



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE  
DELLA RICERCA

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

SKETCHED AND TRUNCATED POLYNOMIAL KRYLOV SUBSPACE METHODS: MATRIX SYLVESTER EQUATIONS

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Palitta, D., Schweitzer, M., Simoncini, V. (2024). SKETCHED AND TRUNCATED POLYNOMIAL KRYLOV SUBSPACE METHODS: MATRIX SYLVESTER EQUATIONS. MATHEMATICS OF COMPUTATION, ON LINE FIRST, 1-32 [10.1090/mcom/4002].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/997350> since: 2024-11-26

*Published:*

DOI: <http://doi.org/10.1090/mcom/4002>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

# SKETCHED AND TRUNCATED POLYNOMIAL KRYLOV SUBSPACE METHODS: MATRIX SYLVESTER EQUATIONS

DAVIDE PALITTA, MARCEL SCHWEITZER, AND VALERIA SIMONCINI

ABSTRACT. Thanks to its great potential in reducing both computational cost and memory requirements, combining sketching and Krylov subspace techniques has attracted a lot of attention in the recent literature on projection methods for linear systems, matrix function approximations, and eigenvalue problems. Applying this appealing strategy in the context of linear matrix equations turns out to be far more involved than a straightforward generalization. These difficulties include analyzing well-posedness of the projected problem and deriving possible error estimates depending on the sketching properties. Further computational complications include the lack of a natural residual norm estimate and of an explicit basis for the generated subspace.

In this paper we propose a new sketched-and-truncated polynomial Krylov subspace method for Sylvester equations that aims to address all these issues. The potential of our novel approach, in terms of both computational time and storage demand, is illustrated with numerical experiments. Comparisons with a state-of-the-art projection scheme based on rational Krylov subspaces are also included.

## 1. INTRODUCTION

In the recent literature sketching has been successfully combined with Krylov subspace methods for the solution of linear systems of equations and eigenvalue problems [4, 5, 33, 47] as well as the numerical approximation of the action of a matrix function on a vector [15, 24, 36]. In this paper we address another central problem in numerical linear algebra, namely the solution of large-scale Sylvester matrix equations of the form

$$(1.1) \quad \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{C}_1\mathbf{C}_2^\top,$$

where  $\mathbf{A} \in \mathbb{R}^{n_1 \times n_1}$ ,  $\mathbf{B} \in \mathbb{R}^{n_2 \times n_2}$ , and the right-hand side is supposed to be of low rank, i.e.,  $\mathbf{C}_1 \in \mathbb{R}^{n_1 \times r}$ ,  $\mathbf{C}_2 \in \mathbb{R}^{n_2 \times r}$  with  $r \ll \max\{n_1, n_2\}$ . The low rank of the right-hand side, along with suitable additional conditions on the coefficient matrices, like, e.g., a large enough relative distance between the spectra of  $\mathbf{A}$  and  $-\mathbf{B}$ , implies that the desired solution  $\mathbf{X}$  admits accurate low-rank approximations; see, e.g., [3, 21, 39].

For the sake of simplicity, in the following we focus on the case  $n \equiv n_1 = n_2$ , namely the solution  $\mathbf{X}$  is an  $n \times n$  matrix, but everything we present here can be straightforwardly generalized to the case  $n_1 \neq n_2$ .

The Sylvester equation (1.1) can be encountered in a variety of applications. For instance, several model order reduction techniques give rise to Sylvester equations that feature very large and sparse coefficients  $\mathbf{A}$  and  $\mathbf{B}$  [1]. Similarly, the discretization of certain elliptic and parabolic PDEs yields Sylvester equations of

the form (1.1); see, e.g., [35,37]. Another source of applications are linearizations of nonlinear problems, such as the algebraic Riccati equation, where, in general, the solution of (1.1) is needed in iterative solvers [12]. We refer to the surveys [11,44] for further applications and details.

Our goal is to design an efficient projection procedure for the solution of (1.1) based on the polynomial Krylov subspaces  $\mathcal{K}_d(\mathbf{A}, C_1)$  and  $\mathcal{K}_d(\mathbf{B}^\top, C_2)$ , where the Krylov subspace is defined as

$$(1.2) \quad \mathcal{K}_d(\mathbf{A}, C_1) = \text{Range}\{[C_1, \mathbf{A}C_1, \dots, \mathbf{A}^{d-1}C_1]\},$$

and analogously for  $\mathcal{K}_d(\mathbf{B}^\top, C_2)$ .

Early contributions to matrix equation solvers suggested the use of (1.2) as approximation space as well; see the pioneering work in [41] for the Lyapunov equation, and later contributions in, e.g., [29] for the Sylvester equation. However, it is well-known that the rate of convergence of plain polynomial Krylov subspace methods for Sylvester equations is rather poor in general, and a significant number of iterations is often needed to converge, thus worsening the performance of the overall solution procedure. Attempts to lower the computational cost per iteration and the memory requirements in case of symmetric  $\mathbf{A}$  and  $\mathbf{B}$  can be found in [38]. With the same goals, a restarted procedure is presented in [31]. These contributions come with their limitations and restrictions, so that in practice, projection schemes based on the extended Krylov subspace and the rational Krylov subspace are usually preferred, in spite of their more expensive basis construction; see [44].

Projection-based methods are not the only procedures available in the literature for solving large-scale Sylvester equations. A non-complete list of different solvers sees low-rank Alternating Direction Implicit (ADI) methods [10], sign function methods [8], and hierarchical algorithms [22]. We refer to [11,44] for more detailed accounts of different approaches available for small and large scale problems.

In this paper we significantly improve the performance of polynomial Krylov subspace methods for Sylvester equations by combining the projection scheme with randomized sketching [32,50]. In particular, we tackle two of the main computational issues due to the large number of iterations needed by polynomial Krylov subspace methods: (i) the high cost of the orthogonalization step and (ii) the large storage demand due to the full allocation of the basis of (1.2).

As has been shown for linear systems and matrix functions [15,24,33,36], one way to unlock the potential of sketching is to combine it with a truncated Krylov scheme, i.e., a Krylov subspace method where the constructed basis is only locally orthogonal. The latter approach dramatically reduces the cost of the orthogonalization step. On the other hand, working with a non-orthogonal basis often leads to a convergence delay; such phenomenon is well-documented for linear systems [46] and matrix functions [18]. Sketching comes into play to overcome this drawback. In particular, we show that the use of sketching is able to restore the rate of convergence of the full Krylov subspace method without significantly increasing the overall cost of the truncated technique. We would like to mention that, to the best of our knowledge, plain (non-sketched) truncated polynomial Krylov subspace methods for the solution of matrix equations have also never been proposed in the literature. This can thus be seen as a further contribution of this paper.

Thanks to the truncated orthogonalization step, the whole basis is not required during the basis construction, and it is thus not stored, with significant memory

savings. On the other hand, the full basis is needed to retrieve the final solution. We avoid allocating the whole basis by employing a so-called two-pass strategy. This is a common procedure which in our case consists in first running the sketched and truncated method without storing the basis. Then, once the prescribed level of accuracy is met, the solution is constructed by computing its low-rank factors during a second, cheaper truncated Arnoldi step; see, e.g., [30].

The truncated and sketched Krylov machinery has not been used so far in the solution of matrix equations, and as we will illustrate, its formulation and implementation require special attention to obtain an efficient and reliable method.

The remainder of the paper is organized as follows. In [Section 2](#), we review the basics of polynomial Krylov methods and randomized sketching, as well as the combination of both. [Section 3](#) introduces the general framework for sketched and truncated Krylov approximations for Sylvester equations and discusses how they relate to the standard Arnoldi approximation. In [Section 4](#), we investigate the influence of sketching on the field of values and perform a convergence analysis for Lyapunov equations. In [Section 5](#), we discuss implementation aspects and present an algorithm for the sketched Arnoldi method, before presenting numerical experiments in [Section 6](#). Concluding remarks are given in [Section 7](#). [Appendix A](#) contains an algorithmic description of a truncated Arnoldi method (without sketching), and [Appendix B](#) collects auxiliary results on tensorized subspace embeddings.

We assume that exact arithmetic is employed in all the derivations presented in this paper.

Throughout the manuscript, we use the following notation. Vectors are denoted by regular lowercase letters and block vectors (i.e., thin matrices with  $r$  columns) are denoted by regular uppercase letters. “Block scalars” (i.e., matrices of size  $r \times r$  that would be scalars in the rank-1 case) are denoted by bold-face lowercase letters. All other matrices are denoted by bold-face uppercase letters. At some places, calligraphic font is used to help distinguish quantities connected to specific algorithms, which will be explicitly made clear where it is used. By  $E_i \in \mathbb{R}^{dr \times r}$ , we denote the block vector which is all zero except for an identity matrix in its  $i$ th  $r \times r$  block. Given an  $n \times n$  real matrix  $\mathbf{M}$ , the quantity  $\alpha(\mathbf{M}) := \lambda_{\max}((\mathbf{M} + \mathbf{M}^\top)/2)$  denotes the rightmost point of the field of values of  $\mathbf{M}$ , defined as  $W(\mathbf{M}) = \{\frac{x^* \mathbf{M} x}{x^* x}, \text{ with } 0 \neq x \in \mathbb{C}^n\}$ ; here  $\lambda_{\max}(\cdot)$  denotes the rightmost eigenvalue of the argument symmetric matrix. We say that  $\mathbf{M}$  is negative definite if  $\alpha(\mathbf{M}) < 0$ .

## 2. BACKGROUND MATERIAL

In this section we review a few known tools that will be crucial ingredients in what follows.

**2.1. The block Arnoldi relation.** Krylov subspace methods for the solution of Sylvester equations (1.1) build upon the Arnoldi process [2], or a block version of the Arnoldi process ([42, Section 6.12]) if the rank  $r$  of the right-hand side  $C_1 C_2^\top$  is larger than one. We present all methods in the more general block setting in the following, without writing out the obvious simplifications that are possible when  $r = 1$ .

Given the coefficient matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ , the blocks of vectors  $C_1, C_2 \in \mathbb{R}^{n \times r}$ , and number of Krylov iterations<sup>1</sup>  $d$ , the block Arnoldi method computes nested orthonormal bases  $\mathbf{U}_d \in \mathbb{R}^{n \times dr}$  of  $\mathcal{K}_d(\mathbf{A}, C_1)$  and  $\mathbf{V}_d \in \mathbb{R}^{n \times dr}$  of  $\mathcal{K}_d(\mathbf{B}^\top, C_2)$  via modified block Gram–Schmidt orthonormalization. This yields the *block Arnoldi relations*

$$(2.1) \quad \begin{aligned} \mathbf{A}\mathbf{U}_d &= \mathbf{U}_{d+1}\underline{\mathcal{H}}_d = \mathbf{U}_d\mathcal{H}_d + \mathbf{u}_{d+1}\mathbf{h}_{d+1,d}E_d^\top, \\ \mathbf{B}^\top\mathbf{V}_d &= \mathbf{V}_{d+1}\underline{\mathcal{G}}_d = \mathbf{V}_d\mathcal{G}_d + \mathbf{v}_{d+1}\mathbf{g}_{d+1,d}E_d^\top, \end{aligned}$$

where

$$\underline{\mathcal{H}}_d = \begin{bmatrix} \mathbf{h}_{1,1} & \mathbf{h}_{1,2} & \cdots & \mathbf{h}_{1,d} \\ \mathbf{h}_{2,1} & \mathbf{h}_{2,2} & \cdots & \mathbf{h}_{2,d} \\ & \mathbf{h}_{3,2} & \ddots & \vdots \\ & & \ddots & \mathbf{h}_{d,d} \\ & & & \mathbf{h}_{d+1,d} \end{bmatrix}, \quad \underline{\mathcal{G}}_d = \begin{bmatrix} \mathbf{g}_{1,1} & \mathbf{g}_{1,2} & \cdots & \mathbf{g}_{1,d} \\ \mathbf{g}_{2,1} & \mathbf{g}_{2,2} & \cdots & \mathbf{g}_{2,d} \\ & \mathbf{g}_{3,2} & \ddots & \vdots \\ & & \ddots & \mathbf{g}_{d,d} \\ & & & \mathbf{g}_{d+1,d} \end{bmatrix}$$

are block upper Hessenberg matrices of size  $(d+1)r \times dr$ , with blocks  $\mathbf{g}_{i,j}, \mathbf{h}_{i,j} \in \mathbb{R}^{r \times r}$  and  $\mathcal{G}_d, \mathcal{H}_d$  denoting their upper  $dr \times dr$  submatrix. Further,  $\mathbf{U}_{d+1} = [\mathbf{u}_1, \dots, \mathbf{u}_{d+1}]$ ,  $\mathbf{V}_{d+1} = [\mathbf{v}_1, \dots, \mathbf{v}_{d+1}] \in \mathbb{R}^{n \times (d+1)r}$  contain the basis block vectors  $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^{n \times r}$ , respectively.

Performing  $d$  steps of the two block Arnoldi schemes requires  $2dr$  matrix vector products (half of them with  $\mathbf{A}$  and half of them with  $\mathbf{B}^\top$ ) at a cost of  $\mathcal{O}(dr \text{nnz}(\mathbf{A}) + dr \text{nnz}(\mathbf{B}))$  floating point operations (flops), assuming that  $\mathbf{A}$  and  $\mathbf{B}$  are sparse with  $\text{nnz}(\mathbf{A})$  and  $\text{nnz}(\mathbf{B})$  nonzeros, respectively. In addition, the modified block Gram–Schmidt orthogonalization needs to be carried out, which in total, across all  $d$  iterations, induces a cost of  $\mathcal{O}((rd)^2n)$  flops. This quadratic growth in computational complexity means that with growing iteration number, the orthogonalization step tends to dominate the overall cost of the method. This becomes even more severe on modern high performance architectures where each inner product introduces a global synchronization point requiring communication.

**2.2. Oblivious subspace embeddings.** The basis of sketched Krylov methods is the use of so-called *oblivious subspace embeddings* [16, 43, 50]. These allow to embed a (low-dimensional) subspace  $\mathcal{V}$  of  $\mathbb{R}^n$  into  $\mathbb{R}^s$ ,  $s \ll n$ , such that norms (or inner products) are distorted in a controlled manner.

To be specific, for a given  $\varepsilon \in [0, 1)$ , a matrix  $\mathbf{S} \in \mathbb{R}^{s \times n}$  is called an  $\varepsilon$ -*subspace embedding* for  $\mathcal{V}$  (or for brevity, simply a “*sketching matrix*”) if it fulfills,

$$(2.2) \quad (1 - \varepsilon)\|v\|^2 \leq \|\mathbf{S}v\|^2 \leq (1 + \varepsilon)\|v\|^2,$$

for all vectors  $v \in \mathcal{V}$ , where  $\|\cdot\|$  denotes the Euclidean vector norm. By the parallelogram inequality, this is equivalent to requiring

$$(2.3) \quad \langle u, v \rangle - \varepsilon\|u\|\|v\| \leq \langle \mathbf{S}u, \mathbf{S}v \rangle \leq \langle u, v \rangle + \varepsilon\|u\|\|v\|,$$

for all  $u, v \in \mathcal{V}$ .

In the context of sketched Krylov methods, so-called *oblivious* subspace embeddings are very important. These are embeddings which can be constructed without explicit knowledge of the subspace  $\mathcal{V}$  that shall be embedded. To be precise,  $\mathbf{S}$

<sup>1</sup>Note that all our results easily generalize to the case of two different dimensions  $d_1, d_2$  for the two Krylov spaces, but we restrict to  $d_1 = d_2 \equiv d$  to simplify notation.

is an oblivious  $\varepsilon$ -subspace embedding for subspaces of dimension  $d$ , if—with high probability—condition (2.2) holds for *any* subspace  $\mathcal{V}$  with  $\dim(\mathcal{V}) = d$ . More precisely, the inputs for constructing  $\mathbf{S}$  are the embedding quality  $\varepsilon$ , the sketching dimension  $s$  and an accepted failure probability  $\delta$ . Based on these inputs,  $\mathbf{S}$  is then constructed using probabilistic methods. We refer to [32, Section 8] and [50] for an extensive treatment of oblivious subspace embeddings. As a specific example, assume that  $\mathbf{S}$  is constructed as a subsampled trigonometric transform,

$$\mathbf{S} = \sqrt{\frac{s}{n}} \mathbf{D} \mathbf{N} \mathbf{E},$$

where  $\mathbf{E} \in \mathbb{R}^{n \times n}$  is a diagonal matrix with entries  $\pm 1$  chosen randomly (with equal probability),  $\mathbf{D} \in \mathbb{R}^{s \times n}$  contains  $s$  randomly selected rows of the identity matrix, and  $\mathbf{N} \in \mathbb{R}^{n \times n}$  is a trigonometric transform (e.g., discrete Fourier transform, discrete cosine transform, Walsh–Hadamard transform). Then, one can show that  $\mathbf{S}$  is an oblivious  $\varepsilon$ -subspace embedding with failure probability  $\delta$  if the sketching dimension satisfies  $s = \mathcal{O}(\varepsilon^{-2}(d + \log \frac{n}{\delta}) \log \frac{d}{\delta})$ . When implemented properly, such  $\mathbf{S}$  can be applied at a cost of  $\mathcal{O}(n \log s)$  flops.<sup>2</sup> Numerical evidence suggests that, despite certain academic examples for which it fails, selecting  $s = \mathcal{O}(\varepsilon^{-2} \frac{d}{\delta})$  works well in practice; see, e.g., [48] and [32, Section 9].

All computationally feasible classes of matrices from which one can draw  $\mathbf{S}$  have in common that the necessary sketching dimension  $s$  depends quadratically on the embedding quality  $\varepsilon$ . As a consequence one needs to work with rather crude accuracies, such as  $\varepsilon = 1/\sqrt{2}$  or  $\varepsilon = 1/2$ , i.e., relatively large distortions in (2.2).

Oblivious subspace embeddings are key in our context, as the spaces we want to embed will be certain Krylov subspaces (see Section 2.3 and Section 3 for details), that we do not know a priori, at the start of the method.

We mention in passing that an alternative viewpoint on subspace embeddings is that a subspace embedding  $\mathbf{S}$  induces a semidefinite inner product

$$\langle u, v \rangle_{\mathbf{S}} := \langle \mathbf{S}u, \mathbf{S}v \rangle,$$

which, when restricted to  $\mathcal{V}$ , is an actual (positive definite) inner product; see, e.g., [6, Section 3.1].

This viewpoint is particularly relevant for establishing a connection between our methodology and other randomized Krylov techniques that have been recently proposed in the literature; see the discussion in Section 2.3.

**2.3. The sketched Arnoldi relation.** The combination of Krylov subspace methods with randomized oblivious subspace embeddings in order to increase computational efficiency in the orthogonalization step (and potentially reduce storage requirements and communication) is a topic that has recently gained a lot of traction and was investigated in the context of linear systems and eigenvalue computations [4, 5, 33] as well as in the approximation of matrix functions [13, 15, 24, 36]. The approach taken in [24, 33, 36] advocates the use of partially orthogonalized Krylov bases together with a subspace embedding  $\mathbf{S}$  that allows to “approximately orthogonalize” the Krylov basis at a much lower cost.

<sup>2</sup>When selecting  $\mathbf{S}$  as a matrix with Gaussian random entries, one actually obtains the stronger guarantee that  $s = \mathcal{O}(\varepsilon^{-2}(d + \log \frac{1}{\delta}))$  is sufficient, but these are less attractive computationally because they are more costly to apply.

In order to analyze the resulting methods and allow to compare them to their standard, non-sketched counterparts, a *sketched Arnoldi relation* is derived in [36], which we recall in the following. Assume that a truncated (block) Arnoldi procedure is employed for constructing bases of  $\mathcal{K}_d(\mathbf{A}, C_1)$  and  $\mathcal{K}_d(\mathbf{B}^\top, C_2)$ , where orthogonalization only against the previous  $k < d$  blocks is performed, rather than against all  $d$  previous blocks, followed by the orthogonalization of the columns of the new block; see also [42, Algorithm 6.6] for a discussion of the non-block case. This results in the truncated Arnoldi relations

$$(2.4) \quad \mathbf{A}\mathbf{U}_d = \mathbf{U}_{d+1}\mathbf{H}_d, \quad \mathbf{B}^\top\mathbf{V}_d = \mathbf{V}_{d+1}\mathbf{G}_d,$$

where  $\mathbf{H}_d$  and  $\mathbf{G}_d$  are upper Hessenberg and banded (with upper bandwidth  $kr$ ), and  $\mathbf{U}_d, \mathbf{V}_d$  do not have all orthogonal columns. We denote  $\mathbf{H}_d = [\mathbf{H}_d; H^\top]$ ,  $\mathbf{G}_d = [\mathbf{G}_d; G^\top]$ , where only the last  $r \times r$  blocks of  $H^\top = \mathbf{h}_{d+1,d}E_d^\top$  and  $G^\top = \mathbf{g}_{d+1,d}E_d^\top$  are nonzero.<sup>3</sup>

It is known—e.g., from the linear system case [46]—that basing a projection method on the quantities from (2.4) might delay convergence (and the same effect is also observed for matrix equations in our experiments in Section 6). One possible remedy for this is to combine the truncated block Arnoldi method with oblivious subspace embeddings in order to cheaply (approximately) orthogonalize the non-orthogonal bases  $\mathbf{U}_{d+1}, \mathbf{V}_{d+1}$ , as it was recently proposed for linear systems and matrix function computations in [24, 33, 36].

The basis for analyzing this approach is given by the following “sketched Arnoldi relation” from [36]. We assume that  $\mathbf{S}_U, \mathbf{S}_V$  are sketching matrices for  $\mathcal{K}_{d+1}(\mathbf{A}, C_1)$  and  $\mathcal{K}_{d+1}(\mathbf{B}^\top, C_2)$ , respectively.

**Proposition 2.1.** [Proposition 3.1 in [36], adapted to the block setting] Let  $\mathbf{S}_U\mathbf{U}_{d+1} = \mathbf{Q}_{U,d+1}\mathbf{T}_{U,d+1}$  be a reduced QR decomposition with

$$\mathbf{Q}_{U,d+1} = [\mathbf{Q}_{U,d}, \mathbf{Q}_{U,d+1}] \text{ and } \mathbf{T}_{U,d+1} = \begin{bmatrix} \mathbf{T}_{U,d} & T_H \\ 0^\top & \tau_{d+1} \end{bmatrix}.$$

Then, for the sketched method, the following Arnoldi-like formula holds:

$$(2.5) \quad \mathbf{S}_U\mathbf{A}\mathbf{U}_d = \mathbf{S}_U\mathbf{U}_d(\mathbf{H}_d + R_H E_d^\top) + \mathbf{Q}_{U,d+1}\tau_{d+1}\mathbf{h}_{d+1,d}E_d^\top,$$

with  $R_H = \mathbf{T}_{U,d}^{-1}T_H\mathbf{h}_{d+1,d}$  and  $\mathbf{Q}_{U,d+1} \perp \mathbf{S}_U\mathbf{U}_d$ . Similarly, if

$$\mathbf{S}_V\mathbf{V}_{d+1} = \mathbf{Q}_{V,d+1}\mathbf{T}_{V,d+1}, \quad \mathbf{T}_{V,d+1} = \begin{bmatrix} \mathbf{T}_{V,d} & T_G \\ 0^\top & \theta_{d+1} \end{bmatrix},$$

then

$$(2.6) \quad \mathbf{S}_V\mathbf{B}^\top\mathbf{V}_d = \mathbf{S}_V\mathbf{V}_d(\mathbf{G}_d + R_G E_d^\top) + \mathbf{Q}_{V,d+1}\theta_{d+1}\mathbf{g}_{d+1,d}E_d^\top,$$

where  $R_G := \mathbf{T}_{V,d}^{-1}T_G\mathbf{g}_{d+1,d}$ .

The reduced QR decompositions of  $\mathbf{S}_U\mathbf{U}_{d+1}$  and  $\mathbf{S}_V\mathbf{V}_{d+1}$  occurring in Proposition 2.1 can also be used as a means of orthogonalizing the bases  $\mathbf{U}_d$  and  $\mathbf{V}_d$  with respect to the inner product  $\langle \cdot, \cdot \rangle_{\mathbf{S}}$ , via the transformations  $\mathbf{U}_d \rightarrow \mathbf{U}_d\mathbf{T}_{U,d}^{-1}$  and

<sup>3</sup>Following the convention from [36], we use roman font  $(\mathbf{H}_d, \mathbf{U}_d)$  for quantities related to a non-orthogonal Arnoldi relation and calligraphic font  $(\mathcal{H}_d, \mathbf{u}_d)$  for quantities related to a fully orthogonal Arnoldi relation.

$\mathbf{V}_d \rightarrow \mathbf{V}_d \mathbf{T}_{V,d}^{-1}$ , respectively. Thanks to the  $\varepsilon$ -subspace embedding property (2.3), the bases

$$(2.7) \quad \widehat{\mathbf{U}}_d := \mathbf{U}_d \mathbf{T}_{U,d}^{-1}, \quad \widehat{\mathbf{V}}_d := \mathbf{V}_d \mathbf{T}_{V,d}^{-1},$$

are very well-behaved, conditioned to the considered probability. Indeed, with high probability it holds ([5, Corollary 2.2])

$$\kappa_2(\widehat{\mathbf{U}}_d) \leq \sqrt{\frac{1+\varepsilon}{1-\varepsilon}},$$

where  $\kappa_2$  denotes the two-norm condition number, and similarly for  $\widehat{\mathbf{V}}_d$ . Rewriting the relations (2.5)–(2.6) in terms of the bases  $\widehat{\mathbf{U}}_d, \widehat{\mathbf{V}}_d$  yields the following *whitened-sketched* truncated block Arnoldi (WS-Arnoldi) relations ([36, Section 3])

$$(2.8) \quad \begin{aligned} \mathbf{S}_U \mathbf{A} \widehat{\mathbf{U}}_d &= \mathbf{S}_U \widehat{\mathbf{U}}_d (\widehat{\mathbf{H}}_d + \widehat{\mathbf{H}} E_d^\top) + Q_{U,d+1} \mathbf{h}_{d+1,d} E_d^\top, \\ \mathbf{S}_V \mathbf{B}^\top \widehat{\mathbf{V}}_d &= \mathbf{S}_V \widehat{\mathbf{V}}_d (\widehat{\mathbf{G}}_d + \widehat{\mathbf{G}} E_d^\top) + Q_{V,d+1} \mathbf{g}_{d+1,d} E_d^\top, \end{aligned}$$

where

$$(2.9) \quad \begin{aligned} \widehat{\mathbf{H}}_d &= \mathbf{T}_{U,d} \mathbf{H}_d \mathbf{T}_{U,d}^{-1}, & \widehat{\mathbf{H}} &= T_H \mathbf{h}_{d+1,d} \boldsymbol{\tau}_d^{-1}, \\ \widehat{\mathbf{G}}_d &= \mathbf{T}_{V,d} \mathbf{G}_d \mathbf{T}_{V,d}^{-1}, & \widehat{\mathbf{G}} &= T_G \mathbf{g}_{d+1,d} \boldsymbol{\theta}_d^{-1}. \end{aligned}$$

Finally, from (2.8) we can derive the relations

$$(2.10) \quad \widehat{\mathbf{U}}_d^\top \mathbf{S}_U^\top \mathbf{S}_U \mathbf{A} \widehat{\mathbf{U}}_d = \widehat{\mathbf{H}}_d + \widehat{\mathbf{H}} E_d^\top, \quad \widehat{\mathbf{V}}_d^\top \mathbf{S}_V^\top \mathbf{S}_V \mathbf{B}^\top \widehat{\mathbf{V}}_d = \widehat{\mathbf{G}}_d + \widehat{\mathbf{G}} E_d^\top,$$

which emphasize the role of the right-hand side matrices as projection and reduction of the matrices  $\mathbf{A}$  and  $\mathbf{B}^\top$  onto the generated spaces, in the  $\mathbf{S}_U^\top \mathbf{S}_U$  and  $\mathbf{S}_V^\top \mathbf{S}_V$  inner product, respectively. We stress that the full matrices  $\widehat{\mathbf{U}}_d, \widehat{\mathbf{V}}_d$  are never explicitly constructed in practice, whereas the small matrices  $\widehat{\mathbf{H}}_d, \widehat{\mathbf{G}}_d$  are used to derive the sought after problem approximation.

We end this section by commenting on the connection between the methodology outlined above and other randomized Krylov methods that were recently proposed, in particular those based on the *randomized Gram–Schmidt (RGS) process* [4, 5, 15, 47]. In these methods, one uses a Gram–Schmidt procedure—with respect to the sketched inner product (2.2) instead of the usual Euclidean inner product—for constructing the Krylov basis. In exact arithmetic, this approach yields the same bases as (2.7) if the QR decompositions are computed without interchanging columns. In particular, the WS-Arnoldi relations (2.8) and (2.10) are equivalent to relations obtained in [47, Section 3.2], but they are phrased differently and reveal different structural properties.

Thus, as was already noted in [47], methods based on the RGS process and those based on truncated Arnoldi and sketching are mathematically equivalent. There are several notable differences, though: Performing the RGS process necessarily requires keeping the whole Krylov basis in memory, and the asymptotic cost of the orthogonalization still scales as  $\mathcal{O}((rd)^2 n)$ . Indeed, while inner products are computed with vectors of length  $s$ , all other vector operations have the same cost as in standard Gram–Schmidt. In contrast, when using truncated Arnoldi with sketching, only the last  $k$  basis blocks need to be kept in memory, as the coefficients for the orthogonalization in the semidefinite inner product (2.2) are computed from the sketched bases  $\mathbf{S}_U \mathbf{U}_{d+1}, \mathbf{S}_V \mathbf{V}_{d+1}$ . As we will explain in Section 3, the transformations (2.7) need not be performed explicitly, so that using this approach allows to



use a two-pass strategy for reducing memory consumption; see [Section 5](#) below. In summary, our approach reduces the asymptotic cost and memory consumption of the basis construction in comparison to RGS. On the other hand, methods based on RGS can in general be expected to be more stable, as the truncated block Arnoldi process and the retrospective orthogonalization via  $T_{U,d}^{-1}$  and  $T_{V,d}^{-1}$  can potentially be sources of instability. We refer to [\[15\]](#), where it is illustrated (for the problem of computing  $f(A)b$ , the action of a matrix function on a vector) that RGS-based methods might be successful in certain examples where those based on truncated orthogonalization are not. The precise effects causing this difference in performance are not yet fully understood, though. In particular, they are not merely related to ill-conditioning of the bases from truncated Arnoldi: As examples in [\[24, 36\]](#) as well as in [Section 6](#) below show, the methods often continue to work well even in presence of a numerically singular basis.

### 3. SKETCHED AND TRUNCATED KRYLOV METHODS FOR MATRIX EQUATIONS

We now turn more specifically to the solution of the Sylvester equation [\(1.1\)](#) by means of projection onto the generated spaces; see, e.g., [\[44\]](#). We seek an approximation in the form  $\mathbf{X}_d = \mathbf{U}_d \mathbf{Y}_d \mathbf{V}_d^\top$ , with  $\mathbf{U}_d, \mathbf{V}_d$  generated above and  $\mathbf{Y}_d \in \mathbb{R}^{dr \times dr}$  computed by solving some *surrogate* problem. Using the truncated Arnoldi relations in [\(2.4\)](#) and  $\mathbf{X}_d = \mathbf{U}_d \mathbf{Y}_d \mathbf{V}_d^\top$ , the residual matrix  $\mathbf{R}_d = \mathbf{A}\mathbf{X}_d + \mathbf{X}_d\mathbf{B} - C_1 C_2^\top$  can be written as

$$(3.1) \quad \mathbf{R}_d = [\mathbf{U}_d, \mathbf{U}_{d+1}] \begin{bmatrix} \mathbf{H}_d \mathbf{Y}_d + \mathbf{Y}_d \mathbf{G}_d^\top - E_1 \boldsymbol{\beta}_1 \boldsymbol{\beta}_2^\top E_1^\top & \mathbf{Y}_d E_d \mathbf{g}_{d+1,d}^\top \\ \mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}_d & 0 \end{bmatrix} [\mathbf{V}_d, \mathbf{V}_{d+1}]^\top,$$

where  $C_1 = \mathbf{U}_d E_1 \boldsymbol{\beta}_1$ ,  $C_2 = \mathbf{V}_d E_1 \boldsymbol{\beta}_2$ , and  $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathbb{R}^{r \times r}$ . The matrix  $\mathbf{Y}_d$  can be obtained by, e.g., imposing some optimality condition on the residual matrix, such as orthogonality with respect to the approximation bases, i.e., the so-called Galerkin condition,  $\mathbf{U}_d^\top \mathbf{R}_d \mathbf{V}_d = 0$ . If  $\mathbf{U}_{d+1} = [\mathbf{U}_d, \mathbf{U}_{d+1}]$  and  $\mathbf{V}_{d+1} = [\mathbf{V}_d, \mathbf{V}_{d+1}]$  had orthonormal columns, this condition would correspond to zeroing out the top left term in the inner block matrix in [\(3.1\)](#), thus solving  $\mathbf{H}_d \mathbf{Y}_d + \mathbf{Y}_d \mathbf{G}_d^\top = E_1 \boldsymbol{\beta}_1 \boldsymbol{\beta}_2^\top E_1^\top$  for  $\mathbf{Y}_d$ . Hence, it would also hold that  $\|\mathbf{R}_d\|_F^2 = \|\mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}_d\|_F^2 + \|\mathbf{Y}_d E_{d+1} \mathbf{g}_{d+1,d}^\top\|_F^2$ ; see, e.g., [\[44\]](#). In our setting, not all the columns of  $\mathbf{U}_{d+1}$  and  $\mathbf{V}_{d+1}$  are orthonormal. Nonetheless, by denoting  $\mathbf{X}_d^{\text{TR}} = \mathbf{U}_d \mathbf{Y}_d^{\text{TR}} \mathbf{V}_d^\top$ , we can still compute  $\mathbf{Y}_d^{\text{TR}}$  by solving the same equation, namely

$$\mathbf{H}_d \mathbf{Y}_d^{\text{TR}} + \mathbf{Y}_d^{\text{TR}} \mathbf{G}_d^\top = E_1 \boldsymbol{\beta}_1 \boldsymbol{\beta}_2^\top E_1^\top.$$

Notice that, in contrast to what happens in the full Arnoldi method, this matrix equation is not what would be obtained by imposing a Galerkin condition on the residual matrix, due to the non-orthogonality of the bases<sup>4</sup>. On the other hand, it allows us to derive an approximation without explicitly storing the two bases, and a computable estimate for the residual norm. In particular, for the residual matrix  $\mathbf{R}_d^{\text{TR}}$  stemming from the truncated scheme, the following result holds true.

**Proposition 3.1.** Let  $\mathbf{R}_d^{\text{TR}} = \mathbf{A}\mathbf{X}_d^{\text{TR}} + \mathbf{X}_d^{\text{TR}}\mathbf{B} - C_1 C_2^\top$ ,  $\mathbf{X}_d^{\text{TR}} = \mathbf{U}_d \mathbf{Y}_d^{\text{TR}} \mathbf{V}_d^\top$ , be the residual matrix stemming from the  $d$ th iteration of the truncated Krylov method.

<sup>4</sup>The condition  $\mathbf{U}_d^\top \mathbf{R}_d \mathbf{V}_d = 0$  yields the reduced equation  $\mathbf{U}_d^\top \mathbf{A} \mathbf{U}_d \mathbf{Y}_d \mathbf{V}_d^\top \mathbf{V}_d + \mathbf{U}_d^\top \mathbf{U}_d \mathbf{Y}_d \mathbf{V}_d^\top \mathbf{B} \mathbf{V}_d = \mathbf{U}_d^\top C_1 C_2^\top \mathbf{V}_d$ .

Then

$$(3.2) \quad \|\mathbf{R}_d^{\text{TR}}\|_F \leq \sqrt{dr}(\|\mathbf{Y}_d^{\text{TR}} E_d \mathbf{g}_{d+1,d}^\top\|_F + \|\mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}_d^{\text{TR}}\|_F).$$

*Proof.* The expression in (3.1) is equivalent to writing  $\mathbf{R}_d^{\text{TR}}$  as

$$\mathbf{R}_d^{\text{TR}} = \mathbf{U}_d \mathbf{Y}_d^{\text{TR}} E_d \mathbf{g}_{d+1,d}^\top \mathbf{V}_{d+1}^\top + U_{d+1} \mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}_d^{\text{TR}} \mathbf{V}_d^\top.$$

Therefore, by recalling that  $U_{d+1}^\top U_{d+1} = V_{d+1}^\top V_{d+1} = \mathbf{I}_r$ , it holds

$$\|\mathbf{R}_d^{\text{TR}}\|_F \leq \|\mathbf{U}_d\|_F \|\mathbf{Y}_d^{\text{TR}} E_d \mathbf{g}_{d+1,d}^\top\|_F + \|\mathbf{V}_d\|_F \|\mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}_d^{\text{TR}}\|_F.$$

The result follows by noticing that  $\|\mathbf{U}_d\|_F = \|\mathbf{V}_d\|_F = \sqrt{dr}$ , since  $\mathbf{U}_d, \mathbf{V}_d \in \mathbb{R}^{n \times dr}$  have unit norm columns.  $\square$

The truncated Krylov method we just outlined is reminiscent of truncated methods for linear systems; see, e.g. [42, Sections 6.4.2 and 6.5.6]. To the best of our knowledge, these truncated Arnoldi methods have never been used in the matrix equation setting.

We now turn our attention to how we can integrate sketching and Krylov schemes for matrix equations. If sketching is employed, we look for an approximate solution  $\mathbf{X}_d^{\text{SK}} = \widehat{\mathbf{U}}_d \mathbf{Y}_d^{\text{SK}} \widehat{\mathbf{V}}_d^\top$ . Thanks to the WS-Arnoldi relations (2.8), we generalize the previous relation (3.1) for the residual to get

(3.3)

$$\begin{aligned} & \mathbf{S}_U \mathbf{R}_d^{\text{SK}} \mathbf{S}_V^\top \\ &= \mathbf{S}_U \widehat{\mathbf{U}}_{d+1} \begin{bmatrix} (\widehat{\mathbf{H}}_d + \widehat{H} E_d^\top) \mathbf{Y}_d^{\text{SK}} + \mathbf{Y}_d^{\text{SK}} (\widehat{\mathbf{G}}_d + \widehat{G} E_d^\top)^\top - E_1 \boldsymbol{\beta}_1 \boldsymbol{\beta}_2^\top E_1^\top & \mathbf{Y}_d^{\text{SK}} E_d \mathbf{g}_{d+1,d}^\top \\ \mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}_d^{\text{SK}} & 0 \end{bmatrix} \widehat{\mathbf{V}}_{d+1}^\top \mathbf{S}_V^\top, \end{aligned}$$

where now  $C_1 = \widehat{\mathbf{U}}_d E_1 \boldsymbol{\beta}_1$ ,  $C_2 = \widehat{\mathbf{V}}_d E_1 \boldsymbol{\beta}_2$ . It is thus again natural to compute  $\mathbf{Y}_d^{\text{SK}}$  so as to annihilate the (1,1)-block in the inner matrix, that is

$$(3.4) \quad (\widehat{\mathbf{H}}_d + \widehat{H} E_d^\top) \mathbf{Y}_d^{\text{SK}} + \mathbf{Y}_d^{\text{SK}} (\widehat{\mathbf{G}}_d + \widehat{G} E_d^\top)^\top = E_1 \boldsymbol{\beta}_1 \boldsymbol{\beta}_2^\top E_1^\top.$$

This way of determining  $\mathbf{Y}_d^{\text{SK}}$  stands on more solid grounds than in the truncated case. Indeed, the columns of  $\widehat{\mathbf{U}}_d$  and  $\widehat{\mathbf{V}}_d$  are orthogonal bases with respect to the sketched inner products defined by  $\mathbf{S}_U^\top \mathbf{S}_U$  and  $\mathbf{S}_V^\top \mathbf{S}_V$ , respectively. Thus, solving (3.4) is equivalent to imposing the following Galerkin  $\mathbf{S}_*^\top \mathbf{S}_*$ -orthogonality condition on the residual matrix, namely

$$\widehat{\mathbf{U}}_d^\top \mathbf{S}_U^\top \mathbf{S}_U \mathbf{R}_d^{\text{SK}} \mathbf{S}_V^\top \mathbf{S}_V \widehat{\mathbf{V}}_d = 0.$$

The equivalence follows from the expression (3.3) and the WS-Arnoldi relations (2.10).

For the sketched-and-truncated method, the Frobenius norm of the sketched residual, namely  $\|\mathbf{S}_U \mathbf{R}_d^{\text{SK}} \mathbf{S}_V^\top\|_F$ , can be cheaply evaluated as

$$(3.5) \quad \|\mathbf{S}_U \mathbf{R}_d^{\text{SK}} \mathbf{S}_V^\top\|_F^2 = \|\mathbf{Y}_d^{\text{SK}} E_d \mathbf{g}_{d+1,d}^\top\|_F^2 + \|\mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}_d^{\text{SK}}\|_F^2,$$

where (3.3) was used together with the orthogonality of the bases  $\widehat{\mathbf{U}}_d$  and  $\widehat{\mathbf{V}}_d$  with respect to the sketched inner products. Indeed, this corresponds to using the  $\mathbf{S}_*^\top \mathbf{S}_*$ -norm, supported by the fact that this will be close to the Frobenius norm thanks to the  $\varepsilon$ -subspace embedding property. More precisely, with high probability,

$$(3.6) \quad (1 + \tilde{\varepsilon})^{-1} \|\mathbf{S}_U \mathbf{R}_d^{\text{SK}} \mathbf{S}_V^\top\|_F^2 \leq \|\mathbf{R}_d^{\text{SK}}\|_F^2 \leq (1 - \tilde{\varepsilon})^{-1} \|\mathbf{S}_U \mathbf{R}_d^{\text{SK}} \mathbf{S}_V^\top\|_F^2,$$

where  $\tilde{\varepsilon} = \varepsilon(2 + \varepsilon)$ ; see Appendix B.

**3.1. Distance from the Arnoldi approximation.** It is relevant to study the discrepancy between the solution to the optimal reduced equation and the solution  $\mathbf{Y}_d^{\text{SK}}$  to the sketched reduced equation (3.4). Analogous results can be obtained for the truncated approximation. The full Galerkin method determines the Arnoldi approximation  $\mathbf{X}_d^{\text{FULL}} = \mathbf{u}_d \mathbf{y}_d^{\text{FULL}} \mathbf{v}_d^\top$ , where  $\mathbf{y}_d^{\text{FULL}}$  is such that

$$(3.7) \quad \mathcal{H}_d \mathbf{y}_d^{\text{FULL}} + \mathbf{y}_d^{\text{FULL}} \mathcal{G}_d^\top = E_1 \boldsymbol{\beta}_1^{\text{FULL}} (\boldsymbol{\beta}_2^{\text{FULL}})^\top E_1^\top,$$

with  $\mathbf{u}_d, \mathbf{v}_d$  the orthonormal bases,  $\mathcal{G}_d, \mathcal{H}_d$  the projected matrices from (2.1), and  $C_1 = \mathcal{U}_1 \boldsymbol{\beta}_1^{\text{FULL}}, C_2 = \mathcal{V}_1 \boldsymbol{\beta}_2^{\text{FULL}}$ ; see, e.g., [44].

Thanks to the sketched Arnoldi relation in Proposition 2.1, it is possible to easily represent the sketched approximation in terms of the orthogonal basis  $\mathbf{u}_d$  instead of the basis  $\widehat{\mathbf{U}}_d$ ; see [36, Section 6]. First, note that there exist upper triangular matrices  $\mathcal{T}_U, \mathcal{T}_V$  such that  $\widehat{\mathbf{U}}_d = \mathbf{u}_d \mathcal{T}_U$  and  $\widehat{\mathbf{V}}_d = \mathbf{v}_d \mathcal{T}_V$ . Analogously to how (2.8) follows from (2.7) and (2.9), we can thus derive the relations

$$\begin{aligned} \mathcal{T}_U (\widehat{\mathbf{H}}_d + \widehat{\mathbf{H}} E_d^\top) \mathcal{T}_U^{-1} &= \mathcal{H}_d - \mathbf{u}_d^\top \widehat{\mathbf{U}}_{d+1} \mathbf{h}_{d+1,d} \mathbf{t}_{U,d}^{-1} E_d^\top, \\ \mathcal{T}_V (\widehat{\mathbf{G}}_d + \widehat{\mathbf{G}} E_d^\top) \mathcal{T}_V^{-1} &= \mathcal{G}_d - \mathbf{v}_d^\top \widehat{\mathbf{V}}_{d+1} \mathbf{g}_{d+1,d} \mathbf{t}_{V,d}^{-1} E_d^\top, \end{aligned}$$

where  $\mathbf{t}_{U,d} = E_d^\top \mathcal{T}_U E_d$  and  $\mathbf{t}_{V,d} = E_d^\top \mathcal{T}_V E_d$ . Therefore, by pre- and postmultiplying (3.4) by  $\mathcal{T}_U$  and  $\mathcal{T}_V^\top$ , respectively, we can write  $\mathbf{X}_d^{\text{SK}} = \mathbf{u}_d \mathbf{y}_d^{\text{SK}} \mathbf{v}_d^\top$ , where  $\mathbf{y}_d^{\text{SK}} = \mathcal{T}_U \mathbf{Y}_d^{\text{SK}} \mathcal{T}_V^\top$  solves

$$(3.8) \quad (\mathcal{H}_d - \widehat{\mathbf{R}}_H E_d^\top) \mathbf{y}_d^{\text{SK}} + \mathbf{y}_d^{\text{SK}} (\mathcal{G}_d - \widehat{\mathbf{R}}_G E_d^\top)^\top = E_1 \boldsymbol{\beta}_1^{\text{FULL}} (\boldsymbol{\beta}_2^{\text{FULL}})^\top E_1^\top,$$

with  $\widehat{\mathbf{R}}_H := \mathbf{u}_d^\top \widehat{\mathbf{U}}_{d+1} \mathbf{h}_{d+1,d} \mathbf{t}_{U,d}^{-1}$ ,  $\widehat{\mathbf{R}}_G := \mathbf{v}_d^\top \widehat{\mathbf{V}}_{d+1} \mathbf{g}_{d+1,d} \mathbf{t}_{V,d}^{-1}$ . The right-hand side is derived as  $\mathcal{T}_U E_1 \boldsymbol{\beta}_1 = E_1 \mathbf{t}_{U,1} \boldsymbol{\beta}_1 = E_1 \boldsymbol{\beta}_1^{\text{FULL}}$ . Similarly for  $\boldsymbol{\beta}_2^{\text{FULL}}$ .

Subtracting (3.8) from (3.7) we obtain

$$\mathcal{H}_d (\mathbf{y}_d^{\text{FULL}} - \mathbf{y}_d^{\text{SK}}) + (\mathbf{y}_d^{\text{FULL}} - \mathbf{y}_d^{\text{SK}}) \mathcal{G}_d^\top = -\widehat{\mathbf{R}}_H E_d^\top \mathbf{y}_d^{\text{SK}} - \mathbf{y}_d^{\text{SK}} E_d \widehat{\mathbf{R}}_G^\top.$$

Hence, for  $\mathcal{L} = \mathcal{G}_d \otimes \mathbf{I} + \mathbf{I} \otimes \mathcal{H}_d$ , we have

$$\|\mathbf{X}_d^{\text{FULL}} - \mathbf{X}_d^{\text{SK}}\|_F = \|\mathbf{y}_d^{\text{FULL}} - \mathbf{y}_d^{\text{SK}}\|_F \leq \|\mathcal{L}^{-1}\|_F \|\widehat{\mathbf{R}}_H E_d^\top \mathbf{y}_d^{\text{SK}} + \mathbf{y}_d^{\text{SK}} E_d \widehat{\mathbf{R}}_G^\top\|_F.$$

The above bound estimates the distance  $\|\mathbf{X}_d^{\text{FULL}} - \mathbf{X}_d^{\text{SK}}\|_F$  in terms of a quantity associated with the sketched residual norm, and the spectral term  $\|\mathcal{L}^{-1}\|_F$ , which coincides with the reciprocal of the separation between the matrices  $\mathcal{G}_d$  and  $\mathcal{H}_d$ . Indeed, starting from (3.3) with  $\mathbf{y}_d^{\text{SK}}$  solving the reduced equation, the residual matrix can be written in the (ideal) orthonormal bases as

$$\mathbf{R}_d^{\text{SK}} = \mathbf{u}_{d+1} \begin{bmatrix} 0 & \mathbf{y}_d^{\text{SK}} E_d \widehat{\mathbf{R}}_G^\top \\ \widehat{\mathbf{R}}_H E_d^\top \mathbf{y}_d^{\text{SK}} & 0 \end{bmatrix} \mathbf{v}_{d+1}^\top,$$

so that both addends in the bound factor  $\|\widehat{\mathbf{R}}_H E_d^\top \mathbf{y}_d^{\text{SK}} + \mathbf{y}_d^{\text{SK}} E_d \widehat{\mathbf{R}}_G^\top\|_F$  go to zero as the residual norm goes to zero. Hence, for a well-posed full orthogonalization procedure, we expect that, in case of convergence, the whitened-sketched scheme yields a solution not far from that of the full orthogonalization method.

One could also write an analogous bound for the error norms in terms of the bottom entries of  $\mathbf{y}_d^{\text{FULL}}$ , for which we have a better understanding of the convergence behavior. The proof follows the derivation above, by first substituting  $\mathbf{y}_d^{\text{SK}}$  with

$(\mathbf{y}_d^{\text{SK}} - \mathbf{y}_d^{\text{FULL}}) + \mathbf{y}_d^{\text{FULL}}$  into (3.8) to get

$$\begin{aligned} & (\mathcal{H}_d - \widehat{R}_H E_d^\top)(\mathbf{y}_d^{\text{FULL}} - \mathbf{y}_d^{\text{SK}}) + (\mathbf{y}_d^{\text{FULL}} - \mathbf{y}_d^{\text{SK}})(\mathcal{G}_d - \widehat{R}_G E_d^\top)^\top \\ &= -\widehat{R}_H E_d^\top \mathbf{y}_d^{\text{FULL}} - \mathbf{y}_d^{\text{FULL}} E_d \widehat{R}_G^\top, \end{aligned}$$

so that

$$\|\mathbf{y}_d^{\text{FULL}} - \mathbf{y}_d^{\text{SK}}\|_F \leq \|\widetilde{\mathcal{L}}^{-1}\|_F \|\widehat{R}_H E_d^\top \mathbf{y}_d^{\text{FULL}} + \mathbf{y}_d^{\text{FULL}} E_d \widehat{R}_G^\top\|_F,$$

where  $\widetilde{\mathcal{L}} = (\mathcal{G}_d - \widehat{R}_G E_d^\top) \otimes \mathbf{I} + \mathbf{I} \otimes (\mathcal{H}_d - \widehat{R}_H E_d^\top)$ . On the other hand, the spectral term  $\|\widetilde{\mathcal{L}}^{-1}\|_F$  of the resulting bound is not easy to control as it depends on low-rank modifications to  $\mathcal{H}_d$  and  $\mathcal{G}_d$ .

Following the same reasonings as above, very similar results can be derived for the truncated approach. In this case, we can also borrow some of the results from the theory of truncated algorithms for (vector) linear systems. Indeed, using the vector form, we have that  $x_d^{\text{TR}} = \text{vec}(\mathbf{X}_d^{\text{TR}}) = (\mathbf{V}_d \otimes \mathbf{U}_d) y_d^{\text{TR}}$  with  $y_d^{\text{TR}} = \text{vec}(\mathbf{Y}_d^{\text{TR}})$ . Under this form we can derive a relation with the optimal Arnoldi solution  $x_d^{\text{FULL}}$ , that is the one that keeps both bases orthonormal, namely  $\mathbf{V}_d \otimes \mathbf{U}_d$ . To this end, let  $\mathcal{M} = \mathbf{I} \otimes \mathbf{A} + \mathbf{B} \otimes \mathbf{I}$  and  $\mathcal{W} := \mathbf{V}_d \otimes \mathbf{U}_d$  be the Kronecker sum and product with quantities as in (2.4), and let  $\mathcal{P}_{\mathcal{M}}$  be the oblique projection  $\mathcal{P}_{\mathcal{M}} = \mathcal{M} \mathcal{W} (\mathcal{W}^\top \mathcal{M} \mathcal{W})^{-1} \mathcal{W}^\top$ , where we assume that the inner matrix is nonsingular, that is, the two spaces spanned by  $\mathbf{U}_d, \mathbf{V}_d$  keep growing. The following equality was proved in [46, Th. 3.1],

$$\frac{\|r_d^{\text{FULL}} - r_d^{\text{TR}}\|}{\|r_d^{\text{FULL}}\|} = \frac{\|\mathcal{P}_{\mathcal{M}} w\|}{\|(\mathbf{I} - \mathcal{P}_{\mathcal{M}}) w\|}, \quad w = v_{d+1} \otimes u_{d+1},$$

where  $r_d^{\text{FULL}}, r_d^{\text{TR}}$  are the vector residuals corresponding to the above approximations and for simplicity we have assumed a rank-one right-hand side matrix  $C_1 C_2^\top$ . This *equality* shows that as long as the next vectors of the bases,  $u_{d+1}, v_{d+1}$ , provide significant new information with respect to the previously computed vectors, then the two residuals will be close to each other. In other words, orthogonality is not paramount, only the angle between the new vectors and the previous spaces is.

#### 4. THEORETICAL INVESTIGATION

In this section we explore some of the theoretical aspects related to the presented sketching technique.

**4.1. On the field of values of the projected problem.** It is well-known that for standard Krylov subspace methods for Sylvester equations, the well-posedness of the projected problems can be guaranteed by assuming that the fields of values of  $\mathbf{A}$  and  $-\mathbf{B}$  have empty intersection, namely  $W(\mathbf{A}) \cap W(-\mathbf{B}) = \emptyset$ ; see, e.g., [44]. In the sketching framework, stronger assumptions have to be considered to ensure that equation (3.4) has a unique solution for any  $d$ .

In general, the field of values of the matrix associated with the sketching procedure is not a significant tool to analyze well-posedness, as sketching may substantially change the field of values of  $\mathbf{A}$ . In the following we illustrate the quality of this change. To this end, we first describe a general result on “sketched” fields of values that will then be adapted to our setting.

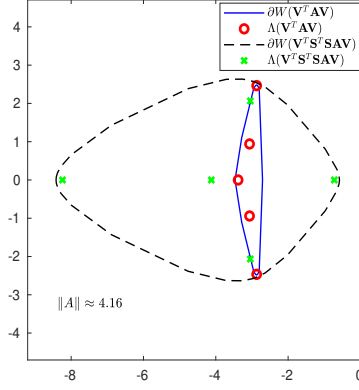


FIGURE 1. **Example 4.2.** Fields of values  $\partial W(\mathbf{V}^\top \mathbf{A} \mathbf{V})$  (solid blue line) and  $\partial W(\mathbf{V}^\top \mathbf{S}^\top \mathbf{S} \mathbf{A} \mathbf{V})$  (dashed black line), together with the corresponding eigenvalues.

**Proposition 4.1.** Let  $\mathbf{S}$  be an  $\varepsilon$ -subspace embedding for  $\mathcal{V}$ . Then for any complex vector  $v$  such that  $\mathbf{A}v \in \mathcal{V}$  it holds that

$$|\Re(v^* \mathbf{S}^\top \mathbf{S} \mathbf{A} v) - \Re(v^* \mathbf{A} v)| \leq \varepsilon \|v\| \|\mathbf{A} v\|,$$

with high probability, where  $\Re(\cdot)$  denotes the real part of a complex number, so that

$$\frac{1}{1+\varepsilon} \Re\left(\frac{v^* \mathbf{A} v}{v^* v}\right) - \varepsilon \|v\| \|\mathbf{A} v\| \leq \Re(v^* \mathbf{S}^\top \mathbf{S} \mathbf{A} v) \leq \frac{1}{1-\varepsilon} \Re\left(\frac{v^* \mathbf{A} v}{v^* v}\right) + \varepsilon \|v\| \|\mathbf{A} v\|.$$

*Proof.* Let  $v = v_1 + v_2$ . Then

$$\begin{aligned} \Re(v^* \mathbf{S}^\top \mathbf{S} \mathbf{A} v) &= \langle \mathbf{S} v_1, \mathbf{S} \mathbf{A} v_1 \rangle + \langle \mathbf{S} v_2, \mathbf{S} \mathbf{A} v_2 \rangle \\ &= \left\langle (I_2 \otimes \mathbf{S}) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, (I_2 \otimes \mathbf{S}) \begin{bmatrix} \mathbf{A} v_1 \\ \mathbf{A} v_2 \end{bmatrix} \right\rangle. \end{aligned}$$

By **Propositions B.1** to **B.2**,  $I_2 \otimes \mathbf{S}$  is an  $\varepsilon$ -subspace embedding for

$$\mathcal{V}_2 := \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x, y \in \mathcal{V} \right\},$$

so that by (2.3),

$$\Re(v^* \mathbf{A} v) - \varepsilon \|v\| \|\mathbf{A} v\| \leq \Re(v^* \mathbf{S}^\top \mathbf{S} \mathbf{A} v) \leq \Re(v^* \mathbf{A} v) + \varepsilon \|v\| \|\mathbf{A} v\|.$$

The factor involving  $1 \pm \varepsilon$  stems from (2.2).  $\square$

The bounds above help us appreciate the distance between elements in the sketched and nonsketched field of values. Indeed, let  $\text{range}(\widehat{\mathbf{U}}) = \mathcal{V}$ . If we write  $v = \widehat{\mathbf{U}}_d y$ , with  $y \neq 0$ , then it holds that  $\|\mathbf{S}_U v\| = \|y\|$  thanks to  $\mathbf{S}_U^\top \mathbf{S}_U$ -orthogonality of  $\widehat{\mathbf{U}}_d$ . Moreover, using **Proposition 2.1**

$$v^* \mathbf{S}_U^\top \mathbf{S}_U \mathbf{A} v = y^* \widehat{\mathbf{U}}_d^\top \mathbf{S}_U^\top \mathbf{S}_U \mathbf{A} \widehat{\mathbf{U}}_d y = y^* (\widehat{\mathbf{H}}_d + \widehat{\mathbf{H}} E_d^\top) y.$$

Hence, points of  $W(\widehat{\mathbf{H}}_d + \widehat{\mathbf{H}} E_d^\top)$  may be distant from  $W(\widehat{\mathbf{U}}^\top \mathbf{A} \widehat{\mathbf{U}})$  as much as  $\mathcal{O}(\|\mathbf{A}\|)$ . The following example makes this point clearer.

**Example 4.2.** Consider the following code, building the Arnoldi basis matrix  $\mathbf{V}$  for the given  $\mathbf{A}$  of size  $n \times n$ ;

```
n=30; rng(35);
A=-toeplitz([3,-1,-0.5,zeros(1,n-3)],[3,1,1,zeros(1,n-3)]);
b=ones(n,1);b=b/norm(b);
K=[b,A*b,A*(A*b), A*(A*(A*b)), A*(A*(A*(A*b)))]; % Krylov matrix
d=size(K,2); s=2*d;
[V,R]=qr(K,0); % Arnoldi basis
S = 1/sqrt(s)*randn(s,n);
```

Note that the Arnoldi relation can be recovered as  $\mathbf{A}*\mathbf{V}(:,1:d-1) = \mathbf{V}*\mathbf{H}$ ; with  $\mathbf{H}=\mathbf{R}(1:d,2:d)/\mathbf{R}(1:d-1,1:d-1)$ ; . The sketching operator  $\mathbf{S}$  is chosen as a random Gaussian matrix [32, 34]. Figure 1 reports the fields of values<sup>5</sup>  $W(\mathbf{V}^\top \mathbf{A} \mathbf{V})$  and  $W(\mathbf{V}^\top \mathbf{S}^\top \mathbf{S} \mathbf{A} \mathbf{V})$ , together with the corresponding eigenvalues. The quantity  $\|\mathbf{A}\|$  is also reported and the distance between the two boundaries clearly reaches the magnitude of this norm.

We next deepen the analysis of the properties of the projected equations (3.4). To simplify the presentation, we assume that the eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$  are simple to avoid working with invariant subspaces.

As a prerequisite for our analysis, we require the following technical lemma. A similar result was proved in [36, Section 7] using the eigenvalue decomposition of the given matrix. To deal with matrix equations, here we generalize the result to the use of the Schur decomposition, which appears to be more appropriate in this context. We stress that this result, together with the subsequent Proposition 4.4, are possible thanks to the low-rank modification form for the projected matrix derived in Proposition 2.1. Expressions for the projected matrix using the pseudoinverse, such as in [13, 15, 24], would not allow one to obtain these simple derivations.

**Lemma 4.3.** *Let  $(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)^\top = \mathcal{Q}\mathcal{R}\mathcal{Q}^*$  be the Schur decomposition of (the transpose of) the given matrix. For  $\eta > 0$ , assume that there exists an index  $\bar{i} \leq d$  such that  $\text{dist}(\mathcal{R}_{ii}, W(\widehat{\mathbf{H}}_d)) = \mathcal{O}(\eta^{-1})$ ,  $i = 1, \dots, \bar{i}$  for an appropriate spectral ordering in the Schur decomposition. Then it holds that  $\mathcal{Q}_{1,i} = \mathcal{O}((\eta/\xi)^d)$ , for  $i = 1, \dots, \bar{i}$ , with  $\xi = \max_{j \leq \bar{i}} |1/(\widehat{\mathbf{H}}_d)_{j+1,j}|$ .*

*Proof.* Let us write  $\mathcal{Q} = [q_1, \dots, q_d]$ . Consider the first column in the Schur decomposition, that is  $(\widehat{\mathbf{H}}_d^\top + e_d \widehat{h}^\top)q_1 = q_1 \mathcal{R}_{1,1}$ . Let  $\mathbf{N} := \widehat{\mathbf{H}}_d^\top$  and recall that  $\mathbf{N}$  is lower Hessenberg. For each row  $k < d$  of this eigenvalue equation we have

$$\mathbf{N}_{k,1:k+1}q_{1:k+1,1} = q_{k,1}\mathcal{R}_{1,1}.$$

Reordering terms,  $\mathbf{N}_{k,k+1}q_{k+1,1} = (\mathbf{N}_{k,k} - \mathcal{R}_{1,1})q_{k,1} + \mathbf{N}_{k,1:k-1}q_{1:k-1,1}$ , from which we obtain

$$q_{k+1,1} = \frac{(\mathbf{N}_{k,k} - \mathcal{R}_{1,1})}{\mathbf{N}_{k,k+1}}q_{k,1} + \frac{\mathbf{N}_{k,1:k-1}}{\mathbf{N}_{k,k+1}}q_{1:k-1,1}.$$

<sup>5</sup>The boundary of the field of values is approximated by the function `fv.m` in [28].

Taking into account the unit norm of  $q_1$  it follows

$$|q_{k+1,1}| \leq \frac{|\mathbf{N}_{k,k} - \mathcal{R}_{1,1}|}{|\mathbf{N}_{k,k+1}|} |q_{k,1}| + \frac{\|\mathbf{N}_{k,1:k-1}\|}{|\mathbf{N}_{k,k+1}|}.$$

Hence, the entries of  $q_1$  satisfy the relation  $|q_{k+1,1}| = \mathcal{O}(\xi\eta^{-1}|q_{k,1}|)$ . Normalization of  $q_1$  yields the sought after result.

For a subsequent Schur vector  $q_i$  with  $1 < i \leq \bar{i}$  we have

$$\mathbf{N}_{k,1:k+1}q_{1:k+1,i} = q_{k,i}\mathcal{R}_{i,i} + q_{k,1:i-1}\mathcal{R}_{1:i-1,i},$$

from which  $\mathbf{N}_{k,k+1}q_{k+1,i} = (\mathbf{N}_{k,k} - \mathcal{R}_{i,i})q_{k,i} + \mathbf{N}_{k,1:k-1}q_{1:k-1,i} + q_{k,1:i-1}\mathcal{R}_{1:i-1,i}$  follows, so that

$$q_{k+1,i} = \frac{(\mathbf{N}_{k,k} - \mathcal{R}_{i,i})}{\mathbf{N}_{k,k+1}}q_{k,i} + \frac{\mathbf{N}_{k,1:k-1}}{\mathbf{N}_{k,k+1}}q_{1:k-1,i} + \frac{\mathcal{R}_{1:i-1,i}}{\mathbf{N}_{k,k+1}}q_{1:k-1,i-1}.$$

Taking once again absolute values we finally obtain

$$|q_{k+1,i}| \leq \frac{|\mathbf{N}_{k,k} - \mathcal{R}_{i,i}|}{|\mathbf{N}_{k,k+1}|} |q_{k,i}| + \frac{\|\mathbf{N}_{k,1:k-1}\|}{|\mathbf{N}_{k,k+1}|} + \frac{\|\mathcal{R}_{1:i-1,i}\|}{|\mathbf{N}_{k,k+1}|},$$

so that  $|q_{k+1,i}| = \mathcal{O}(\xi\eta^{-1}|q_{k,i}|)$ .  $\square$

If some of the eigenvalues of  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  are significantly far from the field of values of  $\widehat{\mathbf{H}}_d$ , the first entry of the corresponding Schur vectors will be far below machine precision, orders of magnitude smaller, especially for  $d \gg 10$ . Therefore, to simplify the derivation, in the following proposition we will assume that these Schur vector entries are in fact zero. Such sparsity property allows us to derive the following result on the rank structure of the solution to the Sylvester equation.

**Proposition 4.4.** *Let  $\widehat{\mathbf{H}}_d^\top + e_d\widehat{h}^\top = \mathcal{Q}\mathcal{R}\mathcal{Q}^*$  and  $\widehat{\mathbf{G}}_d^\top + e_d\widehat{g}^\top = \mathcal{P}\mathcal{K}\mathcal{P}^*$  be Schur decompositions of the given matrices, with  $\mathcal{Q} = [q_1, \dots, q_d]$  and  $\mathcal{P} = [p_1, \dots, p_d]$ . Assume that the spectra of  $\widehat{\mathbf{H}}_d^\top + e_d\widehat{h}^\top$  and  $-(\widehat{\mathbf{G}}_d^\top + e_d\widehat{g}^\top)$  do not intersect. Under the hypotheses of [Lemma 4.3](#) in the limit for  $\eta = 0$ , define the partitions  $\mathcal{Q} = [\mathcal{Q}_0, \mathcal{Q}_1]$  with  $e_1^\top \mathcal{Q}_0 = 0$  and  $e_1^\top \mathcal{Q}_1 \neq 0$ , and  $\mathcal{P} = [\mathcal{P}_0, \mathcal{P}_1]$  with  $e_1^\top \mathcal{P}_0 = 0$  and  $e_1^\top \mathcal{P}_1 \neq 0$ . Then the solution  $\mathbf{Y}_d$  to*

$$(4.1) \quad (\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathbf{Y}_d + \mathbf{Y}_d(\widehat{\mathbf{G}}_d + \widehat{g}e_d^\top)^* = e_1\beta_1\beta_2e_1^*$$

where  $\beta_1, \beta_2 > 0$  are arbitrary, is given by  $\mathbf{Y}_d = \mathcal{Q}_1\mathcal{Z}\mathcal{P}_1^*$  where  $\mathcal{Z}$  solves

$$(4.2) \quad \mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1\mathcal{Z} + \mathcal{Z}\mathcal{P}_1^*(\widehat{\mathbf{G}}_d + \widehat{g}e_d^\top)^*\mathcal{P}_1 = \mathcal{Q}_1^*e_1\beta_1\beta_2e_1^*\mathcal{P}_1.$$

*Proof.* Spectral separation ensures existence and uniqueness of  $\mathbf{Y}_d$ . Left and right multiplication of (4.1) by  $\mathcal{Q}^*$  and  $\mathcal{P}$ , respectively, gives

$$(4.3) \quad \mathcal{Q}^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}\mathcal{Q}^*\mathbf{Y}_d\mathcal{P} + \mathcal{Q}^*\mathbf{Y}_d\mathcal{Q}\mathcal{Q}^*(\widehat{\mathbf{G}}_d + \widehat{g}e_d^\top)^*\mathcal{P} = \mathcal{Q}^*e_1\beta_1\beta_2e_1^*\mathcal{P}.$$

Using the hypotheses,  $\mathcal{Q}^*e_1\beta_1 = [0; q]$  with  $q = \mathcal{Q}_1^*e_1\beta_1$ , and  $\mathcal{P}^*e_1\beta_2 = [0; p]$  with  $p = \mathcal{P}_1^*e_1\beta_2$ . Let us use the partitions

$$\begin{aligned} \mathcal{Q}^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q} &=: \begin{bmatrix} \mathcal{R}_{11}^* & 0 \\ \mathcal{R}_{12}^* & \mathcal{R}_{22}^* \end{bmatrix}, & \mathcal{P}^*(\widehat{\mathbf{G}}_d + \widehat{g}e_d^\top)\mathcal{P} &=: \begin{bmatrix} \mathcal{K}_{11}^* & 0 \\ \mathcal{K}_{12}^* & \mathcal{K}_{22}^* \end{bmatrix}, \\ \mathcal{Q}^*\mathbf{Y}_d\mathcal{P} &=: \begin{bmatrix} \mathcal{Z}_{11} & \mathcal{Z}_{12} \\ \mathcal{Z}_{21} & \mathcal{Z}_{22} \end{bmatrix}, & \mathcal{Q}^*e_1\beta_1\mathcal{P} &=: \begin{bmatrix} 0 & 0 \\ 0 & qp^* \end{bmatrix}. \end{aligned}$$

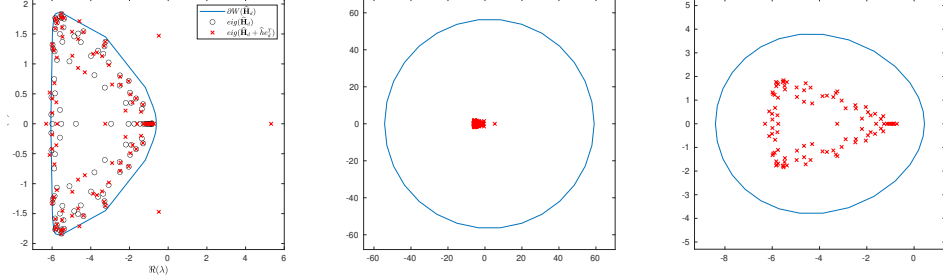


FIGURE 2. **Example 4.5.** Left:  $\partial W(\widehat{\mathbf{H}}_d)$  (blue solid line), spectra of  $\widehat{\mathbf{H}}_d$  (black circles), and  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  (red crosses). Center:  $\partial W(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)$  (blue solid line) and spectrum of  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  (red crosses). Right:  $\partial W(\mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1)$  (blue solid line) and spectrum of  $\mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1$  (red crosses).

Then equating all blocks in (4.3) we obtain

$$(4.4) \quad \mathcal{R}_{11}^* \mathcal{Z}_{11} + \mathcal{Z}_{11} \mathcal{K}_{11} = 0, \quad \mathcal{R}_{12}^* \mathcal{Z}_{11} + \mathcal{R}_{22}^* \mathcal{Z}_{21} + \mathcal{Z}_{21} \mathcal{K}_{11} = 0,$$

and

$$(4.5) \quad \mathcal{R}_{11}^* \mathcal{Z}_{12} + \mathcal{Z}_{11} \mathcal{K}_{12} + \mathcal{Z}_{12} \mathcal{K}_{22} = 0, \quad \mathcal{R}_{12}^* \mathcal{Z}_{12} + \mathcal{R}_{22}^* \mathcal{Z}_{22} + \mathcal{Z}_{21} \mathcal{K}_{12} + \mathcal{Z}_{22} \mathcal{K}_{22} = qp^*.$$

From the left matrix equation in (4.4) we obtain  $\mathcal{Z}_{11} = 0$ , since the spectra of  $\mathcal{R}_{11}$  and  $-\mathcal{K}_{11}$  do not intersect. This in turn leads to  $\mathcal{Z}_{21} = 0$  from the right equation, since  $\mathcal{K}_{11}$  and  $-\mathcal{R}_{22}$  have distinct eigenvalues. Similarly,  $\mathcal{Z}_{12} = 0$  can be obtained from the left matrix equation in (4.5). The right matrix equation in (4.5) gives the sought after solution  $\mathcal{Z} = \mathcal{Z}_{22}$ .  $\square$

In light of the previous result, for  $\mathcal{Q}_1 \in \mathbb{C}^{d \times \ell}$ , we call the *effective field of values* of  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  the set

$$W_{\text{eff}}(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top) = W(\mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1) = \left\{ \frac{v^* \mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1 v}{v^* v}, 0 \neq v \in \mathbb{C}^\ell \right\}.$$

The difference between the field of values and its effective counterpart depends on how large the distance defined in Lemma 4.3 is, and it may be significant if for instance the rank-one modification remarkably changes the eigenvalues. The following example illustrates the difference between the field of values and the effective field of values.

**Example 4.5.** For this example we take inspiration from [36, Example 5.6], and we set  $\mathbf{B} = \mathbf{A}^\top$ . Moreover, we set  $d = 100$  and construct the matrix  $\widehat{\mathbf{H}}_d$  and the vector  $\widehat{h}$  as follows

```
rng(21*pi); d=100;
Hhat=toeplitz([-4, 2, zeros(1,d-2)], [-4, 1/2, 1/2, randn(1,d-3)/20]);
hhat=flipud(linspace(1,20,d))' .* randn(d,1);
```



In **Figure 2** (left) we report  $\partial W(\widehat{\mathbf{H}}_d)$  (blue solid line) along with the spectra of  $\widehat{\mathbf{H}}_d$  (black circles) and  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  (red crosses). This figure shows that, although  $\widehat{\mathbf{H}}_d$  is negative definite, the matrix  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  has eigenvalues with positive real part. In particular, the set  $W(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)$  reported in **Figure 2** (center) significantly trespasses the imaginary axis. This may jeopardize the well-posedness of the projected equation (4.1) and/or the positive semidefiniteness of its solution  $\mathbf{Y}_d$ . This situation is mitigated by looking at the effective field of values of  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$ , namely  $W(\mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1)$ ; see **Figure 2** (right). Indeed,  $W(\mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1)$  is much smaller than  $W(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)$  and all the eigenvalues of  $\mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1$  have negative real part. This means that the matrix  $\mathbf{Y}_d$  can be constructed by first computing the solution  $\mathcal{Z}$  to (4.2) and then setting  $\mathbf{Y}_d = \mathcal{Q}_1\mathcal{Z}\mathcal{Q}_1^*$ . This guarantees both the existence and the positive semidefiniteness of  $\mathbf{Y}_d$ .

**4.2. Convergence analysis for the Lyapunov setting.** For the sake of simplifying the derived bounds, in this section we focus on the Lyapunov equation

$$(4.6) \quad \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^\top = cc^\top,$$

where, for convenience, we assume that the right-hand side has rank 1 and that  $\|c\| = 1$ .

We next show that if  $\mathbf{A}$  and the reduced matrix  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  are both negative definite, a convergence analysis is possible in a very similar way to the standard, non-sketched setting. We closely follow the techniques from [45].

**Theorem 4.6.** *Let  $\mathbf{A}$  be negative definite and let  $\mathbf{X}_d^{\text{SK}}$  denote the sketched-and-whitened Arnoldi approximation for the solution of (4.6). Assume that  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  is negative definite as well and denote  $\alpha_{\max} := \max\{\alpha(\mathbf{A}), \alpha(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\}$ .*

*Further, let  $\mathbb{E} \subset \mathbb{C}^-$  be a compact, convex set containing the field of values of both  $\mathbf{A}$  and  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$ , let  $\Phi_k, k = 0, 1, \dots$  with  $\deg(\Phi_k) = k$  denote the Faber polynomials with respect to  $\mathbb{E}$  and let*

$$(4.7) \quad \exp(\tau z) = \sum_{k=0}^{\infty} f_k(\tau)\Phi_k(z),$$

*be the Faber series of the exponential. Then, with high probability,*

$$(4.8) \quad \|\mathbf{X} - \mathbf{X}_d^{\text{SK}}\| \leq 2\eta_\varepsilon \sum_{k=d}^{\infty} \int_0^{\infty} \exp(\tau\alpha_{\max})|f_k(\tau)|d\tau,$$

*with  $\eta_\varepsilon = \left(1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\right) \left(1 + \frac{1}{\sqrt{1-\varepsilon}}\right)$  and  $\varepsilon$  the parameter controlling the embedding quality; see (2.2) or (2.3).*

*Proof.* We have  $\mathbf{X}_d^{\text{SK}} = \widehat{\mathbf{U}}_d\mathbf{Y}_d^{\text{SK}}\widehat{\mathbf{U}}_d^\top$  with

$$(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathbf{Y}_d^{\text{SK}} + \mathbf{Y}_d^{\text{SK}}(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)^\top = e_1\beta^2e_1^\top,$$

where  $\beta = \|\mathbf{S}_U c\|$ . Therefore, using the standard integral representation for the solution of the Lyapunov equation,

$$\begin{aligned}
\mathbf{X}_d^{\text{SK}} &= \widehat{\mathbf{U}}_d \mathbf{Y}_d^{\text{SK}} \widehat{\mathbf{U}}_d^\top \\
&= \int_0^\infty \widehat{\mathbf{U}}_d \exp(\tau(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)) e_1 \beta^2 e_1^\top \exp(\tau(\widehat{\mathbf{H}}_d + \widehat{t}e_d^\top)^\top) \widehat{\mathbf{U}}_d^\top d\tau, \\
(4.9) \quad &= \int_0^\infty x^{\text{SK}}(\tau) (x^{\text{SK}}(\tau))^\top d\tau,
\end{aligned}$$

with the short-hand notation  $x^{\text{SK}}(\tau) := \widehat{\mathbf{U}}_d \exp(\tau(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)) e_1 \beta$  for the sketched Krylov approximation of  $x(\tau) := \exp(\tau \mathbf{A}) c$ .

Subtracting (4.9) from the integral representation  $\mathbf{X} = \int_0^\infty x(\tau) (x(\tau))^\top d\tau$  gives

$$\begin{aligned}
\|\mathbf{X} - \mathbf{X}_d^{\text{SK}}\| &\leq \int_0^\infty \|x(\tau) (x(\tau))^\top - x_d^{\text{SK}}(\tau) (x_d^{\text{SK}}(\tau))^\top\| \\
&\leq \int_0^\infty (\|x(\tau)\| + \|x_d^{\text{SK}}(\tau)\|) \|x(\tau) - x_d^{\text{SK}}(\tau)\| d\tau \\
(4.10) \quad &\leq (1 + \|\widehat{\mathbf{U}}_d\|) \int_0^\infty \exp(\tau \alpha_{\max}) \|x(\tau) - x_d^{\text{SK}}(\tau)\| d\tau,
\end{aligned}$$

where for the last inequality we have used  $\|x(\tau)\| \leq \exp(\tau \alpha_{\max})$  and  $\|x_d^{\text{SK}}(\tau)\| \leq \exp(\tau \alpha_{\max}) \|\widehat{\mathbf{U}}_d\|$ . To bound the norm inside the integral in (4.10), we use the Faber series (4.7) together with the polynomial exactness property

$$(4.11) \quad \widehat{\mathbf{U}}_d p_{d-1}(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top) e_1 = p_{d-1}(\mathbf{A}) c,$$

for any polynomial of degree at most  $d-1$ , which gives

$$(4.12) \quad x(\tau) - x_d^{\text{SK}}(\tau) = \sum_{k=d}^\infty f_k(\tau) (\Phi_k(\mathbf{A}) c - \widehat{\mathbf{U}}_d \Phi_k(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top) e_1 \beta).$$

Taking norms in (4.12) yields

$$(4.13) \quad \|x(\tau) - x_d^{\text{SK}}(\tau)\| \leq \sum_{k=d}^\infty |f_k(\tau)| (\|\Phi_k(\mathbf{A})\| + \beta \|\widehat{\mathbf{U}}_d\| \|\Phi_k(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\|).$$

By [9, Théorème 1.1], we have  $\|\Phi_k(\mathbf{A})\| \leq 2$ ,  $\|\Phi_k(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\| \leq 2$ , by [5, Corollary 2.2] it holds  $\|\widehat{\mathbf{U}}_d\| \leq \frac{1}{\sqrt{1-\varepsilon}}$  and further  $\beta = \|\mathbf{S}_U c\| \leq \sqrt{1+\varepsilon} \|c\| = \sqrt{1+\varepsilon}$ , because  $\mathbf{S}_U$  is an  $\varepsilon$ -subspace embedding for  $\mathcal{K}_{d+1}(\mathbf{A}, c)$ . Inserting these relations into (4.13) gives

$$\|x(\tau) - x_d^{\text{SK}}(\tau)\| \leq 2 \left(1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\right) \sum_{k=d}^\infty |f_k(\tau)|.$$

Further inserting this into (4.10) and using the bound  $\|\widehat{\mathbf{U}}_d\| \leq \frac{1}{\sqrt{1-\varepsilon}}$  once again completes the proof.  $\square$

Note that for the usually assumed sketching quality  $\varepsilon = 1/\sqrt{2}$ , we have

$$2\eta_\varepsilon \approx 19.45.$$

The result of [Theorem 4.6](#) is very similar to standard results for full Arnoldi in the (non-sketched) setting; see [45]. In that case, a bound like (4.8) holds, where the factor  $2\eta_\varepsilon$  is replaced by 8, one can use  $\exp(\tau \alpha(\mathbf{A}))$  instead of  $\exp(\tau \alpha_{\max})$ ,

and  $\mathbb{E}$  only needs to contain  $W(\mathbf{A})$ , as it then also contains  $W(\mathfrak{H}_d)$ . This shows that as long as  $W(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)$  is not much larger than  $W(\mathbf{A})$ , one can expect the sketched method to converge similarly to the full Arnoldi method. Let us also note that it is well-known that bounds of the form (4.8) are often not tight in practice, also in the deterministic case. In particular, while the convergence slope might often be predicted quite well, the constants involved are often a large overestimate. These bounds are therefore not well-suited to be used as a stopping criterion for the method.

We would like to remark that the polynomial exactness property (4.11) used in the proof of [Theorem 4.6](#) is readily available thanks to the WS-Arnoldi relations (2.8). Indeed, in [19, Theorem 2.7] it has been shown that the only modifications to the upper Hessenberg matrix stemming from the Arnoldi process ( $\widehat{\mathbf{H}}_d$  in our case) that preserve the polynomial exactness are indeed low-rank modifications acting only on the last (block) column. Once again, this result would not be available if we had to rely on expressions of the projected matrices involving pseudoinverses.

For specific shapes of the set  $\mathbb{E}$ , the bound in [Theorem 4.6](#) can be made more explicit. The following corollary exemplifies this for the case that  $\mathbb{E}$  is an ellipse. It depends a lot on the specific  $\mathbf{A}$  at hand whether assuming an enclosing ellipse is a reasonable assumption or not, and we merely show this as an example of the type of bound that can be obtained. For some other cases, e.g., a wedge-shaped set, similar bounds are possible.

**Corollary 4.7.** *Let the assumptions of [Theorem 4.6](#) hold and further assume that  $\mathbb{E}$  is an ellipse in  $\mathbb{C}^-$  with center  $(-c, 0)$ , foci  $(-c, \pm\delta)$  and semi-axes  $a_1$  and  $a_2$ , so that  $\delta = \sqrt{a_1^2 - a_2^2}$ . Define  $\rho_1 = \frac{a_1 + a_2}{2}$  and  $\rho_2 = \frac{c - \alpha_{\max}}{2\rho_1} + \frac{1}{2\rho_1} \sqrt{(c - \alpha_{\max})^2 - \delta^2}$ . Then*

$$\|\mathbf{X} - \mathbf{X}_d^{\text{SK}}\| \leq 2\eta_\varepsilon \frac{1}{\sqrt{(c - \alpha_{\max})^2 - \delta^2}} \frac{\rho_2}{\rho_2 - 1} \left(\frac{1}{\rho_2}\right)^d.$$

*Proof.* The proof works along the same lines as that of [45, Proposition 4.1].  $\square$

The results above may appear to be of rather limited use, since  $W(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)$  may significantly differ from  $W(\mathbf{A})$ . However, the results that led us to introduce the effective field of values of  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$  enable us to obtain more informative convergence bounds. Indeed, under the hypotheses of [Proposition 4.4](#), we have

$$\begin{aligned} \mathbf{X}_d^{\text{SK}} &= (\widehat{\mathbf{U}}_d \mathcal{Q}_1) \mathcal{Z} (\widehat{\mathbf{U}}_d \mathcal{Q}_1)^* \\ &= \int_0^\infty \widehat{\mathbf{U}}_d \mathcal{Q}_1 \exp(\tau(\mathcal{Q}_1^* (\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top) \mathcal{Q}_1) \mathcal{Q}_1^* e_1 \beta^2 e_1^\top \mathcal{Q}_1 \exp(\tau(\mathcal{Q}_1^* (\widehat{\mathbf{H}}_d + \widehat{t}e_d^\top) \mathcal{Q}_1)^*) \mathcal{Q}_1^* \widehat{\mathbf{U}}_d^\top) d\tau \\ &= \int_0^\infty x^{\text{SK}}(\tau) (x^{\text{SK}}(\tau))^\top d\tau, \end{aligned}$$

where

$$x^{\text{SK}}(\tau) = \widehat{\mathbf{U}}_d \exp(\tau(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)) e_1 \beta = \widehat{\mathbf{U}}_d \mathcal{Q}_1 \exp(\tau(\mathcal{Q}_1^* (\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top) \mathcal{Q}_1)) \mathcal{Q}_1^* e_1 \beta,$$

and the last equality follows from noticing that under the proposition hypotheses,  $\mathcal{Q}^* e_1 = [0; \mathcal{Q}_1^* e_1]$ . In particular, in the proof of [Theorem 4.6](#), we have that  $\|x^{\text{SK}}(\tau)\| \leq \exp(\tau\alpha'_{\max})$ , where  $\alpha'_{\max}$  is the rightmost point of  $W_{\text{eff}}(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)$ .

Moreover,

$$\|x(\tau) - x_d^{\text{SK}}(\tau)\| \leq \sum_{k=d}^{\infty} |f_k(\tau)| (\|\Phi_k(\mathbf{A})\| + \beta \|\widehat{\mathbf{U}}_d\| \|\Phi_k(\mathcal{Q}_1^*(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)\mathcal{Q}_1)\|).$$

It thus follows that we can obtain a result such as [Theorem 4.6](#) based on the *effective* field of values of  $\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top$ , which can be significantly closer to that of  $\widehat{\mathbf{H}}_d$  than  $W(\widehat{\mathbf{H}}_d + \widehat{h}e_d^\top)$ .

Corresponding convergence results can be obtained for the Sylvester equation, taking into account spectral properties of both left and right coefficient matrices, with only technical modifications. To avoid proliferation of similar results, we omit this derivation.

## 5. THE ALGORITHMS

We report the complete algorithms presented in the previous sections for solving the Sylvester equation. In [Algorithm 1](#) we depict the sketched-and-truncated approach (with basis whitening) for Sylvester equations. We label this algorithm s-PSKM. The truncated Arnoldi method for (1.1) is illustrated in [Algorithm 2](#) in [Appendix A](#) and is labeled T-PSKM.

Along with the definition of the projected problem and the residual norm expression, the main difference between T-PSKM and s-PSKM lies in the update of the QR factorizations of the sketched bases in [line 9](#) of [Algorithm 1](#). Similarly, the new block upper Hessenberg matrix  $\widehat{\mathbf{H}}_{d+1} = \mathbf{T}_{U,d+1} \mathbf{H}_{d+1} \mathbf{T}_{U,d+1}^{-1}$  is not recomputed from scratch but rather updated: if  $\mathbf{T}_{U,d+1}$  is as in [Proposition 2.1](#), then a direct computation shows that  $\mathbf{T}_{U,d+1}^{-1} = \begin{bmatrix} \mathbf{T}_{U,d}^{-1} & -\mathbf{T}_{U,d}^{-1} \mathbf{T}_H \boldsymbol{\tau}_{d+1}^{-1} \\ & \boldsymbol{\tau}_{d+1}^{-1} \end{bmatrix}$  so that

$$\widehat{\mathbf{H}}_{d+1} = \begin{bmatrix} \widehat{\mathbf{H}}_d + \mathbf{T}_H \mathbf{h}_{d+1,d} \boldsymbol{\tau}_d^{-1} E_d^\top & \widehat{H}_{\text{new}} \\ \boldsymbol{\tau}_{d+1} \mathbf{h}_{d+1,d} \boldsymbol{\tau}_d^{-1} E_d^\top & \boldsymbol{\tau}_{d+1} (-\mathbf{h}_{d+1,d} \boldsymbol{\tau}_d^{-1} E_d^\top \mathbf{T}_H + \mathbf{h}_{d+1,d+1}) \boldsymbol{\tau}_{d+1}^{-1} \end{bmatrix},$$

where  $\widehat{H}_{\text{new}} = -(\widehat{\mathbf{H}}_d + \mathbf{T}_H \mathbf{h}_{d+1,d} \boldsymbol{\tau}_d^{-1} E_d^\top) \mathbf{T}_H \boldsymbol{\tau}_{d+1}^{-1} + \mathbf{T}_{U,d} H \boldsymbol{\tau}_{d+1}^{-1} + \mathbf{T}_H \mathbf{h}_{d+1,d+1} \boldsymbol{\tau}_d^{-1}$  and  $H = [\mathbf{h}_{1,d+1}^\top, \dots, \mathbf{h}_{d,d+1}^\top]^\top \in \mathbb{R}^{dr \times r}$ . A corresponding update for  $\widehat{\mathbf{G}}_{d+1}$  is performed.

At convergence, the matrix  $\mathbf{T}_{U,d}^{-1} \mathbf{Y}_d (\mathbf{T}_{V,d}^{-1})^\top$  is approximated by a low-rank factorization,  $\mathbf{T}_{U,d}^{-1} \mathbf{Y}_d (\mathbf{T}_{V,d}^{-1})^\top \approx \mathbf{Z}^{(1)} (\mathbf{Z}^{(2)})^\top$ , with  $\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)} \in \mathbb{R}^{dr \times \ell}$ ,  $\ell \leq dr$  so that the final approximate solution  $\mathbf{X}_d$  is available as the low-rank product  $\mathbf{X}_d \approx \mathbf{X}^{(1)} (\mathbf{X}^{(2)})^\top = (\widehat{\mathbf{U}}_d \mathbf{Z}^{(1)}) (\widehat{\mathbf{V}}_d \mathbf{Z}^{(2)})^\top$ ; cf. [lines 19 to 20](#) of [Algorithm 1](#). Since  $\widehat{\mathbf{U}}_d, \widehat{\mathbf{V}}_d$  are not stored, to finalize the computation a two-pass strategy is employed; see, e.g. [\[20, 30, 38\]](#). More precisely, the original Arnoldi iteration is performed once again to iteratively construct the two matrices  $\widehat{\mathbf{U}}_d \mathbf{Z}^{(1)}, \widehat{\mathbf{V}}_d \mathbf{Z}^{(2)}$ . Since the orthogonalization coefficients are available in the matrices  $\mathbf{H}_d, \mathbf{G}_d$ , the cost of this second Arnoldi iteration does not involve inner products for the local orthogonalization. To improve efficiency, the reconstruction of  $\widehat{\mathbf{U}}_d \mathbf{Z}^{(1)}$  and  $\widehat{\mathbf{V}}_d \mathbf{Z}^{(2)}$  should be performed in chunks of  $kr$  basis vectors at a time.

[Table 1](#) summarizes the main memory requirements of full Arnoldi and the proposed methods. Both s-PSKM and T-PSKM require the same amount of long vector allocations for fixed truncation parameter  $k$  and number of iterations and we thus report only the storage demand of the former. For  $k, \ell \ll d$  the main difference between the two classes of methods stands in the memory allocations for

---

**Algorithm 1** Sketched-and-truncated Arnoldi method for Sylvester equations (S-PSKM)

---

**Input:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{C}_1 \in \mathbb{R}^{n \times r}$ ,  $\mathbf{C}_2 \in \mathbb{R}^{n \times r}$ ,  $\mathbf{S}_U \in \mathbb{R}^{s \times n}$ ,  $\mathbf{S}_V \in \mathbb{R}^{s \times n_2}$ , integers  $0 < k \leq \maxit \ll n$ ,  $\text{tol} > 0$ ,  $p \geq 1$

**Output:**  $\mathbf{X}^{(1)}$ ,  $\mathbf{X}^{(2)}$  such that  $\mathbf{X}^{(1)}(\mathbf{X}^{(2)})^\top = \mathbf{X}_d$  approximately solves  $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{C}_1\mathbf{C}_2^\top$

- 1: Compute skinny QRs:  $U_1\boldsymbol{\ell} = \mathbf{C}_1$ ,  $V_1\mathbf{s} = \mathbf{C}_2$ ,  $\mathbf{Q}_{U,1}\boldsymbol{\beta}_1 = \mathbf{S}_U\mathbf{C}_1$ ,  $\mathbf{Q}_{V,1}\boldsymbol{\beta}_2 = \mathbf{S}_V\mathbf{C}_2$ , and set  $\mathbf{T}_{U,1} = \boldsymbol{\beta}_1$ ,  $\mathbf{T}_{V,1} = \boldsymbol{\beta}_2$
- 2: **for**  $d = 1, \dots, \maxit$  **do**
- 3:   Compute  $\tilde{\mathbf{U}} = \mathbf{A}U_d$ ,  $\tilde{\mathbf{V}} = \mathbf{B}^\top V_d$
- 4:   **for**  $i = \max\{1, d - k + 1\}, \dots, d$  **do**
- 5:     Set  $\tilde{\mathbf{U}} = \tilde{\mathbf{U}} - U_i\mathbf{h}_{i,d}$  with  $\mathbf{h}_{i,d} = U_i^\top \tilde{\mathbf{U}}$
- 6:     Set  $\tilde{\mathbf{V}} = \tilde{\mathbf{V}} - V_i\mathbf{g}_{i,d}$  with  $\mathbf{g}_{i,d} = V_i^\top \tilde{\mathbf{V}}$
- 7:   **end for**
- 8:   Compute skinny QRs:  $U_{d+1}\mathbf{h}_{d+1,d} = \tilde{\mathbf{U}}$  and  $V_{d+1}\mathbf{g}_{d+1,d} = \tilde{\mathbf{V}}$
- 9:   Update QRs:  $\mathbf{Q}_{U,d+1}\mathbf{T}_{U,d+1} = \mathbf{S}_U[U_d, U_{d+1}]$ ,  
 $\mathbf{Q}_{V,d+1}\mathbf{T}_{V,d+1} = \mathbf{S}_V[V_d, V_{d+1}]$
- 10:   Update  $\widehat{\mathbf{H}}_d = \mathbf{T}_{U,d}\mathbf{H}_d\mathbf{T}_{U,d}^{-1}$ ,  $\widehat{\mathbf{H}} = T_H\mathbf{h}_{d+1,d}\boldsymbol{\tau}_d^{-1}$ ,  
 $\widehat{\mathbf{G}}_d = \mathbf{T}_{V,d}\mathbf{G}_d\mathbf{T}_{V,d}^{-1}$ ,  $\widehat{\mathbf{G}} = T_G\mathbf{g}_{d+1,d}\boldsymbol{\theta}_d^{-1}$
- 11:   **if**  $\text{mod}(d, p) = 0$  **then**
- 12:     Solve  $(\widehat{\mathbf{H}}_d + \widehat{\mathbf{H}}E_d^\top)\mathbf{Y} + \mathbf{Y}(\widehat{\mathbf{G}}_d + \widehat{\mathbf{G}}E_d^\top)^\top = E_1\boldsymbol{\beta}_1\boldsymbol{\beta}_2^\top E_1^\top$  for  $\mathbf{Y}$
- 13:     Compute  $\rho = \sqrt{\|\mathbf{h}_{d+1,d}E_d^\top\mathbf{Y}\|_F^2 + \|\mathbf{Y}E_d\mathbf{g}_{d+1,d}^\top\|_F^2}$
- 14:     **if**  $\rho < \text{tol}$  **then**
- 15:       Go to line 19
- 16:     **end if**
- 17:   **end if**
- 18: **end for**
- 19: Compute (possibly low-rank) factors  $\mathbf{Y}_1, \mathbf{Y}_2$  such that  $\mathbf{Y} \approx \mathbf{Y}_1\mathbf{Y}_2^\top$
- 20: Retrieve  $\mathbf{X}^{(1)} = U_d\mathbf{T}_{U,d}^{-1}\mathbf{Y}_1$ ,  $\mathbf{X}^{(2)} = V_d\mathbf{T}_{V,d}^{-1}\mathbf{Y}_2$  by the two-pass step

---

the generated bases. On the other hand, storage for the solution factors depends on the spectral decay properties of the solution to the reduced problem. Since the polynomial Krylov space generates redundant information, the numerical rank  $\ell$  of  $\mathbf{Y}_d$  may be significantly lower than its dimension<sup>6</sup>, allowing major savings in storing the approximate solution in factored form. We refer to the examples of [Section 6](#) for experimental evidence. The extra allocations concern matrices in the reduced spaces, together with the  $s \times dr$  sketched basis<sup>7</sup>, which is kept orthonormal.

<sup>6</sup>As already mentioned, we are assuming that the exact solution can be well approximated by a matrix of very low rank.

<sup>7</sup>This bit is not present in the overall storage demand of T-PSKM.

Method	Memory Allocation Bases	Memory Allocation Approximate Solution	Extra Terms
Full Arnoldi	$\mathbf{u}_d, \mathbf{v}_d$ $2(d+1)r \cdot n$	$(\mathbf{X}_d^{\text{FULL}})^{(1)}, (\mathbf{X}_d^{\text{FULL}})^{(2)}$ $2\ell \cdot n$	$\mathcal{O}(d^2r^2)$
s-PSKM	$\widehat{\mathbf{U}}_d, \widehat{\mathbf{V}}_d$ $2(k+1)r \cdot n$	$\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$ $2\ell \cdot n$	$\mathcal{O}(d^2r^2 + drs)$
	Cost Bases Construction	Cost Solution Projected Problem	Cost Two-Pass
Full Arnoldi	$\mathcal{O}(\theta + nr^2d^2)$	$\mathcal{O}(r^3d^3)$	-
s-PSKM	$\mathcal{O}(\theta + nr^2dk + sr^2d^2)$	$\mathcal{O}(r^3d^3)$	$\mathcal{O}(\theta + nr^2dk)$

TABLE 1. Memory requirements and main computational costs (in flops) of full Arnoldi and s-PSKM.  $\theta = \mathcal{O}((\text{nnz}(A) + \text{nnz}(B))rd)$ .

In [Table 1](#) we also list the main computational costs of full Arnoldi and s-PSKM in terms of flops. These amount to the bases construction and the solution of the projected problems. s-PSKM also sees the extra cost coming from the second Arnoldi pass. The main difference between the two procedures lies in the orthogonalization cost. The full orthogonalization of Arnoldi requires  $\mathcal{O}(nr^2d^2)$  flops whereas our sketched and truncated procedure cuts the cost down to  $\mathcal{O}(nr^2dk + sr^2d^2)$  where the first term comes from the truncated Arnoldi step whereas the second one is related to the whitening of the sketched bases. The results in [Section 6](#) will show that these savings in the number of flops do indeed lead to less time-consuming solution procedures. Notice that t-PSKM and s-PSKM basically have the same computational cost per iteration. The only difference is in the lack of any sketch-related cost in the former.

Let us also mention that it is possible to use more stable means of orthogonalizing the bases in s-PSKM, e.g., a rank-revealing QR decomposition, as it was already suggested for whitening in [\[40\]](#) or a randomized Householder QR algorithm [\[23\]](#). This substantially increases the computational cost of the algorithm, though: Using a rank-revealing QR decomposition means that it is not possible to update the QR factors from one iteration to the next (cf. [line 9](#) of [Algorithm 1](#)). Instead, these factors have to be recomputed from scratch in every single iteration. Such an approach should therefore only be used for very challenging problems in which s-PSKM becomes unstable. We note that we did not observe this situation in our experiments and therefore did not pursue such an approach any further.

Finally, we remark that our stopping criterion is based on the residual norm or its estimation, in both algorithms. This criterion is checked periodically, not necessarily at each iteration, to save computational efforts.

## 6. NUMERICAL EXPERIMENTS

In this section we report numerical experiments to illustrate the potential of the new algorithms. In all the experiments presented here, we employ the sketching matrix  $\mathbf{S} = \sqrt{s/n} \cdot \mathbf{D}\mathbf{N}\mathbf{E} \in \mathbb{R}^{s \times n}$  where  $\mathbf{E} \in \mathbb{R}^{n \times n}$  is a diagonal matrix containing Rademacher random variables on its diagonal,  $\mathbf{N} \in \mathbb{R}^{n \times n}$  denotes the discrete cosine transform matrix, and  $\mathbf{D} \in \mathbb{R}^{s \times n}$  randomly selects  $s$  rows.<sup>8</sup>

<sup>8</sup>For reproducibility purposes, we set `rng('default')` in all our experiments, and the action of  $\mathbf{N}$  is computed by the Matlab function `dct`.

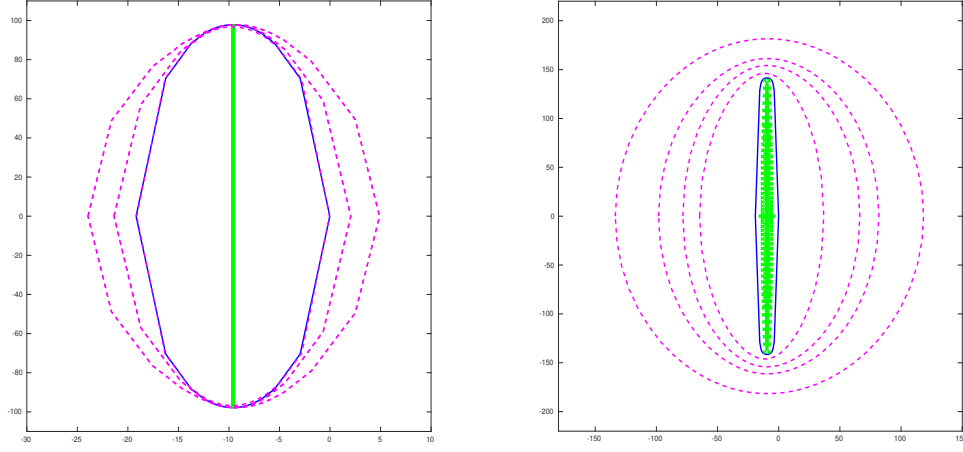


FIGURE 3. **Example 6.1.** Field of values of the coefficient matrix (solid line), corresponding eigenvalues (approximately vertically aligned crosses) and field of values of the reduced matrices, for  $d = 10, 20, 30, 40$  and  $r = 1$  (expanding dashed lines as  $d$  grows). Left:  $\mathbf{A}$  and  $\widehat{\mathbf{H}}_d + \widehat{\mathbf{H}}E_d^\top$ . Right:  $\mathbf{B}$  and  $\widehat{\mathbf{G}}_d + \widehat{\mathbf{G}}E_d^\top$ .

All the experiments have been run using Matlab (version 2022b) on a machine with a 1.2GHz Intel quad-core i7 processor with 16GB RAM on an Ubuntu 20.04.2 LTS operating system. All the small-dimensional Sylvester equations arising in the projection-based procedures we test are solved by the Matlab function `lyap` which implements the Bartels-Stewart algorithm [7].

The code for reproducing the results presented in the following is publicly available online at

<https://github.com/palittaUniB0/Sketched-and-truncated-Krylov4MatrixEqs>

**Example 6.1.** We compare the new sketched-and-truncated method with the full Arnoldi method. Moreover, we analyze the role of the truncation parameter in the performance of the new algorithm.

We consider the matrices  $\mathbf{A}$  and  $\mathbf{B}$  in (1.1) coming from the discretization of a convection-diffusion differential operator of the form

$$(6.1) \quad \mathcal{L}(u) = -\nu\Delta u + w\nabla u,$$

on the unit square  $[0, 1]^2$  by centered finite differences using  $\sqrt{n}$  nodes in each direction, yielding  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ . We consider the same viscosity parameter  $\nu$  in the construction of both  $\mathbf{A}$  and  $\mathbf{B}$  whereas for the convection vectors we use  $w_{\mathbf{A}} = (1, 1)$  for  $\mathbf{A}$  and  $w_{\mathbf{B}} = (3y(1-x^2), -2x(1-y^2))$  for  $\mathbf{B}$ . The block vectors  $C_1, C_2 \in \mathbb{R}^{n \times r}$  in (1.1) have random entries drawn from the standard normal distribution and are such that  $\|C_1 C_2^\top\|_F = 1$ .

In Figure 3 we report the boundary of the field of values of the two coefficient matrices (solid line, left plot for  $\mathbf{A}$  and right plot for  $\mathbf{B}$ ), together with those of the corresponding matrices  $\widehat{\mathbf{H}}_d + \widehat{\mathbf{H}}E_d^\top$  and  $\widehat{\mathbf{G}}_d + \widehat{\mathbf{G}}E_d^\top$ , for  $d = 10, 20, 30, 40$  and  $r = 1$  (dashed lines). To be able to approximate the field of values of  $\mathbf{A}$  and  $\mathbf{B}$  a coarse discretization of the two operators was considered, leading to  $n = 2500$ . In light of this, we also set `maxit`= 100 and `s` = 200. We can appreciate that in both

$\nu$	Full Arnoldi						s-PSKM					
	$p = 1$			$p = 20$			$p = 1$			$p = 20$		
	$d$	Time	Mem	$d$	Time	Mem	$d$	Time	Mem	$d$	Time	Mem
$r = 1$												
0.1	460	117.69	922	460	90.49	922	458	44.10	56	480	15.96	54
0.01	553	185.65	1108	560	133.50	1122	550	77.80	84	560	20.56	84
0.001	745	409.26	1492	760	255.58	1522	770	228.39	174	780	38.38	174
$r = 3$												
0.1	391	455.61	2352	400	304.81	2406	386	242.19	156	400	72.25	156
0.01	501	856.75	3012	520	534.57	3126	496	504.37	238	500	112.88	238
0.001	694	2474.43	4170	700	1052.56	4206	741	2006.41	516	760	276.30	518

TABLE 2. **Example 6.1.** Performance of the full Arnoldi method and s-PSKM as the viscosity  $\nu$  varies. The reported CPU timings are in seconds and  $d$  is the number of iterations at termination. “Mem” indicates the number of vectors of length  $n$  used. The stopping criterion is checked every  $p$  iterations.  $r$  is the right-hand side rank.

cases, and in particular for the matrix  $\mathbf{B}$ , the field of values of the reduced matrix is significantly larger than that of the corresponding coefficient matrix, and it is expanding with growing  $d$ . In principle, this behavior can potentially jeopardize the performance of s-PSKM. However, the following numerical results will show that s-PSKM favorably compares to full Arnoldi. This supports our theoretical findings which indeed suggest that the field of values of the projected matrices is not the right tool to predict the convergence of the solver in the sketched Krylov framework.

To analyze the computational performance of the proposed method, we consider  $n = 90\,000$  and start by comparing the full Arnoldi method and s-PSKM. The threshold for the stopping criterion in full Arnoldi and s-PSKM is  $\text{tol} = 10^{-6}$ . The truncation parameter in s-PSKM is set to  $k = 10$ , the sketching dimension is  $s = 1\,600$  and  $\text{maxit} = 800$ .

In both procedures, the cost of solving the projected equations becomes more and more dominant as the dimensions of the reduced matrices increase, being of the order of  $(rd)^3$ . To mitigate this shortcoming, it is common practice to solve the projected problem and check the stopping criterion every  $p \geq 1$  iterations. We consider  $p \in \{1, 20\}$ .

In Table 2 we report the number of iterations and CPU times of the full Arnoldi method and the sketched-and-truncated scheme, as  $\nu$  varies, and  $r = 1, 3$ . The two schemes perform a very similar number of iterations for all tested values of  $\nu$ . This shows that on this example the basis truncation does not significantly affect (delay) convergence and the implicit use of the  $\mathbf{S}^\top \mathbf{S}$ -orthogonal bases  $\hat{\mathbf{U}}_d$  and  $\hat{\mathbf{V}}_d$  pays off. We recall that in s-PSKM we check the  $\mathbf{S}^\top \mathbf{S}$ -norm of the residual matrix whereas full Arnoldi computes its Frobenius norm. Although rather close thanks to the  $\varepsilon$ -embedding property of the sketching, these two norms may slightly differ (cf. (3.6)), leading to a small gap in the number of iterations needed to achieve a prescribed level of accuracy in the two algorithms.



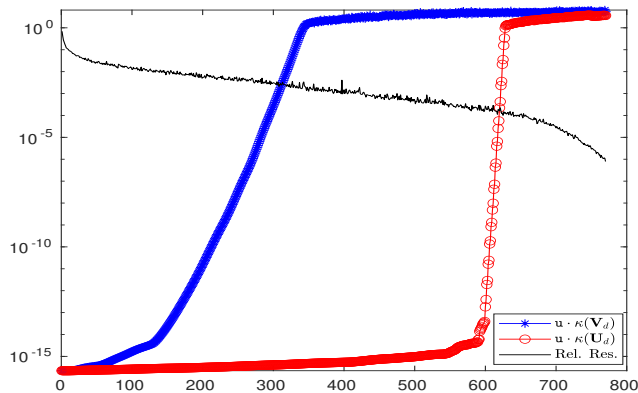


FIGURE 4. **Example 6.1.** Scaled condition number of the bases  $\mathbf{U}_d$  (red line with circles) and  $\mathbf{V}_d$  (blue line with stars), along with the relative Frobenius norm of the residual matrix  $\mathbf{R}_d$  (black solid line).

Thanks to an effective truncation strategy with  $k = 10$ , s-PSKM always requires less CPU time than full Arnoldi, for a similar number of iterations. Moreover, the rather small dimension of the sketching ( $s = 1600$ ), limits the impact of the full orthogonalization of the sketched bases on the overall solution process. We can also observe that using a larger value of  $p$  is much more convenient (up to 85% savings) for s-PSKM than for full Arnoldi (see [Table 2](#) for  $p = 20$ ). This is because in the latter, the cost of constructing a fully orthonormal basis still remains largely dominant.

Our primary concern is memory allocations. We stress that only about  $k = 10$  long vector allocations are required by the truncated approach for each space, as opposed to the full basis. For more on-point insight on the memory requirements, [Table 2](#) (column labeled “Mem”) includes the number of vectors of length  $n$  stored by the two algorithms. With an eye to [Table 1](#), for full Arnoldi memory allocations are mainly due to the bases, whereas for s-PSKM they are mainly due to storing the low-rank factors of the computed solution. The memory requirements of s-PSKM are always one order of magnitude smaller than what is needed by full Arnoldi, making the sketched-and-truncated Krylov subspace method very competitive.

We also point out that the likely ill-conditioning of the non-orthogonal bases  $\mathbf{U}_d$  and  $\mathbf{V}_d$  does not have a strong impact on the convergence of s-PSKM. In [Figure 4](#) we fix  $\nu = 0.001$ ,  $k = 10$ ,  $r = 1$ , and plot  $\mathbf{u} \cdot \kappa_2(\mathbf{U}_d)$  and  $\mathbf{u} \cdot \kappa_2(\mathbf{V}_d)$ , where  $\mathbf{u}$  denotes the unit roundoff, along with  $\|\mathbf{R}_d\|_F$ . We can observe that both  $\mathbf{U}_d$  and  $\mathbf{V}_d$  become numerically singular way before convergence is reached. On the other hand, this potential issue does not significantly affect the trend of  $\|\mathbf{R}_d\|_F$ . This was also noticed in [\[25\]](#).

We next turn our attention to the plain truncated scheme to analyze how different selections of the truncation parameters for the two spaces, namely  $k_A$  and  $k_B$ , may affect performance. In contrast to what happens for s-PSKM (cf. [Table 2](#) for  $k_B = 10$  and  $r = 1$ ), T-PSKM was unable to achieve the prescribed level of

$\nu$	Full Arnoldi						T-PSKM						
	$p = 1$			$p = 20$			$k_B$	$p = 1$			$p = 20$		
	$d$	Time	Mem	$d$	Time	Mem		$d$	Time	Mem	$d$	Time	Mem
$r = 1, k_A = 40$													
0.1	460	117.69	922	460	90.49	922	40	565	107.11	80	580	48.46	80
							60	601	146.25	100	620	75.77	100
0.01	553	185.65	1108	560	133.50	1122	40	671	168.58	98	680	63.01	98
							60	742	237.24	100	760	99.40	100
$r = 3, k_A = 100$													
0.1	391	455.61	2346	400	304.81	2400	100	1196	11013.21	600	1200	1534.59	600
							150	839	3033.78	750	840	930.22	750
0.01	501	856.75	3006	520	534.57	3120	100	1069	6841.95	600	1080	1203.43	600
							150	874	3529.69	750	880	1033.76	750

TABLE 3. **Example 6.1.** Performance of the full Arnoldi method and T-PSKM.  $k_A$  and  $k_B$  denote the truncation parameters employed in T-PSKM for the construction of the left and right space, respectively. The reported CPU timings are in seconds. “Mem” indicates the number of vectors of length  $n$  used.

accuracy for  $k_A = k_B = 10$  within maxit=1000 iterations for any considered value of  $\nu$ . We thus fix  $k_A = 40$  and in Table 3 we report the number of iterations and computational timings of T-PSKM when varying  $\nu$  and  $k_B$ . With these values of  $k_A$  and  $k_B$ , T-PSKM converges and considerations about the use of  $p \geq 1$  similar to the ones discussed for the sketched-and-truncated procedure still hold. Similarly for the memory requirements.

In Table 3 we also report the results achieved by setting  $r = 3$ . We notice that in this case larger values of  $k_A$  and  $k_B$  are needed to be able to meet the required level of accuracy in 2000 iterations. In particular, we used  $k_A = 100$  and  $k_B = 100, 150$ . In spite of these sizable truncation parameters, T-PSKM still shows a significant delay in the number of iterations when compared to full Arnoldi. This drawback makes T-PSKM less appealing in terms of computational time.

Due to the large values of  $k_A$  and  $k_B$ , also the memory requirements of T-PSKM increase. Indeed, in general, these no longer amount to the allocation of the low-rank factors of the approximate solution as in s-PSKM, but rather to the storage of the truncated bases, namely  $(k_A + k_B)r$  vectors of length  $n$ . Nevertheless, the storage demand of T-PSKM is still remarkably smaller than that of full Arnoldi for both  $r = 1, 3$ .

To conclude this example, we would like to highlight a somehow unexpected phenomenon: T-PSKM performs more iterations for  $k_B = 60$  than for  $k_B = 40$ , in spite of the former having more orthogonal basis vectors. This peculiar aspect shows that selecting sensible values of  $k$  for T-PSKM may be a rather difficult task. Further analysis is needed to fully understand this behavior.

**Example 6.2.** For this example we take inspiration from [26, Example 5.1]. We consider the heat equation  $u_t + \Delta u = f$  in the cube  $(-1, 1)^3$  and time interval  $[0, 1]$ , with homogeneous Dirichlet boundary conditions and zero initial value. The

source term is  $f(t, x, y, z) := 10t \sin(t) \cos(2\pi x) \cos(2\pi y) \cos(2\pi z)$  so that its space-time discretized form has rank equal to one. The discretization of this differential problem leads to the following Sylvester equation

$$(6.2) \quad \mathbf{K}\mathbf{X}\mathbf{B}_1 + \mathbf{M}\mathbf{X}\mathbf{B}_2 = c_1 c_2^\top,$$

where  $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{\ell \times \ell}$ , with  $\ell$  the number of time steps, are such that

$$\mathbf{B}_1 = \frac{1}{2(\ell-1)} \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & & -1 & 1 \end{bmatrix},$$

and correspond to the discrete Crank-Nicolson time integrator, whereas  $\mathbf{K}, \mathbf{M} \in \mathbb{R}^{n \times n}$  are, respectively, the stiffness and mass matrix coming from the discretization of the space component of the differential operator by linear finite elements; see [26] for more details.

We recast equation (6.2) in terms of a standard Sylvester equation by inverting  $\mathbf{M}$  and  $\mathbf{B}_1$ . In particular, we consider the Sylvester equation

$$(6.3) \quad \mathbf{M}^{-1}\mathbf{K}\mathbf{X} + \mathbf{X}\mathbf{B}_2\mathbf{B}_1^{-1} = \mathbf{M}^{-1}c_1 c_2^\top \mathbf{B}_1^{-1}.$$

Thanks to the moderate number of time steps  $\ell$  we consider ( $\ell = 800$ ) and the sparsity pattern of  $\mathbf{B}_1$ , we can explicitly construct<sup>9</sup> the matrix  $\mathbf{B}_2\mathbf{B}_1^{-1}$ . On the other hand, the mass matrix  $\mathbf{M}$  inherits the possibly involved sparsity pattern of  $\mathbf{K}$ . Moreover, we employ large values of degrees of freedom  $n$  in space so that  $\mathbf{M}^{-1}\mathbf{K}$  cannot be explicitly computed. The action of this matrix is computed on the fly during the construction of the Krylov subspace  $\mathcal{K}_d(\mathbf{M}^{-1}\mathbf{K}, \mathbf{M}^{-1}c_1)$ . In particular, quantities of the form  $\mathbf{M}^{-1}v$  are approximated by applying the preconditioned conjugate gradient method (PCG) [27] with diagonal (Jacobi) preconditioning. PCG is stopped as soon as the relative residual norm is smaller than  $10^{-8}$ .

Following [26], equation (6.3) can be solved by performing only a left projection, that is, only the left space  $\mathcal{K}_d(\mathbf{M}^{-1}\mathbf{K}, \mathbf{M}^{-1}c_1)$  is constructed. The projected problem has the form

$$\widetilde{\mathbf{H}}_d \mathbf{Y} + \mathbf{Y}\mathbf{B}_2\mathbf{B}_1^{-1} = e_1 \beta_1 c_2^\top \mathbf{B}_1^{-1},$$

where  $\widetilde{\mathbf{H}}_d \in \mathbb{R}^{d \times d}$  stems from the adopted projection technique, namely full Arnoldi ( $\widetilde{\mathbf{H}}_d = \mathcal{H}_d$ ),  $\tau$ -PSKM ( $\widetilde{\mathbf{H}}_d = \mathbf{H}_d$ ), and s-PSKM ( $\widetilde{\mathbf{H}}_d = \widehat{\mathbf{H}}_d + \widehat{H}e_d^\top$ ).

We compare the three schemes as  $n$  varies, in terms of number of iterations and computational time, by checking the stopping criterion every  $p = 20$  iterations, and  $\text{tol} = 10^{-6}$ . For the sketched-and-truncated scheme we adopt  $s = 1200$ ; the truncation parameter takes the value  $k = 3$  in both s-PSKM and  $\tau$ -PSKM.

The results displayed in Table 4 show that full Arnoldi and s-PSKM perform a rather similar number of iterations for all tested values of  $n$  with s-PSKM being always faster than full Arnoldi. On the contrary,  $\tau$ -PSKM needs a much larger number of iterations to converge, at least twice as many as with s-PSKM, leading to correspondingly large CPU time.

<sup>9</sup>This operation is performed by means of the Matlab routine `mrdivide`. On the other hand, by using a specifically adapted procedure, the solution of the reduced Sylvester equation could be performed without explicitly forming  $\mathbf{B}_2\mathbf{B}_1^{-1}$ ; this fact should be considered for finer time discretizations.

$n$	Full Arnoldi			T-PSKM			s-PSKM		
	$d$	Time	Mem	$d$	Time	Mem	$d$	Time	Mem
393968	260	100.93	261	500	172.96	20	260	84.57	16
482768	280	145.43	281	780	317.47	22	280	131.25	16
685214	300	224.10	301	720	352.41	22	280	196.65	16
880370	320	327.02	321	860	531.06	22	340	258.65	16

TABLE 4. **Example 6.2.** Performance of all considered methods for varying  $n$  ( $\ell = 800$ , truncation  $k = 3$ ). The reported CPU timings are in seconds. “Mem” indicates the number of vectors of length  $n$  used.

For this problem, the real potential of s-PSKM and T-PSKM can be appreciated by looking at their memory consumption. Indeed, both schemes require storing one order of magnitude fewer vectors of length  $n$  than what is done in full Arnoldi, making the memory requirements of the former solvers extremely affordable.

**Example 6.3.** In this example we compare the sketched-and-truncated strategy with the rational Krylov subspace method (RKSM); see, e.g. [17]. We consider the matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  in (1.1) coming from the centered finite differences discretization of three-dimensional convection-diffusion operators of the form (6.1) on  $[0, 1]^3$  with  $\nu = 0.005$  and convection vectors  $w_{\mathbf{A}} = (x \sin(x), y \cos(y), e^{z^2-1})$  and  $w_{\mathbf{B}} = ((1 - x^2)yz, 1, e^z)$ . The vectors  $c_1, c_2 \in \mathbb{R}^n$  have normally distributed random entries, and they are such that  $c_1 c_2^\top$  has unit norm.

The shifts employed in RKSM are computed by following the on-line procedure presented in [17]. Moreover, the large shifted linear systems occurring in the basis construction are solved by means of preconditioned BICGstab( $\ell$ ) [49] with no fill-in ILU preconditioning<sup>10</sup>, and similarly for  $\mathbf{B}$ . BICGstab( $\ell$ ) is stopped as soon as the relative residual norm reached  $10^{-8}$ .

The stopping threshold in s-PSKM and RKSM is  $\text{tol} = 10^{-6}$  while  $\text{maxit} = 250$ . Finally, in s-PSKM the truncation parameter was  $k = 3$  while the sketching dimension was  $s = 500$ .

In Table 5 we report the number of iterations, the computational timings and the rank of the solution computed by s-PSKM and RKSM by varying the problem dimension  $n$ . Table 5 also reports the average number of (inner) BICGStab( $\ell$ ) iterations,  $d_{\text{inn}}^{\mathbf{A}}$  and  $d_{\text{inn}}^{\mathbf{B}}$ , to approximately solve the shifted linear systems with  $\mathbf{A}$  and  $\mathbf{B}^\top$ , respectively, during the generation of the rational Krylov spaces. Although RKSM requires a much smaller space to converge, s-PSKM is always faster. Indeed, the solution of two shifted linear systems (one with  $\mathbf{A}$  and one with  $\mathbf{B}^\top$ ) at each RKSM iteration has a detrimental impact on the overall running time. Even though these values remain rather moderate, the BICGStab( $\ell$ ) iteration count increases with  $n$ , making the linear system solution step increasingly more expensive. Such shortcoming may be fixed by employing more robust preconditioning operators. On the other hand, s-PSKM requires only matrix-vector products and the cost of the orthogonalization step is moderate thanks to the truncated approach. Moreover,

<sup>10</sup>We employed the Matlab function `bicgstabl`, and for the preconditioner, `setup=struct('type','nofill');` `[LA,UA]=ilu(A,setup)`.

$n$	s-PSKM				RKSM			
	$d$	Time	rank	Mem	$d (d_{\text{inn}}^A, d_{\text{inn}}^B)$	Time	rank	Mem
125 000	140	2.69	29	58	28 (7.4, 6.9)	9.86	27	58+36
216 000	160	5.36	30	60	28 (7.1, 6.9)	17.57	27	58+36
343 000	180	11.29	31	62	29 (7.8, 7.3)	30.83	27	60+36
512 000	180	21.42	32	64	29 (8.2, 7.7)	48.92	27	60+36
729 000	200	35.42	32	64	29 (8.8, 8.1)	76.60	27	60+36
1 000 000	220	52.56	33	66	29 (9.6, 8.7)	114.91	29	60+36

TABLE 5. **Example 6.3.** Performance of s-PSKM and RKSM. The reported CPU timings are in seconds. The column “rank” indicates the rank of the solution matrix  $\mathbf{X}_d$  and “Mem” indicates the number of vectors of length  $n$  used. For RKSM, the memory requirements are reported as the sum of the memory required for basis allocation and for the preconditioner.

the possible delay coming from the use of locally orthogonal bases is mitigated by sketching.

s-PSKM turns out to be surprisingly competitive when compared to RKSM also in terms of storage allocation. Indeed, as reported in the columns “Mem” of [Table 5](#), the memory requirements of s-PSKM and RKSM are always rather similar. We mention that in case of RKSM we denote the storage demand as the sum of two terms to emphasize the different nature of the allocated components. While the first one is related to the basis storage, the second one comes from the allocation of the preconditioners. Indeed, for this example, storing the four incomplete LU factors is equivalent to storing about 36 vectors of length  $n$ .

## 7. CONCLUSIONS

By fully exploiting the promising combination of sketching and Krylov methods, in this paper we developed a sketched-and-truncated Krylov subspace method for matrix equations. In addition to making polynomial Krylov methods feasible, the approach turns out to be competitive also with respect to state-of-the-art schemes based on rational Krylov subspaces. In particular, the proposed method reduces the storage demand of the overall solution process, and concurrently provides a 50% CPU time speed-up over RKSM. Therefore, the sketched-and-truncated polynomial Krylov subspace method can be a valid candidate for solving large-scale matrix equations for which projection methods based on more sophisticated subspaces struggle due to, e.g., expensive linear systems solves.

The whitened-sketched recurrence produces matrices that can, in theory, endanger the applicability of the generated space as projection method; fortunately, the projected linear equations are usually solvable in practice, and very effective performance resulted in our numerical experiments. We have theoretically explained this perceivable contradiction. Indeed, we have proved that sketching takes place into what we have called the effective field of values of the involved matrices, for which the projected solution can be well-defined.

A convergence analysis for the sketched approximation has also been presented, which conveniently extends results obtained in the literature for the full Arnoldi method.

**Algorithm 2** Truncated Arnoldi method for Sylvester equations ( $\tau$ -PSKM)

---

**Input:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times n}$ ,  $C_1 \in \mathbb{R}^{n \times r}$ ,  $C_2 \in \mathbb{R}^{n \times r}$ , integers  $0 < k \leq \text{maxit} \ll n$ ,  $\text{tol} > 0$ ,  $p \geq 1$

**Output:**  $\mathbf{X}^{(1)}$ ,  $\mathbf{X}^{(2)}$  such that  $\mathbf{X}^{(1)}(\mathbf{X}^{(2)})^\top = \mathbf{X}_d$  approximately solves  $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = C_1 C_2^\top$

- 1: Compute skinny QRs:  $U_1 \boldsymbol{\beta}_1 = C_1$ ,  $V_1 \boldsymbol{\beta}_2 = C_2$
- 2: **for**  $d = 1, \dots, \text{maxit}$  **do**
- 3:   Compute  $\tilde{U} = \mathbf{A}U_d$ ,  $\tilde{V} = \mathbf{B}^\top V_d$
- 4:   **for**  $i = \max\{1, d - k + 1\}, \dots, d$  **do**
- 5:     Set  $\tilde{U} = \tilde{U} - U_i \mathbf{h}_{i,d}$  with  $\mathbf{h}_{i,d} = U_i^\top \tilde{U}$
- 6:     Set  $\tilde{V} = \tilde{V} - V_i \mathbf{g}_{i,d}$  with  $\mathbf{g}_{i,d} = V_i^\top \tilde{V}$
- 7:   **end for**
- 8:   Compute skinny QRs:  $U_{d+1} \mathbf{h}_{d+1,d} = \tilde{U}$  and  $V_{d+1} \mathbf{g}_{d+1,d} = \tilde{V}$
- 9:   **if**  $\text{mod}(d, p) = 0$  **then**
- 10:     Solve  $\mathbf{H}_d \mathbf{Y} + \mathbf{Y} \mathbf{G}_d^\top = e_1 \boldsymbol{\beta}_1 \boldsymbol{\beta}_2 e_1^\top$  for  $\mathbf{Y}$
- 11:     Compute  $\rho = \sqrt{dr}(\|\mathbf{Y} E_d \mathbf{g}_{d+1,d}^\top\|_F + \|\mathbf{h}_{d+1,d} E_d^\top \mathbf{Y}\|_F)$
- 12:     **if**  $\rho < \text{tol}$  **then**
- 13:       Go to line 17
- 14:     **end if**
- 15:   **end if**
- 16: **end for**
- 17: Compute (possibly low-rank) factors  $\mathbf{Y}_1, \mathbf{Y}_2$  such that  $\mathbf{Y} \approx \mathbf{Y}_1 \mathbf{Y}_2^\top$
- 18: Retrieve  $\mathbf{X}^{(1)} = U_d \mathbf{Y}_1$ ,  $\mathbf{X}^{(2)} = V_d \mathbf{Y}_2$  by the two-pass step

---

As a supplementary numerical contribution of this paper, we have derived and analyzed truncated Krylov methods, which appear to be new in the matrix equation context.

## ACKNOWLEDGMENTS

The first and third authors are members of the INdAM Research Group GNCS that partially supported this work through the funded project GNCS2023 “Metodi avanzati per la risoluzione di PDEs su griglie strutturate, e non” with reference number CUP\_E53C22001930001.

## APPENDIX A. ALGORITHMIC DESCRIPTION OF TRUNCATED ARNOLDI

In [Algorithm 2](#) we report the algorithm pseudocode for the truncated polynomial Krylov method for Sylvester equations. Notice that, in principle, one can use the sketched inner product also in the computation of the truncated bases  $U_d$  and  $V_d$ , thus avoiding any inner product in  $\mathbb{R}^n$ . We refrain from reporting also this pseudocode here as it can be easily derived from [Algorithm 2](#).

## APPENDIX B. TENSORIZED SUBSPACE EMBEDDINGS

In our context of solving matrix equations, sketchings of the form  $\mathbf{S}_U \mathbf{M} \mathbf{S}_V^\top$  occur; see, e.g., formula (3.5) involving the norm of the sketched residual.

The embeddings  $\mathbf{S}_U$  and  $\mathbf{S}_V$  are constructed as usual subspace embeddings for the block Krylov spaces  $\mathcal{K}_{d+1}(\mathbf{A}, C_1), \mathcal{K}_{d+1}(\mathbf{B}^\top, C_2)$ , so that (2.2) holds for  $\mathbf{S}_U$  and  $v \in \mathcal{K}_{d+1}(\mathbf{A}, C_1)$  as well as for  $\mathbf{S}_V$  and  $v \in \mathcal{K}_{d+1}(\mathbf{B}^\top, C_2)$ . We now investigate what this implies for  $\|\mathbf{S}_U \mathbf{M} \mathbf{S}_V^\top\|_F$  in relation to  $\|\mathbf{M}\|_F$ , when  $\mathbf{M}$  is of the form  $\mathbf{M} = \mathbf{U}_{d+1} \mathbf{Z} \mathbf{V}_{d+1}^\top$  (as it is the case for the Sylvester equation residual).

Our derivations closely follow the ideas in [14], generalizing their results from random Gaussian matrices to general  $\varepsilon$ -subspace embeddings. We start with the following simple observation.

**Proposition B.1.** *Let  $\mathbf{S} \in \mathbb{R}^{s \times n}$  be an  $\varepsilon$ -subspace embedding of  $\mathcal{V}$  with  $\dim(\mathcal{V}) = d + 1$  and let the columns of  $\mathbf{V}$  form an orthonormal basis of  $\mathcal{V}$ . Then  $\mathbf{S}\mathbf{V}$  is an  $\varepsilon$ -subspace embedding for  $\mathbb{R}^{d+1}$ .*

*Proof.* Let  $z \in \mathbb{R}^{d+1}$ . Then  $\mathbf{V}z \in \mathcal{V}$ , so that by (2.2), we have

$$(1 - \varepsilon)\|\mathbf{V}z\|^2 \leq \|\mathbf{S}\mathbf{V}z\|^2 \leq (1 + \varepsilon)\|\mathbf{V}z\|^2,$$

The result follows by noting that  $\|\mathbf{V}z\| = \|z\|$  due to the orthonormality of the columns of  $\mathbf{V}$ .  $\square$

We further need the following auxiliary result.

**Proposition B.2** (Lemma 1 and Corollary 3 in [14]). *Let  $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{s \times (d+1)}$  be  $\varepsilon$ -subspace embeddings of  $\mathbb{R}^{d+1}$ . Then  $\mathbf{S}_i \otimes \mathbf{I}_n$  and  $\mathbf{I}_n \otimes \mathbf{S}_i, i = 1, 2$  are  $\varepsilon$ -subspace embeddings of  $\mathbb{R}^{n(d+1)}$  and  $\mathbf{S}_1 \otimes \mathbf{S}_2$  is an  $\varepsilon(2 + \varepsilon)$ -subspace embedding of  $\mathbb{R}^{(d+1)^2}$ .*

With these results, we can now state a tensorized embedding property.

**Theorem B.3.** *Let  $\mathbf{S}_U \in \mathbb{R}^{s \times n_1}$  be an  $\varepsilon$ -subspace embedding for the subspace  $\mathcal{U} \subseteq \mathbb{R}^{n_1}$  and  $\mathbf{S}_V \in \mathbb{R}^{s \times n_2}$  be an  $\varepsilon$ -subspace embedding for the subspace  $\mathcal{V} \subseteq \mathbb{R}^{n_2}$ , with  $\dim(\mathcal{U}) = \dim(\mathcal{V}) = d + 1$ . Further, let the columns of  $\mathbf{U}, \mathbf{V}$  contain orthonormal bases of  $\mathcal{U}, \mathcal{V}$ , respectively.*

*Then, for any  $\mathbf{M} = \mathbf{U}\mathbf{Z}\mathbf{V}^\top$ , we have*

$$(B.1) \quad (1 - \tilde{\varepsilon})\|\mathbf{M}\|_F \leq \|\mathbf{S}_U \mathbf{M} \mathbf{S}_V^\top\|_F \leq (1 + \tilde{\varepsilon})\|\mathbf{M}\|_F$$

*with  $\tilde{\varepsilon} = \varepsilon(2 + \varepsilon)$ .*

*Proof.* Using the vectorized form, for any  $\mathbf{M}$ , we have  $\|\mathbf{M}\|_F = \|\text{vec}(\mathbf{M})\|$ . By a direct calculation, we obtain

$$\|\mathbf{S}_U \mathbf{M} \mathbf{S}_V^\top\|_F = \|(\mathbf{S}_V \mathbf{V} \otimes \mathbf{S}_U \mathbf{U}) \text{vec}(\mathbf{Z})\|.$$

By Proposition B.1,  $\mathbf{S}_U \mathbf{U}$  and  $\mathbf{S}_V \mathbf{V}$  are  $\varepsilon$ -subspace embeddings of  $\mathbb{R}^{d+1}$ , so that by Proposition B.2,  $\mathbf{S}_V \mathbf{V} \otimes \mathbf{S}_U \mathbf{U}$ , is an  $\tilde{\varepsilon}$ -subspace embedding of  $\mathbb{R}^{(d+1)^2}$ . Thus,

$$(B.2) \quad (1 - \tilde{\varepsilon})\|\text{vec}(\mathbf{Z})\| \leq \|(\mathbf{S}_V \mathbf{V} \otimes \mathbf{S}_U \mathbf{U}) \text{vec}(\mathbf{Z})\| \leq (1 + \tilde{\varepsilon})\|\text{vec}(\mathbf{Z})\|.$$

Furthermore,  $\|\text{vec}(\mathbf{Z})\| = \|\text{vec}(\mathbf{M})\|$ , because  $(\mathbf{V} \otimes \mathbf{U}) \text{vec}(\mathbf{Z}) = \text{vec}(\mathbf{M})$  and  $\mathbf{V} \otimes \mathbf{U}$  has orthonormal columns. Inserting this into (B.2) completes the proof.  $\square$

*Remark B.4.* The lower bound in (B.1) is only informative if  $1 - \tilde{\varepsilon} > 0$ , or equivalently,  $\varepsilon < \sqrt{2} - 1 \approx 0.4142$ . This means that in order to obtain theoretical guarantees, one needs to impose a stricter subspace embedding condition than when

solving linear systems or approximating the action of matrix functions via sketched Krylov methods (where any  $\varepsilon < 1$  works in principle, and  $\varepsilon = 1/\sqrt{2}$  guarantees that, e.g., the sketched residual norm differs from the true residual norm by a factor of at most 6).

However, our numerical experiments indicate that in practice, a cruder tolerance typically also suffices for obtaining a method that works well.

## REFERENCES

1. A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Society for Industrial and Applied Mathematics, 2005.
2. W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Q. Appl. Math. **9** (1951), 17–29.
3. J. Baker, M. Embree, and J. Sabino, *Fast singular value decay for Lyapunov solutions with nonnormal coefficients*, SIAM J. Matrix Anal. Appl. **36** (2015), no. 2, 656–668.
4. O. Balabanov and L. Grigori, *Randomized block Gram–Schmidt process for solution of linear systems and eigenvalue problems*, arXiv preprint arXiv:2111.14641 (2021).
5. ———, *Randomized Gram–Schmidt process with application to GMRES*, SIAM J. Sci. Comput. **44** (2022), no. 3, A1450–A1474.
6. O. Balabanov and A. Nouy, *Randomized linear algebra for model reduction. Part I: Galerkin methods and error estimation*, Adv. Comput. Math. **45** (2019), no. 5, 2969–3019.
7. R. H. Bartels and G. W. Stewart, *Solution of the matrix equation  $ax + xb = c$* , Commun. ACM **15** (1972), no. 9, 820–826.
8. U. Baur, *Low rank solution of data-sparse Sylvester equations*, Numerical Linear Algebra with Applications **15** (2008), no. 9, 837–851.
9. B. Beckermann, *Image numérique, GMRES et polynômes de Faber*, Comptes Rendus Math. **340** (2005), no. 11, 855–860.
10. P. Benner, R.-C. Li, and N. Truhar, *On the ADI method for Sylvester equations*, Journal of Computational and Applied Mathematics **233** (2009), no. 4, 1035–1045.
11. P. Benner and J. Saak, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM-Mitteilungen **36** (2013), no. 1, 32–52.
12. D. A. Bini, B. Iannazzo, and B. Meini, *Numerical Solution of Algebraic Riccati Equations*, Society for Industrial and Applied Mathematics, 2011.
13. L. Burke and S. Güttel, *Krylov subspace recycling with randomized sketching for matrix functions*, arXiv preprint arXiv:2308.02290 (2023).
14. K. Chen, Q. Li, K. Newton, and S. J. Wright, *Structured random sketching for PDE inverse problems*, SIAM J. Matrix Anal. Appl. **41** (2020), no. 4, 1742–1770.
15. A. Cortinovis, D. Kressner, and Y. Nakatsukasa, *Speeding up Krylov subspace methods for computing  $f(A)b$  via randomization*, SIAM J. Matrix Anal. Appl. **45** (2024), no. 1, 619–633.
16. P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Subspace sampling and relative-error matrix approximation: Column-based methods*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Springer, 2006, pp. 316–326.
17. V. Druskin and V. Simoncini, *Adaptive rational Krylov subspaces for large-scale dynamical systems*, Systems & Control Letters **60** (2011), no. 8, 546–560.
18. M. Eiermann and O. G. Ernst, *A restarted Krylov subspace method for the evaluation of matrix functions*, SIAM J. Numer. Anal. **44** (2006), no. 6, 2481–2504.
19. A. Frommer, K. Lund, and D. B. Szyld, *Block Krylov subspace methods for functions of matrices II: Modified block FOM*, SIAM J. Matrix Anal. Appl. **41** (2020), no. 2, 804–837.
20. A. Frommer and V. Simoncini, *Matrix functions*, Model Order Reduction: Theory, Research Aspects and Applications (W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, eds.), Springer, Berlin Heidelberg, 2008, pp. 275–303.
21. L. Grasedyck, *Existence of a low rank or  $\mathcal{H}$ -matrix approximant to the solution of a Sylvester equation*, Numer. Linear Algebra Appl. **11** (2004), no. 4, 371–389.
22. L. Grasedyck and W. Hackbusch, *A Multigrid Method to Solve Large Scale Sylvester Equations*, SIAM Journal on Matrix Analysis and Applications **29** (2007), no. 3, 870–894.
23. Laura Grigori and Edouard Timsit, *Randomized Householder QR*, Tech. report, 2023.



24. S. Güttel and M. Schweitzer, *Randomized sketching for Krylov approximations of large-scale matrix functions*, SIAM J. Matrix Anal. Appl. **44** (2023), no. 3, 1073–1095.
25. S. Güttel and I. Simunec, *A sketch-and-select Arnoldi process*, arXiv preprint arXiv:2306.03592 (2023).
26. J. Henning, D. Palitta, V. Simoncini, and K. Urban, *Matrix Oriented Reduction of Space-Time Petrov-Galerkin Variational Problems*, Numerical Mathematics and Advanced Applications ENUMATH 2019 (Cham) (Fred J. Vermolen and Cornelis Vuik, eds.), Springer International Publishing, 2021, pp. 1049–1057.
27. M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards **49** (1952), 409–436 (1953). MR 0060307
28. N. J. Higham, *The Matrix Computation Toolbox*, <http://www.ma.man.ac.uk/~higham/mctoolbox>.
29. I. M. Jaimoukha and E. M. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal. **31** (1994), no. 1, 227–251.
30. D. Kressner, *Memory-efficient Krylov subspace techniques for solving large-scale Lyapunov equations*, 2008 IEEE International Conference on Computer-Aided Control Systems, IEEE, 2008, pp. 613–618.
31. D. Kressner, K. Lund, S. Massei, and D. Palitta, *Compress-and-restart block Krylov subspace methods for Sylvester matrix equations*, Numer. Linear Algebra Appl. **28** (2021), no. 1, e2339.
32. P.-G. Martinsson and J. A. Tropp, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numer. **29** (2020), 403–572.
33. Y. Nakatsukasa and J. A. Tropp, *Fast & accurate randomized algorithms for linear systems and eigenvalue problems*, arXiv preprint arXiv:2111.00113 (2021).
34. S. Oymak and J. A. Tropp, *Universality laws for randomized dimension reduction, with applications*, Inf. Inference **7** (2018), no. 3, 337–446.
35. D. Palitta, *Matrix equation techniques for certain evolutionary partial differential equations*, J. Sci. Comput. **87** (2021), no. 99.
36. D. Palitta, M. Schweitzer, and V. Simoncini, *Sketched and truncated polynomial Krylov methods: Evaluation of matrix functions*, arXiv preprint arXiv:2306.06481 (2023).
37. D. Palitta and V. Simoncini, *Matrix-equation-based strategies for convection–diffusion equations*, BIT **56** (2016), no. 2, 751–776.
38. ———, *Computationally enhanced projection methods for symmetric Sylvester and Lyapunov matrix equations*, J. Comput. Appl. Math. **330** (2018), 648–659.
39. T. Penzl, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems & Control Letters **40** (2000), no. 2, 139–144.
40. Vladimir Rokhlin and Mark Tygert, *A fast randomized algorithm for overdetermined linear least-squares regression*, Proc. Natl. Acad. Sci. USA **105** (2008), no. 36, 13212–13217.
41. Y. Saad, *Numerical solution of large Lyapunov equations*, Signal Processing, Scattering, Operator Theory, and Numerical Methods. Proceedings of the international symposium MTNS-89, vol III (Boston) (M. A. Kaashoek, J. H. van Schuppen, and A. C. Ran, eds.), Birkhauser, 1990, pp. 503–511.
42. ———, *Iterative methods for sparse linear systems, 2nd edition*, SIAM, Philadelphia, 2000.
43. T. Sarlós, *Improved approximation algorithms for large matrices via random projections*, 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06), IEEE, 2006, pp. 143–152.
44. V. Simoncini, *Computational methods for linear matrix equations*, SIAM Rev. **58** (2016), no. 3, 377–441.
45. V. Simoncini and V. Druskin, *Convergence analysis of projection methods for the numerical solution of large Lyapunov equations*, SIAM J. Numer. Anal. **47** (2009), no. 2, 828–843.
46. V. Simoncini and D. B. Szyld, *The effect of non-optimal bases on the convergence of Krylov subspace methods*, Numerische Mathematik **100** (2005), 711–733.
47. E. Timsit, L. Grigori, and O. Balabanov, *Randomized orthogonal projection methods for Krylov subspace solvers*, arXiv preprint arXiv:2302.07466 (2023).
48. J. A. Tropp, *Improved analysis of the subsampled randomized Hadamard transform*, Adv. Adapt. Data Anal. **3** (2011), no. 01n02, 115–126.
49. H. A. van der Vorst, *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **13** (1992), 631–644.

50. D. P. Woodruff, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor. Comput. Sci. **10** (2014), no. 1–2, 1–157.

DIPARTIMENTO DI MATEMATICA, (AM)<sup>2</sup>, ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA,  
40126 BOLOGNA, ITALY

*Email address:* `daive.palitta@unibo.it`

SCHOOL OF MATHEMATICS AND NATURAL SCIENCES, BERGISCHE UNIVERSITÄT WUPPERTAL,  
42097 WUPPERTAL, GERMANY

*Email address:* `marcel@uni-wuppertal.de`

DIPARTIMENTO DI MATEMATICA, (AM)<sup>2</sup>, ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA,  
40126 BOLOGNA, AND IMATI-CNR, PAVIA, ITALY

*Email address:* `valeria.simoncini@unibo.it`