



On preconditioner updates for sequences of saddle-point linear systems

Valentina De Simone^{1*}, Daniela di Serafino¹, Benedetta Morini²

¹Dipartimento di Matematica e Fisica,
Università degli Studi della Campania “L. Vanvitelli”, Caserta, Italia

²Dipartimento di Ingegneria Industriale,
Università degli Studi di Firenze, Firenze, Italia

*Email address for correspondence: valentina.desimone@unicampania.it

Communicated by Giorgio Fotia

Received on 10 17, 2017. Accepted on 12 04, 2017.

Abstract

Updating preconditioners for the solution of sequences of large and sparse saddle-point linear systems via Krylov methods has received increasing attention in the last few years, because it allows to reduce the cost of preconditioning while keeping the efficiency of the overall solution process. This paper provides a short survey of the two approaches proposed in the literature for this problem: updating the factors of a preconditioner available in a block LDL^T form, and updating a preconditioner via a limited-memory technique inspired by quasi-Newton methods.

Keywords: saddle-point matrices, sequences of linear systems, preconditioning, factorization updates, limited-memory techniques.

AMS subject classification: 65F08, 65F10, 90C25.

1. Introduction

We provide a short presentation of recent research on preconditioning sequences of large and sparse saddle-point linear systems of the form

$$(1) \quad \mathcal{M}_i x_i = b_i, \quad \mathcal{M}_i = \begin{bmatrix} H_i & A^T \\ A & -C_i \end{bmatrix}, \quad i = 1, \dots, \ell,$$

where $H_i \in \mathbb{R}^{n \times n}$ and $C_i \in \mathbb{R}^{m \times m}$, with $m \leq n$, are symmetric and positive semidefinite, and $\mathcal{M}_i \in \mathbb{R}^{(n+m) \times (n+m)}$ is nonsingular. These sequences appear, e.g., in interior-point methods for quadratic programming and in Lagrangian approaches for the solution of PDE problems, with applications to optimal control, elasticity, polycrystalline aggregates, etc. [1–5].

We focus on the efficient solution of sequence (1) by preconditioned Krylov subspace methods. Preconditioning a single linear system of the sequence is a deeply investigated topic (see, e.g., [1,6,7]), but the computation from scratch of a preconditioner for each linear system may be very expensive in a large-scale setting. On the other hand, given a preconditioner for some matrix \mathcal{M}_i , performing a low-cost update of it for preconditioning a subsequent matrix of the sequence, \mathcal{M}_j , has received attention only in the last few years and offers new perspectives in the iterative solution of saddle-point systems. An efficient preconditioning procedure for the whole sequence (1) is expected to build preconditioners which are less effective than those computed from scratch in terms of linear iterations, but more convenient in terms of computational cost for the overall solution process.

To the best of our knowledge, two approaches have been proposed for updating preconditioners for sequence (1). One of the approaches, presented in Section 2, has been developed for updating constraint preconditioners, i.e., preconditioners of the form

$$\mathcal{P}_i^c = \begin{bmatrix} G_i & A^T \\ A & -C_i \end{bmatrix},$$

where G_i is a symmetric positive definite approximation to H_i [1,6].^a This approach assumes that \mathcal{P}_i^c is available in a suitable factorized form for a matrix \mathcal{M}_i of the sequence, and then builds preconditioners for the subsequent matrices \mathcal{M}_j , with $j > i$, through *low-rank corrections of the Schur complement* of the (1,1) block of \mathcal{P}_i^c [8,9]. The other approach, presented in Section 3, consists in updating a preconditioner available for \mathcal{M}_i by means of a small number of independent vectors, usually gathered during the application of a Krylov solver to $\mathcal{M}_i x_i = b_i$ or to a subsequent matrix. It has been applied to constraint preconditioners, to block diagonal and triangular preconditioners, i.e., preconditioners of the form

$$\mathcal{P}_i^d = \begin{bmatrix} F_i & 0 \\ 0 & W_i \end{bmatrix}, \quad \mathcal{P}_i^t = \begin{bmatrix} F_i & \kappa A^T \\ 0 & -W_i \end{bmatrix},$$

where F_i and W_i are suitable symmetric positive definite matrices and κ is a scalar [6,7], and to sparse approximate factorizations of \mathcal{M}_i . This updating technique is inspired by limited-memory quasi-Newton methods and the resulting preconditioners are called *limited-memory preconditioners* [2,4,10,11]. The reader is referred to [1,6,7] for details on the blocks of \mathcal{P}_i^c , \mathcal{P}_i^d and \mathcal{P}_i^t and the spectral properties of the preconditioned matrices.

^aActually, the assumption of positive definiteness of G_i can be relaxed; for example, when $C = 0$, G_i is required to be positive definite only in the nullspace of A (see [1,6] for details).

2. Updating constraint preconditioners via low-rank corrections of the Schur complement

We focus on the update of a constraint preconditioner \mathcal{P}_i^c where the (1,1) block is defined as $G_i = \text{diag}(H_i)$. This is a common choice when sequence (1) comes from interior point methods for quadratic programming, where H_i is positive definite (see, e.g., [1]). In order to enhance readability, we denote by \mathcal{P}_{seed}^c the *seed* preconditioner \mathcal{P}_i^c , i.e., the preconditioner that has to be updated, and by \mathcal{M}_{seed} the corresponding matrix \mathcal{M}_i ; furthermore, for $j > i$, we remove the index j from any matrix associated with the j th system of the sequence.

The procedure proposed in [8] assumes that a seed constraint preconditioner is available for \mathcal{M}_{seed} in the following factorized form:

$$\mathcal{P}_{seed}^c = \begin{bmatrix} I & 0 \\ AG_{seed}^{-1} & I \end{bmatrix} \begin{bmatrix} G_{seed} & 0 \\ 0 & -S_{seed} \end{bmatrix} \begin{bmatrix} I & G_{seed}^{-1}A^T \\ 0 & I \end{bmatrix},$$

where I denotes the identity matrix of any appropriate size and S_{seed} is the negative Schur complement of the (1,1) block in \mathcal{P}_{seed}^c . A Cholesky factorization of S_{seed} is also assumed to be available:

$$(2) \quad S_{seed} = AG_{seed}^{-1}A^T + C_{seed} = LDL^T.$$

The updating procedure builds a preconditioner \mathcal{P}_{lr} for \mathcal{M} in the form

$$\mathcal{P}_{lr} = \begin{bmatrix} I & 0 \\ AG^{-1} & I \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & -S_{lr} \end{bmatrix} \begin{bmatrix} I & G^{-1}A^T \\ 0 & I \end{bmatrix},$$

where S_{lr} is a low-rank update of S_{seed} aimed at providing a suitable approximation to the Schur complement $S = AG^{-1}A^T + C$ of \mathcal{P}^c .

It can be shown that the real and imaginary parts of the eigenvalues of the $\mathcal{P}_{lr}^{-1}\mathcal{M}$ can be bound essentially in terms of the minimum and maximum eigenvalues of $S_{lr}^{-1}S$ and other quantities depending on how G approximates H (see [8, Theorem 2.1]). The update of S_{lr} is driven by the spectral properties of the preconditioned matrix.

We briefly illustrate how S_{lr} is defined for the case $C_{seed} = C = 0$. S_{lr} takes the following form:

$$(3) \quad S_{lr} = S_{seed} + \bar{A}\bar{K}\bar{A}^T = A(G_{seed}^{-1} + K)A^T,$$

where $K \in \mathbb{R}^{n \times n}$ is a diagonal matrix with $q \ll n$ nonzero entries $(K)_{jj}$ on the diagonal, $\bar{K} \in \mathbb{R}^{q \times q}$ is the principal submatrix of K containing these nonzero entries, $\bar{A} \in \mathbb{R}^{m \times q}$ is made of the corresponding q columns of A , and $G_{seed}^{-1} + K$ accounts for major changes from G_{seed} to G . As mentioned

above, the eigenvalue distribution of $\mathcal{P}_{l_r}^{-1}\mathcal{M}$ depends on the minimum and maximum eigenvalues of $S_{l_r}^{-1}S$, which are, in turn, bounded respectively by the minimum and maximum values of the ratios $(G_{seed}^{-1} + K)_{jj}^{-1}/(G)_{jj}$. Hence, the q nonzero entries $(K)_{jj}$ are defined by selecting the indices j corresponding to the q_1 smallest and the q_2 largest ratios $(G_{seed})_{jj}/(G)_{jj}$, with $q_1 + q_2 = q$, and setting $(K)_{jj} = (G^{-1})_{jj} - (G_{seed}^{-1})_{jj}$. This choice provides bounds on the eigenvalues of $S_{l_r}^{-1}S$ that are between the bounds obtained by setting $S_{l_r} = S_{seed}$, i.e., reusing the preconditioner P_{seed}^c for \mathcal{M} , and those obtained by setting $S_{l_r} = S$, i.e., computing \mathcal{P}^c from scratch (see [8] for details). The low-rank update (3) is performed by updating the Cholesky factorization (2). Although the value of q must be small for limiting the cost of the factorization update, the preconditioning strategy is efficient because of the effect of tightening the eigenvalue bounds, especially when the q selected ratios $(G_{seed})_{jj}/G_{jj}$ are well separated from the remaining ones. Numerical experience shows that in general the quality of the updated preconditioner deteriorates as the distance between \mathcal{M} and \mathcal{M}_{seed} increases, and that periodically recomputing \mathcal{P}_{seed} from scratch is beneficial for the overall performance of the updating strategy [8].

The previous technique can be easily extended to the case where $C_{seed} \neq 0$, $C \neq 0$ and the positions of the nonzero entries of C_{seed} and C are the same [8]. Furthermore, an improvement of the updating strategy is proposed in [9] for this case. Here the low-rank correction of the Schur complement is enriched by performing a further update of S_{l_r} , which takes into account information not included in the first update and expressed as $S_{l_r} + D$, where D is a diagonal positive definite matrix. The procedure proposed in [12] is applied to obtain a cheap approximation of the LDL^T factorization of $S_{l_r} + D$, starting from the factorization of S_{l_r} . Theoretical results and numerical experiments show that the new strategy can be more effective than the procedure based on the low-rank modification alone [9].

Finally, we observe that the previous updating techniques can be applied starting from an inexact Cholesky factorization of S_{seed} , including approximate factorization techniques applicable when the Schur complement is not explicitly available, but matrix-vector products involving it are computable (see, e.g., [13,14]). However, preliminary experiments have shown that starting from inexact factorizations may significantly reduce the robustness of the overall updating procedure [9].

3. Limited memory preconditioning techniques

Limited Memory Preconditioners (LMPs) are based on the idea of improving a *first-level* preconditioner, henceforth referred to as seed precon-

ditioner, with a technique inspired by limited-memory quasi-Newton methods [15]. Initially developed for symmetric positive definite matrices [10,11], LMPs have been extended to saddle-point and to general indefinite matrices, and have been used in different applications, either to improve a preconditioner for a single system or to obtain preconditioners for sequences of linear systems with fixed or varying matrices [2,4,16].

Here we focus on LMPs for sequence (1) and adopt the same notation as in Section 2, with the difference that \mathcal{P}_{seed} and \mathcal{P} denote inverse preconditioners, i.e., approximations to \mathcal{M}_{seed}^{-1} and \mathcal{M}^{-1} , respectively.

Given a symmetric seed preconditioner \mathcal{P}_{seed} , a LMP preconditioner for a matrix \mathcal{M} can be defined as

$$(4) \quad \mathcal{P}_{lmp} = (I - \mathcal{T}\mathcal{M})\mathcal{P}_{seed}(I - \mathcal{M}\mathcal{T}) + \mathcal{T}, \quad \mathcal{T} = \mathcal{S}(\mathcal{S}^T\mathcal{M}\mathcal{S})^{-1}\mathcal{S}^T,$$

where $\mathcal{S} \in \mathbb{R}^{(n+m) \times q}$, $q \ll n + m$, is a suitable matrix such that $\mathcal{S}^T\mathcal{M}\mathcal{S}$ is nonsingular [16].^b When \mathcal{P}_{seed} is nonsymmetric, the LMP can be defined as

$$\tilde{\mathcal{P}}_{lmp} = (I - \mathcal{H}\tilde{\mathcal{T}}\mathcal{H}^T) + \tilde{\mathcal{T}}\mathcal{H}^T, \quad \tilde{\mathcal{T}} = \mathcal{S}(\mathcal{S}^T\mathcal{H}^T\mathcal{H}\mathcal{S})^{-1}\mathcal{S}^T,$$

where \mathcal{H} is the preconditioned matrix $\mathcal{P}_{seed}\mathcal{M}$ [4]. Note that the choice of a small value for q is driven by the computational cost of computing and applying $\mathcal{S}^T\mathcal{M}\mathcal{S}$ or $\mathcal{S}^T\mathcal{H}^T\mathcal{H}\mathcal{S}$. Details on this issue are given in [2,11,16].

Any LMP ensures that the preconditioned saddle-point matrix has at least q eigenvalues equal to 1. Furthermore, the eigenvalues of the preconditioned matrix enjoy a nonexpansion property with respect to the eigenvalues of $\mathcal{P}_{seed}\mathcal{M}$, provided that the columns of \mathcal{S} are the result of projections onto the invariant subspaces associated with the eigenvalues of $\mathcal{P}_{seed}\mathcal{M}$ in the open right half-plane and in the open left one. The columns of \mathcal{S} can be chosen in different ways: they can be linearly independent vectors generated during the application of a Krylov solver to the current system or to a previous system of the sequence, as well as eigenvectors or Ritz vectors associated with the current or a previous system.

In [16] the LMP update (4) is applied to constraint preconditioners for sequences of saddle-point systems with trailing block $C_i = 0$ that arise from interior point methods for quadratic programming. In this case, $\mathcal{S} = [\mathcal{S}_1^T \mathcal{S}_2^T]^T$, where $\mathcal{S}_1 \in \mathbb{R}^{n \times q}$ is full rank and $A\mathcal{S}_1 = 0$. In practice, these properties are satisfied by choosing the columns of \mathcal{S} as suitably normalized conjugate gradient (CG) directions, computed by performing q CG iterations on the previous system of the sequence or on the current one,

^bThe definition given in [2] is equivalent to (4) if \mathcal{P}_{seed} is symmetric positive definite and the preconditioners are applied as split preconditioners by using their Cholesky factorizations.

using an available constraint preconditioner. The updated preconditioner is still a constraint preconditioner, has an eigenvalue at 1 with multiplicity at least $2m + q$ and enjoys the aforementioned nonexpansion property of the eigenvalues. P_{seed} is periodically recomputed to keep the effectiveness of the updated preconditioner.

In [2,4] updates are used to precondition sequences of saddle-point linear systems arising in structural mechanics applications. The saddle-point matrices are either constant or varying through the sequence, with $C_i = 0$ in both cases. Augmented block-diagonal and block-triangular preconditioners, as well as preconditioners resulting from sparse factorizations of filtered saddle-point matrices, are used as seed preconditioners. The motivation for choosing augmented preconditioners is the high nullity of the leading block H_i of the saddle-point matrices. A fixed \mathcal{S} is generally used throughout the sequence; its columns are suitable Ritz vectors of the matrix of the first linear system $\mathcal{M}_1 x_1 = b_1$, recovered during the solution of that system by GMRES(ℓ). However, if the number of GMRES iterations increases too much, recomputing \mathcal{P}_{seed} and \mathcal{S} from the current system is beneficial [4].

For completeness we report that limited-memory techniques are also used to update preconditioners for sequences of saddle-point systems with varying off-diagonal blocks, arising in data assimilation [5]. In this case, computations with the off-diagonal blocks are expensive; hence, given a seed inexact constraint preconditioner with approximated off-diagonal blocks, updated preconditioners are obtained by applying to these blocks quasi-Newton techniques satisfying suitable secant and least-change properties.

Acknowledgements

This work was partially supported by INdAM-GNCS.

REFERENCES

1. M. D’Apuzzo, V. De Simone, and D. di Serafino, On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods, *Computational Optimization and Applications*, vol. 45, pp. 283–310, 2010.
2. S. Gratton, S. Mercier, N. Tardieu, and X. Vasseur, Limited memory preconditioners for symmetric indefinite problems with applications to structural mechanics, *Numerical Linear Algebra with Applications*, vol. 23, pp. 865–887, 2016.
3. J. W. Pearson and J. Gondzio, Fast interior point solution of quadratic programming problems arising from PDE-constrained optimization, *Numerische Mathematik*, vol. 137, pp. 959–999, 2017.

4. S. Mercier, S. Gratton, N. Tardieu, and X. Vasseur, A new preconditioner update strategy for the solution of sequences of linear systems in structural mechanics: applications to saddle point problems in elasticity, *Computational Mechanics*, vol. 60, pp. 969–982, 2017.
5. M. Fischer, S. Gratton, S. Gürol, Y. Tremolét, and X. Vasseur, Low-rank updates in preconditioning the saddle point systems arising from data assimilation problems, *Optimization Methods and Software*, 2016, DOI: 10.1080/10556788.2016.1264398.
6. M. Benzi, G. H. Golub, and J. Liesen, Numerical solution of saddle point problems, *Acta Numerica*, vol. 14, pp. 1–137, 2005.
7. A. Wathen, Preconditioning, *Acta Numerica*, vol. 24, pp. 329–376, 2015.
8. S. Bellavia, V. De Simone, D. di Serafino, and B. Morini, A preconditioning framework for sequences of diagonally modified linear systems arising in optimization, *SIAM Journal on Numerical Analysis*, vol. 50, pp. 3280–3302, 2012.
9. S. Bellavia, V. De Simone, D. di Serafino, and B. Morini, On the update of constraint preconditioners for regularized KKT systems, *Computational Optimization and Applications*, vol. 65, pp. 339–360, 2016.
10. J. L. Morales and J. Nocedal, Automatic preconditioning by limited memory quasi-Newton updating, *SIAM Journal on Optimization*, vol. 10, pp. 1079–1096, 2000.
11. S. Gratton, A. Sartenaer, and J. Tshimanga, On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides, *SIAM Journal on Optimization*, vol. 21, pp. 912–935, 2011.
12. S. Bellavia, V. De Simone, D. di Serafino, and B. Morini, Updating constraint preconditioners for KKT systems in quadratic programming via low-rank corrections, *SIAM Journal on Optimization*, vol. 25, pp. 1787–1808, 2015.
13. S. Bellavia, J. Gondzio, and B. Morini, A matrix-free preconditioner for sparse symmetric positive definite systems and least-squares problems, *SIAM Journal on Scientific Computing*, vol. 35, pp. A192–A211, 2013.
14. V. De Simone and D. di Serafino, A matrix-free approach to build band preconditioners for large-scale bound-constrained optimization, *Journal of Computational and Applied Mathematics*, vol. 268, pp. 82–92, 2014.
15. J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.
16. L. Bergamaschi, V. De Simone, D. di Serafino, and Á. Martínez, BFGS-like updates of constraint preconditioners for sequences of KKT linear systems in quadratic programming, *Numerical Linear Algebra with Applications*.