SIAM J. MATRIX ANAL. APPL.
Vol. 44, No. 1, pp. 359–381

ANALYSIS OF THE TRUNCATED CONJUGATE GRADIENT METHOD FOR LINEAR MATRIX EQUATIONS*

VALERIA SIMONCINI† AND YUE HAO‡

Abstract. The matrix-oriented version of the conjugate gradient (CG) method can be used to approximate the solution to certain linear matrix equations. To limit memory consumption, low-rank reduction of the factored iterates is often employed, possibly leading to disruption of the regular convergence behavior. We analyze the properties of the method in the matrix regime and identify the quantities that are responsible for early termination, usually stagnation, when truncation is in effect. Moreover, we illustrate relations between CG and a projection technique directly applied to the same matrix equation.

Key words. conjugate gradients, linear matrix equations, truncation strategies, low-rank methods

MSC code. 65F30

DOI. 10.1137/22M147880X

1. Introduction.

Multiterm matrix equations

$$A_1XB_1 + A_2XB_2 + \cdots + A_lXB_l = C, \tag{1.1}$$

where $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{m\times m}$, and $C \in \mathbb{R}^{n\times m}$, of low rank $r < \min\{m,n\}$ have recently arisen as a natural algebraic formulation of an increasing number of application problems, such as the discretization of partial differential equations in two or more space variables and also possibly involving time or stochastic variables, the control of discretized dynamical systems (see, e.g., [1], [3], [4], [13], and [32] for more examples of these applications), but also image processing, statistics, and inverse problems in general; see, e.g., [38], [22]. The occurrence of more than two terms, that is $l > 2$, makes the numerical solution particularly challenging, and this led authors to abandon this formulation in the early days [2] or to approach the matrix equation (1.1) mainly from a purely theoretical view point [20]. In the past decade, numerical methods specifically tailored to the solution of (1.1) have successfully emerged, principally following two distinct directions. One class of methods aims to adapt vector approaches to the matrix setting, trying to exploit possible rank structure of the data: these are Krylov subspace methods (see, e.g., [17], [18], [34], [28], and their references), fixed point type iterations [6], [19], low-rank updates [16], etc. In the other direction, reduction techniques have been designed specifically for (1.1) that try to generalize successful

*Received by the editors February 17, 2022; accepted for publication (in revised form) by M. Stoll November 1, 2022; published electronically March 17, 2023.
https://doi.org/10.1137/22M147880X
Funding: The first author is a member of INdAM-GNCS and gratefully acknowledges its support. The work of the second author was partially supported by the China Scholarship Council grant 201906180033, the National Natural Science Foundation of China grants 11471150 and 12161030, and the Hainan Provincial Natural Science Foundation of China grant 121RC537. This work was started during the second author's visit at the Università di Bologna, Italy, August 2019–February 2021.
†Dipartimento di Matematica and AM^2, Alma Mater Studiorum - Università di Bologna, Piazza di Porta S. Donato, 5, I-40127 Bologna, Italy, IMATI-CNR, Pavia, and IAC-CNR, Bari, Italy (valeria.simoncini@unibo.it).
‡High Performance Computing Center, Institute of Applied Physics and Computational Mathematics, Beijing, 100088, China (hao_yue1993@163.com).

methods recently developed for the case $l = 2$; see, e.g., [1], [32]. Approaches that mix these two categories have also been explored; see, e.g., [4], [24], [31]. A convenient hypothesis when aiming at developing a structure-driven method is the fact that C is low rank. Krylov solvers can exploit low rank by including rank constraints. Though methods relying on fixed rank constraints have been developed [36], most solvers impose these constraints dynamically, as the iterations proceed, by applying a truncation procedure to control rank growth.

A convergence analysis of problem (1.1) has only recently been tackled (see, e.g., [15] in the symmetric case). Unfortunately, in its full generality, the theoretical treatment of problem (1.1) is still out of reach. Rank truncation immediately destroys mathematical properties such as global minimization and orthogonality relations. The amount of this damage depends on the type and strictness of the truncation criterion used. If truncation is based on some error norm associated with a specific tolerance, a practical rule of thumb consists in relating this tolerance to the final desired accuracy. However, the whole convergence history can be affected by truncation, especially when the properties of the original methods are not imposed explicitly, as is the case with the conjugate gradient (CG) method. For this algorithm, in exact arithmetic certain orthogonality properties among all generated vectors are satisfied once orthogonality is enforced only locally. Truncation destroys this orthogonality irreparably, eventually leading to stagnation of the whole process (we refer to this low-rank version of CG as "truncated CG" (TCG)). We aim to analyze this striking behavior. Indeed, for a perturbed problem one would expect convergence delay, whereas complete stagnation seems to occur. To be able to analyze in greater detail all quantities involved and have a better handling of the generated spaces, we consider the particular case of (1.1) given by

$$AX + XA + MXM = C, \qquad C = c_1 c_1^\top, \tag{1.2}$$

with A, M, and C symmetric. Nonetheless, many of the presented results are applicable to (1.1) and to linear tensor equations; see, e.g., [17]. To simplify the presentation we will focus on the case when c_1 is a column vector, so that C has rank one.[1] The whole analysis can be generalized to C of (low) rank larger than one. Occasionally we will refer to the case $M = 0$, that is, to the Sylvester equation, to emphasize the new challenges associated with the setting $M \neq 0$. Our theoretical analysis is intended to be a first step towards a better understanding of the performance of iterative methods for solving the general problem in (1.1).

Classically, the problem has been treated by resorting to its Kronecker formulation, giving rise to a standard (vector) linear system. Let $\mathcal{A} = \mathcal{A}_0 + \mathcal{M}$, where

$$\mathcal{A}_0 = A \otimes I + I \otimes A, \quad \mathcal{M} = M \otimes M. \tag{1.3}$$

Then, (1.2) is equivalent to

$$\mathcal{A}x = c, \quad c = \mathrm{vec}(C). \tag{1.4}$$

The vec operator stacks the columns of C one after the other into a single long vector, while for given matrices $H = (h_{ij})_{i=1,\dots,n_H, j=1,\dots,m_H}$, and $B \in \mathbb{R}^{n_B \times m_B}$, the Kronecker product is defined as [14]

[1] We work with the generic nonzero symmetric case and excluding the trivial settings $A = I$ or $M = I$.

$$H \otimes B = \begin{bmatrix} h_{11}B & h_{12}B & \cdots & h_{1m_H}B \\ h_{21}B & h_{22}B & \cdots & h_{2m_H}B \\ \vdots & & & \vdots \\ h_{n_H1}B & h_{n_H2}B & \cdots & h_{n_Hm_H}B \end{bmatrix} \in \mathbb{R}^{n_Hn_B \times m_Hm_B}. \tag{1.5}$$

Throughout the paper we assume that A and M are such that $\mathcal{A}$ is symmetric and positive definite. In the following we denote with $X^\star$ the exact solution to (1.2) and $x^\star = \mathrm{vec}(X^\star)$ the exact solution to (1.4), and we refer to the left-hand side operator as $\mathcal{L}(X) = AX + XA + MXM$, where $\mathcal{L} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$. Though the vector formulation (1.4) can take advantage of a large number of solution strategies, it is now recognized that this form may be unable to preserve some important structural properties of the original matrix equation. Indeed, in addition to possibly large memory requirements, the vector-oriented formulation does not take into account features of the solution matrix $X^\star$ such as numerical low rank[2] and symmetry. These crucial arguments have motivated the large recent interest in developing tailored procedures that can control memory allocations while preserving structural features. This can be achieved by working directly with data in their original context, so that X is treated as a matrix throughout the computation, possibly in factored form.

For $M = 0$ well-established solution methods exist; see [32] for both the large and small scale problems. In particular, for modest matrix dimensions, the solution can be obtained in closed form using a Schur decomposition of A without resorting to the Kronecker formulation. Adding the term MXM with $M \neq 0$ to the matrix equation makes the solution extremely challenging. Except for special cases, no methods exist in the current literature that generalize Schur-based decompositions. Iterative methods thus gain a central role.

Little is known even on the properties of the solution $X^\star$. For instance, the rank of the symmetric solution matrix $X^\star$ is not known a priori. Estimates can be obtained on the decay of the singular values of $X^\star$, that is, of the absolute values of its eigenvalues. Let λ_i, $i = 1, \ldots, n$, be the eigenvalues of $X^\star$, decreasingly ordered in absolute value. Thanks to the properties of the spectral norm, if $\widetilde{X}$ is a rank-m symmetric approximation to $X^\star$, it follows that

$$|\lambda_{m+1}| = \min_{\substack{X \in \mathbb{R}^{n\times n} \\ \mathrm{rank}(X)=m}} \|X^\star - X\| \leq \|X^\star - \widetilde{X}\|, \tag{1.6}$$

where $\|\cdot\|$ is the matrix norm induced by the Euclidean vector norm, namely for $A \in \mathbb{R}^{n\times m}$, $\|A\| = \max_{x\in\mathbb{R}^m, \|x\|=1} \|Ax\|$. In the following, the symbol $\|\cdot\|$ with no subscript stands for the Euclidean norm for vectors, and for the corresponding induced norm for matrices; additional norms will be defined in the next section. The relations in (1.6) say that the error norm $\|X^\star - \widetilde{X}\|$ provides a, not necessarily sharp, upper bound for the $(m+1)$st singular value of $X^\star$. The bound in (1.6) was used in [30] together with classical convergence results for the solution $X^{(k)}$ obtained after k iterations of the ADI method (see, e.g., [37]) to derive upper bounds for the spectral decay behavior of $X^\star$. On the other way around, given a rank-m matrix $\widetilde{X}$, (1.6) indicates that the error norm $\|X^\star - \widetilde{X}\|$ cannot go below $|\lambda_{m+1}|$, the best $\widetilde{X}$ being the one whose spectral decomposition matches that of the first m eigenpairs of $X^\star$. In the following we assume that $X^\star$ can be well approximated by a low-rank matrix.

[2]The *numerical* rank of a matrix is the number of singular values that are above the unit round-off of the considered computational environment.

The existence of such low-rank approximation has been analyzed, for instance, in [1], under various hypotheses on the data.

Taking into account our previous discussion and assuming $\mathcal{A}$ is symmetric and positive definite, a matrix-oriented version of the CG method can be considered [18]. The approach minimizes the error $X^\star - \widetilde{X}$ in a suitable norm, where $\mathrm{vec}(\widetilde{X})$ belongs to a Krylov subspace of growing dimension. This constrained minimization tries to comply with (1.6), though the approximate solution is not designed to have a prescribed numerical rank. In fact, for a zero initial guess it has been observed (see, e.g., [15]) that the numerical rank of the CG approximate solution $\widetilde{X}$ tends to increase, as the iterations proceed, and then to decrease to the final numerical rank as convergence takes place. Hence, how CG behaves as a "matrix-oriented" algorithm provides a new, different perspective, compared to well-established results for the vector setting [21].

Our first aim is to deepen our understanding of the numerical rank evolution of the CG approximate solution $\widetilde{X}$, and to characterize the approximation spaces where $\widetilde{X}$ lives. In particular, this analysis is relevant in the understanding of truncation strategies applied to matrix-oriented CG, in which all matrix iterates are explicitly kept low rank, thus stored in factored form, by truncating the terms that would lead to the rank increase. Moreover, we study loss of orthogonality among computed quantities such as residuals: we derive an inverse proportionality relation between the (perturbed) orthogonality angle and the current residual norm at each iteration, showing that stagnation will occur as soon as the residuals lose their linear independence.

Our second aim is to compare the matrix-oriented CG (without truncation) for (1.2) with a method that explicitly and iteratively builds a low-rank matrix, of increasing rank, and minimizes the error norm in some approximation space by means of a Galerkin condition. We show that a specific choice of approximation space allows one to relate this approximation problem with that of CG, highlighting the (dis)advantages of either approach.

We illustrate our findings with tailored small matrices, whose structures highlight the problems we discuss. We point to the previously cited recent literature for rich experimental evidence with large dimensional equations stemming from various applications. Nonetheless, we stress that even a modest value of n, say a few hundreds, will lead to a vector problem of square the dimension—thus already quite sizable.

1.1. Notation and main definitions. In the following real matrices will be used, and $A^\top$ will denote the transpose of a matrix A. The Frobenius norm of an $n \times m$ real matrix, $\|A\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m A_{i,j}^2$, will be used, together with the matrix norm induced by the Euclidean vector norm, $\|A\| = \max_{x \in \mathbb{R}^m, \|x\|=1} \|Ax\|$, already introduced in the previous section. Given a symmetric and positive definite $n \times n$ matrix A, the A-norm or energy norm is defined as $\|x\|_A^2 = x^\top A x$, where $0 \neq x \in \mathbb{R}^n$. An operator-based energy norm will also be introduced. Exact arithmetic will be assumed throughout.

We will continuously rely on the correspondence between matrix-matrix and vector operations, obtained via the Kronecker product and vec operator. We will freely make use of properties of the Kronecker product, as reported for instance in [14]. We also define the matrix inner product of two matrices $Y, Z \in \mathbb{R}^{n \times m}$ as $\langle Y, Z \rangle = \mathrm{trace}(Y^\top Z)$, and we observe that this corresponds to the vector inner product, that is, $\langle Y, Z \rangle = \mathrm{vec}(Y)^\top \mathrm{vec}(Z)$. In case the two matrices have low rank (here $n = m$), that is, $Y = U_Y V_Y^\top$, $Z = U_Z V_Z^\top$ with $U_Y, V_Y \in \mathbb{R}^{n \times k_Y}$, and $U_Z, V_Z \in \mathbb{R}^{n \times k_Z}$, this inner product can be computed by

$$\text{trace}\left(Y^\top Z\right) = \text{trace}\left(V_Y U_Y^\top U_Z V_Z^\top\right) = \text{trace}\left((U_Y^\top U_Z)(V_Z^\top V_Y)\right). \tag{1.7}$$

In the following, := means that the quantity on the left of the equality is defined by the quantity on the right. Correspondingly, =: means that the quantity on the right of the equality is defined by the quantity on the left.

We end this section with a note on the matrix problem we have chosen to analyze. Problem (1.2) is strictly related to the following form:

$$A_1 Z B_1 + A_2 Z B_2 + Z = F, \tag{1.8}$$

with A_i, B_i, $i = 1, 2$, symmetric. Indeed, for M nonsingular and positive definite, (1.2) can be brought to this form for $A_1 = M^{-1/2} A M^{-1/2} = B_2$, $B_1 = M^{-1} = A_2$, $F = M^{-1/2} C M^{-1/2}$, and $Z = M^{1/2} X M^{1/2}$. However, the seemingly harmless shift term makes the problem very different from the well-known generalized Sylvester equation $A_1 Z B_1 + A_2 Z B_2 = F$. This form does not seem to provide more insight than the form we consider; hence unless explicitly stated we will focus on (1.2).

2. Matrix-oriented CG method. The matrix-oriented CG method simply transforms all vector computations associated with (1.4) into matrix operations, using the vec and Kronecker operators. So, for instance, for the classical approximate solution update (see, e.g., [7, section 10.2] for the vector CG algorithm), it holds that

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \qquad \Leftrightarrow \qquad X^{(k+1)} = X^{(k)} + \alpha_k P^{(k)},$$

where $x^{(k)} = \text{vec}(X^{(k)})$, $p^{(k)} = \text{vec}(P^{(k)})$, and

$$\alpha_k = \frac{\left(r^{(k)}\right)^\top p^{(k)}}{\left(p^{(k)}\right)^\top \mathcal{A} p^{(k)}} = \frac{\text{trace}\left(\left(R^{(k)}\right)^\top P^{(k)}\right)}{\text{trace}\left(\left(P^{(k)}\right)^\top \mathcal{L}\left(P^{(k)}\right)\right)}, \qquad r^{(k)} = \text{vec}\left(R^{(k)}\right);$$

here $\{p^{(k)}\}_{k\geq 0}$ is the sequence of direction vectors determined during the CG recursion, and $r^{(k)} = c - \mathcal{A} x^{(k)}$ is the residual vector associated with $x^{(k)}$. A complete description of the algorithm is postponed to section 3.

The vector and matrix formulations are mathematically equivalent, though some care in the implementation of the matrix inner product is required to avoid unnecessary operations. In particular, formula (1.7) should be used whenever the matrices are kept in factored form. There may be some computational advantages in the matrix iteration in a high performance computing environment; however, the main reason for pursuing a matrix-oriented version is to maintain the possible low-rank structure of the iterates, by using a factorized form. For instance, if $X^{(k)} = X_1^{(k)} (X_1^{(k)})^\top$ with $X_1^{(k)}$ low rank, and similarly for $P^{(k)}$, then

$$X^{(k+1)} = X_1^{(k)} (X_1^{(k)})^\top + \alpha_k P_1^{(k)} (P_1^{(k)})^\top,$$

which is also low rank, with a rank that is in general larger than that of $X^{(k)}$; a more precise structure will be given in subsection 2.2. In case the rank is forced to remain low, truncation can be implemented by taking the best approximation to $[X_1^{(k)}, P_1^{(k)}]$ of fixed rank, for instance. Clearly, the factored form is meaningful only if the right-hand side C either is low rank or can be well approximated by a low-rank matrix. Indeed, even assuming a zero initial approximation $X^{(0)}$, the residual $R^{(0)} = C - \mathcal{L}(X^{(0)})$ will only be low rank if C is. To appreciate the cost relevance of CG applied to (1.2), we notice that without truncation, each multiplication with $\mathcal{A}$

entails $2n$ multiplications with either A or M. The low-rank matrix implementation of CG that takes into account the symmetry of the iteration matrices allows one to significantly reduce this computation, as long as low rank in the iterates is maintained (e.g., r multiplications by A if the symmetric iterate is kept in factored form of rank r). On the other hand, truncation is not without side effects; we refer the reader to [17], [18], [15], [34], [28] for implementation considerations and computational evidence.

To better understand the impact of truncation, we first need to linger over the analysis of the space generated during the matrix recurrence. More precisely, we identify redundant information, which can be purged with no harm, and important vector elements whose elimination determines a degradation of the method performance.

2.1. Analysis of the error matrix. As TCG iterations proceed, that is, as k increases, two facts have been experimentally observed in the literature (see, e.g., [15]):

(i) Singular triplets of $X^{(k)}$ seem to converge in an orderly fashion to those of X.

(ii) The numerical rank of $X^{(k)}$ increases up to some point; then it decreases.

In the following we analyze the CG optimality properties in the matrix context, and how they influence the above two phenomena. We start by recalling that the direction vectors $\{p_k\}_{k\geq 0}$ determined during the CG recursion iteratively generate the following Krylov subspace (assuming $x^{(0)} = 0$):

$$\mathbb{K}_k = \operatorname{span}\{c, \mathcal{A}c, \ldots, \mathcal{A}^{k-1}c\}. \tag{2.1}$$

The same space is spanned by the residuals, $\{r^{(k)}\}_{k\geq 0}$. Moreover, the error norm is minimized in the energy norm associated with the coefficient matrix, that is,

$$\|x^\star - x^{(k)}\|_{\mathcal{A}} = \min_{x\in\mathbb{K}_k} \|x^\star - x\|_{\mathcal{A}}, \tag{2.2}$$

so that a nonincreasing energy norm of the error is ensured [7, section 10.2].

In matrix terms, we first write the norm equivalence

$$\|X\|_{\mathcal{L}}^2 := \operatorname{trace}\left(X^\top \mathcal{L}(X)\right) = \|x\|_{\mathcal{A}}^2, \tag{2.3}$$

where $\mathcal{L}(X) = AX + XA + MXM$, $\mathcal{A}$ is as defined in (1.4) and the energy norm $\|x\|_{\mathcal{A}}$ is as defined in subsection 1.1. Setting $E^{(k)} = X^\star - X^{(k)}$ we can write

$$\|E^{(k)}\|_{\mathcal{L}}^2 = \|x^\star - x^{(k)}\|_{\mathcal{A}}^2 \geq \lambda_{\min}(\mathcal{A})\|x^\star - x^{(k)}\|^2 = \lambda_{\min}(\mathcal{A})\|E^{(k)}\|_F^2.$$

Therefore,

$$\lambda_{\max}(\mathcal{A})^{-\frac{1}{2}} \min_{x\in\mathbb{K}_{k-1}} \|x^\star - x\|_{\mathcal{A}} \leq \|E^{(k)}\|_F \leq \lambda_{\min}(\mathcal{A})^{-\frac{1}{2}} \min_{x\in\mathbb{K}_{k-1}} \|x^\star - x\|_{\mathcal{A}}.$$

Although this inequality does not imply that the quantity $\|E^{(k)}\|_F$ is minimized, it is clear that as k increases, we expect this Frobenius norm to decrease, eventually going towards zero in exact arithmetic. In light of (1.6) and the fact that $\|E^{(k)}\| \leq \|E^{(k)}\|_F$, this explains that the approximation of $X^{(k)}$ to the matrix X occurs in terms of singular values. As convergence takes place the norm of $X - X^{(k)}$ decreases, that is, the leading singular values of $X^{(k)}$ tend to match those of X. On the other hand, below the level of the error norm the singular values of the two matrices X and $X^{(k)}$ can vary significantly. We next formalize this argument.

Let $X^\star = U\Sigma W^\top$ and $X^{(k)} = \tilde{U}\tilde{\Sigma}\tilde{W}^\top$ be the singular value decompositions (SVDs) of the given matrices.[3] Consider the partitionings

[3]Since $X^\star$ and $X^{(k)}$ are symmetric but not necessarily semidefinite, we shall work with singular values rather than eigenvalues.

$$X^\star = [U_1, U_2] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} W_1^\top \\ W_2^\top \end{bmatrix}, \qquad X^{(k)} = [\tilde{U}_1, \tilde{U}_2] \begin{bmatrix} \tilde{\Sigma}_1 & \\ & \tilde{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \tilde{W}_1^\top \\ \tilde{W}_2^\top \end{bmatrix},$$

with Σ_1, and $\tilde{\Sigma}_1$ of size $\ell \times \ell$. Then $E^{(k)} = X^\star - \tilde{U}_1\tilde{\Sigma}_1\tilde{W}_1^\top - \tilde{U}_2\tilde{\Sigma}_2\tilde{W}_2^\top$, so that

$$\left| \|X^\star - \tilde{U}_1\tilde{\Sigma}_1\tilde{W}_1^\top\| - \|\tilde{U}_2\tilde{\Sigma}_2\tilde{W}_2^\top\| \right| \le \|E^{(k)}\|. \tag{2.4}$$

Therefore, the distance between the leading singular triplets of $X^{(k)}$ and those of $X^\star$ is not larger than $\|E^{(k)}\|$ from $\|\tilde{\Sigma}_2\|$. We stress here that the SVD of $X^{(k)}$ is computable, so that one can monitor $\|\tilde{\Sigma}_2\|$. Moreover, $X^{(k)}$ is not assumed to be of rank ℓ; therefore, the analysis based on this partitioning can also be used for increasing rank of $X^{(k)}$. If the partitioning is selected so that $\|\tilde{\Sigma}_2\| \ll \|E^{(k)}\|$, then the inequality above shows that the approximation of the leading triplets of $X^{(k)}$ must be of the order of $\|E^{(k)}\|$. We next formalize this intuition by using a result of Wedin (see, e.g., [33, Theorem V.4.4]). To this end, let $\rho_{r,k} = \|X^\star\tilde{W}_1 - \tilde{U}_1\tilde{\Sigma}_1\|$ and $\rho_{l,k} = \|X^\star\tilde{U}_1 - \tilde{W}_1\tilde{\Sigma}_1\|$. Clearly, $\rho_{r,k} = \|E^{(k)}\tilde{W}_1\| \le \|E^{(k)}\|$ and $\rho_{l,k} = \|E^{(k)}\tilde{U}_1\| \le \|E^{(k)}\|$; hence, both quantities decrease as the error norm does.

THEOREM 2.1. [33, Theorem V.4.4]. *If there exist* $\delta, \alpha > 0$ *such that* $\max \sigma(\Sigma_2) \le \alpha$ *and* $\min \sigma(\tilde{\Sigma}_1) \ge \delta + \alpha$, *then*

$$\max\{\|\sin\Phi\|, \|\sin\Theta\|\} \le \frac{\max\{\rho_{r,k}, \rho_{l,k}\}}{\delta},$$

where Φ *and* Θ *are the matrices of canonical angles between range*(U_1) *and range*$(\tilde{U}_1)$, *and between range*(W_1) *and range*$(\tilde{W}_1)$, *respectively.*

Clearly, in our setting $\Phi = \Theta$. The quantity δ measures how the gap between the converging singular values and the remaining ones influences the actual convergence. This result explains the orderly convergence to the singular triplets of $X^\star$ in item (i) above, as $\|E^{(k)}\|$ decreases.

Example 2.2. Let c_1 be the vector of all ones normalized to have unit norm, and let $A = \texttt{tridiag}(-1, \underline{2}, -1) \in \mathbb{R}^{n\times n}$, $M = \texttt{pentadiag}(-0.5, -0.5, \underline{2.5}, -0.5, -0.5) \in \mathbb{R}^{n\times n}$, and $n = 25$. Matrix-oriented preconditioned CG is employed, with $M \otimes M$ as preconditioner, so that the Kronecker structure of the preconditioned problem is maintained. For each of the first 12 iterations, Figure 1 displays the singular values of $X^\star$ and of $X^{(k)}$, and also the level corresponding to $\|E^{(k)}\|$. As expected, the singular values of $X^{(k)}$ above the error norm level tend to match the corresponding singular values of $X^\star$. What is more noteworthy is that below the error norm level, the discrepancy between the singular values of $X^{(k)}$ and of $X^\star$ is significant, and in practice, clusters of slowly varying singular values can occur for $X^{(k)}$. Below the error level we do not expect the approximate singular values to have the same decay as the exact ones. In fact, since the rank of the iterates may significantly increase at each iteration, the number of nonzero singular values quickly increases, and the singular values have sizable magnitude until $\|E^{(k)}\|$ is sufficiently small.

The previous example illustrates the phenomenon in the item (ii) above. Numerous singular values with magnitude below the error threshold but above the unit round-off emerge as iterations proceed, so that the numerical rank of $X^{(k)}$ grows. As more and more singular values of small magnitude converge, the remaining smaller singular values are necessarily constrained to go towards zero, so that the numerical rank of $X^{(k)}$ decreases towards its final value.

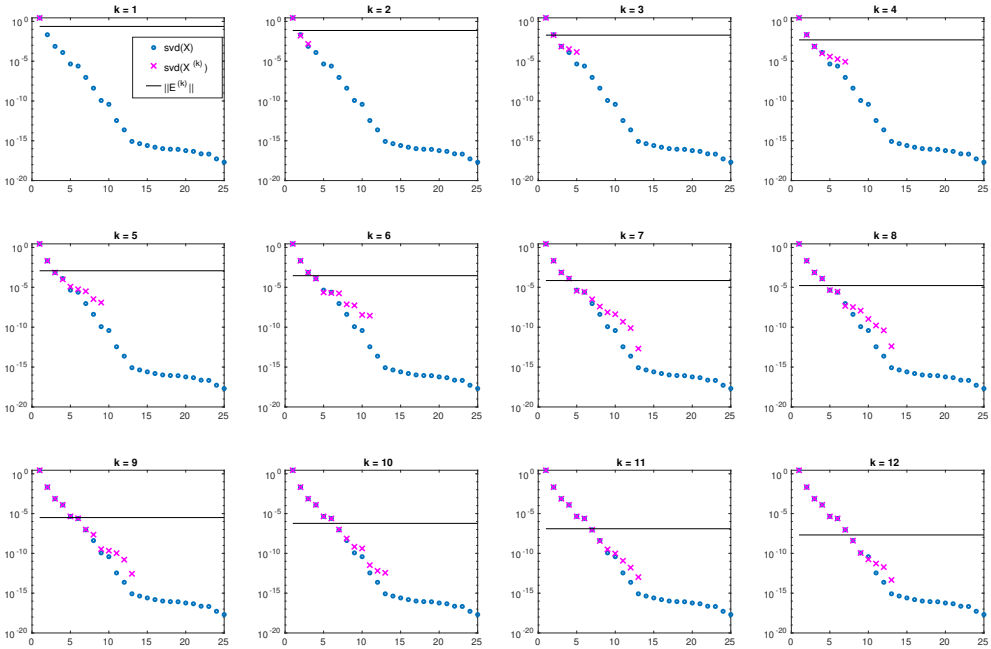

FIG. 1. *Example* 2.2. *Singular values of* $X^\star$ *and of* $X^{(k)}$ *and error threshold for each of the first* 12 *iterations.*

2.2. The CG matrix approximation space. In this section we characterize the matrix approximation space, that is, the vector space containing range$(X^{(k)})$, associated with the matrix-oriented CG method.

Had we $M = 0$ the analysis would simplify, as the following result holds.

PROPOSITION 2.3. *Assume* $M = 0$ *and let* $q \in \mathbb{K}_k$. *For some* $\alpha_0, \dots, \alpha_k \in \mathbb{R}$,

$$q = \sum_{i=0}^{k} \alpha_i \sum_{j=0}^{i} \binom{i}{j} (A^j \otimes A^{i-j}) c = \sum_{0 \le j \le i \le k} \alpha_i \binom{i}{j} (A^j \otimes A^{i-j}) c.$$

Proof. Let $\mathcal{A} = I \otimes A + A \otimes I =: \mathcal{A}_1 + \mathcal{A}_2$ with $\mathcal{A}_1, \mathcal{A}_2$ commuting. We have that $q = \sum_{i=0}^{k} \alpha_i \mathcal{A}^i c$. It holds that $\mathcal{A}^i = (\mathcal{A}_1 + \mathcal{A}_2)^i = \sum_{j=0}^{i} \binom{i}{j} \mathcal{A}_1^{i-j} \mathcal{A}_2^j$, with $\mathcal{A}_1^{i-j} \mathcal{A}_2^j = (I \otimes A)^{i-j} (A \otimes I)^j = (I \otimes A^{i-j})(A^j \otimes I)$. The result follows. ∎

For general nonzero symmetric M the description is more complex. In the following we consider the generic case, where M is full rank and its norm is large enough to make the contribution of the term MXM relevant for the discussion. The matrix $\mathcal{A}$ in (1.4) can be written as $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{M}$ with $\mathcal{M} = M \otimes M$. Clearly, $\mathcal{M}$ does not commute with either other matrix, except in special circumstances.

Assume that $X^{(0)} = 0 := X_1^{(0)} X_1^{(0)\top}$, $R^{(0)} := R_1^{(0)} R_1^{(0)\top}$ with $R_1^{(0)} = c_1$, and $R^{(0)} = P^{(0)} =: P_1^{(0)} P_1^{(0)\top}$, and that at the kth iteration $X^{(k)}$, $R^{(k)}$, and $P^{(k)}$ can be written as $X^{(k)} = X_1^{(k)} G^{(k)} X_1^{(k)\top}$, $R^{(k)} = R_1^{(k)} S^{(k)} R_1^{(k)\top}$, and $P^{(k)} = P_1^{(k)} D^{(k)} P_1^{(k)\top}$, respectively. Then for the $(k+1)$th iteration, there hold (see, e.g., [1] for similar expressions)

(2.5)
$$\begin{aligned}\boxed{X^{(k+1)}} &= X^{(k)} + \alpha_k P^{(k)} = X_1^{(k)} G^{(k)} {X_1^{(k)}}^\top + \alpha_k P_1^{(k)} D^{(k)} {P_1^{(k)}}^\top \\ &= [X_1^{(k)}\ P_1^{(k)}] \begin{bmatrix} G^{(k)} & 0 \\ 0 & \alpha_k D^{(k)} \end{bmatrix} [X_1^{(k)}\ P_1^{(k)}]^\top =: \boxed{X_1^{(k+1)} G^{(k+1)} {X_1^{(k+1)}}^\top};\end{aligned}$$

(2.6)
$$\begin{aligned}\boxed{R^{(k+1)}} &= C - \mathcal{L}(X^{(k+1)}) \\ &= c_1 c_1^\top - A X_1^{(k+1)} G^{(k+1)} {X_1^{(k+1)}}^\top - X_1^{(k+1)} G^{(k+1)} {X_1^{(k+1)}}^\top A^\top \\ &\quad - M X_1^{(k+1)} G^{(k+1)} {X_1^{(k+1)}}^\top M^\top \\ &= [c_1\ AX_1^{(k+1)}\ X_1^{(k+1)}\ MX_1^{(k+1)}] \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & -G^{(k+1)} & 0 \\ 0 & -G^{(k+1)} & 0 & 0 \\ 0 & 0 & 0 & -G^{(k+1)} \end{bmatrix} \\ &\quad \cdot [c_1\ AX_1^{(k+1)}\ X_1^{(k+1)}\ MX_1^{(k+1)}]^\top =: \boxed{R_1^{(k+1)} S^{(k+1)} {R_1^{(k+1)}}^\top};\end{aligned}$$

(2.7)
$$\begin{aligned}\boxed{P^{(k+1)}} &= R^{(k+1)} + \beta_k P^{(k)} \\ &= [R_1^{(k+1)}\ P_1^{(k)}] \begin{bmatrix} S^{(k+1)} & 0 \\ 0 & \beta_k D^{(k)} \end{bmatrix} [R_1^{(k+1)}\ P_1^{(k)}]^\top =: \boxed{P_1^{(k+1)} D^{(k+1)} {P_1^{(k+1)}}^\top}.\end{aligned}$$

By replacing the factors in the recurrence, we obtain[4]

$$\begin{aligned}X^{(k+1)} = [X_1^{(k-1)}\ P_1^{(k-1)}\ R_1^{(k)}\ P_1^{(k-1)}] &\mathrm{blkdiag}(G^{(k-1)}, \alpha_{k-1} D^{(k-1)}, \alpha_k S^{(k)}, \beta_{k-1}\alpha_k D^{(k-1)}) \\ &\cdot [X_1^{(k-1)}\ P_1^{(k-1)}\ R_1^{(k)}\ P_1^{(k-1)}]^\top,\end{aligned}$$

which shows that $X^{(k+1)}$, and thus, $X_1^{(k+1)}$, is naturally rank-deficient in this form. Similarly, we obtain that $P_1^{(k+1)}$ is also naturally rank-deficient. Moreover, we have

$$\begin{aligned}X_1^{(1)} &= \left[X_1^{(0)}\ P_1^{(0)}\right] = c_1, \\ R_1^{(1)} &= \left[c_1\ AX_1^{(1)}\ X_1^{(1)}\ MX_1^{(1)}\right] = [c_1\ Ac_1\ c_1\ Mc_1], \\ P_1^{(1)} &= \left[R_1^{(1)}\ P_1^{(0)}\right] = [c_1\ Ac_1\ c_1\ Mc_1\ c_1], \\ X_1^{(2)} &= \left[X_1^{(1)}\ P_1^{(1)}\right] = [c_1\ c_1\ Ac_1\ c_1\ Mc_1\ c_1],\end{aligned}$$

so that $\mathrm{range}(R_1^{(1)}), \mathrm{range}(P_1^{(1)}), \mathrm{range}(X_1^{(2)}) \subseteq \mathrm{span}\{c_1\ Ac_1\ Mc_1\}$, and

$$R_1^{(2)} = \left[c_1\ AX_1^{(2)}\ X_1^{(2)}\ MX_1^{(2)}\right], \qquad P_1^{(2)} = \left[R_1^{(2)}\ P_1^{(1)}\right] = \left[R_1^{(2)}\ c_1\ Ac_1\ c_1\ Mc_1\ c_1\right],$$

so that $\mathrm{range}(R_1^{(2)}), \mathrm{range}(P_1^{(2)}) \subseteq \mathrm{span}\{c_1\ Ac_1\ Mc_1\ A^2c_1\ AMc_1\ MAc_1\ M^2c_1\}$. By induction, we can see that the rank of $X_1^{(k+1)}$ is the same as that of $P_1^{(k)}$, and the rank of $P_1^{(k)}$ is the same as that of $R_1^{(k)}$. The relations above also show that $R_1^{(1)}$, $P_1^{(1)}$, and $X_1^{(2)}$ are all rank deficient. In other words, although the number of columns grows in

[4] blkdiag defines a block diagonal matrix with input arguments as the diagonal blocks.

the block, the actual rank of the block is lower than the number of computed columns in this form.

Let $Q^{(1)} = [c_1]$, and define the matrix sequence

$$Q^{(k+1)} = \left[Q^{(k)}, AQ^{(k)}, MQ^{(k)}\right],$$

so that $Q^{(2)} = [c_1, Ac_1, Mc_1]$ and $Q^{(3)} = [c_1, Ac_1, Mc_1, A^2c_1, AMc_1, MAc_1, M^2c_1]$. Hence, $R_1^{(k)}, P_1^{(k)} \in \text{range}(Q^{(k+1)})$. As k increases, the columns of $Q^{(k)}$ increasingly build the space[5]

$$\begin{aligned}\mathbb{Q} = \text{span}\big\{ & c_1, \underbrace{Ac_1, Mc_1}, \underbrace{A^2c_1, AMc_1, MAc_1, M^2c_1}, \\ & \underbrace{A^3c_1, A^2Mc_1, AMAc_1, AM^2c_1, MA^2c_1, MAMc_1, M^2Ac_1, M^3c_1}, \cdots \big\},\end{aligned} \tag{2.8}$$

and we denote with $\mathbb{Q}_k$ the smallest subspace of $\mathbb{Q}$ containing the range of $Q^{(k)}$. Thus, we have ($\dim(\mathbb{Q}_{k+1})$ denotes the space dimension of $\mathbb{Q}_{k+1}$)

$$\dim(\mathbb{Q}_{k+1}) \le \dim(\mathbb{Q}_k) + 2^k,$$

that is, the space dimension may grow exponentially, up to its maximum dimension n^2. Let the columns of $\mathscr{Q}^{(k)}$ span $\mathbb{Q}_k$. Then we can write $X_1^{(k)} = \mathscr{Q}^{(k)}\tilde{G}^{(k)}$ for some $\tilde{G}^{(k)}$, so that

$$X^{(k)} = \mathscr{Q}^{(k)}\tilde{G}^{(k)}G^{(k)}\left(\tilde{G}^{(k)}\right)^\top (\mathscr{Q}^{(k)})^\top =: \mathscr{Q}^{(k)}\mathscr{G}^{(k)}\left(\mathscr{Q}^{(k)}\right)^\top. \tag{2.9}$$

This decomposition provides the most genuine low-rank approximation from the generated space. However, it should be stressed that $X^{(k)}$ is not *any* linear combination of the columns of $\mathscr{Q}^{(k)}$. Indeed, for instance, the product $\mathcal{A}c$ yields a special linear combination of $\{c_1, Ac_1, Mc_1\}$, enforcing a constraint on the approximation. This argument generalizes what we have seen for the vector q in Proposition 2.3 to the case of $M \neq 0$. In other words, $X_1^{(k)}$ belongs to a proper subspace of $\mathscr{Q}^{(k)}$, with a possibly much smaller dimension.

We also mention that the CG iteration is unable to capture the underlying matrix $\mathscr{Q}^{(k)}$, so that any truncation strategy directly performed on the next iterate $X^{(k+1)}$ or its factor $X_1^{(k+1)}$ in (2.5) is bound to lose part of the information contained in $\mathbb{Q}_k$.

The important role played by $\mathscr{Q}^{(k)}$ leads one to consider ways to exploit this matrix in a more effective way, without the redundancy created by $\mathbb{Q}_k$. This is possible by using approximation methods directly applied to the original matrix equation, which consists of projecting the solution matrix onto an appropriate subspace. In section 4 we show that an appropriate space is indeed the one generated by $\mathscr{Q}^{(k)}$.

The effect of preconditioning. To speed up convergence, the system $\mathcal{A}x = c$ is usually preconditioned; this was done in Example 2.2, for instance. Hence, a matrix/operator $\mathcal{P}$ is selected and the problem $\mathcal{P}^{-1}\mathcal{A}x = \mathcal{P}^{-1}c$ solved; see, e.g., [7, section 10.3] for a symmetry preserving implementation in the vector case. The operation performed to obtain (1.8) corresponds to preconditioning by $\mathcal{P} = \mathcal{M} = M \otimes M$. For instance, if M is positive definite, the matrix sequence becomes

$$Q^{(1)} = \left[M^{-\frac{1}{2}}c_1\right], \qquad Q^{(k+1)} = \left[Q^{(k)}, M^{-\frac{1}{2}}AM^{-\frac{1}{2}}Q^{(k)}, M^{-\frac{1}{2}}Q^{(k)}\right],\ k = 0, 1, \ldots.$$

[5]Brackets indicate the block of (independent) newly added vectors at each iteration.

Our analysis still holds as long as the preconditioning operator maintains the Kronecker structure in the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$. Following corresponding analyses performed for fixed point iterations (see, e.g., [4]), a natural preconditioned problem is given as

$$X + \mathcal{L}_A^{-1}(MXM) = \mathcal{L}_A^{-1}(C), \qquad \mathcal{L}_A = AX + XA.$$

This procedure is clearly equivalent to the preconditioned problem $\mathcal{A}_0^{-1}\mathcal{A}x = \mathcal{A}_0^{-1}c$ with $\mathcal{A}_0^{-1}\mathcal{A} = I + \mathcal{A}_0^{-1}\mathcal{M}$, where $\mathcal{A}_0$ and $\mathcal{M}$ are as in (1.3).

3. The truncated CG method. In this section, we analyze TCG for solving the matrix equation (1.2). The formal algorithm with truncation, already presented in [18, Alg. 2], is reported in Algorithm 3.1, where $\mathcal{T}$ is the truncation operator. Truncation is performed at lines 5 and 8, whereas for the residual and operator products they are optional. We did not adopt this option in our numerical experiments, to ensure the accurate computation of the residual and the application of the coefficient matrices. Following the discussion in the previous section, matrices are kept in factored form, so that truncation is performed by reducing the rank of the factors $X_1^{(k+1)}$ and $P_1^{(k+1)}$, respectively.

It is experimentally evident that the final attainable accuracy, in terms of relative residual norm, is strictly related to the truncation tolerance; see, e.g., various plots reported in [17]. It particular, it may well occur that the method stagnates at a level above the desirable accuracy. For these reasons, stopping criteria in addition to the relative residual norm should be considered, such as maximum number of iterations and maximum approximation rank.

Low-rank truncation. Different ways to truncate the factorized representation of a matrix $Y = Y_1 Y_2^\top$ can be considered. A simple strategy amounts to fixing a maximum rank equal to k. In this case, the most relevant subspace of dimension k spanned by the columns of Y_1 and Y_2 will be kept. This, however, may dramatically deteriorate the approximation (see section 2.1), unless the sought after solution can be well approximated by a rank-k matrix.

Algorithm 3.1 TCG algorithm for the matrix equation (1.2).

Input: Matrix function $\mathcal{L}$: $\mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, right-hand side $C \in \mathbb{R}^{n\times n}$ in low-rank format. Truncation operator $\mathcal{T}$.

Output: Matrix $X \in \mathbb{R}^{n\times n}$ approximating exact X^*.

1: $X^{(0)} = 0$, $R^{(0)} = C$, $P^{(0)} = R^{(0)}$, $J^{(0)} = \mathcal{L}(P^{(0)})$
2: $\xi_0 = \langle P^{(0)}, J^{(0)} \rangle$, $k = 0$
3: **while** $||R^{(k)}||_F > tol$ do **do**
4: $\quad \alpha_k = \langle R^{(k)}, P^{(k)} \rangle / \xi_k$
5: $\quad X^{(k+1)} = X^{(k)} + \alpha_k P^{(k)}$, $\quad X^{(k+1)} \leftarrow \mathcal{T}(X^{(k+1)})$
6: $\quad R^{(k+1)} = C - \mathcal{L}(X^{(k+1)})$, $\quad$ Optionally: $R^{(k+1)} \leftarrow \mathcal{T}(R^{(k+1)})$
7: $\quad \beta_k = -\langle R^{(k+1)}, J^{(k)} \rangle / \xi_k$
8: $\quad P^{(k+1)} = R^{(k+1)} + \beta_k P^{(k)}$, $\quad P^{(k+1)} \leftarrow \mathcal{T}(P^{(k+1)})$
9: $\quad J^{(k+1)} = \mathcal{L}(P^{(k+1)})$, $\quad$ Optionally: $J^{(k+1)} \leftarrow \mathcal{T}(J^{(k+1)})$
10: $\quad \xi_{k+1} = \langle P^{(k+1)}, J^{(k+1)} \rangle$
11: $\quad k = k + 1$
12: **end while**
13: $X = X^{(k)}$

An error-aware truncation strategy consists of selecting $Y' = Y_1'(Y_2')^\top$, with $Y_1', Y_2' \in \mathbb{R}^{n\times r}$ such that the error matrix $Y - Y'$ is smaller than a threshold, in some relative norm. Nonetheless, a maximum rank value can also be included, so as to limit memory consumption. Let $\sigma_1, \dots, \sigma_r$ be the singular values of Y. The truncation rank $\widetilde{r} \le r$ is the smallest integer satisfying a specified truncation criterion. The following standard criterion based on singular values can be considered:

$$\left(\sigma_{\widetilde{r}+1}^2 + \cdots + \sigma_r^2\right)^{\frac{1}{2}} \le \epsilon_{trunc} \left(\sigma_1^2 + \cdots + \sigma_r^2\right)^{\frac{1}{2}}.$$

To determine the singular values and the new factors, we first compute the reduced (skinny) QR factorization of Y_1 and Y_2, that is, such that $Y_1 = Q_1 R_1, Y_2 = Q_2 R_2$ with upper triangular matrices $R_1, R_2 \in \mathbb{R}^{r\times r}$. Then the SVD

$$R_1 R_2^\top = U \mathrm{diag}(\sigma_1, \dots, \sigma_r) V^\top$$

is computed. Let $\Sigma_{\widetilde{r}} = \mathrm{diag}(\sigma_1, \dots, \sigma_{\widetilde{r}})$. Using MATLAB notation, we then set

$$Y_1' = Q_1 U_{:,1:\widetilde{r}} \Sigma_{\widetilde{r}}^{\frac{1}{2}}, \qquad Y_2' = Q_2 V_{:,1:\widetilde{r}} \Sigma_{\widetilde{r}}^{\frac{1}{2}},$$

and then we obtain the compressed low-rank matrix $Y = Y_1'(Y_2')^\top$. Denoting by $X_{ex}^{(k)}$, $P_{ex}^{(k)}$ the (exact, in exact arithmetic) matrices before truncation, we can write

$$X^{(k)} = \mathcal{T}\left(X_{ex}^{(k)}\right) = X_{ex}^{(k)} + E_X^{(k)}, \quad P^{(k)} = \mathcal{T}(P_{ex}^{(k)}) = P_{ex}^{(k)} + E_P^{(k)},$$

where, by using, for instance, the first truncation strategy, we obtain

$$\frac{||E_X^{(k)}||_F}{||X^{(k)}||_F} < \epsilon_{trun} \quad \text{and} \quad \frac{||E_P^{(k)}||_F}{||P^{(k)}||_F} < \epsilon_{trun}. \tag{3.1}$$

Moreover, in the following we assume that the residual is computed explicitly and not by a recurrence, and that it is not truncated, so that the following holds:

$$\begin{aligned} R^{(k)} &= C - \mathcal{L}\left(X^{(k)}\right) = C - \mathcal{L}\left(X^{(k-1)} + \alpha_{k-1} P^{(k-1)} + E_X^{(k)}\right) \\ &= R^{(k-1)} - \alpha_{k-1} \mathcal{L}\left(P^{(k-1)}\right) - \mathcal{L}\left(E_X^{(k)}\right). \end{aligned}$$

In the following section we deepen our understanding of this loss of orthogonality and provide insight into how the convergence of the truncated version of the method can behave in practice.

Finally, we observe that the truncation strategy may be viewed as a way to reduce the approximation space $\mathbb{Q}_k$ associated with the space in (2.8). However, this truncation strategy is not based on spectral information typically employed in classical subspace enhancements associated with the standard Krylov subspace, such as those in [5], [26], for instance. By specifically focusing on the use of this space, it may be possible to derive more effective truncation strategies; we leave this topic to future investigation.

3.1. Effects of truncation in the CG recurrence. In this section we analyze the effect of truncation on the iterates of the CG recurrence, at each iteration k. We are able to identify the quantities involved in the determination of the final attainable residual norm of the method. For the sake of the presentation, we are going to switch to the vector formulation, which in addition makes the derivation more familiar to

anyone who has seen the standard properties of CG. We also stress that the results of this section apply to the more general equation (1.1) and to tensor linear equations, as only references to the whole matrix $\mathcal{A}$ are made.

We start by introducing some notation. Let $r_{ex}^{(k)} = \mathcal{C} - \mathcal{A}x_{ex}^{(k)}$ be the residual computed by the exact (untruncated) solution iterate $x_{ex}^{(k)}$, and $r^{(k)} = \mathcal{C} - \mathcal{A}x^{(k)}$ be the residual computed by the truncated iterate. Here and in the following, the subscript "ex" denotes vectors before (or without) truncation. At iteration k, the exact vector CG recurrences are given by

$$x_{ex}^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad \alpha_k = \left(r^{(k)}\right)^\top p^{(k)} / \left((p^{(k)})^\top \mathcal{A} p^{(k)}\right),$$
$$p_{ex}^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}, \quad \beta_k = -\left(r^{(k+1)}\right)^\top \mathcal{A} p^{(k)} / \left((p^{(k)})^\top \mathcal{A} p^{(k)}\right).$$

Then,

$$x^{(k+1)} = x_{ex}^{(k+1)} + \boldsymbol{e}_X^{(k+1)}, \qquad p^{(k+1)} = p_{ex}^{(k+1)} + \boldsymbol{e}_P^{(k+1)}, \tag{3.2}$$

with $\|\boldsymbol{e}_X^{(k+1)}\|, \|\boldsymbol{e}_P^{(k+1)}\| \le \epsilon^{(k+1)}$. The error vector corresponds to the matrix truncation $X^{(k+1)} = X_{ex}^{(k+1)} + E_X^{(k+1)}$ described in the previous section; analogously for the recurrence in $p^{(k+1)}$. With this notation, $r^{(k+1)} = r_{ex}^{(k+1)} - \mathcal{A}\boldsymbol{e}_X^{(k+1)}$.

The CG exact iterate minimizes the convex function

$$f(x) = \frac{1}{2} x^\top \mathcal{A} x - \mathcal{C}^\top x \tag{3.3}$$

in the generated Krylov subspace. At each iteration k the coefficient α_k is determined so as to minimize the function f along the direction determined by $p^{(k)}$. The corresponding value is obtained by imposing that $\frac{df}{d\alpha}(x^{(k)} + \alpha p^{(k)}) = 0$. This minimization property is destroyed by the truncation strategy. The iterate $x^{(k+1)} = x_{ex}^{(k+1)} + \boldsymbol{e}_X^{(k)}$ is such that

$$\frac{df}{d\alpha}(x^{(k+1)}) = \left(x_{ex}^{(k+1)} + \boldsymbol{e}_X^{(k+1)}\right)^\top \mathcal{A} p^{(k)} - \mathcal{C}^\top p^{(k)} = \left(\boldsymbol{e}_X^{(k+1)}\right)^\top \mathcal{A} p^{(k)},$$
$$\left|\frac{df}{d\alpha}(x^{(k+1)})\right| \le \epsilon^{(k+1)} \|\mathcal{A}\| \, \|p^{(k)}\|.$$

The relation shows that at least locally, loss of minimization is controlled by the truncation tolerance. Analogously, without truncation the computation of β_k ensures that $(p_{ex}^{(k+1)})^\top \mathcal{A} p^{(k)} = 0$, whereas for the truncated vector $p^{(k+1)}$ it holds that

$$\left(p^{(k+1)}\right)^\top \mathcal{A} p^{(k)} = \left(\boldsymbol{e}_P^{(k+1)}\right)^\top \mathcal{A} p^{(k)}, \quad |\left(p^{(k+1)}\right)^\top \mathcal{A} p^{(k)}| \le \epsilon^{(k+1)} \|\mathcal{A}\| \, \|p^{(k)}\|.$$

Hence, at first sight, local $\mathcal{A}$-orthogonality seems to be controlled by the truncation tolerance. However, what matters when discussing loss of orthogonality is the angle between the two vectors, not just the magnitude of the inner product. Indeed, this and related quantities are more severely affected by truncation, and the angles between these vectors (directions and residuals) play a crucial role. For the iterates with no truncation at iteration k (that is, $\boldsymbol{e}_X^{(k+1)} = 0 = \boldsymbol{e}_P^{(k+1)}$ in (3.2)), the following properties hold:

(i) $(p^{(k)})^\top r_{ex}^{(k+1)} = 0$, enforced by the choice of α_k;
(ii) $(p_{ex}^{(k+1)})^\top \nabla f(x_{ex}^{(k+1)}) (= -(p_{ex}^{(k+1)})^\top r_{ex}^{(k+1)}) = -\|r_{ex}^{(k+1)}\|^2 < 0$, that is, the new direction is a descent direction;

(iii) assuming no truncation for all k's, $\beta_k = \frac{(r_{ex}^{(k+1)})^\top r_{ex}^{(k+1)}}{(r_{ex}^{(k)})^\top r_{ex}^{(k)}}$ and $\beta_k > 0$, that is, the next $p_{ex}^{(k+1)}$ moves along the (positive) direction of $p_{ex}^{(k)}$.

Property (i) ensures that the space keeps growing. None of these relations continues to hold after truncation of $x_{ex}^{(k+1)}$ and $p_{ex}^{(k+1)}$. In particular, the positivity of β_k is crucial for convergence. We stress that this setting is different from what is observed in finite precision analysis, where local orthogonality can be preserved.

We start by making more explicit the influence of the truncation in the vector recurrences.

LEMMA 3.1. *After k iterations of the truncated CG, it holds that*

(i) $(p^{(k)})^\top r^{(k+1)} = -(p^{(k)})^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)}$;

(ii) $(p^{(k+1)})^\top r^{(k+1)} = \|r^{(k+1)}\|^2 - \beta_k (p^{(k)})^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)} + (\boldsymbol{e}_P^{(k+1)})^\top r^{(k+1)}$;

(iii) $$\left(r^{(k+1)}\right)^\top r^{(k)} = \left(-p^{(k)} + \beta_{k-1}p^{(k-1)}\right)^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)} + \beta_{k-1}\left(p^{(k-1)}\right)^\top \mathcal{A}\boldsymbol{e}_X^{(k)} + \left(\boldsymbol{e}_P^{(k)}\right)^\top (\beta_{k-1}\alpha_k \mathcal{A}p^{(k-1)} - r^{(k+1)}).$$

Proof. To obtain (i), we recall that the definition of α_k ensures that $(p^{(k)})^\top r_{ex}^{(k+1)} = (p^{(k)})^\top (r^{(k)} - \alpha_k \mathcal{A}p^{(k)}) = 0$. On the other hand, the residual $r^{(k+1)} = r_{ex}^{(k+1)} - \mathcal{A}\boldsymbol{e}_X^{(k+1)}$ satisfies

$$(p^{(k)})^\top r^{(k+1)} = 0 - (p^{(k)})^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)}. \tag{3.4}$$

To prove (ii), we write

$$\begin{aligned}\left(p^{(k+1)}\right)^\top r^{(k+1)} &= \left(r^{(k+1)} + \beta_k p^{(k)} + \boldsymbol{e}_P^{(k+1)}\right)^\top r^{(k+1)} \\ &= (r^{(k+1)})^\top r^{(k+1)} - \beta_k (p^{(k)})^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)} + \left(\boldsymbol{e}_P^{(k+1)}\right)^\top r^{(k+1)},\end{aligned}$$

where we used the relation $(p^{(k)})^\top r_{ex}^{(k+1)} = 0$.

The proof of (iii) is a little more elaborate. Indeed,

$$\begin{aligned}\left(r^{(k+1)}\right)^\top r^{(k)} &= \left(r^{(k+1)}\right)^\top \left(p^{(k)} - \beta_{k-1}p^{(k-1)} - \boldsymbol{e}_P^{(k)}\right) \\ &= \left(r^{(k+1)}\right)^\top p^{(k)} - \beta_{k-1}\left(r^{(k+1)}\right)^\top p^{(k-1)} - \left(r^{(k+1)}\right)^\top \boldsymbol{e}_P^{(k)} \\ &= -\left(p^{(k)}\right)^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)} - \beta_{k-1}\left(r_{ex}^{(k+1)} - \mathcal{A}\boldsymbol{e}_X^{(k+1)}\right)^\top p^{(k-1)} - \left(r^{(k+1)}\right)^\top \boldsymbol{e}_P^{(k)} \\ &\overset{(3.4)}{=} -\left(p^{(k)}\right)^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)} - \beta_{k-1}\left(r^{(k)} - \alpha_k \mathcal{A}p^{(k)}\right)^\top p^{(k-1)} \\ &\quad + \beta_{k-1}\left(\mathcal{A}\boldsymbol{e}_X^{(k+1)}\right)^\top p^{(k-1)} - \left(r^{(k+1)}\right)^\top \boldsymbol{e}_P^{(k)}\end{aligned}$$

Now,

$$\left(r^{(k)}\right)^\top p^{(k-1)} = -\left(p^{(k-1)}\right)^\top \mathcal{A}\boldsymbol{e}_X^{(k)}, \qquad \left(\mathcal{A}p^{(k)}\right)^\top p^{(k-1)} = \left(\mathcal{A}\boldsymbol{e}_P^{(k)}\right)^\top p^{(k-1)},$$

so that

$$\begin{aligned}\left(r^{(k+1)}\right)^\top r^{(k)} = &-\left(p^{(k)}\right)^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)} + \beta_{k-1}\left(p^{(k-1)}\right)^\top \mathcal{A}\boldsymbol{e}_X^{(k)} \\ &+ \beta_{k-1}\alpha_k\left(\mathcal{A}\boldsymbol{e}_P^{(k)}\right)^\top p^{(k-1)} + \beta_{k-1}\left(\mathcal{A}\boldsymbol{e}_X^{(k+1)}\right)^\top p^{(k-1)} - \left(r^{(k+1)}\right)^\top \boldsymbol{e}_P^{(k)}.\end{aligned}$$

∎

THEOREM 3.2. *Let* $\Delta_k = \max\{\|\boldsymbol{e}_P^{(k)}\|, \|\boldsymbol{e}_X^{(k)}\|, \|\boldsymbol{e}_P^{(k+1)}\|, \|\boldsymbol{e}_X^{(k+1)}\|\}$ *and also* $\delta_k = \min\{\|\boldsymbol{e}_P^{(k)}\|, \|\boldsymbol{e}_X^{(k)}\|, \|\boldsymbol{e}_P^{(k+1)}\|, \|\boldsymbol{e}_X^{(k+1)}\|\}$, *where we assume* $\delta_k > 0$, *that is, truncation occurs. Then*

$$\eta_k \frac{1}{\|\mathcal{A}^{-1}\|} \frac{\delta_k}{\|r^{(k+1)}\|} \leq \frac{|(r^{(k+1)})^\top p^{(k)}|}{\|r^{(k+1)}\|\|p^{(k)}\|} \leq \|\mathcal{A}\| \frac{\Delta_k}{\|r^{(k+1)}\|},$$

with $\eta_k = \frac{|(\mathcal{A}p^{(k)})^\top \boldsymbol{e}_X^{(k+1)}|}{\|\mathcal{A}p^{(k)}\|\,\|\boldsymbol{e}_X^{(k+1)}\|} \in [0,1]$, *and*

$$\beta_k = -\frac{(r_{ex}^{(k+1)})^\top \mathcal{A}p^{(k)} - (\mathcal{A}\boldsymbol{e}_X^{(k+1)})^\top \mathcal{A}p^{(k)}}{(p^{(k)})^\top \mathcal{A}p^{(k)}}.$$

Moreover, for $\gamma = (\|\mathcal{A}p^{(k)}\| + (2|\beta_{k-1}| + |\beta_{k-1}\alpha_k|)\|\mathcal{A}p^{(k-1)}\| + \|r^{(k+1)}\|)/\|r^{(k)}\|$, *it holds that*

$$\frac{|(r^{(k+1)})^\top r^{(k)}|}{\|r^{(k+1)}\|\|r^{(k)}\|} \leq \gamma \frac{\Delta_k}{\|r^{(k+1)}\|}.$$

Proof. The first upper bound is a direct consequence of Lemma 3.1(i). For the lower bound,

$$\begin{aligned}
\frac{|(p^{(k)})^\top r^{(k+1)}|}{\|p^{(k)}\|\,\|r^{(k+1)}\|} &= \frac{|(p^{(k)})^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)}|}{\|p^{(k)}\|\,\|r^{(k+1)}\|} \\
&\geq \frac{|(p^{(k)})^\top \mathcal{A}\boldsymbol{e}_X^{(k+1)}|}{\|\mathcal{A}^{-1}\|\,\|\mathcal{A}p^{(k)}\|\,\|r^{(k+1)}\|} \\
&= \frac{|(\mathcal{A}p^{(k)})^\top \boldsymbol{e}_X^{(k+1)}|}{\|\mathcal{A}p^{(k)}\|\,\|\boldsymbol{e}_X^{(k+1)}\|} \frac{\|\boldsymbol{e}_X^{(k+1)}\|}{\|\mathcal{A}^{-1}\|\,\|r^{(k+1)}\|} \geq \eta_k \frac{1}{\|\mathcal{A}^{-1}\|} \frac{\delta_k}{\|r^{(k+1)}\|}.
\end{aligned}$$

Recalling the definition of β_k, that is, $\beta_k = -(r^{(k+1)})^\top \mathcal{A}p^{(k)}/((p^{(k)})^\top \mathcal{A}p^{(k)})$, the equality for β_k simply follows from substituting $r^{(k+1)} = r_{ex}^{(k+1)} - A\boldsymbol{e}_X^{(k+1)}$ into the numerator.

Finally, using Lemma 3.1(iii) we obtain

$$\begin{aligned}
&|(r^{(k+1)})^\top r^{(k)}| \\
&\quad \leq \|(\mathcal{A}p^{(k)})\|\,\|\boldsymbol{e}_X^{(k+1)}\| + |\beta_{k-1}|\,\|\mathcal{A}p^{(k-1)}\|\|\boldsymbol{e}_X^{(k)}\| + |\beta_{k-1}\alpha_k|\|\mathcal{A}p^{(k-1)}\|\,\|\boldsymbol{e}_P^{(k)}\| \\
&\qquad + |\beta_{k-1}|\,\|\boldsymbol{e}_X^{(k+1)}\|\,\|\mathcal{A}p^{(k-1)}\| + \|r^{(k+1)}\|\,\|\boldsymbol{e}_P^{(k)}\| \\
&\quad \leq \Big(\|(\mathcal{A}p^{(k)})\| + |\beta_{k-1}|\|\mathcal{A}p^{(k-1)}\| + |\beta_{k-1}\alpha_k|\|\mathcal{A}p^{(k-1)}\| \\
&\qquad + |\beta_{k-1}|\|\mathcal{A}p^{(k-1)}\| + \|r^{(k+1)}\| \Big) \Delta_k,
\end{aligned}$$

from which the value of γ and the final bound follow. ∎

The theorem above shows that the cosine of the angle between the direction vector and the next residual grows in a way that is inversely proportional to the current residual norm. The same property holds for the cosine of the angle between two consecutive residuals, with the caveat for γ not to be much greater than $\mathcal{O}(1)$. We emphasize that with no truncation, both inner products should be zero, as the vectors in both pairs are orthogonal to each other. Whenever the cosine of the angle reaches a value close to one, the next residual vector is almost parallel to the previous direction

vector and is thus unlikely to contribute to the expansion of the approximation space. As a result, stagnation of the whole process occurs (see Example 3.3). In addition, and for any number of iterations, for large enough truncation tolerance, the residual and direction vectors build significantly different subspaces.

The expression for β_k shows that for a small residual $r_{ex}^{(k+1)}$, the second term at the numerator may become significant, and if the sign of the two terms is the same, the coefficient β_k may become negative. As soon as β_k becomes negative, the whole CG framework breaks down, with the next direction vector continuing in the previous direction but backwards. Apparently, the procedure is unable to recover (see the example below), leading to overall residual norm stagnation. In fact, from then on the sign of β_k starts to alternate, according to small corresponding modifications in the value of the residual norm; see Example 3.3.

Example 3.3. Let $A \in \mathbb{R}^{n\times n}$, $A = T \otimes I + I \otimes T$, $T = \text{tridiag}(-1, \underline{2}, -1)$, and $M = \text{pentadiag}(-0.5, -1, \underline{3.2}, -1, -0.5) \in \mathbb{R}^{n\times n}$ with $n = 100$ and $C = c_1 c_1^\top$ with c_1 having random entries (from the MATLAB uniform distribution rand [23]). Figure 2 reports the convergence history of the TCG residual norm (dashed thick line) for different values of the truncation threshold (solid thin straight line). Also included are the angle cosines $\frac{|(r^{(k+1)})^\top p^{(k)}|}{\|r^{(k+1)}\|\|p^{(k)}\|}$ (solid thick line) and $\frac{|(r^{(k+1)})^\top r^{(k)}|}{\|r^{(k+1)}\|\|r^{(k)}\|}$ (dashed thinner line). All plots show the inverse correspondence between the residual norm and the two angle cosines. The agreement of the iteration at which the two final plateau values are reached is striking. Figure 3 reports the value of β_k during the whole convergence history, for the same truncation tolerances as in Figure 2. We highlight the oscillation of β_k around zero in correspondence with stagnation, as opposed to the strictly positive values obtained in the untruncated case. For completeness, in Figure 4

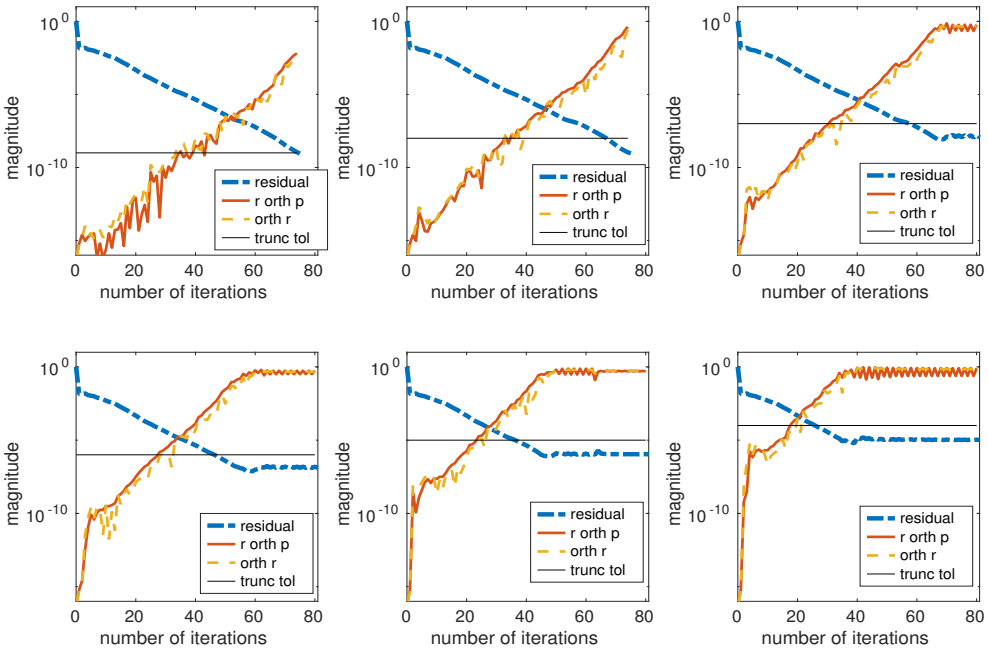

FIG. 2. *Example* 3.3. *Convergence history of truncated CG residual norm (dashed thick black line) for different values of truncation parameter (thin solid line). Loss of orthogonality (cosine of the angles) between consecutive residuals and residual and directions is also reported.*

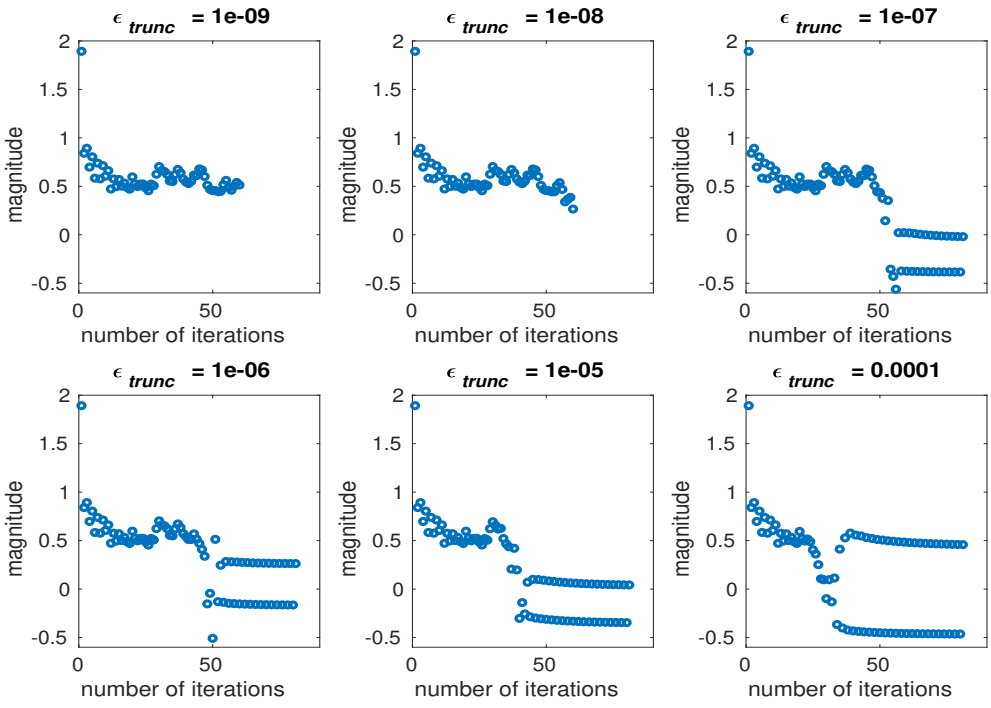

FIG. 3. *Example* 3.3. *Values of the computed* β_k *as the iterations proceed, for different values of truncation tolerance in the runs of Figure* 2.

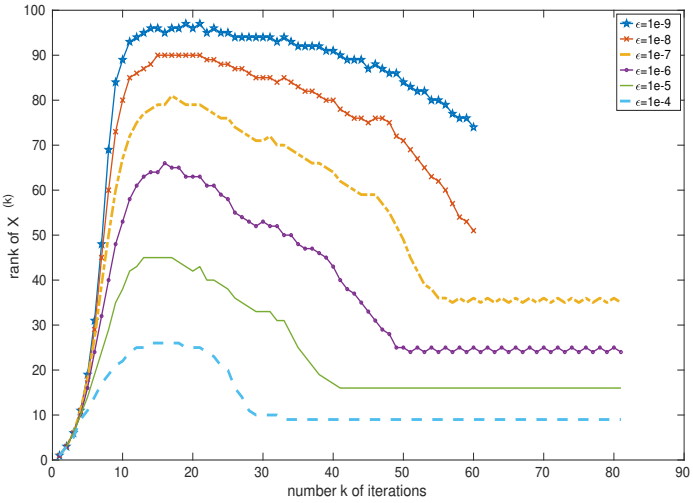

FIG. 4. *Example* 3.3. *Rank of approximate solution* $X^{(k)} = X_1^{(k)}(X_2^{(k)})^\top$ *as the iterations proceed, for different values of the truncation parameter, in the runs of Figure* 2.

we also report the approximate solution rank after truncation as the iterations proceed. The rank growth is consistent with the discussion of section 2.1, as long as the truncation threshold does not affect the action of the error norm decay. Finally, for a selection of truncation tolerance values, Figure 5 shows the singular values of the final basis generated by the residual vectors $[r^{(0)}, \ldots, r^{(k)}]$ (normalized to have unit

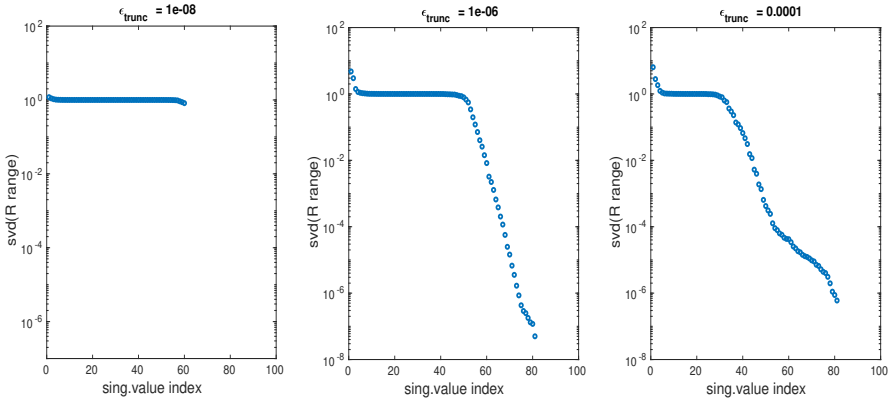

FIG. 5. *Example* 3.3. *Singular values of the matrix* $[r^{(0)}/\|r^{(0)}\|, r^{(1)}/\|r^{(1)}\|, \ldots, r^{(k)}/\|r^{(k)}\|]$, *for a selection of truncation tolerance values in the runs of Figure* 2.

Euclidean norm). For the largest truncation tolerance, these plots illustrate that as the cosine of the angle gets close to one (cf. Figure 2), the residual vectors become linearly dependent, and not only nonorthogonal, leading to a large loss of rank in the approximation space generated by the residual vectors.

The previous experiments illustrate that as long as the residual vectors remain independent, the space keeps growing, and convergence may continue to improve in a way similar to what occurs without truncation. More precisely, preserved local orthogonality seems to be sufficient for the method to advance the approximation with *linear* convergence; we expect superlinear convergence to be lost, as it occurs in finite precision CG; see, e.g., [25] and references therein. The importance of local orthogonality has been stressed in the past to enhance convergence properties of *inexact* preconditioned CG; see, e.g., [8], [27], [9], and their references. Similar pictures can also be observed in analyzing round-off effects in the GMRES orthonormal basis, constructed with the modified Gram–Schmidt algorithm, in which the residual norm stagnates at the level where all linear independence is lost [11].

The following example partially taken from [25] shows that the eigenvalue distribution can in fact play a role in the convergence when truncation is applied. Although of a theoretical nature, this example shows that the convergence of the truncated version of CG may significantly differ from the expected convergence of the untruncated method, and that the convergence curve may diverge from the untruncated one much before the level of the truncation threshold.

Example 3.4. For $n = 100$, we consider $A = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ with

$$\lambda_i = \lambda_1 + \frac{(i-1)}{(n-1)}(\lambda_n - \lambda_1)\rho^{n-i}, \quad \lambda_1 = 0.1, \quad \lambda_n = 100, \quad \rho \in \{0.4, 0.8\}.$$

The matrix A is well known to be a challenging case for CG and it has been extensively used in the literature to analyze the convergence behavior of CG; see [35], and also [21] for a more recent account. Matrix M is taken to be the diagonal matrix with elements logarithmically distributed in the interval $[10^{-2}, 10^{0}]$, and c_1 with all equal components. This choice of M allows us to maintain the specific properties of the matrix A, which classically provide insights into the behavior of CG in finite precision arithmetic observed in the literature for the given distribution of eigenvalues of A

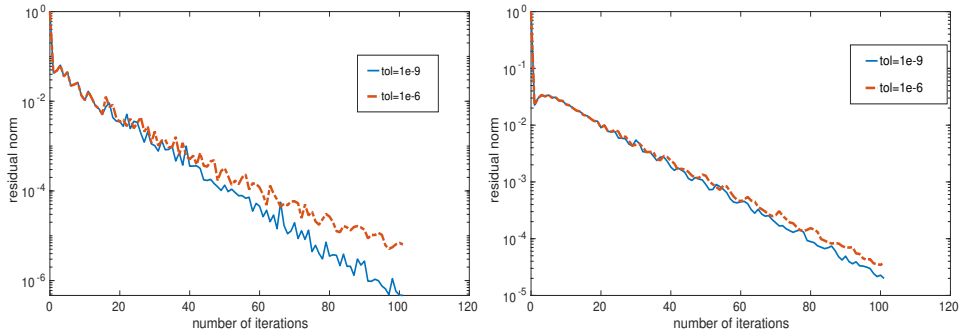

FIG. 6. *Example* 3.4. *Convergence history of TCG for* $\rho = 0.4$ *(left) and* $\rho = 0.8$ *(right) for different truncation tolerances* tol.

[35]. The value of ρ influences the eigenvalue distribution—but not the conditioning of A—with more evenly distributed eigenvalues for larger values of ρ. The plots in Figure 6 show the convergence history of TCG for two truncation tolerances, with $\rho = 0.4$ (left) and $\rho = 0.8$ (right). We can appreciate that a more severe truncation influences the convergence curve well above the truncation tolerance level. Though the observed delay is not dramatic, this experiment illustrates that different truncation tolerances not only influence the final stagnation level but may also influence the whole convergence history; see [18] for other examples where convergence is affected before stagnation level.

An intuitive explanation of the modification in the convergence history is that the approximation space changes as soon as truncation takes place. Our analysis does not provide a rigorous description of this convergence delay, which can perhaps be analyzed by borrowing tools from finite precision arithmetic analysis (see, e.g., [10], [21]); though the type of inexactness is structurally different, the inaccuracy threshold is at a much larger order of magnitude, and the expectation in the achieved accuracy is rather dissimilar. This fascinating connection deserves further study.

4. The Galerkin method. In this section we make a tight connection between the matrix-oriented CG method (with no truncation) and projection methods on (1.2) by showing that they work with the same approximation spaces. Following well-established procedures for the Lyapunov and Sylvester equations [32], it is possible to directly attack (1.2), thus bypassing the Kronecker formulation. An approximate solution in the form $X_k = V_k Y_k V_k^\top$ is sought, where the orthonormal columns of V_k span a specifically selected approximation space of low dimension, and Y_k is a small size matrix, determined by imposing some conditions on the approximations. The space is expanded if the approximation is not good enough; the parameter k accounts for the number of iterations. Let $R_k = C - (AX_k + X_kA + MX_kM)$ be the residual matrix. To determine Y_k a Galerkin condition is imposed, requiring that the residual matrix be orthogonal to the generated space in a matrix sense, that is,

$$V_k^T R_k V_k = 0.$$

Substituting the symmetric form $X_k = V_k Y V_k^T$ for some Y, we obtain

$$(V_k^T AV_k)Y + Y(V_k^T AV_k) + (V_k^T MV_k)Y(V_k^T MV_k) = V_k^T CV_k. \tag{4.1}$$

Therefore, Y_k can be obtained as the solution to the (reduced) matrix equation above, which has the same structure as the original problem, but much smaller dimensions. We refer the reader to, e.g., [13], [12, sec.7] for successful applications of this procedure.

The fundamental step for the effectiveness of this Galerkin procedure is the choice of the approximation space. For $M = 0$, a particularly effective choice is given by the *rational Krylov subspace*, defined as $\text{span}\{c_1, (A+\sigma_1 I)^{-1}c_1, \ldots, \prod_{j=1}^{k}(A+\sigma_j I)^{-1}c_1\}$, where the parameters σ_j, $j = 1, \ldots, k$, can be selected beforehand or adaptively during the space generation; see, e.g., [32] and references therein. As for TCG, an important hypothesis is that the solution X can be well approximated by a low-rank matrix; if this is not the case, an excessively large approximation space and a large reduced problem (4.1) may arise.

Consider the space $\mathcal{K}_k = \text{range}(V_k)$ generated as $V_0 = c_1$, $V_k = [V_{k-1}, Av_k, Mv_k]$, $k = 1, 2, \ldots$, where v_k is the kth vector of V_{k-1}, that is,[6] $V_0 = c_1 =: \underline{v_1}$,

$$\begin{aligned} V_1 &= [V_0, Av_1, Mv_1] =: [v_1, \underline{v_2}, v_3], \\ V_2 &= [V_1, Av_2, Mv_2] =: [v_1, v_2, \underline{v_3}, v_4, v_5], \\ V_3 &= [V_2, Av_3, Mv_3] =: [v_1, v_2, v_3, \underline{v_4}, v_5, v_6, v_7], \quad \text{etc.} \end{aligned}$$

Assuming full rank of the computed matrix, the generated subspace has dimension $\dim(\text{range}(V_k)) = 2k + 1$, $k = 0, 1, \ldots$. Without orthogonalization, the matrix generated by this procedure grows as

$$V_\infty = [c_1, Ac_1, Mc_1, A^2c_1, MAc_1, AMc_1, M^2c_1, A^3c_1, MA^2c_1, AMAc_1, M^2Ac_1, \ldots].$$

Recalling the definition of $\mathbb{Q}_k$ in section 2.2, the following relation holds:

$$\mathbb{Q}_k = \text{range}(V_{2^k-1}). \tag{4.2}$$

As a consequence, as the iterations proceed the dimension of the approximation space grows significantly less in the Galerkin case. However, note that in CG, the subspace actually employed is a constrained subset of $\mathbb{Q}_k$ (see section 2.2).

After k iterations of the Galerkin method V_k appears to contain all terms of $(A + M)^j c_1$, with $j \le k$. However, the space is richer than the space generated by powers of $A + M$. In other words, there exist vectors $p \in \text{range}(V_k)$ that cannot be written as $p = \sum_{j=0}^{k} \alpha_j (A+M)^j c_1$. For instance, $p = \gamma_0 c_1 + \gamma_1 Ac_1 + \gamma_2 Mc_1$ with $\gamma_1 \neq \gamma_2$ cannot be written using only powers of $(A + M)$. The following proposition gives an explicit representation of vectors in range(V_k). Its proof follows from observing that for any $i \le k$, the vector $0 \neq w = \varphi_i(A, M)c_1$ is a linear combination of elements in range(V_k) of exact degree i in at least A or M.

PROPOSITION 4.1. *Let $p \in \text{range}(V_k)$. Then there exist γ_i, $\alpha_{i,\ell}$, and $\beta_{i,\ell}$ such that $p = \sum_{i=0}^{k} \gamma_i \prod_{\ell=0}^{i}(\alpha_{i,\ell}A + \beta_{i,\ell}M)c_1$. In a more compact way, letting $\varphi_i(\xi, \eta) = \prod_{\ell=0}^{i}(\alpha_{i,\ell}\xi + \beta_{i,\ell}\eta)$ be the bivariate polynomial of degree not greater than i, it holds that $p = \sum_{i=0}^{k} \gamma_i \varphi_i(A, M)c_1$.*

We stress that the coefficients γs, αs and βs are not necessarily independent.

The discussion associated with item (i) of section 2.1 carries over to the low-rank matrix $X_k = V_k Y_k V_k^T$ obtained by the Galerkin method. The rank of X_k is at most equal to the dimension of Y_k, and it can thus be monitored. Since the energy norm of the error $X^\star - X_k$ is minimized (see [29]), convergence is focused on

[6] The vector used to expand the space in the next iteration is underlined.

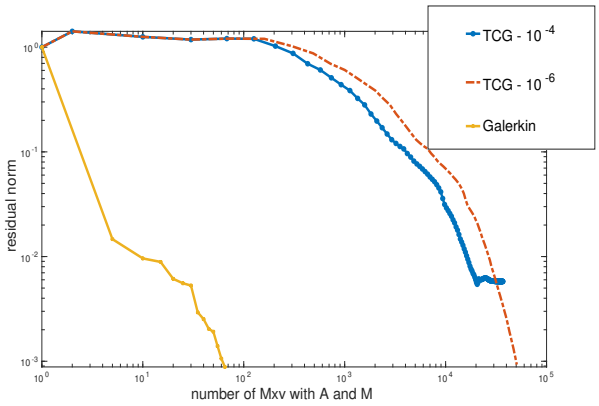

FIG. 7. *Example* 4.2. *Number of matrix-vector multiplies with A and M as iteratios proceed for the Galerkin method and for TCG, using different truncation tolerances.*

matching the leading singular values, while the rank remains under control. Another immediate choice of space is $V_k = [V_{k-1}, M^{-1}Av_k, M^{-1}v_k]$, which corresponds to a premultiplication by M^{-1} of both sides of problem (1.2); see also (1.8). This is the space constructed in the next example.

Example 4.2. The matrix A stems from the centered finite difference discretization of the operator $(-\exp(-xy)u_x)_x - (\exp(xy)u_y)_y$ in the unit square with homogeneous Dirichlet boundary conditions. The discretization leads to a matrix A of dimension $n = 400$. Here $M = \text{tridiag}(-1, \underline{2.1}, -1)$ and c_1 has random entries uniformly distributed in $(0, 1)$. The equation is multiplied from the left and from the right by M^{-1}. In CG this corresponds to using $M \otimes M$ as preconditioner.

The Galerkin method requires a subspace of dimension 131 to reach a relative residual norm less than 10^{-3}. TCG with a truncation tolerance 10^{-6} requires 80 iterations to reach the same residual tolerance, with the final X_1, P_1 having, respectively, 92 and 219 columns, though during the iteration matrices $X_1^{(k)}$, $P_1^{(k)}$ with up to n columns were generated. TCG with a truncation tolerance 10^{-4} did not reach the requested accuracy, stagnating much earlier. Figure 7 reports the number of matrix-vector multiplies with A and M as iterations proceed, for all methods, illustrating the superiority of the Galerkin procedure also with this cost measure.

In the previous experiment we have used an approximation space that is related to that generated by CG to illustrate the previous arguments. Results are not always in favor of the Galerkin method when this space is employed. Other more effective approximation spaces for the given problem can be considered; see, e.g., [3], [13].

5. Conclusions. The matrix version of CG provides a new theoretical and computational framework for the iterative solution of linear matrix equations via Krylov subspaces. We have described some of the relations that influence the actual behavior of the method. In particular, we have characterized how convergence takes place, in terms of the error matrix, and in which way loss of orthogonality plays a role when rank truncation is in action. In addition, a tight connection to Galerkin methods applied to the original problem has been devised. Our analysis can provide new insights for possible future improvements over the basic implementation of TCG, such as local orthogonality imposition and truncation criterion selection.

Acknowledgment. We thank two anonymous reviewers for their careful reading.

REFERENCES

[1] P. Benner and T. Breiten, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numer. Math., 124 (2013), pp. 441–470.
[2] W. G. Bickley and J. McNamee, *Matrix and other direct methods for the solution of linear difference equation*, Philos. Trans. Roy. Soc. London Ser. A, 252 (1960), pp. 69–131.
[3] A. Bünger, V. Simoncini, and M. Stoll, *A low-rank matrix equation method for solving PDE-constrained optimization problems*, SIAM J. Sci. Comput., 43 (2021), pp. S637–S654, https://doi.org/10.1137/20M1341210.
[4] T. Damm, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numer. Linear Algebra Appl., 15 (2008), pp. 853–871.
[5] E. De Sturler, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889, https://doi.org/10.1137/S0036142997315950.
[6] N. S.Ellner and E. L.Wachspress, *New ADI model problem applications*, in Proceedings of the 1986 ACM Fall Joint Computer Conference, ACM, New York, 1986, pp. 528–534.
[7] G. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., The Johns Hopkins University Press, Baltimore, MD, 2013.
[8] G. H. Golub and Q. Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput., 21 (2001), pp. 1305–1320, https://doi.org/10.1137/S1064827597323415.
[9] S. Gratton, E. Simon, D. Titley-Peloquin, and P. L. Toint, *Minimizing convex quadratics with variable precision conjugate gradients*, Numer. Linear Algebra Appl., 28 (2021), e2337.
[10] A. Greenbaum, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.
[11] A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical behavior of the modified Gram-Schmidt GMRES implementation*, BIT, 37 (1997), pp. 706–719.
[12] Y. Hao and V. Simoncini, *The Sherman-Morrison-Woodbury formula for generalized linear matrix equations and applications*, Numer. Linear Algebra Appl., 28 (2021), e2384.
[13] J. Henning, D. Palitta, V. Simoncini, and K. Urban, *Matrix oriented reduction of space-time Petrov-Galerkin variational problems*, in Numerical Mathematics and Advanced Applications, ENUMATH 2019, Springer, New York, 2021, pp. 1049–1058.
[14] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[15] D. Kressner, M. Plešinger, and C. Tobler, *A preconditioned low-rank CG method for parameter-dependent Lyapunov equations*, Numer. Linear Algebra Appl., 21 (2014), pp. 666–684.
[16] D. Kressner and P. Sirković, *Truncated low-rank methods for solving general linear matrix equations*, Numer. Linear Algebra Appl., 22 (2015), pp. 564–583, https://doi.org/10.1002/nla.1973.
[17] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1688–1714, https://doi.org/10.1137/090756843.
[18] D. Kressner and C. Tobler, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1288–1316, https://doi.org/10.1137/100799010.
[19] D. Kressner and A. Uschmajew, *On low-rank approximability of solutions to high-dimensional operator equations and eigenvalue problems*, Linear Algebra Appl., 493 (2016), pp. 556–572.
[20] P. Lancaster, *Explicit solutions of linear matrix equations*, SIAM Rev., 12 (1970), pp. 544–566, https://doi.org/10.1137/1012104.
[21] J. Liesen and Z. Strakos, *Krylov Subspace Methods. Principles and Analysis*, Oxford University Press, Oxford, UK, 2013.
[22] T. Masák, *Covariance Estimation for Random Surfaces beyond Separability*, Ph.D. thesis, Programme doctoral en mathématiques, École polytechnique fédèrale de Lausanne, Lausanne, Switzerland, 2022, N. 9463.
[23] MathWorks, Inc., *MATLAB* 7, r2020b ed., MathWorks, Natick, MA, 2020.
[24] H. Matthies and E. Zander, *Solving stochastic systems with low-rank tensor compression*, Linear Algebra Appl., 436 (2012), pp. 3819–3838.

[25] G. Meurant and Z. Strakos, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.
[26] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171, https://doi.org/10.1137/S0895479893253975.
[27] Y. Notay, *Flexible conjugate gradients*, SIAM J. Sci. Comput., 22 (2000), pp. 1444–1460, https://doi.org/10.1137/S1064827599362314.
[28] D. Palitta and P. Kürschner, *On the convergence of Krylov methods with low-rank truncations*, Numer. Algorithms, 88 (2021), pp. 1383–1417, https://doi.org/10.1007/s11075-021-01080-2.
[29] D. Palitta and V. Simoncini, *Optimality properties of Galerkin and Petrov-Galerkin methods for linear matrix equations*, Vietnam J. Math., 48 (2020), pp. 791–807, https://doi.org/10.1007/s10013-020-00390-7.
[30] T. Penzl, *Eigenvalue decay bounds for solutions of Lyapunov equations: The symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.
[31] S. D. Shank, V. Simoncini, and D. B. Szyld, *Efficient low-rank solutions of generalized Lyapunov equations*, Numer. Math., 134 (2016), pp. 327–342.
[32] V. Simoncini, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441, https://doi.org/10.1137/130912839.
[33] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[34] M. Stoll and T. Breiten, *A low-rank in time approach to PDE-constrained optimization*, SIAM J. Sci. Comput., 37 (2015), pp. B1–B29, https://doi.org/10.1137/130926365.
[35] Z. Strakoš, *On the real convergence rate of the conjugate gradient method*, Linear Algebra Appl., 154/156 (1991), pp. 535–549.
[36] B. Vandereycken and S. Vandewalle, *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2553–2579, https://doi.org/10.1137/090764566.
[37] E. L.Wachspress, *Extended application of alternating direction implicit iteration model problem theory*, J. Soc. Indust. Appl. Math., 11 (1963), pp. 994–1016, https://doi.org/10.1137/0111073.
[38] J. Zhang and J. G. Nagy, *An alternating direction method of multipliers for the solution of matrix equations arising in inverse problems*, Numer. Linear Algebra Appl., 25 (2018), e2123.