Electronic Transactions on Numerical Analysis.
Volume 40, pp. 414-435, 2013.
Copyright © 2013, Kent State University.
ISSN 1068-9613.

# COMPUTING APPROXIMATE EXTENDED KRYLOV SUBSPACES WITHOUT EXPLICIT INVERSION[*]

THOMAS MACH[†], MIROSLAV S. PRANIĆ[‡], AND RAF VANDEBRIL[†]

**Abstract.** It is shown that extended Krylov subspaces—under some assumptions—can be computed approximately without any explicit inversion or system solves involved. Instead, the necessary computations are done in an implicit way using the information from an enlarged standard Krylov subspace.

For both the classical and extended Krylov spaces, the matrices capturing the recurrence coefficients can be retrieved by projecting the original matrix on a particular orthogonal basis of the associated (extended) Krylov space. It is also well-known that for (extended) Krylov spaces of full dimension, i.e., equal to the matrix size, the matrix of recurrences can be obtained directly by executing similarity transformations on the original matrix. In practice, however, for large dimensions, computing time is saved by making use of iterative procedures to gradually gather the recurrences in a matrix. Unfortunately, for extended Krylov spaces, one is obliged to frequently solve systems of equations.

In this paper the iterative and the direct similarity approach are integrated, thereby avoiding system solves. At first, an orthogonal basis of a standard Krylov subspace of dimension $m_\ell + m_r + p$ and the matrix of recurrences are constructed iteratively. After that, cleverly chosen unitary similarity transformations are executed to alter the matrix of recurrences, thereby also changing the orthogonal basis vectors spanning the large Krylov space. Finally, only the first $m_\ell + m_r - 1$ new basis vectors are retained resulting in an orthogonal basis approximately spanning the extended Krylov subspace

$$\mathcal{K}_{m_\ell,m_r}(A,v) = \operatorname{span}\left\{A^{-m_r+1}v,\ldots,A^{-1}v,v,Av,A^2v,\ldots,A^{m_\ell-1}v\right\}.$$

Numerical experiments support the claim that this approximation is very good if the large Krylov subspace approximately contains $\operatorname{span}\left\{A^{-m_r+1}v,\ldots,A^{-1}v\right\}$. This can culminate in significant dimensionality reduction and as such can also lead to time savings when approximating or solving, e.g., matrix functions or equations.

**Key words.** Krylov, extended Krylov, iterative methods, Ritz values, polynomial approximation, rotations, QR factorization

**AMS subject classifications.** 65F60, 65F10, 47J25, 15A16

## 1. Introduction.

There is an intimate relation between orthogonal polynomials, their recurrence relations, and the associated matrix formalism in terms of classical Krylov spaces, the orthogonal basis vectors spanning the spaces, and their recurrences. This link proved to be of bidirectional prosperity for both the polynomial as well as the matrix communities, as illustrated by, e.g., a numerically reliable retrieval of the weights for Gauss quadrature [12, 21] and the convergence analysis of Krylov based algorithms relying on approximation theory and potential theory [18, 19, 31]. Approximations of functions by Laurent polynomials and rational functions have been present for a long time (see [4] and the references therein), but in [26] the matrix analogue in terms of Krylov subspaces was introduced for the first time.

[*] Received February 20, 2013. Accepted October 4, 2013. Published online on December 9, 2013. Recommended by L. Reichel. This research was partially supported by the Research Council KU Leuven, projects OT/11/055 (Spectral Properties of Perturbed Normal Matrices and their Applications), CREA/13/012 (Can Unconventional Eigenvalue Algorithms Supersede the State-Of-The-Art), PFV/10/002 (Optimization in Engineering, OPTEC), by the Fund for Scientific Research–Flanders (Belgium) project G034212N (Reestablishing Smoothness for Matrix Manifold Optimization via Resolution of Singularities), by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), by the Serbian Ministry of Education and Science project #174002 (Methods of Numerical and Nonlinear analysis with Applications), and by the DFG research stipend MA 5852/1-1. This research was initiated during a research visit of M. Pranić, supported by the JoinEU-SEE programme.

[†] Department Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven (Heverlee), Belgium ({Thomas.Mach, Raf.Vandebril}@cs.kuleuven.be).

[‡] Department Mathematics and Informatics, University of Banja Luka, M. Stojanovića, 51000 Banja Luka, Bosnia and Herzegovina (pranic77m@yahoo.com).

Since then rational Krylov spaces have been the subject of many studies; it is therefore impossible to provide an exhaustive listing of all the relevant literature. We attempt to highlight the references closest linked to the extended (pole free) case in the next paragraph. Ruhe initiated this research and constructed several algorithms related to (generalized) eigenvalue computations based on rational Krylov spaces; see e.g., [26, 27, 28, 29]. The relations with matrices and possible numerical issues were investigated in [6, 7, 20, 23]. Fasino proved in [9] that the matrix capturing the recurrence coefficients, though dense, is highly structured and dominated by low rank parts. This low rank structure was already exploited in eigenvalue and inverse eigenvalue problems [34, 35, 36, 37]. An analysis of convergence is presented in [3, 5]. The main bottleneck, however, in the design of these rational iterative methods still remains the computation of the vectors spanning the Krylov subspace, which requires successive system solves [22].

Rational Krylov methods [13] and extended Krylov methods in particular are popular for numerically approximating the action of a matrix function $f(A)$ on a vector $v$ [8, 14, 15, 16]. Extended Krylov subspace methods have also been used to solve Lyapunov equations [17] and have been proven useful in model order reduction [1]. In practice, a rational, extended or classical Krylov space defines a small subspace on which one projects the original matrix or problem, thereby reducing the dimension and leading to an approximate solution.

In an extended Krylov space defined by a matrix $A$ and a vector $v$, not only multiplications with positive powers of $A$ but also with negative powers are admitted. This extra flexibility often allows the extended spaces to be chosen much smaller than the standard Krylov subspaces for achieving a certain accuracy. As a result, the projected problem linked to the extended space can sometimes be much smaller than the corresponding projected problem linked to the standard Krylov subspace, but it still contains the vital properties of the original matrix. When building the extended Krylov subspace, system solves to obtain $A^{-1}v$ are necessary. In the numerical examples in the above mentioned papers, this is often done by using the MATLAB function `backslash` or a direct solver. For large systems, direct solvers often require too much storage or too much computation time. Therefore it is sometimes necessary to switch to an iterative solver, which in turn is again based on a Krylov subspace method. The approach presented here integrates the Krylov subspaces utilized for computing $A^{-k}v,\ k = 1, 2, \ldots,$ with the construction of the desired extended Krylov subspace.

More precisely, the proposed algorithm is initiated by building a large standard Krylov subspace of a certain dimension. After that, the compression procedure is initiated, and cleverly chosen unitary similarity transformations are executed on the matrix capturing the recurrence coefficients. As a result, the matrix of recurrences changes structure and approximates the matrix of recurrences linked to a predefined extended Krylov space. These similarity transformations do not alter the starting vector $v$ but do mix up the Krylov space. Finally, only a subset of all changed Krylov vectors is retained, which now approximate the vectors of the extended space.

Before the new algorithm is presented in Section 4, some essential facts on extended Krylov spaces, rotations, and operations on rotations are reviewed in Section 2. An extension of the implicit Q-theorem for Hessenberg matrices, see, e.g., [10], required for the validation of the results, is given in Section 3. Section 5 is confined to the error estimates introduced by approximating the extended space. In the numerical experiments in Section 6, it is shown that the new approach is feasible for some but not all cases: experiments for approximating matrix functions, approximately solving Lyapunov equations, computational timings, and visualizations of the behavior of the Ritz values are included.

## 2. Preliminaries.

The novel algorithm mostly relies on manipulating the QR factorization of the matrix of recurrences, where the matrix $Q$ itself is factored in essentially $2 \times 2$

rotations. This section elucidates transformations involving rotations (Section 2.2), and links the appearance of negative and positive powers of $A$ in the extended Krylov subspace to the ordering of the rotations when factoring the $Q$-factor in the QR factorization of the matrix of recurrences (Section 2.3). At first, after notational conventions, Krylov and extended Krylov spaces are introduced (Section 2.1).

The following notation is employed throughout this paper: matrices are typeset as upper case letters $A$, vectors as lower case $v$. Matrix elements are denoted as $A_{i,j}$ and MATLAB's colon notation is used, e.g., $A_{:,1:k}$ stands for the first $k$ columns of $A$. The Hermitian conjugate of a matrix $A$ is marked by a superscripted asterisk $A^*$. The $i$th standard basis vector is denoted by $e_i$ and $I_i$ stands for the $i \times i$ identity matrix.

### 2.1. Krylov and extended Krylov spaces.

Let $A \in \mathbb{C}^{n\times n}$ be a matrix and $v \in \mathbb{C}^n$ a vector. The *Krylov subspace*[1] $\mathcal{K}_m(A, v)$ is defined as

$$\mathcal{K}_m(A, v) = \operatorname{span}\left\{v, Av, A^2v, \ldots, A^{m-1}v\right\}.$$

Closely related is the *Krylov matrix* defined by $K_m(A, v) = [v, Av, A^2v, \ldots, A^{m-1}v]$. We use a calligraphic $\mathcal{K}$ for the space and a non-calligraphic $K$ for the matrix; the same convention holds for the extended Krylov subspace, which is defined below.

If the dimension of $\mathcal{K}_m(A, v)$ is $m$, then there exists an orthogonal matrix $V \in \mathbb{C}^{n\times m}$ such that

$$\operatorname{span}\left\{V_{:,1:k}\right\} = \operatorname{span}\left\{v, Av, A^2v, \ldots, A^{k-1}v\right\} \qquad \forall k \leq m. \tag{2.1}$$

An *extended* Krylov subspace is of the form

$$\mathcal{K}_{m_r,m_\ell}(A, v) = \operatorname{span}\left\{A^{-m_r+1}v, \ldots, A^{-1}v, v, Av, A^2v, \ldots, A^{m_\ell-1}v\right\}.$$

When building such a space, vectors are added one by one, either on the left (negative powers) or on the right (positive powers). To record which vector enlarges the subspace in each step, a *selection vector* $s$ is introduced, determining which vector from the bilateral sequence

$$\ldots, A^{m_\ell}v, A^{m_\ell-1}v, \ldots, A^2v, A^1v, v, A^{-1}v, A^{-2}v, \ldots, A^{-m_r+1}v, A^{-m_r}v, \ldots \tag{2.2}$$

is chosen next. To make the ordering in the bilateral sequence consistent with forthcoming deductions, the positive powers of $A$ are defined to be the left ($\ell$) sequence and the negative powers the right ($r$) sequence. The selection vector $s$ only comprises elements $\ell$ and $r$. The first vector of the extended space is always $v$. The second vector is $Av$ chosen from the left if $s_1 = \ell$ or $A^{-1}v$ selected from the right for $s_1 = r$. The $i$th successive vector in the extended Krylov space is taken left whenever $s_{i-1} = \ell$ or right if $s_{i-1} = r$, and it is selected next to the last picked vector on that side of the bilateral sequence. An alternative notation to $\mathcal{K}_{m_\ell,m_r}(A, v)$ is $\mathcal{K}_{s,m}(A, v)$, where $s$ is the selection vector and $m = m_\ell + m_r - 1$ is the number of vectors taken out of (2.2) to generate the extended Krylov space. The number of times $\ell$ appears in the first $m - 1$ components of $s$ equals $m_\ell$, and $m_r$ corresponds to the number of occurrences of $r$.

EXAMPLE 2.1. For example, a Krylov space's selection vector has only values $\ell$. The selection vector accompanying a pure (only inverse powers involved) extended Krylov space only comprises values $r$. The alternating occurrence of $\ell$'s and $r$'s leads to an extended Krylov space of the form

$$\mathcal{K}_{s,m}(A, v) = \operatorname{span}\left\{v, Av, A^{-1}v, A^2v, A^{-2}v, A^3v, A^{-3}v, \ldots\right\},$$

[1] For brevity we will call in the remainder of the paper the classical or standard Krylov subspace just Krylov subspace.

which, for unitary matrices, links closely to CMV matrices [32]. We come back to this in Example 2.5. There is no particular reason to restrict oneself to periodic vector successions, e.g., $s = \left[r\,\ell\,r\,r\,r\,\ell\,r\,\ldots\right]$ corresponds to

$$\mathcal{K}_{s,m}(A,v) = \operatorname{span}\left\{v, A^{-1}v, Av, A^{-2}v, A^{-3}v, A^{-4}v, A^{2}v, A^{-5}, \ldots\right\}.$$

It is well-known that in the Krylov space, the matrix of recurrences $H = V^{*}AV \in \mathbb{C}^{m\times m}$, often also named the *projected counterpart*, is an upper Hessenberg matrix (i.e., $H_{i,j} = 0$, for all $i > j+1$). In the extended case, however, this does not longer hold. The structure of the projected counterpart is examined in Section 2.3 and relies on concepts introduced in the next section.

**2.2. Rotations and their manipulations.** Rotations [11] (also called Givens or Jacobi transformations) are commonly used to set entries in a matrix to zero, e.g., in order to retrieve the QR decomposition of a matrix.

DEFINITION 2.2. *Matrices $G(i,j,\theta)$ which are equal to the identity, except for the positions $G_{i,i} = \cos(\theta)$, $G_{i,j} = \sin(\theta)$, $G_{j,i} = -\overline{\sin(\theta)}$, and $G_{j,j} = \overline{\cos(\theta)}$ are named rotations.*

We will restrict ourselves to rotations $G(i,i{+}1,\theta)$ acting on neighboring rows or columns, abbreviated as $G_i$. A rotation $G$ is unitary, that is, $G$ applied to a vector leaves the 2-norm unchanged. By the *action of a rotation*, we mean the effect that $G$ has on the rows/columns of the matrix to which it is multiplied. To keep track of the action of a rotation, we typically represent them graphically by a bracket having arrows pointing to the rows respectively columns affected, e.g.,

$$\begin{smallmatrix}\ulcorner\!\!\to\\ \llcorner\!\!\to\end{smallmatrix}\begin{bmatrix}\times & \times\\ 0 & \times\end{bmatrix} = \begin{bmatrix}\times & \times\\ \times & \times\end{bmatrix}.$$

When forming a product of several rotations, their order and actions clearly matter. We say that they are organized in a particular *series* of rotations or satisfy a certain *pattern*.

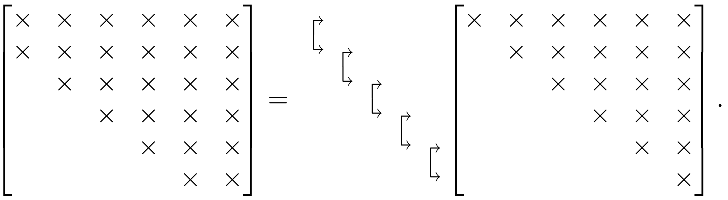
In this paper, we will nearly always operate on the QR factorization and in particular, on the factorization of the matrix $Q$ into rotations, which we also address as a *rotational factorization*. The role of the upper triangular matrix $R$ is inconsequential as one can transfer rotations from the left to the right through the upper triangular matrix without destroying its upper triangularity and without altering the pattern of the rotations involved. More precisely, applying a rotation acting on neighboring rows from the left to an upper triangular matrix introduces a non-zero entry on the sub-diagonal. One can always restore the upper triangular structure by eliminating this entry by a rotation from the right (the elements marked with a tilde are the only ones affected):

$$\begin{smallmatrix}\ulcorner\!\!\to\\ \llcorner\!\!\to\end{smallmatrix}\begin{bmatrix}\times & \times & \times & \times\\ 0 & \times & \times & \times\\ 0 & 0 & \times & \times\\ 0 & 0 & 0 & \times\end{bmatrix} = \begin{bmatrix}\times & \times & \times & \times\\ 0 & \tilde{\times} & \tilde{\times} & \tilde{\times}\\ 0 & \tilde{\times} & \tilde{\times} & \tilde{\times}\\ 0 & 0 & 0 & \times\end{bmatrix} = \begin{bmatrix}\times & \tilde{\times} & \tilde{\times} & \times\\ 0 & \tilde{\times} & \tilde{\times} & \tilde{\times}\\ 0 & 0 & \tilde{\times} & \tilde{\times}\\ 0 & 0 & 0 & \times\end{bmatrix}\begin{smallmatrix}\ulcorner\!\!\to\\ \llcorner\!\!\to\end{smallmatrix}\ .$$

This operation, passing rotations from one side to the other is called a *transfer*. Of course, one can transfer rotations from the right to the left as well. Moreover, let $Q$ be a matrix factored into $2\times 2$ rotations obeying a particular pattern. Transferring one rotation after the other through the upper triangular matrix shows that the rotational pattern remains unaffected. This means that a matrix $A$ having an RQ factorization $A = \hat{R}\hat{Q}$ admits a QR factorization $A = QR$, where the rotational factorizations of $Q$ and $\hat{Q}$ obey the same pattern.

**2.3. The projected counterpart, extended Krylov spaces, and patterns in the QR factorization.** This section discusses the connection between the extended Krylov subspace and the structure of the QR factorization of the projected counterpart.

Let us first consider an $n \times n$ Hessenberg matrix. Its QR decomposition can be written as a descending series of rotations times an upper triangular matrix, e.g.,



The unitary matrix $Q$ is thus decomposed into $n-1$ rotations according to a *position vector* $p = [\ell\,\ell\,\ell\,\ell\,\ell]$, which captures the positioning of successive rotations with respect to each other: an entry $p_i = \ell$ signifies that the rotation $G_i$ is positioned to the left of the rotation $G_{i+1}$, whereas $p_i = r$ indicates that $G_i$ is positioned to the right of $G_{i+1}$.

When going from classical Krylov spaces to extended Krylov spaces, one can no longer guarantee the projected counterpart to remain of Hessenberg form. Nevertheless these matrices, let us name them *extended Hessenberg* matrices, share major properties with the classical Hessenberg matrix when comparing their QR factorizations. Each extended Hessenberg matrix admits a QR factorization with Q factored into $n-1$ rotations $G_i$ for $i = 1, \ldots, n-1$. Recall that $G_i$ acts on neighboring rows $i$ and $i+1$. Due to noncommutativity, it clearly matters whether, for $|i-j| = 1$, $G_i$ is positioned to the left or to the right of $G_j$. So the mutual arrangement of successive rotations is stored in the position vector, uniquely characterizing the rotational pattern in the QR factorization of an extended Hessenberg matrix.

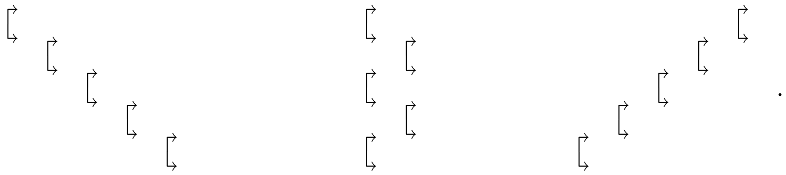
DEFINITION 2.3. *Let $A$ be a matrix having a QR decomposition $A = QR$. If the unitary matrix $Q$ admits a decomposition into at most $n-1$ rotations all acting on different pairs of neighboring rows, then we will call $A$ an extended Hessenberg matrix.*

*If $Q$ can be decomposed into exactly $n-1$ rotations differing from the identity, we will call $A$ an unreduced extended Hessenberg matrix.*

Whenever $A$ is of extended Hessenberg form, the matrix $Q$, with $A = QR$ being a QR factorization, will also be of extended Hessenberg form.

EXAMPLE 2.4. Equation (2.3) displays the rotational pattern of the $Q$-factors showing up in the QR factorization of a Hessenberg (left), a CMV matrix (center), and an inverse Hessenberg matrix (right).

(2.3)



In [36, 37] the link between extended Hessenberg matrices and extended Krylov spaces is examined. The position and selection vector nicely tie together both concepts: they are identical. Therefore, from now on, we will limit ourselves to the selection vector for both concepts. Summarizing, consider an extended Krylov space $\mathcal{K}_{s,m}(A, v)$ determined by its selection vector $s$. Let $V \in \mathbb{C}^{n \times m}$ be an orthogonal basis for this extended space such that

$$\operatorname{span}\{V_{:,1:k}\} = \mathcal{K}_{s,k}(A, v) \qquad \forall k \le m. \tag{2.4}$$

Then the matrix $V^*AV \in \mathbb{C}^{m\times m}$ will be of extended Hessenberg form. More precisely, the $Q$-factor in the QR decomposition of $V^*AV$ admits a decomposition into $m-1$ rotations $G_i$ acting on rows $i$ and $i+1$, where $G_i$ is positioned to the left of $G_{i+1}$ if $s_i = \ell$ or positioned to the right for $s_i = r$.

EXAMPLE 2.5. Reconsider Examples 2.1 and 2.4. Classical Krylov subspaces can be identified with a selection vector of only $\ell$'s and hence with a descending series of rotations as on the left of (2.3). It is not hard to see that a classical Krylov space generated by $A^{-1}$, results in a projected counterpart $V^*A^{-1}V$ being of Hessenberg form. Obviously, its inverse $V^*AV$ will thus be of inverse Hessenberg form. Both the pure extended space and the inverse Hessenberg matrix are described by a selection vector of solely $r$'s. The alternating vector $s = [\ell\, r\, \ell\, r\, \ldots]$ results in a *zigzag* shaped pattern, associated with the CMV decomposition.

## 3. The implicit Q-theorem for the extended case.

Given a matrix $A$ and a vector $v$, the selection vector has a strong impact on the structure and *essential uniqueness* of the projected counterpart, as shown in the next theorem. With *essential uniqueness* of the projected counterpart we mean uniqueness up to unitary similarity with a diagonal matrix. When considering essential uniqueness of the matrix $V$ of orthogonal vectors, we mean uniqueness up to unimodular scaling of each column.

THEOREM 3.1 (From [36, 37]). *Let $A$ be a non-singular matrix, $s$ a selection vector, and let $V$ and $\hat{V}$ be two unitary matrices sharing the first column, i.e., $Ve_1 = \hat{V}e_1$. Assume that both projected counterparts are QR-factored as*

$$QR = H = V^*AV \qquad \text{and} \qquad \hat{Q}\hat{R} = \hat{H} = \hat{V}^*A\hat{V}. \tag{3.1}$$

*If $Q$ and $\hat{Q}$ are extended Hessenberg matrices factored into non-identity rotations following the ordering imposed by $s$, then the matrices $H$ and $\hat{H}$ are essentially the same.*

Theorem 3.1 is an extension of the so called implicit Q-theorem for Hessenberg matrices, stating that once the matrix structure—determined by the selection vector—and the first vector $Ve_1$ are fixed, everything else is implicitly defined. For our purpose, this theorem is not general enough: we require essential uniqueness of a part of the projected counterparts (typically of a strictly smaller dimension than the matrix). In this case, the matrices $V$ and $\hat{V}$ are not necessarily square anymore, the associated selection vector(s) need only be defined for the first $k$ components, and we cannot guarantee all rotations to be different from the identity. Generalizing this, we first reformulate Theorem 3.1 dealing with reducible matrices.

THEOREM 3.2. *Let $A$ be a non-singular matrix, $s$ a selection vector, and let $V$ and $\hat{V}$ be two unitary matrices sharing the first column, i.e., $Ve_1 = \hat{V}e_1$. Assume both projected counterparts are QR-factored as in (3.1). Denote the individual rotations appearing in the rotational factorizations of $Q$ and $\hat{Q}$ as $G_i^Q$ and $G_i^{\hat{Q}}$, respectively, where the subscript $i$ indicates that the rotation acts on rows $i$ and $i+1$. Assume both patterns of rotations satisfy the ordering imposed by $s$. Define $\hat{k}$ as the minimal $i$ for which either $G_i^Q$ or $G_i^{\hat{Q}}$ equal the identity, i.e.,*

$$\hat{k} = \min_i \left\{ 1 \le i \le n-2, \text{ such that } G_i^Q = I \text{ or } G_i^{\hat{Q}} = I \right\},$$

*and if no such rotation exists, set $\hat{k} = n-1$. Then the upper left $\hat{k} \times \hat{k}$ parts of $H$ and $\hat{H}$ are essentially the same, as are the first $\hat{k}$ columns of $V$ and $\hat{V}$.*

Theorem 3.2 follows directly from the more general Theorem 3.5, which we prove below.

COROLLARY 3.3. *Under the assumptions of Theorem 3.2 and for $\hat{k} = n-1$, the two tuples ($V$,$H$) and ($\hat{V}$,$\hat{H}$) are essentially unique as a result of the unitarity of $V$ and $\hat{V}$.*

*Proof*. If $\hat{k} = n-1$, then according to Theorem 3.2 the first $n-1$ columns of $V$ are essentially fixed. Since $\operatorname{span}\{V\} = \mathbb{C}^n$, the last column is then fixed as well. □

Theorem 3.2 states again a property related to a full projection, i.e., for square matrices $V$ and $\hat{V}$. Obviously, the conclusions are not the same when relaxing this condition as illustrated in the following example.

EXAMPLE 3.4. Take a $5 \times 5$ diagonal matrix $A = \operatorname{diag}(1,2,3,4,5)$ and starting vector $v = [1,1,1,1,1]^T$. Consider two Krylov spaces not of full dimension

$$\mathcal{K} = \operatorname{span}\left\{v, Av, A^2 v\right\} \qquad \text{and} \qquad \hat{\mathcal{K}} = \operatorname{span}\left\{v, Av, A^{-1} v\right\}.$$

The associated orthogonal matrices $V$ and $\hat{V}$ are

$$V = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{10}} & \frac{2}{\sqrt{14}} \\ \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{10}} & \frac{-1}{\sqrt{14}} \\ \frac{1}{\sqrt{5}} & 0 & \frac{-2}{\sqrt{14}} \\ \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{10}} & \frac{-1}{\sqrt{14}} \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{10}} & \frac{2}{\sqrt{14}} \end{bmatrix} \qquad \text{and} \qquad \hat{V} = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{10}} & \frac{.52}{3\alpha} \\ \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{10}} & \frac{-.425}{3\alpha} \\ \frac{1}{\sqrt{5}} & 0 & \frac{-.37}{3\alpha} \\ \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{10}} & \frac{-.065}{3\alpha} \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{10}} & \frac{.34}{3\alpha} \end{bmatrix},$$

having $\alpha^2 = 7.0775$. Using $V$ and $\hat{V}$ in the similarity transformation, we get for $H = V^* A V$ and $\hat{H} = \hat{V} A \hat{V}$

$$H = \begin{bmatrix} 3 & -\sqrt{2} & \\ -\sqrt{2} & 3 & \sqrt{\frac{14}{10}} \\ & \sqrt{\frac{14}{10}} & 3 \end{bmatrix} \qquad \text{and} \qquad \hat{H} = \begin{bmatrix} 3 & -\sqrt{2} & \\ -\sqrt{2} & 3 & 1.1089 \\ & 1.1089 & 2.3133 \end{bmatrix}.$$

Obviously both $H$ and $\hat{H}$ admit an identical pattern in the Q-factor of both QR factorizations, and secondly the matrices $V$ and $\hat{V}$ share the first column. Nevertheless, the projected counterparts are non-identical, neither are the third column vectors of $V$ and $\hat{V}$.

The difference is subtle. Only considering the selection vector associated to the projected counterparts, we see that $s = [\ell]$ suffices. For the Krylov space, however, as long as it has not reached its full dimension, the selection vectors $s = [\ell\,\ell]$ and $\hat{s} = [\ell\, r]$ differ and are vital to reconstruct the spaces $\mathcal{K}$ and $\hat{\mathcal{K}}$. We modify Theorem 3.2 accordingly.

THEOREM 3.5. *Let $A$ be a non-singular $n \times n$ matrix, $s$ and $\hat{s}$ be two selection vectors, and let $\underline{V}$ and $\underline{\hat{V}}$ be two $n \times (m+1)$, (with[2] $m < n$) rectangular matrices having orthonormal columns and sharing the first column $\underline{V} e_1 = \underline{\hat{V}} e_1$. Let $V$ and $\hat{V}$ be the principal leading submatrices of size $n \times m$ of $\underline{V}$ and $\underline{\hat{V}}$, respectively. Consider*

$$\begin{aligned} AV &= VH + r_m w_m^* = \underline{V}\,\underline{H} = \underline{V}\,Q\,R, \\ A\hat{V} &= \hat{V}\hat{H} + \hat{r}_m \hat{w}_m^* = \underline{\hat{V}}\,\underline{\hat{H}} = \underline{\hat{V}}\,\hat{Q}\,\hat{R}, \end{aligned} \tag{3.2}$$

*with $r_m, \hat{r}_m \in \mathbb{C}^n$, $w_m, \hat{w}_m \in \mathbb{C}^m$, $H, \hat{H} \in \mathbb{C}^{m \times m}$, $\underline{H}, \underline{\hat{H}} \in \mathbb{C}^{(m+1)\times m}$, and with the QR decompositions $H = QR$ and $\hat{H} = \hat{Q}\hat{R}$ of $H$ and $\hat{H}$, respectively, where $Q$ and $\hat{Q}$ are decomposed into a series of rotations ordered as imposed by $s$ and $\hat{s}$. Define $\hat{k}$ as follows*

$$\hat{k} = \min_i \left\{ 1 \leq i \leq m-1 \text{ such that, } G_i^Q = I, G_i^{\hat{Q}} = I, \text{ or } s_i \neq \hat{s}_i \right\}, \tag{3.3}$$

[2] The case $m = n$ requires a reformulation of (3.2) and is therefore excluded. One can fall back on Theorem 3.2.

*and if no such $\hat{k}$ exists, set $\hat{k}$ equal to $m$. Then the first $\hat{k}$ columns of $V$ and $\hat{V}$ and the upper left $\hat{k} \times \hat{k}$ blocks of $V^*AV$ and $\hat{V}^*A\hat{V}$ are essentially the same.*

In Example 3.4 we have $s_1 = \hat{s}_1$ and $s_2 \neq \hat{s}_2$ and thus $\hat{k} = 2$. This example confirms that $H_{1:2,1:2} = \hat{H}_{1:2,1:2}$. To actually prove Theorem 3.5, Lemma 3.6 is required.

LEMMA 3.6. *Let $H$ be an $m \times m$ matrix with $HP_k$ being of (rectangular) extended Hessenberg form for $1 \leq k < n$, where $P_k = [I_k, 0]^T \in \mathbb{R}^{m \times k}$. Assume that the unitary matrix $Q$, where $QR = HP_k$, has the first $k$ rotations in its rotational factorization ordered according to the selection vector $s$. Then $K_{s,k}(H, e_1)$ is upper triangular.*

The proof is identical to the proof of Theorem 3.7 from [37]: the clue is the necessity of having element $s_i$ available to make a statement for the $(i+1)$st subspace and to have non-identity rotations as well. Let us now prove Theorem 3.5.

*Proof of Theorem 3.5.* First we need to increase the matrices $V$, $H$, and their variants with a hat in size. Let $V_e$ and $\hat{V}_e$ be augmented square unitary matrices, sharing the first columns with $V$ and $\hat{V}$, respectively. The enlarged matrices $H_e$ and $\hat{H}_e$ are defined as the projected counterparts $V_e^* A V_e = H_e$ and $\hat{V}_e^* A \hat{V}_e = \hat{H}_e$. By Theorem 3.6, with $\hat{k}$ as in (3.3), we have $K_{s,\hat{k}}(H_e, e_1) = K_{s,n-1}(H_e, e_1)P_{\hat{k}}$ and $K_{\hat{s},\hat{k}}(\hat{H}_e, e_1) = K_{\hat{s},n-1}(\hat{H}_e, e_1)P_{\hat{k}}$ both upper triangular. Elementary computations provide us with

$$V_e K_{s,n-1}(H_e, e_1) = K_{s,n-1}(V_e H_e V_e^*, V_e e_1) = K_{s,n-1}(A, V_e e_1) = K_{s,n-1}(A, V e_1),$$

and similarly $\hat{V}_e K_{\hat{s},n-1}(\hat{H}_e, e_1) = K_{\hat{s},n-1}(A, \hat{V} e_1)$. Combining everything and projecting onto the first columns leads to

$$V_e K_{s,n-1}(H_e, e_1) P_{\hat{k}} = K_{s,n-1}(A, V e_1) P_{\hat{k}} = K_{\hat{s},n-1}(A, \hat{V} e_1) P_{\hat{k}} = \hat{V}_e K_{\hat{s},n-1}(\hat{H}_e, e_1) P_{\hat{k}}.$$

Uniqueness of the partial QR factorizations of the outer left and outer right factorizations yields the essential equality of the first $\hat{k}$ vectors of $V$ and $\hat{V}$. The rest follows trivially. □

## 4. An implicit extended Krylov subspace algorithm.

Building an extended Krylov subspace typically requires solving some linear systems. In this section, an algorithm for approximately computing an extended Krylov subspace without explicit system solves is presented.

To clarify the description of the algorithm (see Algorithm 1 for a pseudo-code version), it is accompanied by an example having selection vector $s = [\ell\, r\, \ldots]$. First, an oversampling parameter $p$ is chosen and the Krylov subspace $\mathcal{K}_{\tilde{m}}(A, v)$ with dimension $\tilde{m} = |s| + 1 + p$ (here $|s|$ equals the length of the vector $s$) is constructed. This oversampling parameter $p$ determines how many vectors in addition are put into the Krylov subspace before the transformation to the extended space starts. A large value of $p$ increases the computational cost of the algorithm, but it will also improve the approximation to the extended Krylov subspaces. Let $V$ be an orthogonal matrix forming a basis of $\mathcal{K}_{\tilde{m}}(A, v)$ satisfying (2.1). We have $AV = VH + re_{\tilde{m}}^*$ with $H$ in Hessenberg form.

Second, the QR decomposition of $H = QR$ using a series of rotations is computed[3]:

$$\begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \times & \cdots & \times & \times \\ & & \times & \times & \cdots & \times & \times \\ & & & \ddots & \ddots & \vdots & \times \\ & & & & \times & \times & \times \\ & & & & & \times & \times \end{bmatrix} = \underbrace{\begin{array}{ccccc} \Lsh & & & & \\ & \Lsh & & & \\ & & \Lsh & & \\ & & & \ddots & \\ & & & & \Lsh \end{array}}_{=Q} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \times & \cdots & \times & \times \\ & & \times & \times & \cdots & \times & \times \\ & & & \times & \cdots & \times & \times \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \times & \times \\ & & & & & & \times \end{bmatrix}}_{=R}.$$

In the third step, $H$ is transformed via unitary similarity transformations to the desired shape corresponding to the extended Krylov subspace having selection vector $s = [\ell\, r \ldots]$. The first rotation must always remain unaltered, since $V$'s first column must stay fixed. The first entry in $s$ is an $\ell$, entailing the second rotation to be on the right-hand side of the first one. Since this is already the case in the example, nothing remains to be done. The next entry is an $r$, meaning the third rotation must be brought to the other side. To this end, all the rotations starting from the third one are transferred through the upper triangular[4] $R$:

$$AV = V \begin{array}{cc} \Lsh & \\ & \Lsh \end{array} \begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \times & \cdots & \times & \times \\ & & \times & \times & \cdots & \times & \times \\ & & & \times & \cdots & \times & \times \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \times & \times \\ & & & & & & \times \end{bmatrix} \underbrace{\begin{array}{ccc} \Lsh & & \\ & \ddots & \\ & & \Lsh \end{array}}_{=W} + re_{\tilde m}^*.$$

To execute a similarity transformation on the Hessenberg matrix $H$, we multiply with $W^*$ from the right-hand side and set $\tilde V = VW^*$. As a result, we obtain

$$A\tilde V = \tilde V \underbrace{\underbrace{\begin{array}{ccccc} \Lsh & & & & \\ & \Lsh & & & \\ \Lsh & & & & \\ & & \ddots & & \\ & & & & \Lsh \end{array}}_{=\tilde Q} \begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \times & \cdots & \times & \times \\ & & \times & \times & \cdots & \times & \times \\ & & & \times & \cdots & \times & \times \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \times & \times \\ & & & & & & \times \end{bmatrix}}_{=\tilde H} + re_{\tilde m}^* W^*.$$

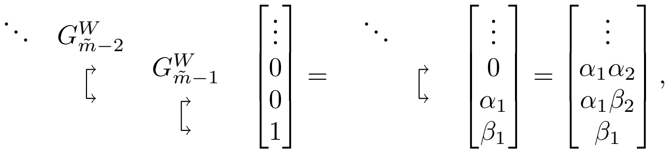
Note that $W$ is an orthogonal matrix and hence also $\tilde V$. The first three rotations in $\tilde H$ have now the shape for a selection vector beginning with $[\ell\, r]$. Next, all the other entries in $s$ are dealt with. If the entry in $s$ is $r$, the trailing rotations are transferred to the right and brought back to the left by similarity transformations. If the next entry is $\ell$, nothing is done. This procedure is repeated until the end of $s$ is reached; as a result $\tilde H$ is in the desired form.

[3] Probably there are much more economical manners of retrieving the QR factorization of $H$, e.g., by storing $H$ directly in factored form and updating the factors as in the SYMMLQ case [25]. This is, however, beyond the goal of this paper.

[4] Whenever the matrix $H$ is highly structured, e.g., tridiagonal, the QR decomposition partially destroys the existing structure. Typically, however, a new, exploitable structure emerges. We do not want to defer too much from the core message of the current paper and as such do not inspect this in detail.

We now have an approximation to the extended Krylov subspace with too many vectors. So in the fourth and last step, the first $|s|+1$ columns of $V$ and the upper $(|s|+1) \times (|s|+1)$ block of $H$ is retained.

Selecting only part of the entire decomposition introduces an approximation error (see Section 5) as also the residual is affected by the previous transformations and part of it gets ignored. More precisely, the original residual $re_{\tilde{m}}^*$ is transformed into $re_{\tilde{m}}^* W^*$, with $We_{\tilde{m}}$ of the following form


$\ddots$
$G_{\tilde{m}-2}^W$
$G_{\tilde{m}-1}^W$
$\vdots$ 0 0 1
=
$\ddots$
$\vdots$ 0 $\alpha_1$ $\beta_1$
=
$\vdots$ $\alpha_1\alpha_2$ $\alpha_1\beta_2$ $\beta_1$
,

with $G_{\tilde{m}-i}^W\big|_{\tilde{m}-i:\tilde{m}-i+1,\tilde{m}-i:\tilde{m}-i+1} = \begin{bmatrix} \alpha_i & \beta_i \\ -\bar{\beta}_i & \bar{\alpha}_i \end{bmatrix}$ and $|\alpha_i|, |\beta_i| \leq 1$. The product $|\prod_i \alpha_i|$ is expected to be smaller than one and is possibly decaying to zero fast, of course depending on the properties of $H$, $A$, and $\mathcal{K}_{\tilde{m}}(A, v)$. So, if the first $|s|+1$ entries of $(e_{\tilde{m}}^* W^*)_{1:|s|+1}$ are negligibly small, then we can apply Corollary 4.1 and know that we have computed a good approximation.

COROLLARY 4.1. *Having computed $\tilde{V}$ and $\tilde{H}$ as described above, assuming the matrix $r(e_{\tilde{m}}^* W^*)_{1:|s|+1}$ is zero, and none of the rotations in the factorization of $\tilde{Q}$ equals the identity, then $\tilde{V}$ and $\tilde{H}$ are essentially the same as if $V$ were computed as the orthogonal basis of the extended Krylov subspace $\mathcal{K}_{s,|s|+1}(A, v)$ and $H = V^*AV$.*

*Proof.* The first rotation remains unaltered and as such $Ve_1 = \tilde{V}e_1$. Applying Theorem 3.5 yields the result. □

It will be shown in Section 5 that this algorithm works well in practice if $A^{-1}v$ has a good approximation within the space spanned by $V$.

**5. Error bounds.** In this section we will show that the algorithm computes a good approximation to the extended Krylov subspace if $A^{-1}v$ is well approximated in the large Krylov subspace $\mathcal{K}_{\tilde{m}}(A, v)$.

For our analysis, a matrix $\tilde{A}$ is needed for which the algorithm will not approximate but compute the exact extended Krylov subspace linked to the original matrix $A$. Consider the matrix

$$\tilde{A} = A - rv_{\tilde{m}}^*. \tag{5.1}$$

Corollary 4.1 implies that Algorithm 1 computes the exact solution if the residual $\|r\|$ is zero. Obviously $\tilde{A}v_i = Av_i, \forall i < \tilde{m}$, since $V$ has orthonormal columns, implying that up to size $\tilde{m}$, the Krylov subspaces $\mathcal{K}_{\tilde{m}}(A, v)$ and $\mathcal{K}_{\tilde{m}}(\tilde{A}, v)$ are identical. Because of

$$\tilde{A}v_{\tilde{m}} = Av_{\tilde{m}} - rv_{\tilde{m}}^* v_{\tilde{m}} = VH_{:,\tilde{m}},$$

we obtain $\tilde{A}V = VH$. Hence $\tilde{A}$ is a matrix for which the algorithm computes the exact extended Krylov subspace identical to the computed approximation when applying the algorithm to $A$. The difference $\|\tilde{A} - A\|_2$ is, however, a too large overestimation to be an adequate error measure because even when the algorithm produces a good approximation, the norm can be large.

First, assume that in the selection vector $s$ only one $r$ appears, and so the extended Krylov subspace contains only a single vector $A^{-1}v$ besides positive powers of $A$ times $v$. This means in fact that the algorithm computes $\mathcal{K}_{s,|s|+1}(\tilde{A}, v)$ instead of $\mathcal{K}_{s,|s|+1}(A, v)$.

**Algorithm 1:** Computing an extended Krylov subspace without inversion.

```
Input: $A \in \mathbb{C}^{n\times n}$, $v \in \mathbb{C}^n$, $s$, e.g., $s = \left[\ell\, r\, \ell\, r \ldots\right]$, oversampling parameter $p$
Output: $H$, $V$ with $AV = VH + V_{:,m+1}e_m^* + \varrho h^* \approx VH + V_{:,m+1}e_m^*$
$\tilde{m} := |s| + 1 + p$; $m := |s| + 1$;
Compute $V$ spanning the Krylov subspace $\mathcal{K}_{\tilde{m}}(A, v)$, $H := V^*AV$, and
    $\varrho := (AV - VH)e_{\tilde{m}}$, with $AV = VH + re_{\tilde{m}}^*$ and $e_{\tilde{m}} = I_{:,1:\tilde{m}}$;
$h := e_{\tilde{m}}$;
Compute the QR-factorization of $H = QR$ into $\tilde{m} - 1$ rotations
    $G_1G_2 \ldots G_{\tilde{m}-1} := Q$ and an upper triangular $R$;
for $j = 1, \ldots, |s|$ do
    if $s(j) == r$ then
        Compute the $RQ$-factorization of $R\prod_{i=j+1}^{\tilde{m}-1} G_i := \prod_{i=j+1}^{\tilde{m}-1} G_i R$;
        $V := V\prod_{i=\tilde{m}-1}^{j+1} G_i^*$;
        $h := \prod_{i=\tilde{m}-1}^{j+1} G_i h$;
    end
end
if $\|\varrho\|_2 \, \|h_{1:m}\|_2$ is small enough then
    $V := V_{:,1:m}$, $H := H_{1:m,1:m}$;
    return $V$ and $H$;
else
    Choose a larger $p$ and start again;
end
```

Note that the Krylov subspaces $\mathcal{K}_{s,|s|+1}(A, v)$ and $\mathcal{K}_{s,|s|+1}(\tilde{A}, v)$ are both spanned by the vectors $v, Av, A^2v, \ldots, A^{|s|-1}v$ and by $A^{-1}v$, respectively $\tilde{A}^{-1}v$. Hence, the norm of the difference between the last two vectors, $\|A^{-1}v - \tilde{A}^{-1}v\|_2$, is a measure of the accuracy of the computed extended Krylov space approximation. In Lemma 5.1 this norm is linked to the approximation accuracy of $A^{-1}v$ in the subspace $\mathcal{K}_{\tilde{m}}(A, v) = \operatorname{span}\{V\}$, which can be quantified by $\left\|(I - VV^*)A^{-1}v\right\|$.

LEMMA 5.1. *Take $A \in \mathbb{C}^{n\times n}$ and let $\tilde{A}$ be as in (5.1). Let $V$ be the matrix of orthonormal columns spanning $\mathcal{K}_{\tilde{m}}(A, v) = \mathcal{K}_{\tilde{m}}(\tilde{A}, v)$. Set $\gamma = \|VV^*A(I - VV^*)\|_2$, and assume that $H = V^*AV$ is invertible. Then*

$$\left\|A^{-1}v - \tilde{A}^{-1}v\right\|_2 \leq \left(1 + \gamma \left\|H^{-1}\right\|_2 \left\|V^*\right\|_2\right) \left\|(I - VV^*)A^{-1}v\right\|_2 .$$

*Proof*. It follows from $\tilde{A}V = VH$ that $\tilde{A}^{-1}V = VH^{-1}$ and $\tilde{A}V = VV^*AV$. We have (for all norms)

$$\begin{aligned} \left\|A^{-1}v - \tilde{A}^{-1}v\right\| &\leq \left\|(I - VV^*)A^{-1}v\right\| + x\left\|VV^*A^{-1}v - \tilde{A}^{-1}v\right\| \\ &\leq \left\|(I - VV^*)A^{-1}v\right\| + \left\|\tilde{A}^{-1}\tilde{A}VV^*A^{-1}v - \tilde{A}^{-1}v\right\| \\ &\leq \left\|(I - VV^*)A^{-1}v\right\| + \left\|\tilde{A}^{-1}VV^*AVV^*A^{-1}v - \tilde{A}^{-1}v\right\| . \end{aligned} \tag{5.2}$$

The projection of $v$ on $V$ is again $v$, hence $v = VV^*v$. As $VV^*$ is a projection, the identity $VV^* = VV^*VV^*$ holds. Using the sub-multiplicativity of the 2-norm, the second norm in (5.2) can be bounded as

$$\begin{aligned}(5.3)\qquad &\left\|\tilde{A}^{-1}(VV^*)VV^*AVV^*A^{-1}v - \tilde{A}^{-1}(VV^*)v\right\|_2 \\ &\qquad \leq \left\|\tilde{A}^{-1}VV^*\right\|_2 \left\|VV^*AVV^*A^{-1}v - v\right\|_2 .\end{aligned}$$

Furthermore,

$$(5.4)\qquad \left\|\tilde{A}^{-1}VV^*\right\|_2 = \left\|VH^{-1}V^*\right\|_2 \leq \underbrace{\|V\|_2}_{=1} \left\|H^{-1}\right\|_2 \|V^*\|_2 .$$

By combining (5.3), (5.4), and the following estimate [30, Proposition 2.1]

$$\left\|VV^*AVV^*A^{-1}v - v\right\|_2 \leq \gamma \left\|(I - VV^*)A^{-1}v\right\|_2 ,$$

the proof is completed. □

This lemma tells us that Algorithm 1 computes a good approximation to the sought extended Krylov subspace if $A^{-1}v$ is approximated well enough in $\mathcal{K}_{\tilde{m}}(A, v)$.

**6. Numerical experiments.** In Section 6.1 we compare the accuracy of the novel approach at first for the examples from [15], where explicit matrix inversions are used to approximate matrix functions (Examples 6.1–6.3), and secondly (Example 6.4 taken from [17]), we illustrate the possible gain in compression with the new approach when approximately solving Lyapunov equations. In Section 6.2, the behavior of the Ritz values is examined when executing the compression technique. And finally in Section 6.3, the computational complexity of the new method is analyzed.

**6.1. Accuracy of approximating matrix functions.** The approach of computing the extended Krylov subspace implicitly is suitable for approximating (some) matrix functions as the following numerical experiments show. The experiments for Examples 6.1–6.3 are taken from Jagels and Reichel in [15]. Four different selection vectors are used: with no $r$'s, with an $r$ at every second entry, every third, and every fourth entry. In this section the variable $m$, determining which vectors and submatrix to retain, is always taken as $|s| + 1$. The computations are performed in MATLAB. The main idea behind these examples is to show that one can do equally well as in [15] without explicit inversions, whenever the inverse operation of $A$ on $v$ is approximated well enough in the large subspace.

The implicit extended Krylov subspace method is used for the approximation of $f(A)v$. We have $H = V^*AV$, so $f(A)v$ can be approximated by

$$f(A)v \approx Vf(H)V^*v = Vf(H)e_1 \|v\|_2 .$$

Three functions were tested: $f(x) = \frac{\exp(-x)}{x}$, $f(x) = \log(x)$, and $f(x) = \frac{1}{\sqrt{x}}$. It is known that in these cases the approximations stemming from extended Krylov subspaces are often quite good. In Figures 6.1–6.6, the plotted error is a measure of the relative distance between $f(A)v$ and its approximation.

EXAMPLE 6.1. In this example, we demonstrate that we are able to reproduce the figures from [15, Examples 5.1–5.2], meaning that the implicit approach performs equally well as the explicit one. Consider a $1000 \times 1000$ symmetric positive definite Toeplitz matrix $A$ having entries

$$a_{i,j} = \frac{1}{1 + |i - j|} .$$

In Figures 6.1 and 6.2 we report the relative error of approximating $f(A)v$ for different selection vectors. In Figure 6.1 for $f(x) = \frac{\exp(-x)}{x}$ and in Figure 6.2 for $f(x) = \log(x)$. The vector $v$ has normally distributed random entries with mean zero and variance one. The oversampling parameter is $p = 100$. It is known that both functions can be approximated well by extended Krylov subspaces, and as a result, an almost identical behavior as in [15, Figures 5.1–5.2] is observed.

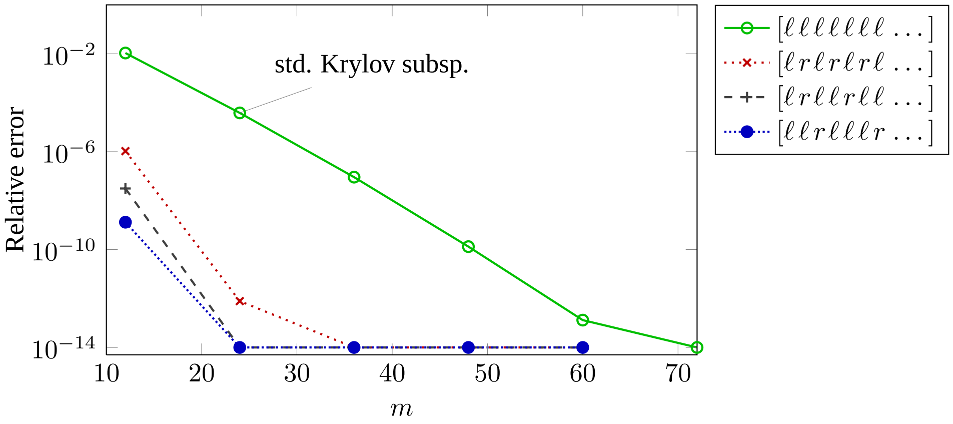


FIG. 6.1. *Relative error in approximating* $f(A)v$ *for* $f(x) = \frac{\exp(-x)}{x}$ *for various selection vectors* $s$ *and* $m = 12, 24, 36, 48, 60$.

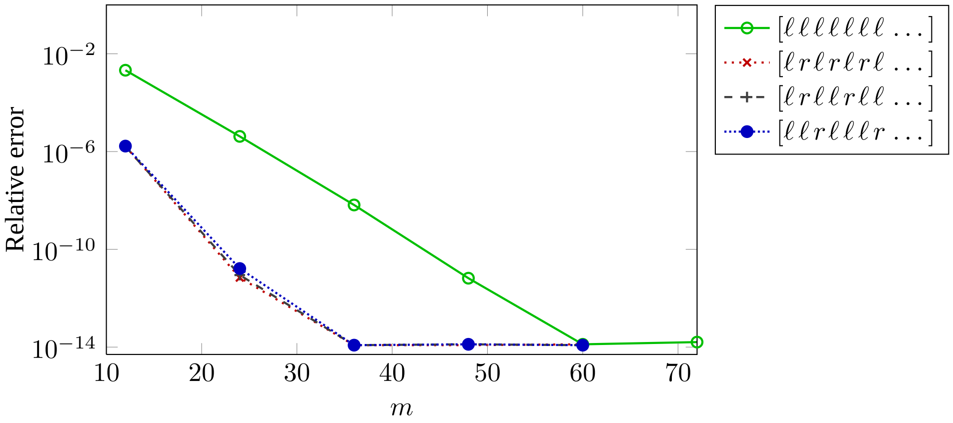


FIG. 6.2. *Relative error in approximating* $f(A)v$ *for* $f(x) = \log(x)$ *for various selection vectors* $s$ *and* $m = 12, 24, 36, 48, 60$.

EXAMPLE 6.2. In this example, the matrix $A$ arises from the discretization of the operator $L(u) = \frac{1}{10}u_{xx} - 100u_{yy}$ on the unit square as in [15, Examples 5.4–5.5]. The results are less accurate, but still reasonable approximations are retrieved. For the discretization in each direction, a three point stencil with 40 equally distributed interior points has been used. Together with a homogeneous boundary condition, this yields a $1600 \times 1600$ symmetric positive matrix $A$. The starting vector $v$ is chosen to be $v_j = \frac{1}{\sqrt{40}}$, for all $j$. Figure 6.3 displays the relative approximation error for $f(x) = \frac{\exp(-x)}{x}$ and Figure 6.4 for $f(x) = \frac{1}{\sqrt{x}}$.

We notice that the oversampling parameter $p = 100$ is not large enough, as the subspace $\mathcal{K}_{\tilde{m}}(A, v)$, depicted by the upper green line in Figure 6.3 is not approximating $A^{-1}v$ nor $f(A)v$ up to a satisfactory accuracy. After truncation (for $p = 100$), we arrive at the middle lines revealing an accuracy for the extended space almost identical as for the large untruncated Krylov space (depicted again by the green line containing, however, $p = 100$ additional vectors). The Krylov subspace of dimension $112$ can thus be reduced to an approximated extended Krylov subspace with only $12$ vectors, while retaining an almost identical relative error. The error of the approximated space with $12$ vectors is more than $3$ orders smaller than the error for a Krylov subspace of dimension $12$, which corresponds to the top green line.

An even larger oversampling parameter of $200$ is tested (corresponding to the bottom line in Figure 6.3) and a reduction of the dimension from $212$ of the classical Krylov space to $12$ for the extended Krylov subspace is observed without loss of accuracy. Moreover, the accuracy achieved with the approximated space is even $6$ orders better than the one attained by the classical Krylov space of only $12$ vectors.

In Figure 6.4, corresponding to $f(x) = \frac{1}{\sqrt{x}}$, almost the same behavior is observed when reducing a space of dimension $136$ respectively $236$ to an extended Krylov subspace of dimension $36$ with a selection vector $[\ell\, r\, \ell\, r\, \ell\, r\, \ldots]$.

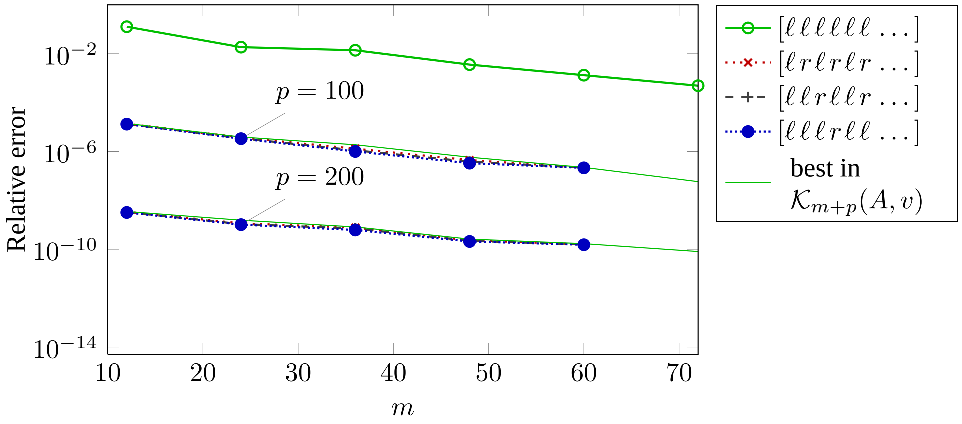


FIG. 6.3. *Relative error in approximating $f(A)v$ for $f(x) = \frac{\exp(-x)}{x}$ for various selection vectors $s$ and $m = 12, 24, 36, 48, 60$.*

EXAMPLE 6.3. In this example, a matrix $A$ for which $A^{-1}v$ does not lie in the Krylov subspace is taken. The algorithm is expected to fail here. The matrix $A$ is a symmetric indefinite matrix of the following form

$$A = \begin{bmatrix} B & C \\ C^* & -B \end{bmatrix} \in \mathbb{R}^{1000\times 1000},$$

with a tridiagonal matrix $B$ with 2's on the diagonal and $-1$'s on the subdiagonals and $C$ is a matrix with all entries zero except for a $1$ in the lower left corner. The setting of Example 6.1 is repeated here for approximating $f(A)v$ with $f(x) = \frac{\exp(-x)}{x}$. Figure 6.5 reveals an equally bad performance as in the Krylov case.

In [15], the extended Krylov subspace was successful in the approximation of $f(A)v$ because of the use of explicit solves with the MATLAB backslash function. In practice, however, such solvers are not always available and often other iterative solvers are used to solve these systems of equations, which would lead to similar problems as observed here.

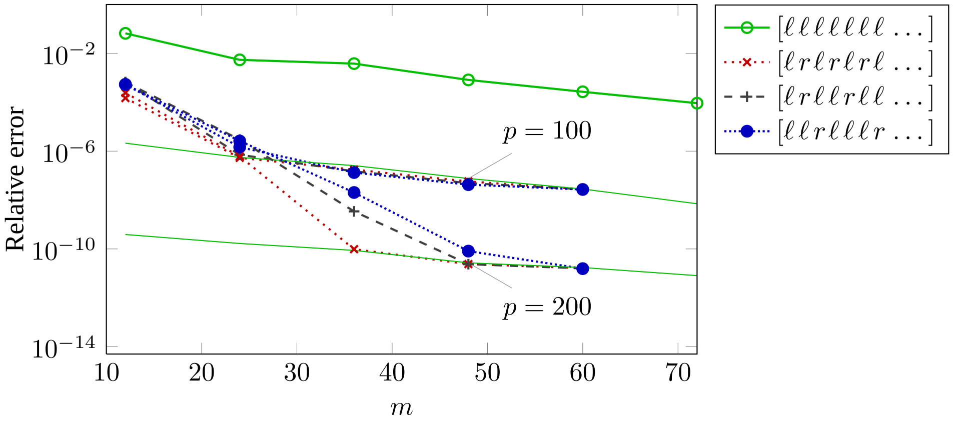


FIG. 6.4. *Relative error in approximating $f(A)v$ for $f(x) = \frac{1}{\sqrt{x}}$ for various selection vectors $s$ and $m = 12, 24, 36, 48, 60$.*

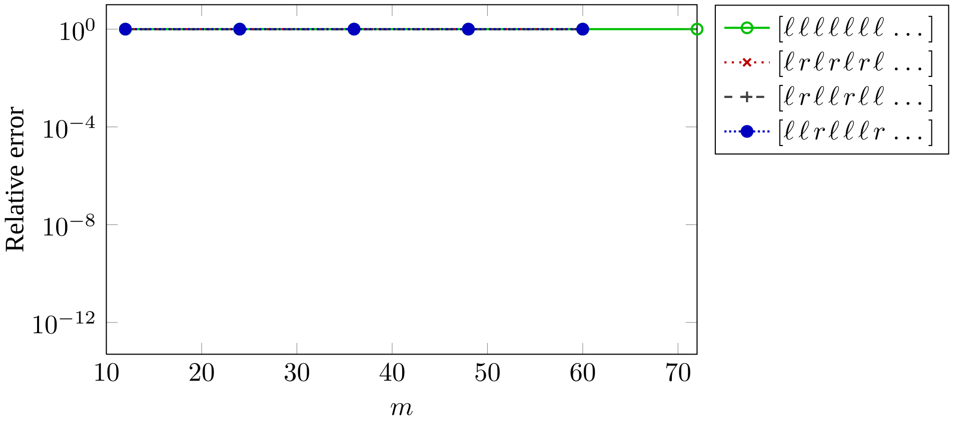


FIG. 6.5. *Relative error in approximating $f(A)v$ for $f(x) = \frac{1}{\sqrt{x}}$ for various selection vectors $s$ and $m = 12, 24, 36, 48, 60$.*

EXAMPLE 6.4. In this example [17, Example 4.2], the implicit extended Krylov subspace method is used for solving Lyapunov equations. The matrix $A \in \mathbb{R}^{5000\times 5000}$ is a diagonal matrix having eigenvalues $\lambda = 5.05 + 4.95\cos(\theta), \theta \in [0, 2\pi]$. The Lyapunov equation $AX + XA^* + BB^* = 0$ is considered with $B$ a vector with normally distributed entries with variance one and mean zero. In Figure 6.6 we report the relative difference (in the 2-norm) of the approximation $\tilde{X}$ computed via

$$\tilde{X} = V_{:,1:\tilde{m}} Y V_{:,1:\tilde{m}}^*,$$

where $Y$ is the solution of

$$\tilde{H}Y + Y\tilde{H} + (V_{:,1:r}^* B)(V_{:,1:r}^* B)^* = 0 \tag{6.1}$$

and the exact solution computed with the MATLAB function `lyapchol`. An oversampling parameter $p = 50$ was chosen. Compared to the standard Krylov subspace, the dimension of the small Lyapunov equation in (6.1) can be reduced by 50–65% without loss of accuracy.

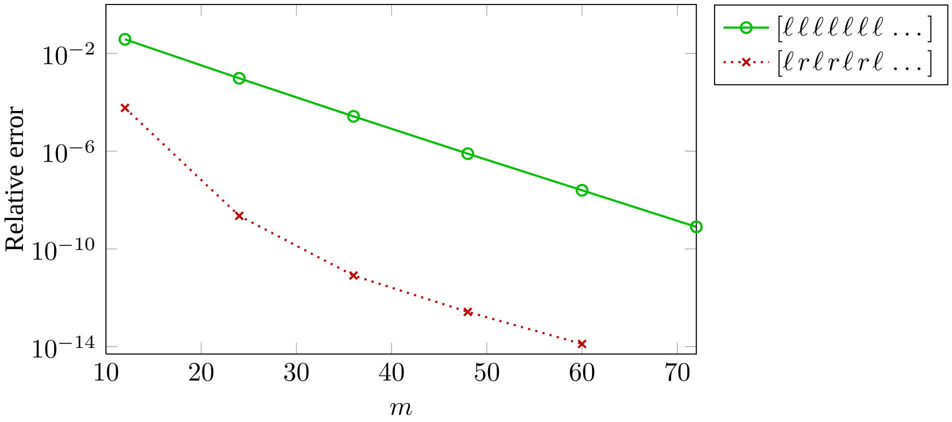


FIG. 6.6. *Relative error in the approximate solutions of $AX + XA^* + BB^* = 0$ for $m = 12, 24, 36, 48, 60$.*

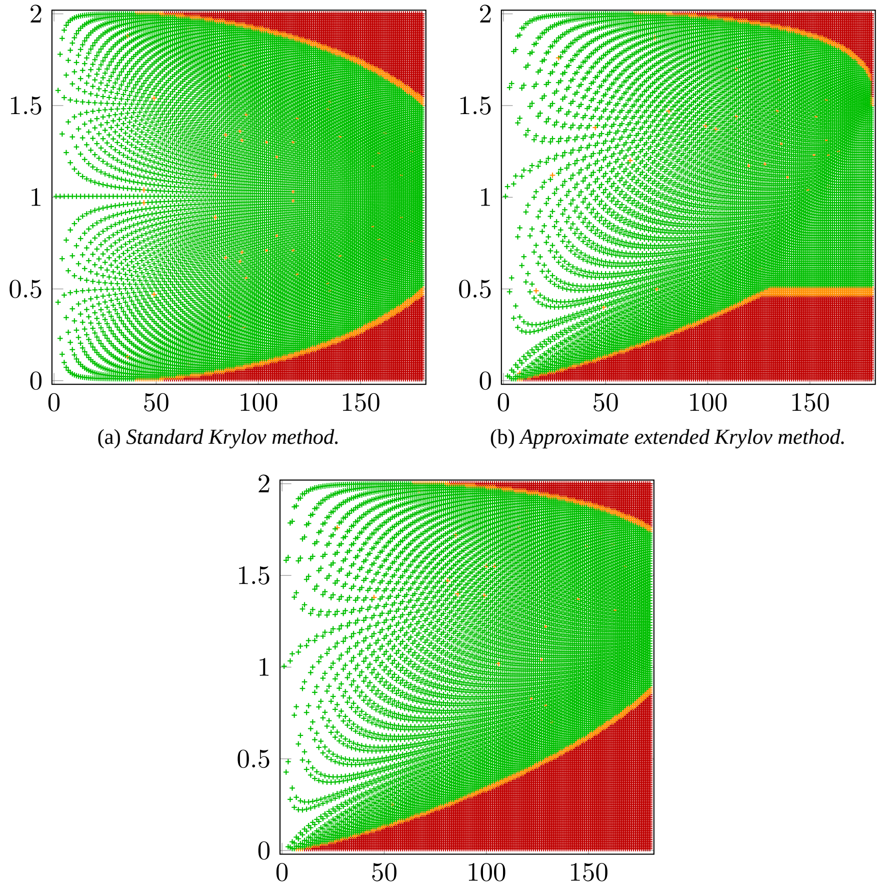
**6.2. Ritz values.** In the next three examples, we would like to highlight the fact that the algorithm starts with the information from the Krylov subspace and then squeezes this information into a smaller extended space. The experiments reveal that the truncated subspace will try to keep possession of all information linked to the extended space as long as possible.

In the next three examples, so-called Ritz plots (see Figures 6.7, 6.8, and 6.10) are depicted. In all these examples, the matrices under consideration have eigenvalues residing in a real interval; this interval corresponds to the range shown on the y-axis. The x-axis ranges from 0 to $m$, with $m$ being the dimension of $\mathcal{K}_m(A, v)$ or $\mathcal{K}_{s,m}(A, v)$. For each $0 < k < m$ on the x-axis, the eigenvalues of $V^*_{:,1:k}AV_{:,1:k}$, with $V$ as in (2.1) or (2.4), named the *Ritz values*, are computed and plotted parallel to the y-axis. Red crosses reveal Ritz values approximating eigenvalues quite well, having absolute error smaller than $1\,\mathrm{e}{-7.5}$. Yellow crosses represent good approximations with errors between $1\,\mathrm{e}{-7.5}$ and $1\,\mathrm{e}{-5}$, the green markers represent reasonable approximations, i.e., errors between $1\,\mathrm{e}{-5}$ and $1\,\mathrm{e}{-2.5}$ and the blue ones the remaining Ritz values.

EXAMPLE 6.5. Consider a simple diagonal matrix of size $200 \times 200$ with equal distributed eigenvalues between 0 and 2 and a uniform starting vector consisting solely of 1's. At first, the Krylov subspace of dimension $m = 180$ is computed for this matrix. A classical convergence pattern of the Ritz values, where first the extreme eigenvalues are found, is observed in Figure 6.7a. The second plot, Figure 6.7b, shows the Ritz values obtained after the truncation algorithm is applied to approximate an extended Krylov subspace; in this case the selection vector contains alternating $\ell$'s and $r$'s. The truncation is initiated once the Krylov subspace of size 180 was reached. Again the Ritz values according to the number of Krylov vectors retained are plotted. We start with dimension 180, and so it cannot be better than the final column of Figure 6.7a. Furthermore, the algorithm is also unable to outperform the results displayed in the third plot, Figure 6.7c, since this plot shows the eigenvalues for the exact extended spaces of dimension up to 180.

To envision what happens more clearly, a video (`equal_spaced_pos_HQ.mp4`) is generated[5]. The animation first shows the Ritz value plots for the classical Krylov space. The Ritz values are plotted concurrently while increasing the subspace's size. After dimension 180 is reached, the final column is separated from the plot and put on hold at the right on the screen, the classical Ritz values are kept in the background in gray. Next the Ritz value plot for the extended space is generated. One can now clearly see the difference between the

[5] The videos are also available at `http://people.cs.kuleuven.be/~thomas.mach/extKrylov/`.



(a) *Standard Krylov method.*

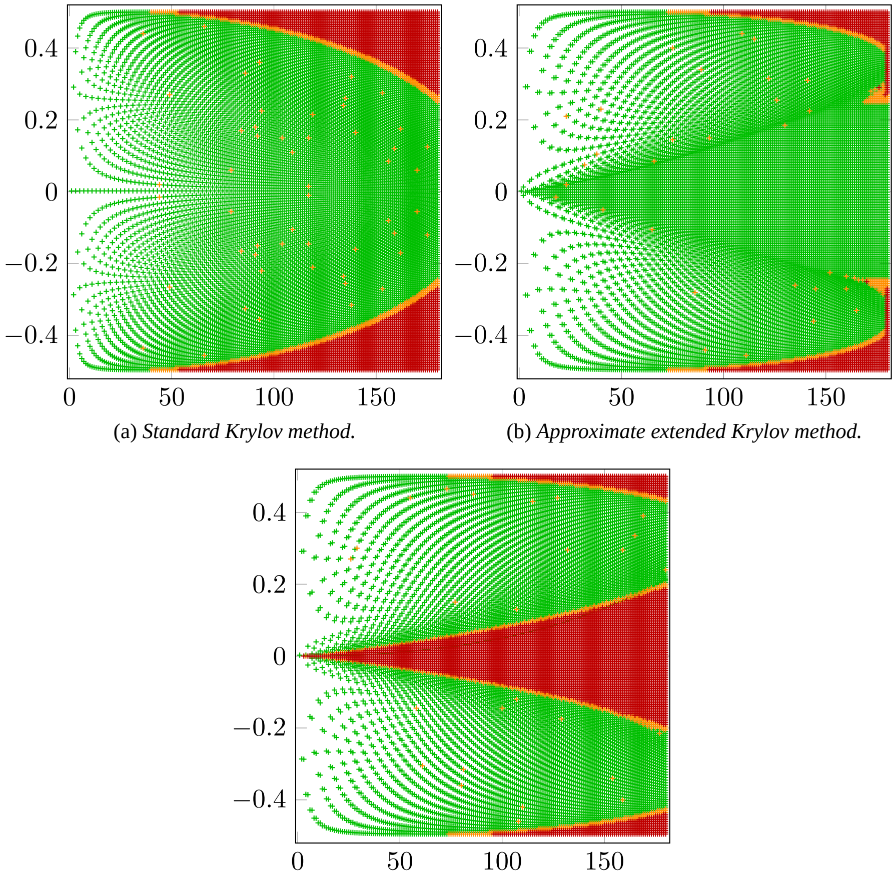
(b) *Approximate extended Krylov method.*

(c) *Extended Krylov method.*

FIG. 6.7. *Ritz plots for equal spaced eigenvalues in* $[0, 2]$.

extended and the classical case, where obviously the emphasis of the extended case is more towards zero. Now the interesting part starts: the extended space is kept where it is, and we start the truncation algorithm based on the Ritz values positioned on the outer right. The outer right vector moves back into the picture, and in each consecutive truncation step (diminishing of the subspace size), the Ritz values from the extended space are overwritten by the ones of the truncated space. Now one clearly sees how the truncation algorithm tries hard to match the extended space, but is strongly limited by the initially available information. Eventually, the truncation plot almost entirely integrates in the extended plot.

EXAMPLE 6.6. In the second example again a diagonal matrix is taken with equal distributed eigenvalues but now between $-\frac{1}{2}$ and $\frac{1}{2}$. We observe that the traditional Krylov method as before first locates the outer eigenvalues (Figure 6.8a). The extended Krylov method on the other hand (Figure 6.8c), due to its pole at zero, converges rapidly to the interior eigenvalues. The truncation strategy starts with the information from the standard Krylov space and tries to approximate the extended space as good as possible. Figure 6.8b visualizes that the truncation strategy tries to retain as much information as possible from the interior of the spectrum and rapidly disposes of the information near the edges. It is expected that the truncation strategy will fail in delivering accurate results when used for, e.g., approximating matrix functions. Again a video (`equal_spaced_sym_HQ.mp4`) is generated along



(a) *Standard Krylov method.*

(b) *Approximate extended Krylov method.*

(c) *Extended Krylov method.*

FIG. 6.8. *Ritz plots for equal spaced eigenvalues in* $[-.5, .5]$.

the same lines as in Example 6.5. In this case we see that the truncation algorithm quickly throws away most of the valuable information in its attempt to approximate the extended space. This is caused by the clear discrepancy between the approximations reached by the classical and the extended Krylov spaces.

EXAMPLE 6.7. In the final example again a diagonal matrix was taken with eigenvalues according to the distribution (see Figure 6.9)

$$\frac{\alpha+1}{2}(1-|x|)^{\alpha},$$

where $\alpha = -\frac{3}{4}$, as in [19]. The distribution shows that most of the eigenvalues are located at the boundaries $-1$ and $1$. Based on potential theory [18, 19], one knows that for this distribution first the inner eigenvalues, located around $0$, are found by classical Krylov methods. This implies that the classical Krylov space will have a similar goal as the extended Krylov approach namely first finding the eigenvalues around the origin. As before, Figures 6.10a–6.10c are generated. In this case the truncation strategy will work very well. A visualization video (`heavy_tail_HQ.mp4`) is also available.

**6.3. Computational efficiency.** In this section we investigate the computational efficiency of the new algorithm with respect to matrix function evaluations. Assume a matrix

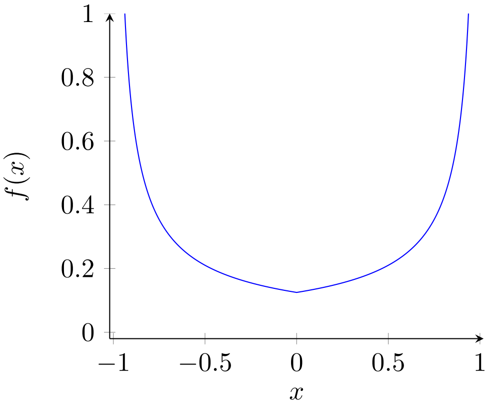
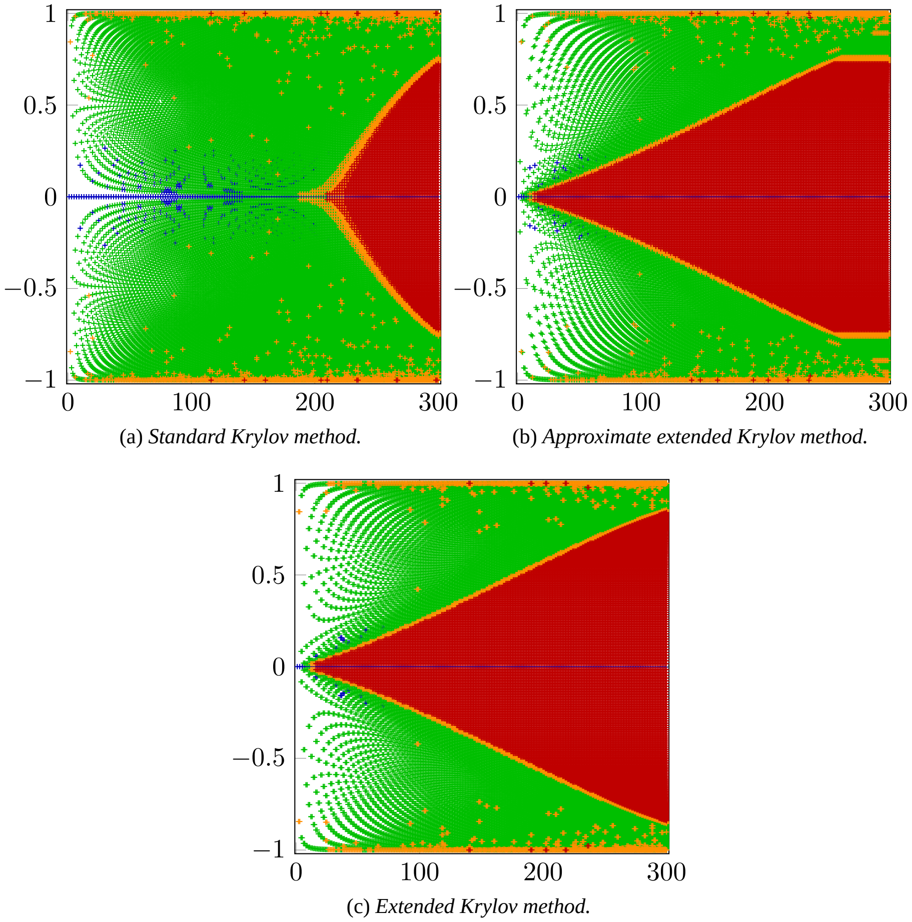


FIG. 6.9. *Eigenvalue distribution.*



(a) *Standard Krylov method.*

(b) *Approximate extended Krylov method.*

(c) *Extended Krylov method.*

FIG. 6.10. *Ritz plots for strong eigenvalue concentrations near the borders of* $[-1, 1]$.

linked to a Krylov space of dimension $|s| + p + 1$ is built and then truncated to an extended space of dimension $|s| + 1$. In practice it is impossible to estimate the time required for building the Krylov space because typically the matrix vector multiplications are the dominant factor and its complexity heavily depends on the algorithm or structures used. As this time is identical for both approaches, we do not report on it. Bare in mind, however, that overall it might occur to be the dominating computation. Nevertheless, even in this case, the proposed method is able to significantly reduce the size of the subspace resulting in equivalently significant memory savings.

So, for now, we neglect the time needed to construct the Krylov space and only investigate the forthcoming computations on the projected counterparts of sizes $|s|+1$ and $|s|+p+1$ including the time required for executing the compression. Each parameter $\ell$ in the selection vector $s$ implicates a transfer of at most $|s| + p$ rotations through an upper triangular matrix. Such a transfer costs $\mathcal{O}(|s| + p)$ flops. As there are at most $|s|$ $\ell$'s, we have an upper bound of $\mathcal{O}\left(|s|(|s| + p)^2\right)$ to complete the truncation process. Additionally, the transferred rotations are applied to $V$. This costs $\mathcal{O}(n)$ per rotation, where $n$ is the dimension of $A$, or $\mathcal{O}\left(n|s|(|s| + p)\right)$ in total. Naturally this is not the total complexity, and additional computations are exerted on the truncated and untruncated projected counterpart. For instance, assume this second phase to have cubical complexity. Then we arrive at a total cost of $\mathcal{O}\left((|s| + p)^3\right)$ for the untruncated matrix and at $\mathcal{O}\left(|s|(|s| + p)\right) + \mathcal{O}\left(|s|^3\right)$ operations for the truncated matrix. Clearly the turning point to arrive at cheaper algorithms is attained early.

EXAMPLE 6.8. The same operator as in Example 6.2 is used but now discretized with 70 equal distributed interior points, so that $A$ becomes a matrix of size $4900 \times 4900$. On the dense matrix $A$, the computation of $f(A)v$ relying on the MATLAB function `expm` took $18.4$ s. Due to the properties of $A$, a large oversampling parameter $p = 1600$ is required to achieve good results. For the Krylov subspace of dimension $1604$, $0.66$ s were needed to compute $f(A)v$ with a relative accuracy of $5.15\,e{-}11$. With the reduction approach, one is able to reduce the Krylov subspace to an extended Krylov subspace of dimension 4 ($s = [\ell\, r\, \ell]$) in $0.59$ s. Within this subspace one can compute $f(A)v$ to the same accuracy as in the large Krylov subspace in $0.001$ s. The computation of the large Krylov subspace was the most expensive part of the computation and took $126.6$ s.[6]

EXAMPLE 6.9. In this example a plain flop count is depicted. Let $A$ be a matrix of size $n \times n$ with $n = 10,000$. Again the computation of $f(A)v$ is the goal, which is conducted via the eigendecomposition of the matrix $A$ or the compressed matrix $V^*AV$. Assume this cost $15n^3$ with $n$ being the dimension of $A$ respectively $V^*AV$. Once the Krylov subspace of dimension $|s| + p + 1$ (costs are about $2n(|s| + p)^2$ flops) is computed, one can continue in two different ways. Either one directly computes the eigendecomposition or one first compresses the Krylov space and then computes the eigendecomposition. The compression requires about $|s|(2n(|s|+p)+2(|s|+p)^2)$ flops. Together, it requires $15|s|^3+|s|(2Nn+2n^2)$ flops versus $15(|s| + p)^3$ for the direct computation. For different values of $|s|$ and $|s| + p$, the flop counts are shown in Figure 6.11.

**7. Conclusions.** We have presented a new algorithm which often computes sufficiently accurate approximations to extended Krylov subspaces without using explicit inversion or explicit solves of linear systems. The numerical examples clearly illustrate these claims whenever the larger subspace approximates the action of $A^{-1}$ on the starting vector $v$ well enough. If, however, this constraint was not satisfied, it was shown that the presented approach was

[6] The computation of the Krylov subspace was done without any special tricks or optimization. This explains the large gap to the $18.4$ s for the computation for the full dense matrix.

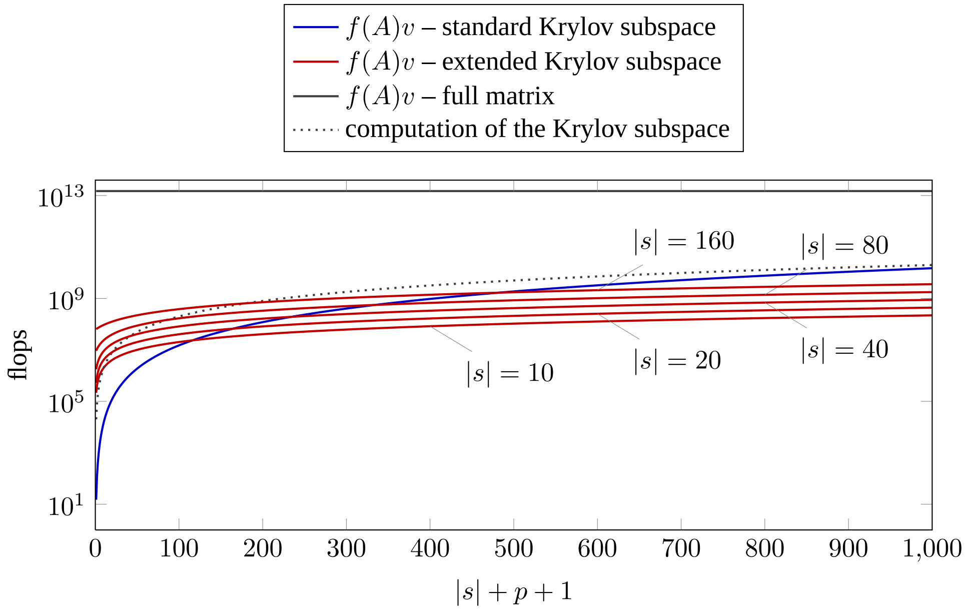


FIG. 6.11. *Complexity plot.*

able to significantly reduce the size of the Krylov space by bringing it to extended form without notable loss of accuracy with respect to the larger space. A larger compression can have multiple advantages such as reduced storage costs and reduced operation counts for subsequent computations. A final set of numerical experiments illustrates this latter statement revealing a nonneglectable reduction of computational efforts.

This research poses quite some questions. How is this related to the implicitly restarted Lanczos method [2, 24, 33] and can this truncation be used for restarts? Is it possible to go from extended Lanczos to rational Lanczos allowing the usage of shifts? Are there good heuristics to determine the selection vectors, the size of the initial large Krylov space, and the dimension of the truncated part?

**Acknowledgments.** The authors thank Karl Meerbergen for his valuable suggestions on the extended Krylov method, Carl Jagels and Lothar Reichel for helping us with the examples from [15] used in the numerical experiments, and the referees for their valuable comments.

## REFERENCES

[1] A. C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.
[2] C. BEATTIE, M. EMBREE, AND D. SORENSEN, *Convergence of polynomial restart Krylov methods for eigenvalue computations*, SIAM Rev., 47 (2005), pp. 492–515.
[3] B. BECKERMANN, S. GÜTTEL, AND R. VANDEBRIL, *On the convergence of rational Ritz-values*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1740–1774.
[4] A. BULTHEEL AND M. VAN BAREL, *Linear Algebra, Rational Approximation and Orthogonal Polynomials*, North-Holland, Amsterdam, 1997.
[5] A. CHESNOKOV, K. DECKERS, AND M. VAN BAREL, *A numerical solution of the constrained weighted energy problem*, J. Comput. Appl. Math., 235 (2010), pp. 950–965.
[6] K. DECKERS AND A. BULTHEEL, *Rational Krylov sequences and orthogonal rational functions*, Report TW499, Departement Computerwetenschappen, Katholieke Universiteit Leuven, Leuven, 2008.
[7] G. DE SAMBLANX, K. MEERBERGEN, AND A. BULTHEEL, *The implicit application of a rational filter in the RKS method*, BIT, 37 (1997), pp. 925–947.

[8] V. DRUSKIN AND L. KNIZHNERMAN, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 755–771.

[9] D. FASINO, *Rational Krylov matrices and QR-steps on Hermitian diagonal-plus-semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 743–754.

[10] J. G. F. FRANCIS, *The QR transformation. II.*, Comput. J., 4 (1962), pp. 332–345.

[11] G. H. GOLUB AND C. F. V. LOAN, *Matrix computations*, 4th ed., Johns Hopkins University Press, Baltimore, 2013.

[12] G. H. GOLUB AND J. H. WELSCH, *Calculation of Gauss quadrature rules*, Math. Comp., 23 (1969), pp. 221–230.

[13] S. GÜTTEL, *Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection*, GAMM-Mitt., 36 (2013), pp. 8–31.

[14] C. JAGELS AND L. REICHEL, *The extended Krylov subspace method and orthogonal Laurent polynomials*, Linear Algebra Appl., 431 (2009), pp. 441–458.

[15] ———, *Recursion relations for the extended Krylov subspace method*, Linear Algebra Appl., 434 (2011), pp. 1716–1732.

[16] L. KNIZHNERMAN AND V. SIMONCINI, *A new investigation of the extended Krylov subspace method for matrix function evaluations*, Numer. Linear Algebra Appl., 17 (2010), pp. 615–638.

[17] ———, *Convergence analysis of the extended Krylov subspace method for the Lyapunov equation*, Numer. Math., 118 (2011), pp. 567–586.

[18] A. B. J. KUIJLAARS, *Which eigenvalues are found by the Lanczos method?*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 306–321.

[19] ———, *Convergence analysis of Krylov subspace iterations with methods from potential theory*, SIAM Rev., 48 (2006), pp. 3–40.

[20] R. LEHOUCQ AND K. MEERBERGEN, *Using generalized Cayley transformations within an inexact rational Krylov sequence method*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 131–148.

[21] G. LÓPEZ LAGOMASINO, L. REICHEL, AND L. WUNDERLICH, *Matrices, moments, and rational quadrature*, Linear Algebra Appl., 429 (2008), pp. 2540–2554.

[22] K. MEERBERGEN, *Dangers in changing the poles in the rational Lanczos method for the Hermitian eigenvalue problem*, Tech. Report RAL-TR-1999-025, Rutherford Appleton Laboratory, Chilton, UK, 1999.

[23] ———, *Changing poles in the rational Lanczos method for the Hermitian eigenvalue problem.*, Numer. Linear Algebra Appl., 8 (2001), pp. 33–52.

[24] R. MORGAN, *On restarting the Arnoldi method for large non-symmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.

[25] C. PAIGE AND M. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[26] A. RUHE, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.

[27] ———, *Rational Krylov algorithms for nonsymmetric eigenvalue problems, II: Matrix pairs*, Linear Algebra Appl., 197/198 (1994), pp. 283–296.

[28] ———, *The Rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices*, BIT, 34 (1994), pp. 165–176.

[29] ———, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput, 19 (1998), pp. 1535–1551.

[30] Y. SAAD, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.

[31] ———, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[32] B. SIMON, *CMV matrices: Five years after*, J. Comput. Appl. Math., 208 (2007), pp. 120–154.

[33] D. C. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[34] M. VAN BAREL, D. FASINO, L. GEMIGNANI, AND N. MASTRONARDI, *Orthogonal rational functions and diagonal plus semiseparable matrices*, in Advanced Signal Processing Algorithms, Architectures, and Implementations XII, F. T. Luk, ed., vol. 4791 of Proceedings of SPIE, Bellingham, WA, 2002, pp. 167–170.

[35] ———, *Orthogonal rational functions and structured matrices*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 810–829.

[36] R. VANDEBRIL, *Chasing bulges or rotations? A metamorphosis of the QR-algorithm*, SIAM J. Matrix Anal. Appl, 32 (2011), pp. 217–247.

[37] R. VANDEBRIL AND D. S. WATKINS, *A generalization of the multishift QR algorithm*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 759–779.