SIAM J. OPTIM.
Vol. 30, No. 3, pp. 1954–1979

A CLASS OF APPROXIMATE INVERSE PRECONDITIONERS BASED ON KRYLOV-SUBSPACE METHODS FOR LARGE-SCALE NONCONVEX OPTIMIZATION*

MEHIDDIN AL-BAALI†, ANDREA CALICIOTTI‡, GIOVANNI FASANO§, AND MASSIMO ROMA‡

Abstract. We introduce a class of positive definite preconditioners for the solution of large symmetric indefinite linear systems or sequences of such systems, in optimization frameworks. The preconditioners are iteratively constructed by collecting information on a reduced eigenspace of the indefinite matrix by means of a Krylov-subspace solver. A spectral analysis of the preconditioned matrix shows the clustering of some eigenvalues and possibly the nonexpansion of its spectrum. Extensive numerical experimentation is carried out on standard difficult linear systems and by embedding the class of preconditioners within truncated Newton methods for large-scale unconstrained optimization (the issue of major interest). Although the Krylov-based method may provide modest information on matrix eigenspaces, the results obtained show that the proposed preconditioners lead to substantial improvements in terms of efficiency and robustness, particularly on very large nonconvex problems.

Key words. large indefinite linear systems, Krylov-subspace methods, preconditioning, conjugate gradient methods, large-scale nonconvex optimization

AMS subject classifications. 90C06, 90C30

DOI. 10.1137/19M1256907

1. Introduction.

Consider solving the symmetric indefinite linear system

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. It is assumed that n is large and no sparsity pattern of A is given. The importance of efficiently solving such linear systems is well known. It is commonly claimed that "In scientific computing most computational time is spent in solving systems of linear equations" [64]. We propose a class of preconditioners for solving (1.1) and also focus on the solution of sequences of systems $Ax = b_i$ or $A_i x = b_i$, $i = 1, \ldots, N$, with particular reference to the framework of the truncated Newton method for large-scale unconstrained optimization.

When the systems are large, direct methods are not usually applicable, and iterative methods become mandatory. The most commonly used methods belong to the Krylov-subspace class [37, 64]. In exact arithmetic, this class converges in a finite number of steps. In practice, however, it is recognized that *preconditioning* is an essential tool for obtaining an efficient solver [5, 41, 63], because the aim of any preconditioning strategy is to improve the spectral properties of the system. Indeed, the rate of convergence of preconditioned Krylov-based methods is strongly affected

*Received by the editors April 17, 2019; accepted for publication (in revised form) May 18, 2020; published electronically July 29, 2020.
https://doi.org/10.1137/19M1256907
Funding: The third author wishes to thank the INdAM-GNCS Instituto Nazionale di Alta Matematica, Gruppo Nazionale per il Calcolo Scientifico, and the CNR-INSEAN Marine Technology Research Institute for their support.

†Department of Mathematics, Sultan Qaboos University, Muscat, Oman (albaali@squ.edu.om).
‡Dipartimento di Ingegneria Informatica, Automatica e Gestionale "A. Ruberti," SAPIENZA, Università di Roma, Roma, Italy (caliciotti@diag.uniroma1.it, roma@diag.uniroma1.it).
§Dipartimento di Management, Università Ca' Foscari, Venezia, Italy (fasano@unive.it).

by the distribution of the eigenvalues of the preconditioned matrix. For large-scale systems, a preconditioning strategy must be *matrix-free*. Thus, a matrix cannot be stored, but the matrix-vector product Ay can be provided by means of some available routine, and a preconditioning matrix must be handled efficiently also. (See [5] for a survey on preconditioning techniques for large linear systems.)

Additional safeguard is needed whenever A is indefinite. In this case, the conjugate gradient method (CG) may fail to solve the linear system. Moreover, for indefinite systems, the relation between the performance of the iterative method and the spectral properties of the preconditioned matrix is far from obvious. To overcome the first drawback, some methods have been proposed such as MINRES and SYMMLQ [56], GMRES [59], and the recently proposed planar-CG methods (see [28, 29, 30] and the references therein).

Of course, matrix-free preconditioners are the only ones that can be used in the framework of truncated Newton methods for large-scale optimization. These methods (also called *Newton–Krylov methods*) are commonly adopted methods for minimizing a twice continuously differentiable function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ [53]. They are based on two nested loops: the outer and the inner iterations. On each outer iteration k, the search direction d_k is computed by approximately solving the linear system $\nabla^2 f(x_k)d = -\nabla f(x_k)$ for d, by means of a Krylov-subspace method. For general nonconvex functions with the Hessian $\nabla^2 f(x_k)$ possibly indefinite, a sequence of symmetric indefinite linear systems must be solved. Here, the importance of preconditioning strategies is well known since the paper of Nash [52]. The interest in the truncated Newton methods is still very strong, for example, in the framework of optimization methods for large-scale machine learning [11, section 6.1]. The recent paper [45] shows the importance of preconditioners within the truncated Newton methods for large-scale linear classification problems.

Interesting classes of preconditioners, particularly suited for solving a sequence of symmetric linear systems, have been proposed in [7, 9, 10, 26, 27, 39, 40, 49, 50]. In particular, [40] introduces a class of limited memory preconditioners (LMP) for solving positive definite linear systems with multiple right-hand sides, extended to the indefinite case in [39]. An automatic preconditioning strategy (PREQN) is proposed in [50] and tested within a Hessian-free Newton method. Both LMP and PREQN are based on quasi-Newton limited memory L-BFGS Hessian approximation [54]. Moreover, [49] proposes a preconditioning strategy that can be applied to particular saddle-point problems, while [7] deals with efficient preconditioning techniques for sequences of KKT linear systems. We also recall the matrix-free approximate inverse preconditioners in [33, 34] (which represent the forerunners of the preconditioners proposed in this paper), the DSPREC preconditioner [57] based on a dynamic scaling, and the band preconditioners of [23, 47].

Specific mention is mandatory for the approaches known as *deflation preconditioning* [61] and *spectral preconditioning* [36]. Both methods are popular for unsymmetric systems and rely on a Krylov-based procedure, which exploits some spectral properties of A, in order to build a preconditioner M. The structure of M is such that the spectrum of MA includes several eigenvalues equal to one. These approaches typically require Krylov-based methods (e.g., GMRES in the unsymmetric case) to explore, after h iterations, an h-dimensional subspace that is invariant under a linear transformation given by the matrix A. If the columns of $V_h \in \mathbb{R}^{n\times h}$ span this subspace, the matrix[1]

[1] The superscript H indicates the conjugate transpose of a matrix.

$$M = I - V_h V_h^H + V_h \left(V_h^H A V_h\right)^{-1} V_h^H \tag{1.2}$$

can be used as preconditioner for the unsymmetric linear system, as h eigenvalues of MA are provably equal to one (see section 4 of [61]).

Here we take advantage of the literature on deflation preconditioning and spectral preconditioning in order to

- reinforce this last result, by building preconditioners whose structure resembles (1.2), but *without requiring any basis of an invariant subspace for* A;
- exploit the structure of the proposed preconditioners, so that some information on the remaining $n - h$ eigenvalues can be provided, without resorting to the introduction of the δ-pseudospectrum of an auxiliary matrix (compare with [61, Theorem 4.1]).

In particular, we propose a class of (matrix-free) positive definite preconditioners to be used for the solution of large indefinite symmetric linear systems (1.1). The preconditioners are constructed by using information gained from any Krylov-subspace method. They are iteratively built by using (but not performing) an implicit factorization of the system matrix, obtained as a byproduct of a Krylov-subspace method. We draw our inspiration from the class of approximate inverse (AINV) preconditioners (see [6], section 5.1.2 of [5], and [9, 10]), because ideally a good preconditioner should mimic A^{-1} and the AINV preconditioners usually work well in practice.

The proposed preconditioners (which we call AINV$\mathcal{K}$ class) represent a significant enhancement relative to those in [33] and [34]. The preconditioners introduced in [33] are characterized by a simpler structure, where CG is used so that early termination might occur. Conversely, those presented in [34] have a structure analogous to the class we study here but with the following limiting features: (i) the approach of [34] only copes with the positive definite case; (ii) the spectral analysis performed therein is different, as it only refers to singular values. Unlike [34], the AINV$\mathcal{K}$ class applies to indefinite problems and has proved to be effective in practical solution of large problems. We report a refined theoretical spectral analysis for the AINV$\mathcal{K}$ class of preconditioners, in terms of the eigenvalues and condition number of the preconditioned matrix. In particular, we analyze the clustering of eigenvalues and possibly the nonexpansion of the spectrum of the preconditioned matrix.

The AINV$\mathcal{K}$ preconditioners have been paired here with the Krylov-based solver SYMMBK [19] and numerically tested. First, experimentation has been performed on standard difficult indefinite linear systems. Then, we focused on using AINV$\mathcal{K}$ within truncated Newton methods to improve the efficiency and robustness of such methods for large-scale nonconvex unconstrained optimization. The results of extensive numerical experimentation seem to confirm this aim. (For the sake of brevity, we report them only for nonlinear optimization problems, but for further material see [31, 32].) In particular, by embedding AINV$\mathcal{K}$ within a linesearch-based implementation of a truncated Newton method, we show good performance of the method on a set of large test problems consisting of all CUTEst [38] unconstrained problems whose dimension ranges from 10^3 to 10^6. Theoretical and numerical comparison with the LMP preconditioners have been carried out here, showing a preference for AINV$\mathcal{K}$. LMP proved to be efficient on different classes of positive definite linear systems, and for the AINV$\mathcal{K}$ preconditioners a complete analysis in the indefinite case shows analogous efficiency on nonconvex optimization problems.

The paper is organized as follows. Section 2 introduces some preliminaries concerning Krylov-subspace methods and symmetric indefinite factorizations. In section 3, the proposed AINV$\mathcal{K}$ class of preconditioners is introduced, and section 4

studies the relation between the AINV$\mathcal{K}$ and LMP classes. The spectral properties of the AINV$\mathcal{K}$ class are investigated in section 5, while section 6 deals with its computational cost. The results of extensive numerical experimentation are reported in section 7.

Let $\lambda_{\min}(A)$, $\lambda_i(A)$, and $\lambda_{\max}(A)$ denote the smallest, ith, and largest eigenvalues of a matrix A and $\kappa(A)$ its condition number, $\|v\|$ be the Euclidean norm of a vector v, and I_n be the identity matrix of order $n \geq 1$. Finally, $e_h \in \mathbb{R}^n$ is the hth unit vector.

2. Preliminaries. The CG method and the Lanczos process (the latter coupled with a procedure for solving a tridiagonal system) are usually adopted to tackle system (1.1). As is well known, they are equivalent (in exact arithmetic) as long as A is positive definite, and the relationship between them is well established (see [37, 62, 64] and [22, Chapter 5]). Indeed, both methods compute different bases for the same Krylov subspace. However, CG, though cheaper, presents a major drawback: it does not cope with the indefinite case. We now recall some details on Krylov-subspace methods that are used later.

2.1. Krylov-subspace methods. Consider any Lanczos-based Krylov-subspace method for solving symmetric linear system (1.1), and suppose that h steps of the method are performed with $h \leq n-1$. Then orthonormal vectors $u_1, \ldots, u_{h+1}$ are generated such that

$$AR_h = R_h T_h + \rho_{h+1} u_{h+1} e_h^T, \tag{2.1}$$

where $R_h = [u_1 \cdots u_h] \in \mathbb{R}^{n \times h}$ and $T_h \in \mathbb{R}^{h \times h}$ is a certain tridiagonal matrix (see (2.2) below). If the scalar $\rho_{h+1} \neq 0$ (i.e., the Krylov-subspace method has not yet come to convergence), then the columns of R_h are not invariant with respect to A.

To have a better intuition on the generation of the orthonormal vectors u_i, in the case of the two commonest Krylov-subspace methods, they coincide with the *Lanczos vectors* for the Lanczos process and with the *normalized residuals* for the CG-based methods. These vectors are used for transforming system (1.1) into a tridiagonal one, and the methods differ only in the way the resulting tridiagonal system is solved by factorization. Premultiplying (2.1) by R_h^T, and assuming orthonormality, we obtain the important explicit relationship

$$T_h = R_h^T A R_h. \tag{2.2}$$

2.2. Factorization of the tridiagonal matrix. If the matrix A is positive definite, (2.2) implies that T_h is also positive definite, so that it can be decomposed as

$$T_h = L_h D_h L_h^T, \tag{2.3}$$

where $L_h \in \mathbb{R}^{h \times h}$ is a unit lower bidiagonal matrix and $D_h \in \mathbb{R}^{h \times h}$ is a diagonal matrix [37]. Since T_{h+1} has T_h as an $h \times h$ diagonal submatrix, the decomposition of T_{h+1} can be easily obtained from that of T_h.

When A is indefinite, the decomposition (2.3) may not exist or may be numerically unstable. Therefore, methods for solving large indefinite linear systems should be based on reliable *symmetric indefinite factorizations* of the tridiagonal matrix T_h. One possibility is the Bunch and Kaufman decomposition [13],

$$T_h = L_h B_h L_h^T, \tag{2.4}$$

where $B_h \in \mathbb{R}^{h\times h}$ is block diagonal with each block of dimension at most two. Its distinguishing feature is to perform 2×2 pivot in place of a single element, 1×1, whenever in T_{h+1} a small pivot would cause instability of the decomposition. To maintain numerical stability, a suitable rule due to Bunch [12] is adopted for choosing a 1×1 or 2×2 pivot, avoiding growth in the matrix elements (see [14, 15, 42] and [43, Chapter 11]). A recursive procedure based on the generation of a sequence of tridiagonal matrices of order h was proposed in [19], and the resulting algorithm is known as SYMMBK (see also [48]). If at iteration h a 1×1 pivot is adopted, the next iterate will be x_{h+1}, while if a 2×2 pivot takes place, the next iterate will be x_{h+2}.

Other factorizations (when T_h is indefinite), used by Paige and Saunders [56], can be found in SYMMLQ, which computes a numerically stable factorization $T_h = L_h Q_h$ (with L_h lower triangular and Q_h orthogonal), and MINRES, which uses QR factors of a suitable tridiagonal matrix.

2.3. Subproblems definitions for Lanczos-based solvers. It is important to highlight that, at each iteration of a Lanczos-based solver, a subproblem is solved to generate iterates of the form $x_h = R_h y_h$ for some $y_h \in \mathbb{R}^h$, which approximate a solution of (1.1), without computing y_h explicitly. Indeed, setting $b = \rho_1 u_1$, since (2.1) can be rewritten as $AR_h = R_{h+1}T_{h+1,h}$, where

$$R_{h+1} = [R_h \quad u_{h+1}], \qquad T_{h+1,h} = \begin{bmatrix} T_h \\ \rho_{h+1} e_h^T \end{bmatrix}, \tag{2.5}$$

we have

$$r_h = b - Ax_h = \rho_1 u_1 - R_{h+1}T_{h+1,h}y_h = R_{h+1}\left(\rho_1 e_1 - T_{h+1,h}y_h\right).$$

Therefore, Lanczos-based solvers aim to make $\rho_1 e_1 - T_{h+1,h}y_h$ small. CG considers the first h equations and determines $y_h \in \mathbb{R}^h$ from the subproblem $T_h y = \rho_1 e_1$. SYMMLQ focuses on the first $h-1$ equations and solves $\min_y \|y\|$ such that $T_{h,h-1}^T y = \rho_1 e_1$ by applying the LQ factorization. MINRES solves $\min_y \|T_{h+1,h}y - \rho_1 e_1\|$ by using the QR factorization. (See [60, Table 3.1] and also [20, 21].) SYMMBK uses the same subproblem $T_h y = \rho_1 e_1$ as CG except when it would be numerically unsafe to compute x_h. In this case, y_h and x_h are undefined, but the factorization (2.4) exists for T_{h+1}, and the next subproblem $T_{h+1}y = \rho_1 e_1$ is used to compute x_{h+1}. This strategy can be viewed as a look-ahead of 1.

3. The AINV$\mathcal{K}$ class of preconditioners. Our class of preconditioners, also based on deflation preconditioning, extends that in [34] to the indefinite case. Considering any Lanczos-based Krylov-subspace method for solving the symmetric system (1.1), we state the following standard assumption.

Assumption 3.1. Consider relation (2.1). The tridiagonal matrix T_h is irreducible, and decomposition (2.4) exists, where B_h is either 1×1 or 2×2 block diagonal and L_h is unit lower bidiagonal.

This assumption is not restrictive because the factorization (2.4) always exists, even without permutation of rows and columns of T_h (recall SYMMBK in subsection 2.2). However, the numerical stability is strictly dependent on the choice of the pivoting strategy (see [42] for details on the existence and the stability of such decomposition). Moreover, since T_h is assumed to be irreducible, it cannot have multiple eigenvalues.

For convenience, we let

$$B_h = \text{blkdiag}_{1\le j\le m}\{E_h^j\},$$

where either $E_h^j \in \mathbb{R}$ or $E_h^j \in \mathbb{R}^{2\times 2}$ for $j = 1, \ldots, m$. For the latter case we find the eigendecomposition of each 2×2 diagonal block:

$$E_h^j = U_h^j D_h^j (U_h^j)^T,$$

where $D_h^j = \text{diag}\{d_h^{j_1}, d_h^{j_2}\}$ with $d_h^{j_1}$ and $d_h^{j_2}$ being the eigenvalues of E_h^j, and the columns of U_h^j are orthogonal eigenvectors ($(U_h^j)^T U_h^j = I$). For blocks consisting of one element, we note that $E_h^j = d_h^{j_1}$ and $U_h^j \equiv 1$. Taking the absolute value of all the diagonal elements $d_h^{j_i}$, $i \in \{1, 2\}$ (see also [35]), we define the matrix

$$|B_h| \equiv \text{blkdiag}_{1\le j\le m} \left\{ U_h^j |D_h^j| (U_h^j)^T \right\} = U_h |D_h| U_h^T,$$

where

$$U_h = \text{blkdiag}_{1\le j\le m} \left\{ U_h^j \right\}, \qquad |D_h| = \text{blkdiag}_{1\le j\le m} \left\{ |D_h^j| \right\}, \tag{3.1}$$

$$|D_h^j| \in \left\{ |d_h^{j_1}|, \text{diag}\{|d_h^{j_1}|, |d_h^{j_2}|\} \right\}. \tag{3.2}$$

Similar to (2.4), we define

$$|T_h| \equiv L_h |B_h| L_h^T \tag{3.3}$$

which has the following properties: (i) $|T_h|$ is positive definite for any h, and $|T_h| = T_h$ if T_h is positive definite; (ii) $|T_h| = L_h U_h |D_h| U_h^T L_h^T$; (iii) $|T_h|^T = |T_h|$ and

$$T_h |T_h|^{-1} = \left[|T_h|^{-1} T_h \right]^T = L_h \widehat{I}_h L_h^{-1}, \tag{3.4}$$

where $\widehat{I}_h = B_h |B_h|^{-1}$ is block-diagonal, including at most 2×2 diagonal blocks, with eigenvalues equal to either 1 or -1.

We now introduce the AINV$\mathcal{K}$ class of preconditioners. For $h \ge 1$, we define the preconditioner matrix

$$M_h(a, W_h) \equiv \left[I_n - R_{h+1} R_{h+1}^T \right] + \; R_{h+1} \mathcal{T}_h^{-1} R_{h+1}^T, \tag{3.5}$$

where

$$\mathcal{T}_h = \begin{bmatrix} |\widehat{T}_h| & a e_h \\ a e_h^T & 1 \end{bmatrix}, \tag{3.6}$$

$W_h = \text{diag}\{w_1^2, \ldots, w_h^2\}$, a and $\{w_i\}$ are scalars, R_{h+1} is defined in (2.5), and

$$|\widehat{T}_h| \equiv L_h U_h \left(W_h |D_h| \right) U_h^T L_h^T. \tag{3.7}$$

We note that the matrix in the square brackets of (3.5) can be written as

$$\hat{R}_{h+1} \hat{R}_{h+1}^T = I_n - R_{h+1} R_{h+1}^T, \tag{3.8}$$

where the columns of $\hat{R}_{h+1} \in \mathbb{R}^{n\times(n-h-1)}$ are orthonormal. Observe that $W_h |D_h|$ is diagonal and $\widehat{T}_h = T_h$ if $W_h = I_h$. In what follows, to simplify the notation, we denote $|D_h|$ by $\mathcal{D}_h$.

The structure of AINV$\mathcal{K}$ in (3.5) can be described as follows: the first term represents a projector onto the subspace orthogonal to the range of R_{h+1}, while the

second one may be interpreted as a suitable approximation of A^{-1} (exploiting the *spectral theorem*) over the Krylov subspace $\text{span}\{u_1, \dots, u_{h+1}\}$. Moreover, two (user-dependent) parameters are present: the scalar a and the matrix W_h. The effect of W_h is to emphasize the information about A collected by the Krylov-subspace method, because it acts as a scaling of the matrix $\mathcal{D}_h$. For the simple choice $W_h = I_h$, $M_h(a, I_h)$ can be regarded as an approximate inverse preconditioner [58], without scaling. However, numerical evidence suggests other choices for W_h (see section 7).

The AINV$\mathcal{K}$ preconditioners (3.5) extend those proposed in [33, 34]. Indeed, the preconditioners proposed in [33] are merely built using the CG method and can be obtained as a particular case of (3.5) by setting $a = 0$ and $W_h = I_h$. In [34] the class of preconditioners proposed is characterized by $a \neq 0$ and $W_h = \nu^2 I_h$, where ν is a scalar, but only the positive definite case is considered, implying $|\widehat{T}_h| = \widehat{T}_h$. Moreover, the spectral analysis performed in [34] is only in terms of singular values of the (unsymmetric) preconditioned matrix, which may not yield direct information on the convergence properties of a Krylov-subspace method. Recalling that $\rho_{h+1} \neq 0$ in (2.1) implies that the subspace $\text{span}\{u_1, \dots, u_h\}$ is not invariant, we consider a more general framework than [3] and most of the approaches based on deflation preconditioning.

4. Relation between the AINV$\mathcal{K}$ and LMP classes. In this section, we highlight the relation between AINV$\mathcal{K}$ and LMP in [40]. The LMP preconditioners use the L-BFGS quasi-Newton updates to build an approximate inverse preconditioner for (1.1), when A is positive definite. An extension of LMP to the indefinite case has been proposed more recently in [39], where a theoretical analysis of the Ritz-LMP variant is also reported. Here, the extended LMP preconditioners are themselves possibly indefinite (see [39, section 3.3]), so that some additional care is mandatory when used in specific contexts.

In principle, since the L-BFGS update is used in place of Krylov-subspace methods, the idea behind LMP is different from our proposal. However, recalling the relationship between the directions computed by the L-BFGS and CG methods (see also [62]) when A is positive definite, it is not surprising that the two approaches show great similarities. In this case we can obtain a formal complete relationship between AINV$\mathcal{K}$ and LMP. Conversely, on indefinite linear systems, the two proposals differ greatly, as the AINV$\mathcal{K}$ preconditioners are always positive definite for suitable values of a in (3.5), while LMP preconditioners might be indefinite. Thus, in the indefinite case a full comparison seems hardly possible. With

$$V_h \equiv I_n - P_h(P_h^T A P_h)^{-1} P_h^T A,$$

the LMP class is characterized by

$$H_h = V_h H_0 V_h^T + P_h(P_h^T A P_h)^{-1} P_h^T, \tag{4.1}$$

where $P_h = [p_1 \ \cdots \ p_h]$ is any $n \times h$ matrix of rank $h \leq n$ and H_0 is a symmetric positive definite matrix. Note that this formula defines multiple BFGS updates and satisfies the so-called generalized quasi-Newton condition $H_h A P_h = P_h$ (it is proposed in [16] with further details in [1] and the references therein). No assumption is required on the column vectors p_j apart from their linear independence.

Following the reasoning in section 2 of [40], where the expression of H_h is derived from the fact that the columns of P_h are A-conjugate directions, we can take $p_1, \dots, p_h$ as the conjugate directions computed by CG (or L-BFGS; see [55] for the relation

between CG and L-BFGS). Observe that H_h is invariant under scaling of the vectors $p_1, \ldots, p_h$ (see [40, Theorem 3.1]). Thus, if $r_1, \ldots, r_h$ are the corresponding residuals computed by CG up to step h, it follows that the norm of these residuals is bounded away from zero, so that we redefine

$$P_h = \left[\frac{p_1}{\|r_1\|} \ \cdots \ \frac{p_h}{\|r_h\|} \right]. \tag{4.2}$$

Thanks to the relation between the residuals and conjugate directions in CG (see also [62]), we have $P_h = R_h L_h^{-T}$, where

$$R_h = \left[\frac{r_1}{\|r_1\|} \cdots \frac{r_h}{\|r_h\|} \right], \qquad L_h = \begin{bmatrix} 1 & & & & \\ -\frac{\|r_2\|}{\|r_1\|} & 1 & & & \\ & -\frac{\|r_3\|}{\|r_2\|} & 1 & & \\ & & \ddots & \ddots & \\ & & & -\frac{\|r_h\|}{\|r_{h-1}\|} & 1 \end{bmatrix}. \tag{4.3}$$

Because

$$\begin{aligned} P_h \left(P_h^T A P_h\right)^{-1} P_h^T A &= R_h L_h^{-T} \left[L_h^{-1} R_h^T A R_h L_h^{-T}\right]^{-1} L_h^{-1} R_h^T A \\ &= R_h T_h^{-1} \left[R_h T_h + \rho_{h+1} \frac{r_{h+1}}{\|r_{h+1}\|} e_h^T \right]^T \\ &= R_h R_h^T + \rho_{h+1} R_h T_h^{-1} e_h \frac{r_{h+1}^T}{\|r_{h+1}\|} \equiv Z_h \end{aligned}$$

and $T_h = R_h^T A R_h$, relation (4.1) becomes

$$H_h = (I_n - Z_h)\, H_0\, (I_n - Z_h)^T + \ R_h T_h^{-1} R_h^T. \tag{4.4}$$

With (3.5), (3.7), $W_h = I_h$, and $a = 0$, this implies that H_h and $M_h(a, I_h)$ are generally different. However, recalling that $(I_n - R_h R_h^T)$ is idempotent, setting $\rho_{h+1} = 0$ and $H_0 = I_n$, we obtain $H_h \equiv M_h(0, I_h)$. Thus, if A is positive definite, when the L-BFGS update is complete (i.e., when $\rho_{h+1} = 0$ in (2.1)), the LMP class with $H_0 = I_n$ and the proposed class (with $W_h = I_h$ and $a = 0$) are coincident in exact arithmetic. When $\rho_{h+1} \neq 0$, there is no chance that the two preconditioners coincide.

Remark 4.1. It is possible to show that if CG is used as the Krylov-subspace method for solving (1.1) and for building both the LMP and AINV$\mathcal{K}$ preconditioners, then the two preconditioners simply differ by a rank-2 matrix.

Finally, we acknowledge the relevant role of the *first-level preconditioner* H_0 in the LMP class, which has no immediate counterpart in our proposal. In fact, such a matrix H_0 can be very effective when properly set, in applications where additional problem information is known.

5. Spectral properties of the preconditioned matrix. The spectral properties of the preconditioned matrix $M_h(a, W_h)A$ may strongly affect the behavior of a Krylov-subspace method used for solving the preconditioned linear system. Therefore, we study the spectrum and give indications on the condition number of the preconditioned matrix $M_h(a, W_h)A$.

Given the orthogonal matrix

$$\mathcal{N} = [R_{h+1}\ \hat{R}_{h+1}], \qquad h \le n-1, \tag{5.1}$$

where $\hat{R}_{h+1}$ is defined in (3.8), we may write (3.5) as

$$M_h(a, W_h) = \mathcal{N} \begin{bmatrix} \mathcal{T}_h^{-1} & 0 \\ 0 & I_{n-(h+1)} \end{bmatrix} \mathcal{N}^T, \qquad h \le n-1, \tag{5.2}$$

where $\mathcal{T}_h$ is defined in (3.6). Recalling that L_h is unit lower bidiagonal ($L_h^{-1} e_h = e_h$ and $L_h e_h = e_h$), setting

$$\Delta_h = 1 - a^2 e_h^T |\widehat{T}_h|^{-1} e_h, \tag{5.3}$$

using the identity

$$\mathcal{T}_h = \begin{bmatrix} I_h & 0 \\ a e_h^T |\widehat{T}_h|^{-1} & 1 \end{bmatrix} \begin{bmatrix} |\widehat{T}_h| & 0 \\ 0 & \Delta_h \end{bmatrix} \begin{bmatrix} I_h & a|\widehat{T}_h|^{-1} e_h \\ 0 & 1 \end{bmatrix} \tag{5.4}$$

and (3.7), and assuming $\Delta_h \neq 0$, we obtain the factorization

$$\mathcal{T}_h^{-1} = \widetilde{\mathcal{L}}_h^T \widetilde{\mathcal{L}}_h, \qquad \widetilde{\mathcal{L}}_h = \begin{bmatrix} W_h^{-1/2} \mathcal{D}_h^{-1/2} U_h^T L_h^{-1} & 0 \\ -a \Delta_h^{-1/2} e_h^T |\widehat{T}_h|^{-1} & \Delta_h^{-1/2} \end{bmatrix}. \tag{5.5}$$

The following theorem characterizes the spectrum of the preconditioned matrix, showing that like spectral preconditioners, the AINV$\mathcal{K}$ preconditioners (3.5) cluster a certain number of eigenvalues of the preconditioned matrix $M_h(a, W_h)A$. An important interlacing property is proved between the eigenvalues of $M_h(0, W_h)A$ and those of the indefinite matrix A.

THEOREM 5.1. *In* (1.1), *suppose Assumption* 3.1 *holds, consider the AINV$\mathcal{K}$ class of preconditioners* (3.5), *and assume* $\Delta_h \neq 0$. *Then*

(i) *if* $\Delta_h > 0$, *then* $M_h(a, W_h)$ *is positive definite;*
(ii) $M_h(a, W_h)A$ *has at least* $(h-2)$ *eigenvalues in* $\{\pm 1/w_1^2, \ldots, \pm 1/w_h^2\}$;
(iii) *if* B_h *in* (2.4) *is diagonal, then* $M_h(a, W_h)A$ *has at least* $(h-1)$ *eigenvalues in* $\{\pm 1/w_1^2, \ldots, \pm 1/w_h^2\}$;
(iv) *if* A *is positive definite, then* $M_h(a, W_h)A$ *has at least* $(h-1)$ *eigenvalues in* $\{1/w_1^2, \ldots, 1/w_h^2\}$;
(v) *if* A *is positive definite and* $a = w_h^2 \rho_{h+1}$, *then* $M_h(a, W_h)A$ *has at least* h *eigenvalues in* $\{1/w_1^2, \ldots, 1/w_h^2\}$;
(vi) *if* $a = 0$, *there exist* $n - h - 2$ *eigenvalues* $\lambda_i(M_h(0, W_h)A)$, $i \in \{1, \ldots, n\}$, *which do not coincide with the eigenvalues in items* (ii), (iii), (iv), *and* (v), *such that*

$$\lambda_{\min}(A) \le \lambda_i(M_h(0, W_h)A) \le \lambda_{\max}(A);$$

(vii) *let* $\rho_{\max} = \max\{|\rho_{h+1}|, |\rho_{h+2}|\}$, $\sigma_{\min} = \min_{1 \le i \le h}\{\sigma_i / w_i^2\}$, *and* $\sigma_{\max} = \max_{1 \le i \le h}\{\sigma_i / w_i^2\}$, *where* σ_i *is the* i*th eigenvalue of* $\hat{I}_h$ *in* (3.4). *If* $a = 0$, *then, for* $i = 1, \ldots, n$

$$\begin{aligned} \min\{\sigma_{\min}, \lambda_{\min}(A)\} + \mathcal{O}(\rho_{\max}) &\le \lambda_i(M_h(0, W_h)A) \\ &\le \max\{\sigma_{\max}, \lambda_{\max}(A)\} + \mathcal{O}(\rho_{\max}). \end{aligned} \tag{5.6}$$

Proof. Item (i) immediately follows from (5.2), (5.3), and (5.4). For (ii), letting

$$\hat{\mathcal{L}}_h = \begin{bmatrix} \widetilde{\mathcal{L}}_h^T & 0 \\ 0 & I_{n-(h+1)} \end{bmatrix}$$

we obtain (by (5.2) and (5.5))

$$M_h(a, W_h) = \mathcal{N}\hat{\mathcal{L}}_h\hat{\mathcal{L}}_h^T\mathcal{N}^T$$

and (by (2.1) and (2.2))

$$\begin{aligned} \mathcal{N}^T A\mathcal{N} &= \begin{bmatrix} R_h^T AR_h & R_h^T Au_{h+1} & R_h^T A\hat{R}_{h+1} \\ u_{h+1}^T AR_h & u_{h+1}^T Au_{h+1} & u_{h+1}^T A\hat{R}_{h+1} \\ \hat{R}_{h+1}^T AR_h & \hat{R}_{h+1}^T Au_{h+1} & \hat{R}_{h+1}^T A\hat{R}_{h+1} \end{bmatrix} \\ &= \begin{bmatrix} T_h & \rho_{h+1}e_h & 0 \\ \rho_{h+1}e_h^T & u_{h+1}^T Au_{h+1} & \rho_{h+2}e_1^T \\ 0 & \rho_{h+2}e_1 & \hat{R}_{h+1}^T A\hat{R}_{h+1} \end{bmatrix}, \end{aligned} \tag{5.7}$$

where ρ_{h+1} and ρ_{h+2} are defined in (2.1). Moreover, $B_h = U_h D_h U_h^T$ (by (3.2)) gives, for suitable scalars α_h, β_h,

$$W_h^{-1/2}\mathcal{D}_h^{-1/2}U_h^T L_h^{-1}e_h = W_h^{-1/2}\mathcal{D}_h^{-1/2}U_h^T e_h \;=\; \begin{cases} \alpha_h e_h & \text{if } U_h^m \in \mathbb{R}, \\ \alpha_h e_h + \beta_h e_{h-1} & \text{otherwise;} \end{cases}$$

$$\begin{aligned} T_h|\widehat{T}_h|^{-1}e_h &= L_h B_h L_h^T L_h^{-T} U_h \mathcal{D}_h^{-1} W_h^{-1} U_h^T L_h^{-1} e_h \\ &= L_h B_h U_h \mathcal{D}_h^{-1} W_h^{-1} U_h^T e_h = \begin{cases} \alpha_h e_h & \text{if } U_h^m \in \mathbb{R}, \\ \alpha_h e_h + \beta_h e_{h-1} & \text{otherwise;} \end{cases} \end{aligned}$$

$$e_h^T|\widehat{T}_h|^{-1}e_h = e_h^T L_h^{-T} U_h \mathcal{D}_h^{-1} W_h^{-1} U_h^T L_h^{-1} e_h = e_h^T U_h \mathcal{D}_h^{-1} W_h^{-1} U_h^T e_h;$$

$$\begin{aligned} |\widehat{T}_h|^{-1}e_h &= L_h^{-T} U_h W_h^{-1} \mathcal{D}_h^{-1} U_h^T L_h^{-1} e_h \\ &= L_h^{-T} U_h W_h^{-1} \mathcal{D}_h^{-1} U_h^T e_h = \begin{cases} \alpha_h e_h & \text{if } U_h^m \in \mathbb{R}, \\ \alpha_h e_h + \beta_h e_{h-1} & \text{otherwise.} \end{cases} \end{aligned}$$

Note that λ is an eigenvalue of $M_h(a, W_h)A$ if and only if it is an eigenvalue of $\hat{\mathcal{L}}_h^T\mathcal{N}^T A\mathcal{N}\hat{\mathcal{L}}_h$. Observe that

$$\hat{\mathcal{L}}_h^T\mathcal{N}^T A\mathcal{N}\hat{\mathcal{L}}_h = \begin{bmatrix} M_{1,1} & M_{1,2} & 0 \\ M_{1,2}^T & M_{2,2} & M_{2,3} \\ 0 & M_{2,3}^T & M_{3,3} \end{bmatrix}, \tag{5.8}$$

where

$$\begin{aligned} M_{1,1} &= \left[W_h^{-1/2}\mathcal{D}_h^{-1/2}U_h^T L_h^{-1}\right] T_h \left[L_h^{-T} U_h \mathcal{D}_h^{-1/2} W_h^{-1/2}\right] \\ &= W_h^{-1/2}\mathcal{D}_h^{-1/2} D_h \mathcal{D}_h^{-1/2} W_h^{-1/2} = \mathrm{diag}_{1\le i\le h}\{\pm 1/w_i^2\}, \end{aligned} \tag{5.9}$$

and for suitable scalar $\bar{\alpha}_h, \tilde{\alpha}_h, \bar{\beta}_h, \tilde{\beta}_h$,

$$M_{1,2} = W_h^{-1/2}\mathcal{D}_h^{-1/2}U_h^T L_h^{-1}\left(-a\Delta_h^{-1/2}T_h|\widehat{T}_h|^{-1}e_h + \rho_{h+1}\Delta_h^{-1/2}e_h\right)$$

$$= \begin{cases} W_h^{-1/2}\mathcal{D}_h^{-1/2}U_h^T L_h^{-1}(\bar{\alpha}_h e_h) = \tilde{\alpha}_h e_h & \text{if } U_h^m \in \mathbb{R}, \\ W_h^{-1/2}\mathcal{D}_h^{-1/2}U_h^T L_h^{-1}(\bar{\alpha}_h e_h + \bar{\beta}_h e_{h-1}) = \tilde{\alpha}_h e_h + \tilde{\beta}_h e_{h-1} & \text{otherwise,} \end{cases}$$

$$M_{2,2} = \Delta_h^{-1}\left(-ae_h^T|\widehat{T}_h|^{-1}T_h + \rho_{h+1}e_h^T\right)\left(-a|\widehat{T}_h|^{-1}e_h\right) +$$
$$\Delta_h^{-1}\left(-a\rho_{h+1}e_h^T|\widehat{T}_h|^{-1}e_h + u_{h+1}^T A u_{h+1}\right)$$
$$= \frac{1}{\Delta_h}\left(a^2 e_h^T|\widehat{T}_h|^{-1}T_h|\widehat{T}_h|^{-1}e_h - 2a\rho_{h+1}e_h^T|\widehat{T}_h|^{-1}e_h + u_{h+1}^T A u_{h+1}\right),$$

$$M_{2,3} = \rho_{h+2}\Delta_h^{-1/2}e_1^T,$$
$$M_{3,3} = \hat{R}_{h+1}^T A \hat{R}_{h+1}.$$

Thus, if $E_h^m \in \mathbb{R}$, $M_h(a, W_h)A$ has at least $h-1$ eigenvalues in $\{\pm 1/w_1^2, \ldots, \pm 1/w_h^2\}$ (corresponding to $h-1$ eigenvectors in the set $\{e_1, \ldots, e_h\}$); otherwise, $E_h^m \in \mathbb{R}^{2\times 2}$ so that the last property holds with $h-1$ replaced by $h-2$.

For (iii), we have $|T_h| = L_h|B_h|L_h^T$ with B_h diagonal and $E_h^m \in \mathbb{R}$. Thus, using the reasoning of (ii) we obtain the result.

For (iv), since A is positive definite, T_h is positive definite too and can be factorized as $T_h = L_h D_h L_h^T$, where D_h is diagonal. Hence, (iii) implies (iv).

Item (v) is a special case of (iv), and recalling that A positive definite implies that U_h in (3.1) is diagonal; by the choice $a = w_h^2\rho_{h+1}$, we see that $M_{1,2} = 0$ in (5.8), which is reduced to the matrix

$$\begin{bmatrix} M_{1,1} & 0 & 0 \\ 0 & M_{2,2} & M_{2,3} \\ 0 & M_{2,3}^T & M_{3,3} \end{bmatrix}.$$

Thus, $M_h(a, W_h)A$ has at least h eigenvalues in $\{1/w_1^2, \ldots, 1/w_h^2\}$ (corresponding to the eigenvectors $\{e_1, \ldots, e_h\}$).

For item (vi) $\Delta_h = 1$. By (5.8), along with the definition of $M_{2,3}$, we have that $\lambda_i(M_h(0, W_h)A) = \lambda_i(M_{3,3})$ is the eigenvalue associated with an eigenvector of $M_{3,3}$ that lies in the subspace $\text{span}\{e_2, \ldots, e_{n-h-1}\}$. We may choose $n-h-2$ orthogonal eigenvectors of $M_{3,3}$ in $\text{span}\{e_2, \ldots, e_{n-h-1}\}$, so that by Poincaré's separation theorem (see also Corollary 4.3.16 of [44]) we obtain (vi).

For item (vii), by a generalization of the monotonicity theorem (or Weyl's inequality) (see, e.g., Theorems 8.4.9 and 8.4.11 of [8]), we have for any symmetric matrices B and C that

$$\lambda_i(B) + \lambda_{\min}(C) \le \lambda_i(B + C) \le \lambda_i(B) + \lambda_{\max}(C).$$

Since $L_h^{-1}e_h = e_h$, when $a = 0$ (which yields $\Delta_h = 1$), the matrix $\hat{\mathcal{L}}_h^T\mathcal{N}^T A\mathcal{N}\hat{\mathcal{L}}_h$ in (5.8) is reduced to

$$\begin{bmatrix} M_{1,1} & \rho_{h+1}W_h^{-1/2}\mathcal{D}_h^{-1/2}U_h^T e_h & 0 \\ \rho_{h+1}e_h^T U_h \mathcal{D}_h^{-1/2}W_h^{-1/2} & u_{h+1}^T A u_{h+1} & \rho_{h+2}e_1^T \\ 0 & \rho_{h+2}e_1 & M_{3,3} \end{bmatrix} \equiv B + C,$$

where

$$B = \begin{bmatrix} M_{1,1} & 0 & 0 \\ 0 & u_{h+1}^T A u_{h+1} & 0 \\ 0 & 0 & M_{3,3} \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & \rho_{h+1} W_h^{-1/2} \mathcal{D}_h^{-1/2} U_h^T e_h & 0 \\ \rho_{h+1} e_h^T U_h \mathcal{D}_h^{-1/2} W_h^{-1/2} & 0 & \rho_{h+2} e_1^T \\ 0 & \rho_{h+2} e_1 & 0 \end{bmatrix}.$$

It follows from Gershgorin's circle theorem and observing that for some scalars α_h, β_h,

$$\rho_{h+1} W_h^{-1/2} \mathcal{D}_h^{-1/2} U_h^T e_h = \begin{cases} \alpha_h e_h & \text{if } U_h^m \in \mathbb{R}, \\ \alpha_h e_h + \beta_h e_{h-1} & \text{otherwise,} \end{cases}$$

we have from (5.9) and (3.4)

$$\begin{aligned} \min\left\{\sigma_{\min}, u_{h+1}^T A u_{h+1}, \lambda_{\min}(M_{3,3})\right\} + O\left(\rho_{\max}\right) &\le \lambda_i(M_h(0, W_h)A) \\ &\le \max\left\{\sigma_{\max}, u_{h+1}^T A u_{h+1}, \lambda_{\max}(M_{3,3})\right\} + O\left(\rho_{\max}\right). \end{aligned}$$

By Poincaré's separation theorem the ith eigenvalue of $\hat{R}_{h+1}^T A \hat{R}_{h+1}$ satisfies

$$\lambda_{\min}(A) \le \lambda_i(M_{3,3}) \le \lambda_{\max}(A).$$

Since u_{h+1} has unit norm, (5.7) implies

$$\lambda_{\min}(A) \le u_{h+1}^T A u_{h+1} \le \lambda_{\max}(A).$$

Therefore (5.6) holds. ∎

Of course, $\hat{R}_{h+1}$ in the proof of Theorem 5.1 always exists such that $\mathcal{N}$ in (5.1) is orthogonal. However, $\hat{R}_{h+1}$ is neither built nor used, and it is introduced only for theoretical purposes.

It is worth comparing the results in Theorem 5.1 with similar results for the preconditioners of the LMP class in [40], concerning clustering of eigenvalues and nonexpansion of the spectrum of the preconditioned matrix. First note that the results in Theorem 5.1 hold when T_h is positive definite or indefinite. Theorem 5.1 is more general than Lemma 3.3 and Theorem 3.4 of [40], except the following: when A is positive definite, the result in item (vii) is weaker than that in [40], because of the presence of $\mathcal{O}(\rho_{\max})$ in (5.6). Also observe that the extension of the LMP class to symmetric indefinite systems in [39] leads to weaker results than those in Theorem 5.1. One specific reason for this conclusion relies on the fact that while the AINV$\mathcal{K}$ preconditioners are always positive definite (see (i) of Theorem 5.1), on indefinite linear systems LMP preconditioners can be possibly indefinite.

We emphasize that (5.6) can be simplified when A is positive definite as follows.

PROPOSITION 5.2. *Assume A is positive definite, and consider the AINV$\mathcal{K}$ preconditioners* (3.5) *with $a = 0$ and $W_h = I_h$. Let $\sigma_{\min} = \min_{1 \le i \le h}\{\sigma_i / w_i^2\}$, where σ_i is the ith eigenvalue of $\hat{I}_h$ in* (3.4). *Also let $\tau = \|w\|^2 / w^T A w$, where w is any nonzero vector. Then, the condition number of the preconditioned matrix $M_h(0, I_h)\tilde{A}$, where $\tilde{A} = \tau \sigma_{\min} A$, can be bounded according to*

$$\kappa(M_h(0, I_h)\tilde{A}) \le \frac{\lambda_{\max}(\tilde{A}) + \mathcal{O}(\rho_{\max})}{\lambda_{\min}(\tilde{A}) + \mathcal{O}(\rho_{\max})}. \tag{5.10}$$

Proof. As in [40], we note that

$$\lambda_{\min}(\tau\sigma_{\min}A) \leq \frac{w^T(\tau\sigma_{\min}A)w}{\|w\|^2} \leq \lambda_{\max}(\tau\sigma_{\min}A).$$

Thus, given (1.1) with A positive definite and $a = 0$ (which implies $\Delta_h = 1$), we can always multiply it by the scalar $\tau\sigma_{\min}$ such that (5.6) becomes

$$\lambda_{\min}(\tilde{A}) + \mathcal{O}(\rho_{\max}) \ \leq \ \lambda_i(M_h(0, W_h)\tilde{A}) \ \leq \ \lambda_{\max}(\tilde{A}) + \mathcal{O}(\rho_{\max}),$$

which implies (5.10). ∎

When the Krylov-subspace method approaches the solution, then both $\rho_{h+1} \to 0$ and $\rho_{h+2} \to 0$. Thus, using Proposition 5.2 with A positive definite, $a = 0$, and $W_h = I_h$, we have

$$\lim_{\substack{\rho_{h+1} \to 0 \\ \rho_{h+2} \to 0}} \kappa(M_h(0, I_h)\tilde{A}) \leq \kappa(\tilde{A}). \tag{5.11}$$

This result is weaker than the analogous result in Lemma 3.3 and Theorem 3.4 of [40]. However, as shown in section 7, the proposed AINV$\mathcal{K}$ preconditioner performs better than that in [40]. Section 6 also shows that AINV$\mathcal{K}$ preconditioners require less memory and computational cost.

Remark 5.3. Since λ is an eigenvalue of the preconditioned matrix $M_h(a, W_h)A$ if and only if λ is an eigenvalue of $\hat{\mathcal{L}}_h^T\mathcal{N}^T A\mathcal{N}\hat{\mathcal{L}}_h$ (see the proof of Theorem 5.1), we can derive some guidelines for the choice of the parameters a and W_h in (3.5). Indeed, they may be set in order to impose conditions like

$$\mathrm{tr}\begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{1,2}^T & M_{2,2} \end{bmatrix} = h + 1, \qquad \mathrm{tr}\,[M_{1,1}] = h$$

(see also [4] for details), which tend to force the clustering of the eigenvalues of the matrix in (5.8) to either $+1$ or -1.

5.1. Issues on the condition number of $M_h(a, W_h)A$. We now describe some theoretical results concerning the bounding of the condition number of the preconditioned matrix $M_h(a, W_h)A$ in the general case $a \neq 0$ and $W_h = \mathrm{diag}\{w_1^2, \ldots, w_h^2\}$.

PROPOSITION 5.4. *Consider $M_h(a, W_h)$ in (3.5) with $h \leq n-1$. Let $\mu_1 \leq \cdots \leq \mu_h$ be the eigenvalues of $|\widehat{T}_h|$ in (3.7), which are not all coincident. If Δ_h in (5.3) is positive, we have*

$$\kappa(M_h(a, W_h)A) \ \leq \ \frac{\max\left\{1, \tilde{\lambda}_{h+1}\right\}}{\min\left\{1, \tilde{\lambda}_1\right\}}\kappa(A), \tag{5.12}$$

where

$$\tilde{\lambda}_1 = \frac{\gamma_h - (\gamma_h^2 - 4\sigma_h)^{1/2}}{2}, \qquad \tilde{\lambda}_{h+1} = \frac{\gamma_h + (\gamma_h^2 - 4\sigma_h)^{1/2}}{2}, \tag{5.13}$$

$$\gamma_h = -(h-1)\mu_1 + \mathrm{tr}(|\widehat{T}_h|) + 1, \qquad \sigma_h = \frac{\Delta_h \det(|\widehat{T}_h|)}{\mu_h^{h-1}}. \tag{5.14}$$

Proof. Let $\lambda_1 \le \cdots \le \lambda_{h+1}$ be the eigenvalues of $\mathcal{T}_h$ in (3.6). From (5.4) and $\Delta_h > 0$, it follows that

$$\det(\mathcal{T}_h) = \Delta_h \det(|\widehat{T}_h|). \tag{5.15}$$

Therefore, by the Cauchy interlacing properties (Lemma 8.4.4 in [8]) between the eigenvalues $\{\mu_j\}_{j=1,\dots,h}$ and $\{\lambda_i\}_{i=1,\dots,h+1}$ we have the relation

$$\lambda_1 \le \mu_1 \le \lambda_2 \le \mu_2 \le \cdots \le \lambda_h \le \mu_h \le \lambda_{h+1}. \tag{5.16}$$

By (5.16), (3.6), and (5.15) we respectively obtain the following intermediate results:

(i) $\mu_1 \le \lambda_i \le \mu_h, \quad i = 2, \dots, h;$

(ii) $\sum_{i=1}^{h+1} \lambda_i = \operatorname{tr}(|\widehat{T}_h|) + 1;$

(iii) $\prod_{i=1}^{h+1} \lambda_i = \Delta_h \det(|\widehat{T}_h|).$

From (i) we deduce that $(h-1)\mu_1 \le \sum_{i=2}^{h} \lambda_i \le (h-1)\mu_h$, so that from (ii), (iii), (5.16), and recalling that the matrix $\mathcal{T}_h$ in (3.6) is positive definite, it follows that

$$\max\left\{0, -(h-1)\mu_h + \operatorname{tr}(|\widehat{T}_h|) + 1\right\} \le \lambda_1 + \lambda_{h+1} \le -(h-1)\mu_1 + \operatorname{tr}(|\widehat{T}_h|) + 1,$$

$$\frac{\Delta_h \det(|\widehat{T}_h|)}{\mu_h^{h-1}} \le \lambda_1 \lambda_{h+1} \le \frac{\Delta_h \det(|\widehat{T}_h|)}{\mu_1^{h-1}}.$$

From the last inequality (see also points (A) and (B) in Figure 1) in order to compute a lower bound $\tilde{\lambda}_1$ and an upper bound $\tilde{\lambda}_{h+1}$ on the smallest and the largest eigenvalue of the matrix $\mathcal{T}_h$ in (3.6), we have to solve the nonlinear system

$$\tilde{\lambda}_1 + \tilde{\lambda}_{h+1} = \gamma_h, \quad \tilde{\lambda}_1 \tilde{\lambda}_{h+1} = \sigma_h.$$

It follows that

$$\tilde{\lambda}_1 = \frac{\gamma_h - (\gamma_h^2 - 4\sigma_h)^{1/2}}{2}, \qquad \tilde{\lambda}_{h+1} = \frac{\gamma_h + (\gamma_h^2 - 4\sigma_h)^{1/2}}{2} \tag{5.17}$$

are real values, because $\gamma_h^2 - 4\sigma_h = (\tilde{\lambda}_1 - \tilde{\lambda}_{h+1})^2$. Now, from (5.1), the preconditioners $M_h(a, W_h)$ may be rewritten as in (5.2).

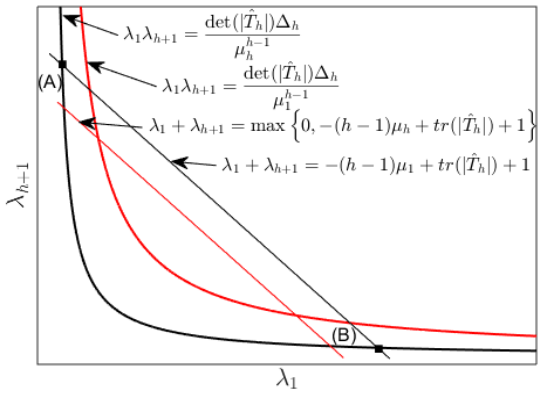

FIG. 1. *Relation between the eigenvalues* λ_1 *and* λ_{h+1} *of the matrix* $\mathcal{T}_h$ *in* (3.6).

Note that the smallest and largest eigenvalues of the matrix

$$G_h = \begin{bmatrix} \mathcal{T}_h & 0 \\ 0 & I_{n-(h+1)} \end{bmatrix},$$

respectively, are

$$\lambda_{\min}(G_h) = \min\{1, \lambda_1\}, \qquad \lambda_{\max}(G_h) = \max\{1, \lambda_{h+1}\}.$$

Thus, it follows from (5.2) and the orthogonality of $\mathcal{N}$ that

$$\|M_h(a, W_h)A\| \;\le\; \lambda_{\max}(A)\|\mathcal{N}\|^2\lambda_{\max}(G_h^{-1}) \;=\; \frac{\lambda_{\max}(A)}{\min\{1, \lambda_1\}}$$

and

$$\|(M_h(a, W_h)A)^{-1}\| \le\; \lambda_{\max}(A^{-1})\|\mathcal{N}^{-1}\|^2\lambda_{\max}(G_h) \;=\; \frac{\max\{1, \lambda_{h+1}\}}{\lambda_{\min}(A)}.$$

Hence, from (5.17) we obtain (5.12). ∎

We observe that as expected, W_h in (3.5) strongly affects the distribution of the eigenvalues of $M_h(a, W_h)A$, as shown by Theorem 5.1, and also its condition number, as indicated by Proposition 5.4. It is worth noting that the bound on $\kappa(M_h(a, W_h)A)$ in (5.12) can be simplified as follows.

PROPOSITION 5.5. *Under the hypotheses of Proposition* 5.4*, inequality* (5.12) *is reduced to*

$$\kappa(M_h(a, W_h)A) \le \xi_h\kappa(A), \tag{5.18}$$

where $\xi_h = \tilde{\lambda}_{h+1}/\tilde{\lambda}_1$ *with* $\tilde{\lambda}_{h+1}$ *and* $\tilde{\lambda}_1$ *given by* (5.13)*. Moreover,*

(i) ξ_h *increases with* $|a|$*, and* $\lim_{|a|\uparrow\omega_h}\xi_h = +\infty$*, where* $\omega_h = (e_h^T|\widehat{T}_h|^{-1}e_h)^{-1/2}$*;*

(ii) *when* $a = 0$*,* ξ_h *attains its minimum*

$$\xi_h^* \;=\; \frac{\gamma_h + \beta_h}{\gamma_h - \beta_h}, \qquad \beta_h = \left(\gamma_h^2 - 4\frac{\det(|\widehat{T}_h|)}{\mu_h^{h-1}}\right)^{1/2}. \tag{5.19}$$

Proof. Imposing the conditions $\tilde{\lambda}_1 \le 1$ and $\tilde{\lambda}_{h+1} \ge 1$ and using (5.13) we obtain $1 - \gamma_h + \sigma_h \le 0$. This inequality is fulfilled by coupling (5.14) and

$$\mathrm{tr}(|\hat{T}_h|) - (h-1)\mu_1 \ge \frac{\det(|\hat{T}_h|)}{\mu_h^{h-1}}.$$

Indeed, the last relation follows from properties of trace and determinant of $|\hat{T}_h|$, and observing that its left-hand side is larger than μ_h, while its right-hand side is smaller than μ_h. Hence, (5.12) implies (5.18).

For item (i), consider (5.3). When $|a|$ is increasing, Δ_h is decreasing, and hence from (5.12)–(5.14) and (5.19), ξ_h is increasing. In addition, when $|a| \to \omega_h$, we have $\sigma_h \to 0$. Then $2\tilde{\lambda}_1 \to 0$ and $2\tilde{\lambda}_{h+1} \to 2\gamma_h > 2$. Thus, we directly have (i).

For (ii), since ξ_h is a continuous function of a, we have from item (i) that ξ_h attains its minimum at $a = 0$. Finally, substituting this value into (5.3) we have $\Delta_h = 1$ and hence by (5.14) $\sigma_h = \det(|\widehat{T}_h|)/\mu_h^{h-1}$, so that ξ_h is reduced by (5.13) to ξ_h^*. ∎

6. Storage and computational cost of the AINV$\mathcal{K}$ preconditioners. Since we are dealing with large-scale problems, we first observe that the case $h \approx n$ in the proposed class of preconditioners (3.5) is not of interest. Indeed, in the literature of preconditioners that make use of a "memory" [50, 51], the latter (which corresponds to the value of h in AINV$\mathcal{K}$) typically does not exceed 10 vectors. Yet this proves sufficient in several applications. The recursion of any preconditioned Krylov-subspace method never uses directly full matrices, because only the product of the preconditioner times a vector is required. The AINV$\mathcal{K}$ class needs matrix-vector products $M_h(a, W_h)v$, $v \in \mathbb{R}^n$. From the structure of $M_h(a, W_h) = I_n + R_{h+1}[\mathcal{T}_h^{-1} - I_{h+1}]R_{h+1}^T$ in (3.5), it is evident that the computational effort in computing $M_h(a, W_h)v$ is given by

$$(h+1)n + n(h+1) + (h+1)^2 \approx 2(h+1)n, \tag{6.1}$$

as $h \ll n$, whether A is indefinite or positive definite. On the other hand, the cost in (6.1) includes $(h+1)^2$ multiplications that are due to the product of $\mathcal{T}_h^{-1}$ (see (3.6)) and a vector, using (5.5).

The quantity (6.1) is competitive with the cost of LMP [40], Ritz-LMP [39], and PREQN preconditioners [50], considering that $h \ll n$. Indeed, with respect to LMP, the proposed class requires only the $(h+1)$ vectors $u_1, \ldots, u_{h+1}$ in place of the $2h$ vectors $u_1, Au_1, \ldots, u_h, Au_h$. Even when A is positive definite, LMP requires computing these $2h$ vectors with a procedure that needs h matrix-vector products, at an additional cost of approximately $3h^2n$ flops [40]. In addition, Ritz-LMP variant requires the full computation of Ritz pairs for an $h \times h$ (possibly indefinite) tridiagonal matrix. Conversely, as by [39, section 3.5.3], Ritz-LMP's storage amounts to $h+2$ vectors, while the application of Ritz-LMP to a vector needs $(4h+9)n$ flops: this is more than twice the amount in (6.1). Finally, since PREQN is a special case of LMP, the computational cost of the AINV$\mathcal{K}$ class is also preferable to that of PREQN.

7. Numerical experiments. We now report the results of extensive numerical experimentation with the AINV$\mathcal{K}$ class of preconditioners (3.5). We first considered the standard test matrices (BCSSTRUC4) from the Harwell–Boeing Sparse Matrix Collection [25] (some results are reported in [31, 32]). We do not describe them here because they were aimed at preliminarily (i) validating the theory in Theorem 5.1 and (ii) highlighting some similarities with the numerical results reported in [40] for the LMP preconditioners.

We then embedded the AINV$\mathcal{K}$ class of preconditioners in a linesearch-based truncated Newton method for the solution of large-scale unconstrained optimization problems, which is our main topic of interest. It requires the solution of a sequence of symmetric linear systems. We show that the overall optimization scheme is efficient, even for $h < 10$ in (3.5). In particular, we use the algorithmic scheme in [33, 34], where some implementation details are described. For simplicity, we set $a = 0$ in (3.5). For $W_h = \mathrm{diag}\{w_1^2, \ldots, w_h^2\}$, we choose $w_1 = w_2 = \cdots = w_h = 100$, with further details given below. Unlike [34], where similar choices of the parameters were carried out in the positive definite case, we tackle indefinite problems, so that the Lanczos process is used in place of the CG algorithm. In particular, we use SYMMBK in [56] for computing the gradient-related Newton-type search direction and iteratively constructing a preconditioner in the AINV$\mathcal{K}$ class.

We assume that h steps of the Lanczos process have been performed. As discussed in subsection 2.2, when A is indefinite, symmetric indefinite factorizations of T_h in (2.4) must be considered. Unlike [33], where use of the CG method might be ques-

tionable, we adopt SYMMBK, which recursively computes the Bunch and Kaufman indefinite factorization of T_h (2.4). Note that the factors for T_{h+1} can be obtained from those of T_h. We recall two main features: (i) a 2×2 pivot may be performed in place of a 1×1 pivot; (ii) the partial pivoting strategy of [12] is adopted for determining the pivot size in order to guarantee numerical stability [14, 15, 19, 42]. By means of this choice, Assumption 3.1 is fulfilled (unlike in [33]), so that the preconditioners belonging to the AINV$\mathcal{K}$ class are still positive definite and can be efficiently computed via the SYMMBK procedure.

All codes were written in Fortran and compiled with gfortran 6 under Linux Ubuntu 18.04. The runs were performed on a PC with Intel Core i7-4790K quad-core 4.00 (up to 4.40) GHz Processor and 32 GB RAM. A Fortran routine that implements SYMMBK is available from the HSL Mathematical Software Library [46] (formerly the Harwell Subroutine Library), namely, routine `HSL_MI02`. This routine includes the eigendecomposition of any 2 by 2 block of B_h in (2.4) (obtained via `DLAEV2` from LAPACK) and incorporates the resulting eigenvectors into L_h, in the form (2.4), given by

$$T_h = L_h B_h L_h^T = \tilde{L}_h \tilde{D}_h \tilde{L}_h^T,$$

where $\tilde{D}_h$ is diagonal and $\tilde{L}$ is lower triangular. Therefore, it is perfectly suited for constructing preconditioners of the AINV$\mathcal{K}$ class. A preconditioner can be provided to `HSL_MI02`. Also observe that SYMMBK computes conjugate directions. Indeed, the columns of P_h, being $R_h = P_h \tilde{L}_h^T$, are scaled conjugate directions, and we set P_h as in (4.2). Hence the preconditioners in (3.5) can be rewritten by replacing R_h by $P_h \tilde{L}_h^T$, and, in particular, we have $R_h |\widehat{T}_h|^{-1} R_h^T = P_h (\mathcal{D}_h^{-1} W_h^{-1}) P_h^T$. Then, the preconditioners (3.5) are completely defined by P_h, $\tilde{L}_h$, and $\mathcal{D}_h$, which are iteratively constructed by SYMMBK.

The preconditioning strategy we adopt (at each outer iteration k) for solving the Newton system $\nabla^2 f(x_k) d = -\nabla f(x_k)$ is similar to that in [33, 34], and it can be briefly summarized as follows: the information gained after a very small number of iterations of SYMMBK is used to construct the preconditioner. Then, the iterations are continued by applying preconditioned SYMMBK. More specifically, the iterations of SYMMBK are stopped as soon as $\bar{h} \ll n$ iterations have been performed and the preconditioner is built; then the inner iterations continue with preconditioned SYMMBK. If the inner iterations are prematurely stopped (by a termination criterion) before $\bar{h}$ steps, we estimate that not enough information is available to build a preconditioner. As pointed out in Remark 5.2 of [33], we prefer this strategy over using information obtained at the previous $(k-1)$th outer iteration, as in [50]. In this way, we use fresh information from the current Hessian $\nabla^2 f(x_k)$ rather than $\nabla^2 f(x_{k-1})$. Indeed, the latter Hessian yields misleading information when x_k is far from x_{k-1}. After careful tuning, we chose $\bar{h} = 7$ as a value analogous to that used for the "memory" of the preconditioner PREQN in [50]. In the practical implementation, the threshold $\bar{h}$ must be dynamically adjusted to avoid terminating the current iteration of SYMMBK whenever a 2×2 pivot is not yet completed.

We use the standard residual-based criterion for terminating the inner iterations [17, 18, 53] and the usual stopping rule

$$\|\nabla f(x_k)\| \leq 10^{-5} \max\{1, \|x_k\|\}$$

for the outer iterations [50, 53]. We consider that a failure occurs on a test problem when the CPU time (in seconds) exceeds 1800. As test set we considered all large-scale unconstrained problems in the CUTEst collection [38]. For each variable-dimension

problem we considered three different dimensions 1000, 10000, and 50000 (in some cases 3000, 12000, 60000), depending on the problem in hand. The resulting test set is composed of 201 test problems, which include convex and nonconvex problems. In what follows we report the results of this experimentation (the value of a to select in (3.5) is given in [34], where the positive definite case and a lower dimension test set are considered). We performed a new tuning of the parameter w on the whole CUTEst test set described above and confirmed that the choice $w_1 = \cdots = w_h = 100$ (adopted in [34]) is still the best one, also on nonconvex problems. Therefore we used this choice in our numerical experiments. We display results using performance profiles [24].

The first results concern a comparison between the unpreconditioned truncated Newton method and the preconditioned one that uses the AINV$\mathcal{K}$ preconditioner (adopting the parameters above). Figure 2 illustrates the performance profiles. With respect to inner iterations, the preconditioned algorithm is best in terms of both efficiency and robustness. As regards the number of function evaluations and CPU time, the efficiency of the two algorithms is comparable, while the preconditioned algorithm is definitely more robust. This is because, even if the preconditioner involves additional computation, it improves the behavior of the algorithm especially on difficult

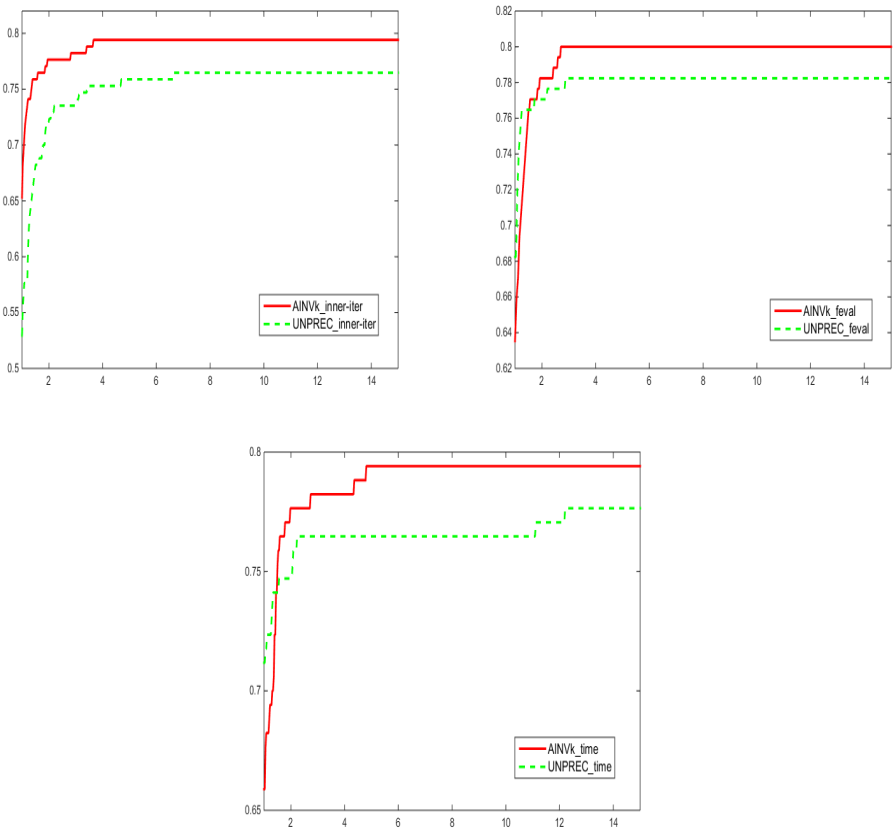

FIG. 2. *Comparison between the unpreconditioned and preconditioned (AINV$\mathcal{K}$) truncated Newton method, in terms of inner iterations (top left), number of function evaluations (top right), and CPU time (bottom).*

problems, so that a certain number of failures occurring in the unpreconditioned case are avoided. It is worth noting that with respect to the proposals in [33, 34], here the use of SYMMBK in place of CG enhances performance.

Other interesting results concern the comparison between two versions of the preconditioned truncated Newton algorithm: the first using AINV$\mathcal{K}$, the second adopting Ritz-LMP [39]. The comparison between the resources used by the two is discussed in section 6. Now we compare their numerical behavior. It is worth recalling that Ritz-LMP requires storage of $\bar{h}+2$ vectors [39], while LMP needs $2\bar{h}$ vectors: this suggests that information exploited by AINV$\mathcal{K}$ might be more similar to that used in Ritz-LMP rather than LMP. The Ritz-LMP preconditioner is computed according to (18)–(19) and (25) of [39] by exploiting the Lanczos procedure in SYMMBK and using Ritz pairs obtained through the LAPACK `DSTEV` routine. Of course other choices (possibly more efficient) are allowed, including the use of the LMP preconditioners with different first-level preconditioners (we did not investigate them here, and we set $H_0 = I_n$ in (4.1)).

Figure 3 illustrates the performance profiles on the whole test set. The better behavior of truncated Newton algorithm using AINV$\mathcal{K}$ is evident. However,

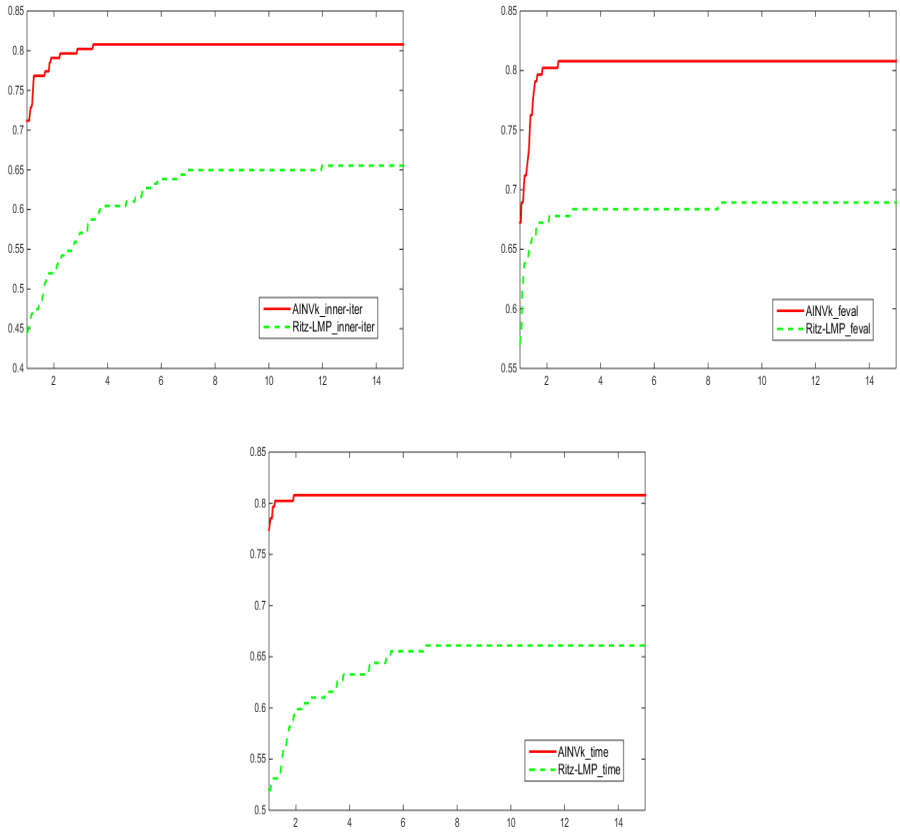

FIG. 3. *Comparison between the AINV$\mathcal{K}$ and Ritz-LMP preconditioners on the whole set of test problems, in terms of inner iterations (top left), number of function evaluations (top right) and CPU time (bottom).*

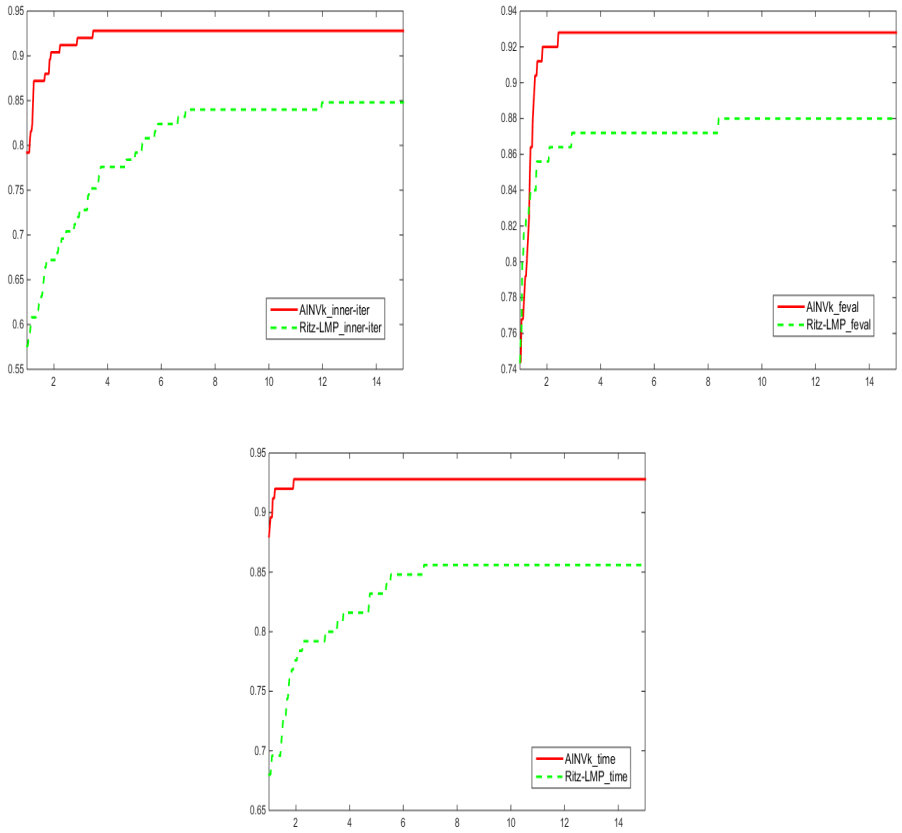

FIG. 4. *Comparison between the AINVK and Ritz-LMP preconditioners on convex problems, in terms of inner iterations (top left), number of function evaluations (top right), and CPU time (bottom).*

this comparison is not completely "fair." Indeed, on nonconvex problems, unlike AINV$\mathcal{K}$ the Ritz-LMP preconditioners are possibly indefinite, so that the preconditioned SYMMBK method we adopted might suffer from reduced performance. In this regard, a similar drawback would also arise in case SYMMBK were replaced by CG. Therefore we repeated the comparison by restricting the test set to all problems where negative curvatures are not encountered. Figure 4 shows the corresponding performance profiles. As expected, on convex problems, the gap between the two algorithms is reduced, so that Ritz-LMP is also slightly preferable to AINV$\mathcal{K}$ in termsof function evaluations. Nevertheless, also in the convex case AINV$\mathcal{K}$ is on average preferable. This suggests that possibly the second term in the right-hand side of (3.5) is carefully built through the spectral theorem without spoiling the information implicitly collected by the Krylov-subspace method on the inverse of A.

These comparisons highlight the effectiveness of the AINV$\mathcal{K}$ preconditioners on indefinite linear systems, and their theoretical properties (stated in section 5) seem to be confirmed, particularly the capability of clustering some eigenvalues of the precon-

ditioned matrix and avoiding expansion of its spectrum. In this sense, the AINV$\mathcal{K}$ preconditioners represent a significant enhancement to the proposal in [34], where only the positive definite case is considered and where the spectral analysis is limited to the singular values of the preconditioned matrix.

To further assess the AINV$\mathcal{K}$ class, we performed another numerical test versus Ritz-LMP by considering problems of huge dimension. We selected all problems from the CUTEst collection whose dimension can be set to 10^6 (allowing maximum CPU time of 3600 seconds). Moreover, in order to carry out a fair comparison with the Ritz-LMP preconditioners, we excluded those problems where negative curvatures were encountered. The resulting test set includes 40 (convex) problems. Figure 5 gives the corresponding comparison between AINV$\mathcal{K}$ and Ritz-LMP. Comparing with Figure 4, we see that the gap in terms of efficiency and robustness is confirmed in favor of AINV$\mathcal{K}$. This suggests that the effectiveness of the AINV$\mathcal{K}$ is not very sensitive to the dimension of the test problems. The additional computational burden due to preconditioning by AINV$\mathcal{K}$ is offset by the reduced number of inner iterations.

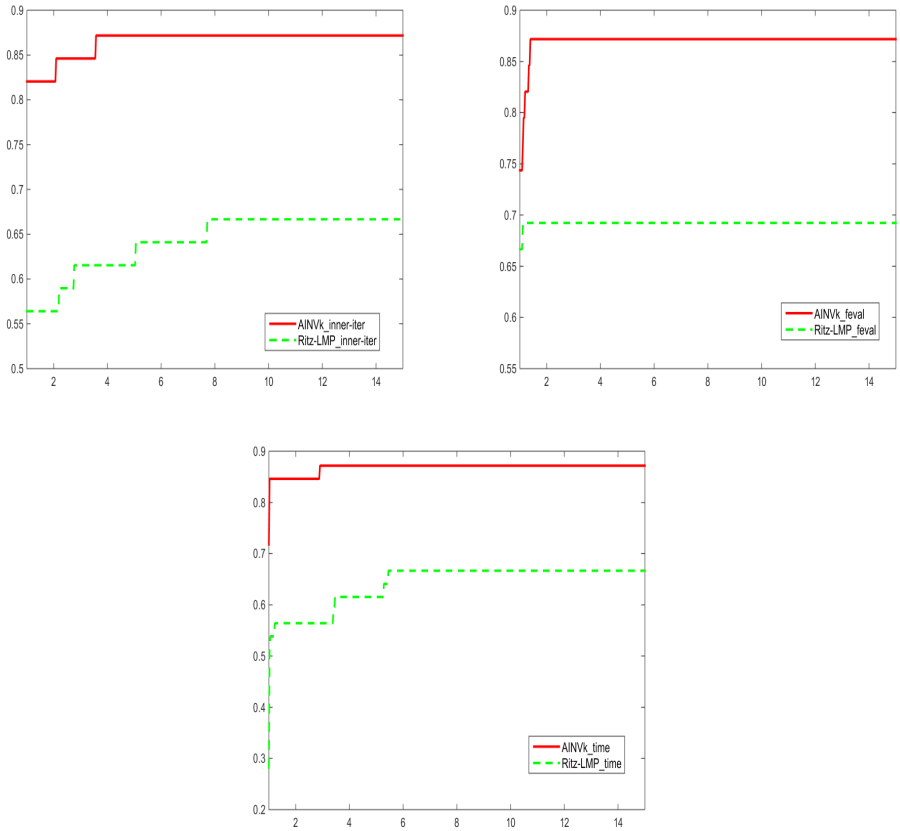

FIG. 5. *Comparison between the AINV$\mathcal{K}$ and Ritz-LMP preconditioners on* huge *convex problems, in terms of inner iterations (top left), number of function evaluations (top right), and CPU time (bottom).*

The effect of preconditioning is even more important on huge problems, when a great number of outer iterations is performed. Indeed, this is the case when the number of the Newton systems to be solved significantly increases.

As illustrated by Figure 5, the value $\bar{h} = 7$ (a relatively low storage) is sufficient for AINV$\mathcal{K}$ in order to collect enough information efficiently on the Hessian. On the contrary, Ritz-LMP using the same number of vectors seems to be less competitive. Indeed (see Appendix A), Ritz-LMP needs a larger number of inner iterations in order to provide even a more poorly scaled Newton-type direction. The larger number of inner iterations also explains the performance of Ritz-LMP with respect to CPU time. An insight into such behavior follows from the subsequent reasoning: the rightmost term in both expressions (3.5) and (4.1) attempts to capture information on the inverse Hessian. However, also in view of Remark 4.1, on the Krylov subspace spanned after the first $\bar{h}$ iterations, Ritz-LMP seems to construct a less accurate approximation of the inverse Hessian compared to AINV$\mathcal{K}$. We recall that, as outlined at the end of section 4, the LMP preconditioners might notably improve their performance by using a proper first-level preconditioner. Indeed, the use of $H_0 = I_n$ (no information on matrix A available at the beginning) along with low memory could limit the LMP preconditioners' efficiency. On the other hand, this shows the good performance of the AINV$\mathcal{K}$ preconditioners on convex problems even if low memory is used and without requiring a first-level preconditioner. As for nonconvex problems, a comparison between AINV$\mathcal{K}$ and LMP may be further enhanced, observing that LMP can be fruitfully coupled with GMRES(m), as in [39, 49].

8. Conclusions. We have proposed a class of general-purpose positive definite preconditioners for the solution of large-scale symmetric indefinite linear systems, which can be extended to a sequence of symmetric systems. The main features of the proposed AINV$\mathcal{K}$ preconditioner are (i) it can be built for both positive definite and indefinite systems; (ii) it is matrix-free and iteratively constructed as a byproduct of the Lanczos-based Krylov-subspace method; (iii) spectral properties of the preconditioned matrix can be provided; (iv) fewer vectors of storage are needed compared to LMP, Ritz-LMP, and PREQN; (v) the computational effort involved in computing the product of the preconditioner times a vector is competitive with existing methods; (vi) the numerical performance of AINV$\mathcal{K}$ proved to be competitive with respect to state-of-the-art methods. In particular, on large and huge optimization test problems, the proposed preconditioners have shown remarkable effectiveness and robustness. Finally, we believe that the combined use of AINV$\mathcal{K}$ preconditioners and *damped techniques* [2] can be worth investigating.

Appendix A. Tables of numerical results. Here we report the complete numerical results for AINV$\mathcal{K}$ (Table 1) and Ritz-LMP (Table 2) on the subset of huge convex problems. *Iter*, *Funct*, *Inner-it*, *Time*, respectively, represent the number of outer iterations, function evaluations, inner iterations, and the overall computational time (in seconds). Unfortunately, it is impossible to combine the two tables on one page for facilitating the comparison of the results. However, the performance profile in Figure 5 shows the improved performance of AINV$\mathcal{K}$.

TABLE 1
Results for AINVK on huge convex problems.

Problem	*n*	*Iter*	*Funct*	*Inner-it*	*Time (s)*
ARWHEAD	1000000	13	148	13	7.35
BDQRTIC	1000000	–	–	–	> 3600
BRYBND	1000000	12	18	21	8.25
CRAGGLVY	1000000	27	62	126	24.21
CURLY10	1000000	–	–	–	> 3600
DIXMAANA	1050000	8	13	8	1.54
DIXMAANB	1050000	6	11	7	1.50
DIXMAANC	1050000	6	12	8	1.56
DIXMAAND	1050000	6	8	8	1.49
DIXMAANE	1050000	21	24	5061	390.68
DIXMAANF	1050000	19	24	18099	1471.16
DIXMAANH	1050000	19	32	42356	3503.84
DIXMAANI	1050000	22	25	4790	373.70
DIXMAANK	1050000	22	35	1557	135.52
DIXMAANL	1050000	14	15	1720	147.61
DQDRTIC	1000000	10	31	17	3.20
DQRTIC	1000000	–	–	–	> 3600
EDENSCH	1000000	14	54	25	5.52
ENGVAL1	1000000	8	21	21	3.26
FLETCBV2	1000000	1	1	0	0.15
FLETCHCR	1000000	17	55	72	11.64
FREUROTH	1000000	17	90	30	7.65
LIARWHD	1000000	37	466	43	19.75
MOREBV	1000000	1	1	0	0.09
NCB20B	1000000	23	105	2952	1710.88
NONDIA	1000000	9	170	8	4.57
NONDQUAR	1000000	19	48	136	15.86
PENALTY1	1000000	60	60	101	15.24
POWELLSG	1000000	31	101	88	7.80
POWER	1000000	–	–	–	> 3600
QUARTC	1000000	–	–	–	> 3600
SCHMVETT	1000000	7	9	26	14.85
SPARSQUR	1000000	32	38	159	136.37
SROSENBR	1000000	14	89	15	3.04
TESTQUAD	1000000	29	65	13219	733.27
TOINTGSS	1000000	2	3	1	0.25
TQUARTIC	1000000	2	2	1	0.32
TRIDIA	1000000	29	121	19159	1177.25
VAREIGVL	1000000	10	44	13	4.52
WOODS	1000000	16	78	21	4.08

TABLE 2
Results for Ritz-LMP on huge convex problems.

Problem	*n*	*Iter*	*Funct*	*Inner-it*	*Time (s)*
ARWHEAD	1000000	13	148	13	7.32
BDQRTIC	1000000	–	–	–	> 3600
BRYBND	1000000	14	20	162	43.52
CRAGGLVY	1000000	27	62	348	82.49
CURLY10	1000000	–	–	–	> 3600
DIXMAANA	1050000	8	13	8	1.56
DIXMAANB	1050000	6	11	7	1.52
DIXMAANC	1050000	6	12	8	1.59
DIXMAAND	1050000	6	8	8	1.50
DIXMAANE	1050000	15	18	2421	469.17
DIXMAANF	1050000	–	–	–	> 3600
DIXMAANH	1050000	–	–	–	> 3600

TABLE 2
(cont.).

Problem	*n*	*Iter*	*Funct*	*Inner-it*	*Time (s)*
DIXMAANI	1050000	15	18	10456	2031.24
DIXMAANK	1050000	–	–	–	> 3600
DIXMAANL	1050000	–	–	–	> 3600
DQDRTIC	1000000	10	31	17	3.20
DQRTIC	1000000	–	–	–	> 3600
EDENSCH	1000000	14	54	25	5.45
ENGVAL1	1000000	8	21	21	3.31
FLETCBV2	1000000	1	1	0	0.15
FLETCHCR	1000000	17	60	684	134.42
FREUROTH	1000000	17	90	30	7.62
LIARWHD	1000000	37	466	43	19.68
MOREBV	1000000	1	1	0	0.10
NCB20B	1000000	20	96	828	594.58
NONDIA	1000000	9	170	8	4.56
NONDQUAR	1000000	–	–	–	> 3600
PENALTY1	1000000	60	60	101	15.66
POWELLSG	1000000	31	101	88	7.83
POWER	1000000	–	–	–	> 3600
QUARTC	1000000	–	–	–	> 3600
SCHMVETT	1000000	6	8	131	51.12
SPARSQUR	1000000	30	32	4440	2345.49
SROSENBR	1000000	14	89	15	3.15
TESTQUAD	1000000	–	–	–	> 3600
TOINTGSS	1000000	2	3	1	0.25
TQUARTIC	1000000	2	2	1	0.32
TRIDIA	1000000	–	–	–	> 3600
VAREIGVL	1000000	10	44	13	4.53
WOODS	1000000	16	78	21	4.12

Acknowledgments. The authors are sincerely thankful to the Associate Editor and the referees for their thorough reading of the manuscript. Their useful comments and suggestions have highly improved the quality of our paper.

REFERENCES

[1] M. Al-baali, *Extra updates for the BFGS method*, Optim. Methods Softw., 13 (2000), pp. 157–179.
[2] M. Al-Baali, A. Caliciotti, G. Fasano, and M. Roma, *Exploiting damped techniques for nonlinear conjugate gradient methods*, Math. Methods Oper. Res., 86 (2017), pp. 501–522.
[3] J. Baglama, D. Calvetti, G. Golub, and L. Reichel, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20 (1998), pp. 243–269.
[4] S. Bellavia, J. Gondzio, and B. Morini, *A matrix–free preconditioner for sparse symmetric positive definite systems and least–squares problems*, SIAM J. Sci. Comput., 35 (2013), pp. A192–A211.
[5] M. Benzi, *Preconditioning techniques for large linear systems: A survey*, J. Comput. Phys., 182 (2002), pp. 418–477.
[6] M. Benzi, J. Cullum, and M. Tůma, *Robust approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 22 (2000), pp. 1318–1332.
[7] L. Bergamaschi, V. De Simone, D. di Serafino, and A. Martinez, *BFGS-like updates of constraint preconditioners for sequences of KKT linear systems in quadratic programming*, Numer. Linear Algebra Appl., 25 (2018), pp. 1–19.
[8] D. Bernstein, *Matrix Mathematics*, 2nd ed., Princeton University Press, Princeton, NJ, 2009.
[9] D. Bertaccini and F. Durastante, *Limited memory block preconditioners for fast solution of fractional partial differential equations*, J. Sci. Comput., 77 (2018), pp. 950–970.
[10] D. Bertaccini and S. Filippone, *Sparse approximate inverse preconditioners on high performance GPU platforms*, Comput. Math. Appl., 71 (2016), pp. 693–711.

[11] L. Bottou, F. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Rev., 60 (2018), pp. 223–311.
[12] J. Bunch, *Partial pivoting strategies for symmetric matrices*, SIAM J. Numer. Anal., 11 (1974), pp. 521–528.
[13] J. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear equations*, Math. Comput., 31 (1977), pp. 163–179.
[14] J. Bunch and R. Marcia, *A pivoting strategy for symmetric tridiagonal matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 911–922.
[15] J. Bunch and R. Marcia, *A simplified pivoting strategy for symmetric tridiagonal systems*, Numer. Linear Algebra Appl., 13 (2006), pp. 865–867.
[16] R. Byrd, R. Schnabel, and G. Shultz, *Parallel quasi-Newton methods for unconstrained optimization*, Math. Program., 42 (1988), pp. 273–306.
[17] A. Caliciotti, G. Fasano, S. G. Nash, and M. Roma, *An adaptive truncation criterion, for linesearch-based truncated Newton methods in large scale nonconvex optimization*, Oper. Res. Lett., 46 (2018), pp. 7–12.
[18] A. Caliciotti, G. Fasano, S. G. Nash, and M. Roma, *Data and performance profiles applying an adaptive truncation criterion, within linesearch-based truncated Newton methods, in large scale nonconvex optimization*, Data in Brief, 17 (2018), pp. 246–255.
[19] R. Chandra, *Conjugate Gradient Methods for Partial Differential Equations*, Ph.D thesis, Yale University, New Haven, CT, 1978.
[20] S.-C. Choi, *Iterative Methods for Singular Linear Equations and Least-Squares Problems*, Ph.D thesis, Stanford University, Stanford, CA, 2006.
[21] S.-C. Choi, C. Paige, and M. Saunders, *MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems*, SIAM J. Sci. Comput., 33 (2011), pp. 1810–1836.
[22] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, MOS-SIAM Ser. Optim., SIAM, Philadelphia, PA, 2000.
[23] V. De Simone and D. di Serafino, *A matrix-free approach to build band preconditioners for large-scale bound-constrained optimization*, J. Comput. Appl. Math., 268 (2014), pp. 82–92.
[24] E. D. Dolan and J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
[25] I. Duff, R. Grimes, and J. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[26] J. Duintjer Tebbens and M. Tůma, *Efficient preconditioning of sequences of nonsymmetric linear systems*, SIAM J. Sci. Comput., 29(5) (2007), pp. 1918–1941.
[27] J. Duintjer Tebbens and M. Tůma, *Preconditioner updates for solving sequences of linear systems in matrix-free environment*, Numer. Linear Algebra Appl., 17 (2010), pp. 997–1019.
[28] G. Fasano, *A CG-type method for the solution of Newton's equation within optimization frameworks*, Optim. Methods Softw., 19 (2004), pp. 267–290.
[29] G. Fasano, *Planar–conjugate gradient algorithm for large–scale unconstrained optimization, Part* 1: *Theory*, J. Optim. Theory Appl., 125 (2005), pp. 523–541.
[30] G. Fasano, *Planar-conjugate gradient algorithm for large-scale unconstrained optimization, Part* 2: *Application*, J. Optim. Theory Appl., 125 (2005), pp. 543–558.
[31] G. Fasano and M. Roma, *A Class of Preconditioners for Large Indefinite Linear Systems, as By-Product of Krylov Subspace Methods: Part* I, Technical Report 4/2011, Dipartimento di Management, Università Ca' Foscari, Venezia, 2011.
[32] G. Fasano and M. Roma, *A Class of Preconditioners for Large Indefinite Linear Systems, as By-Product of Krylov Subspace Methods: Part* II, Technical Report 5/2011, Dipartimento di Management, Università Ca' Foscari, Venezia, 2011.
[33] G. Fasano and M. Roma, *Preconditioning Newton–Krylov methods in nonconvex large scale optimization*, Comput. Optim. Appl., 56 (2013), pp. 253–290.
[34] G. Fasano and M. Roma, *A novel class of approximate inverse preconditioners for large positive definite linear systems in optimization*, Comput. Optim. Appl., 65 (2016), pp. 399–429.
[35] P. E. Gill, W. Murray, D. B. Ponceleon, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.
[36] L. Giraud, S. Gratton, and E. Martin, incremental spectral preconditioners for sequences of linear systems, Appl. Numer. Math., 57 (2007), pp. 1164–1180.
[37] G. Golub and C. Van Loan, *Matrix Computations*, 4th ed., The John Hopkins Press, Baltimore, MD, 2013.

[38] N. I. M. Gould, D. Orban, and P. L. Toint, CUTEst: *A constrained and unconstrained testing environment with safe threads*, Comput. Optim. Appl., 60 (2015), pp. 545–557.
[39] S. Gratton, S. Mercier, N. Tardieu, and X. Vasseur, *Limited memory preconditioners for symmetric indefinite problems with application to structural mechanics*, Numer. Linear Algebra Appl., 23 (2016), pp. 865–887.
[40] 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 J. Optim., 21 (2011), pp. 912–935.
[41] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, PA, 1997.
[42] N. Higham, *Stability of block LDL^T factorization of a symmetric tridiagonal matrix*, Numer. Linear Algebra Appl., 287 (1999), pp. 181–189.
[43] N. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, PA, 2002.
[44] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1999.
[45] C.-Y. Hsia, W.-L. Chiang, and C.-J. Lin, *Preconditioned conjugate gradient methods in truncated Newton frameworks for large-scale linear classification*, in Proceedings of Machine Learning Research, J. Zhu and I. Takeuchi, eds., Asian Conference on Machine Learning, 2018, pp. 1–15.
[46] *The HSL Mathematical Software Library*, http://www.hsl.rl.ac.uk/.
[47] L. Lukšan and J. Vlček, *Efficient tridiagonal preconditioner for the matrix-free truncated Newton method*, Appl. Math. Comput., 235 (2014), pp. 394–407.
[48] R. Marcia, *On solving sparse symmetric linear systems whose definiteness is unknown*, Appl. Numer. Math., 58 (2008), pp. 449–458.
[49] S. Mercier, S. Gratton, N. Tardieu, and X. Vasseur, *A new preconditioner update startegy for the solution of sequences of linear systems in structural mechanics: Application to saddle point problems in elasticity*, Comput. Mech., 60 (2017), pp. 969–982.
[50] J. Morales and J. Nocedal, *Automatic preconditioning by limited memory quasi-Newton updating*, SIAM J. Optim., 10 (2000), pp. 1079–1096.
[51] J. L. Morales and J. Nocedal, *Algorithm* PREQN*: Fortran* 77 *subroutine for preconditioning the conjugate gradient method*, ACM Trans. Math. Software, 27 (2001), pp. 83–91.
[52] S. Nash, *Preconditioning of truncated-Newton methods*, SIAM J. Sci. Stat. Comp., 6 (1985), pp. 599–616.
[53] S. Nash, *A survey of truncated-Newton methods*, J. Comput. Appl. Math., 124 (2000), pp. 45–59.
[54] J. Nocedal, *Updating Quasi-Newton matrices with limited storage*, Math. Comput., 35 (1980), pp. 773–782.
[55] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer-Verlag, New York, 2006.
[56] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[57] M. Roma, *A dynamic scaling based preconditioning for truncated Newton methods in large scale unconstrained optimization*, Optim. Methods Softw., 20 (2005), pp. 693–713.
[58] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.
[59] Y. Saad and M. H. Schultz, *A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.
[60] M. A. Saunders, *Solution of sparse rectangular systems using LSQR and CRAIG*, BIT, 35 (1995), pp. 588–604.
[61] J. Sifuentes, M. Ebree, and R. Morgan, *GMRES convergence for perturbed coefficient matrices, with application to approximate deflation preconditioning*, SIAM J. Matrix Anal. Appl., 34(3) (2013), pp. 1066–1088.
[62] J. Stoer, *Solution of large linear systems of equations by conjugate gradient type methods*, in Mathematical Programming. The State of the Art, A. Bachem, M.Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 540–565.
[63] L. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
[64] H. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, Cambridge, UK, 2003.