ON A CLASS OF LIMITED MEMORY PRECONDITIONERS FOR LARGE SCALE LINEAR SYSTEMS WITH MULTIPLE RIGHT-HAND SIDES*

SERGE GRATTON†, ANNICK SARTENAER‡, AND JEAN TSHIMANGA ILUNGA§

Abstract. This work is concerned with the development and study of a class of limited memory preconditioners for the solution of sequences of linear systems. To this aim, we consider linear systems with the same symmetric positive definite matrix and multiple right-hand sides available in sequence. We first propose a general class of preconditioners, called *Limited Memory Preconditioners* (LMP), whose construction involves only a small number of linearly independent vectors and their product with the matrix to precondition. After exploring and illustrating the theoretical properties of this new class of preconditioners, we more particularly study three members of the class named *spectral-LMP*, *quasi-Newton-LMP* and *Ritz-LMP*, and show that the two first correspond to two well-known preconditioners (see [8] and [20], respectively), while the third one appears to be a new and quite promising preconditioner, as illustrated by numerical experiments.

Key words. preconditioners, linear systems, conjugate gradient, limited memory

AMS subject classifications. 65K05, 65K10, 15A09, 15A15, 15A29

1. Introduction.

Consider a strictly convex quadratic function

$$q(x) = \tfrac{1}{2}x^T Ax - b^T x, \tag{1.1}$$

where A is a $n \times n$ symmetric positive definite matrix and x and $b \in \mathbb{R}^n$. Minimizing this function amounts to solve the linear system $Ax = b$. The most successful iterative method to solve this problem is the conjugate gradient (CG) method [14]. Starting from an initial guess x_0 (with corresponding residual $r_0 = \nabla q(x_0) = Ax_0 - b$), this method generates a sequence of iterates x_k that minimize q over the Krylov subspaces $x_0 + \mathrm{span}(r_0, Ar_0, \ldots, A^{k-1}r_0)$.

It is well known that, to exhibit a fast convergence, the CG method often needs a good *preconditioner*, that can be considered as an approximation of A^{-1}. We concentrate here on the case where a first preconditioner (called here *first-level preconditioner*) is already available (this preconditioner usually depends on the physics of the application that gives rise to a sequence of quadratic functions to minimize, such as in variational data assimilation for instance, see [7, 17, 31]), that is able to cluster most eigenvalues at 1, with relatively few outliers [5, 12, 30]. In order to improve the efficiency of this first-level preconditioner, we propose (and study) in this paper a class of *second-level preconditioners* whose aim is to capture directions in a low dimensional subspace that are left out by the first-level preconditioner and slowing down the convergence of CG.

Because A^{-1} is also the inverse Hessian of the quadratic function q, any inverse Hessian approximation obtained at an acceptable cost, as those encountered in *quasi-Newton methods* [24], is a good canditate to design preconditioners for CG. In particular, as shown in the present paper, *limited memory quasi-Newton methods* can be extended to provide an interesting framework for designing second-level preconditioners. To see that, consider, first, quasi-Newton methods for nonlinear optimization.

*This work was supported by CERFACS and FUNDP.

†CNES, av. E. Belin, and CERFACS, Toulouse, France.

‡Department of Mathematics, University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium.

§CERFACS, av. G. Coriolis, Toulouse, France.

These methods build a quadratic model of the objective function, f say, making use of only one gradient at each iteration to approximate the true Hessian or its inverse, which is good enough to ensure superlinear convergence. In the *BFGS method* (see [24]), the mostly used quasi-Newton method, the inverse Hessian approximation is updated as follows. Let H_{k-1} be a symmetric and positive definite $n \times n$ matrix approximation of the inverse Hessian of a function f at some iterate x_{k-1}, and let $\{s_k, y_k\}$ be a *correction pair* of vectors of $\mathbb{R}^n$ satisfying $s_k^T y_k > 0$. The *BFGS updating formula* to compute a new inverse Hessian approximation H_k based on these data is given by

$$H_k = \left(I_n - \frac{y_k s_k^T}{y_k^T s_k}\right)^T H_{k-1} \left(I_n - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}, \tag{1.2}$$

where I_n is the identity matrix of order n. This formula is derived so as to satisfy the so-called *secant equation* $H_k y_k = s_k$. Note that the vectors s_k and y_k in the BFGS method are defined by

$$s_k = x_k - x_{k-1} \qquad \text{and} \qquad y_k = \nabla f(x_k) - \nabla f(x_{k-1}), \tag{1.3}$$

with the step $s_k = \alpha_k p_k$ obtained from a (possibly inexact) line search[1] along the direction $p_k = -H_{k-1}\nabla f(x_{k-1})$ (see Chapter 8 of [24]).

Assume now that the objective function f is the quadratic function q defined in (1.1). The iterates generated by the BFGS method with exact line search are then identical to the iterates generated by the CG method (see [23]). The gradient difference y_k in (1.3) in that case writes $y_k = r_k - r_{k-1} = \alpha_k A p_k = A s_k$ (where $r_k = Ax_k - b$ denotes the residual at x_k), so that, defining

$$V_k = \left(I_n - A\frac{s_k s_k^T}{s_k^T A s_k}\right), \tag{1.4}$$

one can write

$$V_k^T H_{k-1} V_k = V_k^T (V_{k-1}^T H_{k-2} V_{k-1}) V_k + V_k^T \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T A s_{k-1}} V_k,$$

where we used (1.2) for H_{k-1}. From the fact that the vectors s_i are A-conjugate (thus $s_i^T A s_j = \delta_{ij}$, where δ_{ij} is the Kronecker symbol), follows that $V_k^T s_{k-1} = s_{k-1}$, which in turn implies that

$$V_k^T H_{k-1} V_k = V_k^T (V_{k-1}^T H_{k-2} V_{k-1}) V_k + \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T A s_{k-1}} \tag{1.5}$$

$$= \ldots = V_k^T V_{k-1}^T \ldots V_1^T H_0 V_1 \ldots V_{k-1} V_k + \sum_{i=1}^{k-1} \frac{s_i s_i^T}{s_i^T A s_i}. \tag{1.6}$$

Substituting this last expression in (1.2) now gives

$$H_k = V_k^T V_{k-1}^T \ldots V_1^T H_0 V_1 \ldots V_{k-1} V_k + \sum_{i=1}^{k} \frac{s_i s_i^T}{s_i^T A s_i}.$$

[1] Typically one imposes on α_k to satisfy the well-known Wolfe conditions.

Invoking again the $A-$conjugacy of the s_i, we observe that $V_1 \dots V_{k-1} V_k = I_n - \sum_{i=1}^k A \frac{s_i s_i^T}{s_i^T A s_i}$, which finally shows that

$$H_k = \left(I_n - \sum_{i=1}^k \frac{s_i s_i^T}{s_i^T A s_i} A \right) H_0 \left(I_n - \sum_{i=1}^k A \frac{s_i s_i^T}{s_i^T A s_i} \right) + \sum_{i=1}^k \frac{s_i s_i^T}{s_i^T A s_i}.$$

If we now define the $n \times k$ matrix $S = [s_1, \dots, s_k]$, set $M = H_0$, and note that $(S^T A S)^{-1} = \mathrm{diag}((s_i^T A s_i)^{-1})$, we observe that

$$H_k = \left[I_n - S(S^T A S)^{-1} S^T A \right] M \left[I_n - A S (S^T A S)^{-1} S^T \right] + S(S^T A S)^{-1} S^T.$$

This motivates the following definition, where the matrix M plays the role of the first-level preconditioner introduced above.

DEFINITION 1.1. *Let A and M be symmetric and positive definite matrices of order n and assume that S is any $n \times k$ matrix of rank k, with $k \leq n$. The symmetric matrix*

$$H = \left[I_n - S(S^T A S)^{-1} S^T A \right] M \left[I_n - A S (S^T A S)^{-1} S^T \right] + S(S^T A S)^{-1} S^T \tag{1.7}$$

is called Limited Memory Preconditioner (LMP).

It is the main purpose of this paper to study and investigate the potential of the LMP matrix H when considered as a preconditioner of A. Formula (1.7) is by no mean new. In the framework of multiple secant updates, it appears when equation (3.15) in [27] is particularized for strictly convex quadratic problems. In this case S contains all the steps encountered during the BFGS algorithm and $Y = AS$. Note that the inverse of H, that can be obtained using the Sherman-Morrison formula, is given in [4, 27]. The way our preconditioner is derived is also similar to the one proposed in [20]. Remarkably and independently, formula (1.7) also appears as a preconditioner for domain decomposition in [18, 22], under the name *balancing* preconditioner. This case can be considered as a generalization of the quasi-Newton approach, in the sense that the matrix S does not consist of descent directions generated by the CG algorithm. It is also worth noticing that (1.7) is interpreted in [18] as a two-level multigrid operator.

A first useful property of the LMP (1.7) is the following geometrical property: the LMP is invariant under change of basis of range(S). Indeed, if $Z = SX$, where X is a $k \times k$ invertible matrix, then replacing $Z = SX$ into $Z(Z^T A Z)^{-1} Z^T$ yields $Z(Z^T A Z)^{-1} Z^T = S(S^T A S)^{-1} S^T$, which shows that

$$\begin{aligned} H &= \left[I_n - Z(Z^T A Z)^{-1} Z^T A \right] M \left[I_n - A Z (Z^T A Z)^{-1} Z^T \right] + Z(Z^T A Z)^{-1} Z^T \\ &= \left[I_n - S(S^T A S)^{-1} S^T A \right] M \left[I_n - A S (S^T A S)^{-1} S^T \right] + S(S^T A S)^{-1} S^T. \end{aligned} \tag{1.8}$$

The paper is organized as follows. In Section 2, the main theoretical section of the paper, we explore some basic properties of the LMP. We first recover the property that this preconditioner is such that the preconditioned matrix has a condition number smaller than the condition number of A in most cases. We next address the issue of enforcing the clustering properties when the original matrix has an eigenvalue distribution already presenting some clusters. As already pointed out in [22], the LMP is not scaling invariant, since its effect on a linear system $\gamma A x = \gamma b$ depends on $\gamma > 0$. We present a strategy to choose a value for γ that yields the best possible condition number. At the end of the section, we derive a possible factored form $H = GG^T$, which is of interest for so-called *square root algorithms*, where maintaining

the symmetry and positive definitiveness is a concern. We also describe how this factorization can be used to ensure a numerically robust preconditioning technique when linear least-squares problems are solved with the LSQR or CG algorithms (see [2]).

Section 3 is mainly concerned with implementation details. An algorithm with complexity in $O(k^2n)$ in memory and in number of floating point operations, that resemble the L-BFGS recursion [24, p.225], is presented. In the rest of this section, we show that this memory and computational cost can be further reduced when the columns of S contain particular vectors that are usually available when the solution of linear systems with right-hand sides available in sequence is considered. We present numerical illustrations on linear systems with slowly varying right-hand sides and with matrices taken from the Harwell-Boeing Collection[2] of matrices. We finally show the numerical performance of several variants of the LMP, where different possible choices are considered for S, by showing the reduction of conjugate gradient steps required to achieve convergence. Note that the relevance of the LMP for large scale problems was already demonstrated in [29] on large scale nonlinear least-squares problems. Concluding remarks and perspectives are proposed in Section 4.

2. General properties. As a first elementary property, we have that H in (1.7) equals A^{-1} when S has n columns that span $\mathbb{R}^n$.

PROPOSITION 2.1. *If S is square in Definition 1.1, that is $k = n$, then $H = A^{-1}$.*

Proof. Since $k = n$, it follows from the assumptions of Definition 1.1 that S is a nonsingular matrix. We thus have that $S(S^TAS)^{-1}S^T = S(S^{-1}A^{-1}S^{-T})S^T = A^{-1}$. Replacing this result in (1.7) completes the proof. □

2.1. Non expansion of the spectrum of HA. First observe that since A is symmetric positive definite and S has full rank, the matrix S^TAS is also symmetric positive definite, according to Theorem 4.2.1 in [13, p. 141], so that S^TAS has an inverse and can be factored as $S^TAS = (S^TAS)^{1/2}(S^TAS)^{1/2}$. For further use, we rewrite equation (1.7) as

$$H = [I_n - WW^TA]M[I_n - AWW^T] + WW^T, \tag{2.1}$$

where

$$W \equiv S(S^TAS)^{-1/2}. \tag{2.2}$$

We have

$$W^TAW = (S^TAS)^{-1/2}S^TAS(S^TAS)^{-1/2} = I_k, \tag{2.3}$$

where I_k is the identity matrix of order k. We will refer to the $n \times (n-k)$ matrix $\underline{W}$ as any A-conjugate complement matrix of the $n \times k$ matrix W, whose columns are also A-conjugate and normalized with respect to the A-scalar product. That is, by definition and in addition to $W^TAW = I_k$, W and $\underline{W}$ satisfy

$$\underline{W}^TAW = 0, \tag{2.4}$$

$$\underline{W}^TA\underline{W} = I_{n-k}. \tag{2.5}$$

Let us state a lemma which will play a *central role* in the analysis of the effect of a LMP on matrix A.

[2] http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/

LEMMA 2.2. *Let A and M be symmetric positive definite matrices of order n and H be given by equation (1.7) in Definition 1.1, or, equivalently, by equation (2.1). Then the orthogonal matrix $Q = A^{1/2}\underline{W} \in R^{n\times(n-k)}$ is such that the preconditioned matrix $A^{1/2}HA^{1/2}$ admits the eigen decomposition*

$$A^{1/2}HA^{1/2} = [A^{1/2}W,\ A^{1/2}\underline{W}] \begin{pmatrix} I_k & 0 \\ 0 & V\Lambda V^T \end{pmatrix} \begin{pmatrix} W^TA^{1/2} \\ \underline{W}^TA^{1/2} \end{pmatrix}, \tag{2.6}$$

where V and Λ come from an eigen decomposition of $Q^TA^{1/2}MA^{1/2}Q$, i.e.,

$$V\Lambda V^T = Q^TA^{1/2}MA^{1/2}Q.$$

As consequences, the matrix H is positive definite and

$$\text{spectrum}(HA) = \{1\} \cup \text{spectrum}(Q^TA^{1/2}MA^{1/2}Q). \tag{2.7}$$

Proof. First observe that (2.3)-(2.5) imply that

$$\begin{pmatrix} W^T \\ \underline{W}^T \end{pmatrix} A[W, \underline{W}] = I_n.$$

Since span $[W, \underline{W}] = \mathbb{R}^n$, by premultiplying and postmultiplying this equality by $[W,\ \underline{W}]$ and $[W,\ \underline{W}]^{-1}$, respectively, we can write

$$[W, \underline{W}] \begin{pmatrix} W^T \\ \underline{W}^T \end{pmatrix} A = I_n,$$

or, equivalently,

$$WW^TA + \underline{W}\,\underline{W}^TA = I_n. \tag{2.8}$$

Using this equality, we can now transform the matrix form (2.1) as follows

$$H = (\underline{W}\,\underline{W}^TA)M(A\underline{W}\,\underline{W}^T) + WW^T,$$

which in turn is equivalent to

$$H = [W,\ \underline{W}] \begin{pmatrix} I_k & 0 \\ 0 & \underline{W}^TAMA\underline{W} \end{pmatrix} \begin{pmatrix} W^T \\ \underline{W}^T \end{pmatrix} \tag{2.9}$$

$$= [W,\ \underline{W}] \begin{pmatrix} I_k & 0 \\ 0 & Q^TA^{1/2}MA^{1/2}Q \end{pmatrix} \begin{pmatrix} W^T \\ \underline{W}^T \end{pmatrix}, \tag{2.10}$$

where we have set

$$Q = A^{1/2}\underline{W}. \tag{2.11}$$

Note that Q has orthonormal columns by (2.5), and thus satisfies $Q^TQ = \underline{W}^TA\underline{W} = I_{n-k}$. The structure of H in (2.10) along with the nonsingularity of $[W, \underline{W}]$ and the positive definiteness of $Q^TA^{1/2}MA^{1/2}Q$ reveals the positive definiteness of H. Moreover, again since $Q^TA^{1/2}MA^{1/2}Q$ is symmetric positive definite, its spectral decomposition implies that there must exist an orthogonal matrix V and a diagonal matrix Λ with

positive diagonal entries such that $Q^T A^{1/2} M A^{1/2} Q = \underline{W}^T A M A \underline{W} = V \Lambda V^T$. This, together with (2.10), implies

$$H = [W,\ \underline{W}] \begin{pmatrix} I_k & 0 \\ 0 & V\Lambda V^T \end{pmatrix} \begin{pmatrix} W^T \\ \underline{W}^T \end{pmatrix}. \tag{2.12}$$

Since A is symmetric positive definite, one can write, using (2.12),

$$\begin{aligned} A^{1/2} H A^{1/2} &= A^{1/2} [W,\ \underline{W}] \begin{pmatrix} I_k & 0 \\ 0 & V\Lambda V^T \end{pmatrix} \begin{pmatrix} W^T \\ \underline{W}^T \end{pmatrix} A^{1/2} \\ &= [A^{1/2} W,\ A^{1/2} \underline{W}] \begin{pmatrix} I_k & 0 \\ 0 & V\Lambda V^T \end{pmatrix} \begin{pmatrix} W^T A^{1/2} \\ \underline{W}^T A^{1/2} \end{pmatrix}, \end{aligned} \tag{2.13}$$

which is exactly (2.6) and corresponds to the spectral decomposition of the symmetric positive definite matrix $A^{1/2} H A^{1/2}$, because of the diagonal form in (2.13) and the fact that $[A^{1/2} W,\ A^{1/2} \underline{W}]$ is orthogonal, by (2.3)-(2.5), as well as the matrix V. From (2.6) and the fact that HA and the symmetric matrix $A^{1/2} H A^{1/2}$ are similar and thus have the same set of eigenvalues, we can finally deduce (2.7). □

We are now in position to show the effect of preconditioning the matrix A by H. We will base our analysis on the following result which is related to the clusterization of the spectrum of the preconditioned matrix HA.

THEOREM 2.3. *Let the positive real numbers $\sigma_1, \ldots, \sigma_n$ denote the eigenvalues of MA arranged in nondecreasing order. Then the eigenvalues $\mu_1, \ldots, \mu_n$ of HA can be ordered so that*

$$\begin{cases} \sigma_j \le \mu_j \le \sigma_{j+k} & \text{for } j \in \{1, \ldots, n-k\}, \\ \mu_j = 1 & \text{for } j \in \{n-k+1, \ldots, n\}, \end{cases} \tag{2.14}$$

where the real numbers $\mu_1, \ldots, \mu_{n-k}$ are the eigenvalues of $C \equiv Q^T A^{1/2} M A^{1/2} Q \in \mathbb{R}^{(n-k)\times(n-k)}$. In addition, the condition number $\frac{\max \mu_j}{\min \mu_j}$ of HA can be bounded as follows

$$\frac{\max_{j=1,\ldots,n} \mu_j}{\min_{j=1,\ldots,n} \mu_j} \le \frac{\max\{1, \sigma_n\}}{\min\{1, \sigma_1\}}. \tag{2.15}$$

Proof. For any symmetric matrix X, we denote by $\lambda_j(X)$ its j-th eigenvalue when the eigenvalues are sorted in nondecreasing order. Defining $B = A^{1/2} M A^{1/2} \in \mathbb{R}^{n\times n}$, we first observe that since the two matrices B and MA have the same eigenvalues, $\sigma_j = \lambda_j(B)$ for any $j = 1, \ldots, n$. By Corollary 4.3.16 of [15, p. 190] and the fact that $C = Q^T B Q$ where Q is an othogonal matrix (see Lemma 2.2), we have that, for $j = 1, \ldots, n-k$,

$$\sigma_j = \lambda_j(B) \le \lambda_j(C) \le \lambda_{j+k}(B) = \sigma_{j+k}, \tag{2.16}$$

which, together with (2.7) and the definition of C, yields (2.14). Considering now the extremal eigenvalues of HA gives $\max_{j=1,\ldots,n} \mu_j \le \max\{1, \sigma_n\}$ and $\min_{j=1,\ldots,n} \mu_j \ge \min\{1, \sigma_1\}$, which completes the proof. □

The theorem above shows that at least k eigenvalues of the matrix HA are equal to 1 and that the remaining part of the spectrum satisfies some interlacing property. In the particular case where S contains descent directions obtained on a quadratic function by BFGS, (2.15) is actually related to a known result attributed to Fletcher

(see [10] or [24, Theorem 8.3]). Note also that another possible proof of (2.15) may be derived by combining [21, Theorem 2.6] and [22, Theorem 2.5].

In order to illustrate the eigenvalue clusterization given in Theorem 2.3, we consider a set of three symmetric positive definite matrices that we call `Matrix 1`, `Matrix 2` and `Matrix 3`. Our goal is to *illustrate* the action of a LMP of the form (1.7) on the spectrum of some matrices, not to *explore* the effect of the LMP with respect to all possible eigen-distributions for the matrix to be preconditioned. We thus based our choice on three types of eigenvalue distributions, namely, the whole spectrum above 1, the spectrum "crossing" 1 with a group of eigenvalues in the neighborhood of 1, and finally the spectrum beneath 1.

`Matrix 1` is the 1244×1244 symmetric positive definite matrix BKSST27 from the Harwell-Boeing Sparse Matrix Collection. All its eigenvalues are greater than 1: the smallest is 1.40×10^2 while the largest is 3.5×10^6. `Matrix 2` is constructed from `Matrix 1` by symmetrically applying an incomplete Cholesky factorization to `Matrix 1`, with a drop tolerance of 2×10^{-3}. This first preconditioning step actually allows for a first clusterization at 1. The extremal eigenvalues of `Matrix 2` are then 0.007 and 36.0. `Matrix 3` is borrowed from [20]. This $n \times n$ tridiagonal positive definite matrix has coefficients given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & a & -a/2 & 0 & \cdots & 0 & 0 \\ 0 & -a/2 & a & -a/2 & \cdots & 0 & 0 \\ 0 & 0 & -a/2 & a & \ldots & 0 & 0 \\ & & & & \vdots & & \\ 0 & 0 & 0 & 0 & \ldots & -a/2 & a \end{bmatrix}, \tag{2.17}$$

where, for our purpose, we arbitrarily set $n = 100$ and $a = 10^{-3}$. The smallest and largest eigenvalues of `Matrix 3` are respectively 4.93×10^{-7} and 1.

The experiments consist in *randomly generating* vectors $s_1, \ldots, s_k$, with an arbitrary $k < n$, and to construct the associated LMP matrix H. One then compares the eigen-distribution of $MA = A$ (we assume in these experiments that $M = I_n$), with that of HA, where A is successively `Matrix 1`, `Matrix 2` and `Matrix 3`, choosing $k = 400, 300$ and 25, respectively.

The results are presented in two different categories. The first one concerns Figures 2.1, 2.2 and 2.3, where the eigenvalue distributions are displayed. These figures are given by pairs: the eigenvalue distribution of the matrix A is shown on the left and that of HA, on the right. In the title of each figure, the maximum and minimum eigenvalues of the original and preconditioned matrices are displayed. To clearly highlight the effect of the preconditioner on the spectrum, the same axis ranges are chosen for both A and HA, and horizontal lines are drawn to materialize the extreme eigenvalues. The vertical arrow shows the range of the spectrum and an horizontal line represents the eigenvalue 1. Concerning `Matrix 2` and `Matrix 3`, for which $\sigma = 1$ lies in $[\sigma_1, \sigma_n]$, we observe, as predicted by Theorem 2.3, that since $\max\{1, \sigma_n\} = \sigma_n$ and $\min\{1, \sigma_1\} = \sigma_1$, the spectrum is shrinked by the preconditioner (the length of the vertical arrow is decreased while moving from the spectrum of A to that of HA), and a cluster is created at 1 (the spectrum of HA exhibits a flat horizontal behaviour, corresponding to $\mu = 1$).

For `Matrix 1`, since $1 < \sigma_1$, $\sigma = 1$ is not in the range of the spectrum $[\sigma_1, \sigma_n]$. Theorem 2.3 shows then that the range of the spectrum of the preconditionned matrix is included in $[1, \sigma_n]$, which corresponds to Figure 2.1 (right). In this particular case,

the spectrum is not shrinked, which may not be ideal, since the condition number is not reduced. This situation can however be avoided by replacing the original system $Ax = b$ by $\gamma Ax = \gamma b$, where γ is defined by $\gamma = \frac{u^T u}{u^T Au}$ and u is a nonzero vector. Indeed, from the variationnal characterization of the eigenvalues of γA, if we denote by $R(x) = \frac{\gamma x^T Ax}{x^T x}$ the Rayleigh quotient of γA, we have $\sigma_1(\gamma A) = \min_{x\neq 0} R(x) \leq R(u) = 1 \leq \max_{x\neq 0} R(x) = \sigma_n(\gamma A)$, therefore $\sigma = 1$ is in the range of the spectrum of γA.

The second category of figures (Figures 2.4, 2.5 and 2.6) proposes an alternative view on the spectrum of the original and preconditioned matrices. For that purpose, histograms are displayed, in which eigenvalues have been clustered in classes centered on points corresponding to powers of 10 in the range $[10^{-10} : 10^{+10}]$. From these illustrations, we clearly observe the clusterization at 1 as well as the fact that the rest of the spectrum does not expand when H is applied to A confirming again the results of Theorem 2.3.

2.2. Behaviour with respect to an existing cluster at 1. In some applications, the matrix MA already has an eigenvalue cluster at 1 of dimension $m < n$, thanks to the first-level preconditioner M. In this case, the LMP can be considered as a second-level preconditioner aiming at further improving the spectrum of the matrix. As mentioned earlier, such a situation occurs in the area of variational data assimilation, where, after the so-called (first-level) preconditioning by the background term, A is typically a matrix of order 10^6 consisting in a rank 10^5 modification of the identity matrix. This implies that the cluster at 1 is of dimension $9 \cdot 10^5 = 10^6 - 10^5$ (see [29]).

From Theorem 2.3 follows that a matrix MA having a cluster at 1 of multiplicity m will be such that HA has a cluster at 1 of size k *at least* when H is a LMP. The issue that will be considered in this section is to obtain a better estimate of the true multiplicity p of 1 as eigenvalue of the preconditioned matrix HA. Particular situations of interest will be when $p > k$, or, even better, when $p > \max(m, k)$. The latter case is of practical importance since the cluster at 1 is greater for HA than for MA.

A first situation where the effect of a LMP on an eigenvalue cluster at 1 is easy to characterize is when the columns of S are eigenvectors of MA associated with this cluster. In this case the next proposition shows that $HA = MA$, i.e., the cluster size remains unchanged.

PROPOSITION 2.4. *Suppose that the columns of S are independent eigenvectors of MA associated with the eigenvalue 1. Then we have $H = M$, and therefore $HA = MA$.*

Proof. If the columns of S are eigenvectors of MA associated with the eigenvalue 1, we have $MAS = S$ and straightforward matrix manipulations show that

$$H = \left[I_n - S(S^T AS)^{-1} S^T A\right] M \left[I_n - AS(S^T AS)^{-1} S^T\right] + S(S^T AS)^{-1} S^T = M,$$

which ends the proof. □

Let now consider the general case for S. Our results are expressed using the notion of *harmonic Ritz pair*[3] *of a matrix with respect to a subspace*, that we recall now.

DEFINITION 2.5. *A pair (λ, x) is called an harmonic Ritz pair of A with respect to a subspace $\mathcal{L}$ if $x \in \mathcal{L}$ and $(Ax - \lambda x) \perp A\mathcal{L}$. If λ is an harmonic Ritz value, its*

[3]Not to confuse with *Ritz pair of a matrix with respect to a subspace*, see Section 3.2.2.

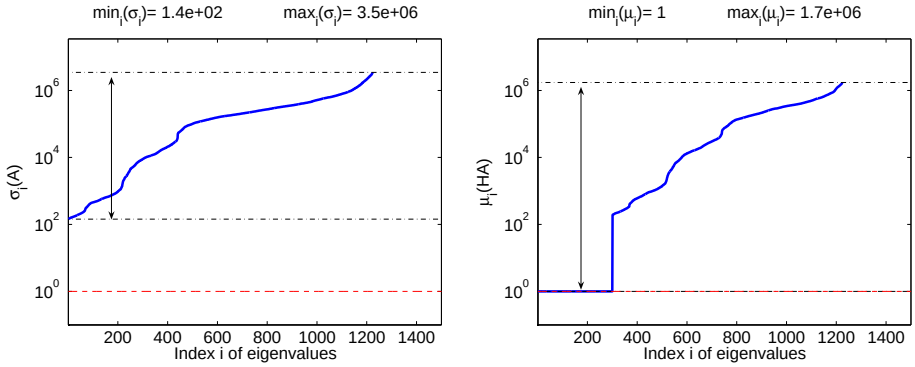

FIG. 2.1. *The eigen-distribution of A and HA for* `Matrix 1`.

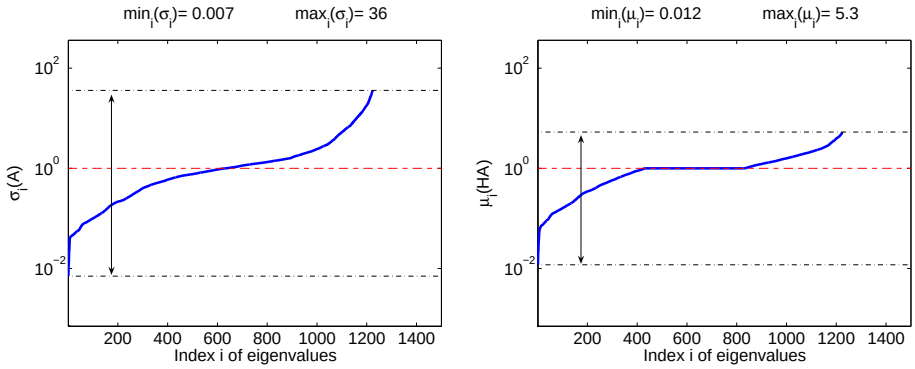

FIG. 2.2. *The eigen-distribution of A and HA for* `Matrix 2`.

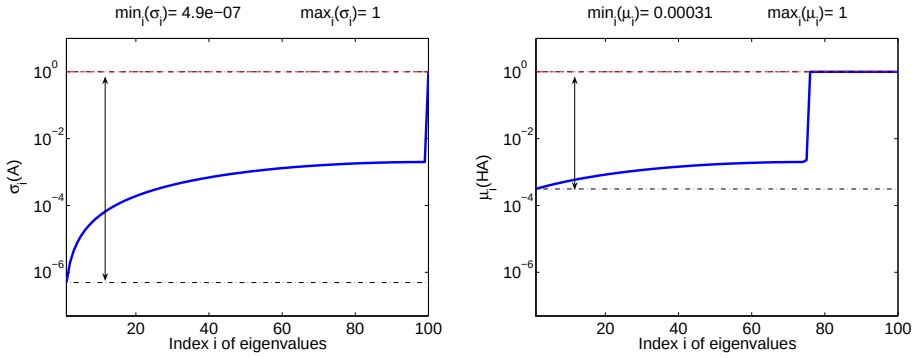

FIG. 2.3. *The eigen-distribution of A and HA for* `Matrix 3`.

multiplicity is the maximum number of independent vectors x_i such that (λ, x_i) is an harmonic Ritz pair of A with respect to $\mathcal{L}$.

PROPOSITION 2.6. *Let $S \in \mathbb{R}^{n\times k}$ be given. The multiplicity of* 1 *as an eigenvalue of HA is $k + n_r$, where n_r is the multiplicity of* 1 *as an harmonic Ritz value of MA with respect to the A-conjugate complement of range(S).*

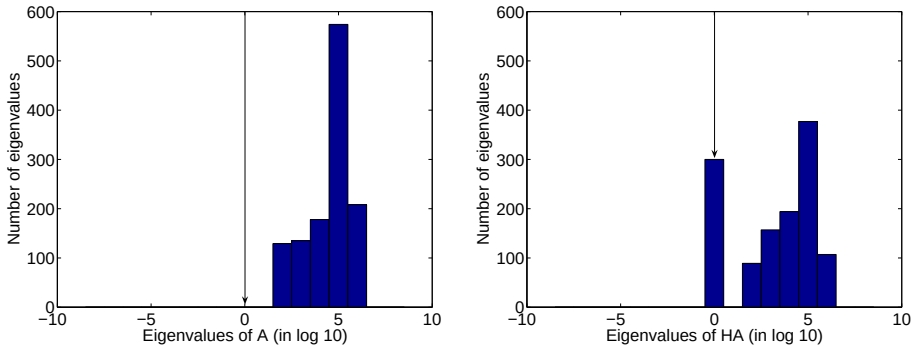

FIG. 2.4. *The histograms of eigen-distribution in classes of A and HA for* `Matrix 1`.

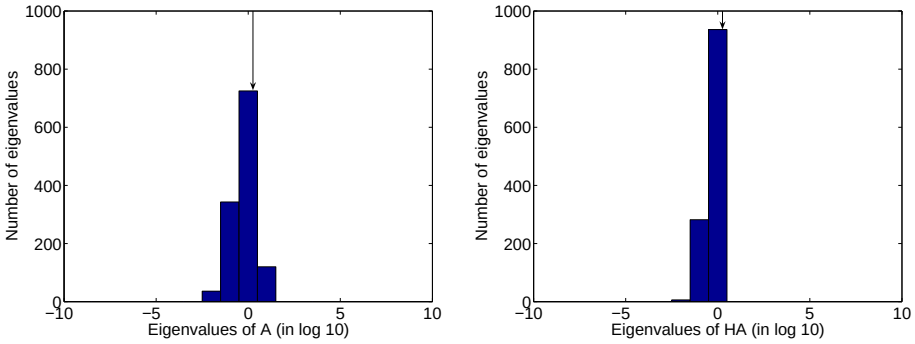

FIG. 2.5. *The histograms of eigen-distribution in classes of A and HA for* `Matrix 2`.

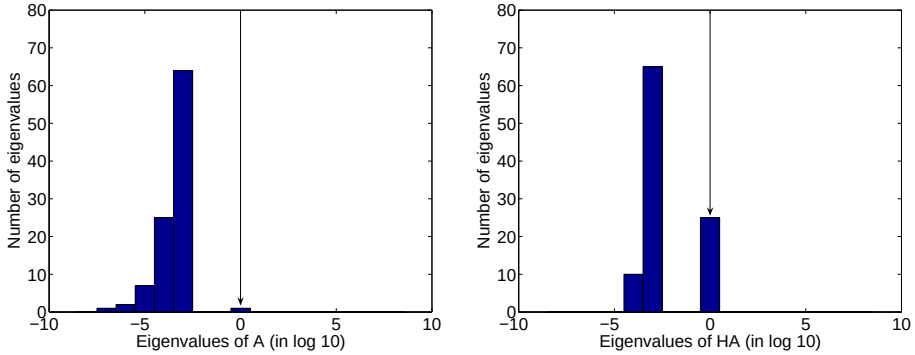

FIG. 2.6. *The histograms of eigen-distribution in classes of A and HA for* `Matrix 3`.

Proof. Let us first characterize the harmonic Ritz values of MA with respect to the A-conjugate complement of range(S) and consider the harmonic Ritz pair (λ, x). By definition of W in (2.2), we have that range(S) = range(W). Since $\underline{W}$ is an A-conjugate complement matrix of W, range$(\underline{W})$ is the A-conjugate complement of range(W) in $\mathbb{R}^n$, so that the pair (λ, x) is an harmonic Ritz pair of MA with respect to range$(\underline{W})$. But by definition of such an harmonic Ritz pair, we have that $x = \underline{W}y$ for some y in $\mathbb{R}^{n-k}$ and $MAx - \lambda x \perp A\underline{W}$, which implies, using (2.5),

that $\underline{W}^T AMA\underline{W}y = \lambda y$. This thus shows that the harmonic Ritz values of MA with respect to range($\underline{W}$) (and thus to the A-conjugate complement of range(S)) are exactly the eigenvalues of $\underline{W}^T AMA\underline{W}$. Using now the spectral decomposition (2.6) of $A^{1/2}HA^{1/2}$, we see that the multiplicity of 1 as eigenvalue of HA is $k + n_r$, where n_r is the multiplicity of 1 as eigenvalue of $Q^T A^{1/2}MA^{1/2}Q = \underline{W}^T AMA\underline{W}$. This completes the proof. □

We know that HA always has a cluster at 1 of size k at least. We explore in the next proposition a situation where the cluster size is strictly greater than k.

PROPOSITION 2.7. *Suppose that MA has an eigenvalue cluster at 1 of multiplicity $m > k$. Then HA has an eigenvalue cluster at 1 of multiplicity $m' \geq m$.*

Proof. Let $C \in \mathbb{R}^{n\times m}$ be a matrix whose columns are linearly independent eigenvectors of MA associated with 1. Denoting $\mathcal{C} = \text{range}(C)$ and $\underline{\mathcal{W}} = \text{range}(\underline{W})$ and noticing that $\dim(\mathcal{C}) = m$ and $\dim(\underline{\mathcal{W}}) = n - k$, we get from the Grassmann formula [19, p. 205] that $\dim(\mathcal{C}+\underline{\mathcal{W}})+\dim(\mathcal{C}\cap\underline{\mathcal{W}}) = \dim(\mathcal{C})+\dim(\underline{\mathcal{W}}) = m+n-k$. This implies, since $n \geq \dim(\mathcal{C} + \underline{\mathcal{W}})$, that $n + \dim(\mathcal{C} \cap \underline{\mathcal{W}}) \geq m + n - k$, i.e., that $\dim(\mathcal{C} \cap \underline{\mathcal{W}}) \geq m - k$. This last inequality shows that there are at least $m - k$ independent vectors that are both in $\mathcal{C}$ and in $\underline{\mathcal{W}}$, i.e., that are eigenvectors of MA associated with 1 belonging to range($\underline{W}$). These $m-k$ vectors are thus, in particular, independent harmonic Ritz vectors of MA with respect to range($\underline{W}$). The multiplicity n_r of Proposition 2.6 (remember that range($\underline{W}$) is the $A-$conjugate complement of range(S)) is therefore greater or equal to $m-k$, so that $m' = n_r+k \geq m-k+k = m$. This completes the proof. □

We continue our investigation about the behaviour of a possible pre-existing eigenvalue cluster of MA at 1. In the next proposition, the effect of the preconditioner is to *increase* by k at least the size of the cluster.

PROPOSITION 2.8. *Suppose that MA has an eigenvalue cluster at 1 of multiplicity m to which correspond m eigenvectors $c_1, \ldots, c_m$. Suppose that $S^T AC = 0$, where $C = [c_1, \ldots, c_m]$. Then HA has an eigenvalue cluster at 1 of multiplicity $m' \geq m+k$.*

Proof. By assumption, $S^T AC = 0$, and thus range(S) and range(C) are $A-$conjugate. But since range(S) = range(W) (see (2.2)), and $\underline{W}$ is an A-conjugate complement of W, we have that range(C) $\subseteq$ range($\underline{W}$). This implies that the m columns of C are eigenvectors of MA belonging to range($\underline{W}$), and thus are, in particular, harmonic Ritz vectors of MA with respect to range($\underline{W}$). The multiplicity n_r of Proposition 2.6 (remember that range($\underline{W}$) is the $A-$conjugate complement of range(S)) is therefore greater or equal to m, so that $m' = n_r + k \geq m + k$, which ends the proof. □

2.3. Scaling properties with respect to $\gamma > 0$ in $\gamma Ax = \gamma b$. It is well known that the conjugate gradient iterates on $\gamma Ax = \gamma b$ are the same for any $\gamma > 0$, provided that the same starting guess x_0 is taken. We assume that $M = I_n$ for the sake of clarity. Let us denote by H_γ the LMP (1.7) where A is replaced by γA

$$H_\gamma = \left[I_n - S(S^T AS)^{-1}S^T A\right]\left[I_n - AS(S^T AS)^{-1}S^T\right] + \frac{1}{\gamma}S(S^T AS)^{-1}S^T, \tag{2.18}$$

and by $x_k(\gamma)$ the $k-$th iterate of the conjugate gradient method on $\gamma Ax = \gamma b$ *preconditioned by* H_γ. In all this section, x_0 is assumed to be fixed in the comparisons made for different values of γ. Our purpose here is to investigate, following [22], whether the convergence of $x_k(\gamma)$ towards $x^\star = A^{-1}b$ can be accelerated by choosing a suitable value for γ.

Two important points are already clear from [22, pp. 1754-1755]. Firstly it is *numerically* shown that $x_k(\gamma)$ depends strongly on γ. Secondly, $x_k(\gamma)$ *does not* depend

on γ if $x_0 = S(S^TAS)^{-1}S^Tb$, which makes this choice for x_0 interesting. However this choice is not always the best one. Taking, if possible, x_0 closer to the solution or lying in a small invariant subspace of A containing the solution could yield interesting convergence properties, since it is known that the number of conjugate gradient steps to achieve convergence is equal to the dimension of the Krylov space $\mathrm{span}\{r_0, \ldots, A^{n-1}r_0\}$ [16].

Defining the matrix $\widetilde{A}(\gamma) = \gamma H_\gamma^{\frac{1}{2}} A H_\gamma^{\frac{1}{2}}$ and the condition number $\kappa(\gamma) = \kappa(\widetilde{A}(\gamma))$, and using the well-known convergence bound for the CG method (see [13, p. 530])

$$\|x_k(\gamma) - x^\star\|_{\widetilde{A}(\gamma)} \leq 2\|x_0 - x^\star\|_{\widetilde{A}(\gamma)} \left(\frac{\sqrt{\kappa(\gamma)} - 1}{\sqrt{\kappa(\gamma)} + 1}\right)^k, \tag{2.19}$$

our goal, in the next theorem, is to determine $\gamma > 0$ that minimizes the condition number $\kappa(\gamma)$.

THEOREM 2.9. *Let the positive real numbers $\mu_1, \ldots, \mu_{n-k}$ denote the eigenvalues of $Q^TAQ \in \mathbb{R}^{(n-k)\times(n-k)}$ arranged in nondecreasing order, where the orthogonal matrix Q is defined in Lemma 2.2. Then the minimum value of the condition number $\kappa(\gamma)$ is achieved for*

$$\gamma = \frac{u^Tu}{u^TQ^TAQu}, \tag{2.20}$$

where u is any nonzero vector, and is equal to $\frac{\mu_{n-k}}{\mu_1}$.

Proof. We know from Lemma 2.2 that, for A and M given,

$$\mathrm{spectrum}(HA) = \{1\} \cup \mathrm{spectrum}(Q^TA^{1/2}MA^{1/2}Q), \tag{2.21}$$

where

$$HA = \left[I_n - S(S^TAS)^{-1}S^TA\right] MA \left[I_n - S(S^TAS)^{-1}S^TA\right] + S(S^TAS)^{-1}S^TA. \tag{2.22}$$

Using (2.18) now gives

$$H_\gamma\gamma A = \left[I_n - S(S^TAS)^{-1}S^TA\right] \gamma A \left[I_n - S(S^TAS)^{-1}S^TA\right] + S(S^TAS)^{-1}S^TA,$$

so that the spectrum of $H_\gamma\gamma A$ is obtained from Lemma 2.2 by taking $M = \gamma I_n$ in equation (2.22), yielding

$$\mathrm{spectrum}(H_\gamma\gamma A) = \{1\} \cup \mathrm{spectrum}(\gamma Q^TAQ).$$

This implies that $\kappa(\gamma)$, which is the ratio between the largest and smallest eigenvalues of $\gamma H_\gamma A$, satisfies

$$\kappa(\gamma) = \frac{\max(1, \gamma\mu_{n-k})}{\min(1, \gamma\mu_1)} \geq \frac{\gamma\mu_{n-k}}{\gamma\mu_1} = \frac{\mu_{n-k}}{\mu_1}.$$

The minimum value $\frac{\mu_{n-k}}{\mu_1}$ for $\kappa(\gamma)$ is thus obtained if $1 \leq \gamma\mu_{n-k}$ and $1 \geq \gamma\mu_1$, that is, if $\mu_1 \leq \frac{1}{\gamma} \leq \mu_{n-k}$. From the variational characterization of the eigenvalues of a symmetric matrix follows that any nonzero vector u is such that $\mu_1 \leq \frac{u^TQ^TAQu}{u^Tu} \leq \mu_{n-k}$. Therefore, for any nonzero vector u, $\gamma = \frac{u^Tu}{u^TQ^TAQu}$ minimizes the condition number $\kappa(\gamma)$, which ends the proof. ∎

Using Theorem 2.9, we see that the use of $\gamma = \frac{u^T u}{u^T Q^T A Q u}$ is interesting when the conjugate gradient is preconditioned by the LMP (1.7), in that it yields the minimum condition number $\frac{\mu_{n-k}}{\mu_1}$ for the preconditioned matrix. Note that the proof of Theorem 2.9 shows that the same optimal condition number is obtained by considering the original system $Ax = b$ (i.e., setting $\gamma = 1$) while taking $M = \frac{u^T u}{u^T Q^T A Q u} I_n$ in the LMP definition. It is also interesting to note that in limited memory BFGS methods [24, p. 226], the choice $M = \frac{s^T y}{y^T y} I_n = \frac{s^T A s}{s^T A^2 s} I_n$ is made, that corresponds to our choice (2.20) if s is in the range of $\underline{W}$, i.e., if $s = A^{-\frac{1}{2}} Q u = \underline{W} u$ (see (2.11)). This choice is made in [24] in order to introduce a reasonable scaling in the descent directions, and is also justified in the present paper as a way of controlling the condition number of the preconditioned system.

2.4. Factored form of LMP. It is well known that the preconditioned CG method does not in principle require the preconditioner in a particular form. But, depending on the way the preconditioner is used in an iterative method, various forms might be needed (see [6, p. 3]): the preconditioner in a *factored* form, the *inverse* of the preconditioner, etc. Furthermore, having the preconditioner available in various forms makes it more flexible as mentioned in [1, pp. 277-302]. In the following proposition, we give some factored expression of the LMP matrix (1.7).

PROPOSITION 2.10. *Let $M = LL^T$, with $L^T \in \mathbb{R}^{n\times n}$, and $S^T A S = R^T R$, with $R \in \mathbb{R}^{k\times k}$, be any factored forms of M and $S^T A S$, respectively. Then, the LMP matrix H of the form (1.7) can be factored as $H = GG^T$, with G given by*

$$G = L - SR^{-1}R^{-T}S^T AL + SR^{-1}X^{-T}S^T L^{-T}, \tag{2.23}$$

where $S^T L^{-T} L^{-1} S = X^T X$, with $X \in \mathbb{R}^{k\times k}$, any factorization of $S^T L^{-T} L^{-1} S$. Equivalently, defining $N \in \mathbb{R}^{n\times k}$ and $Z \in \mathbb{R}^{k\times n}$ by $R^T N^T = S^T$ and $X^T Z = S^T$, we also have

$$G = L - NN^T AL + NZL^{-T}. \tag{2.24}$$

Proof. We proceed by direct computation, showing that $GG^T = H$ using (2.23). First observe that replacing $R^{-1}R^{-T}$ by $(S^T AS)^{-1}$ in (2.23) gives

$$G = L - S(S^T AS)^{-1}S^T AL + SR^{-1}X^{-T}S^T L^{-T}. \tag{2.25}$$

Now since

$$\begin{aligned}\left(L - S(S^T AS)^{-1}S^T AL\right)\left(SR^{-1}X^{-T}S^T L^{-T}\right)^T &= SX^{-1}R^{-T}S^T \\ &\quad -S(S^T AS)^{-1}(S^T AS)X^{-1}R^{-T}S^T \\ &= 0,\end{aligned}$$

and equivalently, by transposition,

$$SR^{-1}X^{-T}S^T L^{-T}\left(L - S(S^T AS)^{-1}S^T AL\right)^T = 0,$$

we have, using (2.25)

$$\begin{aligned}GG^T &= \left(L - S(S^T AS)^{-1}S^T AL\right)\left(L^T - L^T AS(S^T AS)^{-1}S^T\right) \\ &\quad +SR^{-1}X^{-T}S^T L^{-T}L^{-1}SX^{-1}R^{-T}S^T \\ &= \left(I_n - S(S^T AS)^{-1}S^T A\right)LL^T\left(I_n - AS(S^T AS)^{-1}S^T\right) \\ &\quad +SR^{-1}X^{-T}S^T L^{-T}L^{-1}SX^{-1}R^{-T}S^T.\end{aligned}$$

This last equality becomes, using $X^{-T}S^TL^{-T}L^{-1}SX^{-1} = I_k$, $LL^T = M$ and $S^TAS = R^TR$,

$$\begin{aligned} GG^T &= \left(I_n - S(S^TAS)^{-1}S^TA\right) M \left(I_n - AS(S^TAS)^{-1}S^T\right) + S(S^TAS)^{-1}S^T \\ &= H, \end{aligned}$$

by (1.7). This completes the first part of the proof. To prove the second part of the theorem, we replace S by NR or Z^TX in (2.23), which yields

$$\begin{aligned} G &= L - NRR^{-1}R^{-T}R^TN^TAL + NRR^{-1}X^{-T}X^TZL^{-T} \\ &= L - NN^TAL + NZL^{-T}, \end{aligned}$$

which ends the proof. □

Several comments can be made on the factor G obtained in Proposition 2.10. First relatively to the uniqueness of such a result, since $G'G'^T = GG^T$ is equivalent to $G^{-1}G'(G^{-1}G')^T = I_n$, any other square factor G' (including the Cholesky factor of H) can be written $G' = GQ$, where Q is an orthogonal matrix of order n. Concerning now practical computations, some factors G', such as the Cholesky factor of H, may be costly in terms of storage and computational cost when large problems with limited sparsity are considered (see [28]). The factorization we propose can be a reasonable alternative as we discuss now. It first relies on the availability of a factored form for the $n \times n$ matrix $M = LL^T$. If L is not available, computing this matrix will often be as expensive as computing directly the factored form $H = GG^T$. In the favourable case where L is available, the factorizations of the two $k \times k$ matrices $S^TL^{-T}L^{-1}S$ and S^TAS are needed in Proposition 2.10, which should not be a problem for practical situations where k is not too large.

We now give an illustration of the use of the LMP in algorithms where a factored form is *required* for a good numerical robustness. To this purpose, we choose the LSQR and CG algorithms preconditioned by the LMP for solving the least-squares problem

$$\min_{x \in \mathbb{R}^n} \|Jx - b\|_2^2, \tag{2.26}$$

where J is a full rank $m \times n$ matrix and $b \in \mathbb{R}^m$, with $m \geq n$. As described in [2], LSQR requires the preconditioner H to be factored according to $H = GG^T$, whereas CG does not. We consider in this example G as being the factor (2.23) of the LMP given in Proposition 2.10.

The generation of the test problem follows a strategy similar to the one described in [25, Section 8] and in [3, Section 6] and requires three integers m, n (with $m \geq n$) and p. The problem is built in a such way that the solution is known. The matrix of the problem has the singular value decomposition

$$J = Y \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} Z^T,$$

with Y, a $m \times m$ orthogonal matrix, Σ, a diagonal matrix of order n, and Z, a $n \times n$ orthogonal matrix. These three matrices are constructed as follows. The orthogonal matrices Z and Y are built as Householder reflectors by

$$Y = I_m - 2yy^T/y^Ty \quad \text{and} \quad Z = I_n - 2zz^T/z^Tz,$$

where $y = (y_1, \ldots, y_m)^T$ with $y_i = \sin(4i\pi/m)$ and $z = (z_1, \ldots, z_n)^T$ with $z_i = \cos(4i\pi/n)$. By means of the integer p, the singular values of J are set to $\sigma_i = [(n-i)/n]^p$, $i = 1, \ldots, n$, yielding $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$. To build the m-vector b, one first generates two other vectors: the solution of the problem, $x^* \in \mathbb{R}^n$, chosen as

$$x^* = (n-1, n-2, n-3, \ldots, 2, 1, 0)^T,$$

and the corresponding residual, $r^* \in \mathbb{R}^m$, given by

$$r^* = Y \begin{bmatrix} 0 \\ c \end{bmatrix},$$

where the components of the auxiliary vector $c = (c_1, \ldots, c_{m-n})^T \in \mathbb{R}^{m-n}$ are set to $c_i = (-1)^{i+1} i$. Observe that $\|r^*\|_2 = \|c\|_2$ and that $J^T r^* = 0$, since Y is orthogonal. The right-hand side b of (2.26) is then computed as

$$b = Jx^* + r^*.$$

We use for our particular problem the following values: $m = 290$, $n = 10$ and $p = 4$. The LMP is built with $k = 6$ column vectors in the matrix S (see the LMP Definition 1.1). To construct the preconditioner, we first run the unpreconditioned algorithm (LSQR or CG) on the normal equations $J^T Jx = J^T b$ corresponding to Problem (2.26). The iterations are stopped when the iterate x_k is such that the A-norm (here $A = J^T J$) of the solution error $e_k = x_k - x^*$, defined by $\|e_k\|_A = \sqrt{e_k^T A e_k}$, is less than a very small quantity, 10^{-16}, i.e., the convergence is reached in finite precision. The k last descent directions are then retained to build the LMP. A second least-squares problem is then considered with $b' = b + 10^{-5}\delta b$, where δb is a random vector of size m generated with the Matlab `randn` function. For the rest of this section, we focus on the solution of the perturbed least-squares problem: $\min_{x \in \mathbb{R}^n} \|Jx - b'\|_2^2$.

We use the LSQR and CG algorithms preconditioned by the LMP. The results of the experiments are shown in Figure 2.7, where the A-norm of the error of the iterates is plotted versus the number of iterations performed by the Krylov solver. On the left of the figure we have plotted the unpreconditioned runs whereas the preconditioned runs are shown on the right.

First observe that in the unpreconditioned run, LSQR gives a better final accuracy than CG in the A-norm of the error. A similar result is analyzed with the Euclidean norm in [3]. Concerning the preconditioned version, we see that the convergence is better than that obtained in the unpreconditioned cases, which tells us that the preconditioning has a beneficial effect. We also notice that LSQR (preconditioned with a factored form of the LMP) is better compared to CG (preconditioned with the LMP) when a sufficient number of iterations is performed.

To end this section, finally note that it is also possible to find formulas, in the same vein as Proposition 2.10, to compute a factorization *of the inverse* of G (see [28] for further detail).

3. Implementation considerations. In this section, some considerations are made on the implementation of the LMP. We first consider in Section 3.1 the case where the columns of S are general linearly independent vectors. Then we consider the case where the columns of S are vectors that arise in the solution by CG of multiple right-hand side systems $Ax = b_i$. We address situations where the columns of S belong to the Krylov subspaces encountered. A particular attention will be paid on

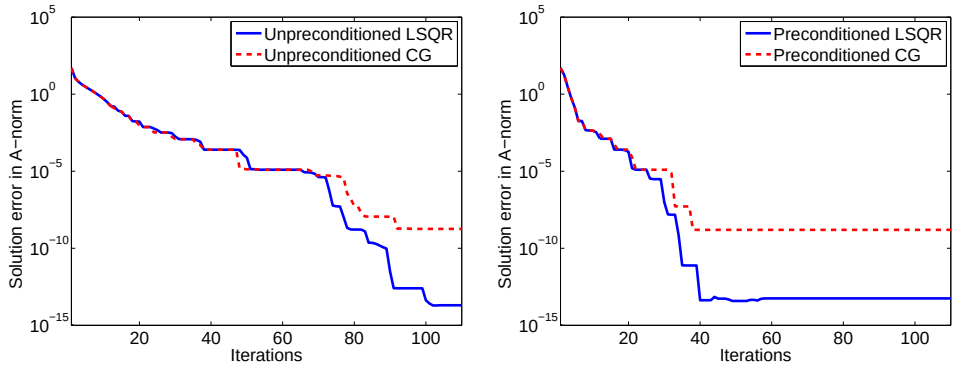

FIG. 2.7. *Result of using the LMP in a least-squares algorithm*

eigenvectors, descent directions and Ritz vectors, and we will show the connections between the resulting LMPs and existing preconditioners, such as the spectral preconditioner and the quasi-Newton preconditioner proposed in [20]. A set of Matlab experiments are shown at the end of the section where the theoretical results are illustrated on the academic test-cases of Section 2.1.

3.1. Memory and flops requirements for general $S \in \mathbb{R}^{n\times k}$. In this section we propose a possible implementation of the LMP preconditioner

$$H = \left[I_n - S(S^TAS)^{-1}S^TA\right] M \left[I_n - AS(S^TAS)^{-1}S^T\right] + S(S^TAS)^{-1}S^T,$$

in the case where S is a general $n \times k$ matrix. As mentioned in Section 1, H is invariant by right multiplication of S by a nonsingular matrix. Our approach consists in replacing S in H by a matrix $Z = [z_1, \ldots, z_k]$ spanning the same space as S, but whose columns are $A-$conjugate (thus $z_i^TAz_j = \delta_{ij}$). As a consequence, H writes $H = \left(I_n - ZZ^TA\right) M \left(I_n - AZZ^T\right) + ZZ^T$, or, if we set $Y = AZ$,

$$H = \left(I_n - ZY^T\right) M \left(I_n - YZ^T\right) + ZZ^T.$$

3.1.1. Memory and flops requirements in the general case for constructing the preconditioner. We decide to maintain in memory the two matrices Z and $Y = AZ$. These two matrices are obtained from S by running the following algorithm. Suppose that the preconditioner associated with $S_j = [s_1, \ldots, s_j]$, for some $1 \leq j < k$, has been stored, which means, using our representation, that Z_j and Y_j are known such that $Z_j^TAZ_j = I_j$, $Y_j = AZ_j$ and range(Z_j) =range(S_j). Let us now consider the preconditioner associated with $S_{j+1} = [S_j, s_{j+1}]$. We define, following a Gram-Schmidt process, $\bar{z} = s_{j+1} - Z_jZ_j^TAs_{j+1} = s_{j+1} - Z_jY_j^Ts_{j+1}$ and $\bar{y} = As_{j+1} - AZ_jY_j^Ts_{j+1} = As_{j+1} - Y_jY_j^Ts_{j+1}$. After normalization by $\rho = \sqrt{\bar{y}^T\bar{z}}$, we get $z_{j+1} = \frac{1}{\rho}\bar{z}$, $y_{j+1} = \frac{1}{\rho}\bar{y}$, and define $Z_{j+1} = [Z_j, z_{j+1}]$ and $Y_{j+1} = [Y_j, y_{j+1}]$. The Gram-Schmidt process produces an $A-$conjugate basis of range(S_{j+1}), which means that $Z_{j+1}^TAZ_{j+1} = I_{j+1}$, range$(Z_{j+1})$ =range(S_{j+1}), and by definition of y_{j+1} we have $Y_{j+1} = AZ_{j+1}$. Therefore, introducing a new vector in S can be done using the following algorithm.

Introduction of a new vector s_{j+1} in $S_j = [s_1, \ldots, s_j]$ $(1 \leq j < k)$
1. $f = Y_j^T s_{j+1}$ (costs $2jn$ flops)
2. $\bar{z} = s_{j+1} - Z_j f$ (costs $2jn$ flops)
3. $\bar{y} = As_{j+1} - Y_j f$ (costs $2jn$ flops and one product by A)
4. Normalization: $\rho = \sqrt{\bar{y}^T \bar{z}}$; $z_{j+1} = \frac{1}{\rho}\bar{z}$; $y_{j+1} = \frac{1}{\rho}\bar{y}$ (costs $4n$ flops)
5. Storage of an additional column in Z_j and Y_j to get $Z_{j+1} = [Z_j, z_{j+1}]$ and $Y_{j+1} = [Y_j, y_{j+1}]$

As a consequence, building $Z \in \mathbb{R}^{n\times k}$ and $Y = AZ$ corresponding to $S \in \mathbb{R}^{n\times k}$ by a repeated use of the procedure above approximately requires k matrix-vector products by A, and $\sum_{j=1}^{k} 6(j-1)n + 4kn \sim 3k^2 n$ additional floating point operations. In terms of storage, $2k$ vectors of length n are required to store both Y and Z.

3.1.2. Memory and flops requirements in the general case for applying the preconditioner. The computation of Hq can be done via the formula $Hq = \left(I_n - ZY^T\right) M \left(I_n - YZ^T\right) q + ZZ^T q$, which we implement using the following procedure that costs $8kn$ floating point operations and one product by M.

Computation of $r = Hq$
1. $f = Z^T q$ (costs $2kn$ flops)
2. $\bar{r} = M(q - Yf)$ (costs $2kn$ flops and one product by M)
3. $r = \bar{r} - Z(Y^T \bar{r} - f)$ (costs $4kn$ flops)

3.2. Particular types of the LMP.

3.2.1. The spectral-LMP. In this section, we investigate the properties of H when $M = I_n$ and when the columns $s_i \in \mathbb{R}^n$ of $S \in \mathbb{R}^{n\times k}$ are linearly independent eigenvectors of A.

PROPOSITION 3.1. *Let v_i, $i = 1, \ldots, k$, be k linearly independent unit eigenvectors of A associated with the eigenvalues λ_i, i.e., $Av_i = \lambda_i v_i$ and $v_i^T v_i = \delta_{ij}$. The LMP matrix H in (1.7) obtained with $M = I_n$ and $S = [v_1, \ldots, v_k]$ then satisfies*

$$H = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\lambda_i}\right) v_i v_i^T \right]. \tag{3.1}$$

Moreover, the factor G satisfying $H = GG^T$ in Proposition 2.10 writes

$$G = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\sqrt{\lambda_i}}\right) v_i v_i^T \right]. \tag{3.2}$$

Proof. If $\Lambda = \mathrm{diag}(\lambda_i)$, we have $AS = S\Lambda$ and $S^T S = I_k$, which shows that $(S^T AS)^{-1} = \Lambda^{-1}$ and $S(S^T AS)^{-1} S^T A = SS^T$. Using these relations in the LMP definition (1.7) together with $M = I_n$ yields

$$H = (I_n - SS^T)(I_n - SS^T) + S\Lambda^{-1} S^T = I_n + S(\Lambda^{-1} - I_k) S^T.$$

This last expression now simply writes

$$\begin{aligned} H &= I_n - \sum_{i=1}^{k} \left(1 - \frac{1}{\lambda_i}\right) v_i v_i^T \\ &= \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\lambda_i}\right) v_i v_i^T \right], \end{aligned}$$

thanks to the orthogonality of the vectors v_i, which completes the proof of (3.1). Consider now (2.23) in Proposition 2.10. Since $R^TR = S^TAS = \Lambda$ and $M = LL^T = I_n$, one can choose $R = \Lambda^{\frac{1}{2}}$ and $L = I_n$, respectively. Then $X^TX = S^TL^{-T}L^{-1}S = S^TS = I_k$, allowing to set $X = I_k$. These choices for R, L and X together with the fact that $S^TA = \Lambda S^T$ then give, by (2.23)

$$\begin{aligned} G &= I_n - S\Lambda^{-1}\Lambda S^T + S\Lambda^{-\frac{1}{2}}S^T \\ &= I_n - S(I_k - \Lambda^{-\frac{1}{2}})S^T. \end{aligned}$$

The final result (3.2) again follows from the orthogonality of the vectors v_i. □

We recognize in (3.1), which we call *spectral LMP*, the spectral preconditioner that is daily used in operational data assimilation (see [8, 9], for example). However, this preconditioner is only of interest when some eigen information is already available, or in situations with multiple right-hand sides. In the latter case indeed, computation of eigenvectors before the solution of the linear systems may be amortized if significant gain is obtained from the LMP and if the number of right-hand sides is large enough.

3.2.2. The Ritz-LMP for multiple right-hand side problems. In this section we suppose that we are solving a sequence of linear systems $Ax = b_i$ using the conjugate gradient algorithm. In that framework, at the end of CG on the first linear system $Ax = b_1$, we have access to all the information contained in the Krylov space. An interesting information, used in the Lanczos algorithm for eigenvalue problems, is the Ritz information. Our key idea is to construct a LMP matrix for the second linear system in which the columns of S are Ritz vectors, by using the equivalence in exact arithmetic between the CG and Lanczos algorithms [13]. The resulting LMP will be called *Ritz-LMP* in what follows. We first give the definition of a *Ritz pair* recalling that the related concept of *harmonic Ritz pair* is presented in Definition 2.5.

DEFINITION 3.2. *A pair (λ, x) is called a Ritz pair of A with respect to a subspace $\mathcal{L}$ if $x \in \mathcal{L}$ and $(Ax - \lambda x) \perp \mathcal{L}$. If λ is a Ritz value, its multiplicity is the maximum number of independent vectors x_i such that (λ, x_i) is a Ritz pair of A with respect to $\mathcal{L}$.*

We remind below the main properties of Ritz pairs (θ_i, z_i), $i = 1, \ldots, k$, obtained from the Lanczos algorithm

$$Az_i = \theta_i z_i + (e_k^T y_i)\delta_k q_{k+1}, \tag{3.3}$$

$$z_i^T A z_j = 0, \qquad z_i^T A z_i \neq 0, \tag{3.4}$$

$$z_i^T z_j = \delta_{ij}, \qquad z_i^T q_{k+1} = 0, \qquad q_k^T q_l = \delta_{kl}, \tag{3.5}$$

where e_k is the kth column of the $k \times k$ identity matrix I_k and δ_k, y_i and q_{k+1} are respectively a scalar and unit vectors defined in the Lanczos algorithm. Note that Ritz vectors are A-conjugate as well as orthonormal. Finally let us mention that the quantity $\rho_i = \|Az_i - \theta_i z_i\| = |(e_k^T y_i)\delta_k|$ gives a residual error bound of the Ritz pair (θ_i, z_i), see (3.3) and Theorem 9.1.2 in [13].

PROPOSITION 3.3. *Let z_i, $i = 1, \ldots, k$, be k Ritz vectors generated during the CG or Lanczos algorithms. Using the notation above, the LMP matrix H in (1.7) obtained with $M = I_n$ and $S = [z_1, \ldots, z_k]$ then satisfies*

$$H = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} - \omega_i^2 \right) z_i z_i^T - \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right], \tag{3.6}$$

where

$$\omega_i = \frac{(e_k^T y_i)\delta_k}{\theta_i}, \tag{3.7}$$

for $i = 1, \ldots, k$. *Moreover,* $H = GG^T$ *with*

$$G = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\sqrt{\theta_i}} \right) z_i z_i^T - \omega_i z_i q_{k+1}^T \right]. \tag{3.8}$$

Proof. Applying the A-conjugacy and orthonormality properties of the Ritz vectors together with the fact that $z_i^T A z_i = \theta_i$, for $i = 1, \ldots, k$, in the LMP definition (1.7), and using $M = I_n$, yields

$$\begin{aligned} H &= \left(I_n - \sum_{i=1}^{k} \frac{z_i z_i^T}{\theta_i} A \right) \left(I_n - \sum_{i=1}^{k} A \frac{z_i z_i^T}{\theta_i} \right) + \sum_{i=1}^{k} \frac{z_i z_i^T}{\theta_i} \\ &= I_n - \sum_{i=1}^{k} \frac{z_i z_i^T}{\theta_i} A - \sum_{i=1}^{k} A \frac{z_i z_i^T}{\theta_i} + \sum_{i=1}^{k} \frac{z_i z_i^T}{\theta_i} A \sum_{j=1}^{k} \frac{A z_j z_j^T}{\theta_j} + \sum_{i=1}^{k} \frac{z_i z_i^T}{\theta_i}. \end{aligned}$$

By (3.3) and (3.5), this last equality simplifies to

$$\begin{aligned} H &= I_n - \sum_{i=1}^{k} \left(1 - \frac{1}{\theta_i} \right) z_i z_i^T + \sum_{i=1}^{k} \frac{(e_k^T y_i)^2 \delta_k^2}{\theta_i^2} z_i z_i^T \\ &\quad - \sum_{i=1}^{k} \frac{(e_k^T y_i)\delta_k}{\theta_i} z_i q_{k+1}^T - \sum_{i=1}^{k} \frac{(e_k^T y_i)\delta_k}{\theta_i} q_{k+1} z_i^T \\ &= I_n - \sum_{i=1}^{k} \left(1 - \frac{1}{\theta_i} - \frac{(e_k^T y_i)^2 \delta_k^2}{\theta_i^2} \right) z_i z_i^T - \sum_{i=1}^{k} \frac{(e_k^T y_i)\delta_k}{\theta_i} \left(z_i q_{k+1}^T + q_{k+1} z_i^T \right). \end{aligned}$$

Thanks to (3.5) again, for $i = 1, \ldots, k$, we derive, using (3.7)

$$H = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} - \omega_i^2 \right) z_i z_i^T - \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right], \tag{3.9}$$

which gives (3.6). Consider now (3.8). By (3.5), one can write

$$\begin{aligned} GG^T &= \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\sqrt{\theta_i}} \right) z_i z_i^T - \omega_i z_i q_{k+1}^T \right] \prod_{j=1}^{k} \left[I_n - \left(1 - \frac{1}{\sqrt{\theta_j}} \right) z_j z_j^T - \omega_j z_j q_{k+1}^T \right]^T \\ &= \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\sqrt{\theta_i}} \right) z_i z_i^T - \omega_i z_i q_{k+1}^T \right] \left[I_n - \left(1 - \frac{1}{\sqrt{\theta_i}} \right) z_i z_i^T - \omega_i z_i q_{k+1}^T \right]^T \end{aligned}$$

$$
\begin{aligned}
&= \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\sqrt{\theta_i}}\right) z_i z_i^T - \omega_i z_i q_{k+1}^T - \left(1 - \frac{1}{\sqrt{\theta_i}}\right) z_i z_i^T + \left(1 - \frac{1}{\sqrt{\theta_i}}\right)^2 z_i z_i^T \right. \\
&\qquad \left. - \omega_i q_{k+1} z_i^T + \omega_i^2 z_i z_i^T \right] \\
&= \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} - \omega_i^2\right) z_i z_i^T - \omega_i (q_{k+1} z_i^T + z_i q_{k+1}^T) \right] \\
&= H,
\end{aligned}
$$

by (3.6).
□

Note the presence of the extra vector (which is not a Ritz vector) q_{k+1} from the Lanczos process on A in the expressions (3.6) and (3.8). This allows to have the factored form in an easier way. Note also some resemblance between the Ritz-LMP and the spectral-LMP versions. Similarities and differences between the two versions are discussed in more detail in Section 3.3.

3.2.3. The quasi-Newton-LMP for multiple right-hand side problems. The quasi-Newton preconditioner has been proposed and described by [20]. The idea of using quasi-Newton formula to update conjugate direction is also found in [26]. In the following, we derive the same preconditioner but as a *member* of our general LMP formulation (1.7). We will refer to this instance as the *quasi-Newton-LMP*.

As in the previous section, we assume that we are solving a sequence of linear systems of matrix A using the conjugate gradient algorithm. The vectors used to build the quasi-Newton-LMP are the conjugate gradient directions p_i produced during the run of an unpreconditioned CG algorithm for the solution of the first linear system. We recall the following well-known properties of conjugate gradient directions together with residuals obtained after k iterations of an unpreconditioned CG run (see [24, pp. 106-110])

$$
\text{for} \quad i = 1, \ldots, k, \quad p_i = -r_i + \beta_i p_{i-1}, \quad \text{where} \quad \beta_i = \frac{p_{i-1}^T A r_i}{p_{i-1}^T A p_{i-1}}, \tag{3.10}
$$

$$
\text{for} \quad i = 1, \ldots, k-1, \quad p_k^T A p_i = 0, \quad r_k^T p_i = 0, \quad r_k^T r_i = 0, \tag{3.11}
$$

$$
\text{for} \quad i = 1, \ldots, k-2, \quad r_k^T A p_i = 0. \tag{3.12}
$$

PROPOSITION 3.4. *Let p_i, $i = 1, \ldots, k$, be k conjugate gradient directions generated during the CG algorithm. The LMP matrix H in (1.7) obtained with $M = H_0$ and $S = [p_1, \ldots, p_k]$ is then inductively given by*

$$
H_k = \left(I_n - \frac{p_k p_k^T A}{p_k^T A p_k}\right) H_{k-1} \left(I_n - \frac{A p_k p_k^T}{p_k^T A p_k}\right) + \frac{p_k p_k^T}{p_k^T A p_k}. \tag{3.13}
$$

Furthermore, let $M = H_0 = I_n$ and the vectors r_i, for $i = 1, \ldots, k$, be available, then we have $H_k = G_k G_k^T$ with

$$
G_k = \left[I_n - p_k \left(\frac{p_k^T A}{p_k^T A p_k} + \frac{1}{\sqrt{p_k^T A p_k}} \frac{r_k^T}{\|r_k\|} \right) \right] G_{k-1}. \tag{3.14}
$$

Proof. The expression (3.13) for H_k is nothing else than the one we started with to define the LMP (1.7) (with $M = H_0$), see (1.2) in Section 1, where $s_k \equiv p_k$ and

$y_k \equiv Ap_k$. Consider now (3.14), we first prove by an obvious induction argument that $H_k = G_k G_k^T$ with

$$G_k = \left[I_n - p_k \left(\frac{p_k^T A}{p_k^T A p_k} - \frac{p_k^T H_{k-1}^{-1}}{\sqrt{p_k^T A p_k}\sqrt{p_k^T H_{k-1}^{-1} p_k}} \right) \right] G_{k-1}. \tag{3.15}$$

Indeed, by assumption we have that $H_0 = I_n = G_0 G_0^T$ (with $G_0 = I_n$). Assuming that $H_{k-1} = G_{k-1} G_{k-1}^T$, a direct computation then gives $H_k = G_k G_k^T$ with G_k given by (3.15). The next step in the proof consists in showing that $H_{k-1}^{-1} p_k = -r_k$, or equivalently $H_{k-1} r_k = -p_k$, using information from the conjugate gradient algorithm. To this aim, let us show by induction that $H_i r_k = r_k$ for $i = 0, \ldots, k-2$. Obviously $H_0 r_k = r_k$ since $H_0 = I_n$. Suppose now that $H_{i-1} r_k = r_k$ for $i = 1, \ldots, k-2$. We can write, by (3.13)

$$\begin{aligned} H_i r_k = & \left[\left(I_n - \frac{p_i p_i^T A}{p_i^T A p_i} \right) H_{i-1} \left(I_n - \frac{A p_i p_i^T}{p_i^T A p_i} \right) + \frac{p_i p_i^T}{p_i^T A p_i} \right] r_k \\ &= H_{i-1} r_k - \frac{p_i p_i^T A H_{i-1} r_k}{p_i^T A p_i} \\ &= r_k - \frac{p_i p_i^T A r_k}{p_i^T A p_i} \\ &= r_k, \end{aligned}$$

using successively (3.11), the induction assumption $H_{i-1} r_k = r_k$ and (3.12). This in particular implies $H_{k-2} r_k = r_k$, which, together with (3.13), (3.11) again for $i = k-1$ and (3.10) for $i = k$, yields

$$\begin{aligned} H_{k-1} r_k = & \left[\left(I_n - \frac{p_{k-1} p_{k-1}^T A}{p_{k-1}^T A p_{k-1}} \right) H_{k-2} \left(I_n - \frac{A p_{k-1} p_{k-1}^T}{p_{k-1}^T A p_{k-1}} \right) + \frac{p_{k-1} p_{k-1}^T}{p_{k-1}^T A p_{k-1}} \right] r_k \\ &= r_k - \frac{p_{k-1} p_{k-1}^T A r_k}{p_{k-1}^T A p_{k-1}} \\ &= r_k - \beta_k p_{k-1} \\ &= -p_k. \end{aligned}$$

Now replacing $H_{k-1}^{-1} p_k$ by $-r_k$ in (3.15) and observing that $p_k^T r_k = -r_k^T r_k$ by (3.10) and (3.11) give

$$G_k = \left[I_n - p_k \left(\frac{p_k^T A}{p_k^T A p_k} - \frac{1}{\sqrt{p_k^T A p_k}} \frac{r_k^T}{\|r_k\|} \right) \right] G_{k-1},$$

which ends the proof. □

3.3. Some comparisons. To draw comparison between the different LMP versions we need some guidelines. To this end observe that (1) spectral-LMP and Ritz-LMP rely on eigen-information and that (2) Ritz-LMP uses Krylov subspaces as quasi-Newton-LMP does. These similarities constitute the basis of the comparisons below.

3.3.1. Spectral-LMP versus Ritz-LMP. We know that both spectral and Ritz-LMP are constructed with information related to the spectrum of the matrix to precondition. The spectral-LMP uses in principle exact eigen-information whereas the Ritz-LMP is based on approximate eigenpairs. Often one does not have exact spectral information at his disposal. On the other hand, computing exact eigen-information for the solution of linear systems is *unacceptably prohibitive*, unless multiple right-hand side situations are considered, in which case this extra cost can be amortized. This is the major drawback of the spectral version of the LMP. Therefore, some implementations of spectral preconditioners use Ritz information instead of exact eigen-information (see [8]). It is then of interest for us to try to appreciate the implication of this approximation of the eigenvalues on the preconditioned matrix. In their paper on the sensitivity of some "spectral" preconditioners, the authors in [11] study the effect of the inexactness of eigenelements on the behavior of the resulting preconditioner, but their study is based on a first-order expansion and is therefore true only asymptotically. Our study, which is restricted to the approximation of eigenvectors by Ritz vectors, does not need to assume that the perturbations are small.

THEOREM 3.5. *Suppose that k Ritz pairs (θ_i, z_i), $i = 1, \ldots, k$, are available from a previous CG run on a linear system $Ax = b$ as described in Section 3.2.2. We denote by H the corresponding Ritz-LMP preconditioner (3.6). Let us denote by $\tilde{H}$ the preconditioner obtained by using the Ritz vectors z_i instead of true eigenvectors v_i in the spectral LMP expression (3.1). If the matrix F denotes the difference $\tilde{H} - H$ and if, as in (3.7), $\omega_i = \frac{(e_k^T y_i)\delta_k}{\theta_i}$, $i = 1, \ldots, k$, then*

$$\|F\|_2 \leq k \left(\max_i(\omega_i^2) + \max_i(|\omega_i|) \right).$$

Proof. We have, by (3.6) and (3.5)

$$\begin{aligned} H &= \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i} - \omega_i^2\right) z_i z_i^T - \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right] \\ &= I_n + \sum_{i=1}^{k} \left[- \left(1 - \frac{1}{\theta_i} - \omega_i^2\right) z_i z_i^T - \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right], \end{aligned}$$

and by (3.1) (with (λ_i, v_i) replaced by (θ_i, z_i)) and (3.5) again

$$\tilde{H} = \prod_{i=1}^{k} \left[I_n - \left(1 - \frac{1}{\theta_i}\right) z_i z_i^T \right] = I_n + \sum_{i=1}^{k} \left[- \left(1 - \frac{1}{\theta_i}\right) z_i z_i^T \right].$$

One can thus write

$$F = \tilde{H} - H = \sum_{i=1}^{k} \left[-\omega_i^2 z_i z_i^T + \omega_i (z_i q_{k+1}^T + q_{k+1} z_i^T) \right].$$

From the orthonormality properties of the vectors z_i and q_{k+1} (see (3.5)), observe that each matrix $(z_i q_{k+1}^T + q_{k+1} z_i^T)$, for $i = 1, \ldots, k$, has 1 and -1 as unique nonzero eigenvalues. These correspond to the eigenvectors $z_i + q_{k+1}$ and $z_i - q_{k+1}$, respectively.

This implies that the 2-norm of F can be bounded as follows

$$\begin{aligned}\|F\|_2 &\leq \sum_{i=1}^{k} \left(\|\omega_i^2 z_i z_i^T\|_2 + |\omega_i| \|z_i q_{k+1}^T + q_{k+1} z_i^T\|_2 \right) \\ &\leq \sum_{i=1}^{k} \left(\|\omega_i^2 z_i z_i^T\|_2 + |\omega_i| \right) \\ &\leq k \left(\max_i(\omega_i^2) + \max_i(|\omega_i|) \right),\end{aligned}$$

which is the desired result. □

We have $\omega_i = \frac{(e_k^T y_i)\delta_k}{\theta_i}$ and since $\rho_i = |(e_k^T y_i)\delta_k|$ represents a residual error bound on Ritz pairs (see Section 3.2.2), we say that the spectral-LMP built with Ritz information will behave like the Ritz-LMP when the information used has well converged (that is, ω_i is small).

Another point in comparing the spectral-LMP with the Ritz-LMP is the memory required to store information for the preconditioner. Propositions 3.1 and 3.3 show that the LMP matrix H need k eigenvectors for the spectral version and $k+1$ for the Ritz one. This is due to the fact that the Ritz-LMP uses a supplementary vector q_{k+1} in its formulation. The amount of necessary memory is thus *nearly the same* in both cases.

3.3.2. Ritz-LMP versus quasi-Newton-LMP. First, we discuss issues related to how vectors information are obtained. The information for quasi-Newton is freely collected from an unpreconditioned CG run. Nevertheless, some applications require, for diagnostic reasons of their data, the approximation of spectral information of the Hessian of a function of error. In such circumstances we can argue that Ritz information are by-products of diagnostic involved procedures. But things are different if that Ritz information is not needed elsewhere than in the preconditioner. In such cases, Ritz information should not be considered as by-product of the iterative solver.

The next theorem shows the equivalence of the two versions of the LMP if *all* available relevant information from a CG-like run is used.

THEOREM 3.6. *The Ritz-LMP and quasi-Newton-LMP preconditioners are analytically equivalent when the space spanned by the descent directions is the same as that spanned by Ritz vectors. This is the case when all available information (descent directions or Ritz vectors), extracted from a CG-like method run on the same matrix, is used in both preconditioners.*

Proof. The analytical equivalence between the Ritz-LMP and quasi-Newton-LMP preconditioners under the assumption that the space spanned by the descent directions is the same as that spanned by Ritz vectors relies on the earlier mentioned principle of invariance of the preconditioner H under change of basis of the columns vectors of the matrix S. We now give a sketch of the proof that, when all available information is used, the spaces spanned by the descent directions and the Ritz vectors coincide (the full proof, that relies on the equivalence between CG and the Lanczos algorithms, can be found in [28]). First, observe that in CG, descent directions as well as residuals span the same (Krylov) subspace, $\mathcal{K}$ say. Second, from the equivalence between the Lanczos and the CG algorithms, it is known that the Lanczos vectors form an *orthonormal* basis of $\mathcal{K}$ (see [13, 28]). Finally, the Ritz vectors are obtained by right multiplication of the Lanczos vectors with the eigenvectors of the tridiagonal matrix arising in the

Lanczos process. This leads to the conclusion that, at step k of Lanczos, the k Ritz vectors and the k descent directions span the same subpsace $\mathcal{K}$. Thus, the Ritz-LMP and quasi-Newton-LMP preconditioners are analytically equivalent. □

Concerning the memory requirement of these versions of the LMP, one should observe from Proposition 3.4 that a total amount of $2k$ vectors is necessary to built the quasi-Newton-LMP. Thus the quasi-Newton-LMP is about *twice more expensive* in storage than the Ritz-LMP which needs only $k+1$ vectors.

3.4. Numerical experiments with constant matrix. To illustrate numerically some of the results we have developed in the previous section, let us present the set of experiments. We choose the framework of solving *linear systems* in sequence.

Experiments consist in comparing performance of the three investigated LMPs: spectral-, Ritz- and quasi-Newton- versions. The systems are based on the matrices presented earlier in Section 2.1. The procedure of conducting the tests is given in the following.

We run, in a first step, k iterations of the unpreconditioned CG algorithm, on a linear system $Ax = b_1$. This first run provides the information for the preconditioner H. After constructing the preconditioner with the k available vectors (descent direction or Ritz vectors), we then run in a second step a preconditioned CG on a new system *with the same matrix* A but whose right-hand side is defined by $b_2 = Ax^*$, where x^* is a random vector. Indeed, knowing the solution of the system enables us to stop the preconditioned CG iterations when the A-norm of the error $[(x_k - x^*)^T A(x_k - x^*)]^{1/2}$ of the iterate is below the tolerance set at 10^{-5} without implementing a sophisticated estimator of this quantity.

Figures are given in pairs. On the left, the figure compares the preconditioners performance on a given system. The plot consists in number of *required preconditioned CG iterations* (ordinate) to reach the given tolerance in the A-norm of the error versus the *number of vectors* (abscissa) used to build the preconditioners. The curves with circles ($\circ$), triangles ($\triangle$) and plus ($+$) are related to the spectral-LMP (with Ritz information), quasi-Newton-LMP and Ritz-LMP, respectively. On the right, we plot the maximum of residual error bounds of the Ritz pairs (θ_i, z_i), $i = 1, 2, \ldots$, defined in Section 3.2.2 for the Ritz information used.

Results are shown in Figures 3.1, 3.2 and 3.3 for systems with `Matrix 1`, `Matrix 2` and `Matrix 3`, respectively. A first observation is that the performance increases with the number of vectors, whatever the preconditioner is. A second point is that the quasi-Newton-LMP gives almost the same result as the Ritz-LMP, as predicted by Theorem 3.6; Concerning now the spectral-LMP, note that it produces, in general, poor results compared to the two other considered LMPs. However, when the error in Ritz pairs (plotted on the right) is small, the spectral (with Ritz-information) starts behaving like the Ritz-LMP, this follows the result given in Theorem 3.5. Finally, we also mention that the three LMP preconditioners have been compared in [29], in a large scale ocean data assimilation system (more than 10^6 unknowns are considered). The superiority of the Ritz-LMP over the spectral-LMP and the quasi-Newton-LMP has been demonstrated for this particular application.

4. Conclusion. Using the approximation properties of limited memory quasi-Newton methods regarding inverse Hessian approximations, we have defined the *LMP class*, a class of limited memory preconditioners, that implements corrections of so-called first-level preconditioners, the latter preconditioners being usually application dependent.

The LMP involves k vectors forming the columns of a matrix S arising in the LMP definition. These vectors can be in theory any set of linearly independent vectors. However, we derived the particular forms taken by the LMP when vectors often

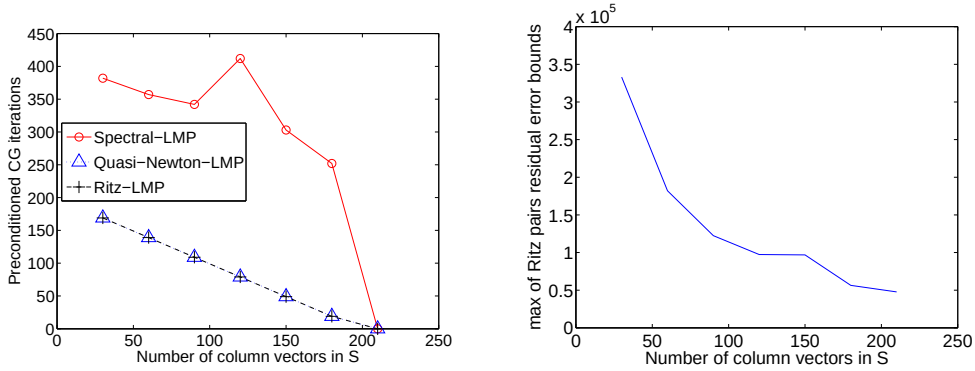

FIG. 3.1. *Performance of three LMPs versus number of columns of S using system of* `Matrix 1`

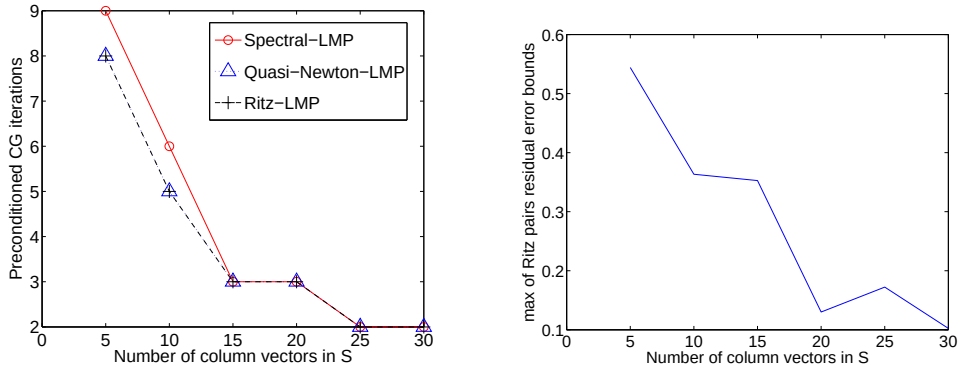

FIG. 3.2. *Performance of three LMPs versus number of columns of S using system of* `Matrix 2`

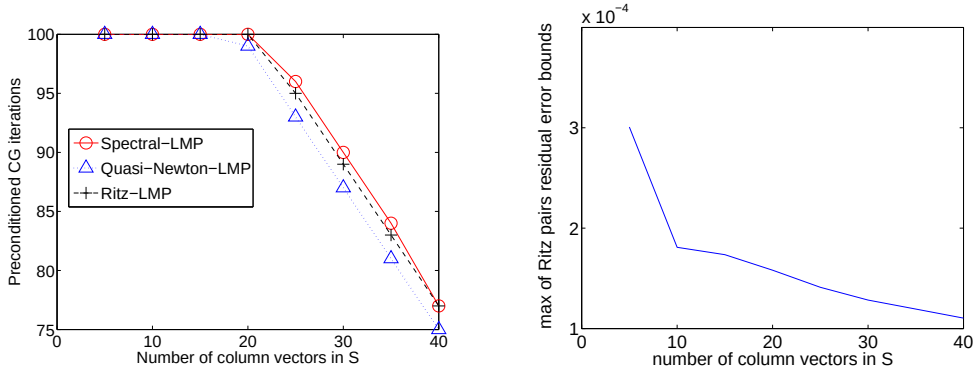

FIG. 3.3. *Performance of three LMPs versus number of columns of S using system of* `Matrix 3`

involved in Krylov methods, such as eigenvectors, Ritz vectors or descent directions, are considered.

A spectral analysis of the preconditioned matrix has been proposed that shows that the LMP is able to cluster at least k eigenvalues at 1. In addition, the behaviour of the eigenvalues of the preconditioned matrix is considered when A already has a cluster at 1. It is also shown that the eigenvalues of the preconditioned matrix enjoy interlacing properties with respect to the eigenvalues of the original matrix A. This spectral analysis has also been applied to the problem of finding a good scaling for the first-level preconditioner M of the LMP. The optimal scaling, that is available in closed form, appears to be closely related to the scaling strategy proposed for the initial inverse Hessian approximation H_0 in the BFGS algorithm (see [24]) .

The implementation of the LMP has also been considered for large scale problems, and an algorithm where applying the preconditioner to a vector costs $8kn$ floating point operations has been proposed; such a cost is consistent with optimized limited memory quasi-Newton methods that are already known. Experiments, reported in [29], show the superiority of the Ritz-LMP over the spectral-LMP and the quasi-Newton-LMP in a large scale ocean data assimilation context (with more then 10^6 unknowns).

Acknowledgments. The authors are indebted to Jorge Nocedal for his useful comments.

REFERENCES

[1] O. Axelsson. Optimal preconditioners based on rate of convergence estimates for the conjugate gradient. *Numerical Functional Analysis and Optimization*, 12:277–302, 2001.
[2] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, USA, 1996.
[3] Å. Björck, T. Elfving, and Z. Strakos. Stability of conjugate gradient and Lanczos methods for linear least squares problems. *SIAM Journal on Matrix Analysis and Applications*, 19(21):720–736, 1998.
[4] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming*, 63:129–156, 1994.
[5] B. Carpentieri, I. S. Duff, and L. Giraud. A class of spectral two-level preconditioners. *SIAM Journal on Scientific Computing*, 25(2):749–765, 2003.
[6] K. Chen. *Matrix Preconditioning Techniques and Applications*. SIAM, Philadelphia, USA, 2005.
[7] J. Derber and F. Bouttier. A reformulation of the background error covariance in the ECMWF global data assimilation system. *Tellus*, 51(2):195–221, 1999.
[8] M. Fisher. Minimization algorithms for variational data assimilation. In *Recent Developments in Numerical Methods for Atmospheric Modelling*, pages 364–385. ECMWF, 1998.
[9] M. Fisher, J. Nocedal, Y. Trémolet, and S.J. Wright. Data assimilation in weather forcasting: A case study in PDE–constrained optimization. Technical report, Optimization Technology Center, 2007.
[10] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.
[11] L. Giraud and S. Gratton. On the sensitivity of some spectral preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 27:1089–1105, 2006.
[12] L. Giraud, S. Gratton, and E. Martin. Incremental spectral preconditioners for sequences of linear systems. *Applied Numerical Mathematics*, 57(11-12):1164–1180, 2007.
[13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.
[14] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of the National Bureau of Standards*, 49:409–436, 1952.
[15] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, England, 1999.
[16] Ilse C. F. Ipsen and Carl D. Meyer. The idea behind Krylov methods. *American Mathematical Monthly*, 105(10):889–899, 1998.

[17] A.C. Lorenc. Modelling of error covariances by 4D-VAR data assimilation. *Q. J. R. Meteorol. Soc.*, 129:3167–3182, 2003.
[18] Jan Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.
[19] C. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, U.S.A., 2000.
[20] J.L. Morales and J. Nocedal. Automatic preconditioning by limited memory quasi-newton updating. *SIAM Journal on Optimization*, 10(3–4):1079–1096, 2000.
[21] R. Nabben and C. Vuik. A comparison of deflation and coarse grid correction applied to porous media flow. *SIAM J. Numer. Anal.*, 42(4):1631–1647, 2004.
[22] R. Nabben and C. Vuik. A comparison of deflation and the balancing preconditioner. *SIAM J. Sci. Comput.*, 27:1742–1759, 2006.
[23] L. Nazareth. A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms. *SIAM Journal on Numerical Analysis*, 16:794–800, 1979.
[24] J. Nocedal and S. J. Wright. *Numerical Optimization*. Series in Operations Research. Springer Verlag, Heidelberg, Berlin, New York, 1999.
[25] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8:43–71, 1982.
[26] M. J. D. Powell. Methods for nonlinear constraints in optimization calculations. In A. Iserles and M. J. D. Powell, editors, *The State of the Art in Numerical Analysis*, pages 325–358, Oxford, England, 1987. Oxford University Press.
[27] R. B. Schnabel. Quasi-newton methods using multiple secant equations. Technical Report CU-CS-247-83, Department of Computer Sciences, University of Colorado, Boulder, USA, 1983.
[28] J. Tshimanga. *On a class of limited memory preconditioners for large-scale nonlinear least-squares problems (with application to variational ocean data assimilation)*. PhD thesis, Department of Mathematics, University of Namur, Namur, Belgium, 2007.
[29] J. Tshimanga, S. Gratton, A. Weaver, and A. Sartenaer. Limited-memory preconditioners with application to incremental four-dimensional variational data assimilation. *Quarterly Journal of the royal Meteorological Society*, 134:751–769, 2008.
[30] H. Waisman, J. Fish, R. S. Tuminaro, and J. Shadid. The generalized global basis method. *International Journal on Numerical Methods in Engineering*, 61(8):1243–1269, 2004.
[31] A.T. Weaver, C. Deltel, E. Machu, S. Ricci, and N. Daget. A multivariate balance operator for variational ocean data assimilation. *Q. J. R. Meteorol. Soc.*, 131:3605–3625, 2005.