Fully Iterative Indefinitely Preconditioned Algorithm for Large Sparse Equality Constrained Nonlinear Programming Problems

Lukšan, Ladislav
1996

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Fully Iterative Indefinitely Preconditioned Algorithm for Large Sparse Equality Constrained Nonlinear Programming Problems

Ladislav Lukšan Jan Vlček

Technical report No. 691

October 1996

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: (+422) 66414244 fax: (+422) 8585789
e-mail: uivt@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Fully Iterative Indefinitely Preconditioned Algorithm for Large Sparse Equality Constrained Nonlinear Programming Problems

Ladislav Lukšan Jan Vlček[1]

Technical report No. 691
October 1996

Abstract

A fully iterative algorithm for large sparse equality constrained nonlinear programming problems is proposed. This algorithm is based on the smoothed conjugate gradient method with special indefinite preconditioner. The efficiency of our algorithm is demonstrated by extensive numerical experiments.

Keywords

Nonlinear programming, sparse problem, equality constraints, truncated Newton method, augmented Lagrangian function, indefinite system, indefinite preconditioner, conjugate gradient method, residual smoothing

[1] This work was supported under grant No. 201/96/0918 given by the Czech Republic Grant Agency

1 Introduction

Consider the problem of finding a point $x^* \in R^n$, such that

$$x^* = \arg \min_{x \in \mathcal{F}} F(x), \tag{1.1}$$

where $\mathcal{F} \subset R^n$ is a feasible set defined by the system of equations

$$\mathcal{F} = \{x \in R^n : c_k(x) = 0, 1 \le k \le m\}. \tag{1.2}$$

where $m \le n$ (in fact we consider only local minimum). Here $F : R^n \to R$ and $c_k : R^n \to R$, $1 \le k \le m$, are twice continuously differentiable functions, whose gradients and Hessian matrices will be denoted by $\nabla F(x)$, $\nabla c_k(x)$, $1 \le k \le m$, and $\nabla^2 F(x)$, $\nabla^2 c_k(x)$, $1 \le k \le m$, respectively. Furthermore, we use the notation $c(x) = [c_1(x), \dots, c_m(x)]^T$ and $A(x) = [a_1(x), \dots, a_m(x)] = [\nabla c_1(x), \dots, \nabla c_m(x)]$ and we suppose that the matrix $A(x)$ has a full column rank. Then the solution $x^* \in R^n$ of the problem (1.1)-(1.2) satisfies the Karush-Kuhn-Tucker (KKT) conditions, i.e. there exists a vector $u^* \in R^m$, such that

$$\begin{aligned} \nabla_x L(x^*, u^*) &= \nabla F(x^*) + A(x^*)u^* = 0, & (1.3) \\ \nabla_u L(x^*, u^*) &= c(x^*) = 0, & (1.4) \end{aligned}$$

where

$$L(x, u) = F(x) + u^T c(x) \tag{1.5}$$

is the Lagrangian function, whose gradient and Hessian matrix will be denoted by

$$\begin{aligned} g(x, u) &= \nabla_x L(x, u) = \nabla F(x) + \sum_{k=1}^{m} u_k \nabla c_k(x), \\ G(x, u) &= \nabla_x^2 L(x, u) = \nabla^2 F(x) + \sum_{k=1}^{m} u_k \nabla^2 c_k(x), \end{aligned}$$

and $(x^*, u^*) \in R^{n+m}$ is the KKT pair (first order necessary conditions). Let $Z(x)$ be the matrix whose columns form an orthonormal basis in the null space of $A^T(x)$ so that $A^T(x)Z(x) = 0$ and $Z^T(x)Z(x) = I$. If, in addition to (1.3)-(1.4), the matrix $Z^T(x^*)G(x^*, u^*)Z(x^*)$ is positive definite, then the point $x^* \in R^n$ is a solution of the problem (1.1)-(1.2) (second order sufficient conditions).

Basic methods for a solution of the problem (1.1)-(1.2) are iterative and their iteration step has the form

$$\begin{aligned} x^+ &= x + \alpha d, & (1.6) \\ u^+ &= u + \alpha v, & (1.7) \end{aligned}$$

where $(d, v) \in R^{n+m}$ is a direction pair ($d \in R^n$ is a direction vector) and $\alpha > 0$ is a stepsize. In this contribution, we confine our attention to methods derived from the

Newton method used for a solution of the KKT system (1.3)-(1.4). The iteration step of the Newton method has the form (1.6)-(1.7), where $\alpha = 1$ and

$$\begin{bmatrix} G(x,u) & A(x) \\ A^T(x) & 0 \end{bmatrix} \begin{bmatrix} d \\ v \end{bmatrix} = - \begin{bmatrix} g(x,u) \\ c(x) \end{bmatrix}. \tag{1.8}$$

This is a system of $n+m$ linear equations with $n+m$ unknowns $(d,v) \in R^{n+m}$, whose matrix is always indefinite. Moreover, the matrix $G(x,u)$ is not positive definite in general even if the matrix $Z(x)^T G(x,u) Z(x)$ is. This fact can lead to some difficulties when standard positive definite preconditioners are used. In this case, it is advantageous to transform the system (1.8) in such a way as to contain, if possible, a positive definite matrix in the left-upper corner. This can often be done by addition of the second equation, multiplied by ρA, to the first equation (cf. Theorem 1), which yields

$$\begin{bmatrix} B & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} d \\ v \end{bmatrix} = - \begin{bmatrix} b \\ c \end{bmatrix}, \tag{1.9}$$

where

$$B = G + \rho A A^T \tag{1.10}$$

and

$$b = g + \rho A c = \nabla F + A u + \rho A c. \tag{1.11}$$

If the matrices A and B are large and sparse, we can solve the system (1.9) either directly using the sparse Bunch-Parlett [1] decomposition or iteratively using the preconditioned conjugate gradient type method. In this contribution, we will concentrate on the preconditioned smoothed conjugate gradient method which has similar properties to the MINRES method but which uses simpler recursions. Great attention will be devoted to efficient preconditioning.

The contribution is organized as follows. In Section 2, we propose some results concerning system (1.9) for direction determination and show the correctness of the Armijo type line search procedure. Section 3 is devoted to studying preconditioners for the KKT system. Section 4 contains a detailed description of our algorithm for large sparse equality constrained nonlinear programming problems together with results obtained by extensive numerical experiments.

In this contribution, we denote by $\|.\|$ the Euclidean (or spectral) norm.

2 Direction determination and stepsize selection

In this section we consider the system (1.9) with an arbitrary value $\rho \geq 0$ even if our algorithm works well with the value $\rho = 0$. The following theorem demonstrates the influence of the value ρ on properties of the system (1.9).

Theorem 1 *Let the matrix $Z^T G Z$ be positive definite. Then there exists a number $\overline{\rho} > 0$, such that the matrix B is positive definite whenever $\rho \geq \overline{\rho}$.*

Proof. See [4].

We will use the augmented Lagrangian function

$$P(\alpha) = F(x + \alpha d) + (u + v)^T c(x + \alpha d) + \frac{\sigma}{2}\|c(x + \alpha d)\|^2, \tag{2.1}$$

with $\sigma \geq \rho$, as a merit function for the stepsize selection. The derivative of this function, for $\alpha = 0$, is given by the formula

$$P'(0) \triangleq \frac{dP(\alpha)}{d\alpha}|_{\alpha=0} = d^T(b + Av) + (\sigma - \rho)d^T Ac. \tag{2.2}$$

The following theorem holds.

Theorem 2 *Let $(d, v) \in R^{n+m}$ be an inexact solution to the system (1.9) such that*

$$\sigma - \rho \geq \frac{2\underline{B}\|d\|^2 - d^T Bd}{c^T c} \tag{2.3}$$

and

$$d^T h + (\sigma - \rho)c^T r \leq \frac{1}{2}(d^T Bd + (\sigma - \rho)c^T c) \tag{2.4}$$

hold, where

$$h = Bd + Av + b \tag{2.5}$$

and

$$r = A^T d + c \tag{2.6}$$

are corresponding residuals. Then $P'(0) \leq -\underline{B}\|d\|^2$.

Proof. Using (2.2)-(2.6) we get

$$\begin{aligned} P'(0) &= d^T(b + Av) + (\sigma - \rho)d^T Ac = -d^T Bd + d^T h + (\sigma - \rho)(c^T r - c^T c) \\ &\leq -\frac{1}{2}(d^T Bd + (\sigma - \rho)c^T c) \leq -\underline{B}\|d\|^2. \end{aligned}$$

□

Theorem 2 shows, that if $\sigma \geq \rho$ is sufficiently large and the system (1.9) is solved sufficiently accurately, then the vector $d \in R^n$ is a descent direction for the merit function (2.1). Besides (2.4), we need additional standard conditions for superlinear rate of convergence of the inexact Newton method (see [3] as an example). These conditions have to be related to the original system (1.8). Using (1.10)-(1.11) and (2.5)-(2.6), we obtain

$$Gd + Av + g = Bd + Av + b - \rho AA^T d - \rho Ac = h - \rho Ar,$$

so that standard conditions for superlinear rate of convergence are satisfied when

$$\|h - \rho Ar\| \leq \omega \min(\|g\|, \overline{g}) \tag{2.7}$$

and

$$\|r\| \leq \omega \min(\|c\|, \overline{c}), \tag{2.8}$$

where $0 \le \omega < 1$ and $\omega \le (\|g\| + \|c\|)$ hold simultaneously. The constants $\overline{g}$ and $\overline{c}$ serve as safeguards against unboundedness. Note that conditions (2.7)-(2.8) are considered separately since individual residuals of the system (1.9) can have considerably different norms.

Let $(d, v) \in R^{n+m}$ be an inexact solution to the system (1.9) satisfying assumptions of Theorem 2. Then we can use the standard Armijo rule for steplength determination i.e. $\alpha > 0$ in (1.6)-(1.7) is chosen so that it is the first member of the sequence $\underline{\beta}^j$, $j = 0, 1, 2, \ldots$, $0 < \underline{\beta} < 1$, such that

$$P(\alpha) - P(0) \le \underline{\varepsilon} \alpha P'(0), \tag{2.9}$$

where $0 < \underline{\varepsilon} < 1$.

In the subsequent considerations, we will assume that the matrix B is uniformly bounded and the matrix

$$K \triangleq \begin{bmatrix} B & A \\ A^T & 0 \end{bmatrix}$$

is uniformly nonsingular, i.e. there exist constants $\overline{B}$ and $\underline{K}$, independent of the curent iteration, such that $\|B\| \le \overline{B}$ and $\|Kw\| \ge \underline{K}\|w\|$ $\forall w \in R^{n+m}$ (the matrix B can be preliminarily modified if $\|B\| > \overline{B}$). Furthermore we will suppose that $\sigma \le \overline{\sigma}$ and that there exist constants $\overline{g}$, $\overline{G}$, $\overline{c}$, $\overline{A}$, independent of the current iteration, such that $\|\nabla F(x + \alpha d)\| \le \overline{g}$, $\|\nabla^2 F(x + \alpha d)\| \le \overline{G}$, $\|c(x + \alpha d)\| \le \overline{c}$, $\|A(x + \alpha d)\| \le \overline{A}$, $\|\nabla^2 c_k(x + \alpha d)\| \le \overline{G}$, $1 \le k \le m$, hold, respectively, for all $0 \le \alpha \le 1$.

Lemma 1 *Let the assumptions of Theorem 2 be satisfied (together with (2.7)-(2.8) and the assumptions of boundedness given above). Then there exists a constant $\overline{K}$, independent of the current iteration, such that*

$$P(\alpha) \le P(0) + \alpha P'(0) + \alpha^2 \overline{K} \|d\|^2 \tag{2.10}$$

$\forall 0 \le \alpha \le 1$.

Proof. Using (2.7)-(2.8) we get

$$\|r\| \le \overline{c}$$

and

$$\|h\| \le \|h - \rho A r\| + \|\rho A r\| \le \overline{g} + \rho \overline{A}\overline{c}.$$

Since (2.5)-(2.6) imply that

$$K \begin{bmatrix} d \\ u + v \end{bmatrix} + \begin{bmatrix} \nabla F + \rho A c \\ c \end{bmatrix} = \begin{bmatrix} h \\ r \end{bmatrix},$$

we can write

$$\underline{K}\|u + v\| \le \left\| K \begin{bmatrix} d \\ u + v \end{bmatrix} \right\| \le 2(\overline{g} + \rho \overline{A}\overline{c} + \overline{c}) \triangleq \underline{K}\overline{U}$$

(ρ is assumed to be constant). Applying the Taylor expansion to (2.1) and using (2.2), we get

$$\begin{aligned} P(\alpha) &\leq P(0) + \alpha P'(0) + \frac{1}{2}\alpha^2 \overline{G}\|d\|^2 + \frac{1}{2}\alpha^2 \sum_{k=1}^{m} |u_k + v_k| \overline{G}\|d\|^2 \\ &\quad + \frac{1}{2}\sigma\alpha^2 \overline{A}^2 \|d\|^2 + \frac{1}{2}\sigma\alpha^2 \sum_{k=1}^{m} |c_k| \overline{G}\|d\|^2 \\ &\leq P(0) + \alpha P'(0) + \frac{1}{2}\alpha^2 \left[(1 + \overline{U}\sqrt{m} + \overline{\sigma c}\sqrt{m})\overline{G} + \sigma \overline{A}^2 \right] \|d\|^2 \\ &\triangleq P(0) + \alpha P'(0) + \alpha^2 \overline{K}\|d\|^2 \end{aligned}$$

$\forall 0 \leq \alpha \leq 1$. □

Theorem 3 *Let the assumptions of Lemma 1 hold and let $d \neq 0$. Then there exist an integer $k \geq 0$ and a number $\underline{\alpha} > 0$, independent of the current iteration, such that the Armijo rule gives the value $\alpha = \underline{\beta}^j$, satisfying (2.9), with $j \leq k$ and $\alpha \geq \underline{\alpha}$. Moreover*

$$P(\alpha) - P(0) \leq -\underline{\alpha \varepsilon B}\|d\|^2. \qquad (2.11)$$

Proof. Using Lemma 1 and Theorem 2, we can write

$$P(\alpha) - P(0) \leq \alpha(P'(0) + \alpha\overline{K}\|d\|^2) \leq \alpha P'(0)(1 - \alpha\frac{\overline{K}}{\underline{B}}),$$

so that (2.9) holds whenever $\alpha \leq (\underline{B}/\overline{K})(1 - \underline{\varepsilon})$. Let $k \geq 0$ be chosen so that it is the lowest integer such that $\underline{\beta}^k \leq (\underline{B}/\overline{K})(1 - \underline{\varepsilon})$ and let $\alpha = \underline{\beta}^j$ be given by the Armijo rule to satisfy (2.9). Then

$$\alpha = \underline{\beta}^j \geq \underline{\beta}^k \geq \underline{\beta}\frac{\underline{B}}{\overline{K}}(1 - \underline{\varepsilon}) \triangleq \underline{\alpha}. \qquad (2.12)$$

Using (2.12) and Theorem 2, we get

$$P(\alpha) - P(0) \leq \alpha\underline{\varepsilon}P'(0) \leq -\underline{\alpha\varepsilon B}\|d\|^2.$$

□

Now we focus our attention on the inexact solution of the equation (1.9). To simplify the notation we put

$$y = \begin{bmatrix} d \\ v \end{bmatrix}, \; z = \begin{bmatrix} b \\ c \end{bmatrix}, \; s = \begin{bmatrix} h \\ r \end{bmatrix},$$

i.e. the system (1.9) will be written in the form $s = Ky + z = 0$. To solve this system, we use the smoothed conjugate gradient method preconditioned by the matrix C. The resulting algorithm is based on the following philosophy. Step 2 realizes basic preconditioned conjugate gradient method. In Step 3, the residual vector is smoothed. Step 4 is devoted to testing a required accuracy. It serves as a switch for using additional decisions. In Step 4, we compute the penalty parameter satisfying the condition (2.3). Step 5 contains condition for descent (2.4) as a termination criterion.

Algorithm 1. Direction determination.

Data: $\rho \geq 0$, $0 < \underline{\sigma} < \overline{\sigma}$, $\underline{B} > 0$, $0 < \omega < 1$, $\overline{c} > 0$, $\overline{g} > 0$.

Step 1: Initiation. Set $\tilde{y}_0 := 0$, $\tilde{s}_0 := z$ $y_0 := \tilde{y}_0$, $s_0 := \tilde{s}_0$ $\omega := \min(\omega, \|s_0\|)$, and $j := 0$.

Step 2: CG iteration. If $j \geq n + m + 3$, then go to Step 6, otherwise set $j := j + 1$. Compute $\tilde{p}_{j-1} := C^{-1}\tilde{s}_{j-1}$ and $\beta_{j-1} := \tilde{p}_{j-1}^T \tilde{s}_{j-1}$. If $j = 1$, then set $p_{j-1} := \tilde{p}_{j-1}$, otherwise set $p_{j-1} := \tilde{p}_{j-1} + (\beta_{j-1}/\beta_{j-2})p_{j-2}$. Compute $q_{j-1} := Bp_{j-1}$ and $\gamma_{j-1} := \beta_{j-1}/p_{j-1}^T q_{j-1}$ and set $\tilde{y}_j := \tilde{y}_{j-1} + \gamma_{j-1}p_{j-1}$, $\tilde{s}_j := \tilde{s}_{j-1} + \gamma_{j-1}q_{j-1}$.

Step 3: Residual smoothing. Compute $\lambda_j := -(s_{j-1} - \tilde{s}_j)^T \tilde{s}_j / \|s_{j-1} - \tilde{s}_j\|^2$ and set $y_j := \tilde{y}_j + \lambda_j(y_{j-1} - \tilde{y}_j)$, $s_j := \tilde{s}_j + \lambda_j(s_{j-1} - \tilde{s}_j)$.

Step 4: Test for sufficient precision. If $\|r_j\| > \omega \min(\|c\|, \overline{c})$, then go to Step 2. Otherwise compute $w_j := h_j - \rho A r_j$. If $\|w_j\| > \omega \min(\|g\|, \overline{g})$, then go to Step 2.

Step 5: Determination of the penalty parameter. Compute the value $\kappa_j := d_j^T B d_j$ and set $\sigma_j := \min(\overline{\sigma}, \max(\underline{\sigma}, \rho, \rho + (2\underline{B}\|d\|^2 - \kappa_j)/\|c\|^2))$. Set $\kappa_j := \kappa_j + (\sigma_j - \rho)\|c\|^2$

Step 6: Test for sufficient descent. Set $\mu_j = d_j^T h_j + (\sigma - \rho)c^T r_j$. If $\mu_j > \kappa_j/2$, then go to Step 2. Otherwise set $y := y_j$, $\sigma := \sigma_j$, $P'(0) := \mu_j - \kappa_j$ and terminate the computation.

Note that the main reason for residual smoothing in Step 3 are conditions (2.7)-(2.8) which require the corresponding norms to be as small as possible.

3 Preconditioning

The main purpose of the preconditioner C is to change a spectrum of the matrix K to obtain more clustered eigenvalues, which usually leads to acceleration of convergence. Ideally, the matrix C should be as close to matrix K as possible but multiplication by the matrix C^{-1} has to be easily computed. If the matrix B is nonsingular, then we can write

$$K^{-1} = \begin{bmatrix} B^{-1} - B^{-1}A(A^T B^{-1} A)^{-1} A^T B^{-1} & B^{-1}A(A^T B^{-1} A)^{-1} \\ (A^T B^{-1} A)^{-1} A^T B^{-1} & -(A^T B^{-1} A)^{-1} \end{bmatrix}, \tag{3.1}$$

so that the matrix C^{-1} should have a similar structure but it should be realized by sparse decompositions. We concentrate our attention to the following preconditioners:

1) Block diagonal positive definite preconditioner

$$C = \begin{bmatrix} LL^T & 0 \\ 0 & D \end{bmatrix} \tag{3.2}$$

(see [8]), where LL^T is an incomplete Gill-Murray [5] decomposition of the matrix B and D is a positive definite diagonal matrix. In this case

$$C^{-1} = \begin{bmatrix} (LL^T)^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix}. \tag{3.3}$$

2) More complicated positive definite preconditioner

$$C = \begin{bmatrix} LL^T & A \\ A^T & A^T(LL^T)^{-1}A + D \end{bmatrix} \tag{3.4}$$

(see [8]), where LL^T is an incomplete Gill-Murray [5] decomposition of the matrix B and D is a positive definite diagonal matrix. In this case

$$C^{-1} = \begin{bmatrix} (LL^T)^{-1} + (LL^T)^{-1}AD^{-1}A^T(LL^T)^{-1} & -(LL^T)^{-1}AD^{-1} \\ -D^{-1}A^T(LL^T)^{-1} & D^{-1} \end{bmatrix}. \tag{3.5}$$

If $LL^T = B$ (i.e. if LL^T is a complete Choleski decomposition of the matrix B), then

$$KC^{-1} = \begin{bmatrix} I & 0 \\ (I + A^TB^{-1}AD^{-1})A^TB^{-1} & -A^TB^{-1}AD^{-1} \end{bmatrix} \tag{3.6}$$

3) New indefinite preconditioner

$$C^{-1} = \begin{bmatrix} D^{-1} - D^{-1}A(R^TR)^{-1}A^TD^{-1} & D^{-1}A(R^TR)^{-1} \\ (R^TR)^{-1}A^TD^{-1} & -(R^TR)^{-1} \end{bmatrix}, \tag{3.7}$$

where R^TR is an incomplete Choleski decomposition of the matrix $A^TD^{-1}A$ and D is a positive definite diagonal matrix. In this case

$$C \approx \begin{bmatrix} D & A \\ A^T & 0 \end{bmatrix}. \tag{3.8}$$

If $R^TR = A^TD^{-1}A$ (i.e. if R^TR is a complete Choleski decomposition of the matrix $A^TD^{-1}A$), then (3.8) holds with equality and

$$KC^{-1} = \begin{bmatrix} I + (BD^{-1} - I)(I - A(A^TD^{-1}A)^{-1}A^TD^{-1}) & (BD^{-1} - I)A(A^TD^{-1}A)^{-1} \\ 0 & I \end{bmatrix}, \tag{3.9}$$

The first two preconditioners were studied in [8]. They were developed for problems with positive definite matrix B, but KKT systems for equality constrained nonlinear programming problems do not have this property in general. Therefore, the efficiency of these preconditioners depends strongly on the value ρ appearing in (1.10) as will be mentioned in the next section. In this section, we will study properties of the indefinite preconditioner defined by (3.7). The main advantage of this preconditioner is a surprisingly excellent efficiency which is experimentally demonstrated in the next section.

Theorem 4 *Let K be nonsingular and (3.8) holds with equality. Then the matrix KC^{-1} has at least $2m$ unit eigenvalues.*

Proof. Let

$$KC^{-1}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}. \tag{3.10}$$

Using the last group of rows of (3.9), we can write $y = \lambda y$, so that $\lambda \neq 1 \Rightarrow y = 0$. Let $\lambda \neq 1$, so that $y = 0$. Using the first group of rows of (3.9), we get

$$(BD^{-1} - I)(I - A(A^T D^{-1} A)^{-1} A^T D^{-1})x = (\lambda - 1)x, \tag{3.11}$$

which is an eigenvalue problem of the dimension n. Since the matrix A has full column rank by the assumption, there exist m linearly independent vectors of the form $x = Au$, for which the left hand side in (3.11) is zero, i.e. which are eigenvectors of (3.11) with zero eigenvalue $\lambda - 1$. This implies that the original problem (3.10) has at most $n - m$ nonunit eigenvalues. □

Theorem 4 shows that the eigenvalues of the matrix KC^{-1} tend to be clustered. If $m = n$, i.e. if the matrix A is square and nonsingular, then all eigenvalues of the matrix KC^{-1} are units. On the other hand the matrix KC^{-1} is nonsymmetric and can have a complex structure. This is a consequence of indefiniteness of the matrix C which does not have a square root. For this reason, we cannot apply the standard estimates to rate of convergence of the conjugate gradient method. On the other hand, the conjugate gradient method with indefinite preconditioner still terminates after at most $n + m$ iterations and it is simpler than methods developed for nonsymmetric systems.

We have examined indefinite preconditioners (3.7) with various diagonal matrices $D = diag\{D_{ii}\}$. The best results were obtained with the choice

$$\begin{aligned} D_{ii} &= \underline{\Delta}, && \text{if} \quad |B_{ii}| < \underline{\Delta}, \\ D_{ii} &= |B_{ii}|, && \text{if} \quad \underline{\Delta} \leq |B_{ii}| \leq \overline{\Delta}, \\ D_{ii} &= \overline{\Delta}, && \text{if} \quad \overline{\Delta} < |B_{ii}|, \end{aligned} \tag{3.12}$$

where $\underline{\Delta} = 10^{-3}$ and $\overline{\Delta} = 10^6$.

4 Numerical experiments

Now we summarize results from the previous section and give a detailed description of our algorithm. This algorithm uses the indefinitely preconditioned smoothed conjugate gradient method for direction determination and the classical Armijo rule for stepsize selection. Before description of the algorithm, we have to note that it is advantageous to use some restart procedure, which treats the cases when the matrix K is unsuitable for direction determination. These cases are characterized by a large value of the derivative $P'(0)$. For this purpose we set $\tau = 10^{-4}$, if $\sigma = \max(\underline{\sigma}, \rho)$, or $\tau = 10^{-1}$, otherwise. If $-P'(0) < \tau \|d\| \|g\|$, then we repeat computation of the direction pair using the diagonal matrix $\tilde{B} = diag\{\tilde{B}_{ii}\}$ instead of B, where

$$\begin{aligned} \tilde{B}_{ii} &= \underline{\Gamma}, \quad \text{if} \quad \frac{\|g\|}{10}|B_{ii}| < \underline{\Gamma}, \\ \tilde{B}_{ii} &= \frac{\|g\|}{10}|B_{ii}|, \quad \text{if} \quad \underline{\Gamma} \le \frac{\|g\|}{10}|B_{ii}| \le \overline{\Gamma}, \\ \tilde{B}_{ii} &= \overline{\Gamma}, \quad \text{if} \quad \overline{\Gamma} < \frac{\|g\|}{10}|B_{ii}|, \end{aligned} \tag{4.1}$$

where $\underline{\Gamma} = 0.005$ and $\overline{\Gamma} = 500.0$. This procedure was obtained experimentally.

Algorithm 2. Equality constrained optimization (CG).

Data: $\rho \ge 0$, $0 < \underline{\sigma} < \overline{\sigma}$, $\underline{B} > 0$, $\overline{B} > 0$, $0 < \overline{\omega} < 1$, $0 < \underline{\beta} < 1$, $0 < \underline{\varepsilon} < 1$, $\overline{\delta} > 0$, $\overline{c} > 0$, $\overline{g} > 0$.

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

Step 1: Initiation. Determine sparsity pattern of the matrix B. Compute the value $F := F(x)$ and the vector $c := c(x)$. Set $u := 0$ and $i := 0$.

Step 2: Termination. Compute the matrix $A := A(x)$ and the vector $g := g(x, u)$. If $\|c\| \le \overline{\delta}$ and $\|g\| \le \overline{\delta}$, then terminate the computation (the solution is found). Otherwise set $i = i + 1$.

Step 3: Approximation of the Hessian matrix. Compute an approximation G of the Hessian matrix $G(x, u)$, using differences of gradient $g(x, u)$ as in [2]. Compute the matrix $B := G + \rho A A^T$ and modify it when its elements are larger than $\overline{B}$.

Step 4: Direction determination. Choose the diagonal matrix D by (3.12) and compute the matrix $A^T D^{-1} A$ together with its incomplete Choleski decomposition $R^T R$. Set $\omega = \min(1/i, \overline{\omega})$. Determine the direction pair (d, v) and the derivative $P'(0)$ using Algorithm 1.

Step 5: Restart. Set $\tau = 10^{-4}$ if $\sigma = \max(\underline{\sigma}, \rho)$, or $\tau = 10^{-1}$, otherwise. If $-P'(0) < \tau \|d\| \|g\|$ then determine the diagonal matrix $\tilde{B}$ by (4.1), set $B := \tilde{B}$ and go to Step 4. Otherwise set $\alpha := 1$ and compute the value of the merit function $P(\alpha)$.

Step 6: Termination of the stepsize selection. If $P(\alpha) - P(0) \le \underline{\varepsilon} \alpha P'(0)$, then set $x := x + \alpha d$, $u := u + \alpha v$ and go to Step 2.

Step 7: Continuation of the stepsize selection. Set $\alpha := \underline{\beta} \alpha$, compute the value of the merit function $P(\alpha)$ and go to Step 6.

The computational efficiency of Algorithm 2 was tested using 18 sparse problems, proposed in [6], which had either 50 or 100 variables. We used parameters $\underline{\sigma} = 1.5$, $\overline{\sigma} = 10^{16}$, $\underline{B} = 10^{-16}$, $\overline{B} = \overline{c} = \overline{g} = 10^{60}$, $\underline{\beta} = 0.5$ $\underline{\varepsilon} = 10^{-4}$, $\overline{\omega} = 0.9$, $\overline{\delta} = 10^{-6}$, in all numerical experiments. Values of the parameter ρ depended on the preconditioner used. We set $\rho = 0$ if indefinite preconditioner (3.7) was applied. In all the other cases we used various nonzero values of the parameter ρ which were chosen to give good results.

The summary of results for all 18 problems is given in Table 1. This table contains the total number of iterations NIT, the total number of function evaluations NFV, the total number of gradient evaluations NGR, the total number of conjugate gradient iterations NCG, the total number of restarts NRS, the total number of failures NFL and the total CPU time on Pentium PC (90 MHz) for double precision arithmetic implementation. The rows correspond to our method (CG) realized by Algorithm 2 with various preconditioners (P0 - without preconditioner, P1 - positive definite preconditioner (3.2), P2 - positive definite preconditioner (3.4), P3 - indefinite preconditioner (3.7)), the range space method (GM+CG) proposed in [6] and the direct method with the Bunch-Parlett (BP) decomposition of the matrix B. All methods presented in Table 1 were implemented using the modular interactive system for universal functional optimization UFO [7].

$n \sim$ 50	NIT	NFV	NGR	NCG	NRS	NFL	CPU
CG-P0	297	395	2153	15309	18	1	21.15
CG-P1	285	408	2063	5100	22	-	12.57
CG-P2	275	436	2041	3951	34	-	13.56
CG-P3	252	319	1828	1025	20	-	5.49
GM+CG	258	298	1847	1089	-	-	5.60
BP	249	344	1775	-	28	-	7.25
$n \sim$ 100	NIT	NFV	NGR	NCG	NRS	NFL	CPU
CG-P0	384	521	2735	42180	37	1	117.87
CG-P1	340	957	2357	7386	27	-	37.02
CG-P2	320	627	2266	4135	36	-	30.32
CG-P3	260	292	1868	1021	18	-	12.25
GM+CG	265	293	1889	1370	-	-	12.52
BP	296	410	2078	-	51	-	16.64

Table 1

For better demonstration of the efficiency of indefinite preconditioner (3.7), we introduce Table 2. This table contains numbers of iterations of preconditioned smoothed conjugate gradient method applied to 18 linear KKT systems. These systems occured in the first iteration of Algorithm 2 applied to 18 test problems which served for experiments reported in the lower part of Table 1

No.	n	m	P0	P1	P2	P3
1	100	98	1465	174	200	7
2	100	93	1214	747	(*)	17
3	100	2	139	328	305	261
4	100	98	1465	174	200	7
5	100	96	1241	90	85	9
6	99	49	606	(*)	(*)	119
7	100	4	162	17	12	7
8	100	98	1188	39	29	11
9	100	6	162	176	161	162
10	100	98	1465	174	200	162
11	98	64	775	811	1095	293
12	97	72	975	272	259	110
13	98	64	577	936	1077	59
14	98	64	516	414	666	67
15	97	72	359	250	283	40
16	97	72	734	76	61	35
17	97	72	734	76	61	35
18	97	72	734	76	61	35

Table 2

All linear systems were solved with high precision ($\varpi = 10^{-12}$). Columns of Table 2 correspond to preconditioners P0, P1, P2, P3, respectively. Asterisks denote failures (more than $10(n+m)$ iterations).

According to results presented in the above tables, we can give several conclusions. First, the most important observation is the fact, that the fully iterative method (CG) with indefinite preconditioner (3.7) applied to the KKT system is more efficient that methods (GM+CG) and (BP) which use direct solvers and, therefore, have larger storage requirements. Second, a very useful property of our algorithm is the fact, that it is not sensitive to ill conditioning or indefiniteness of the matrix B and, therefore, it works well with the choice $\rho = 0$. Third, we have to note, that the conjugate gradient method with indefinite preconditioner (3.7) can theoretically lead to breakdown. However this situation never appeared in our computational experiments.

Bibliography

[1] J.R.Bunch, B.N.Parlett: Direct methods for solving symmetric indefinite systems of linear equations. SIAM J. Numerical Analysis 8 (1971) 639-655.

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

[3] R.S.Dembo, S.C.Eisenstat, T.Steihaug: Inexact Newton methods. SIAM J. Numerical Analysis 19 (1982) 400-408.

[4] R.Fletcher: Practical methods of optimization. Wiley, Chichester, 1987.

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

[6] L.Lukšan, J.Vlček: Efficient algorithm for large sparse equality constrained nonlinear programming problem. Research Report No. V-652, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, Czech Republic, 1995.

[7] L.Lukšan, M.Šiška, M.Tůma, J.Vlček, N.Ramešová: Interactive System for Universal Functional Optimization (UFO), Version 1995. Research Report No. V-662, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, Czech Republic, 1995.

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

[9] R.Weiss: Convergence behavior of generalized conjugate gradient methods. Thesis. University of Karlsruhe, 1990.