CONVERGENCE RATE OF AN OPTIMIZATION ALGORITHM FOR MINIMIZING QUADRATIC FUNCTIONS WITH SEPARABLE
CONVEX CONSTRAINTS

RADEK KUČERA†

Abstract. A new active set algorithm for minimizing quadratic functions with separable convex constraints is proposed by combining the conjugate gradient method with the projected gradient. It generalizes recently developed algorithms of quadratic programming constrained by simple bounds. A linear convergence rate in terms of the Hessian spectral condition number is proven. Numerical experiments, including the frictional three-dimensional (3D) contact problems of linear elasticity, illustrate the computational performance.

Key words. quadratic function, separable convex constraints, active set, conjugate gradient method, projected gradient, convergence rate

AMS subject classifications. 65K05, 90C25

DOI. 10.1137/060670456

1. Introduction. We shall be concerned with solving

\[ \min_{x \in \Omega} f(x), \]

where \( f(x) = \frac{1}{2} x^\top A x - x^\top b, \) \( A \in \mathbb{R}^{n \times n} \) is symmetric positive definite, \( b \in \mathbb{R}^n \), \( \Omega = \Omega_1 \times \cdots \times \Omega_m \), and \( \Omega_i = \{ x_i \in \mathbb{R}^{n_i} : f_i(x_i) \leq 0 \} \) are defined by continuously differentiable convex functions \( f_i : \mathbb{R}^{n_i} \to \mathbb{R} \) so that \( n_i \geq 1, \sum_{i=1}^m n_i = n \). Let us note that the feasible set \( \Omega \) is separable in the sense that each part \( x_i \) of \( x = (x_1^\top, \ldots, x_m^\top) \) is subject to one constraint \( x_i \in \Omega_i \).

This problem includes several independently investigated subproblems originating, for instance, in duality-based methods for the solution of contact problems of linear elasticity:

- If \( n_i = 1 \) and \( f_i(x_i) \equiv l_i - x_i \) with \( l_i \) given, we obtain the simple bound

\[ l_i \leq x_i \]

arising from two-dimensional contact problems [4].

- If \( n_i = 2 \), \( x_i = (x_{2i-1}, x_{2i}) \) \(^\top\) and \( f_i(x_i) \equiv x_{2i-1}^2 + x_{2i}^2 - r_i^2 \) with \( r_i \) given, we arrive at

\[ x_{2i-1}^2 + x_{2i}^2 \leq r_i^2 \]

that can be interpreted as the circular constraint. A source of such constraints is an isotropic friction law for three-dimensional (3D) contact problems [8].

- If an anisotropic friction law is considered and if a finite element tearing and interconnecting (FETI) domain decomposition method is used for solving 3D...
contact problems [11], then (1.3) is replaced by the ellipsoidal constraint,

\[(x_i - s_i)^T F_i (x_i - s_i) \leq r_i^2,\]

with \(F_i \in \mathbb{R}^{2 \times 2}\) positive definite and \(s_i = (s_{2i-1}, s_{2i})^T\) given.

– Finally let us note that the unconstrained case may be described by \(f_i(x_i) = -1\).

Through the whole paper, we shall have in mind large-scale problems in which the Hessian matrix \(A\) is not formed explicitly. In this case, an iterative method is a suitable tool for solving (1.1) since it is the only action of \(A\) which is needed. A class of efficient algorithms appropriate for our research is based on the active set strategy that dates back at least to Polyak [13]. His algorithm solves quadratic programming problems constrained by simple bounds (1.2) using a restarted conjugate gradient method. After each start, a subset of variables is fixed at bounds (the active set) and the conjugate gradient method minimizes \(f\) with respect to remaining variables. The “inner” minimization is terminated if either the minimum is reached or an infeasible iteration is generated. In the first case, some of the fixed variables are released while, in the second case, new variables are added to the active set. In both cases, the value of \(f\) decreases so that the active set can never reappear, and therefore the algorithm converges in a finite number of steps.

As the Polyak algorithm suffers from several drawbacks, it was modified in order to exclude doubts about its efficiency; see discussions in [1, 5]. Here, we mention two improvements relevant for our more general constraints. Firstly, the exact solution of auxiliary “inner” problems can be replaced by an inexact one. We shall use a theoretically supported strategy of an adaptive precision control presented by Friedlander and Martínez [6] and by Dostál [2]. The basic idea is to control the “inner” precision by a ratio of the norms of a violation of the Karush–Kuhn–Tucker (KKT) conditions at fixed and free variables. Secondly, the qualitative progress has been achieved by Dostál and Schöbel [5] using the projected gradient for expanding the active set. It permits rapid changes in the active set without the necessity to perform computationally expensive steps (e.g., backtracking). Moreover, the algorithm has a linear convergence rate in terms of the spectral condition number of the Hessian matrix \(A\).

The scheme of the later algorithm was successfully extended for solving 3D contact problems with an isotropic friction in [11]. As the constraints are circular (1.3), the algorithm does search not only for the active set corresponding to the solution, but also for the positions of the pairs \((x_{2i-1}, x_{2i})^T\) lying on the boundaries of the circles \(x_{2i-1}^2 + x_{2i}^2 = r_i^2\). It is easily seen that the finite terminating property cannot be expected in such cases, and therefore the convergence was proven in [10] by different arguments. On the other hand, it is relatively surprising that a linear convergence rate may be derived like for the simple bound case. This proof is the main goal of the paper.

Let us briefly outline the structure of the paper. After introducing notations in section 2, we prove that the KKT optimality conditions are equivalent to the zero projected gradient. Our algorithm for solving (1.1) is proposed in section 3 in a form suitable for theoretical analysis. Section 4 summarizes auxiliary statements on the projected gradient, while section 5 gives the main result of the paper concerning a linear convergence rate. A practical implementation of the algorithm for simple bounds (1.2) and circular constraints (1.3) is discussed in section 6, and finally, section 7 presents the results of numerical experiments.

2. Preliminaries and notations. We shall always assume that the feasible set \(\Omega\) in (1.1) is nonempty. As \(f\) is the strictly convex function and \(\Omega\) is the convex set,
the existence of a unique solution to (1.1) is guaranteed. We shall denote it by \( x^* \). It is well known that \( x^* \) is fully determined by the KKT conditions [12]. Before giving their appropriate form, we shall introduce notations.

Recall that a continuously differentiable function \( F : \Omega \to \mathbb{R} \) is convex iff

\[
F(y) - F(x) \geq (y - x)^\top \nabla F(x) \quad \forall x, y \in \Omega,
\]

where \( \nabla F \) denotes the gradient of \( F \). The gradient of the objective function \( f \) at \( x \) shall be denoted by

\[
g = g(x) = Ax - b.
\]

Let \( \mathcal{M} \) denote the set of indices so that

\[
\mathcal{M} = \{1, \ldots, m\}.
\]

We shall use the following convention: if \( x \in \mathbb{R}^n \) is a vector, then \( x_i \in \mathbb{R} \) is its \( i \)th entry, \( 1 \leq i \leq n \), and \( x_i \in \mathbb{R}^n \) is its \( i \)th segment, \( i \in \mathcal{M} \), so that \( x = (x_1^\top, \ldots, x_m^\top)^\top \). We shall denote by \( \|x_i\| \) the Euclidean norm of \( x_i \).

In order to exclude pathological situations, we shall assume without loss of generality that \( f_i \) are not identically zero in \( \Omega_i \). In that case \( \Omega_i \) has a nonempty interior, \( \text{int} \Omega_i = \{x_i \in \mathbb{R}^{n_i} : f_i(x_i) < 0\} \), and a possibly nonempty boundary \( \partial \Omega_i = \{x_i \in \mathbb{R}^{n_i} : f_i(x_i) = 0\} \). The convexity of \( f_i \) implies that if \( \partial \Omega_i \neq \emptyset \) and \( x_i \in \partial \Omega_i \), then \( \nabla f_i(x_i) \) is the outward normal vector to \( \partial \Omega_i \) at \( x_i \); see Figure 2.1.a.

It is well known that the solution \( x^* \) to (1.1) is characterized by the existence of Lagrange multipliers \( \lambda_i^* \), \( i \in \mathcal{M} \), such that [12]

\[
g_i^* + \lambda_i^* \nabla f_i(x_i^*) = 0, \quad f_i(x_i^*) \leq 0, \quad \lambda_i^* f_i(x_i^*) = 0, \quad i \in \mathcal{M}, \]

where \( g_i^* \) denotes the \( i \)th segment of \( g^* = g(x^*) \). After eliminating \( \lambda_i^* \), we obtain the following theorem.

**Theorem 2.1.** *The vector \( x^* \in \Omega \) is the solution to (1.1) iff for \( i \in \mathcal{M} \):

\[
f_i(x_i^*) < 0 \quad \text{implies} \quad g_i^* = 0,
\]

\[
f_i(x_i^*) = 0 \quad \text{implies} \quad g_i^* + \frac{\|g_i^*\|}{\|\nabla f_i(x_i^*)\|} \nabla f_i(x_i^*) = 0.
\]

Conditions (2.2) and (2.3) are called the *inner KKT conditions* and the *boundary KKT conditions*, respectively.

As the feasible set \( \Omega \) is separable, the projection \( P_\Omega : \mathbb{R}^n \to \Omega \) can be put together by the projections \( P_{\Omega_i} : \mathbb{R}^{n_i} \to \Omega_i, i \in \mathcal{M} \). Thus we define \( P_\Omega(x) \) for any \( x \in \mathbb{R}^n \) by

\[
P_\Omega(x) = \begin{pmatrix} P_{\Omega_1}(x_1) \\ \vdots \\ P_{\Omega_m}(x_m) \end{pmatrix},
\]

where \( P_{\Omega_i}(x_i) \) are defined by

\[
P_{\Omega_i}(x_i) = \begin{cases} x_i & \text{for } x_i \in \Omega_i, \\ z_i & \text{for } x_i \notin \Omega_i, \end{cases}
\]
with (see Figure 2.1.b)

\begin{align}
(2.5) & \quad f_i(z_i) = 0, \\
(2.6) & \quad z_i + \frac{\|x_i - z_i\|}{\|\nabla f_i(z_i)\|} \nabla f_i(z_i) = x_i.
\end{align}

Let us note that $z_i$ is well defined by (2.5) and (2.6) because $\Omega_i$ is convex. The equality (2.6) yields orthogonality of $P_{\Omega_i}$ and, consequently, of $P_{\Omega}$ that is equivalent to the variational inequality

$$(x - P_{\Omega}(x))^T(y - P_{\Omega}(x)) \leq 0 \quad \forall x \in \mathbb{R}^n \forall y \in \Omega.$$ 

Let us define the projected gradient $\bar{g} = \bar{g}(x)$ at $x \in \Omega$ for fixed $\bar{a} > 0$ by

\begin{equation}
(2.7) \quad \bar{g}(x) = \frac{1}{\bar{a}} (x - P_{\Omega}(x - \bar{a}g(x))).
\end{equation}

This definition enables us to describe the projection of the gradient step without $P_{\Omega}$ so that

$$P_{\Omega}(x - \bar{a}g(x)) = x - \bar{a}\bar{g}(x);$$

see Figure 2.2. In the next theorem, we prove that the zero project gradient represents an alternative optimality criterion to KKT conditions.

**Theorem 2.2.** The vector $x^* \in \Omega$ solves (1.1) iff $\bar{g}(x^*) = 0$. 
Proof. The equality $\overline{g}(x^*) = 0$ is equivalent to
\[(2.8)\]
$$ x_i^* = P_{\Omega_i}(x_i^* - \overline{\alpha}g_i^*), \quad i \in \mathcal{M}. $$
We shall prove that the KKT conditions (2.2) and (2.3) are equivalently satisfied. We distinguish two cases. (i) Let $f_i(x_i^*) < 0$. Then $x_i^* \in \text{int} \Omega_i$, which is equivalent by (2.8) to $x_i^* - \overline{\alpha}g_i^* \in \text{int} \Omega_i$. Therefore
$$ x_i^* = P_{\Omega_i}(x_i^* - \overline{\alpha}g_i^*) = x_i^* - \overline{\alpha}g_i^* $$
so that (2.2) holds. (ii) Let $f_i(x_i^*) = 0$. Using (2.6) with $x_i$ replaced by $x_i^* - \overline{\alpha}g_i^*$ and with $z_i$ replaced by $x_i^*$, we obtain after simple manipulation
$$ g_i^* + \frac{\|g_i^*\|}{\|\nabla f_i(x_i^*)\|} \nabla f_i(x_i^*) = 0 $$
so that (2.3) holds. \[\square\]

We shall decompose $\mathcal{M}$ at $x \in \Omega$ on the free set $\mathcal{F}(x)$ and the active set $\mathcal{A}(x)$ as
\[(2.9)\]
$$ \mathcal{F}(x) = \{i \in \mathcal{M} : f_i(x_i) < 0\}, $$
\[(2.10)\]
$$ \mathcal{A}(x) = \{i \in \mathcal{M} : f_i(x_i) = 0\}. $$

Analogously, we can decompose $\overline{g}$ on the projected free gradient $\overline{\varphi} = \overline{\varphi}(x)$ and the projected boundary gradient $\overline{\beta} = \overline{\beta}(x)$ so that
\[(2.11)\]
$$ \overline{\varphi}_i = \overline{g}_i \quad \text{for} \quad i \in \mathcal{F}(x), \quad \overline{\beta}_i = 0 \quad \text{for} \quad i \in \mathcal{A}(x), $$
$$ \overline{\varphi}_i = \overline{g}_i \quad \text{for} \quad i \in \mathcal{F}(x), \quad \overline{\beta}_i = \overline{g}_i \quad \text{for} \quad i \in \mathcal{A}(x). $$

Thus, the inner KKT conditions (2.2) are satisfied iff $\overline{\varphi}(x) = 0$, and the boundary KKT conditions (2.3) are satisfied iff $\overline{\beta}(x) = 0$. Moreover, we define the free gradient $\varphi = \varphi(x)$ so that
\[(3.1)\]
$$ x^{k+1} = x^k - \overline{\alpha} \overline{\varphi}(x^k); $$

3. Algorithm. In this section we present our algorithm for solving (1.1) in a form convenient for the analysis, while technical details are postponed to section 6. The algorithm exploits a given constant $\Gamma > 0$ to decide on interrupting conjugate gradient iterations and a fixed steplength $\overline{\alpha} \in (0, \|A\|^{-1}]$ defining the projected gradient.

We shall combine three steps to generate a sequence of iterates $\{x^k\}$ that approximates the solution to (1.1):
- the expansion step, which may add indices to the active set, is defined by
the proportioning step, which may release indices from the active set, reads as
\begin{equation}
  x^{k+1} = x^k - \alpha \tilde{\beta}(x^k);
\end{equation}

the conjugate gradient step is given by
\begin{equation}
  x^{k+1} = x^k - \alpha_{cg} p^k, \quad \alpha_{cg} = \frac{g(x^k)^\top p^k}{(p^k)^\top Ap^k},
\end{equation}

where the conjugate gradient directions \( p^k \) are constructed recurrently [7].

The conjugate gradient steps are used to carry out a minimization of the objective function \( f \) efficiently on the interior of the face
\[
W_{\mathcal{A}(x^*)} = \{ x \in \Omega : x_i = x_i^* \text{ for } i \in \mathcal{A}(x^*) \},
\]
where \( x^* \) is determined by the expansion step or by the proportioning step. It requires that the parts of \( p^k \) corresponding to the indices of \( \mathcal{A}(x^*) \) vanish, i.e., \( p_i^k = 0 \) for \( i \in \mathcal{A}(x^*) \). We can easily fulfill this requirement by adapting the classical recurrence generating \( p^k \). We start (or restart) from \( p^s = \varphi(x^*) \) and use
\begin{equation}
  p^k = \varphi(x^k) - \gamma^k p^{k-1}, \quad \gamma^k = \frac{\varphi(x^k)^\top Ap^{k-1}}{(p^{k-1})^\top Ap^{k-1}}, \quad k > s.
\end{equation}
The formulae (3.4) is used while the sequence of the conjugate gradient steps is unbroken. After changing the active set, we must always restart.

Later on, we shall need the following lemma.

**Lemma 3.1.** Let \( x^{k+1} \) be generated by the conjugate gradient step. Then
\[
f(x^{k+1}) \leq f(x^k - \alpha \varphi(x^k)) \quad \forall \alpha \in \mathbb{R}.
\]

**Proof.** It is a well-known property of the conjugate gradient method [7] that
\begin{equation}
  f(x^{k+1}) = \min_{x \in x^* + \text{Span}\{p^s, \ldots, p^k\}} f(x).
\end{equation}
As (3.4) implies \( \varphi(x^k) \in \text{Span}\{p^s, \ldots, p^k\} \) and \( x^k \in x^* + \text{Span}\{p^s, \ldots, p^{k-1}\} \), we obtain \( x^k - \alpha \varphi(x^k) \in x^* + \text{Span}\{p^s, \ldots, p^k\} \). The lemma follows using (3.5).

The last ingredient of our algorithm is the releasing criterion:
\begin{equation}
  \tilde{\beta}(x^k)^\top g(x^k) \leq \Gamma^2 \tilde{\varphi}(x^k)^\top g(x^k).
\end{equation}
If this inequality holds, we call the iterate \( x^k \) strictly proportional. The criterion (3.6) is used to decide which of the steps will be performed.

**Algorithm 3.1.** Let \( x^0 \in \Omega, \quad \Gamma > 0, \quad \text{and} \quad \tilde{\alpha} \in (0, \|A\|^{-1}] \) be given. For \( k \geq 0 \) and \( x^k \) known, choose \( x^{k+1} \) by the following rules:
(i) If \( \tilde{g}(x^k) = 0 \), set \( x^{k+1} = x^k \).
(ii) If \( x^k \) is strictly proportional and \( \tilde{g}(x^k) \neq 0 \), try to generate \( x^{k+1} \) by the conjugate gradient step. If \( x^{k+1} \in \Omega \) and if it does not change the active set, then accept it, else generate \( x^{k+1} \) by the expansion step.
(iii) If \( x^k \) is not strictly proportional, then define \( x^{k+1} \) by the proportioning step.
4. Properties of the projected gradient. In this section we summarize results necessary for the next analysis.

**Lemma 4.1.** It holds that

\[(4.1) \quad \|\tilde{\varphi}(x)\|^2 \leq \varphi(x)^T g(x),\]
\[(4.2) \quad \|\tilde{\beta}(x)\|^2 \leq \beta(x)^T g(x),\]
\[(4.3) \quad \varphi(x)^T g(x) \leq \varphi(x)^T g(x).\]

**Proof.** The definition of the projected gradient (2.7) implies that

\[P_{\Omega_i}(x_i - \tilde{\alpha}g_i) = x_i - \tilde{\alpha}g_i.\]

As \(P_{\Omega_i}\) is orthogonal, we obtain

\[0 \geq (x_i - \tilde{\alpha}g_i - P_{\Omega_i}(x_i - \tilde{\alpha}g_i))^T (x_i - P_{\Omega_i}(x_i - \tilde{\alpha}g_i)) = \tilde{\alpha}^2 (\tilde{g}_i - g_i)^T \tilde{g}_i\]

so that

\[(4.4) \quad \|\tilde{g}_i\|^2 \leq \tilde{g}_i^T g_i.\]

Summing (4.4) over the indices of \(F(x)\) or \(A(x)\), we obtain (4.1) or (4.2), respectively. Using the well-known Cauchy inequality in (4.4), we get \(\|\tilde{g}_i\| \leq \|g_i\|\), and therefore

\[\varphi(x)^T g(x) = \sum_{i \in \mathcal{F}(x)} \tilde{g}_i^T g_i \leq \sum_{i \in \mathcal{F}(x)} \|\tilde{g}_i\| \|g_i\| \leq \sum_{i \in \mathcal{F}(x)} g_i^T g_i = \varphi(x)^T g(x). \] \(\square\)

**Lemma 4.2.** Let \(d \in \mathbb{R}^n, d \neq 0\), and \(x \in \Omega\). Then

\[(4.5) \quad f(x) - f(x - \alpha d) \leq \alpha d^T g(x) \quad \forall \alpha \in \mathbb{R}.\]

Moreover, let \(d^T g(x) \geq 0\), and \(\alpha_d = d^T g(x)/d^T Ad\). Then

\[(4.6) \quad f(x) - f(x - \alpha d) \geq \frac{1}{2} \alpha d^T g(x) \quad \forall \alpha \in [0, \alpha_d].\]

**Proof.** The assertion (4.5) is equivalent to the convexity of \(f\) (compare with (2.1)). For \(\alpha \in [0, \alpha_d]\), we derive

\[f(x - \alpha d) = f(x) - \alpha d^T g(x) + \frac{1}{2} \alpha^2 d^T Ad \leq f(x) - \alpha d^T g(x) + \frac{1}{2} \alpha \alpha_d d^T Ad \]
\[= f(x) - \frac{1}{2} \alpha d^T g(x). \] \(\square\)

**Corollary 4.3.** If \(d\) is replaced in (4.6) by \(\varphi\), \(\tilde{\varphi}\), and \(\tilde{\beta}\), then the corresponding three inequalities hold for all \(\alpha \in [0, \|A\|^{-1}]\).

**Proof.** For \(d = \varphi(x)\), we have

\[\alpha_d = \frac{\varphi(x)^T g(x)}{\varphi(x)^T A \varphi(x)} \geq \frac{\varphi(x)^T \varphi(x)}{\|A\| \|\varphi(x)\|^2} = \|A\|^{-1}.\]

The same follows using Lemma 4.1 for \(d = \tilde{\varphi}(x)\) and \(d = \tilde{\beta}(x)\). \(\square\)
Lemma 4.4. Let \( x \in \Omega \). Then
\[
f(x) - f(x - \alpha \tilde{g}(x)) \leq \alpha (\tilde{\varphi}(x)^\top g(x) + \tilde{\beta}(x)^\top g(x)) \quad \forall \alpha \in \mathbb{R}.
\]

**Proof.** Using (4.5), we obtain
\[
f(x) - f(x - \alpha \tilde{g}(x)) \leq \alpha \tilde{g}(x)^\top g(x) = \alpha \left( \sum_{i \in \mathcal{F}(x)} \tilde{g}_i^\top g_i + \sum_{i \in \mathcal{A}(x)} \tilde{g}_i^\top g_i \right),
\]
and the definitions (2.9) and (2.10) complete the proof. \( \square \)

Lemma 4.5. Let \( x^* \in \Omega \) denote the solution to (1.1), \( \lambda_1 \) denote the smallest eigenvalue of \( A \), and \( x \in \Omega \). Then
\[
f(x) - f(x - \tilde{\alpha} \tilde{g}(x)) \geq \tilde{\alpha} \lambda_1 (f(x) - f(x^*)) \quad \forall \tilde{\alpha} \in [0, \|A\|^{-1}].
\]

**Proof.** For \( \tilde{\alpha} \in (0, \|A\|^{-1}] \), we define
\[
F(y) = \tilde{\alpha} f(y) + \frac{1}{2} (y - x)^\top (I - \tilde{\alpha} A)(y - x).
\]
Notice that \( I - \tilde{\alpha} A \) is positive semidefinite. Therefore
\[
F(y) \geq \tilde{\alpha} f(y),
\]
(4.7) \[ \nabla F(y) = \tilde{\alpha} g(y) + (I - \tilde{\alpha} A)(y - x) = y - x + \tilde{\alpha} g(x), \]
and \( F(x) = \tilde{\alpha} f(x), \nabla F(x) = \tilde{\alpha} g(x) \). Let us note that the Hessian of \( F \) is the identity \( I \).

Let \( y \in \Omega \) be arbitrary. The convexity of \( F \), (4.7), and the orthogonality of \( P_\Omega \) yields
\[
F(y) - F(P_\Omega(x - \tilde{\alpha} g(x))) \geq (y - P_\Omega(x - \tilde{\alpha} g(x)))^\top \nabla F(P_\Omega(x - \tilde{\alpha} g(x)))
\]
\[
= -(y - P_\Omega(x - \tilde{\alpha} g(x)))^\top (x - \tilde{\alpha} g(x) - P_\Omega(x - \tilde{\alpha} g(x)))
\]
\[
\geq 0
\]
so that
\[
F(y) \geq F(P_\Omega(x - \tilde{\alpha} g(x))) \quad \forall y \in \Omega.
\]

Using \( \tilde{\alpha} \lambda_1 \leq \|A\|^{-1} \lambda_1 \leq 1 \) and denoting \( d = x^* - x \), we derive
\[
f(x - \tilde{\alpha} g(x)) = f(P_\Omega(x - \tilde{\alpha} g(x)))
\]
\[
\leq \tilde{\alpha}^{-1} F(P_\Omega(x - \tilde{\alpha} g(x))) \leq \tilde{\alpha}^{-1} \min_{y \in \Omega} F(y) \leq \tilde{\alpha}^{-1} \min_{t \in [0,1]} F(x + td)
\]
\[
\leq \tilde{\alpha}^{-1} F(x + \tilde{\alpha} \lambda_1 d) = f(x) + \tilde{\alpha} \lambda_1 d^\top g(x) + \frac{1}{2} \tilde{\alpha} \lambda_1^2 d^\top d
\]
\[
\leq (1 - \tilde{\alpha} \lambda_1) f(x) + \tilde{\alpha} \lambda_1 f(x) + \tilde{\alpha} \lambda_1 d^\top g(x) + \frac{1}{2} \tilde{\alpha} \lambda_1 d^\top Ad
\]
\[
= (1 - \tilde{\alpha} \lambda_1) f(x) + \tilde{\alpha} \lambda_1 f(x^*).
\]

Lemma 4.4 and Lemma 4.5 immediately imply the following result.

**Corollary 4.6.** Let \( x^* \in \Omega \) denote the solution to (1.1), \( \lambda_1 \) denote the smallest eigenvalue of \( A \), and \( x \in \Omega \). Then
\[
\lambda_1 (f(x) - f(x^*)) \leq \tilde{\varphi}(x)^\top g(x) + \tilde{\beta}(x)^\top g(x)
\]
for all \( \tilde{\alpha} \in (0, \|A\|^{-1}] \) defining \( \tilde{\varphi} \) and \( \tilde{\beta} \).
5. Rate of convergence.

**Theorem 5.1.** Let \( x^0 \in \Omega, \Gamma > 0, \) and \( \tilde{\alpha} \in (0, \|A\|^{-1}] \) be given. Let \( x^* \in \Omega \) denote the solution to (1.1), \( \lambda_1 \) denote the smallest eigenvalue of \( A, \) and \( \tilde{\Gamma} = \max\{1, \Gamma^{-1}\}. \) Let \( \{x^k\} \) be the sequence generated by Algorithm 3.1. Then

\[
f(x^{k+1}) - f(x^*) \leq \eta (f(x^k) - f(x^*)) ,
\]

where

\[
\eta = 1 - \frac{\tilde{\alpha} \lambda_1}{2 + 2 \tilde{\Gamma}^2} < 1.
\]

The error in the A-energy norm is bounded by

\[
\|x^k - x^*\|_A^2 \leq 2 \eta^k (f(x^0) - f(x^*)) .
\]

**Proof.** We shall estimate separately all three possible steps of Algorithm 3.1. As \( \tilde{\alpha} \in (0, \|A\|^{-1}] \), we can use the bound (4.6) for each of the steps due to Corollary 4.3.

Let us first assume that \( x^{k+1} \) is generated by the expansion step (3.1). Using (4.6), we obtain

\[
f(x^{k+1}) = f(x^k - \tilde{\alpha} \bar{\varphi}(x^k)) \leq f(x^k) - \frac{1}{2} \tilde{\alpha} \bar{\varphi}(x^k)^\top g(x^k). \tag{5.4}
\]

If \( x^{k+1} \) is generated by the conjugate gradient step (3.3), we use Lemma 3.1, (4.6), and (4.3) so that

\[
f(x^{k+1}) \leq f(x^k - \tilde{\alpha} \bar{\varphi}(x^k)) \leq f(x^k) - \frac{1}{2} \tilde{\alpha} \bar{\varphi}(x^k)^\top g(x^k). \tag{5.5}
\]

Comparing (5.4) and (5.5), we may observe that we have the same estimates for both of the expansion and conjugate gradient steps. These steps are taken only when \( x^k \) is strictly proportional, i.e., when

\[
\tilde{\beta}(x^k)^\top g(x^k) \leq \tilde{\Gamma}^2 \bar{\varphi}(x^k)^\top g(x^k).
\]

After using Corollary 4.6, we get

\[
\bar{\varphi}(x^k)^\top g(x^k) \geq \frac{\lambda_1}{1 + \tilde{\Gamma}^2} (f(x^k) - f(x^*)). \tag{5.6}
\]

The estimates (5.4), (5.5) combined with (5.6) imply that

\[
f(x^{k+1}) - f(x^*) \leq f(x^k) - f(x^*) - \frac{1}{2} \tilde{\alpha} \bar{\varphi}(x^k)^\top g(x^k) \leq \left( 1 - \frac{\tilde{\alpha} \lambda_1}{2 + 2 \tilde{\Gamma}^2} \right) (f(x^k) - f(x^*)). \tag{5.7}
\]

Let us finally assume that \( x^{k+1} \) is generated by the proportioning step (3.2). Again using (4.6), we obtain

\[
f(x^{k+1}) = f(x^k - \alpha \tilde{\beta}(x^k)) \leq f(x^k) - \frac{1}{2} \alpha \tilde{\beta}(x^k)^\top g(x^k). \tag{5.8}
\]
As the proportioning step is taken when 
\[ \tilde{\beta}(x^k)^\top g(x^k) > \Gamma^2 \tilde{\varphi}(x^k)^\top g(x^k), \]
Corollary 4.6 yields
\[ (5.9) \quad \tilde{\beta}(x^k)^\top g(x^k) \geq \frac{\lambda_1}{1 + \Gamma^{-2}} (f(x^k) - f(x^*)). \]
The estimate (5.8) combined with (5.9) implies that
\[ (5.10) \quad f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\tilde{\alpha} \lambda_1}{2 + 2\Gamma^{-2}}\right) (f(x^k) - f(x^*)). \]
Comparing the inequalities (5.10) and (5.7), and taking into account that by definition \( \Gamma \leq \hat{\Gamma} \) and \( \Gamma^{-1} \leq \hat{\Gamma} \), we can see that (5.1) holds.

In order to prove the error bound (5.3), we use (5.1) and the fact that the solution \( x^* \) to (1.1) is characterized by the variational inequality \( (x - x^*)^\top g(x^*) \geq 0 \) for all \( x \in \Omega \). We obtain
\[
\|x^k - x^*\|^2_A = 2(f(x^k) - f(x^*) - (x^k - x^*)^\top g(x^*))
\leq 2(f(x^k) - f(x^*)) \leq 2\eta_k (f(x^0) - f(x^*)). \]

Theorem 5.1 yields the best estimate for \( \Gamma = \hat{\Gamma} = 1 \) and \( \tilde{\alpha} = \|A\|^{-1} \) when
\[ \eta = 1 - \frac{1}{4} \kappa(A)^{-1}. \]
We shall see that this result is in agreement with numerical experiments.

6. Implementation. We shall give the details of the implementation of Algorithm 3.1. Firstly, we describe a general scheme of the algorithm independently on the type of constraints. Then we show how to compute projections on the feasible set in our numerical experiments.

6.1. A general algorithmic scheme. The presented implementation differs from Algorithm 3.1 in that it exploits the current conjugate gradient direction to generate an intermediate iteration \( x^{k+1/2} \) before the expansion step that generates \( x^{k+1} \) from \( x^{k+1/2} \). Such modification does not require any additional matrix-vector multiplication, and the estimate (5.1) remains valid [5].

We describe the algorithm in an easily understandable variant of the Matlab language, in which we do not distinguish a generation of variables by indices unless it is convenient for further references.

It should be noted that the presented implementation is not influenced by the type of constraints defining the feasible set \( \Omega \). The relation to constraints is hidden in the components \( \tilde{\beta}, \tilde{\varphi} \) of the projected gradient \( \tilde{g} \), and in the feasible steplength \( \alpha_f \). The projected gradient can be directly evaluated by its definition (2.7) using \( P_\Omega \).
Algorithm 6.1. Let \( x^0 \in \Omega, \Gamma > 0, \tilde{\alpha} \in (0, \|A\|^{-1}], \) and \( \epsilon > 0 \) be given.

Set \( k = 0, g = Ax^0 - b, p = \varphi(x^0). \) % Initialization.

while \( \|g(x^k)\| > \epsilon \)

if \( \beta(x^k)^\top g(x^k) \leq \Gamma^2 \bar{\varphi}(x^k)^\top g(x^k) \)

\( \alpha_{cg} = g^\top p/p^\top Ap \) % Conjugate gradient steplength.

\( \alpha_f = \max\{\alpha : x^k - \alpha p \in \Omega\} \) % Feasible steplength.

if \( \alpha_{cg} < \alpha_f \)

\( x^{k+1} = x^k - \alpha_{cg} p, g = g - \alpha_{cg} Ap, \gamma = \varphi(x^{k+1})^\top Ap/p^\top Ap, p = \varphi(x^{k+1}) - \gamma p \) % Conjugate gradient steplength.

else

\( x^{k+1/2} = x^k - \alpha_f p \)

\( x^{k+1} = x^{k+1/2} - \tilde{\alpha} \bar{\varphi}(x^{k+1/2}), g = Ax^{k+1} - b, p = \varphi(x^{k+1}) \) % Expansion step.

endif

else

\( x^{k+1} = x^k - \tilde{\alpha} \tilde{\varphi}(x^k), g = Ax^{k+1} - b, p = \varphi(x^{k+1}) \) % Proportioning step.

endif

\( k = k + 1 \)

endwhile

\( x = x^k. \) % Return step.

6.2. Projections and the feasible steplength. In the previous sections, we have exploited the implicit definition of the projections given by (2.5) and (2.6). Now we shall show how to compute \( P_{\Omega_i} \) for simple bounds (1.2) and circular constraints (1.3).

Let us consider the following variant of the problem (1.1):

\[
\begin{align*}
\text{minimize} & \quad f(x), \\
\text{subject to} & \quad x_i \geq l_i, \ i = 1, \ldots, m_1, \\
& \quad x_{m_1+i}^2 + x_{m_1+2i}^2 = r_i^2, \ i = 1, \ldots, m_2,
\end{align*}
\]

(6.1) where \( m_1 + 2m_2 < n \) and \( l_i, r_i \) are given. The feasible set \( \Omega \) in (6.1) is described by \( \Omega_i, i = 1, \ldots, m, \) \( m = m_1 + m_2 + 1, \) of three different types. Recall that each of \( \Omega_i \) is defined by \( \Omega_i = \{x_i \in \mathbb{R}^{n_i} : f_i(x_i) \leq 0\} \) where \( f_i : \mathbb{R}^{n_i} \to \mathbb{R} \) is continuously differentiable and convex. Let us denote the feasible steplength with respect to the \( i \)th constraint by

\[
\alpha_{f,i} = \max\{\alpha : x_i - \alpha p_i \in \Omega_i\}.
\]

(1) Let \( n_i = 1, \) and \( f_i(x_i) \equiv l_i - x_i. \) Then \( x_i \in \Omega_i, i = 1, \ldots, m_1 \) represent the simple bounds in (6.1). The set \( \Omega_i \) describes the half-line so that the projection is given by

\[
P_{\Omega_i}(x_i) = \begin{cases} 
  x_i & \text{for } l_i \leq x_i, \\
  l_i & \text{for } l_i > x_i
\end{cases}
\]

and the feasible steplength by

\[
\alpha_{f,i} = \begin{cases} 
  (x_i - l_i)/p_i & \text{for } p_i > 0, \\
  \infty & \text{for } p_i \leq 0
\end{cases}
\]
(2) Let \( n_{m_1+} = 2 \), \( f_{m_1+i}(x_{m_1+i}) \equiv x_{m_1+i}^T x_{m_1+i} - r_i^2 \), and \( x_{m_1+i} = (x_{m_1+2i-1}, x_{m_1+2i})^T \). Then \( x_{m_1+i} \in \Omega_{m_1+i} \), \( i = 1, \ldots, m_2 \) represent the circular constraints in (6.1). The set \( \Omega_{m_1+i} \) describes the circle with the center at the origin of \( \mathbb{R}^2 \) and with the radius \( r_i \) so that the projection is given by

\[
P_{\Omega_{m_1+i}}(x_{m_1+i}) = \begin{cases} 
  x_{m_1+i} & \text{for } \|x_{m_1+i}\| \leq r_i, \\
  \frac{r_i}{\|x_{m_1+i}\|} x_{m_1+i} & \text{for } \|x_{m_1+i}\| > r_i
\end{cases}
\]

and the feasible steplength by

\[
\alpha_{f,m_1+i} = \begin{cases} 
 (x_i^T p_i + \sqrt{(x_i^T p_i)^2 - (\|x_i\|^2 - r_i^2)\|p_i\|^2})/\|p_i\|^2 & \text{for } p_i \neq 0, \\
\infty & \text{for } p_i = 0.
\end{cases}
\]

(3) Let \( n_m = n - m_1 - 2m_2 \), \( x_m = (x_{m_1+2m_2+1}, \ldots, x_n)^T \), and \( f_m(x_m) = -1 \). Then \( \Omega_m = \mathbb{R}^{n_m} \), i.e., the components of \( x_m \) are unconstrained. For completeness, we define

\[
P_{\Omega_m}(x_m) = x_m \quad \text{and} \quad \alpha_{f,m} = \infty.
\]

Let us conclude that the projection \( P_{\Omega} \) is put together by \( P_{\Omega_i} \) as in (2.4) and the feasible steplength \( \alpha_f \) is given by

\[
\alpha_f = \min\{\alpha_{f,i} : i = 1, \ldots, m\}.
\]

7. Numerical tests. We shall assess the performance of the algorithm by three examples. The first one is the benchmark of [10], in which only circular constraints occur. The second example represents a one-dimensional obstacle problem comprising both simple bounds and circular constraints. A more realistic third example shows the solution of frictional 3D contact problems of linear elasticity by means of a sequence of problems (6.1).

Let us note that either we shall use Algorithm 6.1 with \( x^0 = 0 \), \( \Gamma = 1 \), \( \bar{\alpha} = \|A\|^{-1} \), and \( \epsilon = 10^{-5}\|b\| \), or we shall comment different choices.

Example 7.1. Let us consider the problem (6.1) for \( n = 12 \), \( m_1 = 0 \), and \( m_2 = 6 \) with the five-diagonal matrix \( A \):

\[
A = (a_{ij}), \quad a_{ii} = 4, \quad a_{ii+1} = a_{ii-1} = -1, \quad b = Ay, \quad r = (2, 1, 0.5, 2, 10^{-3}, 154)^T,
\]

and \( y = (2, 1, 0.5, 0, 0, 11, 10^{-5}, -1, 1, -0.1, 4.1 \cdot 10^{-4}, 143)^T \). The solution \( x^* \in \mathbb{R}^{12} \) has three active constraints so that \( A(x^*) = \{2, 3, 5\} \). Here, we shall denote Algorithm 6.1 by QPC and the algorithm of [10] by QPQ. In Table 7.1 we compare QPQ and QPQ by the number of matrix-vector multiplications for various \( \Gamma \). The algorithm QPC has better performance in all cases. Moreover, the experiments are in agreement with the conclusions of Theorem 5.1.

In order to explain the progress, we mention the ideas of QPQ. This algorithm is constructed by the KKT conditions that are given here in Theorem 2.1. Introducing the turned boundary gradient \( \beta = \beta(x) \) at \( x \in \Omega \) as

\[
\beta_i = 0 \quad \text{for } i \in \mathcal{F}(x), \quad \beta_i = g_i + \frac{\|g_i\|}{\|\nabla f_i(x_i)\|} \nabla f_i(x_i) \quad \text{for } i \in \mathcal{A}(x),
\]

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
Table 7.1
Comparisons of the algorithms QPQ and QPC.

<table>
<thead>
<tr>
<th>( \Gamma )</th>
<th>100</th>
<th>10</th>
<th>5</th>
<th>1</th>
<th>0.5</th>
<th>0.4</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.01</th>
<th>0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPQ</td>
<td>43</td>
<td>41</td>
<td>39</td>
<td>32</td>
<td>37</td>
<td>40</td>
<td>48</td>
<td>41</td>
<td>38</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>QPC</td>
<td>30</td>
<td>24</td>
<td>21</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td>22</td>
<td>22</td>
<td>26</td>
<td>37</td>
<td>50</td>
</tr>
</tbody>
</table>

Fig. 7.1. The iteration history of QPQ and QPC (logarithmic scale).

we can define the turned gradient by \( \nu = \nu(x) = \varphi(x) + \beta(x) \) where \( \varphi(x) \) is the free gradient (2.9). The solution \( x^* \) to the problem (1.1) is then characterized by \( \nu(x^*) = 0 \). The proportioning step of QPC is replaced in QPQ by the step deactivation, in which \( \beta \) replaces \( \tilde{\beta} \) (for a different steplength). It is easy to show that the deactivation step releases all indices from the current active set, for which the boundary KKT conditions (2.3) are not satisfied. After each release, the inner KKT conditions are violated usually on the same level. This observation is a practical consequence of the fact that the functions \( \beta(x) \) and \( \nu(x) \) are discontinuous (in contrast to \( \tilde{\beta} \) and \( \tilde{\varphi} \)).

The typical situation is drawn in Figure 7.1 (left), where the norm of the projected gradient (solid) and the turned gradient (dotted) are depicted. In Figure 7.1 (right) we can see that oscillations arising in QPQ are eliminated in QPC. Comparing the stagnation levels, we can conclude that QPC is considerably more robust. Notice that the turned gradient did not curiously recognize the solution in QPQ due to round-off errors.

Example 7.2. In the second example, we solve

\[
\min \frac{1}{2} \int_0^1 \|x'(t)\|^2 \, dt - \int_0^1 x(t)^T f(t) \, dt
\]

subject to \( x = (X_1, X_2)^T \in \mathcal{K} \), where

\[
\mathcal{K} = \{ x \in (H^1_0(0,1))^2 : \ X_2(t) \geq l \text{ on } (0,0.5), \ |x(t)| \leq r \text{ on } (0.5,1) \},
\]

and \( f(t) = (36\pi^2 \sin 6\pi t, -4\pi^2 \sin 2\pi t)^T \). This problem describes the loaded wire (see Figure 7.2) that is partially above the plan far off the distance \( l \) and partially inside the cylindrical tube of the radius \( r \). A finite element discretization on a regular grid with \( n \) degrees of freedom leads to the problem (6.1) where \( m_1 = m_2 = n/4, \ l_i = l, \) and \( r_i = r \). In tables below we summarize numbers of matrix-vector multiplications and information on active and free constraints as

\[
n_{b,A} : n_{b,F} / n_{c,A} : n_{c,F},
\]
ON MINIMIZING CONSTRAINED QUADRATIC FUNCTIONS

Fig. 7.2. Geometry of the wire.

Table 7.2

\[ r = 2, \text{ only simple bounds are active in the solution.} \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( l = -1.5 )</th>
<th>( l = -1 )</th>
<th>( l = -0.8 )</th>
<th>( l = -0.5 )</th>
<th>( l = -0.1 )</th>
<th>( l = 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0.8/0.8</td>
<td>1.7/0.8</td>
<td>1.7/0.8</td>
<td>3.5/0.8</td>
<td>4.4/0.8</td>
<td>5.3/0.8</td>
</tr>
<tr>
<td>64</td>
<td>0.16/0.16</td>
<td>1.15/0.16</td>
<td>1.15/0.16</td>
<td>4.12/0.16</td>
<td>6.10/0.16</td>
<td>9.7/0.16</td>
</tr>
<tr>
<td>128</td>
<td>0.32/0.32</td>
<td>0.31/0.32</td>
<td>3.29/0.32</td>
<td>6.26/0.32</td>
<td>12.23/0.32</td>
<td>19.13/0.32</td>
</tr>
<tr>
<td>256</td>
<td>0.64/0.64</td>
<td>1.63/0.64</td>
<td>4.60/0.64</td>
<td>10.54/0.64</td>
<td>23.41/0.64</td>
<td>37.27/0.64</td>
</tr>
<tr>
<td>512</td>
<td>0.128/0.128</td>
<td>1.127/0.128</td>
<td>8.120/0.128</td>
<td>21.107/0.128</td>
<td>45.83/0.128</td>
<td>73.55/0.128</td>
</tr>
<tr>
<td>1024</td>
<td>0.256/0.256</td>
<td>1.255/0.256</td>
<td>16.240/0.256</td>
<td>40.216/0.256</td>
<td>89.167/0.256</td>
<td>146.110/0.256</td>
</tr>
</tbody>
</table>

Table 7.3

\[ r = 0, \text{ both simple bounds and circular constraints are active in the solution.} \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( r = 1.4 )</th>
<th>( r = 1 )</th>
<th>( r = 0.5 )</th>
<th>( r = 0.3 )</th>
<th>( r = 0.01 )</th>
<th>( r = 0.001 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>5.8/2.6</td>
<td>6.2/4.4</td>
<td>7.1/5.3</td>
<td>7.1/5.3</td>
<td>8.0/8.0</td>
<td>8.0/8.0</td>
</tr>
<tr>
<td>64</td>
<td>10.6/2.14</td>
<td>11.5/5.11</td>
<td>13.3/6.10</td>
<td>14.2/9.7</td>
<td>16.0/16.0</td>
<td>16.0/16.0</td>
</tr>
<tr>
<td>128</td>
<td>20.12/4.28</td>
<td>22.10/5.27</td>
<td>26.6/10.22</td>
<td>29.3/16.16</td>
<td>32.0/31.1</td>
<td>32.0/32.0</td>
</tr>
<tr>
<td>256</td>
<td>60.8/6.20</td>
<td>62.0/324</td>
<td>324/239</td>
<td>96/41</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>39.25/4.60</td>
<td>45.19/8.56</td>
<td>52.12/18.46</td>
<td>57.7/26.38</td>
<td>64.0/58.6</td>
<td>64.0/64.0</td>
</tr>
<tr>
<td>1024</td>
<td>155.101/6.25</td>
<td>177.79/22.24</td>
<td>208.48/60.196</td>
<td>228.28/95.161</td>
<td>253.3/219.37</td>
<td>256.0/249.7</td>
</tr>
</tbody>
</table>

\[ n_{b,A}, n_{b,F}, n_{c,A}, \text{ and } n_{c,F} \text{ are numbers of active simple bounds, free simple bounds, active circular constraints, and free circular constraints, respectively.} \]

(i) Let \( r = 2 \) so that no circular constraint is active in the solution; see Table 7.2. In this case, Algorithm 6.1 reduces to a variant of the algorithm of [5] with the finite terminating property. Let us note that the column labeled \( l = -1.5 \) corresponds to the unconstrained problem, in which our algorithm became the conjugate gradient method.

(ii) Let \( l = 0 \) so that both simple bounds and circular constraints may be active in the solution; see Table 7.3. It turns out that the algorithm is more efficient in situations when the constraints are tighter, i.e., when \( r \) and \( l \) are near zero so that the number of active constraints is considerably higher than the number of free constraints.
Example 7.3. Let us consider the steel brick lying on a rigid foundation as it is shown in Figure 7.3. The brick occupies in the reference configuration the domain \( \omega \subset \mathbb{R}^3 \), whose boundary \( \partial \omega \) is split into three nonempty disjoint parts \( \gamma_u \), \( \gamma_p \), and \( \gamma_c \) with different boundary conditions. The zero displacements are prescribed on \( \gamma_u \), whereas the surface tractions act on \( \gamma_p \). On \( \gamma_c \), we consider the contact conditions, i.e., the nonpenetration and the effect of friction. The elastic behavior of the brick is described by Lamé equations that, after finite element discretization, lead to a symmetric positive definite stiffness matrix \( K \in \mathbb{R}^{3n_c \times 3n_c} \) and to a load vector \( f \in \mathbb{R}^{3n_c} \). Moreover, we introduce full rank matrices \( N, T_1, T_2 \in \mathbb{R}^{m_c \times 3n_c} \) projecting displacements at contact nodes to normal and tangential directions, respectively, and we denote 
\[
B = (N^\top, T_1^\top, T_2^\top)^\top \in \mathbb{R}^{3m_c \times 3n_c}.
\]

The contact problem with Tresca friction uses the friction law, in which the slip bound \( r \in \mathbb{R}^{m_c}_+ \) depends on the normal contact stress \( \lambda_\nu \in \mathbb{R}^{m_c}_+ \) by 
\[
r = F \lambda_\nu,
\]
where \( F > 0 \) is a coefficient of friction. As \( r \) is the input for (7.1) and \( \lambda_\nu \) is the output, the problem with Tresca friction defines the mapping 
\[
\Psi : \mathbb{R}^{m_c}_+ \rightarrow \mathbb{R}^{m_c}_+ : r \mapsto F \lambda_\nu.
\]
It is easily seen that a fixed point of \( \Psi \) solves the problem with Coulomb friction, i.e., a point \( r \) such that \( \Psi(r) = r \). Notice that \( \Psi \) is contractive for sufficiently small \( F \),
and then there is a unique fixed point. Moreover, successive approximations can be used for its computation:

\[ r^0 \in \mathbb{R}^{m_c} \text{ given; for } k = 1, 2, \ldots \text{ set } r^k = \Psi(r^{k-1}). \]

As the evaluation of \( \Psi \) requires us to solve (7.1), we can repeatedly apply Algorithm 6.1. In order to perform these computations efficiently, we initialize each iteration by results from the previous one. Finally we terminate if

\[ \|r^k - r^{k-1}\|/\|r^k\| \leq 10^{-4}. \]

In our numerical experiments, we consider the steel brick \( \omega = (0, 3) \times (0, 1) \times (0, 1) \) partitioned into \( 3N \times N \times N \) cubes by trilinear finite elements for \( N = 4, 6, 8, 10, 12, \) and 14. The size of problems solved by Algorithm 6.1 is \( n = 3m_c = 9N(N + 1) \). In Table 7.4, we report CPU time in seconds (time), the number of successive approximations (iter), the total complexity by matrix-vector multiplications (nQ), and the relative complexity (nQ/n). The computations are carried out in Matlab 7 on Pentium(R)4, 3GHz, 512MB. The obtained results are promising; especially, \( n_Q \) is only mildly dependent on the finite element discretization so that the relative complexity considerably decreases for finer grids.

8. Comments and conclusions. We have analyzed a new active set algorithm for minimizing strictly convex quadratic functions with separable convex constraints. It generalizes a recently developed algorithm of quadratic programming constrained by simple bounds [5]. It should be noted that we did not need any requirement on nondegeneracy of the problem so that our algorithm is globally convergent for both the nondegenerate as well as the degenerate case.

The main goal is the proof of a linear convergence rate, which is in an optimal case described by the factor \( \eta = 1 - \frac{1}{4} \kappa(A)^{-1} \) in terms of the condition number of the Hessian matrix \( A \). Notice that \( \eta \) is not influenced by constraints. The key assumption is the restricted steplength \( \tilde{\alpha} \) defining the projected gradient \( \tilde{g} \), i.e., \( \tilde{\alpha} \leq \|A\|^{-1} \), which means that the components \( \tilde{\varphi} \) and \( \tilde{\beta} \) of \( \tilde{g} \) are descent directions. We use them for adding/releasing indices to/from the active set in the step expansion/proportioning. Unfortunately, our analysis requires us to replace the conjugate gradient steplengths by \( \tilde{\alpha} \), which seems to be too restrictive, and the obtained convergence rate may be a bit pessimistic. That is the case for simple bound problems. If the constraints are conic (e.g., circular), the algorithm usually performs few valuable conjugate gradient steps combined with expansion steps, and then it alternates many proportioning steps with short conjugate gradient steps. The convergence rate is more realistic in such situations.
The algorithm presented here is an important ingredient in the numerical solution of 3D contact problems. It was shown in [3] that FETI domain decomposition methods are scalable for the frictionless contact. Our paper enables us to extend this result for frictional problems. It will be published in forthcoming papers.

Another class of algorithms relevant for our research (contact problems) is based on a specific semismooth Newton method that is identical again with an active set strategy [9]. The fundamental discrepancy is the fact that it allows infeasible iterates. In this case, the convergence rate is superlinear but requires a sufficiently accurate initial approximation. In general, the computational performance is comparable.

REFERENCES