# Matrix-Free Interior Point Method

Jacek Gondzio[*]


School of Mathematics
and Maxwell Institute for Mathematical Sciences
The University of Edinburgh
Mayfield Road, Edinburgh EH9 3JZ
United Kingdom.

## Abstract

In this paper we present a redesign of a linear algebra kernel of an interior point method to avoid the explicit use of problem matrices. The only access to the original problem data needed are the matrix-vector multiplications with the Hessian and Jacobian matrices. Such a redesign requires the use of suitably preconditioned iterative methods and imposes restrictions on the way the preconditioner is computed. A two-step approach is used to design a preconditioner. First, the Newton equation system is regularized to guarantee better numerical properties and then it is preconditioned. The preconditioner is *implicit*, that is, its computation requires only matrix-vector multiplications with the original problem data. The method is therefore well-suited to problems in which matrices are not explicitly available and/or are too large to be stored in computer memory. Numerical properties of the approach are studied including the analysis of the conditioning of the regularized system and that of the preconditioned regularized system. The method has been implemented and preliminary computational results for small problems limited to 1 million of variables and 10 million of nonzero elements demonstrate the feasibility of the approach.

*Keywords:* Linear Programming, Quadratic Programming, Matrix-Free, Interior Point Methods, Iterative Methods, Implicit Preconditioner.

---

[*]Email: `J.Gondzio@ed.ac.uk`, URL: `http://maths.ed.ac.uk/~gondzio/`

# 1  Introduction

Interior Point Methods (IPMs) for linear and quadratic programming can solve very large problems in a moderate number of iterations. The best known to date primal-dual methods [25, 36] have a guaranteed worst-case complexity of $\mathcal{O}(\sqrt{n})$ iterations to reach optimality. Their practical performance is much better than that guaranteed by the worst-case complexity analysis. Indeed, they usually converge in a number of iterations which is proportional to $\log n$ [3, 9]. Even with such a small number of iterations these methods can occasionally be computationally expensive.

IPMs rely on Newton method and therefore share the usual disadvantages of second-order methods: at each iteration they require building and solving linear equations corresponding to KKT systems for the barrier subproblem. These computations might sometimes get expensive. This is bound to happen when solving dense problems such as for example quadratic programs arising in support vector machine training [16] or linear programs arising in basis pursuit [8], but it might also occur when solving certain sparse problems in which the patters of nonzero entries in the Hessian and Jacobian matrices render KKT systems difficult for direct approaches as is often the case when dealing with problems arising from discretisations of partial differential equations [5]. As optimization problems get larger and larger this aspect of IPM methodology seems to be a bottleneck for their further development and hampers their ability to solve huge problems some of which cannot even be explicitly formulated. The technique presented in this paper aims at removing these drawbacks.

We redesign interior point methods by replacing direct linear algebra with a suitably preconditioned iterative method. This creates a difficulty because, unlike direct methods which usually provide very accurate directions, iterative methods compute only inexact directions. We accept the negative consequence of such a choice, that is, the need of performing more IPM iterations but expect great benefits from the reduction of effort required to compute inexact solution of Newton systems. The use of inexact Newton method goes back to Dembo et al. [12] and has had a number of applications including those in the context of IPMs [4, 18, 26]. A number of interesting developments have been focused on the analysis of conditions which inexact directions should satisfy to guarantee good convergence properties of the IPM [1, 27]. However the focus of this paper lies elsewhere.

We would like to solve KKT systems at the lowest possible cost. Moreover, we would like to avoid any excessive storage requirements and computations which might hamper the ability of solving huge problems. To achieve the goal we impose a condition that the Hessian and Jacobian matrices can be used only to perform multiplications with. In other words, we redesign the interior point method to work in a *matrix-free* regime.

Iterative methods for KKT systems have attracted a lot of attention in the recent years. 8 out of 10 papers published in a special issue of COAP [22] which was devoted to linear algebra techniques of IPMs addressed the use of iterative methods. The performance of iterative methods and more specifically, of Krylov-subspace methods [24] depends on the spectral properties of the linear system. A plethora of preconditioners have been designed for linear systems, especially for those arising from the discretisation of partial differential equations (PDEs). The reader is referred to [5] for an up-to-date survey of recent developments in this area. KKT systems arising in optimization and saddle point systems arising in PDE discretisations share several common

features but display essential differences as well. The structure of PDE-originated saddle point systems usually enjoys some form of regularity in the sparsity pattern which is a consequence of the discretisation process. In contrast, the sparsity pattern of a usual KKT system does not display any regularity at all. KKT systems arising in barrier subproblems of IPMs have a special feature of added diagonal scaling matrix resulting from the barrier terms. This matrix has a very large condition number (which goes to infinity when barrier term goes to zero) and causes the condition number of the KKT system to be unbounded.

The performance of iterative methods depends on the spectral properties of the linear system. Ideally, one would like all the eigenvalues of the matrix involved to be clustered in one or only a few intervals and/or the condition number of the matrix to be as small as possible [24]. This is an ultimate goal when designing preconditioners for KKT systems arising in IPMs [22, 11]. The choice of a preconditioner faces a particular difficulty because the condition number of these systems can go to infinity when the algorithm approaches an optimal solution. To guarantee a bounded condition number of the preconditioned linear system one has to accept that the condition number of the preconditioner may have to go to infinity as well. This causes all sorts of problems to the iterative method. One of them is that the termination criteria of the method which relies on the residual in the preconditioned space does not guarantee a small residual in the original space.

In this paper we propose a different approach to the problem. Instead of directly preconditioning the original KKT system arising in the barrier subproblem, we propose a two-step procedure. First, we regularize the KKT system to guarantee that its condition number is bounded; then, we design a preconditioner for the regularized system.

We also set additional requirements on the preconditioner: it has to be computed without explicitly forming (and factoring) the KKT system. The only operations the preconditioner is allowed to use are matrix-vector multiplications performed with the Hessian and Jacobian and its transposition. This is necessary to achieve a *matrix-free* implementation of interior point method.

It is worth mentioning at this point an interesting recent work [10] which proposed a matrix-free algorithm for equality constrained nonlinear programming problems. There are essential differences between our approach and [10]. In this paper we deal with interior point methods and need to remedy the ill-conditioning introduced by the presence of barrier terms while [10] focuses on the ill-conditioning that may result from the rank-deficient Jacobians. Secondly, we regularize both the primal and dual variables (which is well-suited to the primal-dual interior point method) while [10] regularizes the Hessian matrix (what in our terms would correspond to regularizing the primal solution only). Finally, and most importantly, we propose a preconditioner which can be computed in a matrix-free regime while [10, p. 1244] uses unpreconditioned MINRES.

The paper is organised as follows. In Section 2 we will introduce the problem and define the notation used. In Section 3 we will give motivations for the use of a particular primal-dual regularization and we will discuss the properties of regularized KKT system. In Section 4 we will analyse the normal equation form of the KKT system. In Section 5 we will introduce the preconditioner and perform the spectral analysis of the preconditioned linear system. In Section 6 we will focus on the computational aspects of the matrix-free interior point method and we will discuss the preliminary results obtained for small scale problems (with number of variables below 1 million and number of nonzero entries below 10 million). Finally, in Section 7

we will give our conclusions and comment on possible further developments of the method.

## 2   KKT System in Interior Point Methods

In this paper we are concerned with the use of primal-dual interior point methods to solve convex quadratic programming problems. We assume that the primal-dual pair of problems has the form

$$
\begin{array}{ll}
\text{Primal} & \text{Dual} \\
\begin{aligned}
\min \quad & c^T x + \tfrac{1}{2} x^T Q x \\
\text{s.t.} \quad & A x = b, \\
& x \geq 0;
\end{aligned}
&
\begin{aligned}
\max \quad & b^T y - \tfrac{1}{2} x^T Q x \\
\text{s.t.} \quad & A^T y + s - Q x = c, \\
& y \text{ free}, \ \ s \geq 0,
\end{aligned}
\end{array}
\tag{1}
$$

where $A \in \mathcal{R}^{m \times n}$ has full row rank $m \leq n$, $Q \in \mathcal{R}^{n \times n}$ is a positive semidefinite matrix, $x, s, c \in \mathcal{R}^n$ and $y, b \in \mathcal{R}^m$. The main computational effort of the primal-dual algorithm consists in the computation of the Newton direction for the barrier subproblem

$$
\begin{bmatrix} -(Q + \Theta^{-1}) & A^T \\ A & 0 \end{bmatrix}
\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}
=
\begin{bmatrix} f \\ d \end{bmatrix},
\tag{2}
$$

where $\Delta x \in \mathcal{R}^n$, $\Delta y \in \mathcal{R}^m$ are Newton directions in the primal and dual space, $f \in \mathcal{R}^n$, $d \in \mathcal{R}^m$ are appropriately computed right-hand-side vectors, $\Theta = X S^{-1}$ is the diagonal scaling matrix resulting from the presence of barrier terms, and $X$ and $S$ are diagonal matrices in $\mathcal{R}^{n \times n}$ with elements of vectors $x$ and $s$ spread across the diagonal, respectively.

The matrix $\Theta$ brings an unwelcome feature to this system. When the algorithm approaches the optimal solution, the primal variables and dual slacks converge to their optimal values and display a partition:

$$
\begin{aligned}
x_j \to x_j^* > 0 \quad \text{and} \quad s_j \to s_j^* = 0, \quad \text{for} \quad j \in \mathcal{B}, \\
x_j \to x_j^* = 0 \quad \text{and} \quad s_j \to s_j^* > 0, \quad \text{for} \quad j \in \mathcal{N}.
\end{aligned}
\tag{3}
$$

As a result the elements $\theta_j, j \in \mathcal{B}$, go to infinity and the elements $\theta_j, j \in \mathcal{N}$, go to zero. Unless the objective is strictly convex, in which case $Q$ is strictly positive definite, (3) implies that the eigenvalues of the (1,1) block in (2) may spread from zero to infinity. In the case of linear programming, when $Q = 0$, this is inevitable. This feature makes the solution of (2) challenging. Surprisingly, direct methods do not seem to suffer from this property. An old result of Dikin [13] (see also Stewart [35]) guarantees that if the optimization problem is feasible then the normal equation linear system (Schur complement obtained by eliminating the (1,1) block from (2)) produces a bounded solution irrespective of the spread of $\Theta$. Indeed, direct methods applied to (2) provide sufficiently accurate solution regardless the ill-conditioning of this linear system [3].

Unfortunately, for iterative (Krylov-subspace) methods the ill-conditioning of $\Theta$ and the resulting ill-conditioning of (2) makes the system intractable unless appropriately preconditioned [6, 20, 28, 29]. We formally state this observation below. Before we do that let us state the assumptions to be satisfied by problem (1) and the primal-dual algorithm applied to solve it.

**Assumptions:**

A1 The optimal solution of (1) exists and satisfies

$$(x^*, y^*, s^*) \quad \text{is} \quad \mathcal{O}(1).$$

A2 All iterates of the primal-dual algorithm stay in the symmetric neighbourhood of the (infeasible) central path

$$N_\beta^s = \{(x, y, s) \in \mathcal{F} : \beta\mu \le x_j s_j \le \frac{1}{\beta}\mu\},$$

for $\beta \in (0,1)$ and $\mathcal{F} = \{(x, y, s) : \|b - Ax\| \le \beta_b\mu\|b - Ax^0\|, \|c - A^T y - s + Qx\| \le \beta_c\mu\|c - A^T y^0 - s^0 + Qx^0\|, x > 0, s > 0\}$, where $(x^0, y^0, s^0)$ is an initial point, $\mu = x^T s/n$ is the barrier parameter and $\beta_b$ and $\beta_c$ are given constants.

As a simple consequence of these assumptions we claim

**Lemma 2.1** *If A1 and A2 are satisfied then*

$$\min_j \theta_j^{-1} = \mathcal{O}(\mu) \quad \text{and} \quad \max_j \theta_j^{-1} = \mathcal{O}(\mu^{-1}). \tag{4}$$

**Proof.** For $j \in \mathcal{B}$, we have $\theta_j^{-1} = \frac{s_j}{x_j} = \frac{x_j s_j}{x_j^2} = \frac{\mathcal{O}(\mu)}{\mathcal{O}(1)} = \mathcal{O}(\mu)$, hence $\min_j \theta_j^{-1} = \mathcal{O}(\mu)$. For $j \in \mathcal{N}$, we have $\theta_j^{-1} = \frac{s_j}{x_j} = \frac{s_j^2}{x_j s_j} = \frac{\mathcal{O}(1)}{\mathcal{O}(\mu)} = \mathcal{O}(\mu^{-1})$, hence $\max_j \theta_j^{-1} = \mathcal{O}(\mu^{-1})$. $\quad\square$

Throughout the paper we will assume that the singular values of $A$ are

$$0 < \sigma_1 \le \sigma_2 \le \cdots \le \sigma_m, \tag{5}$$

and the eigenvalues of $Q$ are

$$0 \le \tau_1 \le \tau_2 \le \cdots \le \tau_n. \tag{6}$$

In other words, we assume that $A$ has full row rank and that $Q$ is positive semi-definite. (We cover the special case of linear programming where $Q = 0$.) To simplify the analysis we will also denote the ordered eigenvalues of $Q + \Theta^{-1}$ as

$$0 < \pi_1 \le \pi_2 \le \cdots \le \pi_n, \tag{7}$$

and observe that

$$\tau_1 + \min_j \theta_j^{-1} \le \pi_1 \quad \text{and} \quad \pi_n \le \tau_n + \max_j \theta_j^{-1}. \tag{8}$$

The augmented system matrix (2) is certainly nonsingular and its eigenvalues are real and satisfy

$$\lambda_{-n} \le \lambda_{-n+1} \le \cdots \le \lambda_{-1} < 0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_m. \tag{9}$$

Following Lemma 2.1 in Rusten and Winther [32] (see also Lemma 2.1 in Silvester and Wathen [34]) we provide bounds for these eigenvalues.

**Lemma 2.2** *The eigenvalues of the augmented system matrix (2) satisfy:*

$$-\frac{1}{2}(\pi_n + \sqrt{\pi_n^2 + 4\sigma_m^2}) \le \lambda_{-n} \tag{10}$$

$$\lambda_{-1} \le -\pi_1 \tag{11}$$

$$\frac{1}{2}(\sqrt{\pi_n^2 + 4\sigma_1^2} - \pi_n) \le \lambda_1 \tag{12}$$

$$\lambda_m \le \frac{1}{2}(\sqrt{\pi_1^2 + 4\sigma_m^2} - \pi_1). \tag{13}$$

**Proof.** See the proof of Lemma 2.1 in [32]. □

Consequently, the eigenvalue with the largest absolute value satisfies

$$\max_j |\lambda_j| = \max\{-\lambda_{-n}, \lambda_m\}$$

$$\le \max\{\frac{1}{2}(\pi_n + \sqrt{\pi_n^2 + 4\sigma_m^2}), \frac{1}{2}(\sqrt{\pi_1^2 + 4\sigma_m^2} - \pi_1)\}$$

$$\le \max\{\pi_n + \sigma_m, \sigma_m\}$$

$$= \pi_n + \sigma_m \le \tau_n + \max_j \theta_j^{-1} + \sigma_m. \tag{14}$$

It is reasonable to assume that $\pi_n \gg 1$ and $\sigma_1 \le 1$ so the left hand side of (12) is

$$\frac{1}{2}(\sqrt{\pi_n^2 + 4\sigma_1^2} - \pi_n) = \frac{2\sigma_1^2}{\sqrt{\pi_n^2 + 4\sigma_1^2} + \pi_n} \approx \frac{\sigma_1^2}{\pi_n}.$$

Hence the eigenvalue with the smallest absolute value satisfies

$$\min_j |\lambda_j| = \min\{-\lambda_{-1}, \lambda_1\}$$

$$\ge \min\{\pi_1, \frac{1}{2}(\sqrt{\pi_n^2 + 4\sigma_1^2} - \pi_n)\}$$

$$\approx \min\{\pi_1, \frac{\sigma_1^2}{\pi_n}\}. \tag{15}$$

We will slightly abuse the notation and write $\min_j |\lambda_j| \ge \min\{\pi_1, \frac{\sigma_1^2}{\pi_n}\}$. Both terms on the right hand side of this inequality may get very small. From (8) $\pi_1 \ge \tau_1 + \min_j \theta_j^{-1}$ and $\frac{\sigma_1^2}{\pi_n} \ge \frac{\sigma_1^2}{\tau_n + \max_j \theta_j^{-1}}$. In a (difficult) case of solving an LP problem we have $\tau_1 = \tau_n = 0$ and observe that the condition number of augmented system (2) becomes

$$\kappa_{aug} \le \frac{\sigma_m + \max_j \theta_j^{-1}}{\min\{\min_j \theta_j^{-1}, \sigma_1^2 / \max_j \theta_j^{-1}\}}.$$

Even if $A$ is well conditioned and $\sigma_1$ and $\sigma_m$ are both $\mathcal{O}(1)$, following Lemma 2.1 we deduce that the condition number of the augmented system (2) may get as large as $\mathcal{O}(\mu^{-2})$. In the purely theoretical case of nondegenerate linear programs (the author has not yet seen a nondegenerate LP in his computational practice), a better bound $\mathcal{O}(\mu^{-1})$ for the condition number of (2) could be derived. This is a consequence of the fact that in nondegenerate case, the columns of $A$ corresponding to variable indices $j \in \mathcal{B}$ form a nonsingular optimal basis.

Summing up, ill-conditioning of (2) is an inevitable consequence of the use of primal-dual interior point methods and the use of logarithmic barrier penalties which contribute a badly behaving term $\Theta^{-1}$ to (2). Independently of whether we consider a computationally relevant degenerate case when $\kappa_{aug} = \mathcal{O}(\mu^{-2})$ or a nondegenerate case (of theoretical interest only) when $\kappa_{aug} = \mathcal{O}(\mu^{-1})$, with the barrier term $\mu$ going to zero, $\kappa_{aug}$ inevitably goes to infinity. This makes the system very difficult for iterative methods.

In the following sections we will discuss a two-step procedure to design a suitable preconditioner for (2).

## 3    Regularized KKT System

Following [21], Saunders [33] suggested adding proximal terms to optimization problems in order to improve the numerical properties of the corresponding KKT systems. Although his development was concerned with linear programs, it can be easily applied to quadratic problems as well. Instead of dealing with (1) we consider the following regularized primal quadratic program

$$
\begin{aligned}
\min \quad & c^T x + \frac{1}{2} x^T Q x + \frac{\gamma^2}{2} x^T x + \frac{1}{2} p^T p \\
\text{s.t.} \quad & Ax + \delta p = b, \\
& x \geq 0, \ p \text{ free};
\end{aligned}
\tag{16}
$$

where $\gamma$ and $\delta$ are (small) positive terms and $p \in \mathcal{R}^m$ is a free variable. It is intuitive that as long as $\gamma$ and $\delta$ are small the solution of (16) does not differ too much from that of (1). Indeed, the term $\frac{1}{2}\gamma^2 x^T x$ in the objective plays negligible role compared to the original objective $c^T x + \frac{1}{2} x^T Q x$. Similarly, the strong penalty for large $p$ in form of its norm added to the objective forces $p$ to be small, and therefore forces the linear constraint $Ax = b$ to be satisfied approximately.

The reduced KKT system corresponding to the barrier subproblem for (16) has the following form

$$
\begin{bmatrix} -(Q + \Theta^{-1} + \gamma^2 I) & A^T \\ A & \delta^2 I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f' \\ d' \end{bmatrix},
\tag{17}
$$

where $f' \in \mathcal{R}^n$, $d' \in \mathcal{R}^m$ are appropriately computed right-hand-side vectors.

The regularization proposed in [2] has a different justification. The primal-dual pair (1) is replaced by two problems: an approximate primal and an approximate dual:

$$
\begin{array}{llll}
\min & c^T x + \frac{1}{2} x^T Q x + \frac{1}{2}(x-x_0)^T R_p(x-x_0) & \max & b^T y - \frac{1}{2} x^T Q x - \frac{1}{2}(y-y_0)^T R_d(y-y_0) \\
\text{s.t.} & Ax = b, & \text{s.t.} & A^T y + s - Qx = c, \\
& x \geq 0; & & y \text{ free}, \ s \geq 0,
\end{array}
\tag{18}
$$

where primal-dual diagonal regularization matrices $R_p \in \mathcal{R}^{n \times n}$ and $R_d \in \mathcal{R}^{m \times m}$ and primal-dual reference points in proximal terms $x_0 \in \mathcal{R}^n$ and $y_0 \in \mathcal{R}^m$ can be chosen dynamically. The following regularized KKT system is solved at each iteration of the primal-dual interior point method [2]

$$
\begin{bmatrix} -(Q + \Theta^{-1} + R_p) & A^T \\ A & R_d \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f'' \\ d'' \end{bmatrix},
\tag{19}
$$

where $f'' \in \mathcal{R}^n$, $d'' \in \mathcal{R}^m$ are appropriately computed right-hand-side vectors.

While (17) adds uniform regularization terms to all diagonal elements of both the (1,1) and (2,2) blocks in the augmented system, (19) allows for a dynamic choice of the regularization terms and achieves similar benefits at a lower cost of introducing less of perturbation to the original reduced KKT system (2). Indeed, dynamic choice of $R_p$ and $R_d$ means that acceptable pivots are not regularized at all while potentially unstable pivots (dangerously close to zero) are suitably regularized to prevent the spread of numerical errors [2]. The reader interested in other regularization techniques is referred to [7, 19] and the references therein.

In this paper we are interested in the primal-dual regularization producing the system (19). To simplify the analysis we assume that the elements of the primal and dual regularizations satisfy

$$\gamma^2 \leq R_{pj} \leq \Gamma^2, \ \forall j \quad \text{and} \quad \delta^2 \leq R_{di} \leq \Delta^2, \ \forall i, \tag{20}$$

where $\Gamma$ and $\Delta$ are given constants. However, later we will relax some of these assumptions.

To compare the numerical properties of augmented systems (19) and (2) we derive bounds on the eigenvalues of (19). As in the previous section, we will assume that the eigenvalues of $\tilde{Q} = Q + \Theta^{-1} + R_p$ are

$$0 < \tilde{\pi}_1 \leq \tilde{\pi}_2 \leq \cdots \leq \tilde{\pi}_n, \tag{21}$$

and observe that

$$\tau_1 + \min_j\{\theta_j^{-1} + R_{pj}\} \leq \tilde{\pi}_1 \quad \text{and} \quad \tilde{\pi}_n \leq \tau_n + \max_j\{\theta_j^{-1} + R_{pj}\}. \tag{22}$$

The eigenvalues of (19) are real and satisfy

$$\tilde{\lambda}_{-n} \leq \tilde{\lambda}_{-n+1} \leq \cdots \leq \tilde{\lambda}_{-1} < 0 < \tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \cdots \leq \tilde{\lambda}_m. \tag{23}$$

**Lemma 3.1** _The eigenvalues of the augmented system matrix (19) satisfy:_

$$-\frac{1}{2}(\tilde{\pi}_n - \delta^2 + \sqrt{(\tilde{\pi}_n + \delta^2)^2 + 4\sigma_m^2}) \ \leq \ \tilde{\lambda}_{-n} \tag{24}$$

$$\tilde{\lambda}_{-1} \ \leq \ -\tilde{\pi}_1 \tag{25}$$

$$\frac{1}{2}(\sqrt{(\tilde{\pi}_n + \delta^2)^2 + 4\sigma_1^2} - \tilde{\pi}_n + \delta^2) \ \leq \ \tilde{\lambda}_1 \tag{26}$$

$$\tilde{\lambda}_m \ \leq \ \frac{1}{2}(\sqrt{(\tilde{\pi}_1 + \Delta^2)^2 + 4\sigma_m^2} - \tilde{\pi}_1 + \Delta^2). \tag{27}$$

**Proof.** Let $(u, v)$, $u \in \mathcal{R}^n$, $v \in \mathcal{R}^m$ denote an eigenvector associated with the eigenvalue $\tilde{\lambda}$. The eigenpair satisfies the following system of equations

$$-\tilde{Q}u + A^T v \ = \ \tilde{\lambda}u \tag{28}$$

$$Au + R_d v \ = \ \tilde{\lambda}v. \tag{29}$$

Suppose $\tilde{\lambda} < 0$. From (29) we have $(\tilde{\lambda}I - R_d)v = Au$. We substitute for $v$ in (28) and multiply the equation with $u^T$ to obtain

$$u^T \tilde{Q}u + u^T A^T (R_d - \tilde{\lambda}I)^{-1} Au \ = \ -\tilde{\lambda}u^T u.$$

Taking the largest possible terms on the left-hand-side of this equation yields

$$\tilde{\pi}_n u^T u + (\delta^2 - \tilde{\lambda})^{-1} \sigma_m^2 u^T u \; \geq \; -\tilde{\lambda} u^T u,$$

which can be managed into the following inequality

$$\tilde{\lambda}^2 + \tilde{\lambda}(\tilde{\pi}_n - \delta^2) - \tilde{\pi}_n \delta^2 - \sigma_m^2 \; \leq \; 0.$$

Any negative $\tilde{\lambda}$ which satisfies this quadratic inequality has to satisfy (24).

To prove (25) we multiply equations (28) and (29) with $u^T$ and $v^T$, respectively and subtract the second equation from the first one to obtain

$$u^T \tilde{Q} u + \tilde{\lambda} u^T u \; = \; \tilde{\lambda} v^T v - v^T R_d v.$$

We consider $\tilde{\lambda} < 0$ again hence the right-hand-side of this equation is nonpositive and therefore $u^T \tilde{Q} u + \tilde{\lambda} u^T u \leq 0$. Rearranging this inequality we obtain

$$\tilde{\pi}_1 u^T u \; \leq \; u^T \tilde{Q} u \; \leq \; -\tilde{\lambda} u^T u,$$

which yields (25).

Let us now consider the case of $\tilde{\lambda} > 0$. Using (28) we get $A^T v = (\tilde{Q} + \tilde{\lambda} I) u$. We substitute for $u$ in (29) and multiply the equation with $v^T$ to obtain

$$v^T A (\tilde{Q} + \tilde{\lambda} I)^{-1} A^T v + v^T R_d v \; = \; \tilde{\lambda} v^T v. \tag{30}$$

Taking the smallest possible terms on the left-hand-side of this equation yields

$$(\tilde{\pi}_n + \tilde{\lambda})^{-1} \sigma_1^2 v^T v + \delta^2 v^T v \; \leq \; \tilde{\lambda} v^T v,$$

which after rearrangements gives the following inequality

$$\tilde{\lambda}^2 + \tilde{\lambda}(\tilde{\pi}_n - \delta^2) - \tilde{\pi}_n \delta^2 - \sigma_1^2 \; \geq \; 0.$$

Any positive $\tilde{\lambda}$ which satisfies this quadratic inequality has to satisfy (26).

Finally, to get the last remaining bound, we consider again (30) and for $\tilde{\lambda} > 0$ take the largest possible terms on its left-hand-side to get

$$(\tilde{\pi}_1 + \tilde{\lambda})^{-1} \sigma_m^2 v^T v + \Delta^2 v^T v \; \geq \; \tilde{\lambda} v^T v,$$

which after rearrangements gives the following inequality

$$\tilde{\lambda}^2 + \tilde{\lambda}(\tilde{\pi}_1 - \Delta^2) - \tilde{\pi}_1 \Delta^2 - \sigma_m^2 \; \leq \; 0.$$

Any positive $\tilde{\lambda}$ which satisfies this quadratic inequality has to satisfy (27). $\qquad \square$

We observe that the bounds on the eigenvalues with the largest possible absolute value $\lambda_{-n}$ and $\lambda_m$ given by inequalities (24) and (27), respectively, are similar to those obtained for the original system (2) and given by (10) and (13), respectively. On the other hand, the bounds on small eigenvalues have been changed significantly by the presence of primal and dual regularization. A comparison of (25) and (11) shows that $\tilde{\lambda}_{-1}$ is much better bounded away from zero than $\lambda_{-1}$. Moreover, (25) allows for a selective regularization which should be applied only to the

smallest elements $\theta_j^{-1}$, see (22). The elements $\theta_j^{-1}$ which are large enough do not have to be perturbed at all and the corresponding terms of $R_{pj}$ can be set to zero thus reducing the perturbation to the original system. A comparison of (26) and (12) reveals that $\tilde{\lambda}_1$ is shifted away from zero by dual regularization. Indeed, $\sqrt{(\tilde{\pi}_n + \delta^2)^2 + 4\sigma_1^2} - \tilde{\pi}_n \geq \delta^2$ and therefore $\frac{1}{2}(\sqrt{(\tilde{\pi}_n + \delta^2)^2 + 4\sigma_1^2} - \tilde{\pi}_n + \delta^2) \geq \delta^2$. Summing up, $\min_j |\tilde{\lambda}_j| = \min\{-\tilde{\lambda}_{-1}, \tilde{\lambda}_1\}$ is much larger than that for the original system (15) and the condition number of the regularized augmented system (19) is therefore significantly smaller than that of the original system (2).

## 4    From Augmented System to Normal Equation

In this paper we are concerned with the use of iterative methods to solve the reduced KKT system (2) and its regularized form (19). A first decision to make is whether an iterative method should be applied to the indefinite augmented system formulation of KKT conditions such as (2) and (19) or to a reduced positive definite normal equation (Schur complement) formulation of these equations. The use of iterative methods such as conjugate gradients for indefinite systems may require some extra safeguards [17]. The subject has attracted a lot of attention in recent years [28, 31]. It requires care to be taken about preconditioners [20, 6] as well and/or special versions of the projected conjugate gradient method to be employed [23, 14]. The reduced positive definite normal equation formulation of (2) or (19) allows for a straightforward application of conjugate gradients and we will use it in this paper.

Normal equation formulations are obtained by elimination of $\Delta x$ from the augmented system and take the following forms

$$A(Q + \Theta^{-1})^{-1}A^T \Delta y = h \tag{31}$$

and

$$(A(Q + \Theta^{-1} + R_p)^{-1}A^T + R_d)\Delta y = h'', \tag{32}$$

for original system (2) and regularized system (19), respectively. We will start the analysis by providing bounds on the condition numbers of these two normal equation systems. We will refer once again to the notation introduced in earlier sections with singular values of $A$ given by (5), eigenvalues of $Q$ given by (6) and eigenvalues of $Q + \Theta^{-1}$ and $Q + \Theta^{-1} + R_p$ given by (7) and (21), respectively. We will denote the eigenvalues of (31) as

$$0 < \eta_1 \leq \eta_2 \leq \cdots \leq \eta_m \tag{33}$$

and the eigenvalues of (32) as

$$0 < \tilde{\eta}_1 \leq \tilde{\eta}_2 \leq \cdots \leq \tilde{\eta}_m. \tag{34}$$

**Lemma 4.1** *The eigenvalues of the normal equation matrix (31) satisfy:*

$$\sigma_1^2(\tau_n + \max_j \theta_j^{-1})^{-1} \leq \eta_1 \tag{35}$$

$$\eta_m \leq \sigma_m^2(\tau_1 + \min_j \theta_j^{-1})^{-1}. \tag{36}$$

We omit an easy proof of this Lemma. Instead, we will comment on a special case of linear programming, when $Q = 0$ and $\tau_1 = \tau_n = 0$. We observe that for well-centered iterates

combining Lemma 2.1 and Lemma 4.1 yields $\eta_1 \geq \sigma_1^2 \mathcal{O}(\mu)$ and $\eta_m \leq \sigma_m^2 \mathcal{O}(\mu^{-1})$, hence

$$\kappa_{NE} \leq \frac{\sigma_m^2 \mathcal{O}(\mu^{-1})}{\sigma_1^2 \mathcal{O}(\mu)} \leq \kappa^2(A) \mathcal{O}(\mu^{-2}). \tag{37}$$

With the extra assumption that the linear program is nondegenerate we can improve this bound. Indeed, for a nondegenerate problem the optimal partition (3) defines a set of columns $j \in \mathcal{B}$ corresponding to a nonsingular basis matrix and yields $\eta_1 \geq \sigma_1^2 \mathcal{O}(1)$. In such case $\kappa_{NE} \leq \kappa^2(A) \mathcal{O}(\mu^{-1})$. Clearly, as $\mu$ goes to zero the bounds of $\kappa_{NE}$ go to infinity and indicate that the original system (31) gets very ill-conditioned.

The regularized system (32) displays much better numerical properties.

**Lemma 4.2** *The eigenvalues of the normal equation matrix (32) satisfy:*

$$\sigma_1^2 (\tau_n + \max_j (\theta_j^{-1} + R_{pj}))^{-1} + \min_i R_{di} \leq \tilde{\eta}_1 \tag{38}$$

$$\tilde{\eta}_m \leq \sigma_m^2 (\tau_1 + \min_j (\theta_j^{-1} + R_{pj}))^{-1} + \max_i R_{di}. \tag{39}$$

Again, we skip a simple proof and analyze a special case of LP, when $Q = 0$ and $\tau_1 = \tau_n = 0$. Using (20), the bounds (38) and (39) are simplified

$$\delta^2 \leq \tilde{\eta}_1 \tag{40}$$

$$\tilde{\eta}_m \leq \sigma_m^2 \gamma^{-2} + \Delta^2, \tag{41}$$

and the condition number of the regularized normal equation matrix satisfies

$$\tilde{\kappa}_{NE} \leq \frac{\sigma_m^2 \gamma^{-2} + \Delta^2}{\delta^2} = \frac{\sigma_m^2 + \gamma^2 \Delta^2}{\gamma^2 \delta^2}. \tag{42}$$

The bound does not depend on $\mu$. It depends only on the level of regularization and $\sigma_m$, the largest singular value of $A$. It is reasonable to assume that $\sigma_m$ can be kept moderate, say, $\mathcal{O}(1)$. Indeed, this can be achieved with appropriate scaling of the optimization problem. The regularization terms $\gamma^2$ and $\delta^2$ could be kept large enough to guarantee a good bound on $\tilde{\kappa}_{NE}$.

One has to realise however that large regularization terms $\gamma^2$ and $\delta^2$ mean that (19) differs significantly from (2) and Newton direction computed from (19) may have little in common with the true primal-dual interior point direction obtained from (2). In our approach we look for a compromise solution where $\gamma^2$ and $\delta^2$ are small (to keep (19) close to (2)) but large enough to provide a reasonable bound in (42).

To conclude the analysis of this section we will recall a standard result regarding the speed of convergence of the conjugate gradient (CG) algorithm [24, eq. (2.15), p. 17] which guarantees that an error is systematically reduced at every CG iteration by a factor of $\frac{\kappa^{1/2}-1}{\kappa^{1/2}+1}$. Assuming a uniform dual regularization $R_{di} = \delta^2$, $\forall i$, that is $\delta^2 = \Delta^2$, and assuming that the regularization terms are small ($\delta \ll \sigma_m$ and $\gamma \ll \sigma_m$) from (42) we obtain

$$\tilde{\kappa}_{NE}^{1/2} \leq (1 + \frac{\sigma_m^2}{\gamma^2 \delta^2})^{1/2} \approx \frac{\sigma_m}{\gamma \delta} \gg 1,$$

and then deduce

$$\frac{\kappa^{1/2} - 1}{\kappa^{1/2} + 1} \approx \frac{1 - \frac{\gamma\delta}{\sigma_m}}{1 + \frac{\gamma\delta}{\sigma_m}} \approx 1 - 2\frac{\gamma\delta}{\sigma_m}. \tag{43}$$

The last equality shows that the larger the regularizations in the primal and dual spaces the better (smaller) the condition number of (19) is, and consequently the convergence of CG algorithm applied to (32) is faster.

Following our earlier comments we want to avoid over-regularization in order to keep (19) close to (2). Hence we will use very small regularizations and we will have to accept that $\gamma\delta \ll \sigma_m$. In this case the speed of convergence predicted by (43) is not enough to satisfy our needs. Indeed, so far we have only met the first objective of our approach: we have used the primal-dual regularization to guarantee that the condition number of (32) is bounded and independent of the barrier parameter $\mu$. We do not intend to apply the CG method directly to (32). The second important feature of our approach is the use of a suitable preconditioner for this regularized normal equation system. The design and analysis of this preconditioner are subject of next section.

## 5 Preconditioner for Regularized Normal Equation

We observe an important feature of the regularized normal equation matrix in (32)

$$G_R = A(Q + \Theta^{-1} + R_p)^{-1}A^T + R_d, \tag{44}$$

which has been captured by the bounds on its eigenvalues in Lemma 4.2.

All eigenvalues of this matrix remain in an interval determined by (38) and (39). For our choice of regularization (20) all small eigenvalues are clustered above the lower bound (40) and the large eigenvalues are spread and may reach up to the upper bound (41). Our preconditioner $P \in \mathcal{R}^{m \times m}$ attempts to identify the largest eigenvalues of $G_R$ and guarantee that $\kappa(P^{-1}G_R) \ll \kappa(G_R)$. This is achieved by computing a partial Cholesky decomposition of $G_R$ with complete pivoting choosing the largest (diagonal) pivots from $G_R$ and the updated Schur complements. We compute only a *partial* decomposition, that is we truncate the process after producing the first $k$ columns of Cholesky corresponding to $k$ largest pivots in $G_R$, where $k$ is a predetermined number such that $k \ll m$. We compute

$$G_R = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & S \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}, \tag{45}$$

where $L = \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix}$ is a trapezoidal matrix which contains the first $k$ columns of Cholesky factor of $G_R$ (with triangular $L_{11} \in \mathcal{R}^{k \times k}$ and $L_{21} \in \mathcal{R}^{(m-k) \times k}$ containing the remaining part of Cholesky columns), $D_L \in \mathcal{R}^{k \times k}$ is a diagonal matrix formed by $k$ largest pivots of $G_R$ and $S \in \mathcal{R}^{(m-k) \times (m-k)}$ is the Schur complement obtained after eliminating $k$ pivots.

If we had set $k = m$, (45) would have determined an exact Cholesky decomposition of $G_R = LD_LL^T$ obtained with complete pivoting. Since $G_R$ is a positive definite matrix complete diagonal pivoting is an unnecessary luxury, a very costly one if $G_R$ is a sparse matrix! Indeed, for

sparse $G_R$ modern implementations of Cholesky decomposition [15] ignore the values of pivots and choose their order based entirely on the grounds of preserving sparsity in $L$. Our setting of computing (45) is different: we truncate the Cholesky decomposition after producing merely $k$ ($k \ll m$) first columns and we insist on these columns to eliminate $k$ largest pivots from $G_R$. We accept the fact that for sparse $G_R$ the first $k$ columns of $L$ obtained using complete diagonal pivoting will be much denser than those which would have been computed if ordering for sparsity had been used.

An important feature of our approach is that partial Cholesky decomposition (45) is computed using an *implicit* process in which neither $G_R$ nor its Schur complements need to be fully formulated. Only the diagonal and selected columns of the Schur complements are calculated.

We will precondition (45) with

$$
P = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & D_S \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}, \tag{46}
$$

where $D_S$ is a diagonal of $S$. We will first analyse a simplified version of the preconditioner

$$
P_0 = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & I \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}, \tag{47}
$$

in which the diagonal of the Schur complement has been replaced with an identity matrix.

Let us order diagonal elements of $D_L$ and $D_S = diag(S)$ as follows

$$
\underbrace{d_1 \geq d_2 \geq \cdots \geq d_k}_{D_L} \geq \underbrace{d_{k+1} \geq d_{k+2} \geq \cdots \geq d_m}_{D_S}. \tag{48}
$$

The inequalities are the consequence of complete diagonal pivoting applied when decomposing $G_R$. We will exploit this ordering to argue that the condition number of the preconditioned matrix $P_0^{-1} G_R$ is much better than that of $G_R$.

Using the symmetric decomposition $P_0 = E_0 E_0^T$, where

$$
E_0 = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L^{1/2} & \\ & I \end{bmatrix} \tag{49}
$$

it is easy to derive

$$
E_0^{-1} G_R E_0^{-T} = \begin{bmatrix} I & \\ & S \end{bmatrix}. \tag{50}
$$

From Lemma 4.2 we deduce that the smallest eigenvalues of both $G_R$ and $E_0^{-1} G_R E_0^{-T}$ satisfy (38) and for our particular choice of regularizations they are both greater than or equal to $\delta^2$, see (40). As the largest eigenvalues of a positive definite matrix cannot exceed the trace of the matrix, we observe that

$$
\tilde{\eta}_{max}(G_R) \leq trace(G_R) \quad \text{and} \quad \tilde{\eta}_{max}(E_0^{-1} G_R E_0^{-T}) \leq k + trace(S).
$$

Our rule of choosing the largest possible pivots when computing partial Cholesky decomposition corresponds to a greedy heuristic which is likely to reduce the trace of the resulting Schur

complement at the fastest possible rate. Therefore we expect that the condition number of $E_0^{-1} G_R E_0^{-T}$ is decreasing rapidly with an increase of the number of columns $k$ allowed in the partial Cholesky decomposition. Actually we expect that $k + trace(S) \ll trace(G_R)$.

We have performed our analysis for a simpler preconditioner (47). Similarly we may decompose $P = EE^T$, where

$$E = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L^{1/2} & \\ & D_S^{1/2} \end{bmatrix} \tag{51}$$

and observe that

$$E^{-1} G_R E^{-T} = \begin{bmatrix} I & \\ & D_S^{-1/2} S D_S^{-1/2} \end{bmatrix}. \tag{52}$$

A comparison of (52) and (50) reveals that the difference between $P$ and $P_0$ is an extra Jacobi-type preconditioner applied to the Schur complement $S$.

The reduction of augmented system (19) to normal equation (32) makes sense only if matrix $Q + \Theta^{-1} + R_p$ is easily invertible. This is of course the case if we deal with a linear programming ($Q = 0$) or a separable quadratic programming ($Q$ is a diagonal matrix). We can extend the approach to the case when $Q$ is a band matrix or when $Q$ displays some other sparsity pattern which leads to a trivially invertible matrix. However, if $Q$ is a general sparse matrix then its inverse may actually be quite dense [15] and the reduction of augmented system to normal equation may lead to a serious loss of efficiency. In such case we would not perform the reduction but work with the augmented system.

To maintain the matrix-free feature of the approach we define $\bar{Q} = \text{diag}\{Q\} + \Theta^{-1} + R_p$, compute $\bar{G}_R = A\bar{Q}^{-1}A^T + R_d$ and its partial Cholesky decomposition

$$\bar{G}_R = \begin{bmatrix} \bar{L}_{11} & \\ \bar{L}_{21} & I \end{bmatrix} \begin{bmatrix} \bar{D}_L & \\ & \bar{S} \end{bmatrix} \begin{bmatrix} \bar{L}_{11}^T & \bar{L}_{21}^T \\ & I \end{bmatrix}. \tag{53}$$

As before, the off-diagonal elements of the Schur complement $\bar{S}$ should not be computed because they are dropped to produce the following approximation of $\bar{G}_R$

$$\bar{P} = \begin{bmatrix} \bar{L}_{11} & \\ \bar{L}_{21} & I \end{bmatrix} \begin{bmatrix} \bar{D}_L & \\ & \bar{D}_S \end{bmatrix} \begin{bmatrix} \bar{L}_{11}^T & \bar{L}_{21}^T \\ & I \end{bmatrix} = \bar{L}\bar{D}\bar{L}^T, \tag{54}$$

where $\bar{D}_S$ is a diagonal of $\bar{S}$. Finally, we define the preconditioner for the augmented system (19)

$$P_{aug} = \begin{bmatrix} I & \\ -A\bar{Q}^{-1} & \bar{L} \end{bmatrix} \begin{bmatrix} -\bar{Q} & \\ & \bar{D} \end{bmatrix} \begin{bmatrix} I & -\bar{Q}^{-1}A^T \\ & \bar{L}^T \end{bmatrix}. \tag{55}$$

The computation of such a preconditioner and its application to a vector requires only the multiplications with the Jacobian matrix and the diagonal of the Hessian matrix and the operations with the partial Cholesky matrix $\bar{P}$. Therefore $P_{aug}$ satisfies the requirements of matrix-free regime. Obviously, since the augmented system (19) and the preconditioner (55) are indefinite we cannot use the conjugate gradient algorithm any longer. Instead, we can use for example the LSQR algorithm of Paige and Saunders [30].

# 6 Implementation and Numerical Results

The matrix-free interior point method has been implemented in HOPDM [2, 9]. We use an infeasible primal-dual path following algorithm in which the Newton search directions are computed from the regularized system (19). We reduce this system to (32) and solve it using the preconditioned conjugate gradient (PCG) algorithm with the preconditioner (46). In all our runs the linear systems are solved to a relative accuracy $\varepsilon_{CG} = 10^{-3}$, that is, the CG terminates when the residual in equation (32) satisfies $\|r_q\| \leq \varepsilon_{CG}\|r_0\|$, where $r_q$ and $r_0$ are the residuals at iterations $q$ and zero, respectively. As an additional safeguard against the loss of efficiency, we interrupt the CG after a maximum of 20 steps regardless of the residual reduction achieved by that time. We have used small level of regularizations: $\gamma^2$ varied in our computations between $10^{-6}$ and $10^{-3}$, and $\delta^2$ was set up to $10^{-6}$.

Two variants of the preconditioner (46) have been implemented. If $A$ is a sparse matrix then the preconditioner is computed and stored as a sparse matrix as well, that is, $L$ is stored as a collection of sparse columns. Due to complete pivoting applied to $G_R$ when computing the partial Cholesky decomposition the columns of $L$ fill-in very quickly. (The sparsity pattern of a new column in Cholesky matrix is a result of merging the column of the Schur complement with all or a subset of Cholesky columns computed earlier.) Therefore we do not expect the approach to be competitive for sparse problems. The reason for solving a few sparse examples is to demonstrate the performance of the method on well-known test problems. If $A$ is a dense matrix or is available only implicitly as an operator providing results of multiplications $Au$ and $A^T v$ then dense partial Cholesky is computed.

An important feature of our approach is that preconditioner $P$ given by (46) and based on partial Cholesky decomposition is *implicit*. We do not formulate the matrix $G_R$ except for a few elements of it. The computations start from the diagonal of $G_R$ which is easy to calculate when $Q$ is diagonal or has another easily invertible form. Once the largest pivot in $G_R$ is determined the appropriate column of $G_R$ is computed and used to form the first column of $L$ in (45) and to update the diagonal of $G_R$ to become a diagonal $D_S$ of the first Schur complement. The largest element in $D_S$ is selected to be the next pivot to enter $D_L$, then the appropriate column of $G_R$ is formed and used to produce the next column in the partial Cholesky matrix $L$ and to update the diagonal of the Schur complement. The process continues until the predetermined number $k$ of columns of $L$ are calculated. To guarantee the efficiency of the approach we keep the rank of $L$ really small, $k = 2, 5, 10, 20$ or $50$ at most.

It is not easy to compare computational complexity of direct and matrix-free approaches for sparse case because the degree of sparsity and the speed of generating fill-in in partial Cholesky are difficult to quantify. However we can perform such a comparison for the dense case. We assume that $Q$ is diagonal (or zero), $A \in \mathcal{R}^{m \times n}$ is dense, partial Cholesky (45) contains $k$ columns and the preconditioned conjugate gradient algorithm performs $q$ steps. We report only dominating terms. Computing the preconditioner would require building a diagonal of $A(Q + \Theta^{-1} + R_p)^{-1}A^T + R_d$ which involves $2mn$ floating point operations (flops), building $k$ columns of this matrix costs $kmn$ flops and computing a trapezoidal $L$ in (45) costs $\frac{1}{2}mk^2$ flops. A single PCG iteration needs one multiplication with $L, L^T$ and $A(Q + \Theta^{-1} + R_p)^{-1}A^T + R_d$ so it costs $2mk + 2mn$ flops. Assuming that $q$ steps of PCG are performed the overall cost of

solving one linear system (32) is

$$Cost(PCG) = (2 + k + 2q)mn + (\frac{k^2}{2} + 2qk)m. \tag{56}$$

This number should be compared with the cost of a direct approach, which comprises building full $A(Q + \Theta^{-1} + R_p)^{-1}A^T + R_d$ in $\frac{1}{2}mn^2$ flops and computing its Cholesky decomposition which requires $\frac{1}{3}m^3$ flops giving the total cost

$$Cost(Direct) = \frac{1}{2}m^2n + \frac{1}{3}m^3. \tag{57}$$

It is clear that as long as $k \ll m$ and $q \ll m$ the matrix-free approach wins the competition.

The storage requirements of the preconditioner are limited to $k$ columns of length $m$ for the partial Cholesky matrix $L$ and one column of length $m$ to remember the diagonal of the Schur complement $S$. It is trivial to determine the maximum rank $k$ of the partial Cholesky decomposition as a function of memory available for the preconditioner. The *limited-memory* feature of our approach is an important advantage which should become essential when solving trully large problems.

We first report the results for sparse test examples coming from Kennington's collection of network problems `osa-07` and `osa-14` and train crew scheduling examples from Mittelmann's collection `rail507` and `rail2586`. Tables 1 and 2 are self-explanatory. We report in them the problem dimensions and the solution statistics, respectively. CPU times in bold indicate the winner which in all cases is the direct approach. Although matrix-free interior point method is slower on these problems it is not very far behind the direct approach, which is encouraging.

| Prob | Dimensions | | |
|------|------|------|------|
| | rows | columns | nonzeros |
| osa-07 | 1119 | 23949 | 167643 |
| osa-14 | 2338 | 52460 | 367220 |
| rail507 | 507 | 63009 | 472358 |
| rail2586 | 2586 | 920683 | 8929459 |

Table 1: Dimensions of sparse problems.

| Prob | Direct | | Matrix-Free IPM | | | | | |
|------|------|------|------|------|------|------|------|------|
| | HOPDM | | rank=10 | | rank=20 | | rank=50 | |
| | iters | time | iters | time | iters | time | iters | time |
| osa-07 | 12 | **1.46** | 17 | 4.21 | 15 | 3.56 | 15 | 3.85 |
| osa-14 | 15 | **5.22** | 15 | 8.50 | 15 | 8.66 | 18 | 11.84 |
| rail507 | 18 | **7.20** | 16 | 9.72 | 16 | 9.45 | 17 | 11.56 |
| rail2586 | 22 | **318.32** | 27 | 531.39 | 28 | 625.54 | 26 | 434.97 |

Table 2: Solution statistics for sparse problems.

In our second experiment two particular Netlib problems `fit1d` and `fit2d` are solved. These problems are very small for today's standards. Their constraint matrices are narrow but long ($m \ll n$). Direct approach is obviously very efficient for these problems having to deal with a sparse Cholesky decomposition of dimension merely 24 or 25. The matrix-free approach delivers comparable performance even with a partial Cholesky decomposition of rank $k = 2$ as can be seen in Tables 3 and 4. Since these problems are small and the timings may be inaccurate we do not indicate the winners with bold print.

| Prob | Dimensions | | |
|---|---|---|---|
| | rows | columns | nonzeros |
| fit1d | 24 | 1026 | 14430 |
| fit2d | 25 | 10500 | 138018 |

Table 3: Dimensions of Netlib problems.

| Prob | Direct | | Matrix-Free IPM | | | | | |
|---|---|---|---|---|---|---|---|---|
| | HOPDM | | rank=2 | | rank=5 | | rank=10 | |
| | iters | time | iters | time | iters | time | iters | time |
| fit1d | 16 | 0.07 | 12 | 0.10 | 12 | 0.10 | 11 | 0.08 |
| fit2d | 18 | 1.22 | 12 | 1.20 | 11 | 1.12 | 12 | 1.19 |

Table 4: Solution statistics for Netlib problems.

Finally, in the third experiment we solved randomly generated dense test examples which attempt to mimic basic pursuit problems [8]. Their dimensions are given in Table 5 and the solution statistics are reported in Table 6. For these problems the matrix-free approach is a clear winner. We could use a partial Cholesky decomposition preconditioner (46) with a small rank $k = 10, 20$ or 50 and we observed the fast convergence of the PCG algorithm. In early IPM iterations, PCG converged in 5-8 steps, in the "middle" of optimization, PCG usually required more iterations occasionally matching the limit of 20 steps, and towards the end of optimization, when IPM approached an optimal solution, the convergence of PCG improved again and the sufficient reduction of the residual was achieved in 10-15 steps on average.

| Prob | Dimensions | | |
|---|---|---|---|
| | rows | columns | nonzeros |
| Pb1 | 200 | 1000 | 200000 |
| Pb2 | 500 | 10000 | 5000000 |
| Pb3 | 1000 | 8000 | 8000000 |

Table 5: Dimensions of dense problems.

| Prob | Direct | | Matrix-Free IPM | | | | | |
|------|--------|--------|--------|--------|--------|--------|--------|--------|
|      | HOPDM  |        | rank=10 | | rank=20 | | rank=50 | |
|      | iters  | time   | iters  | time   | iters  | time   | iters  | time   |
| Pb1  | 5      | 1.41   | 5      | **0.27** | 6    | 0.45   | 5      | 0.67   |
| Pb2  | 6      | 69.83  | 6      | **10.02** | 6   | 11.58  | 6      | 15.87  |
| Pb3  | 6      | 222.06 | 5      | 13.15  | 5      | **11.71** | 5    | 20.03  |

Table 6: Solution statistics for dense problems.

## 7    Conclusions

We have discussed in this paper the matrix-free implementation of interior point method for linear and quadratic programming. The method allows for an implicit formulation of the optimization problem in which matrices $Q$ and $A$ do not have to be stored. Instead, they are only used to compute matrix-vector products. The design of the method relies on the use of iterative methods to compute Newton directions. The KKT systems are first regularized to guarantee a bounded condition number and then preconditioned with a partial Cholesky decomposition of the normal equation matrix. The preconditioner is computed without explicit use of matrices defining the optimization problem: it needs only matrix-vector products with the problem matrices. Moreover, the way it is computed easily allows for a limited-memory implementation. The method has been implemented and the preliminary computational results for small to medium scale problems demonstrate its feasibility. An implementation for large scale problems is an ongoing effort and we expect to report on it shortly in another paper.

## References

[1] G. AL-JEIROUDI AND J. GONDZIO, _Convergence analysis of inexact infeasible interior point method for linear optimization_, Journal of Optimization Theory and Applications, 141 (2009), pp. 231–247.

[2] A. ALTMAN AND J. GONDZIO, _Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization_, Optimization Methods and Software, 11-12 (1999), pp. 275–302.

[3] E. D. ANDERSEN, J. GONDZIO, C. MÉSZÁROS, AND X. XU, _Implementation of interior point methods for large scale linear programming_, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, 1996, pp. 189–252.

[4] S. BELLAVIA, _An inexact interior point method_, Journal of Optimization Theory and Applications, 96 (1998), pp. 109–121.

[5] M. BENZI, G. GOLUB, AND J. LIESEN, _Numerical solution of saddle point problems_, Acta Numerica, 14 (2005), pp. 1–137.

[6] L. BERGAMASCHI, J. GONDZIO, AND G. ZILLI, _Preconditioning indefinite systems in interior point methods for optimization_, Computational Optimization and Applications, 28 (2004), pp. 149–171.

[7] J. CASTRO AND J. CUESTA, *Quadratic regularizations in an interior point method for primal block-angular problems*, Tech. Rep. UPC-DEIO DR 2008-07, Dept. of Statistics and Oper. Res., Universitat Politecnica de Catalunya, 08034 Barcelona, 2008.

[8] S. CHEN, D. DONOHO, AND M. SAUNDERS, *Atomic decomposition by basis pursuit*, SIAM Journal on Scientific Computing, 20 (1998), pp. 33–61.

[9] M. COLOMBO AND J. GONDZIO, *Further development of multiple centrality correctors for interior point methods*, Computational Optimization and Applications, 41 (2008), pp. 277–305.

[10] F. CURTIS, J. NOCEDAL, AND A. WÄCHTER, *A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians*, SIAM Journal on Optimization, 20 (2009), pp. 1224–1249.

[11] M. D'APUZZO, V. D. SIMONE, AND D. D. SERAFINO, *On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods*, Computational Optimization and Applications, ?? (2009), pp. ???–??? Available on-line: `http://www.springerlink.com/content/100249/?k=Serafino`.

[12] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 400–408.

[13] I. I. DIKIN, *On the speed of an iterative process*, Upravlaemye Sistemy, 12 (1974), pp. 54–60.

[14] H. DOLLAR, N. GOULD, W. SCHILDERS, AND A. WATHEN, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point problems*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 170–189.

[15] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct methods for sparse matrices*, Oxford University Press, New York, 1987.

[16] M. FERRIS AND T. MUNSON, *Interior point methods for massive support vector machines*, SIAM Journal on Optimization, 13 (2003), pp. 783–804.

[17] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis Dundee 1975, G. Watson, ed., Springer-Verlag, Berlin, New York, 1976, pp. 73–89.

[18] R. W. FREUND AND F. JARRE, *A QMR-based interior-point algorithm for solving linear programs*, Mathematical Programming, 76 (1997), pp. 183–210.

[19] M. P. FRIEDLANDER AND P. TSENG, *Exact regularization of convex programs*, SIAM Journal on Optimization, 18 (2007), pp. 1326–1350.

[20] P. E. GILL, W. MURRAY, D. B. PONCELEÓN, AND M. A. SAUNDERS, *Preconditioners for indefinite systems arising in optimization*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 292–311.

[21] P. E. GILL, M. A. SAUNDERS, AND J. R. SHINNERL, *On the stability of Cholesky factorization for symmetric quasi-definite systems*, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 35–46.

[22] J. GONDZIO AND G. TORALDO (EDS), *Linear algebra issues arising in interior point methods*, Special issue of Computational Optimization and Applications, 36 (2007), pp. 137–341.

[23] N. I. M. GOULD, M. E. HRIBAR, AND J. NOCEDAL, *On the solution of equality constrained quadratic problems arising in optimization*, SIAM Journal on Scientific Computing, 23 (2001), pp. 1375–1394.

[24] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*, vol. 16 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[25] M. Kojima, N. Megiddo, and S. Mizuno, *A primal-dual infeasible-interior-point algorithm for linear programming*, Mathematical Programming, 61 (1993), pp. 263–280.

[26] J. Korzak, *Convergence analysis of inexact infeasible-interior-point-algorithm for solving linear progamming problems*, SIAM Journal on Optimization, 11 (2000), pp. 133–148.

[27] Z. Lu, R. D. S. Monteiro, and J. W. O'Neal, *An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming*, SIAM Journal on Optimization, 17 (2006), pp. 287–310.

[28] L. Lukšan and J. Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems*, Numerical Linear Algebra with Applications, 5 (1998), pp. 219–247.

[29] A. R. L. Oliveira and D. C. Sorensen, *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra and its Applications, 394 (2005), pp. 1–24.

[30] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Transactions on Mathematical Software, 8 (1982), pp. 43–71.

[31] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM Journal of Matrix Analysis and Applications, 24 (2002), pp. 368–391.

[32] T. Rusten and R. Winther, *A preconditioned iterative method for saddle point problems*, SIAM Journal of Matrix Analysis and Applications, 13 (1992), pp. 887–904.

[33] M. A. Saunders, *Cholesky-based methods for sparse least squares: The benefits of regularization*, in Linear and Nonlinear Conjugate Gradient-Related Methods, L. Adams and J. L. Nazareth, eds., SIAM, Philadelphia, 1996, pp. 92–100.

[34] D. Silvester and A. Wathen, *Fast iterative solution of stabilised Stokes systems part ii: Using general block preconditioners*, SIAM Journal on Numerical Analysis, 31 (1994), pp. 1352–1367.

[35] G. W. Stewart, *On scaled projections and pseudoinverses*, Linear Algebra and its Applications, 112 (1989), pp. 189–193.

[36] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.