Comput Optim Appl (2012) 51:457–480
DOI 10.1007/s10589-010-9361-3

Matrix-free interior point method

Jacek Gondzio

Received: 15 December 2009 / Published online: 14 October 2010

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

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-

J. Gondzio (✉)
School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Mayfield Road, Edinburgh EH9 3JZ, UK
e-mail: J.Gondzio@ed.ac.uk
url: http://maths.ed.ac.uk/~gondzio/

dual methods [26, 37] 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, 10]. 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 [17] or linear programs arising in basis pursuit [9], but it might also occur when solving certain sparse problems in which the patterns 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. [13] and has had a number of applications including those in the context of IPMs [4, 19, 27]. 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, 28]. 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 [23] 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 [25] 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 [25]. This is an ultimate goal when designing preconditioners for KKT systems arising in IPMs [12, 23]. 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 [11] which proposed a matrix-free algorithm for equality constrained nonlinear programming problems. There are essential differences between our approach and [11]. In this paper we deal with interior point methods and need to remedy the ill-conditioning introduced by the presence of barrier terms while [11] 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 [11] 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 [11, p. 1244] uses unpreconditioned MINRES.

The paper is organised as follows. In Sect. 2 we will introduce the problem and define the notation used. In Sect. 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 Sect. 4 we will analyse the normal equation form of the KKT system. In Sect. 5 we will introduce the preconditioner and perform the spectral

analysis of the preconditioned linear system. In Sect. 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 Sect. 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

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

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, $\Theta = XS^{-1}$ 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 Θ 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 for a linear program display a partition into a strongly complementary pair [37, p. 27]:

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

As a result the elements θ_j, $j \in \mathcal{B}$, go to infinity and the elements θ_j, $j \in \mathcal{N}$, go to zero. (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. For quadratic problems there may exist pairs which are not strictly complementary, which further complicates (3). The spread of eigenvalues makes the solution of (2) challenging. Surprisingly, direct methods do not seem to suffer from this property.

An old result of Dikin [14] (see also Stewart [36]) 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 Θ. 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 Θ and the resulting ill-conditioning of (2) makes the system intractable unless appropriately preconditioned [7, 21, 29, 30]. 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.

Assumption A1 *The optimal solution of* (1) *exists and satisfies*

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

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

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

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* β_b *and* β_c *are given constants.*

As a simple consequence of these assumptions we claim

Lemma 2.1 *If A1 and A2 are satisfied then*

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

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 $\max_j \theta_j^{-1} = \mathcal{O}(\mu^{-1})$. □

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 [33] (see also Lemma 2.1 in Silvester and Wathen [35]) we provide bounds for these eigenvalues.

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

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

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

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

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

Proof See the proof of Lemma 2.1 in [33]. □

Consequently, the eigenvalue with the largest absolute value satisfies

$$\begin{aligned}
\max_j |\lambda_j| &= \max\{-\lambda_{-n}, \lambda_m\} \\
&\le \max\left\{\frac{1}{2}\left(\pi_n + \sqrt{\pi_n^2 + 4\sigma_m^2}\right), \frac{1}{2}\left(\sqrt{\pi_1^2 + 4\sigma_m^2} - \pi_1\right)\right\} \\
&\le \max\{\pi_n + \sigma_m, \sigma_m\} \\
&= \pi_n + \sigma_m \ \le \ \tau_n + \max_j \theta_j^{-1} + \sigma_m.
\end{aligned} \tag{14}$$

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

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

Hence the eigenvalue with the smallest absolute value satisfies

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

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

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

We will slightly abuse the notation and write $\min_j |\lambda_j| \geq \min\{\pi_1, \frac{\sigma_1^2}{\pi_n}\}$. Both terms on the right hand side of this inequality may get very small. From (8) $\pi_1 \geq \tau_1 + \min_j \theta_j^{-1}$ and $\frac{\sigma_1^2}{\pi_n} \geq \frac{\sigma_1^2}{\tau_n + \max_j \theta_j^{-1}}$. 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

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

Even if A is well conditioned and σ_1 and σ_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 Θ^{-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 μ going to zero, κ_{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 [22], Saunders [34] 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

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

where γ and δ are (small) positive terms and $p \in \mathcal{R}^m$ is a free variable. It is intuitive that as long as γ and δ 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:

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

where primal and dual positive definite diagonal regularization matrices $R_p \in \mathcal{R}^{n \times n}$ and $R_d \in \mathcal{R}^{m \times m}$ and primal and dual reference points in proximal terms $x_0 \in \mathcal{R}^n$ and $y_0 \in \mathcal{R}^m$, respectively can be chosen dynamically. The elements on the diagonal of the regularization matrices are small and the reference points x_0 and y_0 change at each iteration and are set to the current primal and dual iterates, respectively, see [2] for more detail. The following regularized KKT system is solved at each iteration of the primal-dual interior point method

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

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 [8, 20] 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 R_{pj} and R_{di} are the jth and ith elements of diagonal matrices R_p and R_d, respectively and Γ and Δ 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*:

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

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

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

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

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

$$u^T\tilde{Q}u + \tilde{\lambda}u^Tu = \tilde{\lambda}v^Tv - v^TR_dv.$$

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

$$\tilde{\pi}_1u^Tu \;\leq\; u^T\tilde{Q}u \;\leq\; -\tilde{\lambda}u^Tu,$$

which yields (25).

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

$$v^TA(\tilde{Q} + \tilde{\lambda}I)^{-1}A^Tv + v^TR_dv = \tilde{\lambda}v^Tv. \tag{30}$$

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

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

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

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

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). □

We observe that the bounds on the eigenvalues with the largest possible absolute value λ_{-n} and λ_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 λ_{-1}. Moreover, (25) allows for a selective regularization which should be applied only to the smallest elements θ_j^{-1}, see (22). The elements θ_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 [18]. The subject has attracted a lot of attention in recent years [29, 32]. It requires care to be taken about preconditioners [7, 21] as well and/or special versions of the projected conjugate gradient method to be employed [15, 24]. 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 Δx 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*:

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

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

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 μ goes to zero the bounds of κ_{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*:

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

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

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 μ. It depends only on the level of regularization and σ_m, the largest singular value of A. It is reasonable to assume that σ_m can be kept moderate, say, $\mathcal{O}(1)$. Indeed, this can be achieved with appropriate scaling of the optimization problem. The regularization terms γ^2 and δ^2 could be kept large enough to guarantee a good bound on $\tilde{\kappa}_{NE}$.

One has to realise however that large regularization terms γ^2 and δ^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 γ^2 and δ^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 [25, 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

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

and then deduce

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

The last approximation 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 μ. 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 $G_R = L D_L L^T$ 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 [16] 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 a special case of LP they are both greater than or equal to δ^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 [16] 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 [31].

We summarize this section by giving a complete algorithm of the matrix-free interior point method on the next page.

The computation of partial Cholesky preconditioner requires access to the diagonal of $\bar{G}_R = A\bar{Q}^{-1}A^T + R_d$ and to selected k columns of this matrix. The computation of infeasibilities ξ_p and ξ_d, the right-hand-side vectors in the reduced KKT system (19), and matrix-vector multiplications in the iterative scheme need to perform numerous matrix-vector multiplications with matrices A, A^T and Q which can all be executed as implicit operations.

The primal-dual Newton direction is computed using multiple centrality correctors [10]. The use of centrality correctors is not an obvious choice in the implementation which relies on iterative methods to solve the linear equations because the effort to compute the preconditioner is relatively small compared to the one needed for every extra backsolve. However, the correctors help to maintain well-centered iterates and stabilize the behaviour of interior point method. Therefore one or two centrality correctors are allowed per interior point iteration in the matrix-free method.

The following default values of parameters are used in our implementation: $k = 20$, $\texttt{maxItKM} = 20$, $\varepsilon_{KM} = 10^{-4}$, $\varepsilon_p = 10^{-4}$, $\varepsilon_d = 10^{-4}$, $\varepsilon_o = 10^{-6}$. The default primal and dual regularizations are set to $\gamma^2 = 10^{-8}$ and $\delta^2 = 10^{-6}$, respectively. It is tempting to use stronger default regularizations $\gamma^2 = 10^{-6}$ and $\delta^2 = 10^{-4}$ to produce a much better conditioned $\bar{G}_R$ and improve the efficiency and accuracy of the preconditioned iterative solver. However, too strong regularizations might hamper the convergence of the primal dual interior point method.

When the regularized indefinite factorization of (19) is computed in the direct approach [2], both primal and dual regularizations are defined dynamically. The matrix-free method works with the normal equation matrix and therefore needs to compute $\bar{Q} = \text{diag}\{Q\} + \Theta^{-1} + R_p$ before defining the necessary parts (a diagonal or a column) of $\bar{G}_R = A\bar{Q}^{-1}A^T + R_d$. Thus the primal regularization is always fixed to $R_p = \gamma^2 I$. The dual regularization R_d is defined dynamically during the computation of the partial Cholesky decomposition of $\bar{G}_R$. For all stable pivots a default value of $R_{di} = \delta^2$ is used. However, for pivots which fall dangerously close to zero ($d_{ii} \leq 10^{-6}$) a stronger regularization term is applied $R_{di} = 10^{-4}$.

MATRIX-FREE INTERIOR POINT METHOD

Input

Define: $\bar{Q} = \text{diag}\{Q\} + \Theta^{-1} + R_p$ and $\bar{G}_R = A\bar{Q}^{-1}A^T + R_d$;
matrix-vector operators: $u \mapsto Au$, $v \mapsto A^T v$, $u \mapsto Qu$ and functions which compute the diagonal and a single column of $\bar{G}_R = A\bar{Q}^{-1}A^T + R_d$: $(x, s) \mapsto \text{diag}\{\bar{G}_R\}$ and $(x, s) \mapsto column_j\{\bar{G}_R\}$, respectively.

Parameters

k rank of the partial Cholesky matrix;
`maxItKM` maximum number of iterations allowed for the Krylov subspace method;
ε_{KM} relative accuracy tolerance in the Krylov method: $\frac{(r^q)^T r^q}{(r^0)^T r^0} \leq \varepsilon_{KM}$;
ε_p, ε_d, ε_o primal feasibility, dual feasibility and optimality tolerances:
IPM stops when $\frac{\|\xi_p^l\|}{1+\|b\|} \leq \varepsilon_p$, $\frac{\|\xi_d^l\|}{1+\|c\|} \leq \varepsilon_d$ and $\frac{(x^l)^T s^l/n}{1+|c^T x^l + 1/2(x^l)^T Q x^l|} \leq \varepsilon_o$.
γ: default primal regularization;
δ: default dual regularization.

Initialize IPM

iteration counter $l = 0$, primal-dual point $x^0 > 0,\ y^0 = 0,\ s^0 > 0$;
barrier parameter $\mu^0 = (x^0)^T s^0/n$;
primal and dual infeasibilities $\xi_p^0 = b - Ax^0$ and $\xi_d^0 = c - A^T y^0 - s^0 + Qx^0$.

Interior Point Method

while $(\frac{\|\xi_p^l\|}{1+\|b\|} > \varepsilon_p$ or $\frac{\|\xi_d^l\|}{1+\|c\|} > \varepsilon_d$ or $\frac{(x^l)^T s^l/n}{1+|c^T x^l + 1/2(x^l)^T Q x^l|} > \varepsilon_o)$ **do**

Update (reduce) the barrier $\mu^{l+1} = 0.1\mu^l$;
Define $R_p = \gamma^2 I$, compute $\Theta = X^l(S^l)^{-1}$ and $\bar{Q} = \text{diag}\{Q\} + \Theta^{-1} + R_p$;
Define proximal points $x_0 = x^l$ and $y_0 = y^l$;
Compute partial Cholesky decomposition of $\bar{G}_R$ (53);
Define the preconditioner:
for LP or separable QP use (54);
for nonseparable QP use (55).
Solve the reduced KKT system (19). Use the Krylov subspace method with the partial Cholesky preconditioner. Terminate when relative accuracy tolerance drops below ε_{KM} or the number of Krylov method iterations reaches the limit `maxItKM`.

Find $\alpha_P = \max\{\alpha : x^l + \alpha\Delta x \geq 0\}$ and $\alpha_D = \max\{\alpha : s^l + \alpha\Delta s \geq 0\}$;
Set $\alpha_P := 0.99\alpha_P$ and $\alpha_D := 0.99\alpha_D$;
Make step
$x^{l+1} = x^l + \alpha_P \Delta x$;
$y^{l+1} = y^l + \alpha_D \Delta y$;
$s^{l+1} = s^l + \alpha_D \Delta s$.
Compute the infeasibilities: $\xi_p^{l+1} = b - Ax^{l+1}$ and $\xi_d^{l+1} = c - A^T y^{l+1} - s^{l+1} + Qx^{l+1}$;
Update the iteration counter: $l := l + 1$.

while

end-

6 Implementation and numerical results

The matrix-free interior point method has been implemented in HOPDM [2, 10]. We use an infeasible primal-dual path following algorithm in which the Newton search directions are computed from the regularized system (19). For linear and separable quadratic problems we reduce this system to (32) and solve it using the preconditioned conjugate gradient (PCG) algorithm with the preconditioner (54). For nonseparable quadratic problems we apply Krylov subspace method directly to the regularized system (19) and precondition it with (55). In most of our runs the linear systems are solved to a relative accuracy $\varepsilon_{KM} = 10^{-4}$, that is, the iterative solver terminates when the residual in the equation satisfies $\|r^q\|^2 \leq \varepsilon_{KM}\|r^0\|^2$, 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 iterative method after a maximum of $\texttt{maxItKM} = 20$ steps regardless of the residual reduction achieved by that time. For nonseparable QPs we used $\varepsilon_{KM} = 10^{-8}$ and $\texttt{maxItKM} = 100$.

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. For linear problems and separable quadratic problems we set $k = 2, 5, 10, 20$ or 50 at most. Such low-rank approximations are not accurate enough to tackle nonseparable quadratic problems. This is the case because for nonseparable QPs the normal equation matrix G_R is approximated with $\bar{G}_R$ in which the matrix Q is already approximated with merely its diagonal. In fact, a low-rank approximation of the (already approximate) normal equation matrix $\bar{G}_R$ is computed. In consequence we work with approximations of both the (1, 1) block and the Schur complement of (19) when constructing the preconditioner (55) and the partial Cholesky of small rank is unable to

provide sufficient quality of the preconditioner. For nonseparable quadratic problems we therefore set $k = 50$, 100 or 200. Let us observe that the spectral analysis of inexact constraint preconditioner [6] applies to (55). It is worth adding however that the inexact constraint preconditioner [6] offers more flexibility in dropping off-diagonal nonzero entries from Q and dropping nonzero entries from A and allows to produce more sparse preconditioner than the one presented in this paper and based on partial Cholesky factorization with complete pivoting.

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

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

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 truly large problems.

We have performed several tests of the matrix-free interior point method. We have used a Dell Precision M60 laptop computer with a 2 GHz (single core) processor and 2 GB of RAM running Linux. HOPDM was compiled with the GNU Fortran compiler `g77` with optimization option `-O2`. We have used the same termination criteria for direct and for matrix-free approaches and set the following tolerances: $\varepsilon_p = 10^{-4}$, $\varepsilon_d = 10^{-4}$ and $\varepsilon_0 = 10^{-6}$.

We first report the results for sparse test examples coming from Kennington's collection of network problems `osa-07` and `osa-14` (http://www.netlib.org/lp/data/kennington/) and set covering problems from the Italian railroad from Mittelmann's

Table 1 Dimensions of sparse problems

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

Table 2 Solution statistics for sparse problems

Prob	Direct		Matrix-free IPM					
	HOPDM		rank = 10		rank = 20		rank = 50	
	iters	time	iters	time	iters	time	iters	time
osa-07	11	**1.41**	17	4.21	15	3.56	15	3.85
osa-14	13	**4.71**	15	8.50	15	8.66	18	11.84
rail507	15	**6.29**	16	9.72	16	9.45	17	11.56
rail2586	20	**293.23**	27	531.39	28	625.54	26	434.97

Table 3 Dimensions of Netlib problems

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

collection `rail507` and `rail2586` (http://plato.asu.edu/ftp/lptestset/rail/). 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.

In our second experiment two problems from Netlib (http://www.netlib.org/lp/data/) `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.

In our third experiment, a few nonseparable quadratic programming problems were solved. We chose problems with significant quadratic term as shown by the number of off-diagonal nonzero entries in the triangular part of matrix Q reported

Table 4 Solution statistics for Netlib problems

Prob	Direct		Matrix-free IPM					
	HOPDM		rank = 2		rank = 5		rank = 10	
	iters	time	iters	time	iters	time	iters	time
fit1d	12	0.06	12	0.10	12	0.10	11	0.08
fit2d	13	1.01	12	1.20	11	1.12	12	1.19

Table 5 Dimensions of nonseparable QP problems

Prob	Dimensions			
	rows	columns	nonzeros A	nonzeros Q
scagr25	471	500	2029	100
25fv47	820	1571	11127	59053
ship12l	1151	5427	21597	60205

Table 6 Solution statistics for nonseparable QP problems

Prob	Direct		Matrix-free IPM					
	HOPDM		rank = 50		rank = 100		rank = 200	
	iters	time	iters	time	iters	time	iters	time
scagr25	11	**0.04**	18	0.44	13	0.36	11	0.18
25fv47	19	**1.36**	63	10.12	38	9.31	27	11.93
ship12l	9	1.20	10	0.61	10	**0.60**	10	0.89

in the last column of Table 5. The problems come from the Maros and Mészáros' collection of quadratic programming test examples (http://www.sztaki.hu/~meszaros/public_ftp/qpdata/brunel/). We needed to change the default settings of the matrix-free method for these problems. Namely, we allow the rank of partial Cholesky matrix to vary between 50 and 200. Additionally, we increased the accuracy requirement of the Krylov subspace solver to $\varepsilon_{KM} = 10^{-8}$ and the limit of iterations to $\texttt{maxItKM} = 100$. The results collected in Table 6 show that with the rank equal to 50, the matrix-free method struggled to reach optimal solution for two problems `scagr25` and `25fv47` and a higher rank of partial Cholesky preconditioner was required.

Finally, in the fourth experiment we solved randomly generated dense test examples which attempt to mimic basic pursuit problems [9]. Their dimensions are given in Table 7 and the solution statistics are reported in Table 8. 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 occa-

Table 7 Dimensions of dense problems

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

Table 8 Solution statistics for 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	5	61.77	6	**10.02**	6	11.58	6	15.87
Pb3	5	201.62	5	13.15	5	**11.71**	5	20.03

sionally 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.

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. Al-Jeiroudi, G., Gondzio, J.: Convergence analysis of inexact infeasible interior point method for linear optimization. J. Optim. Theory Appl. **141**, 231–247 (2009)

2. Altman, A., Gondzio, J.: Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. Optim. Methods Softw. **11–12**, 275–302 (1999)
3. Andersen, E.D., Gondzio, J., Mészáros, C., Xu, X.: Implementation of interior point methods for large scale linear programming. In: Terlaky, T. (ed.) Interior Point Methods in Mathematical Programming, pp. 189–252. Kluwer Academic, Dordrecht (1996).
4. Bellavia, S.: An inexact interior point method. J. Optim. Theory Appl. **96**, 109–121 (1998)
5. Benzi, M., Golub, G., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
6. Bergamaschi, L., Gondzio, J., Venturin, M., Zilli, G.: Inexact constraint preconditioners for linear system arising in interior point methods. Comput. Optim. Appl. **36**, 137–147 (2007)
7. Bergamaschi, L., Gondzio, J., Zilli, G.: Preconditioning indefinite systems in interior point methods for optimization. Comput. Optim. Appl. **28**, 149–171 (2004)
8. Castro, J., Cuesta, J.: Quadratic regularizations in an interior point method for primal block-angular problems. Math. Program. (2010). Published online 19 February 2010. doi:10.1007/s10107-010-0341-2
9. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**, 33–61 (1998)
10. Colombo, M., Gondzio, J.: Further development of multiple centrality correctors for interior point methods. Comput. Optim. Appl. **41**, 277–305 (2008)
11. Curtis, F., Nocedal, J., Wächter, A.: A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians. SIAM J. Optim. **20**, 1224–1249 (2009)
12. D'Apuzzo, M., De Simone, V., Di Serafino, D.: On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods. Comput. Optim. Appl. **45**, 283–310 (2010)
13. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact Newton methods. SIAM J. Numer. Anal. **19**, 400–408 (1982)
14. Dikin, I.I.: On the speed of an iterative process. Uprav. Sist. **12**, 54–60 (1974)
15. Dollar, H., Gould, N., Schilders, W., Wathen, A.: Implicit-factorization preconditioning and iterative solvers for regularized saddle-point problems. SIAM J. Matrix Anal. Appl. **28**, 170–189 (2006)
16. Duff, I.S., Erisman, A.M., Reid, J.K.: Direct Methods for Sparse Matrices. Oxford University Press, New York (1987).
17. Ferris, M., Munson, T.: Interior point methods for massive support vector machines. SIAM J. Optim. **13**, 783–804 (2003)
18. Fletcher, R.: Conjugate gradient methods for indefinite systems. In: Watson, G. (ed.) Numerical Analysis Dundee 1975, pp. 73–89. Springer, Berlin, New York (1976).
19. Freund, R.W., Jarre, F.: A QMR-based interior-point algorithm for solving linear programs. Math. Program. **76**, 183–210 (1997)
20. Friedlander, M.P., Tseng, P.: Exact regularization of convex programs. SIAM J. Optim. **18**, 1326–1350 (2007)
21. Gill, P.E., Murray, W., Ponceleón, D.B., Saunders, M.A.: Preconditioners for indefinite systems arising in optimization. SIAM J. Matrix Anal. Appl. **13**, 292–311 (1992)
22. Gill, P.E., Saunders, M.A., Shinnerl, J.R.: On the stability of Cholesky factorization for symmetric quasi-definite systems. SIAM J. Matrix Anal. Appl. **17**, 35–46 (1996)
23. Gondzio, J., Toraldo, G. (eds.): Linear algebra issues arising in interior point methods. Comput. Optim. Appl. **36**, 137–341 (2007). Special issue
24. Gould, N.I.M., Hribar, M.E., Nocedal, J.: On the solution of equality constrained quadratic problems arising in optimization. SIAM J. Sci. Comput. **23**, 1375–1394 (2001)
25. Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations. Frontiers in Applied Mathematics, vol. 16. SIAM, Philadelphia (1995).
26. Kojima, M., Megiddo, N., Mizuno, S.: A primal-dual infeasible-interior-point algorithm for linear programming. Math. Program. **61**, 263–280 (1993)
27. Korzak, J.: Convergence analysis of inexact infeasible-interior-point-algorithm for solving linear programming problems. SIAM J. Optim. **11**, 133–148 (2000)
28. Lu, Z., Monteiro, R.D.S., O'Neal, J.W.: An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming. SIAM J. Optim. **17**, 287–310 (2006)
29. Lukšan, L., Vlček, J.: Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems. Numer. Linear Algebra Appl. **5**, 219–247 (1998)
30. Oliveira, A.R.L., Sorensen, D.C.: A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. Linear Algebra Appl. **394**, 1–24 (2005)

31. Paige, C.C., Saunders, M.A.: LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. **8**, 43–71 (1982)
32. Rozloznik, M., Simoncini, V.: Krylov subspace methods for saddle point problems with indefinite preconditioning. SIAM J. Matrix Anal. Appl. **24**, 368–391 (2002)
33. Rusten, T., Winther, R.: A preconditioned iterative method for saddle point problems. SIAM J. Matrix Anal. Appl. **13**, 887–904 (1992)
34. Saunders, M.A.: Cholesky-based methods for sparse least squares: The benefits of regularization. In: Adams, L., Nazareth, J.L. (eds.) Linear and Nonlinear Conjugate Gradient-Related Methods. pp. 92–100. SIAM, Philadelphia (1996).
35. Silvester, D., Wathen, A.: Fast iterative solution of stabilised Stokes systems part ii: Using general block preconditioners. SIAM J. Numer. Anal. **31**, 1352–1367 (1994)
36. Stewart, G.W.: On scaled projections and pseudoinverses. Linear Algebra Appl. **112**, 189–193 (1989)
37. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM, Philadelphia (1997)