Comput Optim Appl (2016) 65:339–360
DOI 10.1007/s10589-016-9830-4

On the update of constraint preconditioners for regularized KKT systems

Stefania Bellavia[1] · Valentina De Simone[2] · Daniela di Serafino[2] · Benedetta Morini[1]

Received: 10 March 2014 / Published online: 15 February 2016

Abstract We address the problem of preconditioning sequences of regularized KKT systems, such as those arising in interior point methods for convex quadratic programming. In this case, constraint preconditioners (CPs) are very effective and widely used; however, when solving large-scale problems, the computational cost for their factorization may be high, and techniques for approximating them appear as a convenient alternative. Here, given a block LDL^T factorization of the CP associated with a KKT matrix of the sequence, called seed matrix, we present a technique for updating the factorization and building inexact CPs for subsequent matrices of the sequence. We have recently proposed an updating procedure that performs a low-rank correction of the Schur complement of the (1,1) block of the CP for the seed matrix. Now we focus on KKT sequences with nonzero (2,2) blocks and make a step further, by enriching the

Work partially supported by INdAM-GNCS (2015 Projects *Metodi di regolarizzazione per problemi di ottimizzazione e applicazioni* and *Metodi numerici per l'ottimizzazione non convessa o non differenziabile e applicazioni).*

✉ Daniela di Serafino
daniela.diserafino@unina2.it

Stefania Bellavia
stefania.bellavia@unifi.it

Valentina De Simone
valentina.desimone@unina2.it

Benedetta Morini
benedetta.morini@unifi.it

1 Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, viale Morgagni 40, 50134 Florence, Italy

2 Dipartimento di Matematica e Fisica, Seconda Università degli Studi di Napoli, viale A. Lincoln 5, 81100 Caserta, Italy

low-rank correction of the Schur complement by an additional cheap update. The latter update takes into account information not included in the former one and expressed as a diagonal modification of the low-rank correction. Theoretical results and numerical experiments show that the new strategy can be more effective than the procedure based on the low-rank modification alone.

Keywords KKT systems · Constraint preconditioners · Matrix updates · Interior point methods

Mathematics Subject Classification 65F08 · 65F10 · 90C20

1 Introduction

We consider the problem of preconditioning a sequence of regularized KKT systems of the following form:

$$\begin{bmatrix} B_k & A^T \\ A & -\Theta_k \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} f_k \\ g_k \end{bmatrix}, \quad k = 1, 2, \ldots, \tag{1}$$

where $B_k \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $A \in \mathbb{R}^{m\times n}$ is full rank, $m \le n$, and $\Theta_k \in \mathbb{R}^{m\times m}$ is diagonal positive semidefinite. We focus on problems where $\Theta_k \ne 0$ and its sparsity pattern does not change throughout the sequence. We further assume that the above linear systems are large and possibly sparse. Linear systems of this form arise, e.g. in interior point (IP) methods for convex quadratic programming problems [25,34]:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}x^T Qx + c^T x, \\ \text{Subject to } A_1 x - s = b_1, & A_2 x = b_2, \quad x + v = u, \quad (x, s, v) \ge 0, \end{array} \tag{2}$$

where $A_1 \in \mathbb{R}^{m_1\times n}$, $A_2 \in \mathbb{R}^{m_2\times n}$, and s and v are slack variables, used to transform the inequality constraints $A_1 x \ge b_1$ and $x \le u$ into equality constraints.

In this case we have

$$B_k = Q + \Phi_k, \quad \Phi_k = X_k^{-1} W_k + V_k^{-1} T_k, \quad \Theta_k = \begin{bmatrix} Y_k^{-1} S_k & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where X_k, W_k, S_k, Y_k, V_k and T_k are suitable diagonal matrices and $m = m_1 + m_2$. More precisely, letting (x_k, w_k), (s_k, y_k) and (v_k, t_k) be the pairs of complementary variables of problem (2) evaluated at the k-th iteration, we have that X_k, W_k, S_k, Y_k, V_k and T_k are the diagonal matrices where the diagonal is equal to the corresponding lowercase vector, according to the usual IP notation. A nonzero block Θ_k arises whenever the problem has linear inequality constraints, i.e., $m_1 \ne 0$, as in this case Θ_k admits m_1 positive diagonal entries corresponding to the slack variables for the linear inequality constraints; furthermore, the number and the position of the nonzero entries of Θ_k does not change throughout the IP iterations. It is well known that the nonzero entries of Φ_k and Θ_k usually display a drastic difference of magnitude: some of them tend to zero while others go to infinity. The nonzero matrix Θ_k may also arise when regularized IP methods are applied to quadratic programming problems with no linear inequality constraints. In this case Θ_k is diagonal positive definite and its form depends on the dual regularization adopted (see, e.g. [23,26] and the references therein).

Typically, the solution of sequence (1) represents the major computational burden of the IP procedure. Therefore, if the linear systems are solved iteratively, the effectiveness and the efficiency of the preconditioners used affect the overall performance of the IP method. Here we address the case where the systems are solved by an iterative linear solver and constraint preconditioners (CPs) are employed (see, e.g. [10,11,15,19,20,24,28,31]).

Whenever the computational cost for building a CP for each system of the sequence is high, a preconditioner updating strategy can offer a tradeoff between cost and efficiency and can enhance the overall procedure for the solution of problem (1). Given a factorization of the preconditioner for a matrix of the sequence, such a strategy builds the preconditioners for some successive systems of the sequence by suitably modifying the available factorization to take into account the matrix changes. The aim is to form a preconditioner that is computationally cheaper than the one computed from scratch, though preserving efficiency in the solution of the linear systems.

A first proposal for an updating strategy for CP preconditioners has been presented in [5]. It relies on a low-rank modification of the factorized Schur complement of the (1,1) block, which has been designed by exploiting results from [1,33]. Clearly, to keep the computational cost low, only small-rank changes are allowed. In this paper, in order to partially recover the information discarded by the low-rank correction, we propose to perform a further update. We note that a part of the discarded information can be seen as a diagonal positive semidefinite modification of the matrix resulting from the low-rank modification. Then we compute an approximate Cholesky factorization of the diagonally modified matrix by updating the Cholesky factorization of the preconditioner resulting from the low-rank correction. This step is accomplished by using a procedure in the framework given in [4]. Theoretical and numerical results show that the latter procedure can improve the preconditioner updating strategy in [5].

The paper is organized as follows. In Sect. 2 we provide the basis for describing our preconditioner updating procedure along with some theoretical results which will be exploited to analyze it. In Sect. 3 we first recall the updating procedure based on the low-rank correction of the Schur complement and then we present the new updating step, aimed at recovering part of the information discarded by the low-rank approach. In Sect. 4 we show some numerical results illustrating the behavior of the overall updating approach. We give some conclusions in Sect. 5.

Henceforth we use the following notations. We denote by $\|\cdot\|$ the matrix 2-norm. For any symmetric matrix M, we denote by $\lambda(M)$ any eigenvalue of M, and by $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ the minimum and maximum of these eigenvalues; furthermore, we use $\mathrm{diag}(M)$ to indicate the diagonal matrix with the same diagonal entries as M. Finally, for any complex number z, we denote by $\mathcal{R}(z)$ and $\mathcal{I}(z)$ the real and imaginary parts of z.

2 Preliminaries

The updating strategy presented in this paper builds a preconditioner for a generic matrix of the sequence (1),

$$\mathcal{A}_k = \begin{bmatrix} B_k & A^T \\ A & -\Theta_k \end{bmatrix},$$

exploiting information from a preconditioner built for a previous matrix of the sequence, which is denoted by

$$\mathcal{A}_{seed} = \begin{bmatrix} B_{seed} & A^T \\ A & -\Theta_{seed} \end{bmatrix}. \tag{3}$$

We assume that a CP $\mathcal{P}_{seed}$ is available for $\mathcal{A}_{seed}$, having the following form:

$$\mathcal{P}_{seed} = \begin{bmatrix} H_{seed} & A^T \\ A & -\Theta_{seed} \end{bmatrix}, \tag{4}$$

where H_{seed} is the following approximation to B_{seed}:

$$H_{seed} = \text{diag}(B_{seed}).$$

The effectiveness of this preconditioner in the context of IP methods is widely recognized by the optimization community, see, e.g. [6,10,11,21,28].

In order to simplify the notations, henceforth we denote H_{seed} by H. The application of $\mathcal{P}_{seed}$ requires its factorization. We consider the following block LDL^T factorization:

$$\mathcal{P}_{seed} = \begin{bmatrix} I_n & 0 \\ AH^{-1} & I_m \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & -S_{seed} \end{bmatrix} \begin{bmatrix} I_n & H^{-1}A^T \\ 0 & I_m \end{bmatrix}, \tag{5}$$

where I_j is the identity matrix of dimension j and S_{seed} is the negative Schur complement of H in $\mathcal{P}_{seed}$, i.e.,

$$S_{seed} = AH^{-1}A^T + \Theta_{seed}. \tag{6}$$

We also assume that a Cholesky factorization of S_{seed} has been computed:

$$S_{seed} = L_{seed} D_{seed} L_{seed}^T, \tag{7}$$

where L_{seed} is unit lower triangular.

Now we consider a generic system of the sequence and we drop the iteration index k from $\mathcal{A}_k$, B_k and Θ_k in order to simplify the notation. Then, as with the previous definition of $\mathcal{P}_{seed}$, the CP for

$$\mathcal{A} = \begin{bmatrix} B & A^T \\ A & -\Theta \end{bmatrix}, \tag{8}$$

is given by

$$\mathcal{P}_{ex} = \begin{bmatrix} G & A^T \\ A & -\Theta \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ AG^{-1} & I_m \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & -S \end{bmatrix} \begin{bmatrix} I_n & G^{-1}A^T \\ 0 & I_m \end{bmatrix}, \tag{9}$$

where

$$G = \text{diag}(B), \quad S = AG^{-1}A^T + \Theta.$$

Concerning the eigenvalue distribution of $\mathcal{P}_{ex}^{-1}\mathcal{A}$, there exists an eigenvalue at 1 with multiplicity $2m - p$, with $p = \text{rank}(\Theta)$, and $n - m + p$ real positive eigenvalues such that the better G approximates B the more clustered around 1 they are [18]. On the other hand, computing a Cholesky factorization of S may account for a large part of the computational cost for solving the KKT system, and hence replacing S with a computationally cheaper approximation of it, S_{inex}, may be convenient [5,21,29,31].

The resulting *inexact* CP has the following form:

$$\mathcal{P}_{inex} = \begin{bmatrix} I_n & 0 \\ AG^{-1} & I_m \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & -S_{inex} \end{bmatrix} \begin{bmatrix} I_n & G^{-1}A^T \\ 0 & I_m \end{bmatrix}. \tag{10}$$

Approximations of CPs may be also obtained by replacing the constraint matrix A in (9) with a sparse approximations of it [9].

In our work we focus on $\mathcal{P}_{inex}$. The spectral analysis of $\mathcal{P}_{inex}^{-1}\mathcal{A}$ has been addressed in [7,8,32] and further refined in [5]. Here we report some results from [5], which will be exploited to design and analyse our updating procedure. We note that, although the convergence of Krylov solvers for systems with coefficient matrix $\mathcal{P}_{inex}^{-1}\mathcal{A}$ is not fully characterized by the spectrum of the matrix, in many practical cases it depends on the distribution of the eigenvalues. Therefore we are interested in providing bounds on the eigenvalues of $\mathcal{P}_{inex}^{-1}\mathcal{A}$. The bounds given in the next theorem highlight the dependence of the spectrum of $\mathcal{P}_{inex}^{-1}\mathcal{A}$ on that of $S_{inex}^{-1}S$ (see [5, Theorem 2.1]).

Theorem 1 *Let $\mathcal{A}$ and $\mathcal{P}_{inex}$ be the matrices in (8) and (10), and let λ and $[x^T, y^T]^T$ be an eigenvalue of $\mathcal{P}_{inex}^{-1}\mathcal{A}$ and a corresponding eigenvector. Let*

$$X = G^{-\frac{1}{2}}BG^{-\frac{1}{2}} \tag{11}$$

and suppose that $2I_n - X$ is positive definite. Let

$$\bar{\lambda} = \lambda_{\max}\left(S_{inex}^{-1}S\right) \max\left\{2 - \lambda_{\min}(X),\ 1\right\}, \tag{12}$$

$$\underline{\lambda} = \lambda_{\min}\left(S_{inex}^{-1}S\right) \min\left\{2 - \lambda_{\max}(X),\ 1\right\}. \tag{13}$$

Then, $\mathcal{P}_{inex}^{-1}\mathcal{A}$ has at most 2m eigenvalues with nonzero imaginary part, counting conjugates. Furthermore, if λ has nonzero imaginary part, then

$$\frac{1}{2}\left(\lambda_{\min}(X) + \underline{\lambda}\right) \le \mathcal{R}(\lambda) \le \frac{1}{2}\left(\lambda_{\max}(X) + \bar{\lambda}\right); \tag{14}$$

otherwise,

$$\min\left\{\lambda_{\min}(X),\ \underline{\lambda}\right\} \le \lambda \le \max\left\{\lambda_{\max}(X),\ \bar{\lambda}\right\}, \quad \textit{for } y \neq 0, \tag{15}$$

$$\lambda_{\min}(X) \le \lambda \le \lambda_{\max}(X), \quad \textit{for } y = 0. \tag{16}$$

Finally, the imaginary part of λ satisfies

$$|\mathcal{I}(\lambda)| \le \sqrt{\lambda_{\max}\left(S_{inex}^{-1} A G^{-1} A^T\right)}\, \|I_n - X\|. \tag{17}$$

We note that $2I_n - X$ can be made positive semidefinite by scaling X in (11) so that its eigenvalues are not greater than 2. Furthermore, if B is diagonal, then $X = I_n$ and all the eigenvalues of $\mathcal{P}_{inex}^{-1}\mathcal{A}$ are real. In this case $\mathcal{P}_{inex}^{-1}\mathcal{A}$ has at least n unit eigenvalues, with n associated independent eigenvectors of the form $[x^T, 0^T]^T$, and the remaining eigenvalues lie in the interval $[\lambda_{\min}(S_{inex}^{-1}S),\ \lambda_{\max}(S_{inex}^{-1}\, S)]$ (see also [21]).

Finally, from the analysis carried out in [32, Sect. 5] it follows that the number of unit eigenvalues of $\mathcal{P}_{inex}^{-1}\mathcal{A}$ depends also on the number of zero eigenvalues of $S - S_{inex}$. In particular, when $S - S_{inex}$ has l zero eigenvalues, if $S^{-1/2}AA^TS^{-1/2}$ has no unit eigenvalues then $\mathcal{P}_{inex}^{-1}\mathcal{A}$ has l unit eigenvalues with geometric multiplicity l.

3 The preconditioner updating strategy

Our preconditioner updating strategy is based on building S_{inex} by a suitable update of the seed Schur complement S_{seed}. In [5] we presented an updated preconditioner of the form (10), where S_{inex} is a low-rank modification, S_{lr}, of S_{seed}. Here we make a step further, improving the approximation of S provided by S_{lr}, in the case where Θ is a nonzero matrix. Therefore, our approach for building S_{inex} consists of two steps: the first computes S_{lr} through the procedure discussed in [5]; the second employs updating techniques discussed in [3,4] to form an approximate factorization of $S_{lr} + \Delta$, where Δ is a suitable diagonal positive semidefinite matrix containing information not included in the previous step. The latter factorization provides the matrix S_{inex}, henceforth denoted by S_{cu} because it is the result of the combination of two updates. The corresponding inexact preconditioner is

$$\mathcal{P}_{cu} = \begin{bmatrix} I_n & 0 \\ AG^{-1} & I_m \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & -S_{cu} \end{bmatrix} \begin{bmatrix} I_n & G^{-1}A^T \\ 0 & I_m \end{bmatrix}. \tag{18}$$

Next we provide a detailed description of our procedure and analyze the quality of the resulting preconditioner. First, we describe how to perform the low-rank update of

$\mathcal{P}_{seed}$ and summarize related theoretical results given in [5]. Second, we present the subsequent updating step and provide new bounds on the eigenvalues of the preconditioned matrix.

3.1 First step: building S_{lr}

Let $\mathcal{L} = \{i \, : \, \Theta_{ii} \neq 0\}$ and let m_1 be its cardinality. We recall that the set $\mathcal{L}$ does not change throughout the sequence. Furthermore, let $\widetilde{\Theta}_{seed}$ and $\widetilde{\Theta}$ be the $m_1 \times m_1$ diagonal submatrices containing the nonzero diagonal entries of Θ_{seed} and Θ, respectively, and let $\widetilde{I}_m$ be the rectangular matrix consisting of the columns of I_m with indices in $\mathcal{L}$. Then, the Schur complements S_{seed} and S, corresponding to $\mathcal{P}_{seed}$ and $\mathcal{P}_{ex}$, respectively, can be written as follows:

$$S_{seed} = \widetilde{A}\widetilde{H}^{-1}\widetilde{A}^T, \quad S = \widetilde{A}\widetilde{G}^{-1}\widetilde{A}^T,$$

where

$$\widetilde{A} = \begin{bmatrix} A & \widetilde{I}_m \end{bmatrix}, \quad \widetilde{H}^{-1} = \begin{bmatrix} H^{-1} & 0 \\ 0 & \widetilde{\Theta}_{seed} \end{bmatrix}, \quad \widetilde{G}^{-1} = \begin{bmatrix} G^{-1} & 0 \\ 0 & \widetilde{\Theta} \end{bmatrix}. \tag{19}$$

Trivially,

$$S = S_{seed} + \widetilde{A}\left(\widetilde{G}^{-1} - \widetilde{H}^{-1}\right)\widetilde{A}^T. \tag{20}$$

The updating procedure described in [5] consists in finding a low-rank correction of S_{seed} of the form

$$S_{lr} = S_{seed} + \widetilde{A}\widetilde{K}\widetilde{A}^T = \widetilde{A}\left(\widetilde{H}^{-1} + \widetilde{K}\right)\widetilde{A}^T = \widetilde{A}\widetilde{J}^{-1}\widetilde{A}^T, \tag{21}$$

where $\widetilde{K}$, and hence $\widetilde{J}$, is a suitable diagonal matrix. The matrix $\widetilde{J}$ (or, equivalently, $\widetilde{K}$) is chosen with a double goal: tightening the bounds on the eigenvalues provided by Theorem 1 and limiting the cost for computing the Cholesky factorization of S_{lr} as a modification of the Cholesky factorization (7). A key role in achieving the first goal is played by a result in [1], reported next for completeness.

Lemma 2 *Let $U \in \mathbb{R}^{r\times s}$ be full rank and let $E, F \in \mathbb{R}^{s\times s}$ be symmetric positive definite. Then*

$$\lambda_{\min}\left(E^{-1}F\right) \leq \lambda\left(\left(UEU^T\right)^{-1} UFU^T\right) \leq \lambda_{\max}\left(E^{-1}F\right).$$

For any diagonal positive definite matrix $C \in \mathbb{R}^{(n+m_1)\times(n+m_1)}$, let $\gamma(C)$ be the vector of dimension $n + m_1$ with entries $\gamma_i(C)$ equal to the diagonal entries of $C\widetilde{G}^{-1}$ sorted in nondecreasing order, i.e.,

$$\min_{1\leq i\leq n+m_1} \frac{C_{ii}}{\widetilde{G}_{ii}} = \gamma_1(C) \leq \gamma_2(C) \leq \cdots \leq \gamma_{n+m_1}(C) = \max_{1\leq i\leq n+m_1} \frac{C_{ii}}{\widetilde{G}_{ii}}. \tag{22}$$

Lemma 2 yields the bounds

$$\gamma_1(\widetilde{J}) \le \lambda\big(S_{lr}^{-1}S\big) \le \gamma_{n+m_1}(\widetilde{J}). \tag{23}$$

We now give some definitions useful to specify $\widetilde{J}$. Let $l = (l_1, l_2, \ldots, l_{n+m_1})^T$ be the vector of indices such that

$$\gamma_{l_i}(\widetilde{H}) = \frac{\widetilde{H}_{ii}}{\widetilde{G}_{ii}}.$$

Given two real constants $\mu_\gamma \ge 1$ and $\nu_\gamma \in (0, 1]$, and two nonnegative integers q_1 and q_2 such that $q = q_1 + q_2 \le n + m_1$, we define $\widetilde{\Gamma}$ as

$$\widetilde{\Gamma} = \widetilde{\Gamma}_1 \cup \widetilde{\Gamma}_2, \tag{24}$$

where

$$\begin{aligned} \widetilde{\Gamma}_1 &= \left\{i : n + m_1 - q_1 + 1 \le l_i \le n + m_1 \text{ and } \gamma_{l_i}(\widetilde{H}) > \mu_\gamma\right\}, \\ \widetilde{\Gamma}_2 &= \left\{i : 1 \le l_i \le q_2 \text{ and } \gamma_{l_i}(\widetilde{H}) < \nu_\gamma\right\}. \end{aligned} \tag{25}$$

Then we define the matrix $\widetilde{K}$ in (21) by setting

$$\widetilde{K}_{ii} = \begin{cases} \widetilde{G}_{ii}^{-1} - \widetilde{H}_{ii}^{-1} & \text{if } i \in \widetilde{\Gamma}, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

As a consequence, the diagonal entries of $\widetilde{J}$ take the following form:

$$\widetilde{J}_{ii} = \begin{cases} \widetilde{G}_{ii} & \text{if } i \in \widetilde{\Gamma}, \\ \widetilde{H}_{ii} & \text{otherwise.} \end{cases} \tag{27}$$

We also note that $\widetilde{J}^{-1}$ can be written as follows:

$$\widetilde{J}^{-1} = \begin{bmatrix} J^{-1} & 0 \\ 0 & \widetilde{\Theta}_{lr} \end{bmatrix}, \tag{28}$$

where $J \in \mathbb{R}^{n \times n}$ accounts for the changes from H to G, and $\widetilde{\Theta}_{lr} \in \mathbb{R}^{m_1 \times m_1}$ for those from $\widetilde{\Theta}_{seed}$ to $\widetilde{\Theta}$.

Suppose that the sets $\widetilde{\Gamma}_1$ and $\widetilde{\Gamma}_2$ in (25) have cardinality equal to $q_1^* \le q_1$ and $q_2^* \le q_2$, respectively. Then, from (21), $S_{lr} - S_{seed}$ is a correction of rank $q^* = q_1^* + q_2^*$ equal to the cardinality of $\widetilde{\Gamma}$. Once S_{lr} is defined, the low-rank update preconditioner $\mathcal{P}_{lr}$ is simply obtained by setting $S_{inex} = S_{lr}$ in (10), i.e.,

$$\mathcal{P}_{lr} = \begin{bmatrix} I_n & 0 \\ AG^{-1} & I_m \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & -S_{lr} \end{bmatrix} \begin{bmatrix} I_n & G^{-1}A^T \\ 0 & I_m \end{bmatrix}. \tag{29}$$

By combining Theorem 1 with (23) and (27), we get the following bounds on the eigenvalues of $\mathcal{P}_{lr}^{-1}\mathcal{A}$, (see [5, Corollary 3.3]).

Corollary 3 *Let $\mathcal{A}$, $\mathcal{P}_{lr}$ and X be the matrices in (8), (29) and (11), and let λ be an eigenvalue of $\mathcal{P}_{lr}^{-1}\mathcal{A}$. Let $\gamma(\widetilde{H})$ and $\gamma(\widetilde{J})$ be defined as in (22), and let q_1^* and q_2^* be the cardinality of the sets $\widetilde{\Gamma}_1$ and $\widetilde{\Gamma}_2$ in (25), respectively. If $2I_n - X$ is positive definite, then λ satisfies either (14) or (15)–(16) with*

$$\bar{\lambda} \le \gamma_{n+m_1}(\widetilde{J}) \max\left\{2 - \lambda_{\min}(X),\ 1\right\}, \tag{30}$$

$$\underline{\lambda} \ge \gamma_1(\widetilde{J}) \min\left\{2 - \lambda_{\max}(X),\ 1\right\}, \tag{31}$$

where

$$\begin{aligned} \gamma_1(\widetilde{J}) &= \min\left\{1, \min_{i \notin \widetilde{\Gamma}} \gamma_{l_i}(\widetilde{H})\right\} = \min\left\{1, \gamma_{q_2^*+1}(\widetilde{H})\right\}, \\ \gamma_{n+m_1}(\widetilde{J}) &= \max\left\{1, \max_{i \notin \widetilde{\Gamma}} \gamma_{l_i}(\widetilde{H})\right\} = \max\left\{1, \gamma_{n+m_1-q_1^*}(\widetilde{H})\right\}. \end{aligned} \tag{32}$$

Furthermore,

$$|\mathcal{I}(\lambda)| \le \sqrt{\gamma_{n+m_1}(\widetilde{J})}\, \|I_n - X\|. \tag{33}$$

It is clear that the bounds provided by the previous corollary are expected to improve as the set $\widetilde{\Gamma}$ is enlarged, i.e., q is increased. Likewise, smaller (larger) values of μ_γ (ν_γ) may generally improve these bounds. However, the choice of the previous values must take into account the cost for computing the Cholesky factorization of S_{lr} by updating the factorization available for S_{seed}. From (7), (21) and (26) it follows that

$$S_{lr} = L_{seed} D_{seed} L_{seed}^T + \bar{A}\bar{K}\bar{A}^T,$$

where $\bar{A} \in \mathbb{R}^{m \times q^*}$ consists of the columns of $\widetilde{A}$ with indices in $\widetilde{\Gamma}$, and $\bar{K} \in \mathbb{R}^{q^* \times q^*}$ is the diagonal matrix having on the diagonal the nonzero entries of $\widetilde{K}$ corresponding to those indices. Therefore, the Cholesky factorization

$$S_{lr} = L_{lr} D_{lr} L_{lr}^T, \tag{34}$$

is a rank-q^* modification of the factorization (7) and can be computed, e.g. through the update and downdate procedures in [17]. Note that an update is required if $\widetilde{H}_{ii} > \widetilde{G}_{ii}$ and a downdate is required if $\widetilde{H}_{ii} < \widetilde{G}_{ii}$.

The computational cost for building S_{lr} depends on the value of q, and for practical purposes the updating strategy is more convenient than computing the exact preconditioner as long as q is kept fairly small (see [5]). Similarly, values of μ_γ and ν_γ not too close to 1 are used in practice; thus, indices that do not bring significant improvement in the eigenvalue bounds while increasing the cost for the updating strategy (see again [5]) are excluded from $\widetilde{\Gamma}$. Since a large number of entries in $\widetilde{G} - \widetilde{H}$ must be discarded, we now propose a second step of our updating approach for recovering part of the discarded information.

3.2 Second step: updating S_{lr} to get S_{cu}

In order to recover information not included in S_{lr}, we observe that part of this information can be regarded as a positive semidefinite diagonal modification, Δ, of S_{lr}. Therefore, we can compute a low-cost approximate LDL^T factorization of the matrix $S_{lr}+\Delta$ by exploiting the procedures in [3,4,30] to update the factorization (34) of S_{lr}. Furthermore, by expressing $S_{lr}+\Delta$ as $\widetilde{A}\widetilde{J}_\Delta^{-1}\widetilde{A}^T$, with $\widetilde{J}_\Delta$ diagonal positive definite, we can exploit Lemma 2 to derive new eigenvalue bounds.

We consider the diagonal matrix $\widetilde{\Delta} \in \mathbb{R}^{m_1\times m_1}$ defined by

$$\widetilde{\Delta}_{ii} = \begin{cases} \widetilde{\Theta}_{ii} - (\widetilde{\Theta}_{seed})_{ii} & \text{if } i+n \notin \widetilde{\Gamma} \text{ and } \widetilde{\Theta}_{ii} - (\widetilde{\Theta}_{seed})_{ii} > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{35}$$

whose nonzero entries correspond to the ratios $\widetilde{\Theta}_{ii}/(\widetilde{\Theta}_{seed})_{ii}$ that are greater than 1 and have been discarded in the construction of S_{lr}. Then, recalling the definition of $\widetilde{I}_m$ in Sect. 3.1, we set $\Delta \in \mathbb{R}^{m\times m}$ as follows:

$$\Delta = \widetilde{I}_m\, \widetilde{\Delta}\, \widetilde{I}_m^T, \tag{36}$$

and define the matrix

$$S_\Delta = S_{lr} + \Delta. \tag{37}$$

Using S_Δ instead of S_{lr} generally improves the quality of the preconditioner $\mathcal{P}_{cu}$, as shown next. Let

$$\widetilde{\Gamma}_\Delta = \widetilde{\Gamma} \cup \left\{ j :\ j = i+n,\ \text{with } 1 \le i \le m_1 \text{ and } \gamma_{l_j}(\widetilde{H}) = \frac{\widetilde{\Theta}_{ii}}{(\widetilde{\Theta}_{seed})_{ii}} > 1 \right\},$$

and let $\widetilde{\Theta}_\Delta \in \mathbb{R}^{m_1\times m_1}$ the diagonal matrix defined by

$$(\widetilde{\Theta}_\Delta)_{ii} = \begin{cases} \widetilde{\Theta}_{ii} & \text{if } i+n \in \widetilde{\Gamma}_\Delta, \\ (\widetilde{\Theta}_{seed})_{ii} & \text{otherwise.} \end{cases}$$

Then we have

$$S_\Delta = \widetilde{A}\widetilde{J}_\Delta^{-1}\widetilde{A}^T, \quad \widetilde{J}_\Delta^{-1} = \begin{bmatrix} J^{-1} & \\ & \widetilde{\Theta}_\Delta \end{bmatrix},$$

where $\widetilde{A}$ and J are the matrices in (19) and (28), respectively. By reasoning as in Sect. 3.1, we get the following bounds on the eigenvalues of $S_\Delta^{-1}S$:

$$\hat{\gamma}_1 \le \lambda\left(S_\Delta^{-1}S\right) \le \hat{\gamma}_{n+m_1}, \tag{38}$$

where

$$\hat{\gamma}_1 = \gamma_1(\widetilde{J}) = \min\left\{1, \gamma_{q_2^*+1}(\widetilde{H})\right\}, \tag{39}$$

$$\hat{\gamma}_{n+m_1} = \max\left\{1, \max_{i \notin \widetilde{\Gamma}_\Delta} \gamma_{l_i}(\widetilde{H})\right\}. \tag{40}$$

Trivially, $\hat{\gamma}_{n+m_1} \leq \gamma_{n+m_1}(\widetilde{J})$, with $\gamma_{n+m_1}(\widetilde{J})$ given in (32), and the benefit of using S_Δ depends on the size of $\hat{\gamma}_{n+m_1}$ with respect to $\gamma_{n+m_1}(\widetilde{J})$.

On the other hand, using S_Δ in place of S_{lr} requires its factorization. In order to limit the computational cost of the preconditioner, we can compute an approximation to the Cholesky factorization of S_Δ,

$$S_{cu} = L_{cu} D_{cu} L_{cu}^T \simeq S_\Delta, \tag{41}$$

by using the updating procedures mentioned at the beginning of this section. The procedure considered here updates the matrices L_{lr} and D_{lr} in (34) as follows (see [4] for details). Let W be the diagonal matrix with diagonal entries defined by

$$W_{ii} = \frac{(D_{lr})_{ii}}{(D_{lr})_{ii} + \Delta_{ii}}, \quad i = 1, \ldots, m;$$

we define the matrices L_{cu} and D_{cu} as

$$\begin{aligned} &D_{cu} = D_{lr} + \Delta, \\ &(L_{cu})_{jj} = 1, \quad j = 1, \ldots, m, \\ &L_{cu}(j+1:m, j) = W_{jj} L_{lr}(j+1:m, j), \quad j = 1, \ldots, m-1, \end{aligned} \tag{42}$$

where, using the Matlab notation, $L_{lr}(j+1:n, j)$ denotes the strictly lower triangular part of the j-th column of L_{lr}. Note that the sparsity pattern of the factors of S_{lr} is preserved; furthermore, the cost of forming S_{cu} is low, since the cost of D_{cu} is negligible, and the computation of L_{cu} consists in scaling only the nonzero entries of the columns of L_{lr} corresponding to the nonzero entries of Δ. Obviously, if Δ is the zero matrix, the update is not performed and $S_{cu} = S_{lr}$, i.e., $\mathcal{P}_{cu} = \mathcal{P}_{lr}$. This limit case occurs if $\widetilde{\Theta}_{ii} \leq (\widetilde{\Theta}_{seed})_{ii}$ for $i = 1, \ldots, m_1$, or if all the indices $i + n$ associated with the ratios $\widetilde{\Theta}_{ii}/(\widetilde{\Theta}_{seed})_{ii}$ that are greater than 1 belong to $\widetilde{\Gamma}$.

We refer to [3,4] for theoretical results and computational experiences motivating the previous updating approach. Here we report only some results concerning the eigenvalues of $S_{cu}^{-1} S$ (see [4, Theorems 2.2 and 2.4]), which are useful in the spectral analysis of $\mathcal{P}_{cu}^{-1} \mathcal{A}$.

Theorem 4 *Let S_Δ and S_{cu} be the matrices in (37) and (41). For all $\varepsilon > 0$ there exists $\eta > 0$ such that if $\|\Delta\| < \eta$, then*

$$\left|\lambda(S_{cu}^{-1} S_\Delta) - 1\right| < \varepsilon$$

for all the eigenvalues of $S_{cu}^{-1} S_\Delta$. Furthermore, if $S_\Delta - S_{cu}$ has rank $m - l$, then l eigenvalues of $S_{cu}^{-1} S_\Delta$ are equal to 1.

Theorem 5 *Let S_Δ and S_{cu} be the matrices in (37) and (41). For all $\varepsilon > 0$ there exists $\eta > 0$ such that if $\Delta_{ii} > \eta$ for r diagonal entries of Δ, then*

$$\left|\lambda(S_{cu}^{-1} S_\Delta) - 1\right| < \varepsilon$$

for r eigenvalues of $S_{cu}^{-1} S_\Delta$.

The next theorem provides bounds on the eigenvalues of $\mathcal{P}_{cu}^{-1}\mathcal{A}$ in terms of $\hat{\gamma}_1$ and $\hat{\gamma}_{n+m_1}$ and of the minimum and maximum eigenvalues of $S_{cu}^{-1} S_\Delta$.

Theorem 6 *Let $\mathcal{A}$, $\mathcal{P}_{cu}$, S_Δ, S_{cu} and X be the matrices in (8), (18), (37), (41) and (11), and let λ be an eigenvalue of $\mathcal{P}_{cu}^{-1}\mathcal{A}$. Let $\hat{\gamma}_1$ and $\hat{\gamma}_{n+m_1}$ be the scalars in (39) and (40). If $2I_n - X$ is positive definite, then λ satisfies either (14) or (15)–(16) with*

$$\bar{\lambda} \le \hat{\gamma}_{n+m_1}\lambda_{\max}(S_{cu}^{-1} S_\Delta)\max\left\{2 - \lambda_{\min}(X),\ 1\right\}, \tag{43}$$

$$\underline{\lambda} \ge \hat{\gamma}_1\lambda_{\min}(S_{cu}^{-1} S_\Delta)\min\left\{2 - \lambda_{\max}(X),\ 1\right\}. \tag{44}$$

Furthermore,

$$|\mathcal{I}(\lambda)| \le \sqrt{\hat{\gamma}_{n+m_1}\lambda_{\max}(S_{cu}^{-1} S_\Delta)}\,\|I_n - X\|. \tag{45}$$

Proof The eigenvalue problem $S_{cu}^{-1} S w = \lambda w$ is equivalent to

$$S_\Delta^{-\frac{1}{2}} S\, S_\Delta^{-\frac{1}{2}} \bar{w} = \lambda\ S_\Delta^{-\frac{1}{2}}\ S_{cu}\ S_\Delta^{-\frac{1}{2}}\ \bar{w},$$

with $\bar{w} = S_\Delta^{\frac{1}{2}}\, w$. Hence, for any eigenvalue of $S_{cu}^{-1} S$ and any corresponding eigenvector w we have

$$\lambda\left(S_{cu}^{-1} S\right) = \frac{\bar{w}^T S_\Delta^{-\frac{1}{2}} S\, S_\Delta^{-\frac{1}{2}} \bar{w}}{\bar{w}^T S_\Delta^{-\frac{1}{2}}\ S_{cu}\ S_\Delta^{-\frac{1}{2}}\ \bar{w}}.$$

Then, by exploiting (38) we get

$$\lambda\left(S_{cu}^{-1} S\right) \le \frac{\lambda_{\max}\left(S_\Delta^{-1} S\right)}{\lambda_{\min}\left(S_\Delta^{-1} S_{cu}\right)} \le \hat{\gamma}_{n+m_1}\,\lambda_{\max}\left(S_{cu}^{-1} S_\Delta\right)$$

and

$$\lambda\left(S_{cu}^{-1} S\right) \ge \frac{\lambda_{\min}\left(S_\Delta^{-1}\, S\right)}{\lambda_{\max}\left(S_\Delta^{-1} S_{cu}\right)} \ge \hat{\gamma}_1\lambda_{\min}\left(S_{cu}^{-1} S_\Delta\right).$$

Inequalities (43) and (44) follow by recalling (12) and (13).

It remains to prove (45). The eigenvalue problem $S_{cu}^{-1} A\, G^{-1} A^T w = \lambda\, w$ is equivalent to

$$S_{\Delta}^{-\frac{1}{2}} A\, G^{-1} A^T S_{\Delta}^{-\frac{1}{2}} \bar{w} = \lambda S_{\Delta}^{-\frac{1}{2}} S_{cu} S_{\Delta}^{-\frac{1}{2}} \bar{w}.$$

Furthermore, since Θ is positive definite, for any vector $v \in \mathbb{R}^m$

$$v^T S_{\Delta}^{-\frac{1}{2}} A\, G^{-1} A^T S_{\Delta}^{-\frac{1}{2}} v \le v^T S_{\Delta}^{-\frac{1}{2}} \left(A\, G^{-1} A^T + \Theta\right) S_{\Delta}^{-\frac{1}{2}} v,$$

and by matrix similarity

$$\lambda_{\max}\left(S_{\Delta}^{-1} A\, G^{-1} A^T\right) \le \lambda_{\max}\left(S_{\Delta}^{-1} S\right).$$

By reasoning as in the first part of the proof and exploiting the previous inequality we get

$$\begin{aligned} \lambda\left(S_{cu}^{-1} A G^{-1} A^T\right) &\le \frac{\lambda_{\max}\left(S_{\Delta}^{-1} S\right)}{\lambda_{\min}\left(S_{\Delta}^{-1} S_{cu}\right)} \\ &= \hat{\gamma}_{n+m_1} \lambda_{\max}\left(S_{cu}^{-1} S_{\Delta}\right), \end{aligned}$$

and (45) follows from (17). □

The following result is a straightforward consequence of Theorem 6.

Corollary 7 *Let $\mathcal{A}$, $\mathcal{P}_{cu}$, S_{Δ}, S_{cu} and X be the matrices in (8), (18), (37), (41) and (11), and let λ be an eigenvalue of $\mathcal{P}_{cu}^{-1}\mathcal{A}$. Let $\hat{\gamma}_1$ and $\hat{\gamma}_{n+m_1}$ be the scalars in (39) and (40). If $2I_n - X$ is positive definite and, for all the eigenvalues of $S_{cu}^{-1} S_{\Delta}$,*

$$\left|\lambda(S_{cu}^{-1} S_{\Delta}) - 1\right| \le \varepsilon$$

for some $\varepsilon > 0$, then λ satisfies either (14) or (15)–(16) with

$$\begin{aligned} \bar{\lambda} &\le (1+\varepsilon)\hat{\gamma}_{n+m_1} \max\left\{2 - \lambda_{\min}(X),\ 1\right\}, \\ \underline{\lambda} &\ge (1-\varepsilon)\hat{\gamma}_1 \min\left\{2 - \lambda_{\max}(X),\ 1\right\}. \end{aligned}$$

Furthermore,

$$|\mathcal{I}(\lambda)| \le \sqrt{(1+\varepsilon)\hat{\gamma}_{n+m_1}}\, \|I_n - X\|.$$

The previous results show that the bounds (30) and (33) can be effectively improved if $\lambda_{\max}(S_{cu}^{-1} S_{\Delta})$ is not too far from 1. Furthermore, $\lambda_{\min}(S_{cu}^{-1} S_{\Delta})$ must not be too far from 1, in order to avoid a significant deterioration of the bound (31). By Theorems 4 and 5, this happens when the entries of Δ are either all sufficiently small or all sufficiently large. In general, we cannot expect that all the eigenvalues of $S_{cu}^{-1} S_{\Delta}$ are close to 1; nevertheless, the ability of S_{cu} of clustering around 1 some eigenvalues of S_{Δ} when Δ has large entries, provides a way to tighten the bounds on the spectrum of $\mathcal{P}_{cu}^{-1}\mathcal{A}$, as confirmed by numerical experiments.

We conclude this section by summarizing in Algorithm 1 the main steps of the overall updating procedure.

```
Algorithm 1: updating the constraint preconditioner
Given $\mu_\gamma \geq 1$, $\nu_\gamma \in (0, 1]$ and the nonnegative integers $q_1$ and $q_2$,
1. form $\widetilde{H}$ defined in (19) and compute $\gamma_i(\widetilde{H})$, $i = 1, \ldots, n + m_1$, according to (22);
2. set $\widetilde{\Gamma}$ as in (24), $q^*$ as the cardinality of $\widetilde{\Gamma}$, and $\widetilde{J}$ as in (27);
3. compute the factorization in (34), $S_{lr} = L_{lr} D_{lr} L_{lr}^T$, by a rank-$q^*$ correction of $S_{seed}$ in (7);
4. compute the matrix $\Delta$ given in (36);
5. if $\Delta \neq 0$
       compute the approximate factorization $L_{cu} D_{cu} L_{cu}^T$ of $L_{lr} D_{lr} L_{lr}^T + \Delta$ (see (42));
   else
       set $L_{cu} = L_{lr}$, $D_{cu} = D_{lr}$;
6. set
       $\mathcal{P}_{cu} = \begin{bmatrix} I_n & 0 \\ AG^{-1} & L_{cu} \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & -D_{cu} \end{bmatrix} \begin{bmatrix} I_n & G^{-1}A^T \\ 0 & L_{cu}^T \end{bmatrix}.$
```

4 Numerical results

We provide some illustrative examples of the behavior of the preconditioner $\mathcal{P}_{cu}$ built with Algorithm 1, compared with the exact CP preconditioner $\mathcal{P}_{ex}$ in (9) and the updated preconditioner $\mathcal{P}_{lr}$ in (29).

We consider five sequences of KKT systems that arise in the solution, by an IP method, of convex quadratic programming problems with linear inequality constraints. These problems have been obtained by modifying the CUTEst [27] problems CVXQP1, CVXQP3 and MOSARQP1. The modifications have been made because large or variable-size CUTEst convex quadratic programming problems with inequality constraints have Schur complements that are very inexpensive to factorize and thus are useless for our experiments. More precisely, CVXQP1 and CVXQP3, which have non-banded Schur complements, have been modified by changing their original equality constraint $Ax = b$ into $Ax \geq b$ (the modified problems have been identified by appending "-M" to the original names). Furthermore, in order to increase the density of their Schur complements, four nonzero entries per row have been added to the matrix A of CVXQP1-M, and one nonzero entry per row to the matrix A of CVXQP3-M (the denser problems have been identified by appending "-D" and "-D2" to their names, respectively, according to the notation used in [5]). Finally, nonzeros in the positions (i, n), with $mod(i, 10) = 1$, have been added to the matrix A of MOSARQP1, obtaining problem MOSARQP1-D (note that the original problem has a narrow-banded Schur complement). The dimensions n and m of the five problems and the number $nnz(S)$ of nonzero entries of their Schur complements are reported in the second column of Table 1.

The sequences of KKT systems have been obtained by running the Fortran 90 PRQP code, which implements an infeasible inexact potential reduction IP method [11,13, 15], and extracting the KKT matrices arising at each IP iteration and the corresponding right-hand sides. Afterwards these sequences have been solved offline, applying $\mathcal{P}_{ex}$, $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$. The starting point for the IP procedure in PRQP has been built with the STP2 strategy described in [16] and the tolerances on the relative duality gap and the relative infeasibilities have been set to 10^{-6} and 10^{-7}, respectively. Within PRQP the

Table 1 Comparison of $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$: total number of SQMR iterations, refreshes of the preconditioners and execution times

Problem	n	*IPits*	$\mathcal{P}_{lr}$ $(q = 50)$					$\mathcal{P}_{cu}$				
	m $nnz(S)$		*nit*	*nref*	T_{prec}	T_{solve}	T_{tot}	*nit*	*nref*	T_{prec}	T_{solve}	T_{tot}
CVXQP1-M	20,000 10,000 67,976	11–26	920	5	18.364	44.892	63.256	597	5	19.191	29.828	49.019
CVXQP1-M-D	20,000 10,000 240,494	10–28	968	9	41.180	116.079	157.259	750	6	40.253	91.284	131.537
CVXQP3-M	20,000 15,000 155,942	11–30	1066	5	102.734	220.767	323.501	943	5	108.163	195.831	303.994
CVXQP3-M-D2	20,000 15,000 224,396	11–25	632	6	73.632	170.147	243.779	583	5	70.172	155.271	225.443
MOSARQP1-D	22,500 20,000 4259,250	1–24	318	7	31.182	53.708	84.890	227	7	37.985	39.113	77.098

KKT systems have been solved by the SQMR method coupled with the exact CP, using an adaptive tolerance in the stopping criterion, which relates the accuracy in the solution of the KKT system to the quality of the current IP iterate, in the spirit of inexact IP methods [2,12]. Specifically, in PRQP the SQMR iterations were stopped as soon as the norm of the unpreconditioned residual was below a tolerance of the form $\tau = \min\{\max\{\tau_1, 10^{-8}\}, 10^{-2}\|r_0\|\}$, where τ_1 depends on the duality gap value at the current IP iteration and r_0 is the initial unpreconditioned residual (see [12] for more details). The values of τ corresponding to all the systems of each sequence were saved to be used in our experiments.

The numerical experiments have been performed in the Matlab environment, exploiting C code for the construction of $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$. More precisely, the CHOLMOD package [14] has been called, through its MEX interface, to compute the sparse LDL^T factorization of S_{seed} and the low-rank updates and downdates required to build $\mathcal{P}_{lr}$. The updates (42) needed to form $\mathcal{P}_{cu}$ have been implemented in C as Matlab MEX-files too. The preconditioners have been built using $q_1 = q_2 = q/2 = 25$, in order to keep low the overhead of the updating/downdating phase; furthermore, $\mu_\gamma = 10$ and $\nu_\gamma = 0.1$ have been used to select the indices to be put in $\widetilde{\Gamma}$. These choices agree with the results of the experiments with $\mathcal{P}_{lr}$ discussed in [5]. When less than q_1 ratios $\gamma_{l_i}(\widetilde{H}) \geq \mu_\gamma$ (or less than q_2 ratios $\gamma_{l_i}(\widetilde{H}) \leq \nu_\gamma$) were available, the number of ratios $\gamma_{l_i}(\widetilde{H}) \leq \nu_\gamma$ (or $\gamma_{l_i}(\widetilde{H}) \geq \mu_\gamma$) was increased to have a total number of ratios in $\widetilde{\Gamma}$ as close as possible to q. We note that we have not applied any scaling to the matrix X in (11) to ensure positive definiteness of $2I_n - X$, although this is assumed in our theory. Nevertheless, the results generally appear to be in agreement with the theory (see also [5]). The linear systems have been solved using a Matlab implementation of the SQMR method without look-ahead [22], stopping the iterations when the norm of the residual had become smaller than the associated tolerance, as in the solution of the KKT systems within the IP code. A maximum number of 500 iterations has been considered too.

Following [5], the preconditioners have been "refreshed" as explained next. When, for a system of the sequence, the time for computing $\mathcal{P}_{lr}$ or $\mathcal{P}_{cu}$ and solving the preconditioned linear system exceeded 90 % of the time for building the last exact preconditioner and solving the corresponding system, the next system of the sequence was solved using the exact CP in (9). Furthermore, a maximum number of consecutive updates, $k_{\max}$, was also considered; here $k_{\max} = 4$. This strategy aims at avoiding deterioration of the quality of the updated preconditioner and hence excessive increase of the number of SQMR iterations, which may make the updating strategy useless. We note that, by using a refreshing criterion based on the execution time, both the problem and the computing environment are taken into account to get an efficient preconditioning strategy.

The tests have been performed on a six-core Xeon processor with clock frequency of 2.4 GHz, 24 GB of RAM and 12 MB of cache memory, running Ubuntu/Linux 12.04.5 (kernel version 3.2.0_83_generic). CHOLMOD and the C code for the updates (42) have been compiled with gcc 4.6.3, and Matlab R2015a (v. 8.5, 64-bit) has been employed. The `tic` and `toc` Matlab commands have been used to measure the execution times.

Table 2 CVXQP1-M: SQMR iterations and execution times obtained with $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$

IP#	$\mathcal{P}_{lr}$ $(q = 50)$					$\mathcal{P}_{cu}$				
	nit	q^*	T_{prec}	T_{solve}	T_{sum}	*nit*	$q^*/nnz(\Delta)$	T_{prec}	T_{solve}	T_{sum}
11	**9**		**2.701**	**0.504**	**3.205**	**9**		**2.701**	**0.504**	**3.205**
12	13	1	0.102	0.685	0.787	10	1/1696	0.176	0.548	0.724
13	19	14	0.339	0.963	1.302	14	14/1841	0.412	0.744	1.156
14	37	50	0.854	1.848	2.702	27	50/1949	0.928	1.366	2.294
15	106	50	0.752	5.154	5.906	46	50/2084	0.827	2.295	3.122
16	**16**		**2.624**	**0.839**	**3.463**	**16**		**2.624**	**0.839**	**3.463**
17	22	5	0.116	1.102	1.218	20	5/2495	0.191	1.023	1.214
18	41	50	0.647	2.017	2.664	36	50/2486	0.722	1.813	2.535
19	89	50	0.638	4.375	5.013	57	50/2491	0.713	2.833	3.546
20	**20**		**2.681**	**1.029**	**3.710**	**20**		**2.681**	**1.029**	**3.710**
21	37	4	0.201	1.825	2.026	28	4/2537	0.278	1.489	1.767
22	297	50	0.664	14.155	14.819	114	50/2517	0.739	5.512	6.251
23	**37**		**2.629**	**1.874**	**4.503**	**37**		**2.629**	**1.874**	**4.503**
24	86	14	0.251	4.048	4.299	47	14/2537	0.328	2.349	2.677
25	**37**		**2.604**	**1.855**	**4.459**	79	50/2514	0.667	3.756	4.423
26	54	47	0.561	2.619	3.180	**37**		**2.575**	**1.854**	**4.429**
	920		18.364	44.892	63.256	597		19.191	29.828	49.019

The data concerning the application of the exact CP are in bold. The number q^* of low-rank updates and the number $nnz(\Delta)$ of elements used for building $\mathcal{P}_{cu}$ are also reported. The data in the last row are the sums of the corresponding columns

We start the description of our numerical results comparing $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$. For the problems from the CVXQP family, we observe that the the number of elements used for the second step of the update, namely $nnz(\Delta)$ with Δ defined in (35) and (36), is zero or negligible (at most twenty) in the first systems of the sequence (first nine systems for CVXQP1-M-D, and first ten systems for CVXQP1-M, CVXQP3-M and CVXQP3-M-D2). Since no significant benefit can be obtained from the update strategy (42) with such a small number of elements, we exclude these first systems from the following analysis.

The results obtained with $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$ are shown in Table 1. We report the range *IPits* of IP iterations considered for each sequence and, for each preconditioner, the total number *nit* of SQMR iterations performed, the number *nref* of times the preconditioner has been refreshed, the total time T_{prec} for building the preconditioner, the total time T_{solve} for solving the preconditioned system, and the sum T_{tot} of the two times. The times are expressed in seconds. We see that all the runs with $\mathcal{P}_{cu}$ are faster than those with $\mathcal{P}_{lr}$. The gain, in terms of total execution time, varies between 6 and 23 %. Furthermore, for two sequences, the savings obtained in the number of SQMR iterations reduce the number of preconditioner refreshes with respect to the use of $\mathcal{P}_{lr}$.

Details on the solution of the sequences of KKT systems by using $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$ are shown in Tables 2, 3 and 4 for CVXQP1-M, CVXQP3-M-D2 and MOSARQP1-D, respectively. The IP iteration number corresponding to each system of a sequence is

Table 3 CVXQP3-M-D2: SQMR iterations and execution times obtained with $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$

IP#	$\mathcal{P}_{lr}$ ($q = 50$)					$\mathcal{P}_{cu}$				
	nit	q^*	T_{prec}	T_{solve}	T_{sum}	*nit*	$q^*/nnz(\Delta)$	T_{prec}	T_{solve}	T_{sum}
11	**9**		**9.579**	**2.586**	**12.165**	**9**		**9.579**	**2.586**	**12.165**
12	20	6	0.659	5.326	5.985	12	6/2227	1.104	3.340	4.444
13	33	50	2.718	9.070	11.788	19	50/2752	3.181	5.359	8.540
14	**15**		**9.564**	**4.113**	**13.677**	41	50/3282	3.685	10.988	14.673
15	22	1	0.570	6.003	6.573	**17**		**9.536**	**4.470**	**14.006**
16	47	50	2.731	12.458	15.189	28	2/4897	0.879	7.514	8.393
17	**24**		**9.442**	**6.308**	**15.750**	51	5/4970	3.111	14.142	17.253
18	40	7	0.458	10.252	10.710	**28**		**9.550**	**7.262**	**16.812**
19	68	50	2.779	18.468	21.247	43	3/5152	0.943	11.667	12.610
20	**32**		**9.619**	**8.177**	**17.796**	58	50/5156	3.617	16.433	20.050
21	46	2	0.538	12.504	13.042	**36**		**9.595**	**9.719**	**19.314**
22	83	50	3.934	23.209	27.143	56	4/5210	1.199	13.673	14.872
23	**41**		**9.491**	**11.317**	**20.808**	61	30/5204	3.401	14.722	18.123
24	82	18	1.972	21.750	23.722	**43**		**9.772**	**11.509**	**21.281**
25	**70**		**9.578**	**18.606**	**28.184**	81	1/5207	1.020	21.887	22.907
	632		73.632	170.147	243.779	583		70.172	155.271	225.443

The data concerning the application of the exact CP are in bold. The number q^* of low-rank updates and the number $nnz(\Delta)$ of elements used for building $\mathcal{P}_{cu}$ are also reported. The data in the last row are the sums of the corresponding columns

indicated by *IP#*. For each system and for both preconditioners, we report the number *nit* of SQMR iterations, the time T_{prec} for building the preconditioner for that system, the time T_{solve} for solving the preconditioned system, and $T_{sum} = T_{prec} + T_{solve}$. For $\mathcal{P}_{lr}$ we also display the size q^* of the low-rank update performed, i.e., the cardinality of the set $\widetilde{\Gamma}$ in (24), while for $\mathcal{P}_{cu}$ we display both q^* and the number of elements used for the second step of the update, i.e., $nnz(\Delta)$. The data corresponding to refreshes are in bold and the times are expressed in seconds.

We observe that, in general, forming $\mathcal{P}_{cu}$ is inexpensive although $nnz(\Delta)$ is large. Using $\mathcal{P}_{cu}$ may significantly reduce the number of SQMR iterations, especially in the last runs before a refresh. In fact, the performance of both $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$ tends to deteriorate progressively after a refresh, but the use of the diagonal modification in $\mathcal{P}_{cu}$ may considerably improve the effectiveness of $\mathcal{P}_{cu}$ with respect to $\mathcal{P}_{lr}$. This behavior is clearly illustrated, e.g. by the data of the 15th, 19th and 22nd system of CVXQP1-M (Table 2); a similar behavior can be recognized in Tables 3 and 4. Taking into account this numerical evidence and the fact that forming $\mathcal{P}_{cu}$ is simple and inexpensive, the second phase of the update appears worthy to be coupled with the low-rank correction in the update of CP preconditioners for KKT systems.

To provide further insight into the behavior of $\mathcal{P}_{cu}$, in Table 5 we report the maximum and minimum eigenvalues of $S_{lr}^{-1}S$ and $S_{cu}^{-1}S$ from the 12th to the 25th iteration of CVXQP1-M. Note that at the 25th iteration S_{lr} is the matrix which S_{cu} is built from, and not the matrix associated with the preconditioner $\mathcal{P}_{lr}$ considered in Table 2 (with

Table 4 MOSARQP1-D: SQMR iterations and execution times obtained with $\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$

IP#	$\mathcal{P}_{lr}$ $(q = 50)$					$\mathcal{P}_{cu}$				
	nit	q^*	T_{prec}	T_{solve}	T_{sum}	*nit*	$q^*/nnz(\Delta)$	T_{prec}	T_{solve}	T_{sum}
1	**1**		**2.709**	**0.298**	**3.007**	**1**		**2.709**	**0.298**	**3.007**
2	3	1	0.428	0.601	1.029	4	1/2068	0.645	0.741	1.386
3	7	2	0.424	1.278	1.702	5	2/2071	0.641	0.978	1.619
4	13	50	0.480	2.228	2.708	8	50/2076	0.680	1.311	1.991
5	**2**		**2.878**	**0.462**	**3.340**	10	50/2160	0.646	1.709	2.355
6	10	2	0.220	1.720	1.940	**3**		**2.878**	**0.593**	**3.471**
7	11	50	0.427	1.889	2.316	4	1/12158	0.664	0.724	1.388
8	28	50	1.392	4.570	5.962	6	50/11075	2.820	1.110	3.930
9	**4**		**2.769**	**0.779**	**3.548**	**4**		**2.769**	**0.779**	**3.548**
10	8	1	0.427	1.391	1.818	6	1/9866	0.638	1.090	1.728
11	11	2	0.425	1.890	2.315	11	2/9250	0.425	1.890	2.315
12	20	4	0.423	3.297	3.720	11	4/8675	0.655	1.902	2.557
13	**4**		**2.772**	**0.794**	**3.566**	14	50/8412	2.107	2.386	4.493
14	5	1	0.220	0.934	1.154	**4**		**2.824**	**0.796**	**3.620**
15	9	2	0.219	1.545	1.764	7	1/10130	0.463	1.252	1.715
16	17	2	0.218	2.819	3.037	11	2/10129	0.437	1.874	2.311
17	43	50	1.971	6.695	8.666	21	50/10104	2.212	3.465	5.677
18	**5**		**2.769**	**0.954**	**3.723**	**5**		**2.769**	**0.954**	**3.723**
19	15	1	0.425	2.591	3.016	13	1/10130	0.658	2.234	2.892
20	47	50	2.997	7.759	10.756	31	50/10130	3.293	5.050	8.343
21	**6**		**2.812**	**1.095**	**3.907**	**6**		**2.812**	**1.095**	**3.907**
22	21	7	0.589	3.430	4.019	17	7/10130	0.818	2.696	3.514
23	**6**		**2.759**	**1.108**	**3.867**	**6**		**2.759**	**1.108**	**3.867**
24	22	10	0.429	3.581	4.010	19	10/10130	0.663	3.078	3.741
	318		31.182	53.708	84.890	227		37.985	39.113	77.098

The data concerning the application of the exact CP are in bold. The number q^* of low-rank updates and the number $nnz(\Delta)$ of elements used for building $\mathcal{P}_{cu}$ are also reported. The data in the last row are the sums of the corresponding columns

$\mathcal{P}_{lr}$ and $\mathcal{P}_{cu}$ the refresh occurs at different IP iterations). We see that the updating strategy used to build S_{cu} practically does not change the smallest eigenvalue, while it reduces the largest one, according to the decrease observed in the number of iterations (see Table 2).

We conclude this section by comparing the preconditioners $\mathcal{P}_{cu}$ and $\mathcal{P}_{ex}$. In Table 6 we summarize the total number of SQMR iterations and execution times for $\mathcal{P}_{ex}$; the same data for $\mathcal{P}_{cu}$, available in Table 1, are repeated for the sake of readability. The reduction of the total time obtained with $\mathcal{P}_{cu}$ over $\mathcal{P}_{ex}$ varies between 9 and 25 % although, as expected, the number of SQMR iterations performed is larger than with $\mathcal{P}_{ex}$. It is noteworthy that $\mathcal{P}_{cu}$ speeds up the solution of the sequences associated with CVXQP1-M, CVXQP1-M-D and MOSARQP1, where the use of $\mathcal{P}_{lr}$ is not beneficial with respect to $\mathcal{P}_{ex}$. We also report briefly on experiments with inexact CPs obtained by

Table 5 CVXQP1-M: minimum and maximum eigenvalues of $S_{lr}^{-1}S$ and $S_{cu}^{-1}S$

IP#	$\lambda_{\min}\left(S_{lr}^{-1}S\right)$	$\lambda_{\max}\left(S_{lr}^{-1}S\right)$	$\lambda_{\min}\left(S_{cu}^{-1}S\right)$	$\lambda_{\max}\left(S_{cu}^{-1}S\right)$
12	4.15e−1	2.34e+0	4.16e−1	1.39e+0
13	2.02e−1	6.79e+0	2.03e−1	2.07e+0
14	8.63e−2	8.83e+0	8.67e−2	2.64e+0
15	3.77e−2	2.42e+1	3.81e−2	4.27e+0
17	2.85e−1	4.75e+0	2.85e−1	2.88e+0
18	1.35e−1	9.56e+0	1.35e−1	5.52e+0
19	6.47e−2	2.33e+1	6.47e−2	9.84e+0
21	1.97e−1	8.69e+0	1.97e−1	3.59e+0
22	2.72e−2	4.70e+1	2.72e−2	7.99e+0
24	1.02e−1	9.58e+0	1.02e−1	2.96e+0
25	1.44e−2	6.23e+1	1.44e−2	1.13e+1

Table 6 Comparison of $\mathcal{P}_{ex}$ and $\mathcal{P}_{cu}$: total number of SQMR iterations, refreshes for $\mathcal{P}_{cu}$ and execution times

Problem	n	*IPits*	$\mathcal{P}_{ex}$				$\mathcal{P}_{cu}$				
	m $nnz(S)$		*nit*	T_{prec}	T_{solve}	T_{tot}	*nit*	*nref*	T_{prec}	T_{solve}	T_{tot}
CVXQP1-M	20,000	11–26	351	42.035	18.274	60.309	597	5	19.191	29.828	49.019
	10,000										
	67,976										
CVXQP1 -M-D	20,000	10–28	670	63.616	82.000	145.616	750	6	40.253	91.284	131.537
	10,000										
	240,494										
CVXQP3-M	20,000	11–30	520	291.891	113.332	405.223	943	5	108.163	195.831	303.994
	15,000										
	155, 942										
CVXQP3-M-D2	20,000	11–25	424	143.798	113.594	257.392	583	5	70.172	155.271	225.443
	15,000										
	224,396										
MOSARQP1-D	22,500	1–24	92	66.854	18.119	84.973	227	7	37.985	39.113	77.098
	20,000										
	4,259,250										

approximating the exact Schur complement S with an inexact Cholesky factorization of it. These experiments have been performed to further assess the reliability of our updating approach. Unmodified and modified incomplete Cholesky factorizations of S have been computed using the Matlab function `ichol` and drop tolerances ranging from 10^{-1} to 10^{-5}. However, the resulting inexact CPs seem to lack robustness on our sequences, as SQMR failed in the solution of at least one system of each sequence.

5 Conclusions

We have presented a procedure for updating CPs for sequences of regularized KKT systems with the nonzero (2,2) block having a fixed sparsity pattern. This procedure combines a recently proposed technique for updating a block LDL^T factorization of a given seed CP, relying on a low-rank correction of its Schur complement, with a further update of the Schur complement. The latter update is performed to introduce into the preconditioner some information that has been discarded in the low-rank correction. It is based on a low-cost technique for approximating the Cholesky factorization when the matrix undergoes a diagonal positive semidefinite modification.

Theoretical results show that the procedure proposed here provides the possibility of tightening the bounds on the eigenvalues of the preconditioned matrix with respect to the procedure based on the low-rank correction alone, and that its effectiveness depends on the accuracy of the approximate Cholesky factorization computed in the second updating step. Numerical experiments confirm this behavior and show that the new approach can enhance the low-rank strategy.

Acknowledgments We are grateful to William Hager for helping us to understand some details in the implementation of the sparse Cholesky factorization available from the CHOLMOD package. We also appreciate the anonymous referees for their insightful comments, which led us to improve our work.

References

1. Baryamureeba, V., Steihaug, T., Zhang, Y.: Properties of a class of preconditioners for weighted least squares problems. Technical Report No. 170, Department of Informatics, University of Bergen, and Technical Report No. TR99-16, Department of Computational and Applied Mathematics. Rice University, Houston (1999)
2. Bellavia, S.: Inexact interior-point method. J. Optim. Theory Appl. **96**, 109–121 (1998)
3. Bellavia, S., De Simone, V., di Serafino, D., Morini, B.: Efficient preconditioner updates for shifted linear systems. SIAM J. Sci. Comput. **33**, 1785–1809 (2011)
4. Bellavia, S., De Simone, V., di Serafino, D., Morini, B.: A preconditioning framework for sequences of diagonally modified linear systems arising in optimization. SIAM J. Numer. Anal. **50**, 3280–3302 (2012)
5. Bellavia, S., De Simone, V., di Serafino, D., Morini, B.: Updating constraint preconditioners for KKT systems in quadratic programming via low-rank corrections. SIAM J. Optim. **25**, 1787–1808 (2015)
6. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
7. Benzi, M., Simoncini, V.: On the eigenvalues of a class of saddle point matrices. Numer. Math. **103**, 173–196 (2006)
8. Bergamaschi, L.: Eigenvalue distribution of constraint-preconditioned symmetric saddle point matrices. Numer. Linear Algebra Appl. **19**, 754–772 (2012)
9. Bergamaschi, L., Gondzio, J., Venturin, M., Zilli, G.: Inexact constraint preconditioners for linear systems arising in interior point methods. Comput. Optim. Appl. **36**, 137–147 (2007)
10. Bergamaschi, L., Gondzio, J., Zilli, G.: Preconditioning indefinite systems in interior point methods for optimization. Comput. Optim. Appl. **28**, 149–171 (2004)
11. Cafieri, S., D'Apuzzo, M., De Simone, V., di Serafino, D.: On the iterative solution of KKT systems in potential reduction software for large-scale quadratic problems. Comput. Optim. Appl. **38**, 27–45 (2007)
12. Cafieri, S., D'Apuzzo, M., De Simone, V., di Serafino, D.: Stopping criteria for inner iterations in inexact potential reduction methods: a computational study. Comput. Optim. Appl. **36**, 165–193 (2007)

13. Cafieri, S., D'Apuzzo, M., De Simone, V., di Serafino, D., Toraldo, G.: Convergence analysis of an inexact potential reduction method for convex quadratic programming. J. Optim. Theory Appl. **135**, 355–366 (2007)
14. Chen, Y., Davis, T.A., Hager, W.W., Rajamanickam, S.: Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. ACM Trans. Math. Softw. **35**, 22:1–22:14 (2008)
15. D'Apuzzo, M., De Simone, V., di Serafino, D.: On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods. Comput. Optim. Appl. **45**, 283–310 (2010)
16. D'Apuzzo, M., De Simone, V., di Serafino, D.: Starting-point strategies for an infeasible potential reduction method. Optim. Lett. **4**, 131–146 (2010)
17. Davis, T.A., Hager, W.W.: Dynamic supernodes in sparse Cholesky update/downdate and triangular solves. ACM Trans. Math. Softw. **35**, 27:1–27:23 (2009)
18. Dollar, H.S.: Constraint-style preconditioners for regularized saddle point problems. SIAM J. Matrix Anal. Appl. **29**, 672–684 (2007)
19. Dollar, H.S., Gould, N.I.M., Schilders, W.H.A., Wathen, A.J.: Using constraint preconditioners with regularized saddle-point systems. Comput. Optim. Appl. **36**, 249–270 (2007)
20. Dollar, H.S., Wathen, A.J.: Approximate factorization constraint preconditioners for saddle-point matrices. SIAM J. Sci. Comput. **27**, 1555–1572 (2006)
21. Durazzi, C., Ruggiero, V.: Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems. Numer. Linear Algebra Appl. **10**, 673–688 (2003)
22. Freund, R., Nachtigal, N.: Software for simplified Lanczos and QMR algorithms. Appl. Numer. Math. **19**, 319–341 (1995)
23. Friedlander, M.P., Orban, D.: A primal-dual regularized interior-point method for convex quadratic programs. Math. Program. Comput. **4**, 71–107 (2012)
24. Forsgren, A., Gill, P.E., Griffin, J.D.: Iterative solution of augmented systems arising in interior methods. SIAM J. Optim. **18**, 666–690 (2007)
25. Gondzio, J.: Interior point methods 25 years later. Eur. J. Oper. Res. **218**, 587–601 (2012)
26. Gondzio, J.: Matrix-free interior point method. Comput. Optim. Appl. **51**, 457–480 (2012)
27. Gould, N.I.M., Orban, D., Toint, Ph.L.: CUTEst: a constrained and unconstrained testing environment with safe threads. Technical Report RAL-TR-2013-005, STFC Rutherford Appleton Laboratory. Chilton, Oxfordshire (2013)
28. Keller, C., Gould, N.I.M., Wathen, A.J.: Constraint preconditioning for indefinite linear systems. SIAM J. Matrix Anal. Appl. **21**, 1300–1317 (2000)
29. Lukšan, L., Vlček, J.: Indefinitely preconditioned inexact newton method for large sparse equality constrained non-linear programming problems. Numer. Linear Algebra Appl. **5**, 219–247 (1998)
30. Meurant, G.: On the incomplete Cholesky decomposition of a class of perturbed matrices. SIAM J. Sci. Comput. **23**, 419–429 (2001)
31. Perugia, I., Simoncini, V.: Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. Numer. Linear Algebra Appl. **7**, 585–616 (2000)
32. Sesana, D., Simoncini, V.: Spectral analysis of inexact constraint preconditioning for symmetric saddle point matrices. Linear Algebra Appl. **438**, 2683–2700 (2013)
33. Wang, W., O'Leary, D.P.: Adaptive use of iterative methods in predictor-corrector interior point methods for linear programming. Numer. Algorithms **25**, 387–406 (2000)
34. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM, Philadelphia (1997)