Computational Optimization and Applications, 28, 149–171, 2004

Preconditioning Indefinite Systems in Interior Point Methods for Optimization

LUCA BERGAMASCHI berga@dmsa.unipd.it
Department of Mathematical Methods for Applied Sciences, University of Padua, Italy

JACEK GONDZIO J.Gondzio@ed.ac.uk
School of Mathematics, University of Edinburgh, Scotland

GIOVANNI ZILLI zilli@dmsa.unipd.it
Department of Mathematical Methods for Applied Sciences, University of Padua, Italy

Received August 8, 2002; Revised September 8, 2003

Abstract. Every Newton step in an interior-point method for optimization requires a solution of a symmetric indefinite system of linear equations. Most of today's codes apply direct solution methods to perform this task. The use of logarithmic barriers in interior point methods causes unavoidable ill-conditioning of linear systems and, hence, iterative methods fail to provide sufficient accuracy unless appropriately preconditioned. Two types of preconditioners which use some form of incomplete Cholesky factorization for indefinite systems are proposed in this paper. Although they involve significantly sparser factorizations than those used in direct approaches they still capture most of the numerical properties of the preconditioned system. The spectral analysis of the preconditioned matrix is performed: for convex optimization problems all the eigenvalues of this matrix are strictly positive. Numerical results are given for a set of public domain large linearly constrained convex quadratic programming problems with sizes reaching tens of thousands of variables. The analysis of these results reveals that the solution times for such problems on a modern PC are measured in *minutes* when direct methods are used and drop to *seconds* when iterative methods with appropriate preconditioners are used.

Keywords: interior-point methods, iterative solvers, preconditioners

1. Introduction

Every iteration of the interior point method for linear, quadratic or nonlinear programming requires the solution of a possibly *large* and almost always *sparse* linear system

$$\begin{bmatrix} -Q - \Theta_1^{-1} & A^T \\ A & \Theta_2^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{1}$$

In this system, $\Theta_1 \in \mathcal{R}^{n\times n}$ and $\Theta_2 \in \mathcal{R}^{m\times m}$ are diagonal scaling matrices with strictly positive elements. Depending on the problem type one or both matrices Θ_1 and Θ_2 may be present in this system. For linear and quadratic programs with equality constraints $\Theta_2^{-1} = 0$. For nonlinear programs with inequality constraints (and variables without sign restriction) $\Theta_1^{-1} = 0$. The matrices Θ_1 and Θ_2 are well-known to display undesirable properties: as the optimal solution of the problem is approached some elements of Θ tend to zero while others tend to infinity. For ease of the presentation we assume that we deal with convex programs hence the Hessian $Q \in \mathcal{R}^{n\times n}$ is a symmetric positive definite matrix. $A \in \mathcal{R}^{m\times n}$ is the matrix of linear constraints (or the linearization of nonlinear constraints); we assume it has a full rank. The reader interested in interior point methods may consult [30] for an excellent explanation of their theoretical background and [2] for a discussion of implementation issues.

Most of interior point solvers use direct methods and factorize the matrix of (1) into LDL^T form with a unit lower triangular L and an easily invertible D. Since the matrix of (1) is indefinite, D should allow 2 × 2 pivots [3, 10]. An alternative is to transform the system to a quasidefinite one [28]. Quasidefinite matrix has the form $[\begin{smallmatrix} -G & A^T \\ A & F \end{smallmatrix}]$, where G and F are symmetric positive definite matrices and A has full rank. As shown in [28], quasidefinite matrices are strongly factorizable, i.e., a Cholesky-like factorization LDL^T with a diagonal D exists for any symmetric row and column permutation of the quasidefinite matrix. The diagonal matrix D has n negative and m positive pivots.

Following the method of [1] implemented in the HOPDM interior point code we transform the indefinite matrix involved in (1) to a quasidefinite matrix by the use of primal and dual regularization method. Consequently, we deal with the matrix

$$H_R = \begin{bmatrix} -Q - \Theta_1^{-1} & A^T \\ A & \Theta_2^{-1} \end{bmatrix} + \begin{bmatrix} -R_p & 0 \\ 0 & R_d \end{bmatrix},$$

where diagonal positive definite matrices $R_p \in \mathcal{R}^{n\times n}$ and $R_d \in \mathcal{R}^{m\times m}$ can be interpreted as adding proximal terms to the primal and dual objective functions, respectively. In the method of [1] the entries of the regularizing matrices are chosen dynamically: the negligibly small terms are used for all acceptable pivots and the stronger regularization terms are used whenever a dangerously small pivot candidate appears. The use of dynamic regularization introduces little perturbation to the original system because the regularization concentrates uniquely on potentially unstable pivots.

Although direct approach offers a number of advantages such as an easy control of accuracy, it is in some cases prohibitively expensive. This happens, for example, whenever Cholesky-like factor L gets significantly denser than the matrix in (1). In this paper we concentrate on these cases and propose an alternative approach that consists in the use of preconditioned iterative method. For the linear programming case with equality constraints the system (1) is often reduced to normal equations. This produces the symmetric positive definite matrix for which a number of preconditioners have been developed, see for example [4, 29] and the references therein. Following Fletcher [12] we apply an iterative method to the indefinite augmented system. We are aware of the associated breakdown risk at least for some of the iterative methods. However, by dealing with the augmented system we expect to gain more freedom in the design of the sparsity exploiting preconditioner.

The presence of diagonal matrices Θ_1^{-1} and Θ_2^{-1} in the system (1) makes it very ill-conditioned and hence difficult for an iterative approach. The elements of matrices Θ_1 and Θ_2 display a drastic difference of magnitude: some of them tend to zero while others go to infinity. The iterative approach has no chance to converge unless carefully preconditioned.

To be successful the preconditioner should:

- capture most of the numerical properties of (1); and
- be computable at a far lower cost than that of the factorization of (1).

To meet these requirements we propose to keep the matrices Θ_1^{-1} and Θ_2^{-1} in the preconditioner but we simplify the system (1) by dropping out all off-diagonal elements in the matrix Q. Namely, instead of the augmented system

$$H = \begin{bmatrix} -Q - \Theta_1^{-1} & A^T \\ A & \Theta_2^{-1} \end{bmatrix} \tag{2}$$

we will use

$$P = \begin{bmatrix} E & A^T \\ A & \Theta_2^{-1} \end{bmatrix}, \tag{3}$$

where $E = -\mathrm{diag}(Q) - \Theta_1^{-1}$. By dropping all off-diagonal elements from Q we open the possibility of reducing the system with the matrix (3) further to the form of normal equations $-AE^{-1}A^T + \Theta_2^{-1}$. In some cases, such as for example when $m \ll n$, this is a viable alternative.

Matrix P of (3) belongs to a wide class of block-preconditioners that have been studied in numerous applications in the context of partial differential equations, see for example [15, 24] and the references therein, and in the context of optimization [11, 14, 19, 21, 22]. We perform the spectral analysis of $P^{-1}H$ and conclude, as in [19], that for convex optimization problems all the eigenvalues of this matrix are strictly positive. This feature offers much promise for computations. Then we show that the presence of primal and dual regularizations in the matrix H_R and in its preconditioner P_R improves the spectral properties of the preconditioned matrix $P_R^{-1}H_R$.

We have implemented a number of most widely used iterative approaches for indefinite system including BiCGSTAB [27], GMRES [26] and a simplified variant of QMR [13], together with the classical Conjugate Gradient. We have tested the performance of two variations of the preconditioner (3) one using its indefinite Cholesky-like factorization and another reducing it to the normal equations and using the Cholesky factorization of $-AE^{-1}A^T + \Theta_2^{-1}$. The computational experience has confirmed that the iterative methods preconditioned with (3) behave well for a class of (convex) quadratic programming problems. In a moderate number of iterations that usually varies between 10 and 20, the iterative methods provide sufficient accuracy in the solution of linear systems to guarantee the practical convergence to the optimal solution of the problem. In some cases when matrix Q has a considerable number of off-diagonal entries our preconditioned iterative solver runs two orders of magnitude faster than the direct approach.

The paper is organized as follows. In Section 2 we briefly recall the linear algebra issues in the implementation of interior point methods for convex optimization. In Section 3 we recall some basic properties of iterative methods used to solve systems of linear equations. In

Section 4 we introduce the indefinite preconditioner and perform the spectral analysis of the preconditioned matrix. We also discuss the influence of the use of primal-dual regularization method on the properties of the preconditioner. Further we consider two variants of the preconditioner and give arguments that they are both useful in some situations. In Section 5 we discuss the computational experience of the use of preconditioned iterative solvers to solve a number of large convex quadratic programming problems. In Section 6 we give the conclusions and discuss possible further developments of our approach.

2. Linear algebra in interior point methods

Interior point methods for linear, quadratic and nonlinear optimization differ obviously in many details but they rely on the same linear algebra kernel. We discuss briefly two cases of quadratic and nonlinear programming.

2.1. Quadratic programming

Consider the convex quadratic programming problem

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} & Ax = b, \\ & \;\; x \geq 0, \end{array}$$

where $Q \in \mathcal{R}^{n\times n}$ is positive semidefinite matrix, $A \in \mathcal{R}^{m\times n}$ is the full rank matrix of linear constraints and vectors x, c and b have appropriate dimensions. The usual transformation in interior point methods consists in replacing inequality constraints with the logarithmic barriers to get

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^{n} \ln x_j \\ \text{s.t.} & Ax \;= b, \end{array}$$

where $\mu \geq 0$ is a barrier parameter. The Lagrangian associated with this problem has the form:

$$L(x, y, \mu) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - \mu \sum_{j=1}^{n} \ln x_j$$

and the conditions for a stationary point write

$$\begin{array}{l} \nabla_x L(x, y, \mu) = c - A^T y - \mu X^{-1} e + Qx = 0 \\ \nabla_y L(x, y, \mu) = Ax - b = 0, \end{array}$$

where $X^{-1} = \text{diag}\{x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\}$. Having denoted

$$s = \mu X^{-1} e, \qquad \text{i.e.} \quad XSe = \mu e,$$

where $S = \text{diag}\{s_1, s_2, \ldots, s_n\}$ and $e = (1, 1, \ldots, 1)^T$, the first order optimality conditions (for the barrier problem) are:

$$\begin{aligned} Ax &= b, \\ A^T y + s - Qx &= c, \\ XSe &= \mu e \\ (x, s) &\geq 0. \end{aligned} \tag{4}$$

Interior point algorithm for quadratic programming [30] applies Newton method to solve this system of nonlinear equations and gradually reduces the barrier parameter μ to guarantee the convergence to the optimal solution of the original problem. The Newton direction is obtained by solving the system of linear equations:

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix}, \tag{5}$$

where

$$\begin{aligned} \xi_p &= b - Ax, \\ \xi_d &= c - A^T y - s + Qx, \\ \xi_\mu &= \mu e - XSe. \end{aligned}$$

By elimination of

$$\Delta s = X^{-1}(\xi_\mu - S\Delta x) = -X^{-1} S \Delta x + X^{-1} \xi_\mu,$$

from the second equation we get the symmetric indefinite augmented system of linear equations

$$\begin{bmatrix} -Q - \Theta_1^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1} \xi_\mu \\ \xi_p \end{bmatrix}. \tag{6}$$

where $\Theta_1 = XS^{-1}$ is a diagonal scaling matrix. By eliminating Δx from the first equation we can reduce (6) further to the form of normal equations

$$\left(A\left(Q + \Theta_1^{-1}\right)^{-1} A^T\right) \Delta y = b_{QP}.$$

2.2. *Nonlinear programming*

Consider the convex nonlinear optimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) \leq 0, \end{array}$$

where $x \in \mathcal{R}^n$, and $f : \mathcal{R}^n \mapsto \mathcal{R}$ and $g : \mathcal{R}^n \mapsto \mathcal{R}^m$ are convex, twice differentiable. Having replaced inequality constraints with an equality $g(x) + z = 0$, where $z \in \mathcal{R}^m$ is a nonnegative slack variable, we can formulate the associated barrier problem

$$\begin{array}{ll} \min & f(x) - \mu \sum_{i=1}^{m} \ln z_i \\ \text{s.t.} & g(x) + z = 0. \end{array}$$

and write the Lagrangian for it

$$L(x, y, z, \mu) = f(x) + y^T (g(x) + z) - \mu \sum_{i=1}^{m} \ln z_i.$$

The conditions for a stationary point write

$$\begin{aligned} \nabla_x L(x, y, z, \mu) &= \nabla f(x) + \nabla g(x)^T y = 0 \\ \nabla_y L(x, y, z, \mu) &= g(x) + z = 0 \\ \nabla_z L(x, y, z, \mu) &= y - \mu Z^{-1} e = 0, \end{aligned}$$

where $Z^{-1} = \operatorname{diag}\{z_1^{-1}, z_2^{-1}, \ldots, z_m^{-1}\}$. The first order optimality conditions (for the barrier problem) have thus the following form

$$\begin{aligned} \nabla f(x) + \nabla g(x)^T y &= 0, \\ g(x) + z &= 0, \\ YZe &= \mu e \\ (y, z) &\geq 0, \end{aligned} \tag{7}$$

where $Y = \operatorname{diag}\{y_1, y_2, \ldots, y_m\}$. Interior point algorithm for nonlinear programming [30] applies Newton method to solve this system of equations and gradually reduces the barrier parameter μ to guarantee the convergence to the optimal solution of the original problem. The Newton direction is obtained by solving the system of linear equations:

$$\begin{bmatrix} Q(x, y) & A(x)^T & 0 \\ A(x) & 0 & I \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - A(x)^T y \\ -g(x) - z \\ \mu e - YZe, \end{bmatrix}, \tag{8}$$

where

$$\begin{aligned} A(x) &= \nabla g(x) \in \mathcal{R}^{m\times n} \\ Q(x,y) &= \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x) \in \mathcal{R}^{n\times n}. \end{aligned}$$

Using the third equation we eliminate

$$\Delta z = \mu Y^{-1}e - Ze - ZY^{-1}\Delta y,$$

from the second equation and get

$$\begin{bmatrix} -Q(x,y) & A(x)^T \\ A(x) & ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ -\Delta y \end{bmatrix} = \begin{bmatrix} \nabla f(x) + A(x)^T y \\ -g(x) - \mu Y^{-1} e \end{bmatrix}. \tag{9}$$

The matrix involved in this set of linear equations is symmetric and indefinite. For convex optimization problem (when f and g are convex), the matrix Q is positive semidefinite and if f is strictly convex, Q is positive definite and the matrix in (9) is quasidefinite. Similarly to the case of quadratic programming by eliminating Δx from the first equation we can reduce this system further to the form of normal equations

$$\left(A(x)Q(x,y)^{-1}A(x)^T + ZY^{-1}\right)\Delta y = b_{NLP}.$$

The two systems (6) and (9) have many similarities. The main difference is that in (6) only the diagonal scaling matrix Θ_1 changes from iteration to iteration, while in the case of nonlinear programming not only the matrix $\Theta_2 = Z^{-1}Y$ but also the matrices $Q(x, y)$ and $A(x)$ in (9) change in every iteration. Both these systems are indefinite. However, to avoid the need of using 2 × 2 pivots in their factorization we transform them to quasidefinite ones by the use of primal and dual regularization [1].

Our analysis in the following sections is concerned with the quadratic optimization problems, and hence A and Q are constant matrices. However, the major conclusions can be generalized to the nonlinear optimization case.

3. Iterative methods

Direct methods used to compute the Newton direction from the system like (5) or (8) are supposed to produce accurate solutions. In the case of quadratic programming, for example, one would compute $(\Delta x, \Delta y, \Delta s)$ from (5) and then verify that the residuals:

$$\begin{bmatrix} r_p \\ r_d \\ r_\mu \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix} - \begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix}, \tag{10}$$

are small compared with the right-hand-side of (5). Of course, due to the presence of numerical errors and possible ill-conditioning of the linear system involved in these equations, the accuracy may occasionally by lost. However, such situations are rare and can usually be cured for example by the use of iterative refinement method. It is common to ask for the error (the infinity norm of the residuals) to be below the threshold of $\epsilon = 10^{-8}$ [16].

Interior point methods work with a particular perturbation of the optimality conditions in the optimization problems. The perturbation originates from the use of logarithmic barrier functions to handle inequality constraints. Let us write the system (5), where we set $\mu = 0$, as:

$$F(x, y, s) = \begin{bmatrix} G(x, y, s) \\ XSe \end{bmatrix} = 0, \quad (x, s) \geq 0. \tag{11}$$

The interior point algorithm can be viewed as an inexact Newton method [9] applied to system (11). A more general perturbation could be used and the same framework of *inexact* interior point method would apply in which the linear equation system (11) is solved only to an approximate solution [5, 6]. Iterative methods are well suited to such a framework. They gradually reduce the error in (10) and they are usually asked for a significantly less accurate solutions than that delivered by the direct approach.

Using the Newton method, the interior point algorithm for (11) requires to solve at every iteration a linear system of the form

$$\nabla F(v_k)(\Delta v) = -F(v_k) + \sigma_k \mu_k e_0, \tag{12}$$

where $v = (x, y, s) \in \mathcal{R}^{2n+m}$, $\mu_k = (x_k^T s_k)/n$, $\sigma_k \in (0, 1)$ and $e_0 = (0, \ldots, 0, 1, \ldots 1)^T \in \mathcal{R}^{2n+m}$. An interior point method in which the linearized system is solved approximately (by means of an iterative method) may be considered as an *inexact (or truncated) interior point method* [6, 7]. In this framework, the system (12) becomes

$$\nabla F(v_k)(v_{k+1} - v_k) = -F(v_k) + \sigma_k \mu_k e_0 + r_k, \tag{13}$$

where r_k is the residual of the iterative method applied to the linear system. This residual satisfies $\|r_k\| \leq \eta_k \mu_k$, where η_k is, for every k, the *forcing term* of the inexact Newton method [9]. Global convergence of such a scheme may be achieved by means of backtracking.

We have implemented several iterative approaches for indefinite system including BiCGSTAB [12, 27], GMRES [26] and a simplified variant of QMR, QMRs [13], which is particularly suited for symmetric indefinite systems. The reader interested in more detail in the iterative methods for solving nonlinear equations is referred to Kelley [20].

We have also added the classical Conjugate Gradient (PCG). The conjugate gradient method is not guaranteed to converge when applied to an indefinite matrix. However, the particular choice of the preconditioner, which will be presented in the next section, makes the preconditioned matrix very similar to a positive definite one with all eigenvalues strictly positive. In spite of the indefiniteness of the preconditioned system, it is shown in [25] that CG can be efficiently applied to this problem and its asymptotic rate of convergence is approximately the same as that obtained for a positive definite matrix with the same eigenvalues as the original system. In [17] a variant of the PCG is proposed based on the projection of the augmented system onto a basis Z which spans the null space of A.

4. Preconditioners

In this section we shall discuss the properties of the preconditioned matrices involved in (1). For ease of the presentation we shall focus on the quadratic programming case with linear equality constraints hence we will assume that $\Theta_2^{-1} = 0$ (we also drop the subscript in Θ_1). In Section 4.2 we shall show how our results can be extended to the nonlinear programming case.

Due to the presence of matrix Θ^{-1}, the augmented system

$$H = \begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix}. \tag{14}$$

is very ill-conditioned. Indeed, some elements of Θ tend to zero while others tend to infinity as the optimal solution of the problem is approached. The performance of any iterative method critically depends on the quality of the preconditioner in this case.

We apply the iterative method to the indefinite augmented system and hence define the preconditioner also for this system. We use a block-type preconditioner. Such preconditioners are widely used in linear systems obtained from a discretization of partial differential equations [15, 24]. The preconditioners for the augmented system have also been used in the context of linear programming [14, 23] and in the context of nonlinear programming [11, 19, 21, 22, 25]. As was shown in [23], the preconditioners for indefinite augmented system offer more freedom than those for the normal equations. Moreover, the factorization of the augmented system is sometimes much easier than that of the normal equations [2] (this is the case, for example, when A contains dense columns). Hence even in the case of linear programming (in which normal equations is a viable approach) augmented system offers important advantages. For quadratic and nonlinear programming the use of the augmented system is a must and so we deal with the augmented system preconditioners in this paper. On the other hand, we realize that for some specially structured problems such as multicommodity network flows, very efficient preconditioners for the normal equations [8, 18] can also be designed.

The preconditioner needs to involve matrix Θ^{-1} to capture the key numerical properties of (14). However, we need to make it significantly less expensive than the direct factorization of this matrix so we drop out all off-diagonal elements in the matrix Q. Thus, instead of the augmented system (14) we use

$$P = \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix}, \tag{15}$$

Table 1. Values of m, n, nonzeros in A, off-diagonal nonzeros in Q and in the triangular factors L: for augmented matrix nz(L), for preconditioner 1 nz(L)-AS, and for preconditioner 2 nz(L)-NE.

Problem	m	n	nz(A)	nz(Q)	nz(L)	nz(L)-AS	nz(L)-NE
qp500-1	1100	500	56750	89071	181999	181999	604375
qp500-2	1000	500	51788	89071	177037	177037	499438
qp500-3	100	500	5614	89071	128837	10058	4944
qp1000-1	700	1000	15753	68762	491059	248824	231199
CVXQP1	7500	15000	22497	44981	6619618	104659	123450
CVXQP2	3750	15000	11249	44981	5225664	15665	5140
cvxqp1_l	5000	10000	14998	29984	3725045	71833	89241
cvxqp2_l	2500	10000	7499	29984	2754141	10579	3379
cvxqp3_l	7500	10000	22497	29984	4291057	149488	271780
sqp2500_1	2000	2500	52321	738051	3124093	2029613	1909672
sqp2500_2	2000	2500	52319	14345	3504910	2055283	1909275
sqp2500_3	4500	2500	115073	738051	3219994	3186754	9874267
q25fv47	820	1571	11127	59053	111760	51816	31102
qpilotnov	975	2172	13129	391	55741	54757	47015

where $E = -\text{diag}(Q) - \Theta^{-1}$. By dropping all off-diagonal elements from Q we expect an important gain in the sparsity of the Cholesky-like factor, see for example the statistics of CVXQP test problems in Table 1. Indeed, by removing the off-diagonal elements from Q we increase the number of columns in P with the small numbers of nonzero elements: such columns become more attractive candidates for the minimum degree heuristic applied to P. Moreover, we open the possibility of reducing the augmented system to the normal equations form whenever the latter offers any advantages. The preconditioner has one general algebraic form (15) but we may compute it in two equivalent ways.

Preconditioner 1 (AS): Compute the Cholesky-like factorization:

$$P_1 = \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix} = LDL^T. \tag{16}$$

Preconditioner 2 (NE): Reduce the system to normal equations $AE^{-1}A^T$, compute the Cholesky factorization

$$AE^{-1}A^T = L_0 D_0 L_0^T,$$

and use:

$$P_2 = \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ AE^{-1} & I \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & -AE^{-1}A^T \end{bmatrix} \begin{bmatrix} I & E^{-1}A^T \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ AE^{-1} & L_0 \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & -D_0 \end{bmatrix} \begin{bmatrix} I & E^{-1}A^T \\ 0 & L_0^T \end{bmatrix}. \tag{17}$$

The two forms differ only in the implementation, being mathematically equivalent. We can therefore carry out the spectral analysis of the preconditioned matrix for both.

Block preconditioners have been widely used and analyzed in the literature. In particular, in [19, 21, 25] the preconditioner similar to the one used in this paper has been analyzed. We have decided to recall the spectral analysis (Theorem 4.1) because it indicates the ways of improving the numerical properties of the preconditioner. We then extend the spectral analysis for the preconditioned quasidefinite matrix which includes regularization (see Theorem 4.2 and the subsequent Corollaries).

4.1. Spectral analysis

The (left) preconditioned matrix reads

$$P^{-1}H = \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix}^{-1} \begin{bmatrix} C & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} E^{-1}C - E^{-1}A^TM^{-1}AJ & 0 \\ M^{-1}AJ & I_m \end{bmatrix}$$

$$= \begin{bmatrix} X & 0 \\ Y & I_m \end{bmatrix} \tag{18}$$

where

$$M = AE^{-1}A^T, \quad J = E^{-1}C - I, \quad C = -\Theta^{-1} - Q, \quad E = \text{diag}(C).$$

We now prove a theorem which establishes the well-conditioning of preconditioned matrix $P^{-1}H$ relating its non-unit eigenvalues to those of the matrix $E^{-1}C$, whose spectrum is strictly positive. In fact $E^{-1}C$ is similar to $(-E)^{-1/2}(-C)(-E)^{-1/2}$ which is trivially symmetric positive definite under the hypothesis that C is negative definite.

Theorem 4.1. *Assume that A has maximum rank. Then, if $m < n$ at least $2m$ eigenvalues of $P^{-1}H$ are unit, and the other eigenvalues are positive and satisfy*

$$\frac{\lambda_{\max}(P^{-1}H)}{\lambda_{\min}(P^{-1}H)} \leq \kappa(E^{-1}C).$$

If $m \geq n$ all the eigenvalues are unit.

Proof: Matrix $P^{-1}H$ in (18) is block lower triangular. It has m unit eigenvalues and the remaining n eigenvalues are those of matrix

$$X = E^{-1}C - E^{-1}A^T M^{-1}A(E^{-1}C - I).$$

We will now consider two cases: $m < n$ and $m \geq n$.

$\boxed{\text{Case } m < n}$

Matrix X has at least m other unit eigenvalues. In fact write X as

$$X = I + (I - E^{-1}A^T M^{-1}A)(E^{-1}C - I) = I + (I - E^{-1}A^T M^{-1}A)J$$

then for every vector $x \in \mathcal{R}^m$

$$X^T A^T x = (I + J^T(I - A^T M^{-1}AE^{-1}))A^T x = A^T x + J^T(A^T x - A^T x) = A^T x$$

so that $A^T x$ is (left) eigenvector of X with 1 as eigenvalue. The remaining $n-m$ eigenvalues and eigenvectors of $P^{-1}H$ must satisfy

$$\begin{bmatrix} C & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \tag{19}$$

or

$$\begin{cases} Cx + A^T y = \lambda Ex + \lambda A^T y \\ Ax \qquad\quad = \lambda Ax \end{cases}. \tag{20}$$

If $\lambda \neq 1$ the second equation provides $Ax = 0$ (with $x \neq 0$). Let us multiply the first equation by x^T. Recalling that $x^T A^T = 0$ we obtain

$$x^T Cx = \lambda x^T Ex, \Rightarrow \lambda = \frac{x^T Cx}{x^T Ex} = q(E^{-1}C). \tag{21}$$

The last expression is the Rayleigh quotient of the generalized eigenproblem $Cv = \mu Ev$. Since both E and C are negative definite we have for every $x \in \mathcal{R}^n$

$$0 < \lambda_{\min}(E^{-1}C) \leq \frac{x^T Cx}{x^T Ex} \leq \lambda_{\max}(E^{-1}C)$$

and finally

$$\lambda_{\min}(E^{-1}C) \leq \lambda \leq \lambda_{\max}(E^{-1}C). \tag{22}$$

$\boxed{\text{Case } m \geq n}$

From the previous analysis we get that every vector of the form $A^T x$ is the left eigenvector of X. Since A has maximum rank, the space spanned by $A^T x,\ x \in \mathcal{R}^m$ has exactly size n, so that there are n eigenvectors corresponding to the unit eigenvalue. Summarizing, there are $m + n$ eigenvectors corresponding to the unit eigenvalue in this case. □

The claim that $P^{-1}H$ has $2m$ unit eigenvalues was already proved in [19].

In the most interesting case (more often met in the computational practice) when $m < n$, the $2m$ eigenvalues of $P^{-1}H$ have value 1 and the remaining $n - m$ eigenvalues satisfy $\lambda_{\min}(E^{-1}C) \leq \lambda \leq \lambda_{\max}(E^{-1}C)$. It is worth noting that the case when θ_j is very small (hence θ_j^{-1} is very large), the preconditioner brings the corresponding eigenvalues of $P^{-1}H$ close to unit. In the case when θ_j is very large (and θ_j^{-1} is very small), the conditioning of matrix Q plays more important role. Indeed, in this case the preconditioner behaves like the simple Jacobi preconditioner for matrix Q.

In any case the iterative method is very sensitive to the quality of the preconditioner and it is crucial to ensure the best possible accuracy of the factorization used in the computations of the preconditioner. In our implementation the system (15) is regularized at the time of factorization. Following [1] we use both primal and dual regularization and instead of (15) we use the regularized preconditioner

$$P_R = \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix} + \begin{bmatrix} -R_p & 0 \\ 0 & R_d \end{bmatrix}.$$

In the computations of the Newton step we replace the matrix (14) with the regularized one

$$H_R = \begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} + \begin{bmatrix} -R_p & 0 \\ 0 & R_d \end{bmatrix}.$$

The diagonal positive definite matrices $R_p \in \mathcal{R}^{n\times n}$ and $R_d \in \mathcal{R}^{m\times m}$ are chosen during the factorization process. For all pivots that accept stability criteria the negligibly small terms are used while for unstable pivots the stronger regularization terms are used. The reader interested in more detail in this approach is referred to [1]. The regularization technique was shown to be successful in the implementation of the direct approach of the HOPDM interior point solver and it also plays the key role in the stability of our iterative approach. The following theorem provides an insight into its role in the computations.

Theorem 4.2. *Assume that A has maximum rank. Then at least m eigenvalues of $P_R^{-1}H_R$ are unit, and the other eigenvalues are positive and satisfy*

$$\lambda\big(P_R^{-1}H_R\big) = \frac{-x^T C x + \delta}{-x^T E x + \delta} > 0$$

for some $\delta > 0,\ x \neq 0$.

Proof: It can easily be shown that left preconditioned matrix has the same block form as the preconditioned matrix without regularization (Theorem 4.1):

$$P_R^{-1}H_R = \begin{bmatrix} X' & 0 \\ Y' & I_m \end{bmatrix}.$$

As in Theorem 4.1 it follows that $P_R^{-1}H_R$ have m linearly independent eigenvectors associated to the unit eigenvalue.

The other eigenpairs of $P_R^{-1}H_R$ must satisfy

$$\begin{bmatrix} C - R_p & A^T \\ A & R_d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda\begin{bmatrix} E - R_p & A^T \\ A & R_d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \tag{23}$$

or

$$\begin{cases} (C - R_p)x + A^T y = \lambda(E - R_p)x + \lambda A^T y \\ Ax + R_d\, y = \lambda Ax + \lambda R_d\, y. \end{cases} \tag{24}$$

If $\lambda \neq 1$ the second equation provides $Ax + R_d y = 0$ (with $x \neq 0$). Let us multiply the first equation by x^T. Recalling that $x^T A^T = -y^T R_d$ we obtain

$$x^T(C - R_p)x - y^T R_d\, y = \lambda x^T(E - R_p)x - y^T \lambda R_d\, y$$

whence, setting $\delta = x^T R_p\, x + y^T R_d\, y$ (≥ 0)

$$\lambda = \frac{x^T Cx - x^T R_p\, x - y^T R_d\, y}{x^T Ex - x^T R_p\, x - y^T R_d\, y} = \frac{-x^T Cx + \delta}{-x^T Ex + \delta} > 0.$$

The last inequality follows from the negative definiteness of C and E. □

In the proof of the next Corollary we will use the Lemma:

Lemma 4.3. *If A is a symmetric positive definite matrix and $D = \operatorname{diag}(A)$ then the eigenvalues of matrix $D^{-1}A$ are either all ones or are contained in a nontrivial interval $[\alpha, \beta]$ with $0 < \alpha < 1 < \beta$.*

Proof: See for example [25]. □

Corollary 4.4. *The eigenvalues of $P_R^{-1}H_R$ satisfy*

$$\min\{\lambda_{\min}(E^{-1}C), 1\} \leq \lambda_{\min}\bigl(P_R^{-1}H_R\bigr) \leq \lambda_{\max}\bigl(P_R^{-1}H_R\bigr) \leq \max\{\lambda_{\max}(E^{-1}C), 1\}.$$

Proof: For any $\alpha > 0$, $\beta > 0$ and $t > 0$, the function $h(t) = \frac{\alpha+t}{\beta+t}$ is increasing if $\frac{\alpha}{\beta} \leq 1$ and decreasing if $\frac{\alpha}{\beta} > 1$. Now let (x_m, y_m) and (x_M, y_M) be the eigenvectors of $P_R^{-1}H_R$ related to the smallest and the largest eigenvalue, respectively, then

$$\lambda_{\min}\left(P_R^{-1}H_R\right) = \frac{-x_m^T C x_m + \delta}{-x_m^T E x_m + \delta} > \min\left\{1, \frac{x_m^T C x_m}{x_m^T E x_m}\right\} \geq \min\{\lambda_{\min}(E^{-1}C), 1\}$$

$$\lambda_{\max}\left(P_R^{-1}H_R\right) = \frac{-x_M^T C x_M + \delta}{-x_M^T E x_M + \delta} < \max\left\{1, \frac{x_M^T C x_M}{x_M^T E x_M}\right\} \leq \max\{\lambda_{\max}(E^{-1}C), 1\}$$

and hence the thesis. □

Corollary 4.4 indicates that the use of primal and dual regularization improves the clustering of the eigenvalues in the preconditioned matrix.

By a similar argument to that in Corollary 4.4 we prove that the eigenvalues of $E^{-1}C$ and hence those of $P_R^{-1}H_R$ are bounded away from zero.

Corollary 4.5. *Let Q be a symmetric positive definite matrix and define $D = \mathrm{diag}(Q)$. The eigenvalues of $E^{-1}C$ satisfy*

$$\min\{\lambda_{\min}(D^{-1}Q), 1\} \leq \lambda_{\min}(E^{-1}C) \leq \lambda_{\max}(E^{-1}C) \leq \max\{\lambda_{\max}(D^{-1}Q), 1\}.$$

Proof: We write

$$\frac{x^T(Q+\Theta^{-1})x}{x^T E x} = \frac{x^T(Q+\Theta^{-1})x}{x^T(D+\Theta^{-1})x} = \frac{x^T Q x + x^T\Theta^{-1}x}{x^T D x + x^T \Theta^{-1} x} = \frac{x^T Q x + \delta}{x^T D x + \delta},$$

where $\delta = x^T\Theta^{-1}x$, and use similar arguments to those in the proof of Corollary 4.4. □

Let us define $\rho = \lambda_{\min}(D^{-1}Q)$. Clearly $\rho > 0$ because Q is positive definite. This number does not depend on Θ and provides a strictly positive lower bound for the eigenvalues of $E^{-1}C$. By combining Corollaries 4.4 and 4.5 we conclude that $\min\{\rho, 1\}$ is also a positive lower bound for the eigenvalues of $P_R^{-1}H_R$ and the condition number of matrix $P_R^{-1}H_R$ satisfies the following bound

$$\frac{\lambda_{\max}\left(P_R^{-1}H_R\right)}{\lambda_{\min}\left(P_R^{-1}H_R\right)} \leq \max\left\{\frac{\lambda_{\max}(D^{-1}Q)}{\lambda_{\min}(D^{-1}Q)}, \lambda_{\min}^{-1}(D^{-1}Q), \lambda_{\max}(D^{-1}Q)\right\}.$$

The preconditioner analyzed above yields a preconditioned matrix which has actually a large part of the spectrum clustered around one. The following two figures, which are

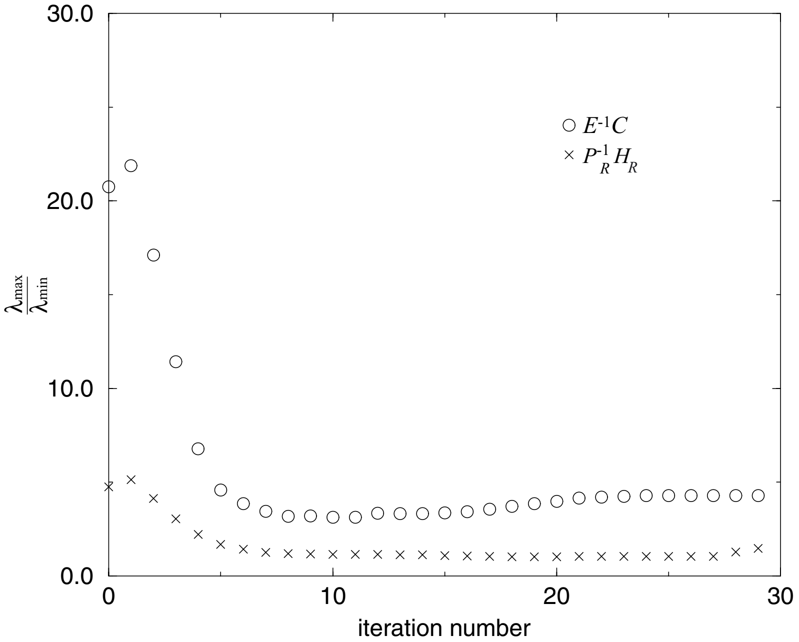

Figure 1. Values of $\kappa(P_R^{-1}H_R)$ and $\kappa(E^{-1}C)$ vs nonlinear iteration number for the `q25fv47` test problem.

related to the `q25fv47` problem with $n = 1571$, $m = 820$ (see the tables in Section 5) give an example of this behavior.

In figure 1 the ratio κ between the largest and the smallest eigenvalue is plotted for the $P_R^{-1}H_R$ and $E^{-1}C$ matrices. From the figure we note that $\kappa(P_R^{-1}H_R)$ is lower than $\kappa(E^{-1}C)$ for every nonlinear iteration, as stated in Corollary 4.4. Moreover $\kappa(P_R^{-1}H_R)$ is smaller than five with the only exception of the second iteration.

In figure 2 we report the number of eigenvalues in the preconditioned matrices which are outside the interval [0.9, 1.1]. When this number is small or even zero it means that the eigenvalues are clustered around one and fast convergence of Krylov solvers can be expected. The figure shows that once again the eigenvalues of $P_R^{-1}H_R$ are better clustered than those of $E^{-1}C$. In 19 nonlinear iterations out of 29 $P_R^{-1}H_R$ has at most one eigenvalue outside [0.9, 1.1].

4.2. Spectral analysis for the nonlinear case

The augmented matrix in the nonlinear case reads

$$H = \begin{bmatrix} -Q & A^T \\ A & ZY^{-1} \end{bmatrix}. \tag{25}$$

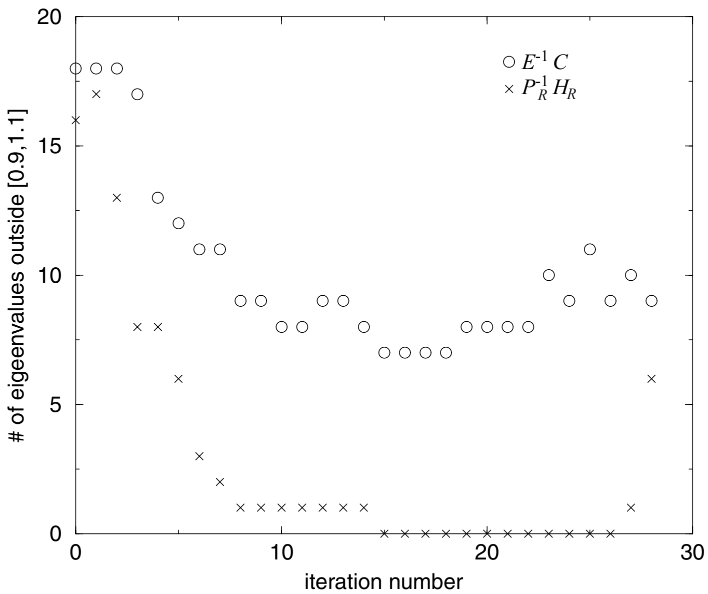

Figure 2. Number of eigenvalues of $P_R^{-1}H_R$ and $E^{-1}C$ outside the interval [0.9, 1.1] vs nonlinear iteration number for the q25fv47 test problem.

The preconditioner is defined by substituting the matrix Q with its diagonal, namely D_Q getting

$$P = \begin{bmatrix} -D_Q & A^T \\ A & ZY^{-1} \end{bmatrix}. \tag{26}$$

The following theorem establishes the well-conditioning of preconditioned matrix $P^{-1}H$ relating its non-unit eigenvalues to those of the matrix $D_Q^{-1}Q$.

Theorem 4.6. *Assume that A has maximum rank. Then at least m eigenvalues of $P^{-1}H$ are unit, and the other eigenvalues are positive and satisfy*

$$\lambda(P^{-1}H) = \frac{x^T Qx + \delta}{x^T D_Q x + \delta} > 0$$

with $\delta = y^T ZY^{-1} y$, for some $x, y, \ x \neq 0$. Moreover,

$$\min\{\lambda_{\min}(D_Q^{-1}Q), 1\} \le \lambda_{\min}(P^{-1}H) \le \lambda_{\max}(P^{-1}H) \le \max\{\lambda_{\max}(D_Q^{-1}Q), 1\}.$$

Proof: The proof of this theorem is analogous to the proofs of Theorem 4.2 and Corollary 4.4. □

5. Numerical results

The method discussed in this paper has been implemented in the context of HOPDM code [16]. HOPDM is an LP/QP solver that uses the higher order primal dual algorithm [1] and contains two factorization methods implemented: the normal equations and the augmented system. The accuracy is controlled in HOPDM through the use of primal and dual regularization. These features have contributed to the ease of the implementation of our preconditioner. All tests have been run on a dual 1.55 GHz Athlon with 1 GB RAM. We have used the pure FORTRAN version of the solver and we have compiled it with the `Portland Group f77` compiler with `-fast -pc 64 -Mvect=prefetch` options.

We have implemented four iterative approaches for indefinite system: BiCGSTAB [27], GMRES [26], QMRs [13], and the classical Conjugate Gradient. We have tested them with both preconditioners P_1 and P_2 from (16) and (17), respectively. We have solved a number of public domain quadratic programs. Some of them come from the three different directories: `brunel`, `cute` and `misc` available via the anonymous `ftp` from: `ftp://ftp.sztaki.hu/pub/oplab/QPDATA/`. Other examples have been made available to us by Professor Hans Mittelmann. The reader interested in them can retrieve AMPL files that generate these examples from `ftp` addresses: `ftp://plato.la.asu.edu/pub/vlgqp.txt` and `ftp://plato.la.asu.edu/pub/randqp.txt`. To avoid reporting excessive numerical results, we have selected a subset of 14 representative quadratic programs for which we give detailed solution statistics.

We start the analysis from the statistics of problems used in our computations. In Table 1 we report problem sizes m, n, the number of nonzero elements in matrix A, nz(A), the number of off-diagonal nonzero elements in Q, nz(Q), and in the three different factorizations: for the complete matrix H, nz(L), for the preconditioner P_1, nz(L)-AS, and for the preconditioner P_2, nz(L)-NE. The matrices Q, H and the preconditioners P_1 and P_2 are symmetric. Therefore we report the number of off-diagonal nonzero elements in the triangular parts stored by the solver. The analysis of results collected in Table 1 reveals that after removing the off-diagonal elements from matrix Q, spectacular gains in the sparsity of factorizations can sometimes be obtained. This is the case, for example, when problems `cvxqpi_l`, `qp500-3`, or `CVXQPi` are solved. Clearly, the more important the reductions in the fill-in of Cholesky factors the faster the computations of the preconditioner are. We would like to draw the reader's attention to the fact that the large number of off-diagonal nonzero elements in Q is not necessarily the only factor that contributes to the fill-in in the triangular factorization. Indeed, the randomly generated problems `qp500_i` have all the same density of Q and yet removing off-diagonal elements from Q leads to dramatic improvement in the factorization only in the case of `qp500_3` problem. It is also worth mentioning that there is no clear winner between the augmented system and the normal equations factorizations. Both approaches are worth to be implemented and a heuristic to make an automatic choice between them incorporated in HOPDM is a useful technique. However, we have disabled this automatic choice and we have forced the solver to use a specific factorization and produce either the preconditioner P_1 or P_2.

Table 2. Solution Statistics: comparisons between direct and iterative (PCG) solver.

	Direct		PCG-AS				PCG-NE			
Problem	Its	Time	Its	Time	LinIt	Avr	Its	Time	LinIt	Avr
qp500-1	14	8.03	17	14.24	347	10	17	83.27	347	10
qp500-2	16	8.73	20	15.45	404	10	20	70.51	404	10
qp500-3	7	2.88	42	6.24	1052	12	42	6.12	1052	12
qp1000-1	14	41.45	22	32.05	851	19	22	30.56	851	19
CVXQP1	7	877.32	12	9.55	397	15	12	9.91	397	15
CVXQP2	9	736.67	11	5.74	399	17	11	5.39	399	17
cvxqp1_l	11	551.51	13	6.41	439	16	13	6.82	439	16
cvxqp2_l	8	268.98	10	3.03	354	16	10	2.92	354	16
cvxqp3_l	8	488.68	10	7.50	332	15	10	10.64	354	16
sqp2500-1	15	804.52	18	486.85	749	20	18	535.56	749	20
sqp2500-2	16	913.82	19	457.43	748	19	19	485.55	748	19
sqp2500-3	18	961.62	24	1147.87	567	11	24	6099.37	567	11
q25fv47	21	3.57	29	3.64	359	6	28	2.82	317	5
qpilotnov	19	1.02	27	1.72	190	3	27	1.60	170	3

Getting a sparse preconditioner is clearly encouraging but it is not yet the guarantee of success. This depends on how many iterations are needed by the iterative method to obtain sufficiently accurate solutions in the linear systems solved for Newton directions. Table 2 collects the results of HOPDM runs on all test examples for three different methods: direct approach (factorization of H) and the preconditioned conjugate gradient method with two different preconditioners P_1 and P_2, respectively.

In all these runs we have used the preconditioned conjugate gradient method with a fixed termination criteria on the relative residual $\frac{\|r_k\|}{\|b\|} \leq \epsilon = 10^{-2}$ and the limit of iterations set to 20. We report in Table 2 the number of interior point iterations, Its, CPU time and for two variants of the preconditioned gradient method, we additionally report the overall number of iterations in the iterative solver, LinIt, and the average number of PCG iterations per backsolve, Avr. The multiple centrality correctors [1] are allowed only with the direct approach: one interior point iteration may then require the solution of several systems of equations. When iterative approach is used each interior point iteration needs the solution of two linear systems: one for the predictor direction and one for the centrality corrector direction [1]. The initialization procedure solves two additional linear systems. Hence if the problem is solved in k iterations with the iterative solver (the number reported in column Its in Table 2), then $2k + 2$ linear systems need to be solved. The analysis of results leads to the following conclusions:

- The preconditioned conjugate gradient method behaves well even with fixed tolerance. There are no failures due to lack of precision in the Newton directions although rather loose tolerance is used. There is certainly room for improvement of these results by tuning this tolerance and adjusting it closer to the needs of the interior point algorithm.

- The use of less precise directions usually increases the number of interior point iterations by no more than 50% (qp500-3 displays an exceptional increase).
- The iterative method is advantageous whenever the factorization of the preconditioner is significantly less expensive than that of the complete augmented system (the problem classes cvxqpi_l and CVXQPi are spectacular examples of it).

Our preconditioners replace matrix Q with its diagonal part. Hence there is no interest in using the preconditioner if the problem is separable (i.e., Q is diagonal). We have also noticed that for problems with a small number of constraints or problems in which matrix Q has only a few off-diagonal entries, the savings resulting from the use of the factorization of the preconditioner (which is only slightly sparser than the KKT system) are insignificant and do not compensate for additional operations required by the iterative method. Therefore in the production code we do not allow to use the iterative method if $m \leq 500$ or $nz(Q) \leq 10$, where nz(Q) denotes the number of off-diagonal nonzero elements in the triangular part of the (symmetric) matrix Q.

We have run the solver on a collection of public domain quadratic programs available from ftp://ftp.sztaki.hu/pub/oplab/QPDATA/ and ftp://plato.la.asu.edu/pub/vlgqp.txt. These test sets contain 46 brunel, 76 cute, 16 misc and 27 vlgqp problems. The numbers of nonseparable problems in these four sets are: 46, 33, 4 and 16, respectively. However, many of them (especially those in the brunel and cute collections) do not meet the criteria to call the iterative solver. The iterative method was used to solve 40 problems overall: 15, 10, 2 and 13 problems from these sets, respectively.

The overall solution time for the 40 test examples is 49909 seconds with the direct method and 4413 seconds with the iterative method. This gives the average time reduction by a factor of 11.3.

We have also tested other iterative methods: BiCGSTAB, GMRES and a simplified variant of QMR. In Table 3 we report these results for the same fixed tolerance on the relative residual $\frac{\|r_k\|}{\|b\|} \leq \epsilon = 10^{-2}$ and the same limit of iterations ITMAX set to 20 for three test problems: qp500_3, cvxqp1_l and CVXQP1.

The CG solver is the fastest (or very close to the fastest) among the four solvers in 13 problems out of 14. However, there are slight differences in terms of CPU time and linear-nonlinear iterations among the solvers. The qp500-3 problem is the only one in which the CG solver is surprisingly slow, since it requires a number of interior point iterations which varies from two to six times more than that of a direct approach.

The tolerance ϵ and the maximum number of iterations (ITMAX) can dramatically affect the nonlinear convergence and the overall CPU time. In Tables 4 and 5 we report the statistics of qp500-3 and cvxqp1_l problems solved with PCG-AS and BiCGSTAB-AS, respectively, with different choices of these two parameters. We note that “oversolving”, that is solving the linear system to a high precision, is not necessary since it produces a slight reduction of the number of nonlinear iterations at the price of a drastic increase of the total linear iterations and hence the CPU time.

Table 3. Comparison of four iterative methods.

		AS preconditioner			NE preconditioner		
Problem	Solver	Its	LinIt	Time	Its	LinIt	Time
cvxqp1_l	GMRES	13	547	9.07	12	508	8.96
	BiCGstab	13	313	8.58	13	314	9.20
	QMRs	13	420	9.96	13	420	10.89
	CG	13	439	6.41	13	439	6.82
CVXQP1	GMRES	32	601	21.02	34	591	21.67
	BiCGstab	12	286	12.67	12	285	13.27
	QMRs	11	353	13.59	11	353	14.28
	CG	12	397	9.55	12	397	9.91
qp500_3	GMRES	22	571	3.45	23	592	3.51
	BiCGstab	15	406	4.19	12	340	3.44
	QMRs	22	583	3.72	22	583	3.60
	CG	42	1052	6.24	42	1052	6.12

Table 4. Performance of PCG with AS preconditioner.

ϵ	ITMAX	Its	LinIt	Time	ϵ	ITMAX	Its	LinIt	Time
10^{-1}	10	44	606	4.60	10^{-1}	2	20	79	3.06
10^{-2}	20	42	1052	6.24	10^{-1}	10	13	287	5.18
10^{-8}	100	10	1548	8.27	10^{-2}	20	13	439	6.41
10^{-8}	1000	13	11125	55.73	10^{-8}	100	13	782	23.04

Influence of tolerance ϵ and maximum number of iterations ITMAX on the nonlinear convergence for qp500-3 problem (left) and cvxqp1_l problem (right).

Table 5. Performance of BiCGstab with AS preconditioner.

ϵ	ITMAX	Its	LinIt	Time	ϵ	ITMAX	Its	LinIt	Time
10^{-1}	10	16	247	3.01	10^{-1}	2	14	59	3.24
10^{-2}	20	15	406	4.19	10^{-1}	10	13	241	7.56
10^{-8}	100	10	1376	13.70	10^{-2}	20	13	313	8.58
10^{-8}	1000	10	6504	64.70	10^{-8}	100	13	770	21.73

Influence of tolerance ϵ and maximum number of iterations ITMAX on the nonlinear convergence for qp500-3 problem (left) and cvxqp1_l problem (right).

6. Conclusions

We have discussed in this paper the use of iterative approaches for the solution of Newton equation system arising in interior point method for quadratic and nonlinear optimization.

The crucial issue in the implementation of such approaches is the use of an appropriate preconditioner that can be computed with a small cost compared with a direct approach and can capture the key numerical properties of the linear system. We have designed such a preconditioner by dropping off-diagonal elements from the Hessian matrix. The preconditioner has well clustered eigenvalues and this clustering can further be improved by the use of the primal and dual regularization method.

We have provided computational evidence that our preconditioner works well for nontrivial quadratic programming problems. In some cases, the use of the preconditioned iterative solver reduces the solution time by a factor of 100 compared with a direct approach.

Acknowledgments

We are grateful to the anonymous referees for constructive comments, resulting in an improved presentation.

References

1. A. Altman and J. Gondzio, “Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization,” Optimization Methods and Software, vol. 11/12, pp. 275–302, 1999.
2. E.D. Andersen, J. Gondzio, C. Mészáros, and X. Xu, “Implementation of interior point methods for large scale linear programming,” in Interior Point Methods in Mathematical Programming, T. Terlaky (Ed.), Kluwer Academic Publishers, pp. 189–252, 1996.
3. M. Arioli, I.S. Duff, and P.P.M. de Rijk, “On the augmented system approach to sparse least-squares problems,” Numerische Mathematik, vol. 55, pp. 667–684, 1989.
4. V. Baryamureeba, “Solution of large-scale weighted least-squares problems,” Numerical Linear Algebra with Applications, vol. 9, pp. 93–106, 2002.
5. V. Baryamureeba and T. Steihaug, “On the convergence of an inexact primal-dual interior point method for linear programming,” Tech. Rep. 188, Department of Informatics, University of Bergen, Norway, 2000.
6. S. Bellavia, “An inexact interior point method,” Journal of Optimization Theory and Applications, vol. 96, pp. 109–121, 1998.
7. L. Bergamaschi and G. Zilli, “Parallel inexact Newton and interior point method,” Ann. Univ. Ferrara - Sez VII - Sc. Mat., XLV, pp. 467–478, 2000.
8. J. Castro, “A specialized interior-point algorithm for multicommodity network flows,” SIAM Journal on Optimization, vol. 10, pp. 852–877, 2000.
9. R.S. Dembo, S.C. Eisenstat, and T. Steihaug, “Inexact Newton methods,” SIAM Journal on Numerical Analysis, vol. 19, pp. 400–408, 1982.
10. I.S. Duff, A.M. Erisman, and J.K. Reid, “Direct Methods for Sparse Matrices,” Oxford University Press: New York, 1987.
11. C. Durazzi and V. Ruggiero, “Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems,” Numerical Linear Algebra with Applications, vol. 10, pp. 673–688, 2003.
12. R. Fletcher, “Conjugate Gradient Methods for Indefinite Systems,” in Numerical Analysis Dundee 1975, G. Watson (Ed.), Springer-Verlag: Berlin, New York, 1976, pp. 73–89.
13. R.W. Freund and N.M. Nachtigal, “Software for simplified Lanczos and QMR algorithms,” Applied Numerical Mathematics, vol. 19, pp. 319–341, 1995.
14. P.E. Gill, W. Murray, D.B. Ponceleon, and M.A. Saunders, “Preconditioners for indefinite systems arising in optimization,” SIAM Journal on Matrix Analysis and Applications, vol. 13, pp. 292–311, 1992.
15. G.H. Golub and A.J. Wathen, “An iteration for indefinite systems and its application to the Navier-Stokes equations,” SIAM Journal on Scientific Computing, vol. 19, pp. 530–539, 1998.

16. J. Gondzio, "HOPDM (version 2.12)—a fast LP solver based on a primal-dual interior point method," European Journal of Operational Research, vol. 85, pp. 221–225, 1995.
17. N.I.M. Gould, M.E. Hribar, and J. Nocedal, "On the solution of equality constrained quadratic problems arising in optimization," SIAM Journal on Scientific Computing, vol. 23, pp. 1375–1394, 2001.
18. J.J. Júdice, J. Patricio, L.F. Portugal, M.G.C. Resende, and G. Veiga, "A study of preconditioners for network interior point methods," Computational Optimization and Applications, vol. 24, pp. 5–35, 2003.
19. C. Keller, N.I.M. Gould, and A.J. Wathen, "Constraint preconditioning for indefinite linear systems," SIAM Journal on Matrix Analysis and Applications, vol. 21, pp. 1300–1317, 2000.
20. C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, vol. 16 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.
21. L. Lukšan and J. Vlček, "Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems," Numerical Linear Algebra with Applications, vol. 5, pp. 219–247, 1998.
22. L. Lukšan and J. Vlček, "Interior point method for nonlinear nonconvex optimization," Tech. Rep. 836, Institute of Computer Science, Academy of Sciences of the Czech Republic, 18207 Prague, Czech Republic, 2001.
23. A.R.L. Oliveira, A new class of preconditioners for large-scale linear systems from interior point methods for linear programming, PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005, 1997.
24. I. Perugia and V. Simoncini, "Block-diagonal and indefinite symmetric preconditioners for mixed finite elements formulations," Numerical Linear Algebra with Applications, vol. 7, pp. 585–616, 2000.
25. M. Rozložník and V. Simoncini, "Krylov subspace methods for saddle point problems with indefinite preconditioning," SIAM Journal of Matrix Analysis and Applications, vol. 24, pp. 368–391, 2002.
26. Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," SIAM Journal on Scientific and Statistical Computing, vol. 7, pp. 856–869, 1986.
27. H.A. Van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant to Bi-CG for the solution of nonlinear systems," SIAM Journal on Scientific and Statistical Computing, vol. 13, pp. 631–644, 1992.
28. R.J. Vanderbei, "Symmetric quasidefinite matrices," SIAM Journal on Optimization, vol. 5, pp. 100–113, 1995.
29. W. Wang and D.P. O'Leary, "Adaptive use of iterative methods in predictor-corrector interior point methods for linear programming," Numerical Algorithms, vol. 25, pp. 387–406, 2000.
30. S.J. Wright, Primal-Dual Interior-Point Methods, SIAM, Philadelphia, 1997.