Computing Approximate Extended Krylov Subspaces without Explicit Inversion

Thomas Mach　　*Miroslav S. Pranić*
Raf Vandebril

Report TW 623, February 2013

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Computing Approximate Extended Krylov Subspaces without Explicit Inversion

Thomas Mach   *Miroslav S. Pranić*
Raf Vandebril

Report TW 623, February 2013

Department of Computer Science, KU Leuven

Abstract

It will be shown that extended Krylov subspaces –under some assumptions– can be retrieved without any explicit inversion or system solves involved. Instead we do the necessary computations of $A^{-1}v$ in an implicit way using the information from an enlarged standard Krylov subspace.

It is well-known that both for classical and extended Krylov spaces, direct unitary similarity transformations exist providing us the matrix of recurrences. 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 required to frequently solve, in some way or another a system of equations. In this article both techniques will be integrated. We start with an orthogonal basis of a standard Krylov subspace of dimension $m + \overline{m} + p$. Then we will apply a unitary similarity built by rotations compressing thereby significantly the initial subspace and resulting in an orthogonal basis approximately spanning the extended Krylov subspace

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

Numerical experiments support our claims that this approximation is very good if the large Krylov subspace contains $\left\{A^{-\overline{m}+1}v, \cdots, A^{-1}v\right\}$ and thus can culminate in nonneglectable dimensionality reduction and as such also can lead to time savings when approximating, e.g., matrix functions.

Keywords : Krylov, extended Krylov, iterative methods, Ritz-values, truncation, rotations, QR factorization
MSC : Primary : 65F60, Secondary : 65F10, 47J25, 15A16.

COMPUTING APPROXIMATE EXTENDED KRYLOV SUBSPACES WITHOUT EXPLICIT INVERSION[*]

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

Abstract. It will be shown that extended Krylov subspaces –under some assumptions– can be retrieved without any explicit inversion or system solves involved. Instead we do the necessary computations of $A^{-1}v$ in an implicit way using the information from an enlarged standard Krylov subspace.

It is well-known that both for classical and extended Krylov spaces, direct unitary similarity transformations exist providing us the matrix of recurrences. 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 required to frequently solve, in some way or another a system of equations. In this article both techniques will be integrated. We start with an orthogonal basis of a standard Krylov subspace of dimension $m + \overline{m} + p$. Then we will apply a unitary similarity built by rotations compressing thereby significantly the initial subspace and resulting in an orthogonal basis approximately spanning the extended Krylov subspace

$$\mathcal{K}_{m,\overline{m}}(A, v) = \text{span}\left\{A^{-\overline{m}+1}v, \cdots, A^{-1}v, v, Av, A^2v, \ldots, A^{m-1}\right\}.$$

Numerical experiments support our claims that this approximation is very good if the large Krylov subspace contains $\left\{A^{-\overline{m}+1}v, \cdots, A^{-1}v\right\}$ and thus can culminate in nonneglectable dimensionality reduction and as such also can lead to time savings when approximating, e.g., matrix functions.

Key words. Krylov, extended Krylov, iterative methods, Ritz-values, truncation, 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 this space, and their recurrences. This link proved to be of bidirectional prosperity for both research incentives, as illustrated by, e.g., numerically reliable retrieving weights for Gauss quadrature [11] and the convergence analysis of Krylov based algorithms relying on approximation theory and potential theory [17, 18, 29]. Approximations of functions by Laurent polynomials and rational functions are around for a long time (see [4] and the references therein), but in [24] the matrix analogue in terms of Krylov subspaces was introduced for the first time.

Since then rational Krylov spaces have been the subject of many studies; it is therefore impossible to provide an exhaustive listing of all relevant literature. We attempt to highlight those references closest related to this publication. As the focus of this manuscript goes to the extended (pole free) case we elaborate more on this in the next paragraph. Ruhe initiated this research and constructed several algorithms related to (generalized) eigenvalue computations based on rational Krylov spaces, e.g., [25–27]. The relations with matrices and potential

[*] The research was partially supported by the Research Council KU Leuven, projects OT/11/055 (Spectral Properties of Perturbed Normal Matrices and their Applications), CoE EF/05/006 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 DFG research stipend MA 5852/1-1. This research was initiated during a research visit of M. Pranić, supported by the JoinEU-SEE programme.

[†] Dept. Computer Science, KU Leuven, Celestijnenlaan 200A, 3000 Leuven, Belgium (`Thomas.Mach@cs.kuleuven.be`).

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

[§] Dept. Computer Science, KU Leuven, Celestijnenlaan 200A, 3000 Leuven, Belgium (`Raf.Vandebril@cs.kuleuven.be`).

numerical issues were further investigated in [6,7,19,21]. Fasino proved in [9] that the matrix capturing the recurrence coefficients, though dense, is highly structured and dominated by low rank parts. Various techniques exploit this low rank structure already [32–35]. An analysis of the convergence is presented in [3,5], but the main bottleneck in the design of these rational iterative methods still remains computing the vectors spanning the Krylov subspace, requiring successive system solves, where we are not even mentioning the numerical difficulties that might arise for some selections of poles [20].

Rational Krylov methods [12], and extended Krylov methods in particular are popular for numerically approximating the action of a matrix function $f(A)$ to a vector v by computing $f(A)v$, e.g., see [8, 13–15]. Extended Krylov subspace have also been exploited to solve Lyapunov equations [16] or for model order reduction [1].

In an extended Krylov space, not only multiplications with positive powers of A, but also with negative powers are admitted. This extra flexibility often admits extended spaces to be chosen much smaller than the standard Krylov subspaces to achieve a certain accuracy. So the matrix A is transformed in $H = V^*AV$, being much smaller than the projected counterpart from the standard Krylov subspace, but still contains the vital properties of A. This means that by using extended Krylov subspaces one can reduce the cost for further computations with H, like $f(H)$ or the solution of the Lyapunov equation $XH + HX + B = 0$.

For the extended Krylov subspace one has to compute $A^{-1}v$. 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 are in turn often based again on a Krylov subspace method. The approach we are going to present here merges the Krylov subspaces utilized in the computation of $A^{-k}v,\ k = 1, 2, \ldots$ with the extended Krylov subspace.

More precisely, our algorithm is initiated by building a large standard Krylov subspace. Once a certain dimension is reached our compression procedure is initiated, and cleverly chosen unitary similarity transformations on the large Krylov subspace are executed. It is crucial to state that these similarities do not alter the starting vector v, but do mix up the Krylov space. As a result a much smaller approximate Krylov subspace is obtained, which is no longer of classical but of extended form. It must be stated that we are free in the selection of our extended space, so we can arbitrarily choose the succession of positive and negative powers k in the matrix vector product $A^k v$.

Before we explain the details of our new algorithm in Section 4, we review in Section 2 some essential facts on extended Krylov spaces, rotations, and operations one can do with them. We provide an extension of the implicit Q-theorem for extended Hessenberg matrices compulsory for the validation of our results in Section 3. Section 5 is confined to the error estimates introduced by the truncation of the standard Krylov space to an approximate extended space. In the numerical experiments in Section 6 we show that our new approach is feasible for some but not all cases where extended Krylov subspaces are used. More specifically, experiments for approximating matrix functions, approximately solving Lyapunov equations, computational timings, and visualizations of the Ritz-value behavior are included.

2. Preliminaries.

In this article the following notation is employed: matrices are typeset as upper case letters A, vectors as lower case v. The Hermitian conjugate of a matrix A is marked by a superscripted asterisk A^*.

Let $A \in \mathbb{C}^{n\times n}$ be a matrix and $v \in \mathbb{C}^n$ a vector. The Krylov subspace $\mathcal{K}_m(A, v)$ is the subspace spanned by $\mathcal{K}_m(A, v) = \text{span}\left\{v, Av, A^2v, \ldots, A^{m-1}v\right\}$. If the dimension of

$\mathcal{K}_m(A, v)$ is m, then there exists an orthogonal matrix V, so that

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

where $V_{:,1:k}$ stands for the first k columns of V (MATLAB colon notation). It is well-known that $H = V^*AV$ is an upper Hessenberg matrix, having $H_{i,j} = 0$, $\forall i > j+1$. If A and v are clear from the context we write short $\mathcal{K}_m$ for $\mathcal{K}_m(A, v)$.

It is beneficial to the understanding of the algorithm to work on a factored form of the associated projected counterpart V^*AV, where V spans a particular (extended) Krylov space. More precisely we will utilize the QR factorization of V, where the matrix Q itself is factored in essentially 2×2 rotations. This section eludes on how to operate on rotations, and links the appearance of negative and positive powers of A in the extended Krylov subspace to the ordering of the rotations in the Q factor stemming from the QR factorization of the projected counterpart.

2.1. Rotations and how to manipulate them. Rotations, and especially so-called Givens or Jacobi rotations, are commonly used to zero some entries of a matrix, e.g., to retrieve its QR decomposition. Let us recall the definition.

DEFINITION 2.1 (From [10]). *Matrices $G(i, j, \theta)$, everywhere equal to the identity, except for the 4 positions $G(i, i) = c = \cos(\theta)$, $G(i, j) = s = \sin(\theta)$, $G(j, i) = -\bar{s}$, and $G(j, j) = \bar{c}$ are called* Givens rotations.

In this paper we will restrict ourselves to transformations of the form $G(i, i+1, \theta)$, acting on neighbored columns resp. rows. Throughout the remainder of the text we will address them as rotations and denote them briefly as G_i.

It is easy to see that rotations are orthogonal. This means especially that applying G to a vector leaves the 2-norm unchanged. When speaking about the action of a rotation, we denote the rows/columns affected when forming the rotation-matrix product. As the action is essential in the development of our algorithm, we will graphically represent each rotator by using brackets having tiny arrows pointing to the rows resp. columns affected, e.g.,

$$\begin{array}{l}\Rsh\\ \Lsh\end{array}\begin{bmatrix}\times & \times\\ 0 & \times\end{bmatrix} = \begin{bmatrix}\times & \times\\ \times & \times\end{bmatrix}.$$

A special case, necessary for our algorithm, is the application of a *descending series* of rotations to e_k, the kth column of the identity matrix, e.g.,

$$\begin{array}{lll}\Rsh & & \\ \Lsh & \Rsh & \\ & \Lsh & \Rsh \\ & & \Lsh\end{array}\begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \begin{array}{ll}\Rsh & \\ \Lsh & \Rsh \\ & \Lsh \\ & \end{array}\begin{bmatrix}0\\0\\s_1\\\overline{c_1}\end{bmatrix} = \begin{bmatrix}s_1s_2s_3\\ s_1s_2\overline{c_3}\\ s_1\overline{c_2}\\ \overline{c_1}\end{bmatrix}.$$

Since $s_i \leq 1$ and $c_i \leq 1$, we have $s_1s_2s_3 \leq 1$ and so on. Clearly the 2-norm is untouched and remains 1, but whereas the weight on the left is concentrated in the trailing element, after applying the rotations, it is smeared out, and distributed over several components.

In this article, as mentioned before, we will nearly always operate on the QR factorization, and in particular on the factorization in rotations of the matrix Q. The role of the upper triangular matrix R is inferior, as we can transfer from the left to the right through the upper triangular matrix without destroying its upper triangularity. More precisely, if we apply a rotation acting on neighbored rows from the left to an upper triangular, then we destroy the sparsity pattern by introducing a non-zero on the sub-diagonal. We can always restore the upper triangular structure by applying a rotation on the corresponding columns (the elements marked with a tilde are the only ones affected by the *transfer* operation):

$$\begin{array}{l} \lceil\!\!\rightarrow \\ \lfloor\!\!\rightarrow \end{array} \quad \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \tilde{\times} & \tilde{\times} & \tilde{\times} \\ 0 & \tilde{\times} & \tilde{\times} & \tilde{\times} \\ 0 & 0 & 0 & \times \end{bmatrix} = \begin{bmatrix} \times & \tilde{\times} & \tilde{\times} & \times \\ 0 & \tilde{\times} & \tilde{\times} & \tilde{\times} \\ 0 & 0 & \tilde{\times} & \tilde{\times} \\ 0 & 0 & 0 & \times \end{bmatrix} \quad \begin{array}{l} \lceil\!\!\rightarrow \\ \lfloor\!\!\rightarrow \end{array} .$$

Of course, one can transfer them from right to left as well. Moreover, let Q be a matrix factored in 2×2 rotations, obeying a particular pattern. Transferring one rotator after the other through the upper triangular matrix shows that the rotational pattern remains unaffected. So, a matrix A having an RQ-factorization $A = \hat{R}\hat{Q}$ admits a QR factorization, where the rotational factorization of Q and $\hat{Q}$ obey the same pattern.

2.2. Non-periodic extended Krylov spaces. Generically an extended Krylov subspace is of the following form

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

When building such a space, one adds vectors to the extended space one by one. For generality, we allow arbitrary successions of multiplications with A and A^{-1}. To track this non-periodic ordering and adding of the vectors a selection vector s is associated to the extended Krylov space. This *selection vector* selects the vectors taken out of the bilateral sequence

$$\ldots, A^m v, A^{m-1}v, \ldots, A^2v, A^1v, v, A^{-1}v, A^{-2}v, \ldots, A^{-\overline{m}+1}v, A^{-\overline{m}}v, \ldots \tag{2.2}$$

to be included next in the extended Krylov space. The first vector of the extended space is always v; the second vector is Av if $s_1 = \ell$, or $A^{-1}v$ for $s_1 = r$. The ith successive vector in the Krylov space is taken left whenever $s_{i-1} = \ell$ or right if $s_{i-1} = r$ out of the bilateral sequence, and next to the last picked vector on that side. An alternative notation to $\mathcal{K}_{m,\overline{m}}(A,v)$ is thus $\mathcal{K}_{s,k}(A,v)$ where $k = m + \overline{m} + 1$ stands for the number of vectors taken out of (2.2) to generate the extended Krylov space. The number of times ℓ appears in the first $k-1$ components of s equals m, and $\overline{m}$ the amount of occurrences of r. The selection vector characterizes thus uniquely the extended Krylov space.

EXAMPLE 2.2. For example, a classical Krylov space's selection vector has only values ℓ. The selection vector accompanying a pure (only inverse powers involved) extended Krylov space only comprises values r. The alternating occurrence of ℓ's and r's leads to an extended Krylov space of the form

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

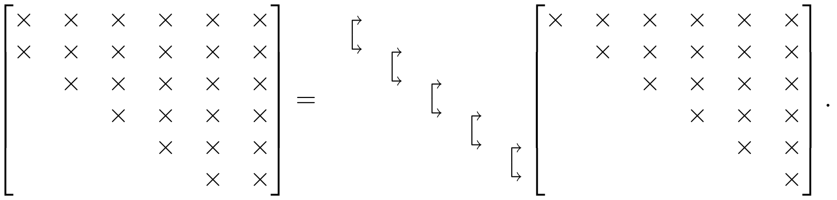
which, for unitary matrices, links closely to CMV-matrices [30], see also Example 2.5.

In our setting there is no particular reason to restrict to periodic vector successions, e.g., $s = \left[r\,\ell\,r\,r\,r\,\ell\,r\,\ldots\right]$ corresponds to

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

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

Let us first consider a Hessenberg matrix. It is well known that its QR decomposition can be written as a descending series of rotations times an upper triangular matrix, e.g.,



The unitary matrix is decomposed into $n-1$ rotations according to a *position vector* $p = [\ell\ell\ell\ell\ell]$, who captures the mutual positioning of successive rotations. An entry $p_i = \ell$ signifies that rotation G_i is positioned to the left of rotation G_{i+1}, whereas $p_i = r$ indicates that G_i is positioned to the right of G_{i+1}.

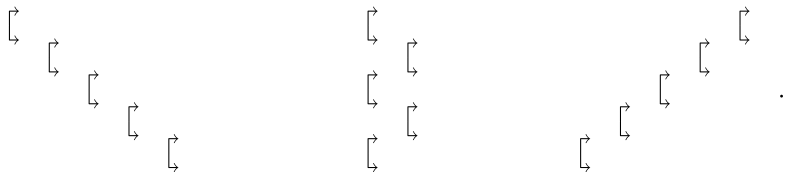
When going from classical Krylov spaces to extended Krylov spaces, there is a price to pay: we cannot guarantee the projected counterpart to remain of Hessenberg form. Nevertheless these matrices, let's name them *extended Hessenberg* matrices, share major properties with the classical Hessenberg matrix when comparing their QR factorizations. Each extended Hessenberg matrix admits a QR factorization with Q factored in $n-1$ 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 rotators 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 neighbored rows, then we will call A an* extended *Hessenberg matrix.*

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

Typically not being unreduced does not pose any problems as this means we have found an invariant subspace.

EXAMPLE 2.4. Equation (2.3) displays the pattern showing up in the QR factorization of a Hessenberg (left), a CMV matrix (middle), and an inverse Hessenberg matrix (right). Only the rotational pattern of the Q factor is depicted.



(2.3)

In [34,35] the link between these extended Hessenberg matrices and the extended Krylov spaces is examined. The position and selection vector nicely tie both concepts together: they are identical. Therefore, from now on we will stick to the selection vector for both concepts. Summarizing, consider an extended Krylov space determined by its selection vector s. Let V be an orthogonal basis for this extended space such that

$$\operatorname{span}\{V_{:,1:k}\} = \mathcal{K}_{s,k}(A, v) \quad \forall k \leq n, \tag{2.4}$$

then the matrix V^*AV will be of extended Hessenberg form. More precisely, the Q factor in the QR decomposition of V^*AV admits a decomposition in $m-1$ 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 2.2 and 2.4. Classical Krylov subspaces identify with a selection vector of only ℓ's and hence a descending sequence of rotators as in the left of (2.3). It is not hard to see that a classical Krylov space, generated for A^{-1}, results in a projected counterpart $V^*A^{-1}V$ being of Hessenberg form, obviously 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 to the CMV decomposition.

3. Implicit Q-theorem for the extended case.

The relations between the position and selection vector proposed in the previous section are quite strict, as shown in the next theorem.

THEOREM 3.1 (From [34, 35]). *Let A be a non-singular matrix and s a selection vector. Let V and $\hat{V}$ be two unitary matrices sharing the first column, i.e. $Ve_1 = \hat{V}e_1$, having both projected counterparts QR factored as*

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

If Q and $\hat{Q}$ can be written as a series of rotations following the ordering imposed by s and all these rotations differ from the identity[1]*, then the matrices H and $\hat{H}$ are essentially the same.*

Theorem 3.1 is an extension of the so called implicit Q-theorem, initially only formulated for Hessenberg matrices, stating that once the matrix structure –determined by the selection vector– and first vector Ve_1 are fixed, everything is implicitly defined. For our purpose, this theorem is not general enough. We require some essential uniqueness of the projected counterparts, which are, of a strictly smaller dimension. Moreover, the associated selection vector need only be defined for the first k components. Let us generalize this. First we reformulate Theorem 3.1 to deal also with reducible matrices.

THEOREM 3.2. *Let A be a non-singular matrix and s a selection vector. Let V and $\hat{V}$ be two unitary matrices sharing the first column, i.e. $Ve_1 = \hat{V}e_1$, having both projected counterparts QR factored as in (3.1), following the ordering imposed by s. Denote the individual rotations appearing in the factorization of Q and $\hat{Q}$ as G_i^Q and $G_i^{\hat{Q}}$ respectively, where i and $i+1$. Define $\hat{k}$ as the minimum i for which either G_i^Q or $G_i^{\hat{Q}}$ equals the identity, i.e.,*

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

if none 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 $\hat{V}$.

We postpone the proof of Theorem 3.2 until Theorem 3.5.

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

Whereas Theorem 3.2 again states something only related to a full projection, i.e., square matrices V and $\hat{V}$, we are typically confronted with rectangular V. Obviously the conclusions are not the same as the following example illustrates.

EXAMPLE 3.4. Take a 5×5 matrix $A = \operatorname{diag}(1,2,3,4,5)$ and initial vector $v = [1,1,1,1,1]^T$. Consider two Krylov spaces not of full dimension: $\mathcal{K} = \operatorname{span}\left\{v, Av, A^2v\right\}$

[1] Otherwise there is welcome breakdown, since we have found an invariant subspace containing v.

and $\hat{\mathcal{K}} = \operatorname{span}\left\{v, Av, A^{-1}v\right\}$. The associated orthogonal matrices V and $\hat{V}$ are

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

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

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

Obviously both H and $\hat{H}$ admit an identical shape 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 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 compulsory 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 matrix and s and $\hat{s}$ be two selection vectors. Let $\underline{V}$ and $\underline{\hat{V}}$ be two $n \times (k+1)$ 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 $n \times k$ submatrices of $\underline{V}$ and $\underline{\hat{V}}$ respectively. Consider*

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

where Q and $\hat{Q}$ are decomposed into a series of rotations ordered as imposed by s and $\hat{s}$.

Then define $\hat{k}$ as follows

$$\hat{k} = \min_i \left\{ 1 \leq i \leq n-2 \text{ such that, } G_i^Q = I, G_i^{\hat{Q}} = I, \text{ or } s_{i-1} \neq \hat{s}_{i-1} \right\}, \tag{3.3}$$

if none such $\hat{k}$ exists, set it equal to $n-1$.[2]

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.

To prove Theorem 3.5 a reformulation of Theorem 3.7 from [35] is required. With e_i the ith standard basis vector is denoted and I_k is the identity matrix of size $k \times k$. A non-calligraphic K represents the *Krylov* matrix having as columns the vectors iteratively constructed for generating the associated Krylov space.

THEOREM 3.6. *Let H be an $n \times n$ matrix, with $HP_{\hat{k}}$ where $P_{\hat{k}} = [I_{\hat{k}}, 0]^T \in \mathbb{R}^{n \times \hat{k}}$ of (rectangular) extended Hessenberg form. Assume that the unitary matrix Q, where $QR = HP_{\hat{k}}$, has its first $\hat{k}$ rotations in its decomposition ordered according to the selection vector s, which is a $(\hat{k}-1)$-tuple.*

[2] The case $\hat{k} = n$ requires a reformulation of (3.2), and is therefore excluded. One can fall back on Theorem 3.2.

Then we have that $S_j = K_{s,j}(H, e_1)$*, for* $1 \leq j \leq \hat{k}$*, is upper triangular.*

The proof is not repeated as the inductive procedure of [35] can be used here as well. The clue is the necessity of having element s_j available to make a statement for the $(j+1)$st subspace and to have nontrivial rotations as well.

Proof. Let us now prove 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 $K_{s,j}(H_e, e_1) P_{\hat{k}} = S_j$ and $K_{\hat{s},j} P_{\hat{k}} = \hat{S}_j$, both upper triangular for $1 \leq j \leq \hat{k}$. Elementary computations provide us

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

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 in the outer left and outer right factorizations provide us the essential equality of the first $\hat{k}$ vectors of V and $\hat{V}$. The rest follows trivially.
$\square$

4. Implicit Extended Krylov Subspace Algorithm.

Building an extended Krylov subspace typically requires solutions of some linear systems. For large matrices these solutions are sometimes computed using iterative solvers based on Krylov subspaces, or block Krylov subspaces if one wishes to solve several systems at once. In this section we will compute an approximation to an extended Krylov subspace without explicit system solves.

Let us explain our algorithm (see Algorithm 1 for a pseudo-code version) with an example having selection vector $s = [\ell\, r \ldots]$. First, we have to choose an oversampling parameter p. This parameter determines how many vectors we additionally put into the standard Krylov subspace before we start the shrinking procedure. A large p makes the whole algorithm more expensive but also the approximation to the extended Krylov subspace more accurate. Then, we compute the Krylov subspace $\mathcal{K}_{\tilde{k}}(A, v)$ with dimension $\tilde{k} = |s| + 1 + p$, where $|s|$ denotes the length of the vector s. Let V be an orthogonal matrix forming a basis of $\mathcal{K}_{\tilde{k}}(A, v)$, satisfying (2.1). The matrix $H = V^* A V$ is upper Hessenberg. We have

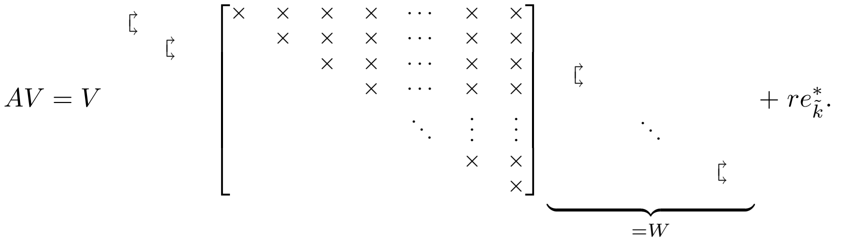
$$AV = VH + r e_{\tilde{k}}^*.$$

In the next step the QR decomposition of $H = QR$ using a series of rotations is computed:

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

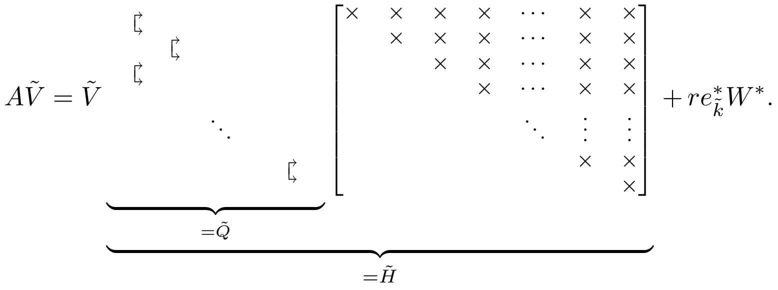
Probably there are much more economical manners of retrieving the QR factorization of H, by storing, e.g., H directly in factored form and updating the factors as in the SYMMLQ case [23]. This is, however, beyond the goal of the current paper.

We will now transform H via unitary similarity transformations into the desired shape, linked to the extended Krylov subspace having selection vector $s = [\ell\, r \ldots]$. The first rotation will always remain unaltered, since we require to keep V's first column untouched. The first entry in s is an ℓ entailing the second rotation to be on the right-hand side of the first one. Since this is already the case we do not have to do anything. The next entry is an r meaning the third rotation must be brought to the other side. To this end, we first transfer all the rotations starting from the third one through the upper triangular R:



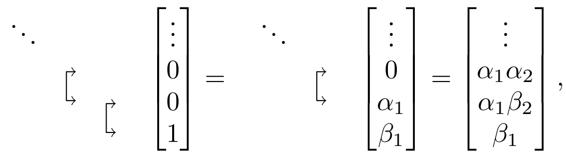
Whenever the matrix H is highly structured, e.g., tridiagonal, the QR decomposition partially destroys the existing structure. Typically sort of structure transfer takes place and a new, exploitable, structure emerges. We do not want to defer too much from the core message of the current article and as such do inspect this in detail.

After multiplying with W^* from the right-hand side we set $\tilde{V} = VW^*$ and get



We 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 kicking off with $[\ell\, r]$. Next we go through all the other entries in s. If the entry in s is r we transfer the trailing rotations to the right and bring them back to the left. If the next entry is ℓ we do nothing. In each step we fix thus one more rotation in the correct position. We repeat this until the end of s is reached to get $\tilde{H}$ in the desired shape.

Now we have an approximation to the extended Krylov subspace with too many vectors. So the last step is to chop off all except the first $|s| + 1$ columns of V and the upper $(|s| + 1) \times (|s| + 1)$ block of H. This chop off is necessary since we have smeared out the residual from the nice shape $re_{\tilde{k}}^*$ to $re_{\tilde{k}}^* W^*$. The key vector is $We_{\tilde{k}}$, having the following form



$\begin{bmatrix} \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix} =$
$\begin{bmatrix} \vdots \\ 0 \\ \alpha_1 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \vdots \\ \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1 \end{bmatrix},$

with $\alpha_1, \alpha_2, \beta_1\beta_2 \leq 1$. The product $\prod_j \alpha_j$ is expected to be smaller than one and is hopefully decaying to zero fast. This of course depends on the properties of H, A, and $\mathcal{K}_{\tilde{k}}(A, v)$.

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., $s = [\ell\, r\, \ell\, r\, \ldots]$, oversampling parameter $p$
Output: $H$, $V$ with $AV = VH + V_{:,k+1}e_k^* + \varrho h^* \approx VH + V_{:,k+1}e_k^*$
$\tilde{k} := |s| + 1 + p$; $k := |s| + 1$;
Compute $V$ spanning the standard Krylov subspace $\mathcal{K}_{\tilde{k}}(A, v)$, $H := V^*AV$, and
    $\varrho := (AV - VH)e_{\tilde{k}}$, with $AV = VH + re_{\tilde{k}}^*$ and $e_{\tilde{k}} = I_{:,1:\tilde{k}}$;
$h := e_{\tilde{k}}$;
Compute the QR-factorization of $QR = H$ into $\tilde{k} - 1$ rotations $G_1 G_2 \ldots G_{\tilde{k}-1} := Q$
    and an upper triangular $R$;
for $j = 1, \ldots, |s|$ do
    if $s(j) == r$ then
        Compute the $RQ$-factorization of $R \prod_{i=j+1}^{\tilde{k}-1} G_i := \prod_{i=j+1}^{\tilde{k}-1} G_i R$;
        $V := V \prod_{i=\tilde{k}-1}^{j+1} G_i^*$;
        $h := \prod_{i=\tilde{k}-1}^{j+1} G_i h$;
    end
end
if $\|\varrho\|_2 \|h_{1:k}\|_2$ is small enough then
    $V := V_{:,1:k}$, $H := H_{1:k,1:k}$;
    return $V$ and $H$;
else
    Choose a larger $p$ and start again;
end
```

The hope is that at least the first $|s| + 1$ entries of $(e_{\tilde{k}}^* W^*)_{1:|s|+1}$ are negligibly small. We will show in Section 5 that this is the case if $A^{-1}v$ has a good approximation within the space spanned by V. If this is the case, the residual $A\tilde{V}_{:,1:|s|+1} - \tilde{V}_{:,1:|s|+1}\tilde{H}_{1:|s|+1,1:|s|+1}$ is dominated by the perturbation $\tilde{V}_{:,|s|+2}\tilde{H}_{|s|+2,1:|s|+1}$. The matrix $\tilde{H}_{1:|s|,1:|s|}$ is an extended Hessenberg matrix, whose QR decomposition admits a factorization of Q into $|s|-1$ rotations ordered as in s.

COROLLARY 4.1. *Having computed $\tilde{V}$ and $\tilde{H}$ as described above, assume the matrix $r(e_{\tilde{k}}^* W^*)_{1:|s|+1}$ is zero, and none of the rotations in $\tilde{Q}$ is 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 $\mathcal{K}_s(A, v)$ 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. □

5. Error bounds.

In this section we will show that our algorithm computes a good approximation to the extended Krylov subspace if $A^{-1}v$ has a good approximation in the large Krylov subspace $\mathcal{K}_{\tilde{k}}$. Therefore, we will first construct a matrix $\tilde{A}$ for which our algorithm computes the exact extended Krylov subspace. From Corollary 4.1 we know that Algorithm 1 computes the exact solution if the residual $\|r\|$ is zero. Let us consider the matrix

$$\tilde{A} = A - rv_{\tilde{k}}^*.$$

Obviously we have $\tilde{A}v_i = Av_i, \forall i < \tilde{k}$, since the columns of V are orthonormal. That means that up to size $\tilde{k}$ the Krylov subspaces $\mathcal{K}_{\tilde{k}}(A, v)$ and $\mathcal{K}_{\tilde{k}}(\tilde{A}, v)$ are identical. Further we have

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

and thus $\tilde{A}V = VH$. Hence $\tilde{A}$ is the sought matrix for which our algorithm computes the exact extended Krylov subspace. Unfortunately, the norm $\|\tilde{A} - A\|_2$ is too large, even if the algorithm produces a good approximation, hence so the norm-wise difference is not a good error measure.

We now 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 that in fact our algorithm computes $\mathcal{K}_{s,k}(\tilde{A}, v)$ instead of $\mathcal{K}_{s,k}(A, v)$. The Krylov subspaces $\mathcal{K}_{s,k}(A, v)$ and $\mathcal{K}_{s,k}(\tilde{A}, v)$ are spanned by the joined vectors $v, Av, A^2v, \ldots, A^{k-2}v$ and by $A^{-1}v$ respectively $\tilde{A}^{-1}v$. Hence the norm $\|A^{-1}v - \tilde{A}^{-1}v\|_2$ is a measure for the accuracy of the computed extended Krylov space approximation. In the next lemma we will link this norm to the approximation accuracy of $A^{-1}v$ in subspace $\mathcal{K}_{\tilde{k}} = \operatorname{span}\{V\}$. This accuracy can be measured by $\left\|(I - VV^*)A^{-1}v\right\|$.

LEMMA 5.1. *Let A and $\tilde{A}$ be defined as above. Let V be the matrix of orthonormal columns spanning $\mathcal{K}_{\tilde{k}}(A, v) = \mathcal{K}_{\tilde{k}}(\tilde{A}, v)$ and $\gamma = \|VV^*A(I - VV^*)\|_2$ and let $H = V^*AV$ be invertible. Then we have that*

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

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

$$\begin{aligned}\left\|A^{-1}v - \tilde{A}^{-1}v\right\| &\leq \left\|(I - VV^*)A^{-1}v\right\| + \left\|VV^*A^{-1}v - \tilde{A}^{-1}v\right\| \\ &\leq \left\|(I - VV^*)A^{-1}v\right\| + \left\|\tilde{A}^{-1}\tilde{A}VV^*A^{-1}v - \tilde{A}^{-1}v\right\| \\ &\leq \left\|(I - VV^*)A^{-1}v\right\| + \left\|\tilde{A}^{-1}VV^*AVV^*A^{-1}v - \tilde{A}^{-1}v\right\|. \qquad (5.1)\end{aligned}$$

The projection of v on V is again v, hence we have $v = VV^*v$. As VV^* is a projection we have $VV^* = VV^*VV^*$. Using the sub-multiplicativity of the 2-norm we can bound the second norm of (5.1) by

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

Further we have

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

The proof is completed by the following bound from [28, Proposition 2.1]

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

□

The 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 $\mathcal{K}_{\tilde{k}}$.

6. Numerical Experiments. In this section we will first compare the accuracy of our novel approach with the examples from [14] where explicit matrix inversions are used to approximate matrix functions. In a second example, taken from [16] and related to the approximate solution of Lyapunov equations, we illustrate the gain in compression with the implicit approach. Thirdly, 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 the approximation of (some) matrix functions as the following numerical experiments will show. For our numerical experiments we use and compare with the examples from Jagels and Reichel in [14]. Four different selection vectors are used: with no r's, with an r every second entry, every third, and every forth entry. In this section the variable k, determining the chop off, is always taken $|s|+1$. The computations are performed in MATLAB. The main idea of these examples is to show that we can do equally well without explicit inversions as in [14], 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 $\tilde{H} = \tilde{V}^* A \tilde{V}$. So $f(A)v$ can be approximated by

$$f(A)v \approx V f(\tilde{H}) \tilde{V}^* v = V f(\tilde{H}) e_1 \left\|v\right\|_2 .$$

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

In the first example we are able to reproduce the figures from [14], meaning that the implicit approach performs equally well as the explicit one.

EXAMPLE 6.1. The first example is a symmetric positive definite Toeplitz matrix A, having

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

We use a 1000×1000 matrix as in [14, Example 5.1-2]. Figure 6.1 and 6.2 show the relative error of the approximation of $f(A)v$ for different selection vectors. In Figure 6.1 $f(x) = \exp(-x)/x$, and in Figure 6.2 $f(x) = \log(x)$. The vector v has normally distributed random entries with mean zero and variance one. It is known that both functions can be approximated by an extended Krylov subspace well. We choose the oversampling parameter $p = 100$ and observe that $A^{-1}v$ has a very good approximation within $\mathcal{K}_{100}(A, v)$. An almost identical behavior as in [14, Fig. 5.1/5.2] is reported.

In the next examples, agreeing with the 4th and 5th one form [14], the results are less good, but still reasonable approximations are retrieved.

EXAMPLE 6.2. In this example the matrix A is the discretization of the operator $L(u) = \frac{1}{10}u_{xx} - 100u_{yy}$ on the unit square. For the discretization in each direction a three point stencil with 40 equal distributed interior points has been used. Together with a homogeneous boundary condition this yields a 1600×1600 symmetric positive matrix A. The initial vector v is chosen to be $v_j = 1/\sqrt{40}, \forall j$. Figure 6.3 shows the relative approximation error for $f(x) = \exp(-x)/x$ and Figure 6.4 for $f(x) = 1/\sqrt{x}$.

We notice that the oversampling parameter $p = 100$ is not large enough, as the subspace $\mathcal{K}_{\tilde{k}}$, depicted by the upper green line in Figure 6.3 is not approximating $A^{-1}v$ nor $f(A)v$ up

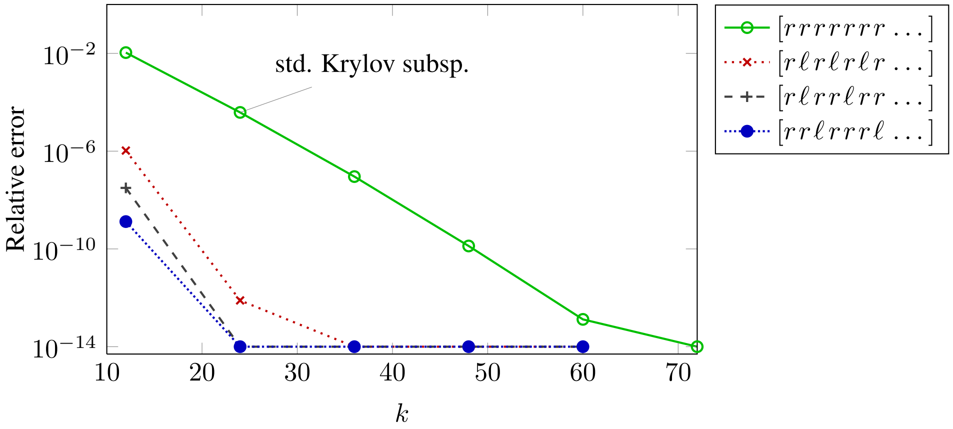


Fig. 6.1: Relative error in approximating $f(x) = \exp(-x)/x$ for various selection vectors s and $k = 12, 24, 36, 48, 60$.

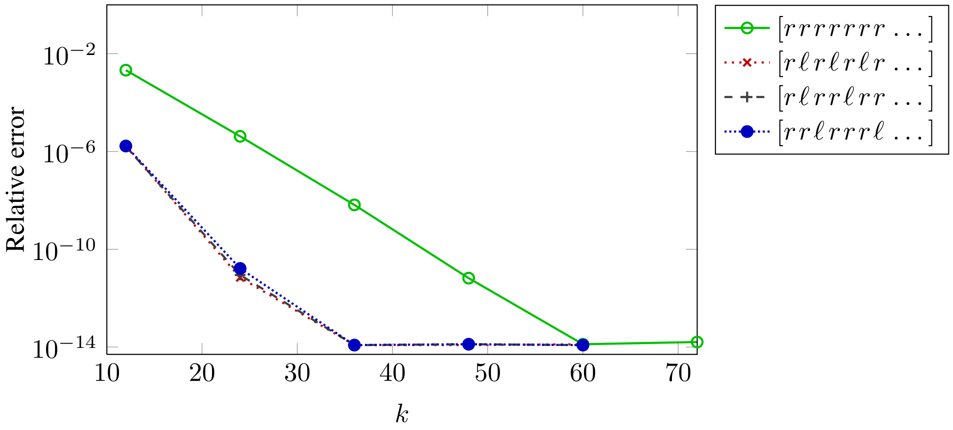


Fig. 6.2: Relative error in approximating $f(x) = \log(x)$ for various selection vectors s and $k = 12, 24, 36, 48, 60$.

to a satisfactory accuracy. After the truncation (for $p = 100$) we arrive at the middle lines revealing an almost identical accuracy for the extended space as for the large untruncated Krylov space (depicted again by the green line containing, however, p additional vectors). We are thus able to reduce the standard Krylov subspace of dimension 112 to an approximation of an extended Krylov subspace with only 12 vectors, while retaining an almost identical relative error for $f(x) = \exp(-x)/x$, which is more than 3 orders smaller than the error for a standard Krylov subspace of dimension 12.

Next a large oversampling parameter of 200 is tested and we notify a reduction of dimension 212 to an extended Krylov subspace containing only 12 vectors, with again a comparable error and, once more an approximation of 6 orders better than the classical Krylov space.

For $f(x) = 1/\sqrt{x}$ we observe almost the same when reducing a space of dimension 136 resp. 236 to an extended Krylov subspace of dimension 36 with selection vector $[r\,\ell\,r\,\ell\,r\,\ell\,\ldots]$.

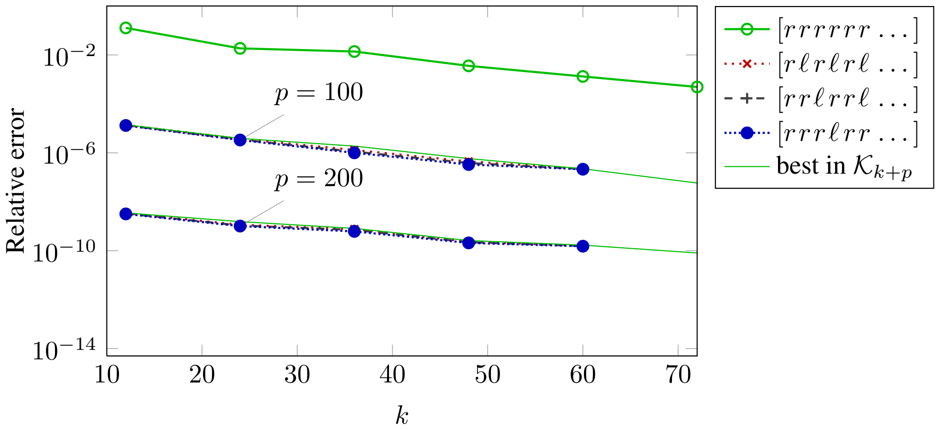


Fig. 6.3: Relative error in approximating $f(x) = \exp(-x)/x$ for various selection vectors s and $k = 12, 24, 36, 48, 60$.

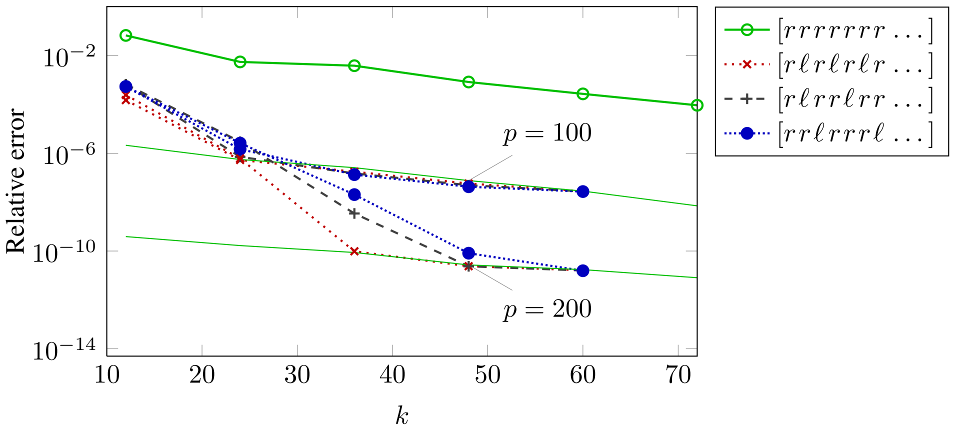


Fig. 6.4: Relative error in approximating $f(x) = 1/\sqrt{x}$ for various selection vectors s and $k = 12, 24, 36, 48, 60$.

The third example uses a matrix A for which $A^{-1}v$ does not lie in the standard Krylov subspace. So our algorithm is expected to fail here.

EXAMPLE 6.3. In this example we use 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 zero entries except for a 1 in the lower left corner. For the computation of Figure 6.5, we approximate again $f(A)v$ with $f(x) = \exp(-x)/x$ and the random v from Example 6.1. We see that we perform equally bad as the standard Krylov space.

In [14], the extended Krylov subspace was successful in the approximation of $f(A)v$, but an explicit solve with the MATLAB function backslash has been used. Such a solver is typically not available for large matrices and, moreover in practice often another iterative solver is used to approximate the inverse multiplication with A, which would lead to similar problems as observed here.

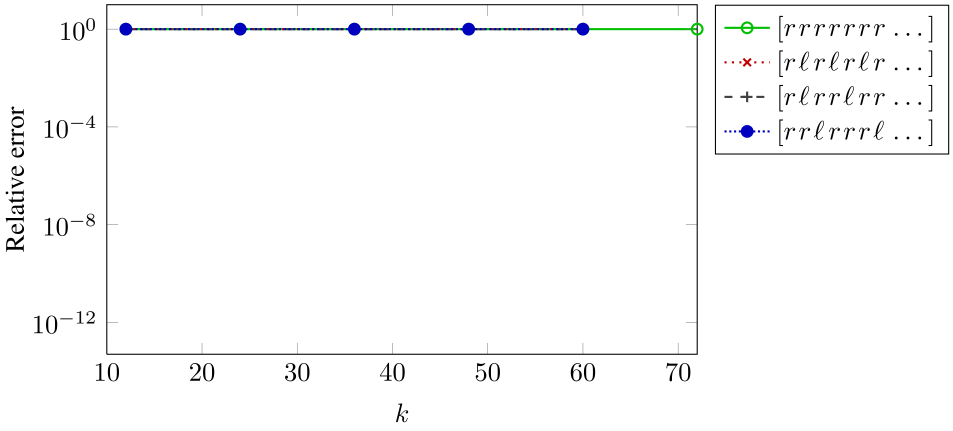


Fig. 6.5: Relative error in approximating $f(x) = 1/\sqrt{x}$ for various selection vectors s and $k = 12, 24, 36, 48, 60$.

The last example in this section shows that one can also use implicit extended Krylov subspaces for the solution of Lyapunov equations

EXAMPLE 6.4. We use the Example 4.2 from [16]. The matrix $A \in \mathbb{R}^{5000\times 5000}$ is a diagonal matrix having eigenvalues

$$\lambda = 5.05 + 4.95\cos(\theta), \qquad \theta \in [0, 2\pi].$$

We solve the Lyapunov equation

$$AX + XA^{*} + BB^{*} = 0,$$

with B a vector with normally distributed entries with variance one and mean zero. The results in Figure 6.6 show the relative 2-norm difference of the approximation $\tilde{X}$ computed via

$$\tilde{X} = V_{1:\tilde{k}} Y V_{1:\tilde{k}}^{*}, \text{ where } Y \text{ is the solution of } \tilde{H}Y + Y\tilde{H} + (V_{1:r}^{*}B)(V_{1:r}^{*}B)^{*} = 0 \qquad (6.1)$$

and the exact solution computed with the MATLAB function `lyapchol`. We have chosen an oversampling parameter $p = 50$. Compared to the standard Krylov subspace we can reduce the dimension of the small Lyapunov equation in (6.1) by 50 – 65% for achieving the same accuracy.

6.2. Ritz values. In the next three examples we would like to highlight the fact that our algorithm commences with the information from the Krylov subspace and then commences to squeeze the available 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.

EXAMPLE 6.5. We choose a simple diagonal matrix of size 200×200, with equal distributed eigenvalues between 0 and 2, and a uniform starting vector consisting solely of 1's. For this matrix we first compute the standard Krylov subspace of dimension 180. In

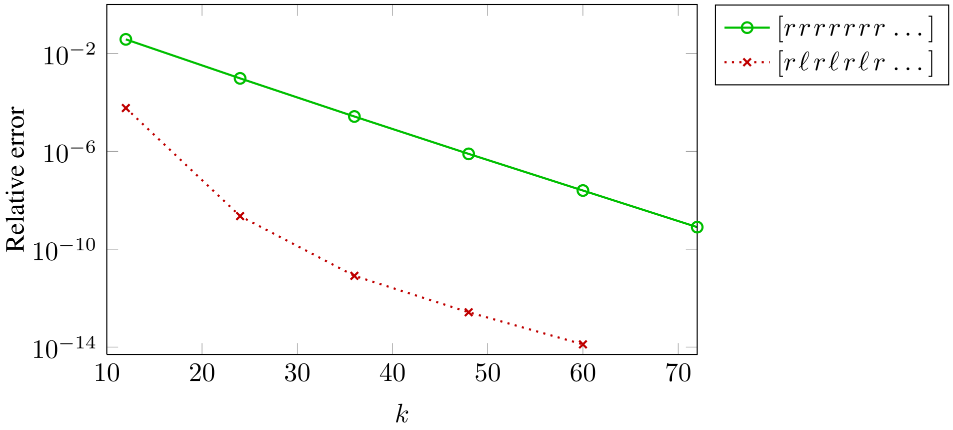


Fig. 6.6: Relative error in approximate solutions of $AX + XA + BB^* = 0$ for $k = 12, 24, 36, 48, 60$.

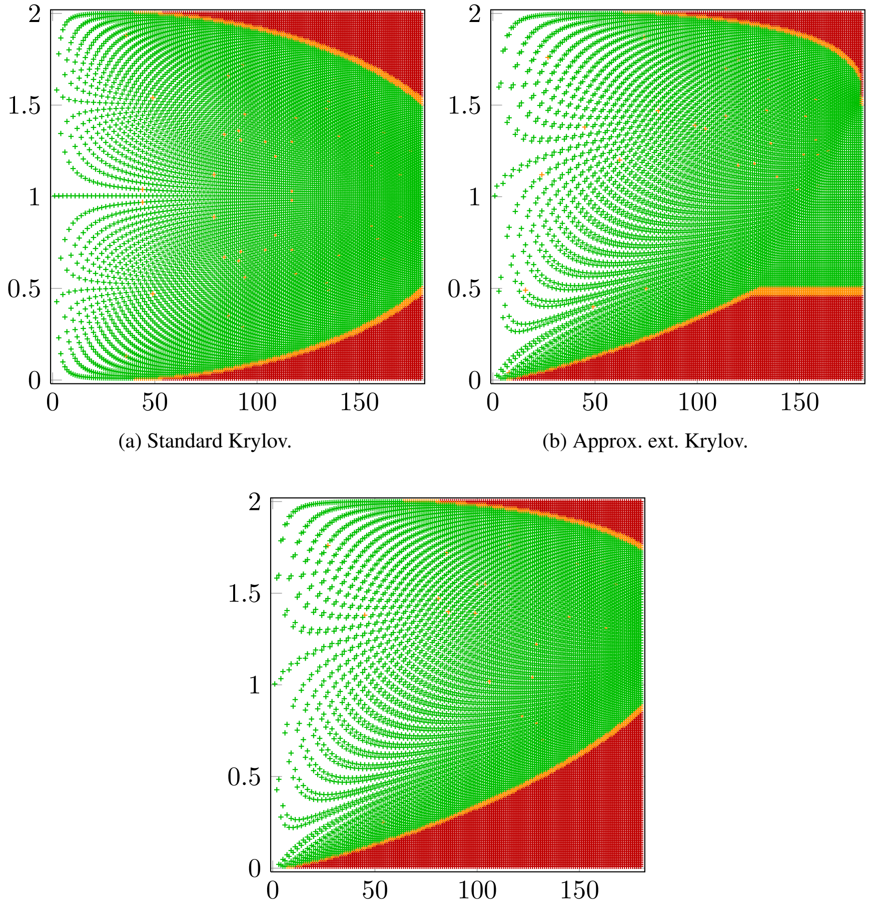
Figure 6.7(a) the Ritz values for standard Krylov subspaces up to dimension 180 are plotted in each iteration step (which equals the number of Krylov vectors generated so far) shown on the x-axis. Red crosses reveal Ritz values approximating eigenvalues quite well, having absolute error smaller than $1\,\text{e}{-7.5}$; yellow crosses are good approximations, with errors between $1\,\text{e}{-7.5}$ and $1\,\text{e}{-5}$; the green markers okay approximations, i.e., errors between $1\,\text{e}{-5}$ and $1\,\text{e}{-2.5}$; and the blue ones the remaining Ritz values. Classical Ritz value convergence is observed, where first the extreme eigenvalues are found.

The next plot, Figure 6.7(b), shows the Ritz values obtained after our truncation algorithm is applied to approximate an extended Krylov subspace, in this case the selection vector contains alternatingly ℓ'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 cannot be better than the final column of the first plot: Figure 6.7(a).

Furthermore, the algorithm is also unable to do better than the third plot, Figure 6.7(c), 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. 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 on 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 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. One sees clearly now how the truncation algorithm tries hard to match the extended space, but is strongly limited by the initial 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 $-1/2$ and $1/2$. We see that traditional Krylov, again



(a) Standard Krylov.

(b) Approx. ext. Krylov.

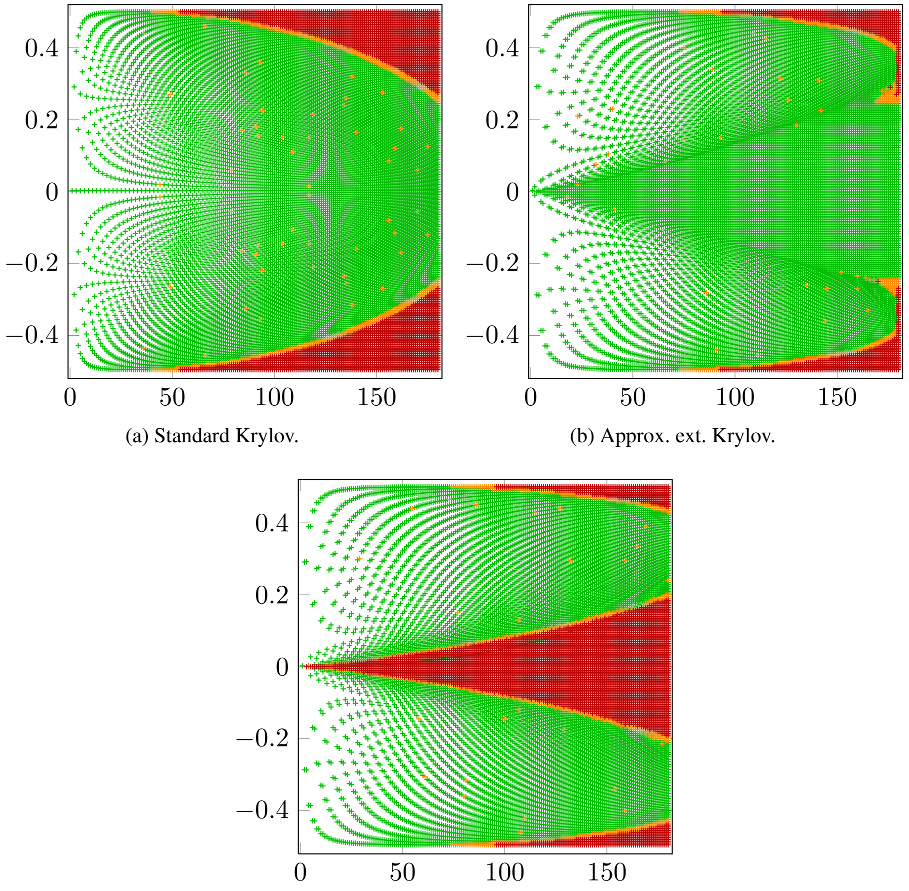
(c) Extended Krylov.

Fig. 6.7: Ritz plots for equal spaced eigenvalues in $[0, 2]$.

first locates the outer eigenvalues (Figure 6.8(a)). Extended Lanczos on the other hand (Figure 6.8(c)), 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 6.8(b) 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 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, 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, but now with the eigenvalues according to the distribution (see Figure 6.9)

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



(a) Standard Krylov.

(b) Approx. ext. Krylov.

(c) Extended Krylov.

Fig. 6.8: Ritz plots for equal spaced eigenvalues in $[-.5, .5]$.

where $\alpha = -3/4$, as in [18]. The distribution shows that most of the eigenvalues are situated at the boundaries -1 and 1. Based on potential theory [17, 18] one knows that for this distribution first the inner eigenvalues, located around 0 are discovered first by classical Krylov. This implies that the classical Krylov space will have a similar goal as the extended Krylov approach namely first find the eigenvalues situated around the origin. Like before, the Figures 6.10(a)-6.10(c) 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 will compare the computational efficiency, and the accuracy of the classical approach and the new algorithm w.r.t. matrix function evaluations. Assume a matrix linked to a Krylov space of dimension $|s| + p$ is built and then truncated to an extended space of dimension $|s|$. In practice it is impossible to estimate the time needed for building the Krylov space, as 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. But, even in this case, the pro-

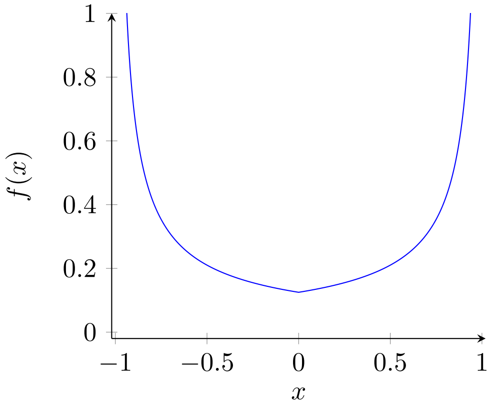


Fig. 6.9: Eigenvalue distribution.

posed method is able to significantly reduce the size of the subspace resulting in equivalently significant memory savings.

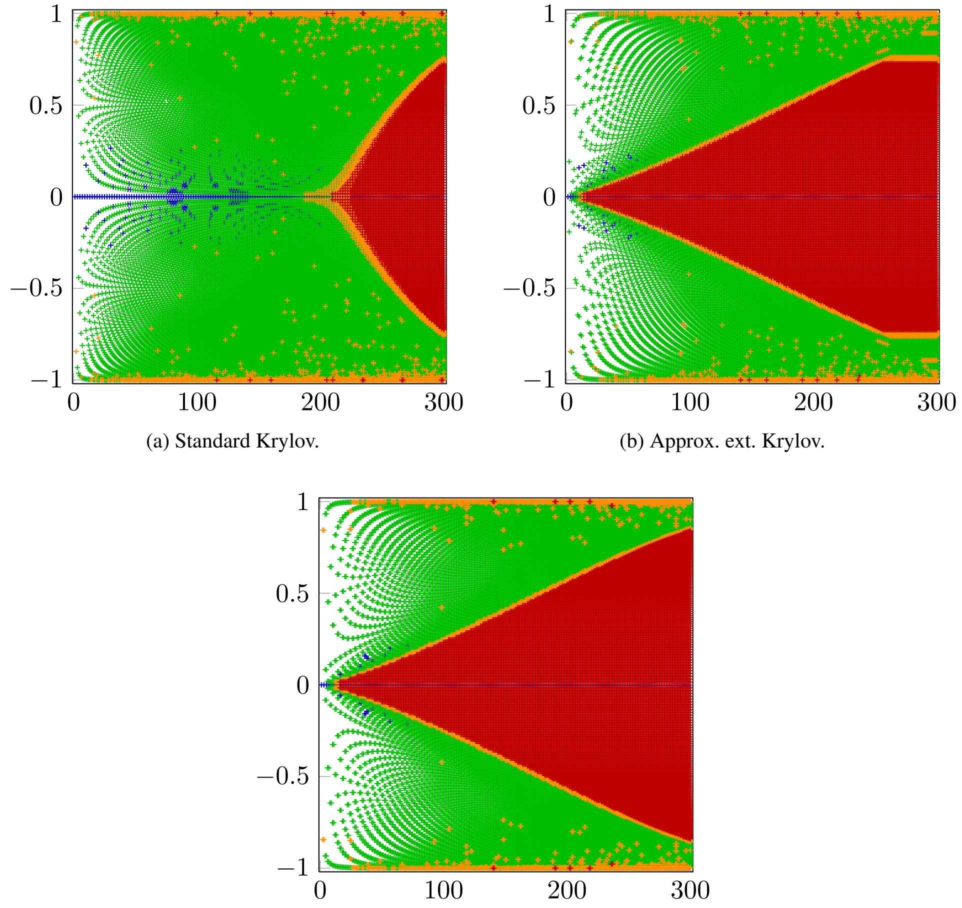
So, for now we forget about the time needed to construct the Krylov space and only investigate the forthcoming computations on the projected counterparts of sizes $|s|$ and $|s|+p$ including the time required for executing the compression. Each parameter ℓ in the selection vector s implicates a transferring of at most $|s|+p$ rotations through an upper triangular matrix. Such a transferring costs $\mathcal{O}(|s|+p)$ flops, as there are at most $|s|$ ℓ's, we have an upper bound of $\mathcal{O}\left(|s|(|s|+p)^2\right)$ to complete the truncation process. Additionally we apply the transferred rotations to V. This costs $\mathcal{O}(n)$, with n the dimension of A, per rotation or $\mathcal{O}\left(n|s|(|s|+p)\right)$ in total. Naturally this is not the end, 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 crossover to arrive at cheaper algorithms is early.

EXAMPLE 6.8. We use again the same operator as in Example 6.2, but we now discretize with 70 equal distributed interior points, so that A is of size 4900×4900. On the dense matrix A the computation of $f(A)v$ based on the optimized MATLAB function `expm` took 18.4 s. Due to the properties of A we need a large oversampling parameter $p=1600$ to achieve good results. On the standard Krylov subspace of dimension 1604 we needed 0.66 s to compute the $f(A)v$ with an relative accuracy of $5.15\,e{-11}$. With our reduction approach we were able to reduce the Krylov subspace to an extended Krylov subspace of dimension 4 ($[\ell\, r\, \ell]$) in 0.59 s. Within this subspace we could 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.[3]

So far we have only an MATLAB implementation for Algorithm 1. This means the part of complexity $\mathcal{O}\left(|s|(|s|+p)\right)$ requiring more time per flop as the approximation of $f(A)v$, which is of complexity $\mathcal{O}\left((|s|+p)^3\right)$ resp. $\mathcal{O}\left(|s|^3\right)$. We assume that a better implementation would increase the advantage of our new approach.

To remove the effect of different degrees of optimization we are going to use a plain flop count plot in the next example.

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



(a) Standard Krylov.

(b) Approx. ext. Krylov.

(c) Extended Krylov.

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

EXAMPLE 6.9. *In this example we only show flop counts. Let A be a matrix of size $n \times n$ with $n = 10,000$. We assume that we again compute $f(A)v$ and to do so use the eigendecomposition of the matrix A or a compressed matrix V^*AV. Let this cost $15N^3$, with N the dimension of A resp. V^*AV. Once we have computed the Krylov subspace of dimension $|s| + p$ (costs about $2n(|s| + p)^2$) we can continue on two different ways. On the one hand we can continue directly computing the eigendecomposition, or on the other hand we can first do the compression to an extended Krylov subspace of dimension $|s|$. This compression requires about $|s|(2n(|s|+p)+2(|s|+p)^2)$. Together we need $15|s|^3+|s|(2Nn+2n^2)$ versus $15(|s| + p)^3$. 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, often computing sufficiently good 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,

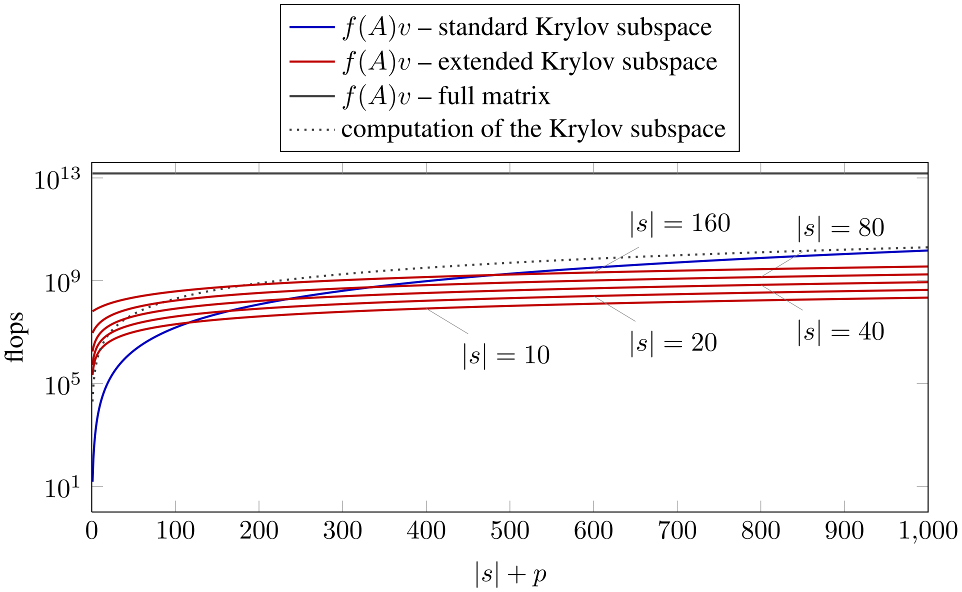


Fig. 6.11: Complexity plot.

however, this constraint was not satisfied, it was shown that the presented approach was able to significantly reduce the size of the Krylov space by bringing it to extended form, without notable loss of accuracy w.r.t. the larger space. A larger compression can have multiple advantages, such as reduced storage costs, and a reduced operation count for forthcoming computations. A final set of numerical experiments illustrated this latter statement revealing a nonneglectable reduction of computational efforts.

This research poses quite some questions. How is this related to implicitly restarted Lanczos [2, 22, 31], 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, size of the initial large Krylov space, size of the truncated part, and so forth.

Acknowledgments. The authors would like to thank Karl Meerbergen for his valuable suggestions on extended Krylov, and Carl Jagels and Lothar Reichel for helping us with the examples from [14] used in the numerical experiments.

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, PA, 2005.

[2] C. Beattie, M. Embree, and D. Sorensen, *Convergence of polynomial restart Krylov methods for eigenvalue computations*, SIAM Review, 47 (2005), pp. 492–515.

[3] B. Beckermann, S. Güttel, and R. Vandebril, *On the convergence of rational Ritz-values*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1740–1774.

[4] A. Bultheel and M. Van Barel, *Linear Algebra, Rational Approximation and Orthogonal Polynomials*, vol. 6 of Studies in computational mathematics, North-Holland, Elsevier Science B.V., Amsterdam, Netherlands, 1997.

[5] A. Chesnokov, K. Deckers, and M. Van Barel, *A numerical solution of the constrained weighted energy problem*, Journal of Computational and Applied Mathematics, 235 (2010), pp. 950–965.

[6] 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.

[7] K. DECKERS AND A. BULTHEEL, *Rational Krylov sequences and orthogonal rational functions*, Tech. Rep. TW499, Katholieke Universiteit Leuven, Departement Computerwetenschappen, 2008.

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

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

[10] G. H. GOLUB AND C. F. V. LOAN, *Matrix computations*, vol. 3, Johns Hopkins University Press, 1996.

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

[12] S. GÜTTEL, *Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection*. submitted.

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

[14] C. JAGELS AND L. REICHEL, *Recursion relations for the extended Krylov subspace method*, Linear Algebra and its Applications, 434 (2011), pp. 1716–1732.

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

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

[17] A. B. J. KUIJLAARS, *Which eigenvalues are found by the Lanczos method?*, SIAM Journal on Matrix Analysis and Applications, 22 (2000), pp. 306–321.

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

[19] R. LEHOUCQ AND K. MEERBERGEN, *Using generalized Cayley transformations within an inexact rational Krylov sequence method*, SIAM Journal on Matrix Analysis and Applications, 20 (1998), pp. 131–148.

[20] K. MEERBERGEN, *Dangers in changing the poles in the rational Lanczos method for the Hermitian eigenvalue problem*, ral-tr-1999-025, Rutherford Appleton Laboratory, Chilton, UK, 1999.

[21] ———, *Changing poles in the rational Lanczos method for the hermitian eigenvalue problem.*, Numerical Linear Algebra with Applications, 8 (2001), pp. 33–52.

[22] R. MORGAN, *On restarting the Arnoldi method for large non-symmetric eigenvalue problems*, Mathematics of Computation, 65 (1996), pp. 1213–1230.

[23] C. PAIGE AND M. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

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

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

[26] ———, *Rational krylov algorithms for nonsymmetric eigenvalue problems, II: Matrix pairs*, Linear Algebra and its Applications, 197/198 (1994), pp. 283–296.

[27] ———, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM Journal on Scientific Computing, 19 (1998), pp. 1535–1551.

[28] Y. SAAD, *Krylov subspace methods for solving large unsymmetric linear systems*, Mathematics of Computation, 37 (1981), pp. 105–126.

[29] ———, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, Pennsylvania, USA, second ed., 2003.

[30] B. SIMON, *CMV matrices: Five years after*, Journal of Computational and Applied Mathematics, 208 (2007), pp. 120–154.

[31] D. C. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 357–385.

[32] 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, Washington, USA, 2002, pp. 167–170.

[33] ———, *Orthogonal rational functions and structured matrices*, SIAM Journal on Matrix Analysis and Applications, 26 (2005), pp. 810–829.

[34] R. VANDEBRIL, *Chasing bulges or rotations? A metamorphosis of the QR-algorithm*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 217–247.

[35] R. VANDEBRIL AND D. S. WATKINS, *A generalization of the multishift QR algorithm*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 759–779.