63)–(65) at least on every $(m_1 + m_2 + 1)$th iteration. To demonstrate this claim, we note that, for any k at which $s_k \neq \mathbf{0}_n$, either we have $\alpha_k = 1$ (in which case we reach the minimizer of f on the current working set $\mathcal{W}_k$, so that the next iteration will yield $s_{k+1} = \mathbf{0}_n$), or else a constraint is added to the working set $\mathcal{W}_k$. Since the number of indices in $\mathcal{W}_k$ is less than $m_1 + m_2 + 1$, the algorithm will encounter the $s_k = \mathbf{0}_n$ case after at most $m_1 + m_2 + 1$ iterations.

Thus, the algorithm finds the global minimum of f on its current working set periodically, and, having done so, it never visits this working set again. Since there are only a finite number of possible working sets, the algorithm will terminate at the solution p^*. ∎

We should note that, when M is indefinite, if we adopt the strategy of using the alternating direction method of multipliers (see the next subsection) to get the initial point and the initial working set, then the active-set method also converges in practice. Therefore, in our numerical experiments, we also test our method on the ETR problem for the case when M is indefinite.

5.2. ADMM for the initial point and the working set. For the active-set method, different choices of the initial working set lead to different iteration sequences. If we choose the initial working set properly, the active-set method may converge by a few iterations. In the implementation of our method, we use the alternating direction method of multipliers (ADMM) to get an initial point p_0 and an initial working set $\mathcal{W}_0$. The ADMM has been well studied in the literature, and therefore we will not discuss it in detail. We refer the reader to [11] for some reviews on the ADMM.

We may rewrite the ETR subproblem equivalently as

$$
\begin{aligned}
\min \quad & \frac{1}{2}p^T M p + q^T p \\
\text{s.t.} \quad & Ap = \beta, \\
& Bp + y - \gamma = \mathbf{0}_n, \\
& p - z = \mathbf{0}_n, \\
& \|z\|_2 \le \Delta, \\
& y \ge \mathbf{0}_n.
\end{aligned}
\tag{70}
$$

The augmented Lagrangian of the above problem is

$$
\begin{aligned}
\mathcal{L}_\rho(p, y, z, \lambda_1, \lambda_2, \lambda_3) = {} & \frac{1}{2}p^T M p + q^T p - \lambda_1^T(Ap - \beta) - \lambda_2^T(Bp + y - \gamma) - \lambda_3^T(p - z) \\
& + \frac{\rho}{2}\|Ap - \beta\|_2^2 + \frac{\rho}{2}\|Bp + y - \gamma\|_2^2 + \frac{\rho}{2}\|p - z\|_2^2,
\end{aligned}
$$

where $\rho > 0$ is a parameter. So the iterative scheme of the ADMM for problem (70) is

$$\begin{aligned}
p_{k+1} &= \arg\min \mathcal{L}_\rho(p, y_k, z_k, \lambda_{1,k}, \lambda_{2,k}, \lambda_{3,k}),\\
y_{k+1} &= \arg\min_{y \geq \mathbf{0}_n} \mathcal{L}_\rho(p_{k+1}, y, z_k, \lambda_{1,k}, \lambda_{2,k}, \lambda_{3,k}),\\
z_{k+1} &= \arg\min_{\|z\| \leq \Delta} \mathcal{L}_\rho(p_{k+1}, y_{k+1}, z, \lambda_{1,k}, \lambda_{2,k}, \lambda_{3,k}),\\
\lambda_{1,k+1} &= \lambda_{1,k} - \rho(Ap_{k+1} - \beta),\\
\lambda_{2,k+1} &= \lambda_{2,k} - \rho(Bp_{k+1} + y_{k+1} - \gamma),\\
\lambda_{3,k+1} &= \lambda_{3,k} - \rho(p_{k+1} - z_{k+1}).
\end{aligned}$$

If M is positive definite, then, by [16, Theorem 3.1], $\{p_k, y_k, z_k\}$ converges to the solution of (70). When M is indefinite, if $\{p_k, y_k, z_k, \lambda_{1,k}, \lambda_{2,k}, \lambda_{3,k}\}$ has a cluster point $\{p^*, y^*, z^*, \lambda_1^*, \lambda_2^*, \lambda_3^*\}$, then these points are critical points of $\mathcal{L}_\rho(\cdot)$, by [22, Theorem 1]. When (p^*, y^*, z^*) is feasible in (70), $\{p^*, \lambda_1^*, \lambda_2^*, \lambda_3^*\}$ is a KKT point of the ETR subproblem (1)–(4). Since the ADMM converges to modest accuracy very quickly, we use it to get the initial point and the initial working set. Numerical experiments reveal that this strategy is very efficient.

6. Numerical results. In this section, we report our numerical experiments comparing our method with the TOMLAB solver [24] for solving the problem (1)–(4). The TOMLAB optimization environment is a powerful optimization platform and modeling language for solving optimization problems. TOMLAB has a wide range of optimization toolboxes that support global optimization, integer programming, linear, quadratic programming, etc., for MATLAB optimization problems.[1] Our objective is to show the efficiency of the block Lanczos method for large-scale ETR subproblems. For each instance, we use three methods to solve it:

(1) the solver combining the block Lanczos method with the active-set algorithm, denoted by B-A;
(2) the solver combining the block Lanczos method with the TOMLAB solver, denoted by B-T;
(3) the TOMLAB solver, denoted by TL.

We evaluate the performance of these methods for the following three cases:

(a) M is a large-scale, dense, positive definite matrix;
(b) M is a large-scale, dense, indefinite matrix;
(c) M is a large-scale sparse matrix.

All experiments are conducted in MATLAB R2015b on a Dell desktop computer Intel Core i5 with 3.20 GHz CPU (s) and 4.00 GB RAM.

For cases (a) and (b), we generate random instances of the problem (1)–(4) for n varying from 4000 to 10000, and m varying from 10 to 50. For every given (n, m), we randomly generate 10 instances and record the average numerical performance of these instances.

For case (a), the matrix M can be generated as follows:

$$Q = \mathrm{orth}(\mathrm{rand}(n, n)), \quad \vartheta = \mathrm{unifrnd}(1, 1e6, 1, n), \quad \text{and} \quad M = Q^T \mathrm{diag}(\vartheta) Q. \tag{71}$$

In this way, the condition number of M ranges from 10^2 to 10^6. Next, we generate p^*, $\sigma^* \geq 0$, $\lambda^* \geq \mathbf{0}_{m_1}$, $\mu^* \geq \mathbf{0}_{m_2}$, and $\upsilon \geq \mathbf{0}_{m_2}$ randomly. Let $\beta = Ap^*$, $\gamma = Bp^* + \upsilon$, and let $q = -(M + \sigma^* I)p^* - A^T\lambda^* - B^T\mu^*$. Then we generate a random instance of the ETR subproblem with the optimal solution p^*.

[1] For more information, see https://tomopt.com.

For case (b), we generate Q and ϑ as in (71). Let E be the set whose elements are randomly selected from $\{1,\ldots,n\}$, and let ϑ' be the vector such that $\vartheta'_i = -\vartheta_i$ for all $i \in E$ and $\vartheta'_i = \vartheta_i$ otherwise. Then let $M = Q^T \mathrm{diag}(\vartheta')Q$. Next, we generate a random instance, as above, except that we choose σ^* such that, for some $\sigma > 0$, $M + \sigma^* I_n + \sigma(A^T A + B^T_{I(p^*)} B_{I(p^*)})$ is positive definite.

Table 1
Test matrices M.

Matrix	n	nnz	Sparsity	Application
fxm3_6	5026	94026	0.37%	Optimization problem
man_5796	5976	225046	0.63%	Structural problem
EX6	6545	295680	0.69%	Combinatorial problem
nd3k	9000	3279690	4.05%	2D/3D problem
nemeth01	9506	725054	0.80%	Theoretical/quantum chemistry problem sequence
net25	9520	401200	0.44%	Optimization problem
fv1	9604	85264	0.09%	2D/3D problem
flowmeter5	9669	67391	0.07%	Model reduction problem
PGPgiantcompo	10680	48632	0.04%	Undirected multigraph
rajat06	10922	46983	0.04%	Circuit simulation problem
t2dah	11445	176117	0.13%	Model reduction problem
ca-HepPh	12008	237010	0.16%	Undirected graph
stokes64s	12546	140034	0.09%	Computational fluid dynamics problem
Pres_Poisson	14822	715804	0.33%	Computational fluid dynamics problem
OPF_3754	15435	141478	0.06%	Power network problem
Dubcova1	16129	253009	0.10%	2D/3D problem
ramage02	16830	2866352	1.01%	Computational fluid dynamics problem
cvxqp3	17500	114962	0.04%	Optimization problem
memplus	17758	99147	0.03%	Circuit simulation problem
ca-AstroPh	18772	396160	0.11%	Undirected graph
Si5H12	19896	738598	0.19%	Theoretical/quantum chemistry problem
chipcool0	20082	281150	0.07%	Model reduction problem
ns3Da	20414	1679599	0.40%	Computational fluid dynamics problem

For case (c), our test sparse matrices, whose dimensions are greater than 5000, are taken from the University of Florida sparse matrix collection [18]. The detailed characteristics of these matrices are listed in Table 1, where n is the dimension, nnz is the number of nonzero entries, and nnz/n^2 is the sparsity. Since some matrices are not symmetric, when testing our algorithms we set $M = (M + M^T)/2$. For each M, 10 instances of β, γ, and q are generated randomly as in cases (a) and (b), and we record the average numerical performance of these instances.

As discussed in section 4, the block Lanczos method for (1)–(4) can be regarded as solving (12) by the Lanczos method. By Theorem 4.2 and Lemma 4.3, the value of ϱ in (61) converges to zero as k increases. Thus, we terminate our method if either

$$\|W_{k+1} x^*([(k-1)(m+1)+1] : k(m+1))\|_2 \le 10^{-12}$$

or $k(m+1) > \frac{n}{2}$ holds.

6.1. Testing for large-scale, dense, positive definite matrices. All numerical results are summarized in Tables 2–7. In each table, "CPU (s)" represents the computational time and "Err" represents the relative error of the ETR subproblem

$$e = \frac{\|p^* - \tilde{p}\|_2}{\|p^*\|_2},$$

where p^* is the exact solution of the original problem (1)–(4), and $\tilde{p}$ is the calculated solution. When the average computational time of a solver is more than three hours, we indicate it by the symbol "$*$".

Table 2
Comparison with the TOMLAB solver when M is positive definite.

		$n = 4000$			$n = 6000$		
		B-A	B-T	TL	B-A	B-T	TL
$m_1 = 10$	CPU (s)	15.1571	25.6515	52.7849	27.5740	29.9247	104.9745
$m_2 = 10$	Err	2.0810e−09	5.7244e−08	2.2621e−06	1.0513e−09	7.1562e−07	3.7539e−06
$m_1 = 20$	CPU (s)	21.8549	21.5344	91.4193	41.7945	31.2068	201.4890
$m_2 = 20$	Err	5.0751e−09	4.2942e−08	1.9014e−06	1.5343e−08	6.6164e−08	3.2196e−06
$m_1 = 30$	CPU (s)	46.3277	23.9315	189.2039	109.9492	34.9728	484.0337
$m_2 = 30$	Err	2.0723e−09	2.7647e−08	1.6441e−06	5.2772e−08	4.5793e−08	3.0061e−06

Table 3
Numerical results when n is large.

			$m_1 = 10$, $m_2 = 10$	$m_1 = 20$, $m_2 = 20$	$m_1 = 30$, $m_2 = 30$	$m_1 = 40$, $m_2 = 40$	$m_1 = 50$, $m_2 = 50$
$n = 8000$	B-A	CPU (s)	68.2418	116.8027	125.8394	163.2301	224.2105
		Err	1.6975e−12	1.2806e−12	2.0093e−10	5.7495e−09	9.1830e−08
	B-T	CPU (s)	51.5802	52.4968	53.9222	54.9941	56.0875
		Err	1.4163e−07	7.7029e−08	9.4726e−08	6.9037e−08	1.3571e−07
	TL	CPU (s)	202.5817	507.4010	1.0181e3	3.7342e3	4.8714e3
		Err	6.7550e−06	6.6103e−06	5.5764e−06	5.8541e−06	4.2033e−06
$n = 10000$	B-A	CPU (s)	59.6534	127.6969	379.6404	456.4134	478.6114
		Err	1.6676e−12	8.1241e−13	4.7001e−12	5.6368e−10	3.8120e−09
	B-T	CPU (s)	73.5004	81.4680	88.2570	86.9354	89.1433
		Err	1.6409e−07	1.1271e−07	1.0400e−07	9.2842e−08	8.8812e−08
	TL	CPU (s)	*	*	*	*	*
		Err	*	*	*	*	*

Table 4
Comparison with the TOMLAB solver when M is indefinite.

		$n = 4000$			$n = 6000$		
		B-A	B-T	TL	B-A	B-T	TL
$m_1 = 10$	CPU (s)	22.8261	25.2476	51.8642	40.9477	33.8412	111.7909
$m_2 = 10$	Err	3.8388e−08	4.3245e−06	4.3245e−06	2.1695e−09	2.5145e−07	7.4381e−06
$m_1 = 20$	CPU (s)	42.7168	22.1512	96.8620	63.3341	32.0591	184.8249
$m_2 = 20$	Err	1.1896e−08	1.4162e−07	2.6278e−06	9.0956e−08	1.5344e−07	6.7169e−06
$m_1 = 30$	CPU (s)	54.2135	24.4696	194.8940	144.5491	34.8273	511.5421
$m_2 = 30$	Err	1.3748e−08	8.4309e−08	3.1201e−06	2.7689e−08	1.4530e−07	3.8339e−06

Table 5
Numerical results when n is large.

			$m_1 = 10$, $m_2 = 10$	$m_1 = 20$, $m_2 = 20$	$m_1 = 30$, $m_2 = 30$	$m_1 = 40$, $m_2 = 40$	$m_1 = 50$, $m_2 = 50$
$n = 8000$	B-A	CPU (s)	97.6684	240.4277	250.6100	233.9501	252.6979
		Err	1.0858e−12	1.3739e−10	7.0969e−10	5.4589e−07	7.0781e−07
	B-T	CPU (s)	51.1267	52.0337	55.8732	57.1357	57.1721
		Err	4.3554e−07	1.3503e−07	3.2103e−07	2.4184e−07	1.2503e−06
	TL	CPU (s)	213.9407	487.2720	1.1052e3	3.7631e3	4.9869e3
		Err	6.7550e−06	6.6103e−06	5.5764e−06	5.8541e−06	4.2033e−06
$n = 10000$	B-A	CPU (s)	252.6979	321.5923	653.0346	741.8329	926.1189
		Err	5.9636e−12	2.8035e−11	2.5723e−11	1.7367e−09	2.1623e−08
	B-T	CPU (s)	75.1252	83.4110	86.3980	88.7041	89.3292
		Err	4.3320e−07	2.0314e−07	1.8428e−07	1.8068e−07	1.4978e−07
	TL	CPU (s)	*	*	*	*	*
		Err	*	*	*	*	*

Table 6
Numerical results for sparse matrices whose dimensions are less than 10000.

Matrix			$m_1 = 10$ $m_2 = 10$	$m_1 = 20$ $m_2 = 20$	$m_1 = 30$ $m_2 = 30$	$m_1 = 40$ $m_2 = 40$	$m_1 = 50$ $m_2 = 50$
fxm3_6	B-A	CPU (s)	9.7837	16.9676	39.5539	14.2081	13.5771
		Err	1.7315e−15	3.0296e−15	3.0337e−15	8.6302e−15	6.7382e−15
	B-T	CPU (s)	60.5393	24.8962	24.6642	83.4317	79.9378
		Err	6.3091e−07	1.2353e−06	3.1814e−06	6.7825e−07	5.0658e−07
	TL	CPU (s)	143.1346	202.5209	324.3657	429.5980	479.5999
		Err	9.3401e−06	1.4109e−05	2.3784e−05	1.6187e−05	2.6957e−05
man_5976	B-A	CPU (s)	33.5854	24.4866	36.5061	116.2190	31.6673
		Err	2.7262e−15	3.4997e−15	6.8693e−15	5.0344e−15	2.8130e−14
	B-T	CPU (s)	285.6493	58.6651	62.7424	29.0201	71.6723
		Err	4.4326e−05	1.9740e−06	3.5521e−06	4.5179e−06	5.0060e−06
	TL	CPU (s)	104.7539	215.8944	361.3810	433.3687	654.2471
		Err	7.6282e−04	1.0358e−05	2.3437e−05	2.4273e−05	2.4841e−05
EX6	B-A	CPU (s)	26.8690	11.5208	27.7868	47.2995	104.8835
		Err	2.3332e−15	2.5367e−14	2.0536e−08	1.2869e−09	1.3946e−10
	B-T	CPU (s)	72.4482	1.2518e3	1.9560e3	99.3487	87.2869
		Err	1.7525e−06	8.2279e−07	4.1241e−06	2.1347e−06	9.0148e−05
	TL	CPU (s)	214.5745	213.9743	958.8988	611.7951	1.9954e+03
		Err	3.1192e−05	8.7876e−06	4.7879e−05	1.0981e−05	5.0408e−04
nd3k	B-A	CPU (s)	99.4634	154.5309	89.6359	28.8885	89.5609
		Err	6.2593e−15	1.4142e−09	6.3277e−15	9.5644e−12	1.0595e−08
	B-T	CPU (s)	36.5428	35.8646	36.5221	32.8895	35.9024
		Err	8.0959e−07	9.2065e−06	1.1654e−06	3.2932e−06	1.8042e−05
	TL	CPU (s)	266.1533	578.7664	1.2525e3	2.7020e3	5.1906e+03
		Err	1.4178e−05	1.2863e−04	9.4521e−05	1.2483e−05	6.3261e−05
nemeth01	B-A	CPU (s)	108.2570	98.0125	99.0669	113.5002	121.9845
		Err	2.8059e−15	4.6788e−15	3.4897e−15	6.1700e−08	1.1520e−12
	B-T	CPU (s)	12.6211	37.5528	75.3023	366.5196	103.1170
		Err	**0.6773**	**0.6329**	3.1143e−06	**0.3916**	**0.5298**
	TL	CPU (s)	426.0148	468.4940	1.2342e+03	2.1748e+03	1.8006e+03
		Err	1.0859e−05	7.5401e−06	4.5847e−06	9.7934e−05	1.3955e−08
net25	B-A	CPU (s)	107.7949	109.8193	95.0551	77.7841	85.9987
		Err	2.0553e−15	3.5862e−15	3.3888e−15	3.0677e−15	2.9593e−15
	B-T	CPU (s)	286.8756	305.1889	272.5080	63.7878	64.9366
		Err	**1.6759**	**1.0740**	**0.9375**	**1.0412**	**0.5282**
	TL	CPU (s)	1.1266e+03	2.2451e+03	945.8087	3.1366e+03	*
		Err	1.4601e−06	1.8892e−05	1.7396e−05	3.1036e−05	*
fv1	B-A	CPU (s)	117.2612	107.3773	117.2678	116.5816	108.4142
		Err	1.8676e−15	2.2884e−15	7.2910e−15	1.0156e−14	2.2138e−14
	B-T	CPU (s)	28.6951	30.1017	27.1641	27.1284	29.0069
		Err	1.1642e−07	2.7156e−07	1.1833e−06	1.8980e−06	3.0522e−06
	TL	CPU (s)	204.0208	441.8608	751.2324	1.1352e+03	1.5996e+03
		Err	1.9973e−06	6.2905e−07	2.2456e−06	1.2979e−06	1.4979e−06

From Tables 2 and 3, we can see that all methods need more CPU time to converge as n increases. B-A and B-T require less computational time than TL to achieve a quite accurate solution. The advantage of the two methods is especially obvious when n is large, which demonstrates that the block Lanczos method is suitable for large-scale problems.

The computational cost of TL increases rapidly as n increases. When $n = 10000$, TL cannot find the optimal solution within three hours. It is observed that, when n becomes larger, B-A achieves a greater accuracy. Part of the reason for this is that we use the relative accuracy.

From Tables 2 and 3, we can see that, when m increases, the computational cost of B-A increases rapidly. Thus, B-A is suitable for the instances with $n \gg m$. It is remarkable that the size of m has less influence on the performance of B-T.

TABLE 7
Numerical results for sparse matrices whose dimensions are greater than 10000.

Matrix	B-A		B-T		TL	
	CPU (s)	Err	CPU (s)	Err	CPU (s)	Err
PGPgiantcompo	124.9722	2.8730e−15	61.5588	1.4213e−06	*	*
rajat06	191.8491	6.8409e−15	34.0700	7.1382e−07	*	*
t2dah	186.7441	4.3078e−15	390.1279	**0.7752**	*	*
ca-HepPh	244.5563	6.0076e−15	56.2770	6.2279e−07	*	*
stokes64s	201.3240	9.1571e−15	41.8400	2.5992e−06	*	*
Pres_Poisson	364.5897	1.1214e−13	64.2966	2.3397e−05	*	*
Dubcova1	186.8153	6.6727e−15	103.8588	9.6312e−07	*	*
ramage02	493.1655	3.7629e−15	103.0569	1.3111e−05	*	*
memplus	550.5148	1.8573e−14	273.9491	1.5471e−06	*	*
ca-AstroPh	1.2944e+03	6.4801e−15	1.4569e+03	**0.7043**	*	*
Si5H12	514.7270	3.8184e−15	218.5472	4.4504e−06	*	*
chipcool0	569.9937	1.2108e−14	318.1621	7.5939e−07	*	*
ns3Da	553.4633	3.9779e−14	220.5074	1.6852e−06	*	*

6.2. Testing for large-scale, dense, nondefinite matrices. In this subsection, we compare B-A and B-T with TL for nonconvex ETR subproblems. The numerical results are shown in Tables 4 and 5.

There are some observations from Tables 4 and 5 to be made here. We can see that B-A and B-T have a better performance than TL. The combining method, B-T, is much faster than TL, the original TOMLAB solver. Surprisingly, the accuracy has also been improved. For $n = 10000$, the ETR subproblem is successfully solved by B-A and B-T. Comparing B-T with B-A, B-T is faster while B-A achieves a higher accuracy.

6.3. Testing for large-scale sparse matrices. For the sparse problems in Table 1, we report in Table 6 the numerical results of seven problems whose dimensions are less than 10000. When $n > 10000$, we give the numerical results for the case when $m_1 = m_2 = 50$ in Table 7, which does not include the results of three problems—flowmeter5, OPF_3754, and cvxqp3—because all these methods failed for those problems.

From Tables 6 and 7, we can see that B-A performs better than B-T and TL, and it can always get a quite accurate solution. B-T is much faster than TL, but fails for four problems (nemeth01, net25, t2dah, and ca-AstroPh). The reason for this is that the TOMLAB solver reaches the maximum iteration number before it can find the solution of the projected subproblem.

7. Conclusion and future work. In this paper, we presented a block Lanczos method for the ETR subproblem. Numerical experiments demonstrate that the block Lanczos method is effective for large-scale problems. In particular, the performance of the B-T method, which combines the block Lanczos method with the TOMLAB solver, is much better than the original TOMLAB solver. Theoretical analysis of error bounds for the optimal value, the optimal solution, and the multipliers is also presented. These theoretical results not only provide the factors to determine the accuracy of the approximation solution, but also can be useful in designing certain stopping criteria for the block Lanczos method.

The encouraging computational results tell us that the block Lanczos method is suitable for large-scale quadratic programming problems with linear constraints. Thus, one of the topics for our future research will be algorithms for large-scale quadratic programming problems with linear and box constraints. When the eigenvalues of B are scattered, the computational cost of the block Lanczos algorithm will increase rapidly as n increases. Which strategy should be employed to overcome this difficulty is also a part of our future work.

Appendix A. Block Lanczos (Ruhe's variant) algorithm.

Algorithm 4 Block Lanczos algorithm (Ruhe's variant).

Input:
M, $\mathcal{C}$.
Output:
W_{k+1}, Q_k, T_k.

$[V_1, R] = qr(\mathcal{C})$ and set $\{t_{ij}\}_{i,j=1}^{km_0} = \mathbf{0}$;
for $j = m_0 : km_0$ **do**
  $r = j - m_0 + 1$;
  $w_1 = Mv_r$;
  for $i = 1 : r - 1$ **do**
    $w_1 = w_1 - t_{ir}v_i$;
  end for
  for $i = r : j$ **do**
    $t_{ir} = \langle w_1, v_i \rangle$;
    if $i \neq r$ **then**
      $t_{ri} = t_{ir}$;
    end if
    $w_1 = w_1 - t_{ir}v_i$;
  end for
  if $j < km_0$ **then**
    $t_{j+1\ r} = \|w_1\|$;
    $v_{j+1} = w_1/t_{j+1\ r}$;
    $t_{r\ j+1} = t_{j+1\ r}$;
  end if
end for
for $j = 2 : m_0$ **do**
  $r = km_0 - m_0 + j$;
  $w_j = Mv_r$;
  for $i = 1 : r - 1$ **do**
    $w_j = w_j - t_{ir}v_i$;
  end for
  for $i = r : km_0$ **do**
    $t_{ir} = \langle w_j, v_i \rangle$;
    if $i \neq r$ **then**
      $t_{ri} = t_{ir}$;
    end if
    $w_j = w_j - t_{ir}v_i$;
  end for
end for
$W_{k+1} = [w_1, w_2, \ldots, w_{m_0}]$; $Q_k = [V_1, V_2, \ldots, V_k]$; $T_k = \{t_{ij}\}_{i,j=1}^{km_0}$.

Acknowledgments. The authors are grateful to the Associate Editor and the referees for their careful reading and very useful comments and suggestions, all of which led to a significant improvement of the paper.

REFERENCES

[1] S. Adachi, S. Iwata, Y. Nakatsukasa, and A. Takeda, *Solving the trust-region subproblem by a generalized eigenvalue problem*, SIAM J. Optim., 27 (2017), pp. 269–291.

[2] W. Ai and S. Zhang, *Strong duality for the CDT subproblem: A necessary and sufficient condition*, SIAM J. Optim., 19 (2009), pp. 1735–1756.

[3] J. Baglama, *Dealing with linear dependence during the iterations of the restarted block Lanczos methods*, Numer. Algorithms, 25 (2000), pp. 23–36.

[4] H. H. Bauschke and J. M. Borwein, *On projection algorithms for solving convex feasibility problems*, SIAM Rev., 38 (1996), pp. 367–426.

[5] H. H. Bauschke, J. M. Borwein, and W. Li, *Strong conical hull intersection property, bounded linear regularity, Jameson's property (G), and error bounds in convex optimization*, Math. Program., 86 (1999), pp. 135–160.

[6] A. Beck and Y. C. Eldar, *Strong duality in nonconvex quadratic optimization with two quadratic constraints*, SIAM J. Optim., 17 (2006), pp. 844–860.

[7] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton University Press, Princeton, NJ, 2009.

[8] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[9] D. Bertsimas, D. B. Brown, and C. Caramanis, *Theory and applications of robust optimization*, SIAM Rev., 53 (2011), pp. 464–501.

[10] D. Bienstock and A. Michalka, *Polynomial solvability of variants of the trust-region subproblem*, in Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, PA, 2014, pp. 380–390.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learning, 3 (2011), pp. 1–122.

[12] S. Burer and K. M. Anstreicher, *Second-order-cone constraints for extended trust-region subproblems*, SIAM J. Optim., 23 (2013), pp. 432–451.

[13] S. Burer and B. Yang, *The trust region subproblem with non-intersecting linear constraints*, Math. Program., 149 (2015), pp. 253–264.

[14] J. V. Burke and S. Deng, *Weak sharp minima revisited, part* II*: Application to linear regularity and error bounds*, Math. Program., 104 (2005), pp. 235–261.

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

[16] X. Cai, D. Han, and X. Yuan, *The direct extension of ADMM for three-block separable convex minimization models is convergent when one function is strongly convex*, Comput. Optim. Appl., 66 (2017), pp. 39–73.

[17] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, PA, 2000.

[18] T. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), pp. 1–25.

[19] J. B. Erway and P. E. Gill, *A subspace minimization method for the trust-region step*, SIAM J. Optim., 20 (2010), pp. 1439–1461.

[20] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.

[21] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, PA, 1997.

[22] K. Guo, D. Han, Z. W. Wang, and T. Wu, *Convergence of ADMM for multi-block nonconvex separable optimization models*, Front. Math. China, 12 (2017), pp. 1139–1162.

[23] N. Ho-Nguyen and F. Kilinc-Karzan, *A second-order cone based approach for solving the trust-region subproblem and its variants*, SIAM J. Optim., 27 (2017), pp. 1485–1512.

[24] K. Holmstrom, A. O. Goran, and M. M. Edvall, *User's Guide for TOMLAB/SOL*1, TOMLAB Optimization Inc., San Diego, CA, 2008.

[25] V. Jeyakumar and G. Y. Li, *Strong duality in robust convex programming: Complete characterizations*, SIAM J. Optim., 20 (2010), pp. 3384–3407.

[26] V. Jeyakumar and G. Y. Li, *Trust-region problems with linear inequality constraints: Exact SDP relaxation, global optimality and robust optimization*, Math. Program., 147 (2014), pp. 171–206.

[27] S. Lu and S. M. Robinson, *Normal fans of polyhedral convex sets: Structures and connections*, Set-Valued Anal., 16 (2008), pp. 281–305.

[28] K. F. Ng and W. H. Yang, *Regularities and their relations to error bounds*, Math. Program., 99 (2004), pp. 521–538.

[29] J. Nocedal and S. J. Writht, *Numerical Optimization*, 2nd ed., Springer, Berlin, 2006.

[30] T. K. Pong and H. Wolkowicz, *The generalized trust region subproblem*, Comput. Optim. Appl., 58 (2014), pp. 273–322.

[31] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with*

applications to large scale minimization, Math. Program., 77 (1997), pp. 273–299.
[32] A. Ruhe, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Math. Comp., 33 (1979), pp. 680–687.
[33] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.
[34] M. Salahi, A. Taati, and H. Wolkowicz, *Local nonglobal minima for solving large-scale extended trust-region subproblems*, Comput. Optim. Appl., 66 (2017), pp. 223–244.
[35] J. F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Math. Oper. Res., 28 (2003), pp. 246–267.
[36] Z. Wang and Y. Yuan, *A subspace implementation of quasi-Newton trust region methods for unconstrained optimization*, Numer. Math., 104 (2006), pp. 241–269.
[37] Y. Ye and S. Zhang, *New results on quadratic minimization*, SIAM J. Optim., 14 (2003), pp. 245–267.
[38] J. Yuan, M. Wang, W. Ai, and T. Shuai, *A necessary and sufficient condition of convexity for SOC reformulation of trust-region subproblem with two intersecting cuts*, Sci. China Math., 59 (2016), pp. 1127–1140.
[39] Y. Yuan, *A review on subspace methods for nonlinear optimization*, in Proceedings of the International Congress of Mathematicians 2014, Vol. IV, Kyung Moon, Seoul, 2014, pp. 807–827; available at http://www.icm2014.org/.
[40] Y. Yuan, *Recent advances in trust region algorithms*, Math. Program., 151 (2015), pp. 249–281.
[41] L. Zhang, C. Shen, and R. Li, *On the generalized Lanczos trust-region method*, SIAM J. Optim., 27 (2017), pp. 2110–2142.
[42] X. Y. Zheng and K. F. Ng, *Metric regularity and constraint qualifications for convex inequalities on Banach spaces*, SIAM J. Optim., 14 (2003), pp. 757–772.