Interior Point Method for Nonlinear Nonconvex Optimization

Lukšan, Ladislav
2001

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Interior-point method for nonlinear nonconvex optimization

Ladislav Lukšan and Jan Vlček

Technical report No. 836

September 2001

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodrenskou v 2, 182 07 Prague 8, Czech Republic
phone: (+4202) 6884244 fax: (+4202) 8585789
e-mail: uivt@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Interior-point method for nonlinear nonconvex optimization [1]

Ladislav Lukšan and Jan Vlček[2]

Technical report No. 836
September 2001

Abstract

In this paper, we propose an algorithm for solving nonlinear nonconvex programming problems, which is based on the interior-point approach. Main theoretical results concern direction determination and step-length selection. We split inequality constraints into active and inactive to overcome problems with stability. Inactive constraints are eliminated directly while active constraints are used to define symmetric indefinite linear system. Inexact solution of this system is obtained iteratively using indefinitely preconditioned conjugate gradient method. Theorems confirming efficiency of several indefinite preconditioners are proved. Furthermore, new merit function is defined, which includes effect of possible regularization. This regularization can be used to overcome problems with near linear dependence of active constraints. The algorithm was implemented in the interactive system for universal functional optimization UFO. Results of extensive numerical experiments are reported.

Keywords

Nonlinear programming, interior-point methods, indefinite systems, indefinite preconditioners, preconditioned conjugate gradient method, merit functions, algorithms, computational experiments

[1] Work supported by grant GA ČR 201/00/0080.

[2] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic, E-mail: luksan@cs.cas.cz, vlcek@cs.cas.cz

INTERIOR POINT METHOD FOR NONLINEAR NONCONVEX OPTIMIZATION [3]

Ladislav Lukšan, Jan Vlček, Institute of Computer Science AVČR,
Pod vodárenskou věží 2, 182 07 Praha 8, Czech republic and Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

1 Introduction

In this contribution, we are concerned with a general nonlinear programming problem:

(NP): Find the minimum of function $f(x)$ on the set given by constraints $c_I(x) \leq 0$ and $c_E(x) = 0$, where $f : R^n \to R$, $c_I : R^n \to R^{m_I}$ and $c_E : R^n \to R^{m_E}$ are twice continuously differentiable mappings ($c_I \leq 0$ is considered by elements) and $I = \{1, \ldots, m_I\}$, $E = \{m_I + 1, \ldots, m_I + m_E\}$.

Necessary conditions (the KKT - conditions) for the solution of problem (NP) (if gradients of active constraints are linearly independent) have the following form

$$\begin{aligned} g(x,u) &= 0, && (1.1)\\ c_I(x) &\leq 0, \quad u_I \geq 0, \quad u_I^T c_I(x) = 0, && (1.2)\\ c_E(x) &= 0, && (1.3) \end{aligned}$$

where

$$g(x,u) = \nabla f(x) + \sum_{i \in I} u_i \nabla c_i(x) + \sum_{i \in E} u_i \nabla c_i(x) = \nabla f(x) + A_I(x) u_I + A_E(x) u_E \quad (1.4)$$

and $A_I(x) = [\nabla c_i(x) : i \in I]$, $A_E(x) = [\nabla c_i(x) : i \in E]$. Here $u_I \in R^{m_I}$, $u_E \in R^{m_E}$ are vectors of Lagrange multipliers.

We use the idea of interior-point methods, which is based on the introduction of a slack vector $s_I \in R^{m_I}$ and the transformation of the original problem to the following sequence of problems with the logarithmic barrier function:

(IP): For a given value $\mu > 0$, find the minimum of function $f(x) - \mu e^T \ln(S_I) e$ with constraints $c_I(x) + s_I = 0$ and $c_E(x) = 0$, where e is the vector with unit elements and $S_I = \mathrm{diag}(s_i : i \in I)$ (we assume that $\mu \to 0$).

The logarithmic barrier term is used to ensure the inequality $s_I \geq 0$ implicitly (a detailed explanation of this idea can be found in [12], [27], [28]).

[3] This work was supported by grant GAČR No. 201/00/0080

Necessary conditions (the KKT - conditions) for the solution of problem (IP) (if gradients of active constraints are linearly independent) have the following form

$$\begin{aligned} g(x,u) &= 0, && (1.5)\\ S_I U_I e - \mu e &= 0, && (1.6)\\ c_I(x) + s_I &= 0, && (1.7)\\ c_E(x) &= 0, && (1.8) \end{aligned}$$

where $g(x,u)$ is given by (1.4) and $U_I = \mathrm{diag}(u_i : i \in I)$. Note that (1.5)–(1.8) with $s_I \geq 0$ is equivalent to (1.1)–(1.3) if $\mu = 0$. Linearizing (1.5)–(1.8), we get a step of the Newton method

$$\begin{bmatrix} G & 0 & A_I & A_E \\ 0 & U_I & S_I & 0 \\ A_I^T & I & 0 & 0 \\ A_E^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s_I \\ \Delta u_I \\ \Delta u_E \end{bmatrix} = - \begin{bmatrix} g \\ S_I U_I e - \mu e \\ c_I + s_I \\ c_E \end{bmatrix}, \qquad (1.9)$$

where $g = g(x,u)$ and $G = G(x,u) = \nabla^2 f(x) + \sum_{i \in I} u_i \nabla^2 c_i(x) + \sum_{i \in E} u_i \nabla^2 c_i(x)$. We assume, that matrix of this system is nonsingular (more details are given in Section 2).

The algorithm of an interior-point method can be roughly described in the following form. For given vectors $x \in R^n$, $s_I \in R^{m_I}$, $u_I \in R^{m_I}$, $u_E \in R^{m_E}$ such that $s_I > 0$, $u_I > 0$ and a given barrier parameter $\mu > 0$, we determine direction vectors Δx, Δs_I, Δu_I, Δu_E by solving linear system equivalent to (1.9) (more details are given in Section 2). Furthermore, we choose step-length $0 < \alpha \leq \overline{\alpha}$, and set $x := x + \alpha \Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta u_I)$, $u_E := u_E + \alpha \Delta u_E$, where $s_I(\alpha, \Delta s_I) > 0$ and $u_I(\alpha, \Delta u_I) > 0$ are functions of α depending on Δs_I and Δu_I, which are chosen by a suitable strategy (more details are given in Section 3). Finally, we determine a new barrier parameter $\mu > 0$ (see Section 4). Notice that condition $s_I > 0$ is necessary for the definition of a logarithmic barrier function and condition $u_I > 0$, motivated by (1.6), is necessary for the construction of an efficient preconditioner used in Section 2.

Interior-point methods were developed for solving linear, quadratic or convex programming problems, since they have a polynomial complexity in these cases, see [28] (simplex type methods do not have this property). Application of interior-point methods to nonlinear nonconvex programming is a quite new approach, which is suitable especially for large-scale sparse problems (see, e.g., [12], [27]). We have immediately influenced by ideas proposed in [3], [23], [24], [25]. The main claim of this paper consists in some new ideas concerning direction determination and step-length selection. In Section 2, we propose a variant of indefinitely preconditioned conjugate gradient method for solving a system equivalent to (1.9). We prove several new theorems concerning its convergence. In Section 3, we propose a new merit function and prove that the direction vector, which is obtained by an inexact solution of a system equivalent to (1.9), is descent for this function. The effect of regularization, which can be used to overcome problems with linear dependence of active constraints, is included into the merit function. The algorithm proposed in Section 5 was implemented in interactive system for universal functional optimization UFO [15]. Section 6 contains results of extensive numerical experiments. Some ideas used in this contribution were motivated by [16] and [18].

2 Direction determination

System (1.9) is nonsymmetric with the dimension $n + m_E + 2m_I$. This system can be symmetrized and reduced by the elimination of vector Δs_I. One has

$$\Delta s_I = -U_I^{-1} S_I (u_I + \Delta u_I) + \mu U_I^{-1} e \tag{2.1}$$

so that

$$\begin{bmatrix} G & A_I & A_E \\ A_I^T & -U_I^{-1} S_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u_I \\ \Delta u_E \end{bmatrix} = - \begin{bmatrix} g \\ c_I + \mu U_I^{-1} e \\ c_E \end{bmatrix}. \tag{2.2}$$

Equation (1.6) implies that $S_I U_I e \approx \mu e$ and if $\mu \to 0$, then either $u_i \to 0$ or $s_i \to 0$ holds for every index $i \in I$. Now we split the set of inequality constraints to an active subset, where $\hat{s}_I \leq \varepsilon_I \hat{u}_I$, and an inactive subset, where $\check{s}_I > \varepsilon_I \check{u}_I$. In the same way, we split and denote other quantities corresponding to inequality constraints. By elimination of inactive equations, we obtain

$$\Delta \check{u}_I = \check{S}_I^{-1} \check{U}_I (\check{c}_I + \check{A}_I^T \Delta x) + \mu \check{S}_I^{-1} e \tag{2.3}$$

so that

$$\begin{bmatrix} \hat{G} & \hat{A}_I & A_E \\ \hat{A}_I^T & -\hat{U}_I^{-1} \hat{S}_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \hat{u}_I \\ \Delta u_E \end{bmatrix} = - \begin{bmatrix} \hat{g} \\ \hat{c}_I + \mu \hat{U}_I^{-1} e \\ c_E \end{bmatrix}, \tag{2.4}$$

where

$$\begin{aligned} \hat{G} &= G + \check{A}_I \check{S}_I^{-1} \check{U}_I \check{A}_I^T, \\ \hat{g} &= g + \check{A}_I \check{S}_I^{-1} \check{U}_I \check{c}_I + \mu \check{A}_I \check{S}_I{}^{-1} e. \end{aligned}$$

Both matrices $\hat{G}$ and $\hat{U}_I^{-1} \hat{S}_I$ are bounded (we assume that G and A are bounded) and if the strict complementarity conditions $\lim_{\mu \to 0} (s_i + u_i) > 0 \ \forall i \in I$ hold (recall that $s_i > 0$ and $u_i > 0$), then one has $\lim_{\mu \to 0} \hat{U}_I^{-1} \hat{S}_I = 0$. This property is very useful, since system (2.4) can be efficiently preconditioned by the way described in [16] if $\hat{U}_I^{-1} \hat{S}_I$ has small elements.

Using this approach, we can split equality (2.1) into two equalities. We obtain

$$\Delta \hat{s}_I = -\hat{U}_I^{-1} \hat{S}_I (\hat{u}_I + \Delta \hat{u}_I) + \mu \hat{U}_I^{-1} e, \tag{2.5}$$

$$\Delta \check{s}_I = -(\check{c}_I + \check{A}_I^T \Delta x + \check{s}_I) \tag{2.6}$$

after arrangements. Vector $\Delta \hat{u}_I$ is determined by solving system (2.4) and vector $\Delta \check{u}_I$ is computed from (2.3). Matrix $\check{S}_I^{-1} \check{U}_I$ is bounded and if the strict complementarity conditions hold, then $\lim_{\mu \to 0} \check{S}_I^{-1} \check{U}_I = 0$. Since $S_I U_I e \sim \mu e \to 0$, vectors $\mu \check{S}_I^{-1} e \sim \check{U}_I e$ and $\mu \hat{U}_I^{-1} e \sim \hat{S} e$ used in (2.3) and (2.5) are also bounded.

2.1 Indefinitely preconditioned conjugate gradient method

To simplify the notation in the subsequent analysis, we rewrite system (2.4) in the form

$$K\bar{d} = \begin{bmatrix} \hat{G} & \hat{A}_I & A_E \\ \hat{A}_I^T & -\hat{M}_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ \hat{d}_I \\ d_E \end{bmatrix} = \begin{bmatrix} b \\ \hat{b}_I \\ b_E \end{bmatrix} = \bar{b} \tag{2.7}$$

(i.e., we introduce vectors $\bar{d}$ and $\bar{b}$). Here $\hat{M}_I = \hat{U}_I^{-1}\hat{S}_I$ is a positive definite diagonal matrix. We assume that matrix K is nonsingular, which implies that A_E has a full column rank. Therefore, all inversions used in the subsequent analysis make sure. System (2.7), which is symmetric and indefinite of order $n + \hat{m}_I + m_E$, can be solved either directly by using the sparse Bunch-Parlett decomposition or iteratively by using Krylov-subspace methods for symmetric indefinite systems. Motivated by [16] (see also [4], [11], [14], [19], [20]) we first investigate the preconditioner

$$C = \begin{bmatrix} \hat{D} & \hat{A}_I & A_E \\ \hat{A}_I^T & -\hat{M}_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix}, \tag{2.8}$$

where $\hat{D}$ is a positive definite diagonal matrix derived from the diagonal of $\hat{G}$, see [16]. Note that C is nonsingular when A_E has a full column rank (see (2.11)). We restrict to the situation when matrix $\hat{G} - \hat{D}$ is nonsingular. This is an usual situation and also the worst case in some sense (the Krylov subspace used in Theorem 2 has a lover dimension if $\hat{G} - \hat{D}$ is singular, see [14]).

Now we prove three theorems which explain good properties of preconditioner (2.8) and the high efficiency of the conjugate gradient method preconditioned by (2.8). These theorems are generalizations of theorems given in [16] to the case when $m_I \neq 0$.

Theorem 1. *Consider preconditioner (2.8) applied to system (2.7) and assume that $\hat{G} - \hat{D}$ is nonsingular. Then matrix KC^{-1} has at least $\hat{m}_I + 2m_E$ unit eigenvalues but at most $\hat{m}_I + m_E$ linearly independent eigenvectors corresponding to these eigenvalues exist. The other eigenvalues of matrix KC^{-1} are exactly eigenvalues of matrix $Z_E^T\tilde{G}Z_E(Z_E^T\tilde{D}Z_E)^{-1}$, where $[Z_E, A_E]$ is a nonsingular square matrix, $Z_E^T A_E = 0$, $Z_E^T Z_E = I$ and where $\tilde{G} = \hat{G} + \hat{A}_I\hat{M}_I^{-1}\hat{A}_I^T$, $\tilde{D} = \hat{D} + \hat{A}_I\hat{M}_I^{-1}\hat{A}_I^T$. If $Z_E^T\tilde{G}Z_E$ is positive definite then all eigenvalues are positive.*

Proof. First we derive the expression for matrix $KC^{-1} = (C^{-1}K)^T$. To do it, we formally write $C\bar{x} = K\bar{y}$, i.e.

$$\begin{aligned} \hat{D}x + \hat{A}_I\hat{x}_I + A_E x_E &= \hat{G}y + \hat{A}_I\hat{y}_I + A_E y_E \\ \hat{A}_I^T x - \hat{M}_I\hat{x}_I &= \hat{A}_I^T y - \hat{M}_I\hat{y}_I \\ A_E^T x &= A_E^T y. \end{aligned}$$

and using a block structure of C we eliminate vector $\bar{x}$ obtaining $\bar{x} = C^{-1}K\bar{y}$. This is a formally complicated but straightforward procedure so we give the final result (after transposition):

$$KC^{-1} = \begin{bmatrix} (\tilde{G} - \tilde{D})\tilde{P} + I & \hat{H}_I & H_E \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \tag{2.9}$$

where $\tilde{G} = \hat{G} + \hat{A}_I \hat{M}_I^{-1} \hat{A}_I^T$, $\tilde{D} = \hat{D} + \hat{A}_I \hat{M}_I^{-1} \hat{A}_I^T$ and $\tilde{P} = \tilde{D}^{-1} - \tilde{D}^{-1} A_E (A_E^T \tilde{D}^{-1} A_E)^{-1} A_E^T \tilde{D}^{-1}$ (explicit expressions for $\hat{H}_I$ and H_E are not important). Now it is clear that matrix KC^{-1} has at least $\hat{m}_I + m_E$ unit eigenvalues which correspond to unit matrices in its second and third rows. The remaining eigenvalues are eigenvalues of matrix $(\tilde{G} - \tilde{D})\tilde{P} + I$ so they have to satisfy the equation $(\tilde{G} - \tilde{D})\tilde{P}x = (\lambda - 1)x$. Since $\tilde{P}x = 0$ if and only if $x = A_E u$, $u \in R^{m_E}$, we obtain at least m_E additional unit eigenvalues corresponding to the m_E linearly independent eigenvectors $A_E u$, $u \in R^{m_E}$. Now let us assume that $\lambda \neq 1$. Since eigenvalues of KC^{-1} have to satisfy equations

$$\begin{aligned} \hat{G}x + \hat{A}_I \hat{x}_I + A_E x_E &= \lambda(\hat{D}x + \hat{A}_I \hat{x}_I + A_E x_E), \\ \hat{A}_I^T x - \hat{M}_I \hat{x}_I &= \lambda(\hat{A}_I^T x - \hat{M}_I \hat{x}_I), \\ A_E^T x &= \lambda A_E^T x, \end{aligned} \tag{2.10}$$

we obtain

$$\begin{aligned} \hat{A}_I^T x - \hat{M}_I \hat{x}_I &= 0, \\ A_E^T x &= 0, \end{aligned}$$

which gives $x = Z_E u$ and $\hat{x}_I = \hat{M}_I^{-1} \hat{A}_I^T x = \hat{M}_I^{-1} \hat{A}_I^T Z_E u$, where $[Z_E, A_E]$ is a nonsingular square matrix, $Z_E^T A_E = 0$ and $Z_E^T Z_E = I$. Substituting this result into the first equation of (2.10) and premultiplying the resulting equation by Z_E^T, we obtain

$$(Z_E^T \hat{G} Z_E + Z_E^T \hat{A}_I \hat{M}_I^{-1} \hat{A}_I^T Z_E)u = \lambda(Z_E^T \hat{D} Z_E + Z_E^T \hat{A}_I \hat{M}_I^{-1} \hat{A}_I^T Z_E)u$$

so that λ is an eigenvalue of matrix $Z_E^T \tilde{G} Z_E (Z_E^T \tilde{D} Z_E)^{-1}$. It is obvious that $\lambda > 0$ if matrix $Z_E^T \tilde{G} Z_E$ is positive definite. If $\lambda = 1$, then the first equation of (2.10) gives

$$\hat{G}x + \hat{A}_I \hat{x}_I + A_E x_E = \hat{D}x + \hat{A}_I \hat{x}_I + A_E x_E$$

so that $\hat{x}_I \in R^{\hat{m}_I}$ and $x_E \in R^{m_E}$ can be arbitrary and non-singularity of $\hat{G} - \hat{D}$ implies $x = 0$. Therefore, if $\hat{G} - \hat{D}$ is nonsingular, then matrix KC^{-1} has at most $\hat{m}_I + m_E$ linearly independent eigenvenvectors corresponding to the unit eigenvalues. □

There are two important cases. If $\hat{m}_I = 0$, we obtain Theorem 3.3 proposed in [16]. If $m_E = 0$, then matrix KC^{-1} has at least $\hat{m}_I$ unit eigenvalues with $\hat{m}_I$ linearly independent eigenvectors and the other eigenvalues are exactly eigenvalues of matrix $\tilde{G}\tilde{D}^{-1}$. If $\hat{M}_I \to \infty$, then $\tilde{G}\tilde{D}^{-1} \approx \hat{G}\hat{D}^{-1}$, which leads to a diagonally preconditioned system. If $\hat{M}_I \to 0$, then term $\hat{A}_I \hat{M}_I^{-1} \hat{A}_I^T$ dominates and $\tilde{G}\tilde{D}^{-1} \approx \hat{Z}_I^T \hat{G} \hat{Z}_I (\hat{Z}_I^T \hat{D} \hat{Z}_I)^{-1}$, where $[\hat{Z}_I, \hat{A}_I]$ is a nonsingular square matrix, $\hat{Z}_I^T \hat{A}_I = 0$ and $\hat{Z}_I^T \hat{Z}_I = I$.

Theorem 1 shows that the efficiency of a Krylov-subspace method preconditioned by (2.8) does not depend strongly on the choice of active and inactive variables (matrix $\tilde{G} = G + \check{A}_I \check{S}_I^{-1} \check{U}_I \check{A}_I^T + \hat{A}_I \hat{S}_I^{-1} \hat{U}_I \hat{A}_I^T$ is the same for every splitting). Nevertheless, the splitting influences the resulting (inexact) solution of system (1.9), since the equations corresponding to inactive variables are determined exactly by a direct elimination while active variables are obtained inexactly by an iterative process, which is prematurely terminated (a variant of the inexact Newton method described in [9] is used in the algorithm given in Section 5).

Theorem 2. *Consider preconditioner (2.8) applied to system (2.7) and assume that* $\hat{G} - \hat{D}$ *is nonsingular. Then Krylov subspace* $\mathcal{K} = \mathrm{span}\{\bar{r}, KC^{-1}\bar{r}, (KC^{-1})^2\bar{r}, \ldots\}$, *where* $\bar{r} \in R^{n+\hat{m}_I+m_E}$, *has a dimension of at most* $\min(n+1, n-m_E+2)$.

Proof. If $\hat{G}-\hat{D}$ is nonsingular, then matrix $(\tilde{G}-\tilde{D})\tilde{P}+I$ has exactly m_E unit eigenvalues with m_E linearly independent eigenvectors $a_{m_I+1}, \ldots, a_{m_I+m_E}$ (here $a_{m_I+1}, \ldots, a_{m_I+m_E}$ are columns of matrix A_E). The other eigenvectors of $(\tilde{G}-\tilde{D})\tilde{P}+I$ define an invariant subspace of dimension $n-m_E$. Therefore, we can write

$$\begin{aligned} r &= \sum_{i=m_I+1}^{m_I+m_E} \alpha_i a_i + \sum_{j=1}^{n-m_E} \beta_j w_j \\ \hat{H}_I \hat{r}_I &= \sum_{i=m_I+1}^{m_I+m_E} \alpha_i^I a_i + \sum_{j=1}^{n-m_E} \beta_j^I w_j \\ H_E r_E &= \sum_{i=m_I+1}^{m_I+m_E} \alpha_i^E a_i + \sum_{j=1}^{n-m_E} \beta_j^E w_j \end{aligned}$$

where $w_1, \ldots, w_{n-m_E}$ form a basis in the invariant subspace of $(\tilde{G}-\tilde{D})\tilde{P}+I$ (here r, $\hat{r}_I$, r_E are components of $\bar{r}$). Now we use the induction. Assume that

$$(KC^{-1})^k\bar{r} = \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} \alpha_i a_i \\ \hat{r}_I \\ r_E \end{bmatrix} + k \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} (\alpha_i^I + \alpha_i^E) a_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n-m_E} \beta_j^k w_j \\ 0 \\ 0 \end{bmatrix}$$

for some $k \geq 0$ (it is obvious for $k=0$). Using (2.9), we can write

$$\begin{aligned} (KC^{-1})^{k+1}\bar{r} &= \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} \alpha_i a_i \\ \hat{r}_I \\ r_E \end{bmatrix} + k \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} (\alpha_i^I + \alpha_i^E) a_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n-m_E} \gamma_j^k w_j \\ 0 \\ 0 \end{bmatrix} \\ &+ \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} (\alpha_i^I + \alpha_i^E) a_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n-m_E} (\beta_j^I + \beta_j^E) w_j \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} \alpha_i a_i \\ \hat{r}_I \\ r_E \end{bmatrix} + (k+1) \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} (\alpha_i^I + \alpha_i^E) a_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n-m_E} \beta_j^{k+1} w_j \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

where γ_j^k, $1 \leq j \leq n-m_E$, are new coordinates in the invariant subspace spanned by the vectors $w_j \in R^n$, $1 \leq j \leq n-m_E$, and $\beta_j^{k+1} = \gamma_j^k + \beta_j^I + \beta_j^E$, $1 \leq j \leq n-m_E$. Thus we have proved by induction that all the vectors $\bar{r}, KC^{-1}\bar{r}, (KC^{-1})^2\bar{r}, \ldots$ are combinations of $n-m_E+2$ vectors

$$\begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} \alpha_i a_i \\ \hat{r}_I \\ r_E \end{bmatrix}, \quad \begin{bmatrix} \sum_{i=m_I+1}^{m_I+m_E} (\alpha_i^I + \alpha_i^E) a_i \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} w_j \\ 0 \\ 0 \end{bmatrix}, \quad 1 \leq j \leq n-m_E,$$

so Krylov subspace $\mathcal{K}$ has a dimension of at most $n - m_E + 2$. If $m_E = 0$, the second vector is zero, so $\mathcal{K}$ has a dimension of at most $n + 1$ in this case. □

If Krylov subspace $\mathcal{K}$ has a dimension of at most $\min(n + 1, n - m_E + 2)$, then using Krylov-subspace method we obtain the solution of system (2.7) after at most $\min(n + 1, n - m_E + 2)$ iterations.

If $\hat{m}_I = 0$, we obtain Theorem 3.4 proposed in [16]. If $m_E = 0$, then Krylov subspace $\mathcal{K}$ has a dimension of at most $n + 1$ and using Krylov-subspace method we obtain the solution after at most $n + 1$ iterations.

Further interesting property of preconditioner (2.8) is demonstrated on the preconditioned conjugate gradient (PCG) method applied to system (2.7) (see Theorem 3). For simplifying the notation, we use the following algorithmic form, which do not use iteration indices (ω is a precision).

Algorithm PCG

$\bar{d}$ – given, $\quad \bar{r} := \bar{b} - K\bar{d}, \quad \beta := 0,$

while $\|\bar{r}\| > \omega\|\bar{b}\|$ **do**

$$\begin{array}{lll} \bar{t} := C^{-1}\bar{r}, & \gamma := \bar{r}^T\bar{t}, & \beta := \beta\gamma, \\ \bar{p} := \bar{t} + \beta\bar{p}, & \bar{q} := K\bar{p}, & \alpha := \gamma/\bar{p}^T\bar{q}, \\ \bar{d} := \bar{d} + \alpha\bar{p}, & \bar{r} := \bar{r} - \alpha\bar{q}, & \beta := 1/\gamma \end{array}$$

end while.

We use the following notation.

$$\bar{d} = \begin{bmatrix} d \\ \hat{d}_I \\ d_E \end{bmatrix}, \quad \bar{p} = \begin{bmatrix} p \\ \hat{p}_I \\ p_E \end{bmatrix}, \quad \bar{q} = \begin{bmatrix} q \\ \hat{q}_I \\ q_E \end{bmatrix}, \quad \bar{r} = \begin{bmatrix} r \\ \hat{r}_I \\ r_E \end{bmatrix}, \quad \bar{t} = \begin{bmatrix} t \\ \hat{t}_I \\ t_E \end{bmatrix},$$

Lemma 1. *Consider Algorithm PCG with preconditioner (2.8) applied to system (2.7). Assume that initial $\bar{d}$ is chosen in such a way that $\hat{r}_I = 0$ and $r_E = 0$ at the start of the algorithm. Let*

$$\begin{aligned} d &= Z_E d_Z + \tilde{D}^{-1} A_E d_A, \\ p &= Z_E p_Z + \tilde{D}^{-1} A_E p_A, \\ t &= Z_E t_Z + \tilde{D}^{-1} A_E t_A \end{aligned}$$

and

$$\begin{aligned} r &= \tilde{D} Z_E (Z_E^T \tilde{D} Z_E)^{-1} r_Z + A_E (A_E^T \tilde{D}^{-1} A_E)^{-1} r_A, \\ q &= \tilde{D} Z_E (Z_E^T \tilde{D} Z_E)^{-1} q_Z + A_E (A_E^T \tilde{D}^{-1} A_E)^{-1} q_A, \end{aligned}$$

which imply

$$\begin{aligned} r_Z &= Z_E^T r, \qquad r_A = A_E \tilde{D}^{-1} r, \\ q_Z &= Z_E^T q, \qquad q_A = A_E \tilde{D}^{-1} q \end{aligned}$$

(all these decompositions are determined uniquely when A_E has linearly independent columns). Then $\hat{r}_I = 0$, $r_E = 0$ and

$$\begin{aligned}
t_Z &:= (Z_E^T \tilde{D} Z_E)^{-1} r_Z, \quad t_A = 0, \\
\bar{r}^T \bar{t} &= r_Z^T t_Z, \\
p_Z &:= t_Z + \beta p_Z, \qquad p_A = 0, \\
q_Z &:= Z_E^T \tilde{G} Z_E p_Z, \\
\bar{p}^T \bar{q} &= p_Z^T q_Z, \\
d_Z &:= d_Z + \alpha p_Z, \\
r_Z &:= r_Z - \alpha q_Z, \\
d_A &:= d_A + \alpha p_A = d_A
\end{aligned}$$

in all iterations of Algorithm PCG (since $p_A = 0$, d_A remains unchanged).

Proof. Although the proof is carried out by induction, we omit iterative indices and use assignments ":=" to simplify the notation. The inductive assumptions are $\hat{r}_I = 0$, $r_E = 0$, $p_A = 0$ and $\hat{p}_I = \hat{M}_I^{-1} \hat{A}_I^T Z_E p_Z$ (we can set $p_A = 0$ and $\hat{p}_I = \hat{M}_I^{-1} \hat{A}_I^T Z_E p_Z$ at the start of the algorithm, since $\beta = 0$ in the first iteration). Using a straightforward elimination as in the proof of Theorem 1, we obtain

$$C^{-1} = \begin{bmatrix} \tilde{P} & \tilde{P} \hat{A}_I \hat{M}_I^{-1} & \tilde{D}^{-1} A_E C_E^{-1} \\ \hat{M}_I^{-1} \hat{A}_I^T \tilde{P} & \hat{M}_I^{-1} \hat{A}_I^T \tilde{D}^{-1} \hat{A}_I \hat{M}_I^{-1} & \hat{M}_I^{-1} \hat{A}_I^T \tilde{D}^{-1} A_E C_E^{-1} \\ C_E^{-1} A_E^T \tilde{D}^{-1} & C_E^{-1} A_E^T \tilde{D}^{-1} \hat{A}_I \hat{M}_I^{-1} & -C_E^{-1} \end{bmatrix}, \tag{2.11}$$

where $\tilde{D} = \hat{D} + \hat{A}_I \hat{M}_I^{-1} \hat{A}_I^T$, $\tilde{P} = \tilde{D}^{-1} - \tilde{D}^{-1} A_E (A_E^T \tilde{D}^{-1} A_E)^{-1} A_E^T \tilde{D}^{-1}$ and $C_E = A_E^T \tilde{D}^{-1} A_E$. Since $\hat{r}_I = 0$ and $r_E = 0$ by the assumption, we can write

$$\begin{bmatrix} t \\ \hat{t}_I \\ t_E \end{bmatrix} = \begin{bmatrix} \tilde{P} \\ \hat{M}_I^{-1} \hat{A}_I^T \tilde{P} \\ (A_E^T \tilde{D}^{-1} A_E)^{-1} A_E^T \tilde{D}^{-1} \end{bmatrix} r,$$

which together with $\tilde{P} A_E = 0$ and $\tilde{P} \tilde{D} Z_E = Z_E$ gives

$$t = \tilde{P}(\tilde{D} Z_E (Z_E^T \tilde{D} Z_E)^{-1} r_Z + A_E (A_E^T \tilde{D}^{-1} A_E)^{-1} r_A) = Z_E (Z_E^T \tilde{D} Z_E)^{-1} r_Z$$

or

$$t_Z = (Z_E^T \tilde{D} Z_E)^{-1} r_Z, \quad t_A = 0.$$

Similarly

$$\hat{t}_I = \hat{M}_I^{-1} \hat{A}_I^T \tilde{P} r = \hat{M}_I^{-1} \hat{A}_I^T t = \hat{M}_I^{-1} \hat{A}_I^T Z_E t_Z$$

Furthermore

$$\begin{aligned}
\bar{r}^T \bar{t} &= r^T t = r^T Z_E t_Z \\
&= (\tilde{D} Z_E (Z_E^T \tilde{D} Z_E)^{-1} r_Z + A_E (A_E^T \tilde{D}^{-1} A_E)^{-1} r_A)^T Z_E t_Z \\
&= r_Z^T (Z_E^T \tilde{D} Z_E)^{-1} Z_E^T \tilde{D} Z_E t_Z = r_Z^T t_Z.
\end{aligned}$$

Since $p_A = 0$ and $\hat{p}_I = \hat{M}_I^{-1}\hat{A}_I^T Z_E p_Z$ by the assumption, we can write

$$\begin{aligned} p_Z &:= t_Z + \beta p_Z, \\ p_A &:= t_A + \beta p_A = 0, \\ \hat{p}_I &:= \hat{t}_I + \beta \hat{p}_I = \hat{M}_I^{-1}\hat{A}_I^T Z_E (t_Z + \beta p_Z). \end{aligned}$$

Thus again $p_A = 0$ and $\hat{p}_I = \hat{M}_I^{-1}\hat{A}_I^T Z_E p_Z$. Now

$$\begin{bmatrix} q \\ \hat{q}_I \\ q_E \end{bmatrix} = \begin{bmatrix} \hat{G} & \hat{A}_I & A_E \\ \hat{A}_I^T & -\hat{M}_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \hat{p}_I \\ p_E \end{bmatrix},$$

which gives

$$q = \hat{G}p + \hat{A}_I \hat{p}_I + A_E p_E = \hat{G} Z_E p_Z + \hat{A}_I \hat{p}_I + A_E p_E$$

(since $p_A = 0$) so

$$q_Z = Z_E^T q = Z_E^T \hat{G} Z_E p_Z + Z_E^T \hat{A}_I \hat{M}_I^{-1} \hat{A}_I^T Z_E p_Z = Z_E^T \tilde{G} Z_E p_Z.$$

Furthermore

$$\hat{q}_I = \hat{A}_I^T p - \hat{M}_I \hat{p}_I = \hat{A}_I^T Z_E p_Z - \hat{M}_I \hat{M}_I^{-1} \hat{A}_I^T Z_E p_Z = 0$$

(since $\hat{p}_I = \hat{M}_I^{-1}\hat{A}_I^T Z_E p_Z$) and

$$q_E = A_E^T p = A_E^T Z_E p_z = 0.$$

Finally

$$\begin{aligned} \bar{p}^T \bar{q} &= p^T q = p_Z^T Z_E^T q \\ &= p_Z^T Z_E^T (\tilde{D} Z_E (Z_E^T \tilde{D} Z_E)^{-1} q_Z + A_E (A_E^T \tilde{D}^{-1} A_E)^{-1} q_A) \\ &= p_Z^T Z_E^T \tilde{D} Z_E (Z_E^T \tilde{D} Z_E)^{-1} q_Z = p_Z^T q_Z. \end{aligned}$$

and

$$\begin{aligned} d_Z &:= d_Z + \alpha p_Z \\ r_Z &:= r_Z - \alpha q_Z \\ \hat{r}_I &:= \hat{r}_I - \alpha \hat{q}_I = 0 \\ r_E &:= r_E - \alpha q_E = 0 \end{aligned}$$

□

Theorem 3. *Consider Algorithm PCG with preconditioner (2.8) applied to system (2.7). Assume that initial $\bar{d}$ is chosen in such a way that $\hat{r}_I = 0$ and $r_E = 0$ at the start of the algorithm. Let matrix $Z_E^T \tilde{G} Z_E$ be positive definite. Then:*

(a) Vector $d^* = Z_E d_Z^* + \tilde{D}^{-1} A_E d_A^*$ (the first part of vector $\bar{d}^*$ which solves equation $K\bar{d} = \bar{b}$) is found after $n - m_E$ iterations at most.

(b) *The algorithm cannot break down before d^* is found.*

(c) *Error $\|d - d^*\|$ converges to zero at least R - linearly with quotient*

$$\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1},$$

where κ is the spectral condition number of matrix $Z_E^T \tilde{G} Z_E (Z_E^T \tilde{D} Z_E)^{-1}$.

(d) *If $d = d^*$, then also $\hat{d}_I = \hat{d}_I^*$ and d_E^* can be determined by the formula*

$$d_E^* = d_E + (A_E^T \tilde{D}^{-1} A_E)^{-1} A_E^T \tilde{D}^{-1} r.$$

Proof. (a) Lemma 1 implies that if system (2.7) is solved by conjugate gradient method with preconditioner (2.8), then $d = Z_E d_Z + \tilde{D}^{-1} A_E d_A$, where components d_Z are generated by conjugate gradient method with preconditioner $Z_E^T \tilde{D} Z_E$ applied to system $Z_E^T \tilde{G} Z_E d_Z = b_Z$ and components d_A remain unchanged (explicit expression for b_Z is not important). Since $Z_E^T \tilde{G} Z_E$ is positive definite, we obtain the solution d_Z^* and, therefore, d^* after at most $n - m_E$ iterations.
(b) Since denominators used in the conjugate gradient method with preconditioner (2.8) applied to system (2.7) are the same as denominators used in the conjugate gradient method with preconditioner $Z_E^T \tilde{D} Z_E$ applied to system $Z_E^T \tilde{G} Z_E d_Z = b_Z$ and matrix $Z_E^T \tilde{G} Z_E$ is positive definite, then the algorithm cannot break down before d_Z^* and, therefore, d^* is found.
(c) Since d_A remains unchanged and $Z_E^T Z_E = I$, one has $\|d - d^*\| = \|Z_E(d_Z - d_Z^*)\| = \|d_Z - d_Z^*\|$. Thus we can use standard estimation for the conjugate gradient method with preconditioner $Z_E^T \tilde{D} Z_E$ applied to system $Z_E^T \tilde{G} Z_E d_Z = b_Z$.
(d) Since $d = d^*$ and $\hat{r}_I = 0$, we obtain $\hat{d}_I = \hat{d}_I^*$ from (2.7). Furthermore (2.7) implies

$$\begin{aligned} b - \hat{G} d^* - \hat{A}_I \hat{d}_I^* - A_E d_E &= r, \\ b - \hat{G} d^* - \hat{A}_I \hat{d}_I^* - A_E d_E^* &= 0, \end{aligned}$$

which after subtraction gives $A_E(d_E^* - d_E) = r$. Premultiplying this equation by $A_E^T \tilde{D}^{-1}$, we obtain the solution $d_E^* - d_E = (A_E^T \tilde{D}^{-1} A_E)^{-1} A_E^T \tilde{D}^{-1} r$. □

If $\hat{m}_I = 0$, we obtain Theorem 3.5 proposed in [16]. If $m_E = 0$, then $Z_E = I$ so the algorithm is equivalent to conjugate gradient method with preconditioner $\tilde{D}$ applied to system $\tilde{G} d = b$. Notice that we require positive definiteness of matrix $\tilde{G}$ in this case.

Although notation (2.7) is advantageous for theoretical analysis, we use another notation for implementation purposes. If we set $\hat{A} = [\hat{A}_I, A_E]$ and $\hat{M} = \mathrm{diag}(\hat{M}_I, 0)$, we can write

$$K\bar{d} = \begin{bmatrix} \hat{G} & \hat{A} \\ \hat{A}^T & -\hat{M} \end{bmatrix} \begin{bmatrix} d \\ \hat{d} \end{bmatrix} = \begin{bmatrix} b \\ \hat{b} \end{bmatrix} = \bar{b} \tag{2.12}$$

and

$$C = \begin{bmatrix} \hat{D} & \hat{A} \\ \hat{A}^T & -\hat{M} \end{bmatrix}. \tag{2.13}$$

Theorem 3 assumes the initial $\bar{d}$ is chosen in such a way that $\hat{r} = 0$ at the start of the algorithm. Equation (2.12) implies that this condition is satisfied if we set $\hat{d} = 0$ and $d = (\hat{A}^T\hat{D}^{-1}\hat{A})^{-1}\hat{A}^T\hat{D}^{-1}\hat{b}$.

Expressions for matrices K and C given by (2.12) and (2.13) imply that

$$C^{-1} = \begin{bmatrix} \hat{P} & \hat{D}^{-1}\hat{A}(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1} \\ (\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}\hat{A}^T\hat{D}^{-1} & -(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1} \end{bmatrix}$$

and

$$KC^{-1} = \begin{bmatrix} (\hat{G} - \hat{D})\hat{P} + I & (\hat{G} - \hat{D})\hat{D}^{-1}\hat{A}(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1} \\ 0 & I \end{bmatrix},$$

where $\hat{P} = \hat{D}^{-1} - \hat{D}^{-1}\hat{A}(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}\hat{A}_E^T\hat{D}^{-1}$. Using these formulas, Algorithm PCG can be rewritten in the following form

Algorithm PCGa

d – given, $\hat{d} := 0$,
$r := b - \hat{G}d - \hat{A}\hat{d}$, $\hat{r} := \hat{b} - \hat{A}^T d + \hat{M}\hat{d}$,
$\beta := 0$,

while $\|r\| > \omega\|b\|$ or $\|\hat{r}\| > \omega\|\hat{b}\|$ **do**

$t := \hat{D}^{-1}(r - \hat{A}\hat{t})$, $\hat{t} := (\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}(\hat{A}^T\hat{D}^{-1}r - \hat{r})$,
$\gamma := r^T t + \hat{r}^T\hat{t}$, $\beta := \beta\gamma$,
$p := t + \beta p$, $\hat{p} := \hat{t} + \beta\hat{p}$,
$q := \hat{G}p + \hat{A}\hat{p}$, $\hat{q} := \hat{A}^T p - \hat{M}\hat{p}$,
$\alpha := p^T q + \hat{p}^T\hat{q}$, $\alpha := \gamma/\alpha$,
$d := d + \alpha p$, $\hat{d} := \hat{d} + \alpha\hat{p}$,
$r := r - \alpha q$, $\hat{r} := \hat{r} - \alpha\hat{q}$,
$\beta := 1/\gamma$

end while.

Matrix $(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}$ used in the above algorithm is not computed, but the sparse Choleski decomposition (complete or incomplete) is used instead. Unfortunately, matrix $\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M}$ can be dense if $\hat{A}$ has dense rows. To eliminate this situation, we assume (without loss of generality) that $\hat{A}^T = [\hat{A}_s^T, \hat{A}_d^T]$ and $\hat{D} = \mathrm{diag}(\hat{D}_s, \hat{D}_d)$, where $\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M}$ is sparse and $\hat{A}_d$ consists of dense rows. Then

$$\begin{aligned} (\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1} &= (\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M} + \hat{A}_d^T\hat{D}_d^{-1}\hat{A}_d)^{-1} \\ &= (\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M})^{-1} \\ &- (\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M})^{-1}\hat{A}_d^T\hat{M}_d^{-1}\hat{A}_d(\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M})^{-1} \end{aligned}$$

(by the Woodbury theorem), where

$$\hat{M}_d = \hat{D}_d + \hat{A}_d(\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M})^{-1}\hat{A}_d^T$$

is a (low-dimensional) dense matrix. Again the sparse Choleski decomposition is used instead of $(A_s^T D_s^{-1} A_s + \hat{M})^{-1}$. Notice that this approach is not quite reliable since

matrix $\hat{A}_s^T \hat{D}_s^{-1} \hat{A}_s + \hat{M}$ can be singular even if $\hat{A}^T \hat{D}^{-1} \hat{A} + \hat{M}$ is nonsingular. In this case, the above way cannot be used, since the required inversion does not exist. However, we can use the Bunch-Parlett decomposition of matrix C alternatively to compute $C^{-1}\bar{r}$.

2.2 Linear dependence of gradients of active constraints

Motivating by Tikhonov regularization [22], we use a perturbation of $\hat{M}$ to eliminate singularity (or near singularity) of matrix $\hat{A}^T \hat{D}^{-1} \hat{A} + \hat{M}$. Therefore, we solve equation

$$K(\varepsilon)\bar{d}(\varepsilon) = \begin{bmatrix} \hat{G} & \hat{A} \\ \hat{A}^T & -(\hat{M} + \varepsilon \hat{E}) \end{bmatrix} \begin{bmatrix} d(\varepsilon) \\ \hat{d}(\varepsilon) \end{bmatrix} = \begin{bmatrix} b \\ \hat{b} \end{bmatrix} = \bar{b} \tag{2.14}$$

and use preconditioner

$$C(\varepsilon) = \begin{bmatrix} \hat{D} & \hat{A} \\ \hat{A}^T & -(\hat{M} + \varepsilon \hat{E}) \end{bmatrix}. \tag{2.15}$$

where $\hat{E}$ is a positive semidefinite diagonal matrix (e.g., $\hat{E} = I$) and $\varepsilon > 0$ is a small number. Note that a near singularity of matrix $\hat{A}^T \hat{D}^{-1} \hat{A} + \hat{M}$ is caused by a near linear dependence of gradients of active constraints. In this case, vector $\hat{d}$ obtained from (2.12) tends to infinity and the interior-point method usually fails. The following theorem shows that the regularization can eliminate this phenomenon.

Theorem 4. *Consider system (2.14) with nonsingular $\hat{G}$. Then*

$$\frac{1}{2} \frac{\mathrm{d}(\hat{d}^T(\varepsilon) \hat{E} \hat{d}(\varepsilon))}{\mathrm{d}\varepsilon} = -\hat{d}^T(\varepsilon) \hat{E} (\hat{A}^T \hat{G}^{-1} \hat{A} + \hat{M} + \varepsilon \hat{E})^{-1} \hat{E} \hat{d}(\varepsilon). \tag{2.16}$$

If there is a number $\overline{\varepsilon} \geq 0$ such that $\hat{A}^T \hat{G}^{-1} \hat{A} + \hat{M} + \varepsilon \hat{E}$ is positive definite $\forall \varepsilon \geq \overline{\varepsilon}$, then (2.16) is negative (if $\hat{d}(\varepsilon) \neq 0$) $\forall \varepsilon \geq \overline{\varepsilon}$ and $\hat{d}^T(\varepsilon) \hat{E} \hat{d}(\varepsilon) \to 0$ if $\varepsilon \to \infty$.

Proof. Differentiating (2.14) by ε, we obtain

$$K(\varepsilon)\bar{d}'(\varepsilon) + K'(\varepsilon)\bar{d}(\varepsilon) = 0,$$

which gives

$$\begin{bmatrix} \hat{G} & \hat{A} \\ \hat{A}^T & -(\hat{M} + \varepsilon \hat{E}) \end{bmatrix} \begin{bmatrix} d'(\varepsilon) \\ \hat{d}'(\varepsilon) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \hat{E} \end{bmatrix} \begin{bmatrix} d(\varepsilon) \\ \hat{d}(\varepsilon) \end{bmatrix}.$$

Using partial elimination, one has

$$d'(\varepsilon) = -\hat{G}^{-1} \hat{A} \hat{d}'(\varepsilon)$$

and

$$\hat{d}'(\varepsilon) = -(\hat{A}^T \hat{G}^{-1} \hat{A} + \hat{M} + \varepsilon \hat{E})^{-1} \hat{E} \hat{d}(\varepsilon).$$

Now

$$\frac{1}{2} \frac{\mathrm{d}(\hat{d}^T(\varepsilon) \hat{E} \hat{d}(\varepsilon))}{\mathrm{d}\varepsilon} = \hat{d}^T(\varepsilon) \hat{E} \hat{d}'(\varepsilon),$$

so (2.16) follows from the previous equality. If there is a number $\overline{\varepsilon} \geq 0$ such that $\hat{A}^T \hat{G}^{-1} \hat{A} + \hat{M} + \varepsilon \hat{E}$ is positive definite $\forall \varepsilon \geq \overline{\varepsilon}$, then

$$v^T (\hat{A}^T \hat{G}^{-1} \hat{A} + \hat{M} + \varepsilon \hat{E}) v \geq v^T (\hat{A}^T \hat{G}^{-1} \hat{A} + \hat{M} + \overline{\varepsilon} \hat{E}) v \geq \underline{\lambda} v^T v > 0 \tag{2.17}$$

$\forall v \neq 0$, $\forall \varepsilon \geq \overline{\varepsilon}$, due to positive semidefiniteness of $\hat{E}$ ($\underline{\lambda}$ is the minimum eigenvalue of positive definite matrix $\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M} + \overline{\varepsilon}\hat{E}$). Thus (2.16) is negative (if $\hat{d}(\varepsilon) \neq 0$) $\forall \varepsilon \geq \overline{\varepsilon}$. Furthermore, using partial elimination of (2.14), we obtain

$$d(\varepsilon) = -\hat{G}^{-1}\hat{A}\hat{d}(\varepsilon) - \hat{G}^{-1}b,$$

$$(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M} + \varepsilon\hat{E})\hat{d}(\varepsilon) = \hat{A}^T\hat{G}^{-1}b - \hat{b}$$

and (2.17) implies that $\|\hat{d}(\varepsilon)\| \leq \|\hat{A}^T\hat{G}^{-1}b - \hat{b}\|/\underline{\lambda}$ $\forall \varepsilon \geq \overline{\varepsilon}$. If $\varepsilon \to \infty$ then

$$\begin{aligned} \hat{d}^T(\varepsilon)\hat{E}\hat{d}(\varepsilon) &= \frac{1}{\varepsilon}\left(\hat{d}^T(\varepsilon)(\hat{A}^T\hat{G}^{-1}b - \hat{b}) - \hat{d}^T(\varepsilon)(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{d}(\varepsilon)\right) \\ &\leq \frac{1}{\varepsilon}\frac{\|\hat{A}^T\hat{G}^{-1}b - \hat{b}\|^2}{\underline{\lambda}}\left(1 + \frac{\|\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M}\|}{\underline{\lambda}}\right) \to 0. \end{aligned}$$

□

Singularity (or near singularity) of matrix $\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M}$ is usually detected during the Choleski decomposition. If the Gill-Murray modification [13] of the Choleski decomposition is used, then a suitable matrix $\hat{E}$ is obtained as a by-product. Note that the regularization described above deteriorates properties of preconditioner (2.15). If $\hat{E} = \mathrm{diag}(\hat{E}_I, E_E)$ where E_E is nonsingular, then the situation is the same as in case all constraints are inequalities. Thus the Krylov subspace has a dimension of at most $n + 1$ and using Krylov-subspace method we obtain the solution after at most $n + 1$ iterations.

The regularization described above can affect direction vector not negligibly. Therefore, we have to include this effect to the definition of the merit function as is shown in Section 3. For this purpose, we define $E_I = \mathrm{diag}(\hat{E}_I, \check{E}_I)$, where $\check{E}_I = 0$.

2.3 Additional indefinite preconditioners

There are additional indefinite preconditioners for system (2.12) based on the Choleski or the Bunch-Parlett decompositions [10] of a sparse matrix derived from $\hat{G}$. These preconditioners are especially advantageous for convex quadratic programming problems where matrix $\hat{G}$ is positive definite. Preconditioner

$$C = \begin{bmatrix} \hat{B} & \hat{A} \\ \hat{A}^T & \hat{A}^T\hat{B}^{-1}\hat{A} - (\hat{M} + \hat{D}) \end{bmatrix}, \tag{2.18}$$

where $\hat{B}$ is a nonsingular approximation of $\hat{G}$ (usually $\hat{B} = \hat{G}$) and $\hat{D}$ is a diagonal matrix such that $\hat{M} + \hat{D}$ is positive definite, is introduced in [11], [16], [21] ($\hat{D}$ should be as close as possible to matrix $\hat{A}^T\hat{B}^{-1}\hat{A}$). In this case,

$$C^{-1} = \begin{bmatrix} \hat{B}^{-1} - \hat{B}^{-1}\hat{A}\hat{C}^{-1}\hat{A}^T\hat{B}^{-1} & \hat{B}^{-1}\hat{A}\hat{C}^{-1} \\ \hat{C}^{-1}\hat{A}^T\hat{B}^{-1} & -\hat{C}^{-1} \end{bmatrix}, \tag{2.19}$$

where $\hat{C} = \hat{M} + \hat{D}$ (recall that $\hat{C}$ is positive definite) and if $\hat{B} = \hat{G}$, then

$$KC^{-1} = \begin{bmatrix} I & 0 \\ (I - \hat{H})\hat{A}^T\hat{G}^{-1} & \hat{H} \end{bmatrix}, \tag{2.20}$$

where $\hat{H} = (\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{C}^{-1}$. Instead of $\hat{B}^{-1}$, we use the Choleski or the Bunch-Parlett decomposition of matrix $\hat{B} \approx \hat{G}$, which is usually sparse.

Preconditioner

$$C = \begin{bmatrix} \hat{B} & \hat{A} \\ \hat{A}^T & -(\hat{M} + \hat{D}) \end{bmatrix}, \tag{2.21}$$

where $\hat{B}$ is an approximation of $\hat{G}$ such that $\hat{B}+\hat{A}(\hat{M}+\hat{D})^{-1}\hat{A}^T$ is nonsingular (usually $\hat{B} = \hat{G}$) and $\hat{D}$ is a diagonal matrix such that $\hat{M} + \hat{D}$ is positive definite, is introduced in [2], [16]. ($\hat{D}$ should be as small as possible). In this case,

$$C^{-1} = \begin{bmatrix} \hat{B}^{-1} - \hat{B}^{-1}\hat{A}\hat{C}^{-1}\hat{A}^T\hat{B}^{-1} & \hat{B}^{-1}\hat{A}\hat{C}^{-1} \\ \hat{C}^{-1}\hat{A}^T\hat{B}^{-1} & -\hat{C}^{-1} \end{bmatrix}, \tag{2.22}$$

where $\hat{C} = \hat{A}^T\hat{B}^{-1}\hat{A}+\hat{M}+\hat{D}$ (we assume that $\hat{B}$ and $\hat{C}$ are nonsingular) and if $\hat{B} = \hat{G}$, then

$$KC^{-1} = \begin{bmatrix} I & 0 \\ (I - \hat{H})\hat{A}^T\hat{G}^{-1} & \hat{H} \end{bmatrix}, \tag{2.23}$$

where $\hat{H} = (\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{C}^{-1}$.

Formula (2.22) is not suitable for computation, since it requires the Choleski or the Bunch-Parlett decomposition of matrix $\hat{C}$, which is usually dense. Therefore, we use the following alternative formula

$$C^{-1} = \begin{bmatrix} \tilde{B}^{-1} & \tilde{B}^{-1}\hat{A}(\hat{M} + \hat{D})^{-1} \\ (\hat{M} + \hat{D})^{-1}\hat{A}^T\tilde{B}^{-1} & (\hat{M} + \hat{D})^{-1}\hat{A}^T\tilde{B}^{-1}\hat{A}(\hat{M} + \hat{D})^{-1} - (\hat{M} + \hat{D})^{-1} \end{bmatrix}, \tag{2.24}$$

where $\tilde{B} = \hat{B} + \hat{A}(\hat{M} + \hat{D})^{-1}\hat{A}^T$. Instead of $\tilde{B}^{-1}$, we use the Choleski or the Bunch-Parlett decomposition of matrix $\tilde{B}$, which is usually sparse (it is dense when $\hat{A}$ has dense columns). Note that nonsingularity of $\hat{B}$ and $\hat{C}$ is not required in (2.24). Nevertheless, matrix $\tilde{B}$ has to be nonsingular. Note also that we can set

$$\hat{H} = \hat{A}^T\tilde{B}^{-1}\hat{A}(\hat{M} + \hat{D})^{-1} \tag{2.25}$$

in (2.23).

Preconditioner (2.21) is based on the regularization. If $\hat{G}$ and $\hat{M} + \varepsilon\hat{E}$ are "sufficiently" nonsingular, we can set $\hat{B} = \hat{G}$ and $\hat{D} = \varepsilon\hat{E}$ so $C(\varepsilon) = K(\varepsilon)$ and the solution of (2.14) is found in the first iteration. This situation also arises when $\varepsilon = 0$ and $m_E = 0$. If $\hat{G}$ and $\hat{M}$ are "sufficiently" nonsingular, we can set $\hat{B} = \hat{G}$ and $\hat{D} = 0$ so $C = K$ and the solution of (2.12) is found in the first iteration.

The following theorems, which are analogies of Theorem 1, Theorem 2, Theorem 3, hold for preconditioners (2.18) and (2.21) with $\hat{B} = \hat{G}$ (we assume that all inversions exist).

Theorem 5. *Consider preconditioner (2.18) (or (2.21)) with $\hat{B} = \hat{G}$ and assume that $\hat{M} + \hat{D}$ is positive definite. Then the matrix KC^{-1} has at least n unit eigenvalues and a full system of n linearly independent eigenvectors corresponding to these eigenvalues exists. The other eigenvalues of matrix KC^{-1} are exactly eigenvalues of matrix*

$(\hat{A}^T\hat{G}^{-1}\hat{A}+\hat{M})\hat{C}^{-1}$, *where* $\hat{C}=\hat{M}+\hat{D}$ *for (2.18)(or* $\hat{C}=\hat{A}^T\hat{B}^{-1}\hat{A}+\hat{M}+\hat{D}$ *for (2.21)). If* $\hat{A}^T\hat{G}^{-1}\hat{A}+\hat{M}$ *is positive definite then all eigenvalues are positive.*

Proof. The proof uses expression (2.20) (or (2.23)) and is introduced in [16] (see also [11]). □

Theorem 6. *Consider preconditioner (2.18) (or (2.21)) with* $\hat{B}=\hat{G}$ *applied to system (2.12). Then Krylov subspace* $\mathcal{K}=\mathrm{span}\{\bar{r},KC^{-1}\bar{r},(KC^{-1})^2\bar{r},\ldots\}$, *where* $\bar{r}\in R^{n+\hat{m}}$, *has a dimension of at most* $m+1$.

Proof. Matrix $\hat{H}=(\hat{A}^T\hat{G}^{-1}\hat{A}+\hat{M})\hat{C}^{-1}$ has a full system of linearly independent eigenvectors, since $\hat{A}^T\hat{G}^{-1}\hat{A}+\hat{M}$ is symmetric and $\hat{C}$ is positive definite. Thus KC^{-1} has a full system of linearly independent eigenvectors by Theorem 5. By the same theorem, KC^{-1} has at most $m+1$ different eigenvalues so the Krylov subspace $\mathcal{K}$ has a dimension of at most $m+1$. □

Lemma 2. *Consider Algorithm PCG with preconditioner (2.18) (or (2.21)) with* $\hat{B}=\hat{G}$ *applied to system (2.12). Assume that initial* $\bar{d}$ *is chosen in such a way that* $r=0$ *at the start of the algorithm. Let matrix* $\hat{A}^T\hat{G}^{-1}\hat{A}+\hat{M}$ *be positive definite. Then* $r=0$ *and*

$$\begin{aligned}
\hat{t} &:= -\hat{C}^{-1}\hat{r},\\
\bar{r}^T\bar{t} &= \hat{r}^T\hat{t},\\
\hat{p} &:= \hat{t}+\beta\hat{p},\\
\hat{q} &:= -(\hat{A}^T\hat{G}^{-1}\hat{A}+\hat{M})\hat{p},\\
\bar{p}^T\bar{q} &= \hat{p}^T\hat{q},\\
\hat{d} &:= \hat{d}+\alpha\hat{p},\\
\hat{r} &:= \hat{r}-\alpha\hat{q},
\end{aligned}$$

in all iterations of Algorithm PCG, where $\hat{C}=\hat{M}+\hat{D}$ *for (2.18) (or* $\hat{C}=\hat{A}^T\hat{G}^{-1}\hat{A}+\hat{M}+\hat{D}$ *for (2.21)).*

Proof. The proof for (2.18) is given in [11]. Consider preconditioner (2.21). Although the proof is carried out by induction, we omit iterative indices and use assignments ":=" to simplify the notation. The inductive assumptions are $r=0$ and $p=-\hat{G}^{-1}\hat{A}\hat{p}$ (we can set $p=-\hat{G}^{-1}\hat{A}\hat{p}$ at the start of the algorithm, since $\beta=0$ in the first iteration). Since $r=0$ by the assumption, we can write

$$\begin{bmatrix} t\\ \hat{t}\end{bmatrix}=\begin{bmatrix}\hat{G}^{-1}\hat{A}\hat{C}^{-1}\hat{r}\\ -\hat{C}^{-1}\hat{r}\end{bmatrix}=-\begin{bmatrix}\hat{G}^{-1}\hat{A}\hat{t}\\ \hat{C}^{-1}\hat{r}\end{bmatrix}$$

by (2.22). Obviously, $\bar{r}^T\bar{t}=\hat{r}^T\hat{t}$. Since $p=-\hat{G}^{-1}\hat{A}\hat{p}$ by the assumption, we can write

$$\begin{aligned}
p &:= t+\beta p=-\hat{G}^{-1}\hat{A}(\hat{t}+\beta\hat{p})\\
\hat{p} &:= \hat{t}+\beta\hat{p}.
\end{aligned}$$

Thus again $p = -\hat{G}^{-1}\hat{A}\hat{p}$. Now

$$\begin{bmatrix} q \\ \hat{q} \end{bmatrix} = \begin{bmatrix} \hat{G} & \hat{A} \\ \hat{A}^T & -\hat{M} \end{bmatrix} \begin{bmatrix} -\hat{G}^{-1}\hat{A}\hat{p} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} 0 \\ -(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{p} \end{bmatrix},$$

which gives $\bar{p}^T\bar{q} = \hat{p}^T\hat{q}$. Finally

$$\begin{aligned} \hat{d} &:= \hat{d} + \alpha\hat{p}, \\ \hat{r} &:= \hat{r} - \alpha\hat{q} = 0. \end{aligned}$$

□

Theorem 7. *Consider Algorithm PCG with preconditioner (2.18) (or (2.21)) with $\hat{B} = \hat{G}$ applied to system (2.12). Assume that initial $\bar{d}$ is chosen in such a way that $r = 0$ at the start of the algorithm. Let matrix $\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M}$ be positive definite. Then:*

(a) Vector $\bar{d}^*$ which solves equation $K\bar{d} = \bar{b}$ is found after m iterations at most.

(b) The algorithm cannot break down before $\bar{d}^*$ is found.

(c) Error $\|\hat{d} - \hat{d}^*\|$ converges to zero at least R - linearly with quotient

$$\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1},$$

where κ is the spectral condition number of matrix $(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{C}^{-1}$,where $\hat{C} = \hat{M} + \hat{D}$ for (2.18) (or $\hat{C} = \hat{A}^T\hat{B}^{-1}\hat{A} + \hat{M} + \hat{D}$ for (2.21)).

Proof. The proof for (2.18) is given in [11]. Consider preconditioner (2.21).
(a) Lemma 2 implies that if system (2.12) is solved by conjugate gradient method with preconditioner (2.21), then $\hat{d}$ are generated by conjugate gradient method with preconditioner $\hat{C}$ applied to system $(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{d} = -\hat{b}$. Since $\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M}$ and $\hat{C}$ are positive definite, we obtain the solution $\hat{d}^*$ after at most m iterations. Since $r^* = 0$, one has $d^* = -\hat{G}^{-1}\hat{A}\hat{d}^*$. Therefore, $\hat{A}^Td^* - \hat{M}\hat{d}^* = \hat{b}$ is equivalent to $(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{d}^* = -\hat{b}$ and since $\hat{d}^*$ solves this system we have $\hat{r}^* = 0$. Thus $\bar{d} = \bar{d}^*$ if $\hat{d} = \hat{d}^*$.
(b) Since denominators used in the conjugate gradient method with preconditioner (2.21) applied to system (2.12) are the same (with exception of the sign) as denominators used in the conjugate gradient method with preconditioner $\hat{C}$ applied to system $(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{d} = -\hat{b}$ and since matrices $\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M}$ and $\hat{C}$ are positive definite, the algorithm cannot break down before $\hat{d}^*$ and, therefore, $\bar{d}^*$ is found.
(c) We can use standard estimation for the conjugate gradient method with preconditioner $\hat{C}$ applied to system $(\hat{A}^T\hat{G}^{-1}\hat{A} + \hat{M})\hat{d} = -\hat{b}$. □

3 Step-length selection

Step-length $0 < \alpha < \overline{\alpha}$ and new vectors $x := x + \alpha\Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta u_I)$, $u_E := u_E + \alpha\Delta u_E$ can be determined in many ways. In all these ways, it is necessary to satisfy condition $s_I > 0$ (to make it possible to define a logarithmic barrier function) and $u_I > 0$ (which is motivated by (1.6) and which guarantee positive definiteness of matrix $M_I = U_I^{-1}S_I$). We have used three simple strategies for computation of $s_I(\alpha, \Delta s_I)$ and $u_I(\alpha, \Delta u_I)$. Strategy 1 uses maximum step-length $\overline{\alpha} = \min(1, \overline{\Delta}/\|\Delta x\|)$ ($\overline{\Delta}$ serves as a safeguard and has a similar significance as a trust-region radius) and handles individual components separately so that $s_i(\alpha, \Delta s_I) = s_i + \alpha_{s_i}\Delta s_i$ and $u_i(\alpha, \Delta u_I) = u_i + \alpha_{u_i}\Delta u_i$, $i \in I$, where

$$\begin{aligned}
\alpha_{s_i} &= \alpha, & \Delta s_i \geq 0,\\
\alpha_{s_i} &= \min\left(\alpha, -\gamma\frac{s_i}{\Delta s_i}\right), & \Delta s_i < 0,\\
\alpha_{u_i} &= \alpha, & \Delta u_i \geq 0,\\
\alpha_{u_i} &= \min\left(\alpha, -\gamma\frac{u_i}{\Delta u_i}\right), & \Delta u_i < 0,
\end{aligned}$$

and $0 < \gamma < 1$ is a coefficient close to unit. Other strategies require bounds

$$\begin{aligned}
\overline{\alpha}_s &= \gamma \min_{i\in I, \Delta s_i<0}\left(-\frac{s_i}{\Delta s_i}\right),\\
\overline{\alpha}_u &= \gamma \min_{i\in I, \Delta u_i<0}\left(-\frac{u_i}{\Delta u_i}\right),
\end{aligned}$$

where $0 < \gamma < 1$ is a coefficient close to unit, and define $s_I(\alpha, \Delta s_I) = s_I + \min(\alpha, \overline{\alpha}_s)\Delta s_I$, $u_I(\alpha, \Delta u_I) = u_I + \min(\alpha, \overline{\alpha}_u)\Delta u_I$. Strategy 2 uses upper bound $\overline{\alpha} = \min(1, \overline{\Delta}/\|\Delta x\|)$. Strategy 3 corresponds to the choice $\overline{\alpha} = \min(1, \overline{\alpha}_s, \overline{\Delta}/\|\Delta x\|)$, which gives $s_I(\alpha, \Delta s_I) = s_I + \alpha\Delta s_I$ for $0 < \alpha \leq \overline{\alpha}$.

A further requirement for the selection of a step-length is the satisfying of a suitable goal criterion. This criterion is usually a merit function which is a combination of the barrier function and a measure of constraint violation. Motivated by [16], we use the following function (which includes the effect of the regularization described in Section 2)

$$\begin{aligned}
P(\alpha) &= f(x + \alpha\Delta x) - \mu e^T \ln(S_I(\alpha, \Delta s_I))e\\
&+ (u_I + \Delta u_I)^T(c_I(x + \alpha\Delta x) + s_I(\alpha, \Delta s_I))\\
&+ (u_E + \Delta u_E)^T c_E(x + \alpha\Delta x)\\
&+ \frac{\sigma}{2}\|c_I(x + \alpha\Delta x) + s_I(\alpha, \Delta s_I) - \varepsilon E_I(u_I(\alpha, \Delta u_I) - u_I)\|^2\\
&+ \frac{\sigma}{2}\|c_E(x + \alpha\Delta x) - \varepsilon E_E\alpha\Delta u_E\|^2, \qquad (3.1)
\end{aligned}$$

where $\sigma \geq 0$. The following theorem holds.

Theorem 8. *Let $s_I > 0$, $u_I > 0$ and let the triple Δx, $\Delta\hat{u}_I$, Δu_E be an inexact*

solution of a regularized system so that

$$\begin{bmatrix} \hat{G} & \hat{A}_I & A_E \\ \hat{A}_I^T & -\hat{U}_I^{-1}\hat{S}_I - \varepsilon \hat{E}_I & 0 \\ A_E^T & 0 & -\varepsilon E_E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \hat{u}_I \\ \Delta u_E \end{bmatrix} + \begin{bmatrix} \hat{g} \\ \hat{c}_I + \mu \hat{U}_I^{-1} e \\ c_E \end{bmatrix} = \begin{bmatrix} r \\ \hat{r}_I \\ r_E \end{bmatrix}, \tag{3.2}$$

where r, $\hat{r}_I$, r_E are parts of the residual vector and let $\Delta \check{u}_I$ and Δs_I be given by (2.3) and (2.1), respectively. Then

$$\begin{aligned} P'(0) &= -(\Delta x)^T G \Delta x - (\Delta s_I)^T S_I^{-1} U_I \Delta s_I - \sigma(\|c_I + s_I\|^2 + \|c_E\|^2) \\ &+ (\Delta x)^T r + \sigma((\hat{c}_I + \hat{s}_I)^T \hat{r}_I + c_E^T r_E). \end{aligned} \tag{3.3}$$

If

$$\sigma > -\frac{(\Delta x^T) G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I}{\|c_I + s_I\|^2 + \|c_E\|^2}, \tag{3.4}$$

and if (2.4) is solved with a sufficient precision, namely if

$$\begin{aligned} (\Delta x)^T r + \sigma((\hat{c}_I + \hat{s}_I)^T \hat{r}_I + c_E^T r_E) &< (\Delta x)^T G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I \\ &+ \sigma(\|c_I + s_I\|^2 + \|c_E\|^2), \end{aligned} \tag{3.5}$$

then $P'(0) < 0$.

Proof. Since $s_I > 0$ and $u_I > 0$, one has $s_I(\alpha, \Delta s_I) = s_I + \alpha \Delta s_I$ and $u_I(\alpha, \Delta u_I) = u_I + \alpha \Delta u_I$ for sufficiently small values of α. Thus differentiating (3.1) by α, we obtain

$$\begin{aligned} P'(0) &= (\Delta x)^T (\nabla f(x) + A_I(u_I + \Delta u_I) + A_E(u_E + \Delta u_E)) \\ &- \mu (\Delta s_I)^T S_I^{-1} e + (\Delta s_I)^T (u_I + \Delta u_I) \\ &+ \sigma (c_I + s_I)^T (A_I^T \Delta x + \Delta s_I - \varepsilon E_I \Delta u_I) \\ &+ \sigma c_E^T (A_E^T \Delta x - \varepsilon E_E \Delta u_E). \end{aligned} \tag{3.6}$$

Using the equality

$$\begin{bmatrix} G & 0 & A_I & A_E \\ 0 & U_I & S_I & 0 \\ A_I^T & I & -\varepsilon E_I & 0 \\ A_E^T & 0 & 0 & -\varepsilon E_E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s_I \\ \Delta u_I \\ \Delta u_E \end{bmatrix} + \begin{bmatrix} g \\ S_I U_I e - \mu e \\ c_I + s_I \\ c_E \end{bmatrix} = \begin{bmatrix} r \\ 0 \\ r_I \\ r_E \end{bmatrix},$$

which is equivalent to (2.1), (2.3) and (3.2), we obtain

$$(\Delta x)^T (\nabla f(x) + A_I(u_I + \Delta u_I) + A_E(u_E + \Delta u_E)) = -(\Delta x)^T G \Delta x + (\Delta x)^T r,$$

$$(\Delta s_I)^T (u_I + \Delta u_I) - \mu (\Delta s_I)^T S_I^{-1} e = -(\Delta s_I)^T S_I^{-1} U_I \Delta s_I,$$

$$(c_I + s_I)^T (A_I^T \Delta x + \Delta s_I - \varepsilon E_I \Delta u_I) = -\|c_I + s_I\|^2 + (\hat{c}_I + \hat{s}_I)^T \hat{r}_I,$$

$$c_E^T (A_E^T \Delta x - \varepsilon E_E \Delta u_E) = -\|c_E\|^2 + c_E^T r_E$$

(since $\check{r}_I = 0$ by (2.3)), which after substituting into (3.6) gives (3.3). If (3.4) holds, then the right-hand side in (3.5) is positive so if (2.4) is solved with a sufficient precision, then (3.5) holds and $P'(0) < 0$ by (3.3). □

Condition (3.4) restricts the choice of parameter σ weakly. If matrix G is positive semidefinite, any value $\sigma \geq 0$ satisfies this condition. In the opposite case, the second term, which is always positive, decreases the value of $P'(0)$ and partially eliminates the influence of the first term.

Theorem 8 gives one possibility for the computation of parameter σ, which implies the inequality $P'(0) < 0$. But it is usually more efficient for practical computation to choose parameter σ as a constant and replace matrix G by a positive definite diagonal matrix if condition $P'(0) < 0$ does not hold, see [18].

If $P'(0) < 0$, we can use a line-search technique. In this case, we set $\alpha = \beta^l\overline{\alpha}$, where $0 < \beta < 1$ is a line search parameter and $l \geq 0$ is a minimum nonnegative integer such that $P(\beta^l\overline{\alpha}) < P(0)$. After determination of $\alpha \leq \overline{\alpha}$, we set $x := x + \alpha\Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta u_I)$, $u_E := u_E + \alpha\Delta u_E$.

The above line-search technique is not always advantageous, since step-length $\alpha = \beta^l\overline{\alpha}$ can be too short, especially if Strategy 3 is used. Therefore, we have tested two additional possibilities. First, we have excluded barrier term $\mu e^T \ln(S_I(\alpha, \Delta s_I))e$ from (3.1), since our strategies guarantee that $s_I(\alpha, \Delta s_I) > 0$ and $u_I(\alpha, \Delta u_I) > 0$. Secondly, we have used the simple choice $\alpha = \overline{\alpha}$ (the first step accepted). In this case, merit function (3.1) serves only to indication of restarts. Surprisingly, the simple choice is very efficient as it is demonstrated in Table 1a–Table 1c.

4 Computation of the barrier parameter

Although computation of the barrier parameter is a crucial part of the interior-point method, we do not bring new results here. We only recapitulate ideas, which were used in our implementation.

If we solved problem (IP) exactly and changed the value of parameter μ consequently, the total number of iterations would be rather large. Therefore, the value of parameter μ is recomputed in every iteration. Most implementations of interior-point methods choose the value μ in such a way that $0 < \mu < s_I^T u_I/m_I$ (or $\mu = \lambda s_I^T u_I/m_I$, where $0 < \lambda < 1$). This case is analyzed in [12] and used in [24]. Computational experience has shown that the algorithm perform best when components $s_i u_i$ of the dot-product in numerator approach zero at a uniform rate. The distance from uniformity can be measured by the ratio

$$\varrho = \frac{\min_{i\in I}(s_i u_i)}{s_I^T u_I/m_I}$$

(also called the centrality measure). Clearly, $0 < \varrho \leq 1$ and $\varrho = 1$ if and only if the condition (1.6) holds. The value λ is then computed by using ϱ. Usually heuristic formulas are used for this purpose. In our implementation, we have used the formula

$$\lambda = 0.1 \min\left(0.05\frac{1-\varrho}{\varrho}, 2\right)^3 \tag{4.1}$$

proposed in [24]. We have also tested other possibilities, e.g., formulas given in [1], but formula (4.1) has shown to be best.

Concerning the local convergence analysis of interior-point methods with various choices of the barrier parameter we refer to [6] and [12]. It is necessary to note that slow decrease of μ can lead to a considerable increase of the total number of iterations, i.e., to a long computational time, but its rapid decrease can lead to the method failure.

5 Description of the algorithm

The above considerations can be summarized in the algorithmic form.

Algorithm 1.

Data: Parameter for the active constraint definition ε_I (e.g. $\varepsilon_I = 0.1$). Minimum precision for the direction determination $0 < \overline{\omega} < 1$ (e.g. $\overline{\omega} = 0.9$). Line-search parameter $0 < \beta < 1$ (e.g. $\beta = 0.5$). Maximum step-length reduction $0 < \gamma < 1$ (e.g. $\gamma = 0.95$ when barrier function (3.1) is used and $\gamma = 0.99$ otherwise). Step bound $\overline{\Delta} > 0$ (e.g. $\overline{\Delta} = 1000$).

Input: Sparsity pattern of matrices $\nabla^2 F$ and A. Initial choice of vector x.

Step 1: *Initiation.* Choose the values $\mu > 0$ (e.g. $\mu = 1$) and $\sigma > 0$ (e.g. $\sigma = 1$). For $i \in I$ set $s_i := \max(-c_i(x), \delta_s)$ and $u_i := \delta_u$, where $\delta_s > 0$ (e.g. $\delta_s = 0.1$) and $\delta_u > 0$ (e.g. $\delta_u = 0.1$). For $i \in E$ set $u_i := 0$. Compute value $f(x)$ and vectors $c_I(x)$, $c_E(x)$. Set $k := 0$.

Step 2: *Termination.* Compute matrix $A := A(x)$ and vector $g := g(x, u)$. If KKT conditions (1.5) - (1.8) with μ sufficiently small are satisfied with a sufficient precision, then terminate the computation. Otherwise set $k := k + 1$.

Step 3: *Approximation of the Hessian matrix.* Compute approximation G of the Hessian matrix $G(x, u)$ by using differences of gradient $g(x, u)$ as in [8].

Step 4: *Direction determination.* Split constraints into active and inactive and build. linear system (2.4). Determine positive definite diagonal matrix $\hat{D}$ as an approximation of the diagonal of $\hat{G}$ and determine a representation of the preconditioner (2.8) (Bunch-Parlett decomposition or Schur complement based representation, see [18]). Writing system (2.4) in the form $K\bar{d} = \bar{b}$, set $\omega = \min(\|\bar{b}\|, 1/k, \overline{\omega})$ and determine direction vectors Δx, $\Delta \hat{u}_I$ and Δu_E as an inexact solution of (2.4) (with the precision $\|K\bar{d} - \bar{b}\| \leq \omega \|\bar{b}\|$) by the preconditioned Krylov-subspace method. Compute vectors $\Delta \check{u}_I$, $\Delta \hat{s}_I$, $\Delta \check{s}_I$ by (2.3), (2.5), (2.6). Compute directional derivative $P'(0)$ of the merit function $P(\alpha)$ by (3.6)

Step 5: *Restart.* If $P'(0) \geq 0$, determine positive definite diagonal matrix D by the procedure given in [18], set $G = D$ and go to Step 4.

Step 6: *Step-length selection.* Define maximum step-length $\overline{\alpha}$ and functions $s_I(\alpha, \Delta s_I)$, $u_I(\alpha, \Delta u_I)$ by one of the strategies described in Section 4. Find the minimum integer $l \geq 0$ such that $P(\beta^l \overline{\alpha}) < P(0)$ (line-search) or set $l = 0$ (first step). Set $\alpha = \beta^l \overline{\alpha}$ and $x := x + \alpha \Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta s_I)$, $u_E := u_E + \alpha \Delta u_E$. Compute value $f(x)$ and vectors $c_I(x)$ and $c_E(x)$.

Step 7: *Barrier parameter.* Determine parameter λ by (4.1), set $\mu = \lambda s_I^T u_I / m_I$ and go to Step 2.

Note that this algorithm does not contain regularization described in Section 2.2 and additional preconditioners described in Section 2.3.

6 Numerical experiments

Algorithm 1 was tested by using five sets each containing 17 test problems. These sets were obtained as a modification of test problems for equality constrained minimization given in [16] and [17], which can be downloaded (together with report [17]) from `http://www.cs.cas.cz/~luksan/test.html` (we excluded Problem 5.8 from our tests, since it consumed more than 50% of the total CPU time). In the first set, equalities $c(x) = 0$ were replaced by inequalities $c(x) \geq 0$. In the second set, equalities $c(x) = 0$ were replaced by inequalities $c(x) \leq 0$ (i.e., this set contains problems LUKVLI1–LUKVLI18 from the cute collection [5]). The third set was generated from the first set by adding box constraints $x \geq 0$. The fourth set was generated from the second set by adding box constraints $x \leq 0$. The fifth set contains inequalities $-1 \leq x \leq 1$ and $-1 \leq c(x) \leq 1$. All problems used have optional dimension; we have chosen dimension with 1000 variables. The results of the tests are listed in nine tables, where NIT is the total number of iterations, NFV is the total number of function evaluations, NFG is the total number of gradient evaluations (NFG is much greater than NIT, since second order derivatives are computed by using gradient differences), NCG is the total number of CG iterations, NRS is the total number of restarts and NFAIL gives the number of failures for a given set (the number of problems which have not been solved). Table 1a–Table 1c correspond to Algorithm 1, where line search is not used and merit function (3.1) serves only to indication of restarts.

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	618	618	4515	23974	16	9.39	-
2	393	393	2824	11817	19	5.66	-
3	737	737	5626	16383	28	10.52	-
4	361	361	2726	13623	2	12.78	2
5	571	574	4026	4086	16	10.59	1

Table 1a: The first step accepted - Strategy 1

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	609	609	4211	42288	8	11.76	-
2	451	451	3222	9738	24	6.99	-
3	650	650	4797	20229	8	10.16	-
4	415	420	2931	8143	8	8.59	2
5	620	620	4454	5210	11	12.15	-

Table 1b : The first step accepted - Strategy 2

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	594	594	4485	31452	14	11.44	-
2	747	747	53182	15681	12	9.06	-
3	1276	1278	10597	50062	36	29.47	1
4	1507	1594	11828	14708	91	20.28	1
5	730	751	5297	9549	35	16.54	1

Table 1c : The first step accepted - Strategy 3

Table 2a–Table 2c contain results obtained using line search with merit function (3.1).

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	529	727	4000	27309	12	10.74	-
2	411	842	2908	15527	90	7.68	1
3	477	714	3559	15423	28	9.60	-
4	345	478	2557	12009	4	17.10	2
5	489	516	3709	5405	37	11.88	-

Table 2a : Line-search - Strategy 1

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	629	911	4610	27946	34	10.56	-
2	497	922	3529	11919	95	7.43	1
3	617	9067	4832	23723	22	11.71	-
4	520	794	3888	10686	5	12.11	1
5	580	685	4365	10717	23	13.66	-

Table 2b: Line-search - Strategy 2

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	697	927	5461	24408	32	12.38	-
2	865	1300	6067	14726	91	10.80	1
3	1772	1882	16857	44363	48	36.42	2
4	1030	1175	8573	13682	11	18.30	2
5	1115	1313	8834	53898	249	61.64	2

Table 2c : Line-search - Strategy 3

Table 2a–Table 3c contain results obtained using line search with the barrier term excluded.

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	641	913	4725	25353	54	11.04	-
2	372	907	2689	12672	28	6.33	1
3	577	809	4407	15501	61	9.74	-
4	352	486	2668	63920	5	6.05	2
5	580	756	4077	6570	46	13.44	-

Table 3a : Barrier term excluded - Strategy 1

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	519	751	3700	16730	16	7.35	-
2	404	988	2969	11773	31	5.96	1
3	907	1462	6779	19291	37	14.33	1
4	380	580	2734	6654	4	6.88	1
5	550	742	3986	6385	30	12.51	-

Table 3b: Barrier term excluded - Strategy 2

Set	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
1	571	639	4432	18932	17	9.90	-
2	764	1281	5442	15590	30	9.86	1
3	1359	1630	11951	32862	33	25.11	2
4	1078	1241	8365	14430	13	25.11	2
5	1273	1342	9381	8523	18	22.53	2

Table 3c : Barrier term excluded - Strategy 3

For a better demonstration of efficiency of our algorithm, we performed additional tests with problems from the widely used CUTE collection [5]. Table 8 contains a list of these problems together with their dimensions. Here n is the number of variables, m is the number of nonlinear constraints, m_A is the number of nonzero elements in the matrix A and m_G is the number of nonzero elements in the matrix G. Column denoted by S refers to the strategy used. Values NIT, NFV, NFG, NCG have the same meaning as in the previous tables.

Problem	n	m	m_A	m_G	S	NIT	NFV	NFG	NCG
BRITGAS	450	360	1576	2779	1	15	15	285	132
CLNLBEAM	1503	1000	4000	6005	1	20	20	140	84
DALLASL	906	667	1812	2814	1	47	47	893	47
EG3	1001	2000	5995	4995	1	44	44	308	443
EIGENB2	420	210	8000	80620	1	8	8	3261	207
EIGENC2	462	231	9261	97923	1	17	17	7531	180
GAUSSELM	819	1296	5802	6175	3	22	22	726	1479
HANGING	1800	1150	6900	13950	1	29	29	609	795
NGONE	100	1273	4996	5050	3	35	35	3535	536
OPTCDEG2	1202	800	2800	4402	1	11	11	88	236
OPTCDEG3	1202	800	2800	4402	1	7	7	56	11
OPTMASS	1210	1005	3216	4627	1	6	6	48	26
READING1	2002	1000	4000	7003	3	35	35	245	352
READING3	2002	1001	4002	7004	3	19	19	133	540
READING4	1001	1000	2000	2001	3	134	135	540	42065
READING5	5001	5000	10000	10001	1	2	3	12	4
READING9	2002	1000	3000	5003	1	11	11	55	53
SINROSNB	1000	999	1998	1999	1	13	13	52	50
SREADIN3	1002	501	2002	3504	3	38	38	266	193
SSNLBEAM	3003	2000	8000	12005	1	19	19	133	125
SVANBERG	1000	1000	9000	9000	1	20	20	380	81
TRAINF	2008	1002	5010	10028	1	51	51	510	284
TRAINH	2008	1002	6012	13036	1	30	30	390	533
ZAMB2	1326	480	2400	6726	1	26	26	312	1645

Table 8 : The first step accepted

7 Conclusions

In this contribution, we describe an implementation of the interior-point method for solving general nonlinear programming problems. The main results are proposed in Section 2. These results are quite general and can be used in many additional applications, e.g., for solution of the Stokes or the Navier-Stokes problems after discretization. For this reason, we have studied additional preconditioners, which are very efficient when G is positive definite. In nonlinear programming, G is often indefinite or even singular (with exception of convex quadratic problems), so preconditioner (2.8) seems to be most robust in the general case. Its efficiency is demonstrated in Section 6, where it is shown that number of CG iterations (NCG) is moderate (for CUTE problems, usually 10 CG iterations was used in one IP iteration). We also propose a certain kind of regularization. More detailed investigation of this approach will be given in an independent paper. Similar approach is studied in [26], where some questions concerning the rate of convergence are answered.

The implementation represented by Algorithm 1 has several non-traditional features. First, the constraints are splitted into active and inactive sets. Equations corresponding to active constraints are solved inaccurately by the preconditioned Krylov-subspace method while the other quantities are obtained by a direct elimination. This way leads to equations which are suitable for iterative solvers. Furthermore, the Lagrange multipliers are forced to be positive. This approach implies several strategies for step-length selection. Computational experiments show that Strategy 1 is usually most efficient, but the other strategies can be successful if this strategy fails. Surprisingly, step-length selection without line search is very efficient, but it requires a more careful choice of the maximum step-length.

	Algorithm 1			Results from [7]		
Problem	n	m	NFV	n	m	NFV
CLNLBEAM	1503	1000	20	303	200	21
DALLASL	906	667	47	906	667	100
EG3	1001	2000	44	101	200	31
GAUSSELM	819	1926	22	819	1926	115
GRIDNETA	924	484	12	924	484	21
GRIDNETD	924	484	12	924	484	19
GRIDNETF	924	484	17	924	484	20
GRIDNETG	924	484	13	924	484	21
GRIDNETI	924	484	16	924	484	28
NGONE	100	1273	35	100	1273	217
OPTCDEG2	1202	800	11	302	200	30
OPTCDEG3	1202	800	7	302	200	22
OPTMASS	1210	1005	6	610	505	15
READING1	2002	1000	35	202	100	52
READING3	2002	1001	19	303	200	12
READING4	1001	1000	135	202	101	77
READING5	5001	5000	3	501	500	6
READING9	2002	1000	11	501	500	15
SINROSNB	1000	999	13	1000	999	90
SREADIN3	1002	501	38	202	101	30
SSNLBEAM	3003	2000	19	303	200	23
SVANBERG	1000	1000	20	1000	1000	18
TRAINF	2008	1002	51	808	402	345
TRAINH	2008	1002	30	808	402	441
ZAMB2	1326	480	26	1326	480	37

Table 9 : Comparison of results

Finally, Table 8 demonstrates the high efficiency of our implementation of the interior-point method. To make it more clear, Table 9 contains a comparison of Algorithm 1 with algorithm described in [7] (we use the best results from the table given in [7]).

Bibliography

[1] M.Argaez, R.A.Tapia, L.Velasquez: Numerical comparison of path-following strategies for a basic interior-point method for nonlinear programming. Report No. CRPC-TR9777-S, Center for Research on Parallel Computation, Rice University, Houston 1998.

[2] O.Axelsson: Preconditioning of indefinite problems by regularization. SIAM J. Numerical Analysis, 16, 58-69, 1979.

[3] H.Y.Benson, D.F.Shanno, J.Vanderbei: Interior-point methods for nonconvex nonlinear programming: Jamming and numerical testing. Report No. ORFE-00-02, Operations Research and Financial Engineering, Princeton University, August 2000.

[4] L.Bergamaschi, J. Gondzio, G.Zilli: Preconditioning indefinite systems in interior-point methods for optimization. Report No. MS-02-001, Department of Mathematics and Statistics, University of Edinburgh, July 2002.

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

[6] R.H.Byrd, G.Liu, J.Nocedal: On the local behavior of an interior point method for nonlinear programming. In Numerical Analysis 1997 (D.F.Griffiths and D.J.Higham, eds.), pp.37-56. Addison Wesley Longman, 1998.

[7] R.H.Byrd, J.Nocedal, R.A.Waltz: Feasible interior methods using slacks for nonlinear optimization. Report No. OTC 2000/11, Optimization Technology Center, December 2000.

[8] A.R.Curtis, M.J.D.Powell, J.K.Reid: On the estimation of sparse Jacobian matrices. IMA Journal of Aplied Mathematics 13, 117-119, 1974.

[9] R.S.Dembo, T.Steihaug: Truncated Newton Algorithms for Large-Scale Optimization. Mathematical Programming, 26, 190-212, 1983.

[10] I.S.Duff, M.Munksgaard, H.B.Nielsen, J.K.Reid: Direct solution of sets of linear equations whose matrix is sparse and indefinite. J. Inst. Maths. Applics., 23, 235-250, 1979.

[11] C.Durazzi, V.Ruggiero: Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems. To appear in Numerical Linear Algebra with Applications.

[12] A.El-Bakry, R.Tapia, T.Tsuchiya, Y.Zhang: On the formulation and theory of Newton interior-point method for nonlinear programming. Journal of Optimization Theory and Applications, 89, 507-541, 1996.

[13] P.E.Gill, W.Murray: Newton type methods for nconstrained and linearly constrained optimization. Mathematical Programming, 7, 311-350, 1974.

[14] C.Keller, N.I.M.Gould, A.J.Wathen: Constraint preconditioning for indefinite linear systems. SIAM J. on Matrix Analysis and Applications, 21, 1300-1317, 2000.

[15] L.Lukšan, M.Tůma, M.Šiška, J.Vlček, N.Ramešová: UFO 2000 - Interactive system for universal functional optimization. Technical Report V-826. Institute of Computer Science, Academy of Sciences, Prague, ICS AS CR, 2000.

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

[17] L.Lukšan, J.Vlček: Sparse and partially separable test problems for unconstrained and equality constrained optimization. Report V-767. Prague, ICS AS CR, 1998.

[18] L.Lukšan, J.Vlček: Numerical experience with iterative methods for equality constrained nonlinear programming problems. Optimization Methods and Software, 16, 257-287, 2001.

[19] I.Perugia, V.Simoncini: Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. Numerical Linear Algebra with Applications, 7, 1-32, 2000.

[20] M.Rozložník, V.Simoncini: Short-term recurrences for indefinite preconditioning of saddle point problems. Report No. SAM 2000-08, SAM, ETH Zurich, 2000

[21] P.S.Vassilevski, D.Lazarov: Preconditioning mixed finite element saddle-point elliptic problems. Numerical Linear Algebra with Applications, 3, 1-20, 1996.

[22] A.N.Tikhonov, V.Y.Arsenin: Solutions of ill-posed problems. Winston, Washington, D.C., 1977.

[23] D.F.Shanno, J.Vanderbei: Interior-point methods for nonconvex nonlinear programming: Orderings and higher-order methods. Mathematical Programming, 87, 303-316, 2000.

[24] J.Vanderbei, D.F.Shanno: An interior point algorithm for nonconvex nonlinear programming. Computational Optimization and Applications, 13, 231-252, 1999.

[25] J.Vanderbei, D.F.Shanno: Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions. Report No. ORFE-00-06, Operations Research and Financial Engineering, Princeton University, December 2000.

[26] L.N.Vicente, S.J.Wright: Local convergence of a primal-dual method for degenerate nonlinear programming. To appear in SIAM J. Optimization.

[27] M.H.Wright: Interior methods for constrained optimization, Acta Numerica, 1991.

[28] Y.Ye: Interior Point Algorithms - Theory and Analysis, Wiley-Interscience Publication, 1997.