Open access • Journal Article • DOI:10.1007/S10589-008-9226-1

On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods

Marco D'Apuzzo, Valentina De Simone, Daniela di Serafino

Institutions: Seconda Università degli Studi di Napoli

Published on: 01 Mar 2010 - Computational Optimization and Applications (Springer US)

Topics: Numerical linear algebra, Interior point method, Karush–Kuhn–Tucker conditions and Linear algebra

DIPARTIMENTO DI MATEMATICA
Complesso universitario, via Vivaldi - 81100 Caserta

ON MUTUAL IMPACT OF NUMERICAL LINEAR ALGEBRA AND LARGE-SCALE OPTIMIZATION WITH FOCUS ON INTERIOR POINT METHODS

Marco D'Apuzzo[1], Valentina De Simone[1] and Daniela di Serafino[1]

PREPRINT: n. 1, impaginato nel mese di marzo 2008
CLASSIFICAZIONE AMS: 65F10, 65K05, 90C30, 90C06

[1] Dipartimento di Matematica, Seconda Università di Napoli, via Vivaldi 43, I-81100 Caserta, Italy

On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods

Marco D'Apuzzo, Valentina De Simone, Daniela di Serafino

Department of Mathematics, Second University of Naples, Caserta, Italy
E-mail: {marco.dapuzzo,valentina.desimone,daniela.diserafino}@unina2.it

Abstract

The solution of KKT systems is ubiquitous in optimization methods and often dominates the computation time, especially when large-scale problems are considered. Thus, the effective implementation of such methods is highly dependent on the availability of effective linear algebra algorithms and software, that are able, in turn, to take into account specific needs of optimization. In this paper we discuss the mutual impact of linear algebra and optimization, focusing on interior point methods and on the iterative solution of the KKT system. Three critical issues are addressed: preconditioning, termination control for the inner iterations, and inertia control.

Keywords: large-scale optimization, interior point methods, KKT system, constraint preconditioners, adaptive stopping criteria, inertia control.

1 Introduction

The strong interplay between numerical linear algebra and optimization has been evident for a long time. Much progress in numerical linear algebra has been spurred by the need of solving linear systems with special features in the context of optimization, and many optimization codes have benefited, in terms of both efficiency and robustness, from advances in numerical linear algebra, coming out also from needs in other fields of scientific computing. This interplay is clearly recognized in the textbook [41] by Gill et al., where a presentation of numerical optimization and numerical linear algebra techniques is provided, highlighting the relations between the two fields in the broader context of scientific computing. A general discussion of the role of numerical linear algebra in optimization in the 20th century is in the essay by O'Leary [65]. She

points out that in any optimization algorithm the work involved "in generating points approximating an optimal point" is often "dominated by linear algebra, usually in the form of solution of a linear system or least squares problem and updating of matrix information". By looking at the connections between the advances in numerical linear algebra and in optimization, O'Leary comes to the conclusion that there is a symbiosis between the two fields and foresees that it will continue in the current century.

In our opinion, this symbiosis is getting stronger and stronger, especially in the context of large-scale optimization problems, where the solution of linear algebra problems often dominates the computation time. A clear signal of this trend is also the organization of events gathering people working in the two fields (see, e.g., [82, 83, 84]). The aim of this paper is just to discuss the mutual impact of recent developments in numerical linear algebra and optimization, focusing on the solution of large-scale nonlinear optimization problems and on iterative linear algebra techniques, where much progress has been made in the last years (see [2, 5, 48, 74] and the references therein).

Thus, we consider the following general nonlinear optimization problem:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{s. t.} & c_{\mathcal{I}}(x) \geq 0, \\ & c_{\mathcal{E}}(x) = 0, \end{array} \tag{1}$$

where $f : \Re^n \longrightarrow \Re$ is the objective function, $c_{\mathcal{I}} : \Re^n \longrightarrow \Re^{m_{\mathcal{I}}}$ and $c_{\mathcal{E}} : \Re^n \longrightarrow \Re^{m_{\mathcal{E}}}$ are the inequality and equality constraints, respectively, and $m_{\mathcal{I}} + m_{\mathcal{E}} \leq n$. We assume that $f, c_{\mathcal{I}}$ and $c_{\mathcal{E}}$ are twice continuously differentiable and some constraint qualification holds, such as the Linear Independence or the Mangasarian-Fromovitz one, so that a solution of problem (1) satisfies the Karush-Kuhn-Tucker (KKT) conditions (see, e.g., [64, Chapter 12]).

General problems of type (1) are often hard to solve and in the last years many research efforts have been devoted to improve optimization algorithms with the double goal of success and high performance over a wide range of problems. However, no single approach has resulted uniformly robust and efficient in tackling nonlinear optimization problems. Among the various methods developed for such problems, two approaches have emerged: Sequential Quadratic Programming (SQP) and Interior Point (IP). Both approaches are based on the idea of moving toward a (local) solution of problem (1) by approximately solving a sequence of "simpler" problems; of course, they strongly differ for the characteristics of these problems and the strategies used to solve them.

As explained in the next section, SQP and IP methods have a common linear algebra kernel; at each iteration they require the solution of the so-called *KKT linear system*:

$$\begin{pmatrix} H & -J^T \\ -J & -D \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}, \tag{2}$$

where $H \in \Re^{n\times n}$, $D \in \Re^{m\times m}$ and $J \in \Re^{m\times n}$, with $m \leq m_{\mathcal{I}} + m_{\mathcal{E}}$. The matrix H is usually (an approximation of) the Hessian of the Lagrangian of problem (1) at the current iteration, and hence it is symmetric and possibly indefinite, J is the Jacobian of some or all the constraints, and D is diagonal and positive semidefinite, possibly null. Note that, in large-scale problems, system (2) is usually sparse. In the following, the matrix of this system is denoted by $\mathcal{K}$ and is called *KKT matrix*.

The solution of KKT systems is often the most computationally expensive task in SQP and IP methods. Thus, the effective implementation of such methods is highly dependent on the availability of effective linear algebra algorithms and software, that are able, in turn, to take into account specific needs of the optimization solvers. Note that KKT systems arise also in the solution of other optimization problems, such as least squares ones, and in the more general context of saddle-point problems [7], therefore very large interest is devoted to this subject. For these reasons, our discussion on linear algebra and optimization is centred on the KKT system.

The remainder of the paper is organized as follows. In Section 2 we show how the KKT system arises in SQP and IP methods for solving problem (1). In Section 3 we report main properties of the KKT matrix, which must be taken into account in solving the related system in the context of optimization. In Section 4 fundamental issues in solving the KKT system are discussed, focusing on interior point methods and iterative linear algebra solvers; preconditioning, adaptive termination of the inner iterations and inertia control are addressed. Finally, in Section 5 we report some experiences in the application of iterative linear algebra techniques in the context of a potential reduction method for quadratic programming. Concluding remarks are given in Section 6.

2 Linear algebra in SQP and IP methods

In order to show how KKT systems arise in SQP and IP methods, we give a sketch of both, presenting only their basic ideas applied to the general problem (1). A deeper discussion of these methods is outside the scope of the paper; many details can be found in the surveys [33, 48, 50] and in the references therein.

Henceforth we use the following notations: $g(x) = \nabla f(x)$ (gradient of the objective function), $L(x, y) = f(x) - y_{\mathcal{I}}^T c_{\mathcal{I}}(x) - y_{\mathcal{E}}^T c_{\mathcal{E}}(x)$ (Lagrangian function of the problem), $H(x, y) \approx \nabla_{xx} L(x, y)$ (approximation to) the Hessian of the Lagrangian function with respect to x, $J_{\mathcal{I}}(x) = \nabla c_{\mathcal{I}}(x)$ and $J_{\mathcal{E}}(x) = \nabla c_{\mathcal{E}}(x)$ (Jacobian matrices of the inequality and equality constraints, respectively). Furthermore, the identity matrix is denoted by I and the vector of all 1's by e; for any vector v the diagonal matrix $\mathrm{diag}(v)$ is denoted by the corresponding uppercase letter V, and, for any vectors v and w, (v, w) is a shorthand for $(v^T, w^T)^T$.

The basic idea of a SQP method is to generate a sequence of approximate (local)

solutions of problem (1), by solving, at each iteration, a Quadratic Programming (QP) problem, such as

$$\begin{array}{ll} \underset{\delta x}{\text{minimize}} & q(\delta x) \equiv \delta x^T g(x) + \frac{1}{2}\delta x^T H(x,y)\delta x \\ \text{s. t.} & c_{\mathcal{I}}(x) + J_{\mathcal{I}}(x)\delta x \geq 0, \\ & c_{\mathcal{E}}(x) + J_{\mathcal{E}}(x)\delta x = 0, \end{array} \tag{3}$$

where $q(\delta x)$ is a quadratic (e.g. Quasi-Newton) approximation of $L(x, y)$ and δx is a search direction. A commonly used strategy to solve the SQP subproblem is based on the active-set approach. It tries to predict the inequality constraints that are active at the solution and solves an equality constrained optimization problem, hence it is called Sequential Equality-constrained Quadratic Programming (SEQP). The quadratic problem (3) reduces to

$$\begin{array}{ll} \underset{\delta x}{\text{minimize}} & q(\delta x) \equiv \delta x^T g(x) + \frac{1}{2}\delta x^T H(x,y)\delta x \\ \text{s.t.} & c_{\mathcal{A}}(x) + J_{\mathcal{A}}(x)\delta x = 0, \end{array} \tag{4}$$

where $\mathcal{A}$ is an estimate of the active set at x, $c_{\mathcal{A}}$ are the constraints corresponding to $\mathcal{A}$ and $J_{\mathcal{A}}$ is the related Jacobian. An optimal solution of problem (4) satisfies the first order optimality conditions for this problem, i.e. it is solution of the linear system

$$\begin{pmatrix} H(x,y) & -J_{\mathcal{A}}(x)^T \\ -J_{\mathcal{A}}(x) & 0 \end{pmatrix} \begin{pmatrix} \delta x \\ y_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} -g(x) \\ c_{\mathcal{A}}(x) \end{pmatrix}, \tag{5}$$

where $y_{\mathcal{A}}$ is the vector of Lagrangian multipliers for the quadratic problem (4), that provides an approximation of the Lagrangian multipliers corresponding to $c_{\mathcal{A}}$ in the original problem (1). This system has the form (2), with $H = H(x, y)$ and $J = J_{\mathcal{A}}(x)$. Systems of this type are obtained also in the case of Sequential Inequality-constrained Quadratic Programming methods, where no a priori prediction of the active set is made [48]. The step δx resulting from the solution of (5) is used to update the current approximation of the solution; actually, a linesearch or trust-region approach must be applied to obtain useful updates.

A fundamental aspect for the effectiveness of SQP methods is the choice of the Hessian approximation $H(x, y)$ at each iteration. The SQP methods also have several critical shortcomings, such as the possibility that the subproblem is not convex, the linearized constraints are inconsistent and the iterates do not converge. For a discussion on these issues and the strategies to deal with them the reader is referred to [48] and the references therein. We only note that any variant of the SQP method outlined here requires the solution of KKT systems.

The key idea of IP methods is to approach a solution of problem (1) by approximately solving a sequence of barrier problems (BPs), depending on a parameter $\mu > 0$.

Under reasonable assumptions, as μ decreases to zero, local minimizers of the BPs exist and describe trajectories converging to local solutions of (1) [33]. The objective function of a BP is generally obtained by adding a logarithmic term (barrier function) to the objective of the original problem, in order to prevent the iterates from prematurely approaching the boundary of the region defined by the inequality constraints. We consider here the following BP:

$$\begin{array}{ll} \underset{x,s}{\text{minimize}} & \phi_\mu(x,s) \equiv f(x) - \mu \sum_{i=1}^{m_\mathcal{I}} ln(s_i) \\ \text{s. t.} & c_\mathcal{I}(x) - s = 0, \\ & c_\mathcal{E}(x) = 0, \\ & s \geq 0, \end{array} \tag{6}$$

where $s = (s_i) \in \Re^{m_\mathcal{I}}$ is a vector of slack variables. This formulation allows inequality constraints to be violated and is generally preferred to formulations where inequality constraints must be strictly satisfied, since the latter requirement may be very difficult to achieve [48].

An approximate solution of (6) can be obtained by applying a Newton step to the KKT conditions of the BP, starting from a previous approximation. This leads to the following system:

$$\begin{pmatrix} H(x,y) & 0 & -J_\mathcal{I}(x)^T & -J_\mathcal{E}(x)^T \\ 0 & S^{-1}Y_\mathcal{I} & I & 0 \\ -J_\mathcal{I}(x) & I & 0 & 0 \\ -J_\mathcal{E}(x) & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta x \\ \delta s \\ \delta y_\mathcal{I} \\ \delta y_\mathcal{E} \end{pmatrix} = \begin{pmatrix} -g(x) + J_\mathcal{I}(x)^T y_\mathcal{I} + J_\mathcal{E}(x)^T y_\mathcal{E} \\ -y_\mathcal{I} + \mu S^{-1} e \\ c_\mathcal{I}(x) - s \\ c_\mathcal{E}(x) \end{pmatrix}, \tag{7}$$

where $y_\mathcal{I}$ and $y_\mathcal{E}$ denote the current approximations of the Lagrangian multipliers. Note that system (7) can be also obtained by applying a Newton step to perturbed KKT conditions for the original problem (1), where the perturbation concerns only the complementarity conditions:

$$S y_\mathcal{I} = \mu e.$$

Another way to approximately solve the BP is based on the SEQP approach, i.e. on the use of a quadratic model of the Lagrangian of the BP subject to linearizations of the equality constraints:

$$\begin{array}{ll} \underset{\delta x, \delta s}{\text{minimize}} & q(\delta x, \delta s) \equiv \delta x^T g(x) + \frac{1}{2} \delta x^T H(x,y) \delta x - \mu \delta s^T S^{-1} e + \frac{1}{2} \delta s^T S^{-1} Y_\mathcal{I} \delta s \\ \text{s. t.} & c_\mathcal{I}(x) + J_\mathcal{I}(x) \delta x - \delta s - s = 0, \\ & c_\mathcal{E}(x) + J_\mathcal{E}(x) \delta x = 0. \end{array} \tag{8}$$

This leads again to the solution of system (7). We observe that the objective function in (8) is a primal-dual model, since it esplicitly involves the dual variable $y_\mathcal{I}$; primal

only models can be also considered, but the primal-dual ones are generally preferred [20].

System (7) (which is of type (2)) is usually reduced to a smaller system of the same type by eliminating δs:

$$\begin{pmatrix} H(x,y) & -J_{\mathcal{I}}(x)^T & -J_{\mathcal{E}}(x)^T \\ -J_{\mathcal{I}}(x) & -Y_{\mathcal{I}}^{-1}S & 0 \\ -J_{\mathcal{E}}(x) & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y_{\mathcal{I}} \\ \delta y_{\mathcal{E}} \end{pmatrix} = \begin{pmatrix} -g(x) + J_{\mathcal{I}}(x)^T y_{\mathcal{I}} + J_{\mathcal{E}}(x)^T y_{\mathcal{E}} \\ c_{\mathcal{I}}(x) - \mu Y_{\mathcal{I}}^{-1} e \\ c_{\mathcal{E}}(x) \end{pmatrix}; \tag{9}$$

here

$$H = H(x,y), \quad J = \begin{pmatrix} J_{\mathcal{I}}(x) \\ J_{\mathcal{E}}(x) \end{pmatrix}, \quad D = \begin{pmatrix} Y_{\mathcal{I}}^{-1}S & 0 \\ 0 & 0 \end{pmatrix}. \tag{10}$$

As explained in Section 3, system (9) may be very ill-conditioned as the iterates approach the solution of problem (1); this must be strongly taken into account when iterative solvers are used, while it is not a severe problem when direct solvers are applied [80, 81]. Other types of BP lead to linear systems of the same form, with the same ill conditioning issues.

We note that linesearch or trust-region strategies, along with fraction-to-the-boundary rules, must be applied to keep s and $y_{\mathcal{I}}$ "sufficiently positive" and to actually make progress toward the solution of the original problem. A fundamental issue for the effectiveness of the IP methods is the update of the barrier parameter; furthermore, as for the SQP methods, suitable strategies must be used to deal with possible non-convexity, inconsistent linearized constraints and non-convergent iterates. Details on these issues can be found in [33, 48] and in the references therein. Again, any variant of the IP method sketched here requires the solution of a system of type (2) at each iteration.

We conclude by observing that a very large amount of work has been devoted to the development, analysis and implementation of IP methods in the last fifteen years, as testified by the large number of papers on this subject and by the software developed [59, 62, 67]. Indeed, in [33] it is pointed out that the development of IP methods has transformed both the theory and practice of constrained optimization, bringing "together areas of optimization that for many years were treated as firmly disjoint", and hence leading to a "revolution" in the field. For these reasons, in the following the KKT system is considered in the context of IP methods.

3 Basic properties of the KKT matrix

For the sake of completeness, we report some properties of the KKT matrix $\mathcal{K}$. Our presentation is far from being exhaustive; for a detailed discussion on this topic the reader is referred to [7] and to the references therein. Without loss of generality, we

assume that, if the diagonal matrix D in (2) has rank $p > 0$, then its nonzero diagonal entries are located in the first p rows, i.e.

$$D = \begin{pmatrix} D_{\mathcal{I}} & 0 \\ 0 & 0 \end{pmatrix},$$

with $D_{\mathcal{I}} \in \Re^{p \times p}$ nonsingular. Accordingly, the matrix J and the vectors w and d in (2) are partitioned as

$$J = \begin{pmatrix} J_{\mathcal{I}} \\ J_{\mathcal{E}} \end{pmatrix}, \quad w = \begin{pmatrix} w_{\mathcal{I}} \\ w_{\mathcal{E}} \end{pmatrix}, \quad d = \begin{pmatrix} d_{\mathcal{I}} \\ d_{\mathcal{E}} \end{pmatrix}.$$

These notations are reminiscent of the fact that, in the context of IP methods, a part of D and J is associated to the inequality constraints while the other part to the equality constraints. We also note that D can be written as

$$D = I_{\mathcal{I}} D_{\mathcal{I}} I_{\mathcal{I}}^T, \tag{11}$$

where $I_{\mathcal{I}} \in \Re^{m \times p}$ consists of the columns of the identity matrix of dimension m that correspond to the nonzero diagonal entries of D.

The following theorem provides sufficient conditions for which the matrix $\mathcal{K}$, with $D = 0$, is nonsingular and hence the KKT system has a unique solution.

Theorem 1 *Assume $D = 0$, J has full rank and $Z \in \Re^{n \times (n-m)}$ is a basis for* $\ker(J)$. *If $Z^T H Z$ is positive definite, then $\mathcal{K}$ is nonsingular.*

A proof of this theorem can be found, e.g., in [64, Lemma 16.1] . Note that $Z^T H Z$ positive definite means that the quadratic problem related to the KKT matrix is strictly convex and hence the KKT system provides its solution.

The previous theorem can be generalized to the case $\text{rank}(D) = p > 0$ (D positive semidefinite), by transforming the KKT system (2) into an equivalent one. Let $z = -D_{\mathcal{I}} I_{\mathcal{I}}^T w$, system (2) can be rewritten as

$$\begin{pmatrix} H & 0 & -J^T \\ 0 & D_{\mathcal{I}}^{-1} & I_{\mathcal{I}}^T \\ -J & I_{\mathcal{I}} & 0 \end{pmatrix} \begin{pmatrix} v \\ z \\ w \end{pmatrix} = \begin{pmatrix} c \\ 0 \\ d \end{pmatrix}; \tag{12}$$

by Theorem 1 the matrix in (12) is nonsingular provided that $(-J \;\; I_{\mathcal{I}})$ has full rank and, for any basis $Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \in \Re^{(n+p) \times (n+p-m)}$ of $\ker(-J \;\; I_{\mathcal{I}})$, the matrix $Z_1^T H Z_1 + Z_2^T D_{\mathcal{I}}^{-1} Z_2$ is positive definite. Of course, under the same hypoteses, the KKT matrix $\mathcal{K}$ is nonsingular too.

A necessary and sufficient condition for the nonsingularity of $\mathcal{K}$ is provided by the following theorem [7, Theorem 3.4]:

Theorem 2 *Assume H is positive semidefinite, J has full rank and D is positive semidefinite. Then $\mathcal{K}$ is nonsingular if and only if* $\ker(H) \cap \ker(J) = \{0\}$.

It is easy to verify that, under the assumptions of Theorem 2, the condition $\ker(H) \cap \ker(J) = \{0\}$ is equivalent to one of the following conditions:

- $Z^T H Z$ is positive definite, if $D = 0$,
- $Z_1^T H Z_1 + Z_2^T D_{\mathcal{I}}^{-1} Z_2$ is positive definite, if $D \neq 0$,

where Z, Z_1, Z_2 and $D_{\mathcal{I}}$ are the matrices defined above. If H is not positive semidefinite, then the positive definiteness of $Z^T H Z$ or $Z_1^T H Z_1 + Z_2^T D_{\mathcal{I}}^{-1} Z_2$ implies that $\ker(H) \cap \ker(J) = \{0\}$, but the converse is not true.[1] We note that Theorem 2 can be easily extended to the case J is rank deficient, by adding to the condition $\ker(H) \cap \ker(J) = \{0\}$ the following one:

$$\ker(D) \cap \ker(J^T) = \{0\}.$$

We note also that the assumption J full rank is not too restrictive, since it means that the quadratic problem related to the KKT conditions has no redundant constraints. The hypotesis that the reduced Hessian is positive semidefinite is not satisfied in general, but, as discussed in Section 4.3, the optimization methods usually try to deal with locally convex problems and hence explicitly or implicitly apply modifications to obtain a positive semidefinite reduced Hessian.

When H is nonsingular, the KKT matrix $\mathcal{K}$ has the following factorization:

$$\mathcal{K} = \begin{pmatrix} I & 0 \\ -JH^{-1} & I \end{pmatrix} \begin{pmatrix} H & 0 \\ 0 & -S_H \end{pmatrix} \begin{pmatrix} I & -H^{-1}J^T \\ 0 & I \end{pmatrix}, \tag{13}$$

where

$$S_H = (D + JH^{-1}J^T), \tag{14}$$

i.e. $-S_H$ is the Schur complement of H in $\mathcal{K}$; the KKT matrix is therefore nonsingular if and only if S_H is nonsingular. Analogously, when D is nonsingular, $\mathcal{K}$ can be factorized as

$$\mathcal{K} = \begin{pmatrix} I & J^T D^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} S_D & 0 \\ 0 & -D \end{pmatrix} \begin{pmatrix} I & 0 \\ D^{-1}J & I \end{pmatrix}, \tag{15}$$

[1] A counterexample is given by

$$H = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad D = 0.$$

In this case $Z = (0 \;\; 0 \;\; 1)$ is a basis of $\ker(J)$ and $Z^T H Z = (-1)$, while $\ker(H) \cap \ker(J) = \{0\}$.

where

$$S_D = H + J^T D^{-1} J, \tag{16}$$

hence $\mathcal{K}$ is nonsingular if and only if S_D is nonsingular. We observe that explicit espressions for the inverse of $\mathcal{K}$ can be easily obtained from (13) or (15); espressions of $\mathcal{K}^{-1}$ are available also when H and D are both singular, but some other condition holds, e.g. $Z^T H Z$ is positive semidefinite and $D = 0$ [7, Section 3.3]. However, such espressions of $\mathcal{K}^{-1}$ are of limited practical use in the context of optimization, mainly because of their computational cost. For example, the explicit computation of the inverse of $\mathcal{K}$ through the factorization (13) may be convenient if H and S_H are "easy to invert", e.g. H is diagonal and $JH^{-1}J^T$ is not too dense.

The factorizations (13) and (15) provide also information on the spectrum of $\mathcal{K}$. Let us remember that the *inertia* of a symmetric matrix $A \in \Re^{q\times q}$ is defined as the triple $In(A) = (i_+, i_-, i_0)$, where i_+, i_- and i_0 are the number of positive, negative and null eigenvalues of A. The *Sylvester law of inertia* states that if A is symmetric and $U \in \Re^{q\times q}$ is nonsingular, then $In(A) = In(UAU^T)$ (see, e.g., [43, Theorem 8.1.17]). If D is positive definite, by applying the Sylvester law of inertia to (15), we have that

$$In(\mathcal{K}) = In(S_D) + (0, m, 0)$$

(a similar result can be obtained from (13)). Therefore, if H is positive definite,

$$In(\mathcal{K}) = (n, m, 0). \tag{17}$$

The previous result holds under more general hypoteses, as a consequence of the following theorems (see [46, Lemma 3.4] and [30, Proposition 2]):

Theorem 3 *Assume* $D = 0$, $\mathrm{rank}(J) = r$ *and* $Z \in \Re^{n\times(n-r)}$ *is a basis of* $\ker(J)$. *Then*

$$In(\mathcal{K}) = In(Z^T H Z) + (r, r, m - r). \tag{18}$$

Theorem 4 *Assume* D *is positive semidefinite, with* $\mathrm{rank}(D) = p$ *and* $0 < p < m$, $\mathrm{rank}(J\ D) = r$ *and* $Z \in \Re^{n\times(n-r+p)}$ *is a basis of* $\ker(J_{\mathcal{E}})$. *Then*

$$In(\mathcal{K}) = In(Z^T(H + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z) + (r - p, r, m - r). \tag{19}$$

From (18) (respectively (19)) we see that that, if J has full rank, the matrix $Z^T H Z$ (respectively $Z^T(H + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z$) is positive definite if and only if (17) holds. Since $Z^T H Z$ and $Z^T(H + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z$ are the reduced Hessian matrices of suitable local quadratic models of the optimization problem (see, e.g., [64]), the inertia of $\mathcal{K}$ reveals if the problem is locally strictly convex at the current iterate. In this context, the inertia (17) is called *correct inertia*. As discussed in Section 4.3, this relation between

convexity and inertia has motivated the interest for suitable modifications of both direct and iterative solvers, in order to detect, during the solution process, if the KKT matrix does not have the correct inertia.

Eigenvalue bounds are also available for the matrix $\mathcal{K}$, which provide information on its condition number; a general result is reported below [73].

Theorem 5 *Assume H is positive definite, D is positive semidefinite and J has full rank. Let ξ_{min} and ξ_{max} be the minimum and the maximum eigenvalue of H, θ_{max} the maximum eigenvalue of D, and σ_{min} and σ_{max} the minimum and the maximum singular value of J and $\lambda(\mathcal{K})$ the spectrum of $\mathcal{K}$. Then each $\lambda \in \lambda(\mathcal{K})$ satisfies*

$$\lambda \in I^- \cup I^+$$

where

$$I^- = \left[\frac{1}{2}\left(\xi_{min} - \theta_{max} - \sqrt{(\xi_{min} + \theta_{max})^2 + 4\sigma_{max}^2}\right), \frac{1}{2}\left(\xi_{max} - \sqrt{\xi_{max}^2 + 4\sigma_{min}^2}\right)\right]$$

and

$$I^+ = \left[\xi_{min}, \frac{1}{2}\left(\xi_{max} + \sqrt{\xi_{max}^2 + 4\sigma_{max}^2}\right)\right].$$

Tighter bounds can be found in [73], while a generalization to the case H positive semidefinite is presented in [68]. From Theorem 5 we see that the condition number $\kappa(\mathcal{K}) = \max_{\lambda\in\lambda(\mathcal{K})} |\lambda| / \min_{\lambda\in\lambda(\mathcal{K})} |\lambda|$ grows, e.g., as ξ_{min} or σ_{min} tends to zero, or as ξ_{max} or θ_{max} or σ_{max} tends to infinity. We note that, in IP methods, the KKT matrix exhibits an increasing inherent ill conditioning as the iterates approach the solution of the optimization problem. A precise estimate of the condition number depends on the particular IP method, but we can generally say that the ill conditioning depends on the fact that, as the IP method progresses, some entries of the matrix D tend to zero while others tend to infinity. We come back to this issue in Section 4, focusing on preconditioning techniques for the iterative solution of the KKT system.

4 Fundamental issues in solving the KKT system

As pointed out in Section 2, the KKT system arising at each step of an IP method for solving the optimization problem (1) is related to the first-order optimality conditions for a local quadratic model of the problem and provides the direction used to update the current approximation of the optimal solution. Therefore, besides common issues in solving linear systems, fundamental aspects must be addressed that are strictly related to the underlying optimization problem and method. On the other hand, advances

in linear algebra give the opportunity of improving the performance of optimization methods, allowing the solution of larger and more complex problems, which, in turn, require more sophisticated linear algebra algorithms. This mutual impact stimulates a feedback mechanism between the fields of linear algebra and optimization.

In this context, an important issue is the influence that the characteristics of the optimization problem and method have on the form of the KKT system. Different types of constraints and different versions of IP methods lead to KKT systems with the same block structure, but different block properties. A solver for the KKT system, either direct or iterative, must be able to exploit such features in order to be effective in the optimization process.

We observe that the KKT system can be solved by reducing it to a smaller system. Assuming D nonsingular, we can use the second block of equations in (2) to eliminate $w = -D^{-1}(Jv + d)$, obtaining the following $n \times n$ *condensed system*:

$$S_D v = c - J^T D^{-1} d, \tag{20}$$

where S_D is the matrix in (16). This system may be significantly smaller than the KKT one if the optimization problem has many constraints; furthermore, its matrix is positive semidefinite (definite) if H is positive semidefinite (definite), while the KKT matrix $\mathcal{K}$ is indefinite. On the other hand, $\mathcal{K}$ is sparse if H and J are sparse, while S_D may result dense even if H and J are sparse, e.g. when J has a dense column, and it is usually denser anyway. Finally, when IP methods are applied to QP problems, at each iteration the matrix of the condensed system must be entirely recomputed, while the KKT system requires only an update of the diagonal entries. In both cases, the sparsity pattern of the matrix is unchanged during the whole IP algorithm. Similar considerations apply to the system obtained by eliminating $v = H^{-1}(J^T w + c)$, provided that H is nonsingular:

$$S_H w = -d - JH^{-1}c, \tag{21}$$

where S_H is defined in (14). As already observed for the factorization (13), the cost for inverting H must be also taken into account, which is generally greater than the cost for computing the diagonal matrix D^{-1}. A further approach for reducing the KKT system is to eliminate the unknowns corresponding to $D_{\mathcal{I}}$, if any, and then apply a projection onto the space of the linearized constraints $J_{\mathcal{E}} v = d_{\mathcal{E}}$. The matrix of the resulting system is the reduced Hessian $Z^T(H + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z$ or $Z^T H Z$, mentioned in Section 3. This approach shows the same drawbacks as the condensed sytems, plus the cost of computing the basis Z of $\ker(J_{\mathcal{E}})$.

Based on the above observations, in large-scale problems the KKT system is generally preferred to the condensed one. This choice is also due to the availability of effective linear algebra techniques for its solution and, at the same time, stimulates

the development of new algorithms. Nevertheless, there are some optimization problems for which the condensed system is a "natural" choice, e.g. problems with bound constraints only, where J is diagonal.

Direct methods are widely used for solving the KKT systems in well-established optimization codes based on IP methods, such as MOSEK [61], LOQO [77, 78], OOQP [37], KNITRO-Interior/Direct [12], IPOPT [79]. In this case, modified Cholesky factorizations [19, 40, 72], that are capable of handling the case the matrix is not positive definite, are usually applied to the condensed system, while LBL^T factorizations, where L is unit lower triangular and B is symmetric block diagonal with 1×1 or 2×2 blocks, are applied to the KKT system. The pivoting strategies by Bunch and Parlett [11], Bunch and Kaufman [10] and Fletcher [29] are generally used in the latter factorization. When the matrices H and D are both positive definite, the KKT matrix is quasi-definite and a "sufficiently accurate" LBL^T factorization with B diagonal can be computed [76]. Since the matrices to be factorized are often sparse, suitable reordering strategies are exploited to deal with the fill-in problem. As observed at the end of Section 2, when using direct methods the inherent ill-conditioning of the KKT matrix is not a severe problem and, under pretty general assumptions, these methods are able to compute a search direction accurate enough to advance toward the optimal solution.

When the problem is large-scale, the cost of the factorizations may be prohibitive in terms of memory and time, thus limiting the effective use of optimization codes. This has motivated in the last years an increasing research activity devoted to the development of suitable iterative techniques for solving the KKT systems. Examples of optimization codes implementing such techniques are available, such as KNITRO-Interior/CG [12], HOPDM [44] and PRQP (see Section 5). Due to the ill conditioning of the KKT system, effective preconditioners must be used to obtain useful search directions. As explained in Section 4.1, a suitable choice of the preconditioning matrix allows to use the Conjugate Gradient (CG) method, although the matrix $\mathcal{K}$ is indefinite; furthermore, with any symmetric preconditioner, a symmetric variant of QMR (SQMR) [36] can be applied. An interesting issue in the use of iterative methods is the possibility of relating the accuracy of the solution of the KKT system to the quality of the current IP iterate, in order to reduce the computational cost. The basic idea is that during early outer iterations it is useless to require high accuracy in the solution of the KKT system, while it is appropriate to increase the accuracy requirement as the IP iterate approaches the optimal solution. This strategy can be applied through suitable adaptive stopping criteria for the inner iterations. These issues concerning the iterative solution of the KKT system have a significant impact on the effectiveness of IP methods and are discussed in detail in Sections 4.1 and 4.2.

Finally, since the inertia of the KKT matrix $\mathcal{K}$ can reveal if the optimization problem is locally convex at the current iterate, interest has been devoted to the development of solvers that detect if $\mathcal{K}$ has the correct inertia and, if this is not the case, apply

suitable modification to K. This topic, which is important for the success of nonlinear optimization methods, is the subject of Section 4.3.

4.1 Preconditioning the KKT system

As pointed out in [7], the development of high-quality preconditioners for the KKT system needs exploiting the particular block structure of the system, possibly together with information about the meaning and structure of the variuos blocks. The quality of standard algebraic preconditioners, such as incomplete factorizations and sparse approximate inverses, has often turned out to be not statisfactory for KKT systems. On the other hand, preconditioners that are tailored to the special structure of the KKT system and use knowledge of the origin of the system would guarantee better perfomance.

First experiments with the iterative solution of the KKT system have been concerned with the application of the CG method to condensed forms of this system. Most of these experiments have been performed in the context of linear and quadratic programming, and diagonal preconditioners or incomplete Cholesky factorizations (ICF) have been the usual choices [18, 53, 56, 58]. CG with spanning-tree preconditioners [69] and preconditioned LSQR [39] have been also used.

For the reasons explained at the beginning of Section 4, in the last years the KKT system (2) has been preferred to the condensed one. Furthermore, it has been noted in [66] that every preconditioner for the condensed system induces a preconditioner for the KKT system, while the converse is not true. In this context, first esperiments have been performed using symmetric Krylov subspace solvers, such as SYMMLQ coupled with suitable symmetric preconditioners [38] and SQMR, with a modified SSOR preconditioner [34]. A class of preconditioners reducing the KKT system to positive definite systems, and hence allowing the use of CG, has been proposed in [66].

The most popular preconditioner for the KKT system is the so-called *Constraint Preconditioner* (CP), which is a matrix having the same block structure as the KKT matrix, with the (1,2) and (2,1) blocks unchanged. CPs date back to the 70's [1], but in the last decade they have been extensively applied to the KKT system, not only in the optimization context (see e.g. [7, 8, 13, 23, 24, 28, 31, 45, 47, 54, 57, 70]). We consider the following symmetric indefinite CP:

$$\mathcal{P} = \begin{pmatrix} M & -J^T \\ -J & -D \end{pmatrix}, \tag{22}$$

where the (1,1) block M is some symmetric approximation of the matrix H. The name Constraint Preconditioner refers to the fact that $\mathcal{P}$ is the coefficient matrix of a KKT system associated to an optimization problem with the same constraints as the original problem. It is worth noting that M should be chosen so that the resulting

preconditioner is nonsingular (see Theorems 1 and 2 in Section 3) and is significantly "easier to invert" than the original KKT matrix. A common choice is M diagonal, e.g. $M = diag(H)$. A CP variant where D instead of H is approximated has been analysed in [28]; the resulting preconditioner is generally more expensive to apply, unless optimization problems with a particular structure are considered.

The spectral properties of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{K}$ have been investigated in many papers [6, 8, 23, 28, 31, 54, 57]. The following result holds for the case $D = 0$ [54].

Theorem 6 *Assume $D = 0$, J has full rank and $Z \in \Re^{n\times(n-m)}$ is a basis for* $\ker(J)$. *Furthermore, let M be such that $\mathcal{P}$ in (22) is nonsingular. Then the preconditioned matrix $\mathcal{P}^{-1}\mathcal{K}$ has*

- *an eingenvalue at 1 with multiplicity $2m$;*
- *$n - m$ eigenvalues that are defined by the generalized eigenproblem $Z^T H Z u = \lambda Z^T M Z u$.*

As pointed out in [54], if either $Z^T H Z$ or $Z^T M Z$ is positive definite, then all the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{K}$ are real. In view of Theorem 6, we observe that the better M approximates H, the more clustered around 1 are the eigenvalues of $\mathcal{P}^{-1}\mathcal{K}$. The results of the previous theorem can be extended to the case D positive semidefinite, as follows [23].

Theorem 7 *Assume D is positive semidefinite, with $rank(D) = p > 0$, J has full rank, $Z \in \Re^{n\times(n-m+p)}$ is a basis for* $\ker(J_E)$. *Furthermore, let M be such that $\mathcal{P}$ in (22) is nonsingular. Then the preconditioned matrix $\mathcal{P}^{-1}\mathcal{K}$ has*

- *an eingenvalue at 1 with multiplicity $2m - p$;*
- *$n - m + p$ eigenvalues that are defined by the generalized eigenproblem $Z^T(H + J_I^T D_I^{-1} J_{\mathcal{I}})Zu = \lambda Z^T(M + J_I^T D_I^{-1} J_{\mathcal{I}})Z$.*

As for the case $D = 0$, if either $Z^T(H + J_I^T D_I^{-1} J_{\mathcal{I}})Z$ or $Z^T(M + J_I^T D_I^{-1} J_{\mathcal{I}})Z$ is positive definite, then the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{K}$ are real.

Due to the indefiniteness of CP, $\mathcal{P}^{-1}\mathcal{K}$ is generally nonsymmetric. Thus, Krylov methods for nonsymmetric matrices, such as GMRES [71] and QMR [35], should be used. Unlike GMRES, which uses a long-term recurrence for generating an orthogonal basis for the corresponding Krylov subspace, QMR is based on a short-term recurrence, but generates a nonorthogonal basis. Note that, due to the symmetry of $\mathcal{K}$ and $\mathcal{P}$, SQMR can be applied to the preconditioned KKT system, which is transpose-free, and hence computationally more efficient.

It is interesting to note that, although the preconditioned KKT matrix is indefinite, the specific form of CP allows to use the CG algorithm for solving the KKT system, provided that a suitable choice of the starting guess is made. The fundamental conditions for using CP with CG are:

(i)
- if $D = 0$, the matrices $Z^T H Z$ and $Z^T M Z$ are positive definite, where Z is a basis for $\ker(J)$;
- if $rank(D) = p$, $0 < p < m$, the matrices $Z^T(H + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z$ and $Z^T(M + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z$ are positive definite, where Z is a basis for $\ker(J_{\mathcal{E}})$;
- if D is positive definite, the matrices $H + J^T D^{-1} J$ and $M + J^T D^{-1} J$ are positive definite;

and

(ii) a starting guess (v^0, w^0) is chosen such that the second block of equations in (2) is satisfied, i.e.

$$Jv^0 + Dw^0 = -d. \tag{23}$$

It can be shown that, when $D = 0$, CG with CP applied to the KKT system behaves as CG applied to

$$(Z^T H Z)u = Z^T(c - Hv^0), \tag{24}$$

with preconditioner $Z^T M Z$ [47]; when $\text{rank}(D) = p > 0$, $0 < p < m$, CG with CP behaves as CG applied to

$$(Z^T(H + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z)u = Z^T(c - J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} d_{\mathcal{I}}) - Z^T(H + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})v^0, \tag{25}$$

with preconditioner $Z^T(M + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z$ [23]; finally, when D is nonsingular (i.e. positive definite), CG with CP is equivalent to applying CG to the condensed system (20) with preconditioner $M + J^T D^{-1} J$. In the latter case, it can be shown that the $\mathcal{K}$-norm of the error of CG method is strictly positive if the error is different from null vector, i.e. the KKT matrix acts on the error at each CG step as a positive definite matrix, although it is indefinite [13]. Note that this does not hold if the matrix D is not positive definite. For instance, if $D = 0$ then the $\mathcal{K}$-norm of the error of CG method may be zero even if the error is not the null vector; therefore the preconditioned CG method may not be able to decrease the residual and hence to compute an accurate approximation of the solution [70]. Safeguard strategies, such as the diagonal scaling discussed in [70] and the residual update strategy presented in [47], must be applied to overcome this problem.

We observe that systems (24), (25) and (20) are the first order optimality conditions of reduced quadratic optimization problems, obtained by eliminating all the constraints from the original quadratic problems. In other words, if condition (ii) holds, CP acts

on the problem as a projector onto the space defined by the linear constraints. For this reason, CG coupled with CP is called the *Preconditioned Projected CG* method.

We finally note that, if the assumptions of Theorem 6 hold and $Z^T M Z$ is positive definite, then, in exact arithmetic, any optimal Krylov method, i.e. any method that generates an orthogonal basis for the corresponding Krylov subspace, terminates after at most $n - m + 2$ steps [54]. Such result has been extended to the case D positive semidefinite [23]. Indeed, if the assumptions of Theorem 7 hold and $Z^T(M + J_{\mathcal{I}}^T D_{\mathcal{I}}^{-1} J_{\mathcal{I}})Z$ is positive definite, then, in exact arithmetic, any optimal Krylov method terminates after at most $\min\{n - m + p + 2, n + m\}$ steps.

The application of CP within a Krylov method is usually accomplished either through its direct LBL^T factorization or the LBL^T factorization of the matrix $S_M = D + JM^{-1}J^T$, which is the Schur complement of the block $-M$ in $\mathcal{P}$. The second strategy exploits the block decomposition of $\mathcal{P}$ given by (13) with H replaced by M; it can be applied if S_M is nonsingular, which is true if M is positive definite and J has full rank. Methods based on this approach are preferred to methods based on a straight factorization of $\mathcal{P}$ if M is very simple (e.g. diagonal) and the cost of solving linear systems involving the Schur complement is not too high. As for the condensed system, the effectiveness of such strategy can suffer from the loss of sparsity in forming the Schur complement, especially when J has a dense column, and in this case a sparse factorization of the CP is a more viable strategy.

A recently proposed approach for building and applying the CP is based on implicit Schilders factorizations [24, 26]. For example, in the case $D = 0$, CP (22) is partitioned as:

$$P = \begin{pmatrix} M_{11} & M_{12} & J_1^T \\ M_{21} & M_{22} & J_2^T \\ J_1 & J_2 & 0 \end{pmatrix}, \tag{26}$$

where each corner block is of dimension $m \times m$. By assuming that the block J_1 is nonsingular and by choosing $L_1 \in \Re^{m \times m}$ and $L_2 \in \Re^{(n-m)\times(n-m)}$, with L_2 nonsingular, $\mathcal{P}$ can be factorized as

$$P = \begin{pmatrix} J_1^T & 0 & L_1 \\ J_2^T & L_2 & E \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} D_1 & 0 & I \\ 0 & D_2 & 0 \\ I & 0 & 0 \end{pmatrix} \begin{pmatrix} J_1 & J_2 & 0 \\ 0 & L_2^T & 0 \\ L_1^T & E^T & I \end{pmatrix}, \tag{27}$$

where

$$D_1 = J_1^{-T} M_{11} J_1^{-1} - L_1^T J_1^{-1} - J_1^{-T} L_1,\ D_2 = L_2^{-1}(M_{22} - J_2^T D_1 J2 - E J_2 - J_2^T E^T) L_2^{-T},$$

$$E = M_{21} J_1^{-1} - J_2^T D_1 - E J_2 - J_2^T L_1 J_1^{-1}.$$

Factorization (27) allows to define M and to reproduce the block J from suitably chosen matrices D_1, D_2 and E. Furthermore, using this factorization allows to build and apply

the preconditioner either in $O((n-m)^3)$ flops, if $2m \leq n$, or in $O(m^3)$ flops, if $2m > n$. In [24] it is shown how incomplete factorizations of type (27) can be used to reduce the total number of flops required by the PCG algorithm. Numerical experiments on a set of KKT systems associated to QP problems from the CUTEr collection have shown that, in some cases, the use of implicit Schilders factorizations can reduce the solution time up to a factor of 5 compared to the use of a direct factorization of the preconditioner. However, satisfying the necessary assumption that J_1 is nonsingular is a nontrivial practical issue of this approach, and "low-cost" strategies should be used for ensuring that J_1 is nonsingular.

In all the previous cases, the factorization of CP may still account for a large part of the computational cost of a single IP iterations and, as pointed out in [47] "methods which avoid this are urgently needed". This has motivated the interest toward suitable CP approximations, that are cheaper to apply.

Recent attempts in this direction are concerned with the use of approximate factorizations of CP. In [57] an inexact CP for the case $D = 0$ is analysed, which is obtained by replacing the zero (2,2)-block with a diagonal positive definite matrix D_1 and by applying an incomplete Cholesky factorization of the Schur complement $M + J^T D_1^{-1} J$. In [6], an inexact CP is obtained without changing a priori the block $D = 0$ in the preconditioner, but using an incomplete Cholesky factorization directly on the Schur complement $JM^{-1}J^T$. Spectral bounds for the preconditioned matrices are provided in both cases.

An alternative approach is presented in [25], which consists in using CP with a zero (2,2)-block even if the KKT matrix may have $D \neq 0$. This idea is motivated by the observation that, when the IP method progresses toward the solution, the entries in D generally get smaller. It has be shown that if $rank(D) = p > 0$, then the preconditioned matrix has at least $2(m-p)$ unit eigenvalues, and, if the entries of D are very small, one may expect that additional $2p$ eigenvalues get clustered around 1. This may allow a significant reduction of the number of inner steps at the last outer IP iterations. As pointed out in [25], the use of such an approximate CP allows to perform its factorization once as long as J remains constant and M in (22) is chosen to remain constant too.

Another approach is proposed in [9], where an approximate CP is obtained by replacing the block J with a sparse approximation $\bar{J}$. The spectral analysis of the resulting preconditioned matrix suggests that a good approximate CP should keep $rank(J - \bar{J})$ as small as possible in order to limit the number of complex eigenvalues, while keeping the smallest singular value of $\bar{J}$ as large as possible, in order to improve the clustering of the eigenvalues around 1. Furthermore, care must be taken to avoid rank deficient approximations.

Finally, CP approximations are obtained by reusing for multiple IP iterations the CP that has been computed at a certain iteration. In the context of optimization, this

idea has been already applied to incomplete Cholesky factorization preconditioners, coupled with the CG method, for the solution of the condensed system [18, 53]. In [16], the reuse of CP is investigated in the context of an infeasible primal-dual Potential Reduction method for convex quadratic programming. A computational study of different adaptive strategies is presented, aimed at selecting the IP iterations at which the CP has to be recomputed. The basic idea of such strategies is to reuse the CP until its effectiveness deteriorates in terms of inner iterations required to solve the system. Some results obtained with this approach are reported in Section 5.

We observe that the use of an approximate preconditioner is expected in general to increase the number of inner iterations; therefore, to have an effective approximation strategy, the time saved in the application of the preconditioner should pay off the time increase due to the larger iteration count. Furthermore, the approximate preconditioner should allow the computation of solutions accurate enough to guarantee the convergenge of the IP method. Finally, the use of approximate CPs generally does not allow to apply the CG algorithm, thus other Krylov solvers must be coupled with it.

4.2 Termination control

The application of an iterative method to the KKT system (2) produces a solution $u = (v, w)$ which satisfies

$$\mathcal{K}u = h + r, \tag{28}$$

where $h = (c, d)$ is the right-hand side in (2) and $r = (r_c, r_d)$ is the residual vector. To describe an adaptive stopping criterion for the inner iterations, we consider, without loss of generality, the KKT system (9) arising in the IP method described in Section 2. In this case the blocks H, J and D in $\mathcal{K}$ are partitioned as in (10) and

$$v = \delta x, \quad c = -g(x) + J_{\mathcal{I}}(x)^T y_{\mathcal{I}} + J_{\mathcal{E}}(x)^T y_{\mathcal{E}},$$
$$w = \begin{pmatrix} \delta y_{\mathcal{I}} \\ \delta y_{\mathcal{E}} \end{pmatrix}, \quad d = \begin{pmatrix} c_{\mathcal{I}}(x) - \mu Y_{\mathcal{I}}^{-1} e \\ c_{\mathcal{E}}(x) \end{pmatrix}, \quad r_d = \begin{pmatrix} r_{\mathcal{I}} \\ r_{\mathcal{E}} \end{pmatrix}.$$

Once an approximation of $(\delta x, \delta y_{\mathcal{I}}, \delta y_{\mathcal{E}})$ has been computed, the corresponding vector δs is recovered from

$$\delta s = -Y_{\mathcal{I}}^{-1} S \delta y_{\mathcal{I}} - Se - \mu Y_{\mathcal{I}}^{-1} e,$$

obtaining an approximate direction $\bar{u} = (\delta x, \delta s, \delta y_{\mathcal{I}}, \delta y_{\mathcal{E}})$ which satisfies

$$\bar{\mathcal{K}}\bar{u} = \bar{h} + \bar{r}\ , \tag{29}$$

where

$$\bar{\mathcal{K}} = \begin{pmatrix} H(x,y) & 0 & -J_{\mathcal{I}}(x)^T & -J_{\mathcal{E}}(x)^T \\ 0 & Y_{\mathcal{I}} & S & 0 \\ -J_{\mathcal{I}}(x) & I & 0 & 0 \\ -J_{\mathcal{E}}(x) & 0 & 0 & 0 \end{pmatrix}$$

and

$$\bar{h} = \begin{pmatrix} -g(x) + J_{\mathcal{I}}(x)^T y_{\mathcal{I}} + J_{\mathcal{E}}(x)^T y_{\mathcal{E}} \\ -Sy_{\mathcal{I}} + \mu e \\ c_{\mathcal{I}}(x) - s \\ c_{\mathcal{E}}(x) \end{pmatrix}, \quad \bar{r} = \begin{pmatrix} r_c \\ Y_{\mathcal{I}} r_{\mathcal{I}} \\ r_{\mathcal{I}} \\ r_{\mathcal{E}} \end{pmatrix}.$$

We observe that system (29) can be regarded as an inexact Newton step on the unperturbed KKT conditions for the original optimization problem:

$$\bar{\mathcal{K}} u = \tilde{h} + \tilde{r} + \bar{r}\,, \tag{30}$$

where

$$\tilde{h} = \bar{h} - \tilde{r}, \quad \tilde{r} = (0, \mu e, 0, 0) \tag{31}$$

and $\tilde{r} + \bar{r}$ is the residual vector. Under suitable hypotheses, the convergence of this inexact Newton method is ensured if, for each outer iteration, a value ξ exists, $\xi \leq \alpha < 1$, such that [21]

$$\|\tilde{r} + \bar{r}\| \leq \xi \|\tilde{h}\|. \tag{32}$$

A sequence of such values ξ is called forcing sequence. Inequality (32) gives the possibility of relating the accuracy of the approximate solution u of the KKT system to the quality of the current IP iterate, thus allowing to reduce the computational cost of the solution of the system, and hence the overall cost of the IP method [3, 14, 52]. The idea is to use adaptive inner iteration stopping criteria that require low accuracy when the outer IP iterate is far from the optimal solution and to require higher accuracy as soon as the IP iterate approaches the solution. If the inner iterations are stopped when the residual in (28) satisfies

$$\|r\| + \|Y_{\mathcal{I}} r_{\mathcal{I}}\| \leq \eta\mu, \quad 0 < \eta < 1, \tag{33}$$

then we have

$$\|\bar{r}\| \leq \|r\| + \|Y_{\mathcal{I}} r_{\mathcal{I}}\| \leq \eta\mu, \quad 0 < \eta < 1. \tag{34}$$

From (34) we have (see [4])

$$\|\tilde{r} + \bar{r}\| \leq \mu\sqrt{p} + \eta\mu,$$

with $p = rank(D)$. Assuming that $\mu = \sigma \frac{s^T y_{\mathcal{I}}}{p}$, with $0 \leq \sigma < \alpha$, since $\frac{s^T y_{\mathcal{I}}}{\sqrt{p}} \leq \|\tilde{h}\|$, we have

$$\|\tilde{r} + \bar{r}\| \leq (\sigma + \eta)\|\tilde{h}\|,$$

and, by choosing η such that $\xi = (\sigma + \eta) \leq \alpha < 1$, the sequence of values ξ results to be a forcing sequence for the inexact Newton method applied to the unperturbed KKT conditions. Stopping criterion (33) relates the size of the residual to the barrier parameter μ, which represents the perturbation in the KKT conditions of the original

optimization problem. As the IP iterate approaches the optimal solution, μ reduces and hence the accuracy required in the inexact solution increases.

We note that the presence of $Y_{\mathcal{I}}$ in (33) implies that the size of the current approximation of the Lagrangian multipliers associated to the inequality constraints may affect the number of inner iterations. However, numerical experiments carried out in the context of PR methods for convex QP problems have shown that $Y_{\mathcal{I}}$ can be neglected, i.e. the following stopping criterion can be used

$$\|r\| + \|r_{\mathcal{I}}\| \leq \eta\mu, \tag{35}$$

without significantly affect the performance of the PR methods [14]. Suitable relaxations of (35), aimed at reducing the number of inner iterations without deteriorating the convergence of the IP method, have been also studied in [14].

4.3 Inertia control

As observed in Section 3, the inertia of the KKT matrix can reveal if the optimization problem is locally strictly convex at the current iterate. Thus, it is important that the method used to solve the KKT systems is able to detect if the KKT matrix has correct inertia. In the case of wrong inertia, the optimization process can continue by using techniques based on either implicitly or explicitly modifying the KKT matrix $\mathcal{K}$ into a matrix $\bar{\mathcal{K}}$ which has correct inertia, i.e. such that $In(\bar{\mathcal{K}}) = (n, m, 0)$. Of course, $\tilde{\mathcal{K}}$ should be such that solving the linear system involving the matrix $\tilde{\mathcal{K}}$ allows to obtain a descent direction for a suitable merit function related to the original optimization problem. Usually, only the H block of $\mathcal{K}$ is modified into $\tilde{H}$, i.e.

$$\tilde{\mathcal{K}} = \begin{pmatrix} \tilde{H} & -J^T \\ -J & -D \end{pmatrix}, \tag{36}$$

and this modification may be regarded as a local convexification of the optimization problem. Modifications of J or D could be also considered, but they do not have any meaning in terms of the underlaying optimization problem. The ability of a solver to reveal and modify the inertia of $\mathcal{K}$ is referred to as inertia control, and hence a solver that has this capability is referred to as inertia-controlling solver.

The inertia control has been mainly investigated in the context of the factorization of the KKT matrix [30, 32, 42, 51]. To illustrate the basic idea of inertia-controlling factorizations, we consider a LBL^T factorizations of $\mathcal{K}$, where, due to the Sylvester law of inertia, the matrix B has the same inertia as $\mathcal{K}$. The inertia of B is readily available, because of the particular block-diagonal structure of this matrix. This inertia can be found out during the factorization process, by imposing additional restrictions on the pivot selection, which are based on controlling the inertia of the submatrix factorized so

far [30]. When the KKT matrix has wrong inertia, an inertia-controlling factorization should have the ability to modify B into $\tilde{B}$ such that $In(\tilde{B}) = (n, m, 0)$ and the related matrix $\tilde{\mathcal{K}} = \Pi L \tilde{B} L^T \Pi^T$ has the form (36), where Π is a suitable permutation matrix.

As for the direct approach, it is important that an iterative solver is able to detect if the KKT matrix has correct inertia. General Krylov methods, such as GMRES and QMR. are unsuitable for controlling the inertia, because these solvers are not able to estimate the number of positive and negative eigenvalues of an indefinite matrix. Conversely, the CG method is able to detect a wrong inertia, since it is indicated by the occurrence of a direction p of negative curvature, i.e. such that $p^T \mathcal{K} p < 0$. In this case, suitable techniques can be used to overcome the problem, i.e. to compute useful search directions.

Some options have been proposed, based on either implicitly or explicitly modifying the KKT matrix. If a linesearch approach is used, the CG method may be terminated as soon as a negative curvature is discovered, in which case either the steepest descent direction (if the negative curvature occurs at the first CG step) or the current CG approximation of the solution can be used as search direction [21, 22]. In a trust-region approach, the CG algorithm may be continued in the subspace of positive curvature whilst gathering negative curvature information [49]. Among the approaches based on the modification of the coefficient matrix, we mention a strategy for handling negative curvature recently presented in [63] in the context of a trust-region approach for unconstrained optimization. The basic idea of such strategy, named shifted CG, is that, once a negative curvature is encountered, an estimate of a negative eigenvalue of the coefficient matrix is computed; then, CG continues using the "shifted" matrix obtained by adding the opposite of such an estimate to each diagonal entry. If a negative curvature is discovered again at the next CG step, this procedure is repeated. When no negative curvature is found, if the current approximation of the solution is inside the trust-region, then the shift is removed; otherwise, it is kept unchanged. We observe that the shifted CG should allow to overcome a possible drawback of the standard CG when applied in the context of trust region approach, that is once the CG iterates leave the trust region they will not return inside it [75].

5 Some experiences in the context of Quadratic Programming

We report our experiences in the application of iterative linear algebra techniques, discussed in the previous sections, to the KKT systems arising in an IP method for the solution of QP problems. These techniques have been used within the *PRQP* optimization solver [14, 15, 16], implementing a Potential Reduction (PR) method by Mizuno et al. [60] for the solution of the convex QP problem

$$\begin{array}{ll} \underset{x}{\text{minimize}} & q(x) \equiv \dfrac{1}{2} x^T Q x + c^T x \\ \text{s. t.} & Ax - \gamma = b, \\ & Gx = d, \\ & x + \vartheta = u, \\ & (x, \gamma, \vartheta) \geq 0, \end{array} \tag{37}$$

and of its dual problem

$$\begin{array}{ll} \underset{x,y,s,t}{\text{maximize}} & p(x, y, s, t) \equiv -\dfrac{1}{2} x^T Q x + b^T y + d^T s - u^T t \\ \text{s. t.} & Qx - A^T y - G^T s + t - \omega = -c, \\ & (x, y, t, \omega) \geq 0, \end{array} \tag{38}$$

where $Q \in \Re^{n \times n}$ is symmetric positive semidefinite, $A \in \Re^{m_1 \times n}$ and $G \in \Re^{m_2 \times n}$ have full rank, $m = m_1 + m_2 \leq n$, x, y, s, t are the (actual) primal and dual variables, and $\gamma, \vartheta, \omega$ are the slack variables.

At each iteration, the PR method requires the solution of a linear system resulting from the application of a Newton step to suitably perturbed KKT conditions for the pair of problems (37)-(38). By eliminating some variables, this system can be reduced to the form (2), where

$$H = Q + E, \quad E = X^{-1}\Omega + \Theta^{-1}T, \quad J = \begin{pmatrix} A \\ G \end{pmatrix}, \quad D = \begin{pmatrix} Y^{-1}\Gamma & 0 \\ 0 & 0 \end{pmatrix},$$

$$v = \delta x, \quad w = \begin{pmatrix} \delta y \\ \delta s \end{pmatrix},$$

with δx, δy and δs search directions corresponding to x, y and s. Note that, if problem (37) does not have the inequality constraints $Ax - \gamma = b$, then $D = Y^{-1}\Gamma$, while if it does not have the equality constraints $Gx - d = 0$, then $D = 0$; in both cases, if the problem does not have the bound constraints $x + \theta = u$, it is $E = X^{-1}\Omega$. Furthermore, in case of bound constraints only (i.e. $x + \theta = u, (x, \theta) \geq 0$), the KKT naturally reduces to the condensed form (21), where S_D is defined in (16) and $H = Q + E$, $E = X^{-1}\Omega$, $J = I$ and $D = T^{-1}\Theta$.

Different choices for the pair iterative solver / preconditioner are available in PRQP. In case of bound constraints only, the condensed form of the KKT system is solved, using the CG method and the limited-memory incomplete Cholesky factorization preconditioner presented in [55]. This preconditioner allows to specify a priori, through a suitable parameter, the amount of fill-in for the incomplete factorization, without the need of using a drop tolerance; thus it has a predictable memory behaviour, which is a desirable feature in large-scale optimization algorithms. An adaptive choice of the fill-in

parameter is performed at each PR step; a suitable diagonal scaling of the condensed matrix is also applied to improve the quality of this matrix [17].

In the other cases, the KKT system (2) is considered; either the CG or the SQMR (with no look ahead) solvers have been implemented, coupled with the "exact" CP having $M = diag(H)$ as (1,1) block or with the "reused" CP (see Section 4.1). The default choice for the preconditioner is the exact CP; it is applied through the sparse LBL^T factorization provided by the MA27 suite of routines [27] from the Harwell Subroutine Library. In this case, the CG method with the starting guess (23) is used by default; the SQMR method can be also chosen by the user. The reused CP strategy is coupled with SQMR, since CG cannot be generally used with approximated CPs. During the PR iterations, the preconditioner is not updated until one of the following two conditions holds: i) the number of SQMR iterations has exceeded twice the number of SQMR iterations performed when the current preconditioner was used the first time (i.e. as an exact CP), ii) the number of times the current preconditioner has been reused exceeds the number of times the previous preconditioner was reused. Other criteria can be chosen to decide when to update the preconditioner; furthermore, CG and SQMR can be also applied alternately, i.e. CG when the CP is used for the first time and SQMR in all the remaining cases (see [16] for details).

The direct solution of either the condensed or the KKT system has been also implemented in PRQP, using MA27; this generally results more efficient on problems with moderate size or with diagonal Hessian matrix (in the latter case the preconditioner is equal to the KKT matrix). No strategies for inertia control are provided, since convex problems are considered.

The preconditioned CG or SQMR iterations are stopped using the following criterion:

$$||r|| \leq tol, \tag{39}$$

where r is the residual corresponding to the approximate solution of the KKT (or condensed) system and

$$tol = \begin{cases} \min\left\{\eta\frac{\Delta}{\rho}, tol_{\Delta_q}\right\}, & \text{if } \Delta_q \leq \zeta \cdot tol_{\Delta_q} \\ \eta\frac{\Delta}{\rho}, & \text{otherwise} \end{cases}; \tag{40}$$

here $\Delta = x^T\omega + y^T\gamma + t^T\theta$ is the duality gap at the current PR iteration and $\Delta_q = \Delta/(1+q(x))$, tol_{Δ_q} is the accuracy required on Δ_q (i.e. $\Delta_q \leq tol_{\Delta_q}$) in order to stop the PR iterations, $\rho \geq n + m + \sqrt{n+m}$ is a parameter of the logarithmic potential function, and $\zeta, \eta \geq 1$ (see [14] for more details). Criterion (39)-(40) allows to adapt the accuracy requirement in the solution of the KKT system to the quality of the current iterate, since Δ decreases toward 0 as the PR iterates approach the solution of the primal and dual QP problems. This stopping criterion results from an application of

the termination control strategy described in Section 4.2 (here Δ/ρ plays the same role as the barrier parameter μ in (35)); however, criterion (39)-(40) is also naturally suggested by the convergence theory of the inexact PR method [14, 15].

The PRQP solver has been run on several convex test problems, varying the iterative method, the preconditioner and related parameters. For the sake of brevity, we report here a selection of the results, which is representative of the effects of the implemented linear algebra techniques on the behaviour of the optimization solver applied to large-scale problems. These results concern seven test problems from the CUTEr collection and three test problems obtained from CUTEr problems by modifying the constraints $Ax = b$ into $Ax - \gamma = b$, in order to have large problems with inequality constraints and nondiagonal Hessian. Details on the test problems are reported in Table 1. The selected problems have been solved either with the direct or the iterative approach. In the latter case both the exact and the reused CP, coupled with the corresponding default solvers, have been applied. The parameter ρ has been set equal to $\nu(n + m)$, with $\nu = 6$ if $n+m \geq 15000$ and $\nu = 10$ otherwise. The tolerance tol_{Δ_q} has been set to 10^{-7}; a stronger accuracy, i.e. 10^{-8}, has been required on measures of the primal and dual infeasibilities. A maximum number of 80 PR iterations has been also considered. In (40), the values $\eta = 1$ and $\zeta = 10$ have been used. The tests have been carried out on a personal computer with a 2.53 GHz Pentium IV processor, 1.256 GB of memory and 512 MB of L2 cache, running the Linux operating system (Ubuntu 4.1.2 distribution). The PR code, written in Fortran 77 with a C driver that manages dynamic memory allocation, has been compiled using the g77 3.4.6 and gcc 4.1.3 compilers. Further results can be found in [14, 16].

Tables 2-4 show the number of outer and inner iterations and the execution times obtained by solving the KKT systems with the selected solvers. We see that the iterative solver with the exact CP is faster than the direct solver for all the CVXQP problems; in this case the total time obtained with the iterative solver is two to three orders of magnitude smaller than the total time with the direct solver, due to the strong decrease in the time spent in the LBL^T factorization. As expected, the use of the iterative solver is not effective for the problems AUG3DCQP and LISWET, which have diagonal Hessian matrix; furthemore, there is no gain in using the iterative approach with GOULDQP3 and QPBAND, since the Hessian matrix is such that the reduction of time achieved by factorizing the preconditioner instead of the KKT matrix is hidden by the time spent in the Krylov solver iterations. We note also that the iterative solution of the KKT system practically does not affect the number of PR iterations, i.e. it allows to compute good search directions. The reused CP strategy is able to reduce the execution times of CVXQP1, CVXQP2, CVXQP1-M, CVXQP2-M and AUG3DCQP, i.e. of the problems where the time for the factorization still dominates the total execution time. It is interesting to note that the reused CP strategy makes the execution time of AUG3DCQP smaller than in the case of direct solution of the KKT system, despite the

Problem	n	m_1	m_2	$nnz\ Hessian$	$nnz\ Constr.$
AUG3DCQP	27543	0	8000	27543	50286
CVXQP1	10000	0	5000	39984	14998
CVXQP2	10000	0	2500	39984	7499
CVXQP3	10000	0	7500	39984	22497
GOULDQP3	19999	0	9999	39996	29997
LISWET5	10002	1000	0	10002	30000
QPBAND	50000	25000	0	99999	50000
CVXQP1-M	10000	5000	0	39984	14998
CVXQP2-M	10000	2500	0	39984	7499
CVXQP3-M	10000	7500	0	39984	22497

Table 1: Dimensions and number of nonzero entries of the Hessian matrices (upper, or lower, triangle) and the constraint matrices of the test problems. The modified problems are identified by -M in the name.

Hessian of this problem is diagonal. Finally, the number of PR iterations is unaffected by this strategy.

6 Concluding remarks

We discussed of the solution of the KKT system in the context of IP methods for nonlinear optimization, highlighting the relations between numerical linear algebra and optimization. We focused on large-scale problems and on the iterative linear algebra solvers, addressing, in particular, three fundamental issues which are related to specific needs of IP methods and have a significat impact on their effectiveness: preconditioning of the KKT system, with special attention to CPs, adaptive stopping criteria for the inner iterations, and controlling the inertia of the KKT matrix. We also reported some results obtained by applying iterative linear algebra techniques in PR methods for quadratic programming.

We believe that, in spite of the large amount of work devoted to the previous issues in the last fifteen years, there is still room for improvements and novel techniques, especially in the preconditioning and the inertia control of the KKT systems.

Acknowledgments

This work was partially supported by the Italian Ministry of University and Research (MIUR), PRIN Project *Innovative Problems and Methods in Nonlinear Optimization*,

Direct solution				
Problem	*it*	t_{fact}	t_{solve}	t_{tot}
AUG3DCQP	19	3.04e+1	2.72e−1	3.11e+1
CVXQP1	20	1.16e+4	5.27e+0	1.16e+4
CVXQP2	19	3.03e+3	2.15e+0	3.03e+3
CVXQP3	16	2.63e+4	8.01e+0	2.63e+4
GOULDQP3	21	1.32e+0	8.35e−2	1.82e+0
LISWET5	25	6.16e−1	3.82e−2	8.77e−1
QPBAND	14	1.37e+0	1.16e−1	2.32e+0
CVXQP1-M	26	2.79e+4	1.29e+1	2.79e+4
CVXQP2-M	24	4.32e+3	3.19e+0	4.32e+3
CVXQP3-M	30	4.72e+4	1.40e+1	4.72e+4

Table 2: PRQP with LBL^T factorization: number of iterations and execution times (matrix factorization, system solution and total PR times, in seconds).

Exact CP					
Problem	it_{out}	it_{in}	t_{prec}	t_{solve}	t_{tot}
AUG3DCQP	19	19	3.20e+1	7.28e−1	3.35e+1
CVXQP1	20	351	1.21e+1	3.84e+0	1.63e+1
CVXQP2	19	393	1.48e−1	1.10e+0	1.45e+0
CVXQP3	16	228	6.47e+1	5.76e+0	7.10e+1
GOULDQP3	21	90	5.47e−1	6.70e−1	1.79e+0
LISWET5	25	25	6.27e−1	2.49e−1	1.13e+0
QPBAND	14	901	7.56e−1	1.74e+1	1.91e+1
CVXQP1-M	25	423	1.23e+1	4.30e+0	1.69e+1
CVXQP2-M	24	415	1.82e−1	1.19e+0	1.62e+0
CVXQP3-M	30	606	1.06e+2	1.36e+1	1.20e+2

Table 3: PRQP with CG and exact CP: number of outer and inner iterations and execution times (preconditioner factorization, system solution and total PR times, in seconds).

Reused CP					
Problem	it_{out}	it_{in}	t_{prec}	t_{solve}	t_{tot}
AUG3DCQP	19	300	1.57e+1	7.02e+0	2.31e+1
CVXQP1	20	738	3.64e+0	8.72e+0	1.26e+1
CVXQP2	20	614	5.96e−2	2.15e+0	2.40e+0
CVXQP3	16	607	1.88e+1	1.53e+1	3.42e+1
GOULDQP3	21	591	3.05e−1	5.51e+0	6.23e+0
LISWET5	25	200	2.99e−1	1.28e+0	1.81e+0
QPBAND	14	1531	3.73e−1	3.81e+1	3.92e+1
CVXQP1-M	25	650	5.92e+0	6.81e+0	1.30e+1
CVXQP2-M	24	624	9.44e−2	2.20e+0	2.53e+0
CVXQP3-M	30	858	4.50e+1	1.96e+1	6.50e+1

Table 4: PRQP with SQMR and reused CP: number of outer and inner iterations and execution times (preconditioner factorization, system solution and total PR times, in seconds).

grant no. 2005017083.

References

[1] O. Axelsson, "Preconditioning of indefinite problems by regularization", SIAM J. Numerical Analysis, 16, pp. 58–69, 1979.

[2] O. Axelsson and M. Neytcheva, "Preconditioning methods for linear systems arising in constrained optimization problems", Numerical Linear Algebra with Applications, 10, pp. 3–31, 2003.

[3] V. Baryamureeba and T. Steihaug, "On the Convergence of an Inexact Primal-Dual Interior Point Method for Linear Programming", Reports in Informatics No. 188, Department of Informatics, University of Bergen, Norway, 2000.

[4] S. Bellavia, "Inexact Interior Point method", Journal of Optimization Theory and Applications, 96, pp. 109–121, 1998.

[5] M. Benzi, "Preconditioning Techniques for Large Linear Systems: A Survey", J. Computational Physics, 182, pp. 418–477, 2002.

[6] M. Benzi and V. Simoncini, "On the Eigenvalues of a Class of Saddle Point Matrices", Numerische Mathematik, 103(2), pp. 173-196, 2006.

[7] M. Benzi, G.H. Golub and J. Liesen, "Numerical Solution of Saddle Point Problems", Acta Numerica, 14, pp. 1–137, 2005.

[8] L. Bergamaschi, J. Gondzio and G. Zilli, "Preconditioning Indefinite Systems in Interior Point Methods for Optimization", Computational Optimization and Applications, 28(2), pp. 149–171, 2004.

[9] L. Bergamaschi, J. Gondzio, M. Venturin and G. Zilli, "Inexact constraint preconditioners for linear systems arising in interior point methods", Computational Optimization and Applications, 36 (2-3), pp. 137–147, 2007.

[10] J.R. Bunch and L. Kaufman "Some stable methods for calculating inertia and solving symmetric linear systems", Mathematics of Computation, 31, pp. 163–179, 1977.

[11] J.R. Bunch and B.N. Parlett "Direct methods for solving symmetric indefinite systems of linear equations", SIAM J. Numerical Analysis, 8, pp. 639–655, 1971.

[12] R. Byrd, J. Nocedal, R. Waltz, "KNITRO: An Integrated Package for Nonlinear Optimization", Large-Scale Nonlinear Optimization (G. Di Pillo and M. Roma eds.), pp. 35–59, 2006.

[13] S. Cafieri , M. D'Apuzzo, V. De Simone , D. di Serafino, "On the iterative solution of KKT systems in Potential Reduction software for Large-Scale Quadratic problems", Computational Optimization and Applications, special issue on High Performance Algorithms and Software for Nonlinear Optimization (A. Murli and G. Toraldo eds.), 38, pp. 27–45, 2007.

[14] S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, "Stopping criteria for inner iterations in inexact Potential Reduction methods: a computational study", Computational Optimization and Applications, special issue on Linear Algebra Issues Arising in Interior Point Methods (J. Gondzio, W.W. Hager and G. Toraldo eds.), 36, pp. 165-193, 2007.

[15] S. Cafieri, M. D'Apuzzo, V.De Simone, D. di Serafino and G. Toraldo, "Convergence Analysis of an Inexact Potential Reduction Method for Convex Quadratic Programming", J. Optimization Theory and Applications, 135 (3), pp.355–366, 2007.

[16] S. Cafieri, M. D'Apuzzo, V. De Simone and D. di Serafino, "On the use of an Approximate Constraint Preconditioner in a Potential Reduction Algorithm for Quadratic Programming", Applied and Industrial Mathematics in Italy II, Series on Advances in Mathematics for Applied Sciences, (V. Cutello, G. Fotia and L. Puccio eds.), 75, pp. 220–230, 2007.

[17] S. Cafieri , M. D'Apuzzo, M. Marino, A. Mucherino, G. Toraldo, "Interior point solver for large-scale quadratic programming problems with bound constraints", J. Optimization Theory and Applications, 129, pp. 55–75, 2006.

[18] T. Carpenter and D.N. Shanno, "An interior point method for quadratic programs based on conjugate projected gradients", Computational Optimization and Applications, 2, pp. 5–28, 1993.

[19] S. Cheng and N. Higham, "A modified Cholesky algorithm based on a symmetric indefinite factorization", SIAM J. Matrix Analysis and Applications, 20, pp. 513-561, 1998.

[20] A. Conn, N.I.M. Gould, D. Orban and P.L. Toint, "A primal-dual trust-region algorithm for non-convex nonlinear programming", Mathematical Programming, Ser. B, 87, pp. 215–249, 2000.

[21] R. Dembo, S. Eisentat and T. Steihaug, "Inexact Newton methods", SIAM J. Numerical Analysis, 19, pp. 400–408, 1982.

[22] R. Dembo and T. Steihaug, "Truncated-Newton algorithms for large-scale unconstrained optimization", Mathematical Programming, 26(2), pp. 190–212, 1983.

[23] H.S. Dollar, "Constraint-style preconditioners for regularized saddle-point problems", SIAM J. Matrix Analysis and Applications, 29, pp. 672–684, 2007.

[24] H. S. Dollar, N.I. M. Gould, W.H.A. Schilders and A.J. Wathen, "Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems", SIAM J. Matrix Analysis and Applications, 28, pp. 170–189, 2006.

[25] H.S. Dollar, N.I.M. Gould, W. Schilders and A.J. Wathen, "Using constraint preconditioners with regularized saddle-point systems", Computational Optimization and Applications, 36, pp. 249–270, 2007.

[26] H.S. Dollar and A.J. Wathen, "Approximate factorization constraint preconditioners for saddle-point matrices", SIAM J. Scientific Computing, 27, 2006.

[27] I.S. Duff and J.K. Reid, "The Multifrontal Solution of Indefinite Sparse Symmetric Linear Equations", ACM Transactions on Mathematical Software, 9 (3), pp. 302–325, 1983.

[28] C. Durazzi and V. Ruggiero, "Indefinitely Preconditioned Conjugate Gradient Method for Large Sparse Equality and Inequality Constrained Quadratic Problems", Numerical Linear Algebra with Applications, 10(8), pp. 673–688, 2003.

[29] R. Fletcher, "Factorizing symmetric indefinite matrices", Linear Algebra and its Applications, 14, pp. 257–272, 1976.

[30] A. Forsgren, "Inertia-controlling factorizations for optimization algorithms", Applied Numerical Mathematics, 43, pp. 91–107, 2002.

[31] A. Forsgren P.E. Gill and J. Griffin, "Iterative solution of augmented systems arising in interior methods", SIAM J. Optimization 18, pp. 666–690, 2007.

[32] A. Forsgren, P.E. Gill and W. Murray, "On the identification of local minimizers in inertia-controlling methods for quadratic programming", SIAM J. Matrix Analysis and Applications, 12, pp. 730–746, 1991.

[33] A. Forsgren, P.E. Gill and M.H. Wright, "Interior Methods for Nonlinear Optimization", SIAM Rewiew, 44(4), pp. 525–597, 2002.

[34] R.W. Freund and F. Jarre, "A QMR-based interior-point algorithm for solving linear programs", AT&T Bell Laboratories and Institute für Angewandte Mathematik und Statistik, Tech. Rep., 1995.

[35] R. Freund and N. Nachtigal, 'QMR: a quasi-minimal residual method for non Hermitian linear systems", Numerische Mathematik, 60, pp. 315–339, 1991.

[36] R. Freund and N. Nachtigal, "Software for simplified Lanczos and QMR algorithms", Applied Numerical Mathematics, 19(3), pp. 319–341, 1995.

[37] E.M. Gertz and S.J. Wright, "Object-Oriented Software for Quadratic Programming", ACM Transactions on Mathematical Software, 29, pp. 58–81, 2003. See also `http://www.cs.wisc.edu/~swright/ooqp/`.

[38] P.E. Gill, W. Murray, D.B Ponceleon and M.A. Saunders, "Preconditioners for Indefinite Systems Arising in Optimization", SIAM J. Matrix Analysis and Applications, 13, pp. 292–311, 1992.

[39] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin and M.H. Wright, "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method", Mathematical Programming, 36, pp. 183–209, 1986.

[40] P.E. Gill, W. Murray and M.H. Wright, "Practical Optimization", Academic Press, 1981.

[41] P.E. Gill, W. Murray and M.H. Wright, "Numerical Linear Algebra and Optimization", Volume I, Addison-Wesley, 1990.

[42] G.H. Golub and C. Greif, "Techniques For Solving General KKT Systems", SCCM Program, Computer Science Dept., Stanford University, SCCM-00-11, 2000.

[43] G.H. Golub and C.F. Van Loan, "Matrix Computations", 3rd edition, The Johns Hopkins University Press, 1996.

[44] J. Gondzio, "HOPDM. A Fast LP Solver Based on a Primal-Dual Interior Point Method", European J. Operation Research, 85, pp. 221–225, 1995.

[45] J. Gondzio, W.W. Hager and G. Toraldo, eds., special issue on "Linear Algebra Issues Arising in Interior Point Methods", Computational Optimization and Applications, 36, 2007.

[46] N.I.M. Gould, "On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem", Mathematical Programming, 32, pp. 90–99, 1985.

[47] N.I.M. Gould, M.E. Hribar and J. Nocedal, "On the solution of equality constrained quadratic programming problems arising in optimization", SIAM J. Scientific Computing, 23, pp. 1376–1395, 2001.

[48] N.I.M. Gould, D. Orban and P.L. Toint, "Numerical methods for large-scale nonlinear optimization", Acta Numerica, 14, pp. 299–361, 2005.

[49] N.I.M. Gould, S. Lucidi, M. Roma and P.L. Toint, "Solving the trust-region subproblem using the Lanczos methods", SIAM J. Optimization, 9, pp. 504–525, 1999.

[50] N.I.M. Gould and P.L. Toint, "SQP methods for large-scale nonlinear programming", in System Modelling and Optimization: Methods, Theory and Applications (M. Powell and S. Scholtes, eds.), Kluwer, Dordrecht, The Netherlands, pp. 149-178, 2000.

[51] N.J. Higham and S. Cheng, "Modifying the inertia of matrices arising in optimization", Linear Algebra and its Applications, 275-276, pp. 261–279, 1998.

[52] S. Ito, "Inexact implementation of interior point algorithms for optimal control problems", Lecture Notes in Num. Appl. Anal., 14, pp. 245-248, 1995.

[53] N. Karmakar and K. Ramakrishnan, "Computational results of an interior point algorithm for large scale linear programming", Mathematical Programming, 52, pp. 555–586, 1991.

[54] C. Keller, N.I.M. Gould, and A.J. Wathen, "Constraint preconditioning for indefinite linear systems", SIAM J. Matrix Analysis and Applications, 21(4), pp. 1300–1317, 2000.

[55] C.-J. Lin and J.J. Moré, "Incomplete Cholesky factorizations with limited memory" SIAM J. Scientific Computing, 21, pp. 24–45, 1999.

[56] L. Luksan, C. Matonoha and J. Vlcek, "Interior-point method for non-linear nonconvex optimization", Numerical Linear Algebra with Applications, 11, pp. 431-453, 2004.

[57] L. Luksan and J. Vlcek, "Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems", Numerical Linear Algebra with Applications, 5, pp. 219–247, 1998.

[58] S. Mehrotra, "Implementations of affine scaling methods: Approximate solutions of systems of linear equations using preconditioned conjugate gradient method", ORSA J. Computing, 4(2), pp. 102–118, 1992.

[59] H.D. Mittelmann, "Decision Tree for Optimization Software", `http://plato.asu.edu/guide.html`, 2007.

[60] S. Mizuno, M. Kojima and M.J. Todd, "Infeasible-interior-point primal-dual potential-reduction algorithms for linear programming", SIAM J. Optimization, 5, pp. 52–67, 1995.

[61] MOSEK ApS, "The MOSEK optimization tools version 3.1 (Revision 28). Users manual and reference", 2002. See also `http://www.mosek.com/`.

[62] NEOS Guide: Optimization Software , `http://www-fp.mcs.anl.gov/OTC/Guide/SoftwareGuide/`.

[63] J. Nocedal, "On the solution of very large nonlinear optimization problems", talk at the Eigth SIAM Conference on Optimization, Stockholm, 2005.

[64] J. Nocedal and S.J. Wright, "Numerical Optimization", Springer Series in Operation Research, Springer, 1999.

[65] D. O'Leary, "Symbiosis between Linear Algebra and Optimization", J. Computational and Applied Mathematics, 123, pp. 447–465, 2000.

[66] A. Oliveira and D. Sorensen, "A new class of preconditioners for large-scale linear systems from interior point methods for linear programming", Linear Algebra and its Applications, 394, pp. 1–24, 2005.

[67] Optimization Online home page, `http://www.optimization-online.org/`.

[68] I. Perugia and V. Simoncini, "Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations", Numerical Linear Algebra with Applications, 7, pp. 585–616, 2000.

[69] L. Portugal, M. Resende, G. Veiga and J. Judice, "An efficient implementation of the infeasible primal-dual network flow method", AT & T Bell Laboratories, New Jersey, Tech. Rep., 1994.

[70] M. Rozloznik and V. Simoncini, "Krylov Subspace Methods for Saddle Point Problems with Indefinite Preconditioning", SIAM J. Matrix Analysis and Applications, 24(2), pp. 368–391, 2002.

[71] Y. Saad and M.H. Schultz, "GMRES: a generalized minimal residual algorithm fo solving nonsymmetric linear systems", SIAM J. Scientific and Statistical Computing, 7, pp. 856-869, 1986.

[72] R. B. Schnabel and E. Eskow, "A Revised Modified Cholesky Factorization Algorithm", SIAM J. Optimization, 9(4), pp. 1135–1148, 1999.

[73] D. Silvester and A.J. Wathen, "Fast Iterative Solution of Stabilized Stokes Systems. Part II: Using General Block Preconditioners", SIAM J. Numerical Analysis, 31, 1994, pp. 1352–1367.

[74] V. Simoncini and D. Szyld, "Recent computational developments in Krylov Subspace Methods for linear systems", Numerical Linear Algebra with Applications, 14(1), pp. 1–59, 2007.

[75] T. Steihaug,"The conjugate gradient method and trust regions in large scale optimization", SIAM J. Numerical Analysis, 20(3), pp. 626–637, 1983.

[76] R.J. Vanderbei, "Symmetric quasi-definite matrices", SIAM J. Optimization, 5, pp. 100-113, 1995.

[77] R.J. Vanderbei, "An Interior Point Code for Quadratic Programming", Optimization Methods and Software, 1, pp. 451–484 , 1999.

[78] R.J. Vanderbei and D.N. Shanno, "An Interior Point algorithm for nonconvex nonlinear programming", Computational Optimization and Applications, 13, pp. 231–252, 1999.

[79] A. Wachter and L. Biegler, "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming", Mathematical Programming, 106(1), pp. 25–57, 2006.

[80] M.H. Wright, "Ill-conditioning and computational error in primal-dual interior methods for nonlinear programming", SIAM J. Optimization, 9, pp. 84–111, 1998.

[81] S.J. Wright, "Stability of augmented system factorizations in interior point methods", SIAM J. Matrix Analysis and Applications, 18, pp. 191–222, 1997.

[82] SIAM Conference on Optimization, Minisymposium MS10: Numerical Linear Algebra Issues in Interior Point Methods, Stockholm, May 15-19, 2005, http://www.siam.org/meetings/op05/.

[83] International Conference on Numerical Optimization and Numerical Linear Algebra, Lhasa, Tibet, China, August 8-12, 2005, http://lsec.cc.ac.cn/~yyx/nlao2005.html.

[84] IMA Conference on Numerical Linear Algebra and Optimisation, University of Birmingham, UK September 13-15, 2007, http://www.ima.org.uk/Conferences/numlinalg/numlinalghome.htm.