COMPACT TWO-SIDED KRYLOV METHODS FOR NONLINEAR EIGENVALUE PROBLEMS[*]

PIETER LIETAERT[†], KARL MEERBERGEN[†], AND FRANÇOISE TISSEUR[‡]

Abstract. We describe a generalization of the compact rational Krylov (CORK) methods for polynomial and rational eigenvalue problems that usually, but not necessarily, come from polynomial or rational approximations of genuinely nonlinear eigenvalue problems. CORK is a family of one-sided methods that reformulates the polynomial or rational eigenproblem as a generalized eigenvalue problem. By exploiting the Kronecker structure of the associated pencil, it constructs a right Krylov subspace in compact form and thereby avoids the high memory and orthogonalization costs that are usually associated with linearizations of high degree matrix polynomials. CORK approximates eigenvalues and their corresponding right eigenvectors but is not suitable in its current form for the computation of left eigenvectors. Our generalization of the CORK method is based on a class of Kronecker structured pencils that include as special cases the CORK pencils, the transposes of CORK pencils, and the symmetrically structured linearizations by Robol, Vandebril, and Van Dooren [*SIAM J. Matrix Anal. Appl.*, 38 (2017), pp. 188–216]. This class of structured pencils facilitates the development of a general framework for the computation of both right- and left-sided Krylov subspaces in compact form, and hence allows the development of two-sided compact rational Krylov methods for nonlinear eigenvalue problems. The latter are particularly efficient when the standard inner product is replaced by a cheaper to compute quasi-inner product. We show experimentally that convergence results similar to CORK can be obtained for a certain quasi-inner product.

Key words. nonlinear eigensolver, rational Krylov, nonlinear eigenvalue problem, two-sided Krylov method

AMS subject classifications. 65F15, 65F25, 15A18, 15A22, 41A20, 47J10

DOI. 10.1137/17M1144167

1. Introduction. We consider the computation of selected eigenvalues of the nonlinear eigenvalue problem and their associated right and left eigenvectors, which, for a given matrix-valued function $A : \mathbb{C} \to \mathbb{C}^{n \times n}$ consists of finding triplets (λ, x, y) with $\lambda \in \mathbb{C}$ and nonzero vectors $x, y \in \mathbb{C}^n$ such that

$$(1.1) \qquad A(\lambda)x = 0, \qquad y^* A(\lambda) = 0.$$

Applications are sensitivity analysis of eigenvalues (see [16], [32], and references therein) and eigenvalue optimization [25], [31].

Krylov methods are commonly used for finding eigenvalues of a matrix. The Arnoldi method [30] and the rational Krylov method [29] are one-sided methods, i.e., they build a Krylov space with the matrix. Two-sided methods build two Krylov spaces: one for the matrix and one for its transpose. Two-sided methods are less popular than one-sided methods because of the additional cost for building the left Krylov basis and possible loss of accuracy due to a potentially unstable biorthogonalization

[†]Department of Computer Science, KU Leuven, University of Leuven, 3001 Heverlee, Belgium (pieter.lietaert@cs.kuleuven.be, Karl.Meerbergen@cs.kuleuven.be).

[‡]School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (Francoise. Tisseur@manchester.ac.uk).

process. However, the computation of both right and left eigenvectors requires both Krylov spaces. The prototype two-sided Krylov method is the Lanczos method [4], [18]. The two-sided Arnoldi method builds two Arnoldi processes, one for the right Krylov space and one for the left Krylov space, and uses oblique projections of the matrix on these spaces [28], [36].

When the coefficients of $A(\lambda)$ are polynomials in λ, $A(\lambda)$ is called a matrix polynomial. The eigenvalue problem is then called the polynomial eigenvalue problem. Its properties and solution are pretty well understood. Most numerical methods for computing a selection of eigenvalues strongly rely on the concept of linearization, that is, a linear pencil of larger size than A, that has exactly the same eigenvalues as the matrix polynomial. This generalized eigenvalue problem is then solved by an eigensolver for linear eigenvalue problems. Linearizations can also be used for solving nonlinear eigenvalue problems which are not polynomial, by first approximating the nonlinear functions by polynomials or even rationals. The matrix-valued function from the nonlinear eigenvalue problem can, in general, be written as

$$(1.2) \qquad A(s) = \sum_{j=0}^{m} C_j f_j(s),$$

where $C_j \in \mathbb{C}^{n \times n}$ are constant matrix factors and $f_j(s)$ are nonlinear scalar functions. In applications from physics and engineering such problems often have small m, e.g., three or four, which makes the approximation by a matrix polynomial or rational matrix function a little less complicated, since one just has to approximate the functions $f_j(s)$ by polynomials or rationals [10], [17], [33].

In this paper, we focus on polynomial or rational approximation $P(s)$ of the matrix-valued function $A(s)$ in (1.2). We consider rational Krylov methods for linearizations of $P(\lambda)$, which, in our context, is a pencil of order nd, where d is the degree of the polynomial. The two orthogonal Arnoldi method (TOAR) [21], the compact rational Krylov method (CORK) [34], and related methods [18], [33], [17] all exploit the structure in the linearization so that the memory cost and the cost of the Gram–Schmidt orthogonalization process are reduced by almost a factor d compared to applying a standard Krylov method directly to the linearization. Linearizations with compact representation of the Krylov vectors exist for many polynomial and rational bases. Most common is the companion linearization for matrix polynomials in monomial basis [13], [14]. Linearizations have been introduced and used for solving eigenvalue problems expressed in Newton and rational Newton polynomials [17], [33], Chebyshev polynomials [1], Lagrange polynomials [6], [35], rational monomials [2], [24], and the spectral discretization, which is also a rational approximation [18].

Two-sided methods already exist for the infinite Arnoldi method [12] and the delay Arnoldi method [26]. In this paper, we present a more generic approach that can be used for matrix polynomials or rational matrix-valued functions $P(s)$ that are expressed in polynomial or rational bases satisfying a linear relation. Both TOAR and CORK are designed for a compact representation of the right Krylov vectors $\mathbf{v}$ of the form $\mathbf{v} = (I \otimes Q)\mathbf{u}$, where Q has orthonormal columns and $\mathbf{u}$ has much smaller dimension than $\mathbf{v}$. For the left Krylov vectors, however, the compact structure for the linearizations employed by CORK is, at first sight, completely lost. We show that the left Krylov subspace for CORK, and the right and the left Krylov subspaces for the symmetrically structured linearizations in [27] can also be represented in some compact form, but the latter is more expensive to exploit than for the one-sided CORK method. We show that the cost can be greatly reduced by employing special

inner products in the Krylov methods. Due to the potential breakdown or loss of numerical stability of the two-sided Lanczos method, we instead use the two-sided Arnoldi method, i.e., we build two Krylov spaces, one for the matrix pencil of the linearization and one for its transpose. The eigenvalues and right and left eigenvectors are obtained from the two-sided projection of the pencil. Note that the cost for explicitly computing the oblique projection of the pencil on the left and right Krylov spaces does not increase the computational cost dramatically, at least for moderate size Krylov spaces.

As an alternative to the approach presented in this paper, one can build Krylov spaces using CORK for a rational approximation of both $A(s)$ and $A^T(s)$. The Q-factors, Q_R and Q_L, returned by CORK can be used to compute a two-sided projection of $A(s)$ into the smaller matrix-valued function $Q_L^T A(s) Q_R$. This approach is significantly easier than the method presented in this paper but has several disadvantages: (1) the reduced problem is still a nonlinear eigenvalue problem that needs to be solved; (2) if a method based on linearization is used to solve the projected nonlinear eigenvalue problem, then the linearization can still be of large size; (3) it is not clear how eigenvalues converge for such projection; and (4) it is not clear how to use implicit restarting for such approach.

The outline of this paper is as follows. In section 2, we present the general framework of matrix polynomials using two types of linearizations: the CORK linearization [34] and symmetrically structured linearizations [27]. In section 3, we introduce the compact form of the iteration vectors and present the algorithm for building the Krylov spaces using this form. We also introduce an alternative inner product to reduce the computational cost. Section 4 illustrates the methods by numerical examples, where we compare accuracy and memory cost for various alternatives.

We use lowercase Roman and Greek letters for numbers (e.g., f, ϕ), uppercase Greek letters and lowercase Roman letters to indicate vectors (e.g., Φ or v), capital letters for matrices (e.g., A), boldface lower and upper case letters for block vectors and matrices (e.g., $\mathbf{v}, \mathbf{A}$), and blackboard boldface letters for matrices containing different block structures (e.g., $\mathbb{A}$). We denote by e_j the unit vector with zero entries everywhere except in position j, where it is one. We will often use the $n \times (n-1)$ matrix

$$(1.3) \qquad \check{I}_n = \begin{bmatrix} 0 \\ I_{n-1} \end{bmatrix},$$

which corresponds to the $n \times n$ identity matrix I_n with its first column removed. When partitioning a vector $\mathbf{v} \in \mathbb{C}^{nd}$ in d blocks of length n, $v^{[i]}$ indicates the ith block.

2. Matrix polynomials, rational functions, and their linearizations. Assume the matrix-valued function in (1.2) is approximated by a matrix polynomial or rational matrix function of the form

$$(2.1) \qquad P(s) = P_0 \xi_0(s) + P_1 \xi_1(s) + \cdots + P_d \xi_d(s),$$

where $P_j \in \mathbb{C}^{n \times n}$ are constant coefficient matrices and $\xi_j(s)$ are rational functions or polynomials. This approximation can, for example, be obtained using one of the techniques described in [16, section 6]. We concentrate on basis functions $\xi_j(s)$ that satisfy a linear recurrence relation. These include the following:

(a) shifted and scaled monomials $\xi_j(s) = (s - \sigma)^j / \prod_{i=0}^{j} \beta_i$, which satisfy the

linear relation

$$(2.2) \qquad \xi_0(s) = \frac{1}{\beta_0}, \quad \xi_j(s) = \frac{s - \sigma}{\beta_j} \xi_{j-1}(s), \quad j \geq 1,$$

for some nonzero scaling factors β_j and shift σ;

(b) orthogonal polynomials, which satisfy a three-term recurrence

$$(2.3)$$
$$\xi_{-1}(s) = 0, \quad \xi_0(s) = 1, \quad s\xi_j(s) = \alpha_j \xi_{j+1}(s) + \beta_j \xi_j(s) + \gamma_j \xi_{j-1}(s), \quad j \geq 0,$$

(note that both the shifted and scaled monomials (2.2), and the Chebyshev polynomials

$$T_0(s) = 1, \quad T_1(s) = s, \quad T_{j+1}(s) = 2sT_j(s) - T_{j-1}(s), \quad j \geq 1,$$

are special cases of (2.3));

(c) rational Newton basis functions, which are defined by the recursion

$$(2.4) \qquad \xi_0(s) = \frac{1}{\beta_0}, \quad \xi_j(s) = \frac{s - \sigma_j}{\beta_j(1 - s/\zeta_j)} \xi_{j-1}(s), \quad j \geq 1,$$

for some scaling factors $\beta_0, \ldots, \beta_j$, interpolation nodes $\sigma_1, \ldots, \sigma_j$ and poles $\zeta_1, \ldots, \zeta_j$ (note that scaled Newton polynomials, which are defined by the linear relation

$$\xi_0(s) = \frac{1}{\beta_0}, \quad \xi_j(s) = \frac{s - \sigma_j}{\beta_j} \xi_{j-1}(s), \quad j \geq 1,$$

are a special case of (2.4) corresponding to choosing poles at infinity);

(d) scaled Lagrange polynomials with distinct nodes $\sigma_0, \ldots, \sigma_d$ and scaling factors $\beta_0, \ldots, \beta_d$,

$$(2.5) \qquad \xi_j(s) = \frac{1}{\beta_j} \prod_{\substack{i=0 \\ i \neq j}}^{d} (s - \sigma_i), \quad 0 \leq j \leq d,$$

which satisfy the linear relation

$$\beta_j(s - \sigma_j)\xi_j(s) = \beta_{j-1}(s - \sigma_{j-1})\xi_{j-1}(s), \quad j > 0.$$

The linear recurrence relation between the basis functions can be rewritten in matrix form as

$$(2.6) \qquad M_d \, \Xi_d(s) = sN_d \, \Xi_d(s),$$

where

$$(2.7) \qquad \Xi_d(s) = [\xi_0(s), \xi_1(s), \ldots, \xi_d(s)]^T$$

and $M_d, N_d \in \mathbb{C}^{d \times (d+1)}$. When (2.6) holds with $M_d - sN_d$ of full row rank, then $M_d - sN_d$ is said to be a *full row-rank linear dual basis to* $\Xi_d(s)$ [11], [27, Definition 12]. Note that in the special case $d = 0$, we consider the corresponding dual linear basis to be empty.

Using the linear relation between $\xi_d(s)$ and $\xi_{d-1}(s)$, the matrix polynomial or rational matrix function $P(s)$ in (2.1) is rewritten in the form

$$(2.8) \qquad g(s)P(s) = \sum_{j=0}^{d-1}(A_j - sB_j)\xi_j(s)$$

for some function $g(s)$ and matrices A_j, B_j that depend on the coefficient matrices P_j of $P(s)$ in (2.1)—see, for example, [34], [16, Table 6.6]. Note that $g(s) = 1$ for polynomial bases but $g(s) = (1 - s/\zeta_d)$ for the rational Newton basis.

Define

$$(2.9) \qquad \mathbf{A} - s\mathbf{B} = \begin{bmatrix} A_0 - sB_0 & A_1 - sB_1 & \cdots & A_{d-1} - sB_{d-1} \end{bmatrix} \in \mathbb{C}^{n \times nd},$$

and rewrite (2.8) as

$$(2.10) \qquad g(s)P(s) = (\mathbf{A} - s\mathbf{B})(\Xi_{d-1}(s) \otimes I_n).$$

The pencil

$$(2.11) \qquad \mathbb{A} - s\mathbb{B} = \begin{bmatrix} \mathbf{A} - s\mathbf{B} \\ (M_{d-1} - sN_{d-1}) \otimes I_n \end{bmatrix} \in \mathbb{C}^{nd \times nd}$$

is referred to as the *CORK pencil* [34]. It is a linearization of $P(s)$ when the latter is a matrix polynomial. The CORK pencil satisfies two one-sided factorizations as the next lemma shows.

LEMMA 2.1. *Let* $\Xi_{d-1}(s) = [\xi_0(s), \ldots, \xi_{d-1}(s)]^T$ *be a family of basis functions satisfying the linear relation* $M_{d-1}\Xi_{d-1}(s) = sN_{d-1}\Xi_{d-1}(s)$. *For every* $s \in \mathbb{C}$ *such that the* $d - 1$ *last columns of* $M_{d-1} - sN_{d-1}$ *are linearly independent and* $\xi_0(s) \neq 0$, *the pencil* $\mathbb{A} - s\mathbb{B}$ *in* (2.11) *and* $g(s)P(s)$ *in* (2.10) *satisfy the*

(i) *right-sided factorization*

$$(2.12) \qquad (\mathbb{A} - s\mathbb{B})G(s) = e_1 \otimes g(s)P(s),$$

where $G(s) = \Xi_{d-1}(s) \otimes I_n$ *and* e_1 *is the first column of the* $d \times d$ *identity matrix, and the*

(ii) *left-sided factorization*

$$(2.13) \qquad H(s)(\mathbb{A} - s\mathbb{B}) = e_1^T \otimes g(s)P(s),$$

where $H(s) = \xi_0(s) \begin{bmatrix} I_n & R_\xi^T(s) \end{bmatrix}$ *with* $R_\xi^T(s) = -(\mathbf{A} - s\mathbf{B})(\breve{I}_d(M_{d-1}\breve{I}_d - sN_{d-1}\breve{I}_d)^{-1} \otimes I_n)$ *and* $\breve{I}_d$ *as in* (1.3).

Proof. Statement (i) follows from $(M_{d-1} - sN_{d-1})\Xi_{d-1}(s) = 0$ and (2.10). See also [34].

For statement (ii), we start by rewriting $(M_{d-1} - sN_{d-1})\Xi_{d-1} = 0$ as

$$(2.14) \qquad \xi_0(s)(M_{d-1} - sN_{d-1})e_1 = -(M_{d-1} - sN_{d-1})\breve{I}_d\breve{I}_d^T\Xi_{d-1}(s).$$

Since s is such that $(M_{d-1} - sN_{d-1})\breve{I}_d$ is nonsingular and $\xi_0(s) \neq 0$, we can write (2.14) as

$$(M_{d-1}\breve{I}_d - sN_{d-1}\breve{I}_d)^{-1}(M_{d-1} - sN_{d-1})e_1 = -\breve{I}_d^T\Xi_{d-1}(s)/\xi_0(s).$$

Hence, $(M_{d-1}\check{I}_d - sN_{d-1}\check{I}_d)^{-1}(M_{d-1} - sN_{d-1}) = \left[\, -\check{I}_d^T\Xi_{d-1}(s)/\xi_0(s) \quad I_{d-1} \,\right]$, so that

$$(2.15) \qquad \xi_0(s)\big(I_d - \check{I}_d(M_{d-1}\check{I}_d - sN_{d-1}\check{I}_d)^{-1}(M_{d-1} - sN_{d-1})\big) = e_1^T \otimes \Xi_{d-1}(s).$$

Now, the left-hand side of (2.13) can be rewritten as

$$(2.16) \quad \xi_0(s)(\mathbf{A} - s\mathbf{B}) - \xi_0(s)(\mathbf{A} - s\mathbf{B})(\check{I}_d(M_{d-1}\check{I}_d - sN_{d-1}\check{I}_d)^{-1}(M_{d-1} - sN_{d-1})\otimes I_n).$$

From (2.15), after Kronecker product with I_n, we have that (2.16) equals

$$(\mathbf{A} - s\mathbf{B})(e_1^T \otimes \Xi_{d-1}(s) \otimes I_n),$$

which, by (2.10), equals $e_1^T \otimes g(s)P(s)$. $\qquad\square$

The next result describes the structure of the eigenvectors of $\mathbb{A} - s\mathbb{B}$. It is a direct consequence of [15, Thm. 3.1] applied to the one-sided factorizations (2.12)–(2.13).

THEOREM 2.2. *Let $P(s)$ be the $n \times n$ matrix polynomial or rational matrix-valued function in* (2.10), *let $\mathbb{A} - s\mathbb{B}$ be the $nd \times nd$ pencil in* (2.11), *and let $\lambda \in \mathbb{C}$.*

   (i) *If* (2.12) *holds at λ with $\Xi_{d-1}(\lambda) \neq 0$, then $\Xi_{d-1}(\lambda)\otimes x$ is a right eigenvector of $\mathbb{A} - s\mathbb{B}$ with eigenvalue λ if and only if x is a right eigenvector of $P(s)$ with eigenvalue λ.*

   (ii) *If* (2.13) *holds at λ with $H(\lambda)$ of full rank, then $\left[\, \dfrac{y}{R_\xi(\lambda)y} \,\right]$ is a left eigenvector of $\mathbb{A} - s\mathbb{B}$ with eigenvalue λ if and only if y is a left eigenvector of $P(s)$ with eigenvalue λ.*

The conditions $(M_{d-1} - sN_{d-1})\check{I}_d$ being nonsingular and $\xi_0(s) \neq 0$ are not too strict since we can permute the order of the entries in the basis Ξ_{d-1} if needed. Note that for degree graded polynomial bases, $\xi_0(s)$ is a constant and therefore nonzero.

Example 1. The degree d matrix polynomial expressed in the monomial basis as

$$(2.17) \qquad\qquad P(s) = P_0 + sP_1 + s^2P_2 + \cdots + s^dP_d,$$

can be rewritten in the form (2.8) with $g(s) = 1$, $A_i = P_i$, $B_i = 0$ for $i = 0, \ldots, d-2$ and $A_{d-1} = P_{d-1}$, $B_{d-1} = -P_d$. Then the pencil $\mathbb{A} - s\mathbb{B}$ in (2.11) has the form

$$(2.18) \qquad \mathbb{A} = \begin{bmatrix} P_0 & P_1 & \cdots & P_{d-1} \\ 0 & I_n & & \\ & \ddots & \ddots & \\ & & 0 & I_n \end{bmatrix}, \qquad \mathbb{B} = \begin{bmatrix} 0 & \cdots & 0 & -P_d \\ I_n & 0 & & \\ & \ddots & \ddots & \\ & & I_n & 0 \end{bmatrix},$$

which is close to the first companion form [13], [14].

From Theorem 2.2, the right and left eigenvectors of $\mathbb{A} - s\mathbb{B}$, with eigenvalue λ, have the form

$$\mathbf{x} = \begin{bmatrix} x \\ \lambda x \\ \lambda^2 x \\ \vdots \\ \lambda^{d-1}x \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y \\ -(\sum_{i=1}^d \lambda^{i-1}P_i)^*y \\ -(\sum_{i=2}^d \lambda^{i-2}P_i)^*y \\ \vdots \\ -(P_{d-1} + \lambda P_d)^*y \end{bmatrix}.$$

The blocks of the right eigenvector are just scalar multiples of the right eigenvector x of $P(s)$ whereas the structure of the left eigenvector is more complicated.

As we are interested in computing both right and left eigenvectors of $P(s)$, we also consider a class of symmetrically structured pencils associated with $P(s)$ whose right and left eigenvectors have a similar Kronecker product form. Assuming for simplicity that d is odd, we rewrite $P(s)$ in (2.1) as

$$(2.19) \qquad g(s)P(s) = \sum_{j=0}^{\frac{d-1}{2}} (A_j - sB_j)\phi_j(s)\psi_j(s)$$

for some function $g(s)$ and matrices A_j, B_j that depend on the coefficient matrices P_j of $P(s)$ in (2.1) (note that these differ in general from $g(s)$ and A_j, B_j in (2.8) but for the presentation of our general framework, it is convenient to not introduce new notation here). The $\phi_j(s)$, $j = 1, \ldots, d_\phi$ and the $\psi_j(s)$, $j = 0, \ldots, d_\psi$ with $d_\phi = d_\psi = \frac{d-1}{2}$ are two families of basis functions, each of them satisfying a linear relation such that if

$$(2.20) \qquad \Phi_{d_\phi}(s) = [\phi_0(s), \ldots, \phi_{d_\phi}(s)]^T, \qquad \Psi_{d_\psi}(s) = [\psi_0(s), \ldots, \psi_{d_\psi}(s)]^T,$$

then

$$(2.21) \qquad (M_{d_\phi} - sN_{d_\phi})\Phi_{d_\phi}(s) = 0, \qquad (M_{d_\psi} - sN_{d_\psi})\Psi_{d_\psi}(s) = 0,$$

with $M_{d_\phi}, N_{d_\phi} \in \mathbb{C}^{d_\phi \times (d_\phi+1)}$ and $M_{d_\psi}, N_{d_\psi} \in \mathbb{C}^{d_\psi \times (d_\psi+1)}$. Now define the $n(d_\psi + 1) \times n(d_\phi + 1) = n\left(\frac{d+1}{2}\right) \times n\left(\frac{d+1}{2}\right)$ pencil

$$(2.22) \qquad \mathbf{A} - s\mathbf{B} = \begin{bmatrix} A_0 - sB_0 & & & \\ & A_1 - sB_1 & & \\ & & \ddots & \\ & & & A_{\frac{d-1}{2}} - sB_{\frac{d-1}{2}} \end{bmatrix}$$

and rewrite (2.19) as

$$(2.23) \qquad g(s)P(s) = (\Psi_{d_\psi}(s) \otimes I_n)^T (\mathbf{A} - s\mathbf{B})(\Phi_{d_\phi}(s) \otimes I_n).$$

When $P(s)$ is a matrix polynomial, both $M_{d_\phi} - sN_{d_\phi}$ and $M_{d_\psi} - sN_{d_\psi}$ are of full row-rank with $d_\phi = d_\psi$, and the elements in the polynomial basis $\Phi_{d_\phi}(s)$ (respectively, $\Psi_{d_\psi}(s)$) have no common divisor, Robol, Vandebril, and Van Dooren [27, Thm. 15] show that

$$(2.24) \qquad \mathbb{A} - s\mathbb{B} = \begin{bmatrix} \mathbf{A} - s\mathbf{B} & (M_{d_\psi} - sN_{d_\psi})^T \otimes I_n \\ (M_{d_\phi} - sN_{d_\phi}) \otimes I_n & 0 \end{bmatrix} \in \mathbb{C}^{nd \times nd}$$

is a linearization of $P(s)$. The pencil is symmetrically structured if both $M_{d_\phi} - sN_{d_\phi}$ and $M_{d_\psi} - sN_{d_\psi}$ have the same structure. This is, for example, the case if we use two polynomial bases $\Phi_{d_\phi}(s)$ and $\Psi_{d_\psi}(s)$ of the same type. Furthermore, if both bases are identical, i.e., $\Phi_{d_\phi}(s) = \Psi_{d_\psi}(s)$ and $\mathbf{A} - s\mathbf{B}$ is symmetric, then $\mathbb{A} - s\mathbb{B}$ is symmetric.

The rational matrix function or matrix polynomial $P(s)$ can be rewritten in the form (2.19) in several ways. We describe below a possibility for the standard basis functions (2.2)–(2.5).

(a) If P in (2.1) is expressed in shifted and scaled monomial basis, then, for the ϕ_j and ψ_j in (2.19), we can choose the shifted and scaled monomials

$$(2.25) \quad \phi_0(s) = \frac{1}{\beta_0}, \quad \phi_j(s) = \frac{(s-\sigma)}{\beta_{2j}}\phi_{j-1}(s), \quad j = 0, \ldots, (d-1)/2,$$

$$(2.26) \quad \psi_0(s) = \frac{1}{\beta_1}, \quad \psi_j(s) = \frac{(s-\sigma)}{\beta_{2j+1}}\psi_{j-1}(s), \quad j = 0, \ldots, (d-1)/2$$

so that

$$A_j = \beta_{2j+1}P_{2j} - \sigma P_{2j+1}, \quad B_j = -P_{2j+1}, \quad j = 0, \ldots, (d-1)/2,$$

and $g(s) = 1$.

(b) For Chebyshev polynomials of the first or second kind, see [19].

(c) If the $\xi(s)$ are the scaled rational Newton basis functions in (2.4), then for $(1 - \frac{s}{\xi_1})P(s)$ (i.e., $g(s) = 1 - s/\xi_1$), we can choose

$$(2.27)$$
$$\phi_0(s) = \frac{1}{\beta_0}, \quad \phi_j(s) = \frac{s - \sigma_{2j-1}}{\beta_{2j}(1 - s/\zeta_{2j})}\phi_{j-1}, \quad j = 1, \ldots, (d-1)/2,$$

$$(2.28)$$
$$\psi_0(s) = \frac{1}{\beta_1}, \quad \psi_j(s) = \frac{s - \sigma_{2j}}{\beta_{2j+1}(1 - s/\zeta_{2j+1})}\psi_{j-1}, \quad j = 1, \ldots, (d-1)/2,$$

so that

$$A_j = \beta_{2j+1}P_{2j} - \sigma_{2j+1}P_{2j+1}, \quad B_j = \frac{\beta_{2j+1}}{\xi_{2j+1}}P_{2j} - P_{2j+1}, \quad j = 0, \ldots, (d-1)/2.$$

(e) If P in (2.1) is expressed in the scaled Lagrange basis $\xi_j(s)$, $j = 0, \ldots, d$ with ξ_j as in (2.5), then as suggested in [27], we can choose the scaled Lagrange basis functions

$$\phi_j(s) = \frac{1}{\beta_{2j}}\prod_{\substack{i=0 \\ i \neq j}}^{(d-1)/2}(s - \sigma_{2i}),$$

$$\psi_j(s) = \frac{1}{\beta_{2j+1}}\prod_{\substack{i=0 \\ i \neq j}}^{(d-1)/2}(s - \sigma_{2i+1}), \quad j = 0, \ldots, (d-1)/2$$

so that

$$A_j = -\sigma_{2j+1}\beta_{2j+1}P_{2j} - \sigma_{2j}\beta_{2j}P_{2j+1},$$
$$B_j = -\beta_{2j+1}P_{2j} - \beta_{2j}P_{2j+1}, \quad j = 0, \ldots, (d-1)/2.$$

Note that the expression for $P(s)$ in (2.19) is a particular case of (2.23) with $d_\psi = 0$, $\Psi_{d_\psi} = [1]$, $d_\phi = d - 1$, and $\mathbf{A} - s\mathbf{B}$ as in (2.9). The next result extends Lemma 2.1 to the more general expression for $P(s)$ in (2.23) and its associated pencil $\mathbb{A} - s\mathbb{B}$ in (2.24).

LEMMA 2.3. *Let $\Phi_{d_\phi}(s)$ and $\Psi_{d_\psi}(s)$ be the two families of basis functions in* (2.20) *satisfying the linear relations* (2.21).

(i) *Let $s \in \mathbb{C}$ be such that $\psi_0(s) \neq 0$ and the last d_ψ columns of $M_{d_\psi} - sN_{d_\psi}$ are linearly independent. Then $P(s)$ in (2.23) and the matrix pencil $\mathbb{A} - s\mathbb{B}$ in (2.24) satisfy the right-sided factorization*

$$(2.29) \qquad (\mathbb{A} - s\mathbb{B})G(s) = e_1 \otimes g(s)P(s),$$

where

$$G(s) = \psi_0(s) \begin{bmatrix} I_{n(d_\phi + 1)} \\ R_\psi(s) \end{bmatrix} (\Phi_{d_\phi}(s) \otimes I_n)$$

with $R_\psi(s) = -\left(\left((M_{d_\psi} - sN_{d_\psi})\check{I}_{d_\psi + 1}\right)^{-T} \check{I}_{d_\psi + 1}^T \otimes I_n\right)(\mathbf{A} - s\mathbf{B}) \in \mathbb{C}^{nd_\psi \times n(d_\phi + 1)}$.

(ii) *Let $s \in \mathbb{C}$ be such that $\phi_0(s) \neq 0$ and the last d_ϕ columns of $M_{d_\phi} - sN_{d_\phi}$ are linearly independent. Then $P(s)$ in (2.23) and the matrix pencil $\mathbb{A} - s\mathbb{B}$ in (2.24) satisfy the left-sided factorization*

$$(2.30) \qquad H(s)(\mathbb{A} - s\mathbb{B}) = e_1^T \otimes g(s)P(s),$$

where

$$H(s) = \phi_0(s)(\Psi_{d_\psi}(s)^T \otimes I_n) \begin{bmatrix} I_{n(d_\psi + 1)} & R_\phi^T(s) \end{bmatrix}$$

with $R_\phi^T(s) = -(\mathbf{A} - s\mathbf{B})(\check{I}_{d_\phi + 1}\left((M_{d_\phi} - sN_{d_\phi})\check{I}_{d_\phi + 1}\right)^{-1} \otimes I_n) \in \mathbb{C}^{n(d_\psi + 1) \times nd_\phi}$.

Proof. The proof that the last nd_ϕ rows of (2.29) and the last nd_ψ columns of (2.30) are zero follows from (2.21). The proof that the first $n(d_\psi + 1)$ rows of (2.29) and the first $n(d_\phi + 1)$ columns of (2.30) are equal to $e_1 \otimes g(s)P(s)$ and $e_1^T \otimes g(s)P(s)$, respectively, is similar to the proof of statement (ii) of Lemma 2.1. $\qquad \square$

The next result follows from [15, Thm. 3.1] and the two one-side factorizations in Lemma 2.3.

THEOREM 2.4. *Let $P(\lambda)$ be the $n \times n$ matrix polynomial or rational matrix-valued function in (2.1), let $\mathbb{A} - s\mathbb{B}$ be the $nd \times nd$ pencil in (2.24), and let $\lambda \in \mathbb{C}$.*

(i) *If (2.29) holds at λ with $G(\lambda)$ of full rank, then*
- $\begin{bmatrix} \Phi_{d_\phi}(\lambda) \otimes x \\ R_\psi(\lambda)(\Phi_{d_\phi}(\lambda) \otimes x) \end{bmatrix}$ *is a right eigenvector of $\mathbb{A} - s\mathbb{B}$ with eigenvalue λ if and only if x is a right eigenvector of $P(s)$ with eigenvalue λ.*
- *If $w \in \mathbb{C}^{nd}$ is a left eigenvector of $\mathbb{A} - s\mathbb{B}$ with eigenvalue λ, then $(e_1^T \otimes I_n)w$ is a left eigenvector of $P(s)$ with eigenvalue λ provided that it is nonzero.*

(ii) *If (2.30) holds at λ with $H(\lambda)$ of full rank, then*
- *If $z \in \mathbb{C}^{nd}$ is a right eigenvector of $\mathbb{A} - s\mathbb{B}$ with eigenvalue λ, then $(e_1^T \otimes I_n)z$ is a right eigenvector of P with eigenvalue λ provided that it is nonzero.*
- $\begin{bmatrix} \overline{\Psi_{d_\psi}(\lambda)} \otimes y \\ R_\phi(\lambda)(\overline{\Psi_{d_\psi}(\lambda)} \otimes y) \end{bmatrix}$ *is a left eigenvector of $\mathbb{A} - s\mathbb{B}$ with eigenvalue λ if and only if y is a left eigenvector of P with eigenvalue λ.*

Note that we can again use permutation matrices in order to relax the conditions on $(M_{d_\phi} - sN_{d_\phi})\check{I}_{d_\phi + 1}$, $(M_{d_\psi} - sN_{d_\psi})\check{I}_{d_\psi + 1}$ being nonsingular, and on $\phi_0(s)$ and $\psi_0(s)$ being nonzero.

Example 2. For matrix polynomials of odd degree expressed in the monomial basis (2.17), we select $\phi_j(s) = s^j$, $\psi_j(s) = s^j$ for $j = 0, \ldots, \frac{d-1}{2}$, so that $g(s) = 1$, $A_j = P_{2j}$, $B_j = -P_{2j+1}$ for $j = 0, \ldots, \frac{d-1}{2}$. The $nd \times nd$ pencil (2.24) is, in this case, a permuted Fiedler linearization [3]:

$$\mathbb{A} = \left[\begin{array}{ccccc|ccc} P_0 & & & & & 0 & & \\ & P_2 & & & & I_n & \ddots & \\ & & \ddots & & & & \ddots & 0 \\ & & & P_{d-1} & & & & I_n \\ \hline 0 & I_n & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & I_n & & & & \end{array} \right], \quad \mathbb{B} = \left[\begin{array}{ccccc|ccc} -P_1 & & & & & I_n & & \\ & -P_3 & & & & 0 & \ddots & \\ & & \ddots & & & & \ddots & I_n \\ & & & -P_d & & & & 0 \\ \hline I_n & 0 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & I_n & 0 & & & & \end{array} \right].$$

It follows from Theorem 2.4 that the left and right eigenvectors of $\mathbb{A} - s\mathbb{B}$ with eigenvalue λ have the form

$$\mathbf{x} = \begin{bmatrix} x \\ \lambda x \\ \vdots \\ \lambda^{\frac{d-1}{2}} x \\ -(\sum_{i=2}^{d} \lambda^{i-1} P_i)x \\ -(\sum_{i=4}^{d} \lambda^{i-2} P_i)x \\ \vdots \\ -(\lambda^{\frac{d-1}{2}}(P_{d-1} + \lambda P_d))x \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y \\ \bar{\lambda} y \\ \vdots \\ \bar{\lambda}^{\frac{d-1}{2}} y \\ -(\sum_{i=2}^{d} \lambda^{i-1} P_i)^* y \\ -(\sum_{i=4}^{d} \lambda^{i-2} P_i)^* y \\ \vdots \\ -(\lambda^{\frac{d-1}{2}}(P_{d-1} + \lambda P_d))^* y \end{bmatrix},$$

where x and y are right and left eigenvectors for P with eigenvalue λ.

In what follows, we consider our matrix valued function $P(s)$ to be in the form (2.23) with corresponding pencil (2.24), which includes the following three special cases:

$\mathbf{L}_1$: the CORK pencil for which
- $d_\psi = 0$, $\Psi_{d_\psi}(s) = [1]$, $d_\phi = d - 1$, and $\Phi_{d_\phi}(s) = \Xi_{d-1}(s)$, with $\Xi_{d-1}(s)$ as in (2.7),
- $\mathbf{A} - s\mathbf{B}$ has the form (2.9) and $(M_{d_\psi} - sN_{d_\psi})^T \otimes I_n$ is the empty matrix.

$\mathbf{L}_2$: the transpose of the CORK pencil, i.e., $\mathbf{L}_2 = \mathbf{L}_1^T$ for which
- $d_\psi = d - 1$, $\Psi_{d_\psi}(s) = \Xi_{d-1}(s)$, $d_\phi = 0$, and $\Phi_{d_\phi}(s) = [1]$,
- $\mathbf{A} - s\mathbf{B}$ is the transpose of (2.9) and where A_i, B_i, $i = 0, \ldots, d-1$ are also transposed, and $(M_{d_\phi} - sN_{d_\phi}) \otimes I_n$ is the empty matrix.

$\mathbf{L}_S$: the symmetrically structured pencil (2.24) for which
- $d_\psi = d_\phi = (d-1)/2$, and $\Psi_{d_\psi}(s)$ and $\Phi_{d_\psi}(s)$ are as in (2.20),
- $\mathbf{A} - s\mathbf{B}$ has the form (2.22).

3. Compact Krylov methods. The (one-sided) rational Krylov method for the $nd \times nd$ pencil $\mathbb{A} - s\mathbb{B}$, outlined in Algorithm 1, builds an orthonormal basis $\mathbf{V}_{k+1} = [\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}] \in \mathbb{C}^{nd \times (k+1)}$ for the (right) Krylov space

$$(3.1) \qquad \text{span}\{\mathbf{V}_{k+1}\} = \text{span}\left\{\mathbf{v}_1, (\mathbb{A} - \sigma_1\mathbb{B})^{-1}\mathbb{B}\mathbf{v}_1, \ldots, (\mathbb{A} - \sigma_k\mathbb{B})^{-1}\mathbb{B}\mathbf{v}_1\right\}$$

for k distinct shifts $\sigma_1, \ldots, \sigma_k$ [29]. If all the shifts are equal, the method is equivalent to the shift-and-invert Arnoldi method, which builds the Krylov space

$$\text{span}\left\{\mathbf{v}_1, (\mathbb{A} - \sigma_1\mathbb{B})^{-1}\mathbb{B}\mathbf{v}_1, \ldots, ((\mathbb{A} - \sigma_1\mathbb{B})^{-1}\mathbb{B})^k\mathbf{v}_1\right\}.$$

Algorithm 1 Rational Krylov algorithm.

Input: $\mathbb{A}, \mathbb{B} \in \mathbb{C}^{nd \times nd}$, a nonzero starting vector $\mathbf{v}_1 \in \mathbb{C}^{nd}$, and k shifts $(\sigma_j)_{j=1}^k$.

1: Normalize $\mathbf{v}_1$ such that $\|\mathbf{v}_1\|_2 = 1$.
2: **for** $j = 1, \ldots, k$ **do**
3:      Compute $\mathbf{w} = (\mathbb{A} - \sigma_j\mathbb{B})^{-1}\mathbb{B}\mathbf{v}_j$.
4:      Orthogonalize: $\widetilde{\mathbf{v}}_{j+1} = \mathbf{w} - \sum_{i=1}^j h_{ij}\mathbf{v}_i$, where $h_{ij} = \mathbf{v}_i^* \mathbf{w}$ for $i = 1, \ldots, j$.
5:      Set $h_{j+1,j} = \|\widetilde{\mathbf{v}}_{j+1}\|_2$, normalize $\widetilde{\mathbf{v}}_{j+1}$: $\mathbf{v}_{j+1} = \widetilde{\mathbf{v}}_{j+1}/h_{j+1,j}$.
6: **end for**

Returns $\mathbf{V}_{k+1} = [\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}] \in \mathbb{C}^{nd \times (k+1)}$ and $\underline{H}_k = (h_{ij}) \in \mathbb{C}^{(k+1) \times k}$.

The matrix $\mathbf{V}_{k+1}$ satisfies the rational Krylov recurrence relation

$$(3.2) \qquad \mathbb{A}\mathbf{V}_{k+1}\underline{H}_k = \mathbb{B}\,\mathbf{V}_{k+1}\underline{K}_k,$$

where $\underline{H}_k$, $\underline{K}_k \in \mathbb{C}^{(k+1) \times k}$ are upper Hessenberg matrices with $\underline{H}_k = (h_{ij})$ and $\underline{K}_k = \underline{I}_k + \underline{H}_k \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$. The Ritz pairs $(\lambda_i, \mathbf{x}_i)$ of $\mathbb{A} - s\mathbb{B}$ can then be obtained from the eigenpairs (λ_i, s_i) of the generalized eigenvalue problem

$$(3.3) \qquad K_k s_i = \lambda_i H_k s_i, \quad s_i \neq 0,$$

where H_k, K_k are the $k \times k$ upper parts of $\underline{H}_k$ and $\underline{K}_k$ by setting $\mathbf{x}_i = \mathbf{V}_{k+1}\underline{H}_k s_i$.

An orthonormal basis $\mathbf{Z}_{k+1} = [\mathbf{z}_1, \ldots, \mathbf{z}_{k+1}]$ for the left Krylov space associated with the pencil $\mathbb{A} - s\mathbb{B}$ is obtained by applying Algorithm 1 to $\mathbb{A}^T$ and $\mathbb{B}^T$. A two-sided rational Krylov method is obtained by applying Algorithm 1 to both $\mathbb{A} - s\mathbb{B}$ and $\mathbb{A}^T - s\mathbb{B}^T$. We then use the computed orthonormal bases $\mathbf{V}_{k+1}$ and $\mathbf{Z}_{k+1}$ to project $\mathbb{A} - s\mathbb{B}$ onto a smaller pencil

$$(3.4) \qquad A - sB = \mathbf{Z}_{k+1}^T(\mathbb{A} - s\mathbb{B})\mathbf{V}_{k+1}$$

of size $(k+1) \times (k+1)$. The right and left Ritz vectors $\mathbf{x}$ and $\mathbf{y}$ of $\mathbb{A} - s\mathbb{B}$ are spanned by $\mathbf{V}_{k+1}$ and $\mathbf{Z}_{k+1}$, respectively, i.e., $\mathbf{x} = \mathbf{V}\hat{x}$ and $\mathbf{y} = \overline{\mathbf{Z}}\hat{y}$, where $\hat{x}$ and $\hat{y}$ are right and left eigenvectors of $A - sB$, respectively. The triplet $(\lambda, \mathbf{x}, \mathbf{y})$ forms a desired approximate eigenvalue-eigenvector triplet for $\mathbb{A} - s\mathbb{B}$.

The size nd of the iteration vectors in Algorithm 1 is large, which can lead to a heavy memory burden and an expensive orthogonalization process. Also, solving the large scale system in step 3 of Algorithm 1, without exploiting its structure, is prohibitively expensive in memory for large n. To alleviate these expenses, we use the Kronecker structure of the pencil $\mathbb{A} - s\mathbb{B}$ to express the iteration vectors in a compact format that considerably reduces the memory cost, and the cost of the Gram–Schmidt orthogonalization process. We describe this compact format in section 3.1 and explain how to compute the iteration vectors efficiently, which includes the efficient solution of the linear system in step 3 of Algorithm 1. Section 3.2 explains how to make use of the compact form of the vectors to compute the inner products needed for the Gram–Schmidt orthogonalization process. We present the compact rational Krylov algorithm in section 3.3. Section 3.4 explains how to efficiently compute the reduced eigenvalue problem (3.4). Section 3.5 presents further reductions on the memory consumption and computational cost. We close the section with a comment on implicit filtering in section 3.6, which we need for improving the numerical stability of the two-sided method.

3.1. Compact form of the iteration vectors. We show that the iteration vectors $\mathbf{w}$ and $\mathbf{v}_j$ in Algorithm 1 can be written in a compact form involving Kronecker products. Consider the pencil $\mathbb{A} - s\mathbb{B}$ in (2.24), associated with the expression for $P(s)$ in (2.23), and the solution $\mathbf{w}$ to the linear system

$$(3.5) \qquad\qquad (\mathbb{A} - s\mathbb{B})\mathbf{w} = \mathbb{B}\mathbf{v}$$

for some shift s, such that $\mathbb{A} - s\mathbb{B}$ is nonsingular. Consider also vectors $\mathbf{v}$ and $\mathbf{w}$ of length nd that we partition as follows:

$$(3.6) \qquad\qquad \mathbf{v} = \begin{bmatrix} \mathbf{v}^{[\phi]} \\ \mathbf{v}^{[\psi]} \end{bmatrix} \quad , \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}^{[\phi]} \\ \mathbf{w}^{[\psi]} \end{bmatrix},$$

with $\mathbf{v}^{[\phi]}, \mathbf{w}^{[\phi]} \in \mathbb{C}^{n(d_\phi+1)}$ and $\mathbf{v}^{[\psi]}, \mathbf{w}^{[\psi]} \in \mathbb{C}^{nd_\psi}$. The vector $\mathbf{v}^{[\phi]}$ is itself partitioned into $d_\phi + 1$ vectors $v^{[0]}, \ldots, v^{[d_\phi]}$ of length n, and $\mathbf{v}^{[\psi]}$ consists of the d_ψ vectors $v^{[d_\phi+1]}, \ldots, v^{[d_\phi+d_\psi]}$ of length n.

Assume that the conditions of Lemma 2.3 with respect to $M_{d_\psi} - sN_{d_\psi}$, $M_{d_\phi} - sN_{d_\phi}$, and $\phi_0(s)$ are satisfied. By Theorem 2.4, if s is not an eigenvalue of $\mathbb{A} - s\mathbb{B}$, then it is not an eigenvalue of $P(s)$ so that $P(s)$ is nonsingular. We can transform the linear system (3.5) to an equivalent system by multiplying it on the left with

$$\begin{bmatrix} H(s) \\ \check{I}_{d_\psi+1}^T \otimes I_n & 0 \\ 0 & I_{d_\phi} \otimes I_n \end{bmatrix},$$

where $H(s)$ is as in Lemma 2.3(ii) and is assumed to be of full row rank. By using Lemma 2.3(ii) to replace $H(s)(\mathbb{A} - s\mathbb{B})$ with $e_1^T \otimes g(s)P(s)$, we obtain the linear system

$$(3.7) \quad \begin{bmatrix} e_1^T \otimes g(s)P(s) & 0 \\ (\check{I}_{d_\psi+1}^T \otimes I_n)(\mathbf{A} - s\mathbf{B}) & \check{I}_{d_\psi+1}^T(M_{d_\psi} - sN_{d_\psi})^T \otimes I_n \\ (M_{d_\phi} - sN_{d_\phi}) \otimes I_n & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}^{[\phi]} \\ \mathbf{w}^{[\psi]} \end{bmatrix}$$

$$= \begin{bmatrix} H(s) \\ \check{I}_{d_\psi+1}^T \otimes I_n & 0 \\ 0 & I_{d_\phi} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{B} & N_{d_\psi}^T \otimes I_n \\ N_{d_\phi} \otimes I_n & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}^{[\phi]} \\ \mathbf{v}^{[\psi]} \end{bmatrix}.$$

The solution $\mathbf{w}$ to (3.7) can be obtained as follows:

$$(3.8) \qquad w^{[0]} = g(s)^{-1}P(s)^{-1}H(s)\mathbb{B}\mathbf{v},$$
$$(3.9) \quad (\check{I}_{d_\phi+1}^T \otimes I_n)\mathbf{w}^{[\phi]} = \big(E_\phi(s)N_{d_\phi} \otimes I_n\big)\mathbf{v}^{[\phi]} - E_\phi(s)(M_{d_\phi} - sN_{d_\phi})e_1 \otimes w^{[0]},$$
$$\mathbf{w}^{[\psi]} = \big(F_\psi(s)^T \otimes I_n\big)\mathbf{B}\mathbf{v}^{[\phi]} + \big((N_{d_\psi}F_\psi(s))^T \otimes I_n\big)\mathbf{v}^{[\psi]}$$
$$(3.10) \qquad\qquad - \big(F_\psi(s)^T \otimes I_n\big)(\mathbf{A} - s\mathbf{B})\mathbf{w}^{[\phi]},$$

where

$$(3.11) \quad E_\phi(s) = \big((M_{d_\phi} - sN_{d_\phi})\check{I}_{d_\phi+1}\big)^{-1}, \qquad F_\psi(s) = \check{I}_{d_\psi+1}\big((M_{d_\psi} - sN_{d_\psi})\check{I}_{d_\phi+1}\big)^{-1}.$$

In what follows, we will repeatedly use the fact that if $\mathbf{a} = (X \otimes I_n)\mathbf{b}$ for some $\mathbf{a}, \mathbf{b} \in \mathbb{C}^{dn}$ and $X \in \mathbb{C}^{d \times d}$, then the jth block $a^{[j]}$ of length n of $\mathbf{a}$ is a linear combination of the d blocks $b^{[1]}, b^{[2]}, \ldots, b^{[d]}$ of $\mathbf{b}$. We will need the following result.

LEMMA 3.1. *Let* $r_k = rank \, \widetilde{V}_k$, *where*

$$(3.12) \qquad \widetilde{V}_k = \big[v_1^{[0]}, \ldots, v_1^{[d_\phi]}, v_2^{[0]}, \ldots, v_2^{[d_\phi]}, \ldots, v_k^{[0]}, \ldots, v_k^{[d_\phi]}\big] \in \mathbb{C}^{n \times (d_\phi+1)k},$$

with $v_j^{[\ell]}$ *denoting the* $(\ell+1)$*st block of length* n *of the* j*th column of the matrix* $\mathbf{V}_k$
in Algorithm 1. *Then,*

    (i) $r_1 \le \min(d_\phi + 1, n)$,
    (ii) $r_k = r_{k-1}$ *or* $r_k = r_{k-1} + 1$ *for* $k > 1$.

Proof. Statement (i) is obvious. Let us prove (ii). Let the columns of $Q_k \in \mathbb{C}^{n \times r_k}$
form an orthonormal basis for the column range of $\widetilde{V}_k$. Since $\mathbf{v}_k$ for $k > 1$ is a
linear combination of $\mathbf{w}_{k-1}$ and $\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}$ (see step 4 of Algorithm 1), the blocks
$v_k^{[i]}$, $i = 0, \ldots, d_\phi$, of $\mathbf{v}_k^{[\phi]}$ lie in the column range of $[Q_{k-1}, w_{k-1}^{[0]}, \ldots, w_{k-1}^{[d_\phi]}]$. But,
by (3.9), with $\mathbf{w}^{[\phi]} \equiv \mathbf{w}_{k-1}^{[\phi]}$ and $\mathbf{v}^{[\phi]} \equiv \mathbf{v}_{k-1}^{[\phi]}$, we see that $w_{k-1}^{[j]}$, $1 \le j \le d_\phi$ is a linear
combination of $v_{k-1}^{[0]}, v_{k-1}^{[1]}, \ldots, v_{k-1}^{[d_\phi]}$, and $w_{k-1}^{[0]}$, so that

$$\text{range}(Q_k) = \text{range}\big([Q_{k-1}, w_{k-1}^{[0]}, \ldots, w_{k-1}^{[d_\phi]}]\big) = \text{range}\big([Q_{k-1}, w_{k-1}^{[0]}]\big),$$

i.e., r_k can be at most $r_{k-1} + 1$, which proves the lemma. $\qquad\square$

We introduce the auxiliary matrix

$$(3.13) \quad L_k = [A_0, A_1, \ldots, A_{d_{\mathrm{AB}}-1}, B_0, B_1, \ldots, B_{d_{\mathrm{AB}}-1}](I_{2d_{\mathrm{AB}}} \otimes Q_k) \in \mathbb{C}^{n \times 2d_{\mathrm{AB}} r_k},$$

where $Q_k \in \mathbb{C}^{n \times r_k}$ is the matrix with orthonormal columns defined in the proof of
Lemma 3.1 and d_{AB} is the number of nonzero $n \times n$ blocks in the pencil $\mathbf{A} - s\mathbf{B}$, used
to define $\mathbb{A} - s\mathbb{B}$.

THEOREM 3.2 (compact form of iteration vector $\mathbf{v}_k$). *Let* $\mathbf{v}_k \in \mathbb{C}^{nd}$, $k > 1$, *be*
the iteration vector computed at step $k-1$ *of the rational Krylov algorithm applied*
to the pair $(\mathbb{A}, \mathbb{B})$ *associated with* $P(s)$ *in* (2.19). *Assume that* $s = \sigma_k$ *is such that*
$\mathbb{A} - s\mathbb{B}$ *is nonsingular and the conditions of Lemma* 2.3 *with respect to* $M_{d_\psi} - sN_{d_\psi}$,
$M_{d_\phi} - sN_{d_\phi}$, *and* $\phi_0(s)$ *are satisfied. Then,*

$$(3.14) \qquad \mathbf{v}_k^{[\phi]} = (I_{d_\phi+1} \otimes Q_k)\mathbf{u}_k^{[\phi]} \quad \text{for some } \mathbf{u}_k^{[\phi]} \in \mathbb{C}^{r_k(d_\phi+1)},$$

where $r_k = rank \, \widetilde{V}_k$ *with* $\widetilde{V}_k$ *as in* (3.12) *and the columns of* $Q_k \in \mathbb{C}^{n \times r_k}$ *form an*
orthonormal basis for $\text{range}(\widetilde{V}_k)$. *Moreover, with* L_k *as in* (3.13), *if* $d_\psi \ne 0$ *and the*
lower part of the initial vector $\mathbf{v}_1^{[\psi]}$ *is chosen to be of the form* $\mathbf{v}_1^{[\psi]} = (I_{d_\psi} \otimes L_1)\mathbf{u}_1^{[\psi]}$
for some $\mathbf{u}_1^{[\psi]} \in \mathbb{C}^{2r_1 d_{AB} d_\psi}$, *then*

$$(3.15) \qquad \mathbf{v}_k^{[\psi]} = (I_{d_\psi} \otimes L_k)\mathbf{u}_k^{[\psi]} \quad \text{for some } \mathbf{u}_k^{[\psi]} \in \mathbb{C}^{2r_k d_{AB} d_\psi}.$$

Proof. The compact form for $\mathbf{v}_k^{[\phi]}$ follows from Lemma 3.1 and its proof. For $\mathbf{v}_k^{[\psi]}$,
we proceed by induction on k. The compact form clearly holds for $k = 1$. Assume it
holds for $k-1$, i.e., $\mathbf{v}_{k-1}^{[\psi]} = (I_{d_\psi} \otimes L_{k-1})\mathbf{u}_{k-1}^{[\psi]}$ for some $\mathbf{u}_{k-1}^{[\psi]} \in \mathbb{C}^{2r_{k-1}d_{AB}d_\psi}$. From
Algorithm 1, $\mathbf{v}_k^{[\psi]} = \alpha\mathbf{w}_{k-1}^{[\psi]} - \alpha \sum_{i=1}^{k-1} h_{ij}\mathbf{v}_i^{[\psi]}$ with α such that $\|\mathbf{v}_k\|_2 = 1$. Now, from
(3.13)–(3.14) and the induction hypothesis, for the two first terms in the expression
for $\mathbf{w}_{k-1}^{[\psi]}$ in (3.10), it is easy to see that

$$\big(F_\psi(s)^T \otimes I_n\big)\mathbf{B}\mathbf{v}_{k-1}^{[\phi]} + \big((N_{d_\psi}F_\psi(s))^T \otimes I_n\big)\mathbf{v}_{k-1}^{[\psi]} = (I_{d_\psi} \otimes L_{k-1})\mathbf{a}$$

for some $\mathbf{a} \in \mathbb{C}^{2r_{k-1}d_{AB}d_\psi}$. It follows from (3.8)–(3.9) and (3.14) that

$$(3.16) \qquad \mathbf{w}_{k-1}^{[\phi]} = (I_{d_\phi+1} \otimes Q_k)\mathbf{t}_k^{[\phi]} \text{ for some } \mathbf{t}_k^{[\phi]} \in \mathbb{C}^{r_k(d_\phi+1)}$$

so that $\left(F_\psi(s)^T \otimes I_n\right)(\mathbf{A} - s\mathbf{B})\mathbf{w}_{k-1}^{[\phi]} = (I_{d_\psi} \otimes L_k)\mathbf{b}$ for some $\mathbf{b} \in \mathbb{C}^{2r_k d_{AB}d_\psi}$. Hence,

$$(3.17) \qquad \mathbf{w}_k^{[\psi]} = (I_{d_\psi} \otimes L_k)\mathbf{t}_k^{[\psi]} \text{ for some } \mathbf{t}_k^{[\psi]} \in \mathbb{C}^{2r_k d_{AB}d_\psi}$$

and $\mathbf{v}_k^{[\psi]} = \alpha\mathbf{w}_{k-1}^{[\psi]} - \alpha\sum_{i=1}^{k-1} h_{ij}\mathbf{v}_i^{[\psi]} = (I_{d_\psi} \otimes L_k)\mathbf{u}_k^{[\psi]}$ for some $\mathbf{u}^{[\psi]} \in \mathbb{C}^{2r_k d_{AB}d_\psi}$, which concludes the proof. $\qquad \square$

For the compact forms (3.14)–(3.17) to hold, $\mathbf{v}_1^{[\psi]}$ must lie in the column range of L_1. Recall that, by Lemma 3.1, the size of Q_k increases by at most one at each iteration and that of L_k by at most $2d_{AB}$ so that L_k can end up having a large number of columns. We explain in section 3.5 how to reduce the number of columns of L_k, which can be as little as mr_k, where m is the number of matrices defining the nonlinear eigenvalue problem via the matrix-valued function in (1.2) and is usually small. Now, with an appropriate choice of the starting vector $\mathbf{v}_1$, $r_1 = \text{rank}(Q_1) = 1$ and Q_k has at most rank $r_k = k$. So we encourage the reader to think of Q_k as being $n \times k$ and L_k to be n times a small multiple of k.

We drop the subscript k in what follows and write

$$(3.18) \qquad \mathbf{v} = \begin{bmatrix} \mathbf{v}^{[\phi]} \\ \mathbf{v}^{[\psi]} \end{bmatrix} = \begin{bmatrix} (I_{d_\phi+1} \otimes Q)\mathbf{u}^{[\phi]} \\ (I_{d_\psi} \otimes L)\mathbf{u}^{[\psi]} \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} \mathbf{w}^{[\phi]} \\ \mathbf{w}^{[\psi]} \end{bmatrix} = \begin{bmatrix} (I_{d_\phi+1} \otimes Q)\mathbf{t}^{[\phi]} \\ (I_{d_\psi} \otimes L)\mathbf{t}^{[\psi]} \end{bmatrix}$$

for some $\mathbf{u}^{[\phi]}, \mathbf{t}^{[\phi]} \in \mathbb{C}^{r(d_\phi+1)}$ and some $\mathbf{u}^{[\psi]}, \mathbf{t}^{[\psi]} \in \mathbb{C}^{2rd_{AB}d_\psi}$, and where r is the rank of Q. We still have to present a practical way to compute the short vectors $\mathbf{t}^{[\phi]}$ and $\mathbf{t}^{[\psi]}$, and then $\mathbf{u}^{[\phi]}, \mathbf{u}^{[\psi]}$ defining the next iteration vector $\mathbf{v}$. Note that, since L is not a full rank matrix, $\mathbf{u}^{[\psi]}$ and $\mathbf{t}^{[\psi]}$ are not unique.

To compute $\mathbf{t}$, it is convenient to introduce $\mathbf{A}_L, \mathbf{B}_L \in \mathbb{C}^{2rd_{AB}(d_\psi+1)\times r(d_\phi+1)}$ such that

$$(3.19) \qquad \mathbf{A}(I_{d_\phi+1} \otimes Q) = (I_{d_\psi+1} \otimes L)\mathbf{A}_L \qquad \mathbf{B}(I_{d_\phi+1} \otimes Q) = (I_{d_\psi+1} \otimes L)\mathbf{B}_L.$$

The matrices $\mathbf{A}_L$ and $\mathbf{B}_L$ are in general of much smaller dimension than $\mathbf{A}$ and $\mathbf{B}$. Note that for $\mathbf{A}, \mathbf{B}$ in (2.9) for which $d_{AB} = d = d_\phi + 1$, we have that

$$\mathbf{A}_L = [e_1, e_2, \ldots, e_{d_{AB}}] \otimes I_r, \qquad \mathbf{B}_L = [e_{d_{AB}+1}, e_{d_{AB}+2}, \ldots, e_{2d_{AB}}] \otimes I_r,$$

and for $\mathbf{A}, \mathbf{B}$ in (2.22) for which $d_{AB} = d_\phi + 1 = d_\psi + 1$, we have

$$\mathbf{A}_L = \begin{bmatrix} e_1 & & & \\ & e_2 & & \\ & & \ddots & \\ & & & e_{d_\psi+1} \end{bmatrix} \otimes I_r, \qquad \mathbf{B}_L = \begin{bmatrix} e_{d_\psi+2} & & & \\ & e_{d_\psi+3} & & \\ & & \ddots & \\ & & & e_{2d_\psi+2} \end{bmatrix} \otimes I_r,$$

where e_j is the jth column of the $2d_{AB} \times 2d_{AB}$ identity matrix.

If $\mathbf{w}$ is the solution to the linear system $(\mathbb{A} - s\mathbb{B})\mathbf{w} = \mathbb{B}\mathbf{v}$ at iteration k of Algorithm 1, then the Q in the compact form for $\mathbf{w}$ in (3.18) corresponds to Q_{k+1}, which has not been formed at this point. It is then necessary to compute $w^{[0]}$ as the solution to the linear system in (3.8) first and update Q accordingly (recall that

range$(Q_{k+1}) = $ range$([Q_k, w^{[0]}]))$. But the Q in the compact form for $\mathbf{v}$ is Q_k, which is known at this stage. So we can take advantage of the compact form of $\mathbf{v}$ to reduce the cost of the computation of $w^{[0]}$, starting with the right-hand side $H(s)\mathbb{B}\mathbf{v}$, which we rewrite as

$$(3.20) \qquad b := H(s)\mathbb{B}\mathbf{v} = LH_L(s)\mathbb{B}_L\mathbf{u},$$

where

$$(3.21) \quad H_L(s) = \phi_0(s)(\Psi_{d_\psi}(s)^T \otimes I_{2rd_{AB}}) \left[I_{2rd_{AB}(d_\psi+1)} - (\mathbf{A}_L - s\mathbf{B}_L)(F_\phi(s) \otimes I_r) \right]$$

with $F_\phi(s)$ as in (3.11) (and ϕ in place of ψ), and

$$(3.22) \qquad \mathbb{B}_L = \begin{bmatrix} \mathbf{B}_L & N_{d_\psi}^T \otimes I_{2rd_{AB}} \\ N_{d_\phi} \otimes I_r & 0 \end{bmatrix}.$$

This follows from (3.19) and the fact that

$$(N_{d_\phi} \otimes I_n)(I_{d_\phi+1} \otimes Q) = (I_{d_\phi} \otimes Q)(N_{d_\phi} \otimes I_r),$$
$$(N_{d_\psi}^T \otimes I_n)(I_{d_\psi} \otimes L) = (I_{d_\psi+1} \otimes L)(N_{d_\psi}^T \otimes I_{2rd_{AB}}).$$

Now, replacing $\mathbf{v}^{[\phi]}$ and $\mathbf{v}^{[\psi]}$ in (3.9)–(3.10) by their compact form in (3.18), using (3.19) and $Q^*Q = I_r$, a straightforward computation shows that

$$t^{[0]} = Q^*w^{[0]},$$
$$(3.23) \quad (\check{I}_{d_\phi+1}^T \otimes I_r)\mathbf{t}^{[\phi]} = (E_\phi(s)N_{d_\phi} \otimes I_r)\mathbf{u}^{[\phi]} - E_\phi(s)(M_{d_\phi} - sN_{d_\phi})e_1 \otimes t^{[0]},$$
$$\mathbf{t}^{[\psi]} = \left(F_\psi(s)^T \otimes I_{2rd_{AB}} \right)\left(\mathbf{B}_L(\mathbf{u}^{[\phi]} + s\mathbf{t}^{[\phi]}) - \mathbf{A}_L\mathbf{t}^{[\phi]} \right)$$
$$+ \left(N_{d_\psi}F_\psi(s)^T \otimes I_{2rd_{AB}} \right)\mathbf{u}^{[\psi]}.$$

As mentioned before, the expression for $\mathbf{t}^{[\psi]}$ is not unique but $\mathbf{t}^{[\psi]}$ defines $\mathbf{w}^{[\psi]}$ uniquely.

In what follows we provide simpler expressions for $H_L(s)$ in (3.21) and $\mathbf{t}$ in (3.23) that are also easier to compute. We introduce the matrix $\mathbb{A}_L$ that has the structure of $\mathbb{B}_L$ in (3.22) but with $\mathbf{A}_L, M_{d_\psi}^T$, and M_{d_ϕ} in place of $\mathbf{B}_L, N_{d_\psi}^T$, and N_{d_ϕ}, respectively. We will also need

$$\check{\mathbb{I}}_L = \begin{bmatrix} \check{I}_{d_\psi+1}^T \otimes I_{2rd_{AB}} & \\ & I_{d_\phi} \otimes I_r \end{bmatrix} \quad \text{and} \quad \check{\mathbb{I}}_R = \begin{bmatrix} \check{I}_{d_\phi+1} \otimes I_r & \\ & I_{d_\psi} \otimes I_{2rd_{AB}} \end{bmatrix}.$$

Using an approach similar to the one we used in the proof of Lemma 2.1, we can show that if we let

$$P_L(s) = (\Psi_{d_\psi}(s) \otimes I_{2rd_{AB}})^T(\mathbf{A}_L - s\mathbf{B}_L)(\Phi_{d_\phi}(s) \otimes I_r),$$

then the $r\left(2d_{AB}(d_\psi+1) + d_\phi\right) \times r\left(2d_{AB}d_\psi + (d_\phi+1)\right)$ pencil $\mathbb{A}_L - s\mathbb{B}_L$ and the $2rd_{AB} \times r$ rational matrix function $P_L(s)$ are related by the one-sided factorization

$$(3.24) \qquad H_L(s)(\mathbb{A}_L - s\mathbb{B}_L) = e_1^T \otimes P_L$$

with $H_L(s)$ as in (3.21). Now if we combine the first $2rd_{AB}$ columns of (3.21) to (3.24) with its first r columns removed, we obtain that $H_L(s)$ is a solution to the linear system

$$(3.25) \qquad H_L(s) \left[\begin{bmatrix} I_{2rd_{AB}} \\ 0 \end{bmatrix} \quad (\mathbb{A}_L - s\mathbb{B}_L)\check{\mathbb{I}}_R \right] = \phi_0\psi_0 \begin{bmatrix} I_{2rd_{AB}} & 0 \end{bmatrix}.$$

The matrix on the right-hand side of $H_L(s)$ is block upper triangular with nonsingular diagonal blocks. This follows from the fact that $\breve{I}_{d_\psi+1}^T (M_{d_\psi} - sN_{d_\psi})^T$ and $(M_{d_\phi} - sN_{d_\phi})\breve{I}_{d_\phi+1}$ have full rank. Hence $H_L(s)$ is uniquely determined by (3.25). It is not difficult to check that $\mathbf{t}$ in (3.23) is a solution to the $\big(2rd_{\mathrm{AB}}d_\psi + r(d_\phi + 1)\big) \times \big(2rd_{\mathrm{AB}}d_\psi + r(d_\phi + 1)\big)$ block lower triangular linear system

$$(3.26) \qquad \begin{bmatrix} e_1^T \otimes I_r \\ \breve{\mathbb{I}}_L(\mathbb{A}_L - s\mathbb{B}_L) \end{bmatrix} \begin{bmatrix} \mathbf{t}^{[\phi]} \\ \mathbf{t}^{[\psi]} \end{bmatrix} = \begin{bmatrix} Q^* w^{[0]} \\ \breve{\mathbb{I}}_L \mathbb{B}_L \mathbf{u} \end{bmatrix}.$$

That the matrix of this linear system is nonsingular follows from the same arguments we used to prove nonsingularity of the system (3.25). Hence $\mathbf{t}$ is uniquely determined by (3.26). This presents an alternative to the explicit expression for $\mathbf{t}$ in (3.23). Note that solving (3.26), using a sparse direct linear solver, e.g., is, in general, not cheaper than (3.23), but it is less challenging than working out (3.23).

3.2. Orthogonalization. We now describe how to implement the inner product of $\mathbf{v}$ and $\mathbf{w}$ in (3.18) needed in the Gram–Schmidt orthogonalization in step 4 of Algorithm 1. Since Q has orthonormal columns,

$$\mathbf{v}^*\mathbf{w} = (\mathbf{u}^{[\phi]})^*\mathbf{t}^{[\phi]} + \big([A_0 Q, \ldots, B_{d_{\mathrm{AB}}-1}Q]\mathbf{u}^{[\psi]}\big)^* \big([A_0 Q, \ldots, B_{d_{\mathrm{AB}}-1}Q]\mathbf{t}^{[\psi]}\big),$$

where the second term in the summation disappears when $d_\psi = 0$ (e.g., for the CORK pencil $\mathbf{L}_1$ in (2.11)) but has a nonnegligible cost to compute when $d_\psi \neq 0$. In the latter case we propose to replace the Euclidean inner product $\mathbf{v}^*\mathbf{w}$ with a quasi-inner product that is cheaper to evaluate. We consider both

$$(3.27) \qquad \langle \mathbf{v}, \mathbf{w} \rangle_\phi = (\mathbf{u}^{[\phi]})^*\mathbf{t}^{[\phi]} \quad \text{and} \quad \langle \mathbf{v}, \mathbf{w} \rangle_{\phi,\psi} = \begin{bmatrix} \mathbf{u}^{[\phi]} \\ \mathbf{u}^{[\psi]} \end{bmatrix}^* \begin{bmatrix} \mathbf{t}^{[\phi]} \\ \mathbf{t}^{[\psi]} \end{bmatrix}.$$

These are both quasi-inner products since $\langle \mathbf{v}, \mathbf{v} \rangle_\phi = 0$ does not imply $\mathbf{v} = 0$ and when L is not of full column rank, there exists $\mathbf{u}^{[\psi]} \neq 0$ such that $\mathbf{v} = 0$ but $\langle \mathbf{v}, \mathbf{v} \rangle_{\phi,\psi} \neq 0$. The latter situation should not lead to unlucky breakdown. We compare numerically the two quasi-inner products in (3.27) in section 4.

3.3. The overall algorithm. The compact rational Krylov algorithm is presented in Algorithm 2.

It can be applied to any linearization of the form (2.24), including those discussed in section 2: CORK ($\mathbf{L}_1$), T-CORK ($\mathbf{L}_2$), and TS-CORK ($\mathbf{L}_s$). Similarly to Algorithm 1, Algorithm 2 applied to the transposes of $\mathbb{A}$ and $\mathbb{B}$ returns a basis $\mathbf{Z}$ in compact format for the left Krylov subspace. Note that, with regards to numerical stability, we use reorthogonalization for both the orthogonalization with Euclidean inner product in step 6 and for the orthogonalization with the alternative inner products in steps 13 and 14 [7].

Algorithm 2 requires the storage of Q, and $\mathbf{u}_1, \ldots, \mathbf{u}_{k+1}$. Recall that L is not stored. Assume that the starting vector takes the form (4.1), defined in section 4, which implies $r_1 = 1$. Storing k iteration vectors in the compact format requires $O(nk + k^2(d_\phi + 1 + 2d_{\mathrm{AB}}d_\psi))$ floating point numbers, compared to $O(nk(d_\phi + d_\psi + 1))$ for the full unstructured vector format. Note that the compact format is only beneficial for sufficiently large scale problems. The orthogonalization cost when the $\langle \cdot, \cdot \rangle_\phi$ inner

---

Algorithm 2 Compact Rational Krylov algorithm.

Input: Coefficients of $P(s)$, choice of pencil $\mathbb{A} - s\mathbb{B}$ in (2.24), a starting vector represented by $\mathbf{u}_1$, Q_1 with $Q_1^* Q_1 = I$ and L_1 as in (3.13), a set of k shifts $\sigma_1, \ldots, \sigma_k$, and a (quasi-) inner product $\langle \cdot, \cdot \rangle$ such as $\langle \cdot, \cdot \rangle_\phi$ or $\langle \cdot, \cdot \rangle_{\phi,\psi}$ in (3.27).

1: Normalize $\mathbf{v}_1 = \begin{bmatrix} (I_{d_\phi+1} \otimes Q_1)\mathbf{u}_1^{[\phi]} \\ (I_{d_\psi} \otimes L_1)\mathbf{u}_1^{[\psi]} \end{bmatrix}$ such that $\langle \mathbf{v}_1, \mathbf{v}_1 \rangle = 1$.

2: **for** $j = 1, \ldots, k$ **do**

3:     Form the right-hand side $b = L_j H_{L_j}(\sigma_j)\mathbb{B}_{L_j}\mathbf{u}_j$ with H_L computed using (3.21), (3.22) or (3.25), (3.22).

4:     Solve $P(\sigma_j)w^{[0]} = b$.

5:     **if** $w^{[0]}$ does not lie in the column range of Q_j **then**

6:         Use Gram–Schmidt to orthogonalize $w^{[0]}$ w.r.t. the columns of Q_j.

7:         Expand Q_j: $Q_{j+1} := [Q_j, w^{[0]}]$ so that $Q_{j+1}^* Q_{j+1} = I$.

8:         Expand L_j with $A_i w^{[0]}$ and $B_i w^{[0]}$ to form L_{j+1}. (Note that this is not done explicitly, but corresponds to an expansion of A_L and B_L with additional rows and columns.)

9:         Pad $\mathbf{u}_i$ with zeros so $\mathbf{v}_i^{[\phi]} = (I_{d_\phi+1} \otimes Q_{j+1})\mathbf{u}_i^{[\phi]}$ and $\mathbf{v}_i^{[\psi]} = (I_{d_\psi} \otimes L_{j+1})\mathbf{u}_i^{[\psi]}$, $i = 0, \ldots, j$.

10:     **else**

11:         $Q_{j+1} = Q_j$.

12:     **end if**

13:     Compute $\mathbf{t}$ in (3.23) or as the solution of (3.26).

14:     Compute $h_{ij} = \langle \mathbf{v}_i, \mathbf{w} \rangle$, $i = 1, \ldots, j$.

15:     Orthogonalize $\begin{bmatrix} \tilde{\mathbf{u}}^{[\phi]} \\ \tilde{\mathbf{u}}^{[\psi]} \end{bmatrix} = \begin{bmatrix} \mathbf{t}^{[\phi]} \\ \mathbf{t}^{[\psi]} \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1^{[\phi]}, \ldots, \mathbf{u}_j^{[\phi]} \\ \mathbf{u}_1^{[\psi]} \ldots, \mathbf{u}_j^{[\psi]} \end{bmatrix} h_{1:j,j}$.

16:     Compute $h_{j+1,j} = \sqrt{\langle \widetilde{\mathbf{v}}_{j+1}, \widetilde{\mathbf{v}}_{j+1} \rangle}$, where $\widetilde{\mathbf{v}}_{j+1} = \begin{bmatrix} (I_{d_\phi+1} \otimes Q_{j+1})\tilde{\mathbf{u}}^{[\phi]} \\ (I_{d_\psi} \otimes L_{j+1})\tilde{\mathbf{u}}^{[\psi]} \end{bmatrix}$.

17:     Normalize $\begin{bmatrix} \mathbf{u}_{j+1}^{[\phi]} \\ \mathbf{u}_{j+1}^{[\psi]} \end{bmatrix} = \frac{1}{h_{j+1,j}} \begin{bmatrix} \tilde{\mathbf{u}}^{[\phi]} \\ \tilde{\mathbf{u}}^{[\psi]} \end{bmatrix}$.

18: **end for**

Returns $\mathbf{V}_{k+1}$ in compact form as $\{\mathbf{u}_1, \ldots, \mathbf{u}_{k+1}\}$, Q_{k+1}, and $\underline{H}_k = (h_{ij}) \in \mathbb{C}^{(k+1)\times k}$.

---

product is used is $O(nk^2) + O((d_\phi + 1)k^2)$ floating point operations as for one-sided CORK. For the $\langle \cdot, \cdot \rangle_{\phi,\psi}$ inner product, the cost is $O(nk^2) + O((d_\phi + 1 + 2d_{\mathrm{AB}}d_\psi)k^2)$ floating point operations.

3.4. Computation of the projection. Let $\mathbf{V}$ be the basis in compact form returned by Algorithm 2 applied to $\mathbb{A} - s\mathbb{B}$, and let $\mathbf{Z}$ be the basis in compact form returned by Algorithm 2 when applied to $\mathbb{A}^T - s\mathbb{B}^T$,

$$\mathbf{V} = \begin{bmatrix} I_{d_\phi+1} \otimes Q_V & \\ & I_{d_\psi} \otimes L_V \end{bmatrix} \begin{bmatrix} \mathbf{U}_V^{[\phi]} \\ \mathbf{U}_V^{[\psi]} \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} I_{d_\psi+1} \otimes Q_Z & \\ & I_{d_\phi} \otimes L_Z \end{bmatrix} \begin{bmatrix} \mathbf{U}_Z^{[\psi]} \\ \mathbf{U}_Z^{[\phi]} \end{bmatrix}.$$

Here, note that the bases Ψ and Φ are swapped for $\mathbf{Z}$, since $\mathbf{Z}$ is formed using $\mathbb{A}^T$ and $\mathbb{B}^T$. Then, we have

$$\begin{bmatrix} I_{d_\psi+1} \otimes Q_Z & \\ & I_{d_\phi} \otimes L_Z \end{bmatrix}^* (\mathbb{A} - s\mathbb{B}) \begin{bmatrix} I_{d_\phi+1} \otimes Q_V & \\ & I_{d_\psi} \otimes L_V \end{bmatrix}$$
$$= \begin{bmatrix} I_{d_\psi+1} \otimes Q_Z^* L_V & \\ & I_{d_\phi} \otimes L_Z^* Q_V \end{bmatrix} (\mathbb{A}_L - s\mathbb{B}_L),$$

and therefore

$$\mathbf{Z}^*(\mathbb{A} - s\mathbb{B})\mathbf{V} = \begin{bmatrix} (I_{d_\psi+1} \otimes L_V^* Q_Z)\mathbf{U}_Z^{[\psi]} \\ (I_{d_\phi} \otimes Q_V^* L_Z)\mathbf{U}_Z^{[\phi]} \end{bmatrix}^* (\mathbb{A}_L - s\mathbb{B}_L) \begin{bmatrix} \mathbf{U}_V^{[\phi]} \\ \mathbf{U}_V^{[\psi]} \end{bmatrix}.$$

An efficient way to evaluate $Q_Z^* L_V$ and $L_Z^* Q_V$ is to first compute $\hat{A}_i = Q_Z^* A_i Q_V$ and $\hat{B}_i = Q_Z^* B_i Q_V$.

3.5. Reduction of the size of L. The size of L and, consequently, $\mathbf{u}^{[\psi]}$, can be large, in particular, when d_{AB} is large. We show via examples that the number of nonzero columns of L is usually much smaller than $2rd_{\mathrm{AB}}$ and explain how to reduce the size of L and as a consequence, the length of $\mathbf{u}^{[\psi]}$.

Example 3. Consider the matrix polynomial in the monomial basis from Example 1. The relation between the coefficient matrices P_i, $i = 0, \ldots, d$ and the CORK coefficient matrices A_i, B_i $i = 0, \ldots, d-1$ is given by

$$[A_0, \ldots, A_{d-1}, B_0, \ldots, B_{d-1}] = [P_0, \ldots, P_d](T \otimes I_n)$$
$$\text{with} \quad T = \begin{bmatrix} \hat{I}_{d+1} & -e_{d+1}e_d^T \end{bmatrix} \in \mathbb{R}^{(d+1)\times 2d},$$

where $\hat{I}_{d+1}$ is the $(d+1) \times (d+1)$ identity matrix with its last column removed. For $d = 3$, e.g., we have

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{4\times 6}.$$

For any polynomial of degree d, a minimal formulation consists of at most $d + 1$ different matrices A_i, B_i, so that T has at most $d + 1$ rows. Then

$$L = [A_0, \ldots, A_{d-1}, B_0, \ldots, B_{d-1}](I_{2d} \otimes Q) = [P_0, \ldots, P_d](T \otimes I_n)(I_{2d} \otimes Q)$$
$$= [P_0, \ldots, P_d](I_{d+1} \otimes Q)(T \otimes I_r)$$
$$= L_G(T \otimes I_r),$$

where $L_G = [P_0, \ldots, P_d](I_{d+1} \otimes Q) \in \mathbb{C}^{n \times r(d+1)}$ with $d + 1 < 2d_{\mathrm{AB}}$. The iteration vectors then take the form

$$\mathbf{w} = \begin{bmatrix} (I_{d_\phi+1} \otimes Q)\mathbf{u}^{[\phi]} \\ (I_{d_\psi} \otimes L_G)\mathbf{u}^{[\psi]} \end{bmatrix},$$

where $\mathbf{u}^{[\psi]} \in \mathbb{C}^{r(d+1)d_\psi}$, which can be much shorter in length than $\mathbf{u}^{[\psi]}$ in (3.18). When we use L_G instead of L, we also need to replace A_L and B_L by $A_{L,G}$ and $B_{L,G}$, respectively, such that

$$\mathbf{A}(I_{d_\phi+1} \otimes Q) = (I_{d_\psi+1} \otimes L_G)A_{L,G}, \qquad \mathbf{B}(I_{d_\psi+1} \otimes Q) = (I_{d_\psi+1} \otimes L_G)B_{L,G},$$

where

$$A_{L,G} = (I_{d_\psi+1} \otimes T \otimes I_r)A_L, \qquad B_{L,G} = (I_{d_\psi+1} \otimes T \otimes I_r)B_L.$$

The situation improves even more when the matrix coefficients of $P(s)$ are linear combinations of less than $d+1$ matrices. This is the case when $g(s)P(s)$ is the result of the polynomial or rational approximation of a nonlinear problem of the form (1.2) with $m \ll d$. The following example illustrates this.

Example 4. Consider the matrix polynomial with $m = 3$ written as

$$P(s) = C_0 + (s + 2s^2 + 3s^3)C_1 + s^4C_2.$$

The relation between the coefficient matrices C_i of P and the CORK coefficient matrices in (2.9) is described by $[A_0, A_1, A_2, A_3, B_0, B_1, B_2, B_3] = [C_0, C_1, C_2](T \otimes I_n)$, where

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{3\times8}.$$

In Example 4, the size of L is reduced from $n \times 2rd_{\mathrm{AB}}$ to $n \times rm$ and $\mathbf{u}^{[\psi]}$ is reduced from size $2rd_{\mathrm{AB}}d_\psi$ to rmd_ψ. In a similar way, the gun problem described in (4.3) is defined by the four coefficient matrices K, M, W_1, and W_2. If approximated by a rational function of degree 21, as in section 4.1, the resulting coefficient matrices A_i and B_i are just linear combinations of K, M, W_1, and W_2. Using the technique described above, we can reduce the size of $\mathbf{u}^{[\psi]}$ from $42rd_\psi$ to $4rd_\psi$.

3.6. Implicit filter and numerical stability. Instabilities in the Lanczos method can sometimes be cured by an implicit filter [8]. Implicit filtering is also used for one-sided Krylov methods for reducing the chance of spurious eigenvalues [22], [23]. We recall the idea for the one-sided rational Krylov method and then explain how it is applied to the two-sided method. The idea of implicit filtering for the rational Krylov method is due to [9].

The following theorem shows the case for constant shifts $\sigma_1 = \cdots = \sigma_k$. For the general, but more complicated, case of different shifts, we refer to [9].

THEOREM 3.3. *Let* $\mathbf{V}_{k+1}$, $\underline{H}_k$, $\underline{K}_k$ *satisfy* (3.2). *Define the "skinny" QR factorization* $\underline{H}_k = \underline{Y}_kR_k$ *where* $\underline{Y}_k \in \mathbb{C}^{(k+1)\times k}$ *has orthonormal columns and* $R_k \in \mathbb{C}^{k\times k}$ *is upper triangular. Define*
(3.28)
$$\mathbf{V}_k^{(F)} = \mathbf{V}_{k+1}\underline{Y}_k \in \mathbb{R}^{n\times k}, \quad \underline{H}_{k-1}^{(F)} = R_k(\check{I}_{k+1}^T\underline{Y}_k\check{I}_k) \in R^{k\times(k-1)}, \quad \underline{K}_{k-1}^{(F)} = \underline{H}_{k-1}^{(F)} + \sigma_k\underline{I}_k.$$

Then, we have that both $\underline{H}_{k-1}^{(F)}$ *and* $\underline{K}_{k-1}^{(F)}$ *are upper Hessenberg matrices and*

$$\mathbb{A}\mathbf{V}_k^{(F)}\underline{H}_{k-1}^{(F)} = \mathbb{B}\mathbf{V}_k^{(F)}\underline{K}_{k-1}^{(F)}$$

is a Rational Krylov recurrence relation, of order $k - 1$. *Moreover,*

$$\mathrm{Range}(\mathbf{V}_k^{(F)}) = \mathrm{Range}((\mathbb{A} - \sigma_k\mathbb{B})^{-1}\mathbb{B}\mathbf{V}_k).$$

The filtered basis $\mathbf{V}_k^{(F)}$ spans the Krylov space obtained by multiplying the old Krylov space $\mathbf{V}_k$ with $(\mathbb{A} - \sigma_1\mathbb{B})^{-1}\mathbb{B}$. This operation removes the small eigenvalues from the Krylov space. Their Ritz values are usually not good approximations of eigenvalues and may lead to spurious eigenvalues. Therefore, their removal by the filter may improve the numerical behavior of the Krylov method.

In the context of TS-CORK, the new iteration vectors take the form

$$\mathbf{V}_k^{(F)} = \begin{bmatrix} (I_{d_\phi+1} \otimes Q)\mathbf{U}^{[\phi]}\underline{Y}_k \\ (I_{d_\psi} \otimes L)\mathbf{U}^{[\psi]}\underline{Y}_k \end{bmatrix},$$

where Q has rank $r \leq d_\phi + k + 1$. Since they also satisfy a recurrence relation of order $k-1$, we know that there are $Q^{(F)}$, $L^{(F)}$, $(\mathbf{U}^{[\phi]})^{(F)}$, and $(\mathbf{U}^{[\psi]})^{(F)}$ such that

$$\mathbf{V}_k^{(F)} = \begin{bmatrix} (I_{d_\phi+1} \otimes Q^{(F)})(\mathbf{U}^{[\phi]})^{(F)} \\ (I_{d_\psi} \otimes L^{(F)})(\mathbf{U}^{[\psi]})^{(F)} \end{bmatrix},$$

with rank of $Q^{(F)}$ bounded from above by $d_\phi + k$. It may thus happen that $Q^{(F)}$ has smaller rank than Q. As a result, $L^{(F)}$ may also have lower rank than L since it is intimately connected to $Q^{(F)}$. Algorithm 3 shows the different steps.

Algorithm 3 Implicit filter for P-CORK/TS-CORK.

Input: $\underline{H}_k$, $\underline{K}_k$, and $\mathbf{V}_{k+1}$ in compact form, i.e., Q and $\mathbf{U}^{[\phi]}, \mathbf{U}^{[\psi]}$.

1: Compute the QR factorization of $\underline{H}_k$: $\underline{H}_k = \underline{Y}_k R_k$.
2: Update $\underline{H}_{k-1}^{(F)}$ and $\underline{K}_{k-1}^{(F)}$ following (3.28).
3: Form $(\mathbf{U}^{[\phi]})^{(F)} = \mathbf{U}^{[\phi]}\underline{Y}_k$ and $(\mathbf{U}^{[\psi]})^{(F)} = \mathbf{U}^{[\psi]}\underline{Y}_k$.
4: Reshape these matrices as follows:
5: $\mathbb{U} = [(u_1^{[0]})^{(F)}, \ldots, (u_k^{[0]})^{(F)}, \ldots, (u_1^{[d_\psi+d_\phi]})^{(F)}, \ldots, (u_k^{[d_\psi+d_\phi]})^{(F)}]$.
6: Compute the singular values decomposition of $\mathbb{U} = \mathbb{V}\Sigma\mathbb{W}^*$ and truncate the small singular values and associated right and left singular vectors.
7: Form $Q^{(F)} = Q\mathbb{V}$.
8: Replace $(\mathbf{U}^{[\phi]})^{(F)}$ by $\mathbb{V}^*(\mathbf{U}^{[\phi]})^{(F)}$ and $(\mathbf{U}^{[\psi]})^{(F)}$ by $\mathbb{V}^*(\mathbf{U}^{[\psi]})^{(F)}$.
9: If the rank of $Q^{(F)}$ is lower than the rank of Q, drop associated rows and columns of A_L and B_L.

Return: $\underline{H}_{k-1}^{(F)}$, $\underline{K}_{k-1}^{(F)}$, and $\mathbf{V}_k^{(F)}$ in compact form, i.e., $Q^{(F)}$ and $(\mathbf{U}^{[\phi]})^{(F)}, (\mathbf{U}^{[\psi]})^{(F)}$.

For the two-sided method, we apply the algorithm on both right and left subspaces. This produces new $\mathbf{V}$ and $\mathbf{Z}$.

4. Numerical examples. In this section, we illustrate the use of the right and left Krylov spaces generated by Algorithm 2 for computing eigenvalue and eigenvector approximations. We compare the following methods:

CORK: the compact rational Krylov method applied to $\mathbf{L}_1$ as described in [34]. This also corresponds to Algorithm 2 applied to $\mathbf{L}_1$ with the standard (Euclidean) inner product denoted by $\langle \cdot, \cdot \rangle_f$, which does not require additional products with A_i and B_i, since $t^{[\psi]}$ is empty. The Ritz values are obtained from (3.3).

T-CORK: Algorithm 2 applied to $\mathbf{L}_2 = \mathbf{L}_1^T$. The Ritz values are again obtained from (3.3). The iteration vectors cannot be fully orthogonalized without additional products with A_i, B_i. We compare the convergence behavior of Ritz values when using the inner products $\langle \cdot, \cdot \rangle_\phi$, $\langle \cdot, \cdot \rangle_{\phi,\psi}$ and also the standard (expensive) inner product $\langle \cdot, \cdot \rangle_f$. Note that $\langle \cdot, \cdot \rangle_\phi$ only uses the first block, $v_j^{[0]}$, of iteration vector $\mathbf{v}_j$.

RS-CORK: Algorithm 2 applied to $\mathbf{L}_S$. The Ritz values are obtained from (3.3). Both inner products $\langle \cdot, \cdot \rangle_\phi$ and $\langle \cdot, \cdot \rangle_{\phi,\psi}$, are compared. In contrast to T-CORK, $\langle \cdot, \cdot \rangle_\phi$ uses the upper half of the iteration vectors.

LS-CORK: The same as RS-CORK, but Algorithm 2 is applied to $\mathbb{A}^T$ and $\mathbb{B}^T$ of $\mathbf{L}_S = \mathbb{A} - s\mathbb{B}$.

P-CORK: This is the two-sided rational Krylov method, explained in the beginning of section 3, where $\mathbf{L}_1$ is projected on the left and right Krylov spaces constructed by CORK and T-CORK, respectively. This method produces Ritz triplets, i.e., Ritz values and associated right and left Ritz vectors, computed from the projected pencil (3.4).

TS-CORK: This is the two-sided rational Krylov method applied to $\mathbf{L}_S$, where the right and left Krylov spaces are computed with RS-CORK and LS-CORK, respectively. This method produces Ritz triplets, i.e., Ritz values and associated right and left Ritz vectors, computed from the projected pencil (3.4).

As a starting vector we use a vector in the low rank format

$$\mathbf{v}_1 = \begin{bmatrix} f \otimes v_1 \\ \mathbf{0}_{nd_\psi} \end{bmatrix}, \tag{4.1}$$

where $f \in \mathbb{C}^{d_\phi+1}, v_1 \in \mathbb{C}^n \setminus \{0\}$, both with two-norm 1, so that $Q = [v_1] \in \mathbb{C}^{n \times 1}$ and $\mathbf{u}^{[\phi]} = f$ and $\mathbf{u}^{[\psi]} = 0$. In our experiments, we chose $f = e_1$.

In the numerical examples, convergence will be measured by the backward errors on the right and left eigenpair estimates, defined as the scaled residual norms

$$R_x = \frac{\|A(\lambda)x\|_2}{\|A(\lambda)\|_1\|x\|_2}, \qquad R_y = \frac{\|y^*A(\lambda)\|_2}{\|A(\lambda)\|_1\|y\|_2}, \tag{4.2}$$

where x and y correspond to the first n-sized block of the Ritz vectors of the linear pencil.

4.1. Gun problem. Consider the model of a radio-frequency gun cavity available from the NLEVP collection [5], known as the `gun` problem. This problem has the following matrix:

$$A(\lambda) = K - \lambda M + i\sqrt{(\lambda - \sigma_1^2)}W_1 + i\sqrt{(\lambda - \sigma_2^2)}W_2. \tag{4.3}$$

See [20] for the details. The problem is of size $n = 9956$, and $\sigma_1 = 0$, $\sigma_2 = 108.8774$. Since the coefficient matrices are symmetric, we can use $\mathbf{Z} = \overline{\mathbf{V}}$, i.e., a two-sided method is strictly speaking not needed to solve this problem. We use this example to compare the reliability of the CORK, T-CORK, and RS-CORK methods for computing eigenvalues and right eigenvectors using (3.3).

We determine eigenpairs of the nonlinear eigenvalue problem, whose eigenvalues are nearest to 250^2. As in [17] and [34], the eigenvalue problem is approximated by a rational eigenvalue problem expressed in a rational Newton basis of degree 21 and five shifts are used, one at 250^2 and the other four around this point in the complex plane.

First, the impact on the convergence of the alternative inner products is numerically checked by comparing T-CORK using $\langle \cdot, \cdot \rangle_\phi$ and $\langle \cdot, \cdot \rangle_{\phi,\psi}$, with T-CORK using the Euclidean inner product $\langle \cdot, \cdot \rangle_f$. We further compare RS-CORK using $\langle \cdot, \cdot \rangle_\phi$ and $\langle \cdot, \cdot \rangle_{\phi,\psi}$ with CORK. Figure 4.1 shows the residual norms as a function of the iteration count. From top to bottom, the convergence is shown for three specific eigenvalues, namely those three that show fastest convergence for the CORK method. For the other methods, convergence is shown for the values that match best these three Ritz values. Both T-CORK and RS-CORK perform well with $\langle \cdot, \cdot \rangle_{\phi,\psi}$. It is clear that
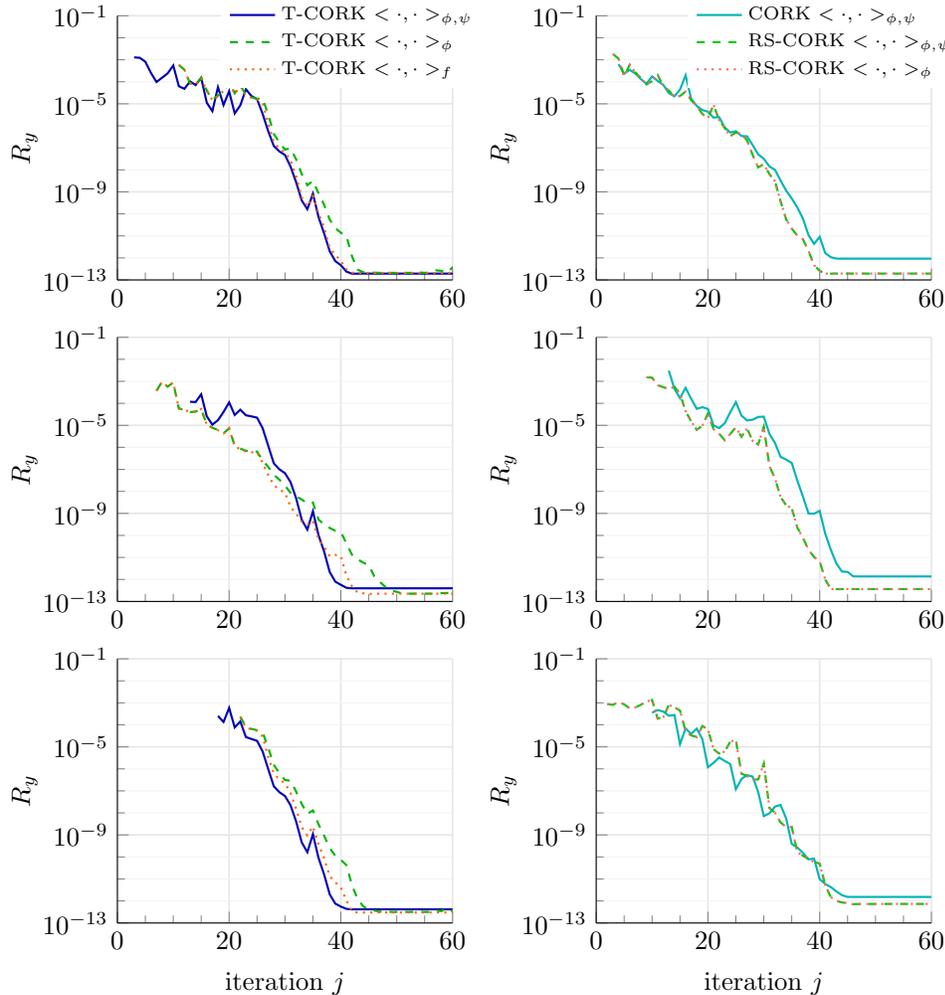
FIG. 4.1. *The residual norms R_x and R_y for three eigenvalues of the gun problem as a function of the iteration count j. Each row corresponds to one eigenvalue.*

T-CORK does not produce small residual norms when $\langle \cdot, \cdot \rangle_\phi$ is used, but RS-CORK performs well. The convergence behaviour of T-CORK is the same for the Euclidean inner product $\langle \cdot, \cdot \rangle_f$ and $\langle \cdot, \cdot \rangle_{\phi,\psi}$ for this example so there is no advantage in using the more expensive Euclidean inner product.

Figure 4.2 illustrates the performance of the symmetrically structured pencil by comparing R_y for T-CORK and LS-CORK for a large number of Ritz values. Both approaches perform comparably well.

The left side of Figure 4.3 shows the memory cost in megabytes for storing the iteration vectors for the rational Krylov method without exploiting the structure of the iteration vectors and the memory use of TS-CORK using the compact representation (3.18). We also compared with a method that we call TS-CORK-R. This latter method uses L_G instead of L, as discussed in section 3.5. The difference lies in the cost of storing $\{\mathbf{u}_1, \ldots, \mathbf{u}_{k+1}\}$. Since $m = 4$ and $d = 21$, the size of $\mathbf{u}_j$ is $211j^2$ without reduction, compared to $51j^2$ for TS-CORK-R. The memory savings become
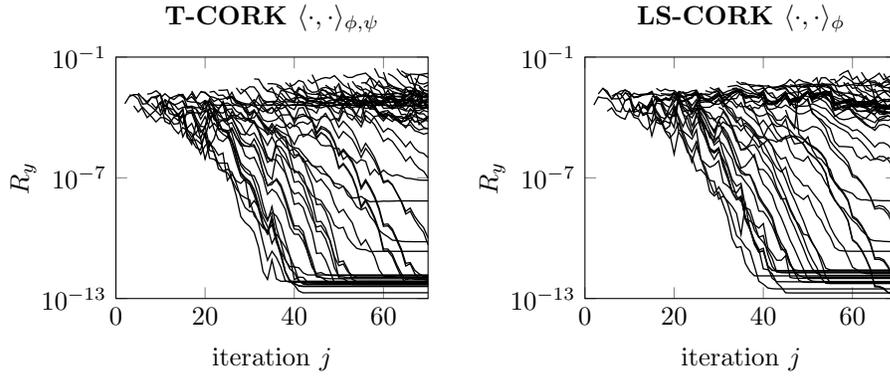
T-CORK $\langle \cdot, \cdot \rangle_{\phi, \psi}$ **LS-CORK** $\langle \cdot, \cdot \rangle_{\phi}$

FIG. 4.2. *All residual norms for the gun problem. (Left) T-CORK with inner product $\langle \cdot, \cdot \rangle_{\phi,\psi}$. (Right) LS-CORK with inner product $\langle \cdot, \cdot \rangle_{\phi}$.*

Gun problem   **DDE problem**
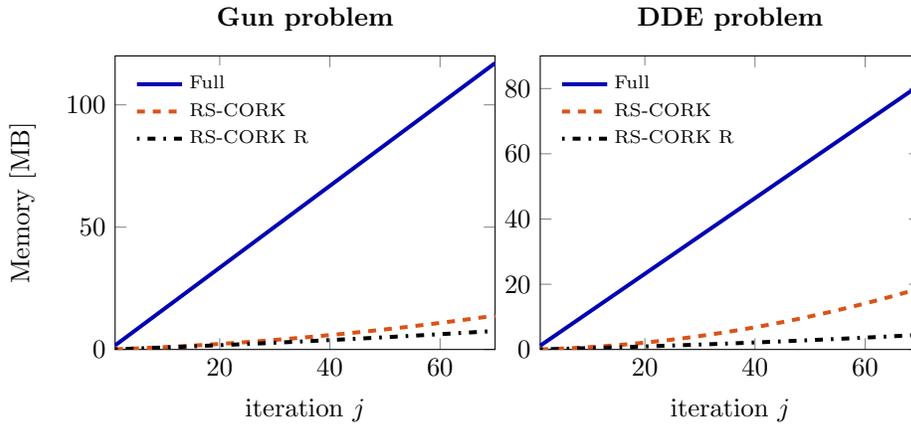
FIG. 4.3. *Memory usage in MB for storing the iteration vectors. "RS-CORK R" stands for the reduction of L to L_G as explained in section 3.5. (Left) For the gun problem. (Right) For the DDE problem.*

Gun problem   **DDE problem**
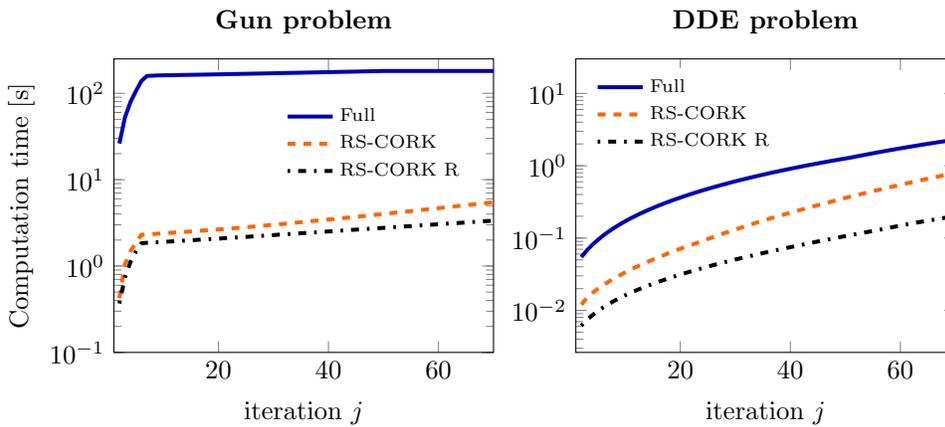
FIG. 4.4. *Computation time as a function of the number of iterations for the full pencil, RS-CORK and RS-CORK R (where L is replaced by L_G; see section 3.5). (Left) For the gun problem. (Right) For the DDE problem.*

more pronounced for a higher number of iterations.
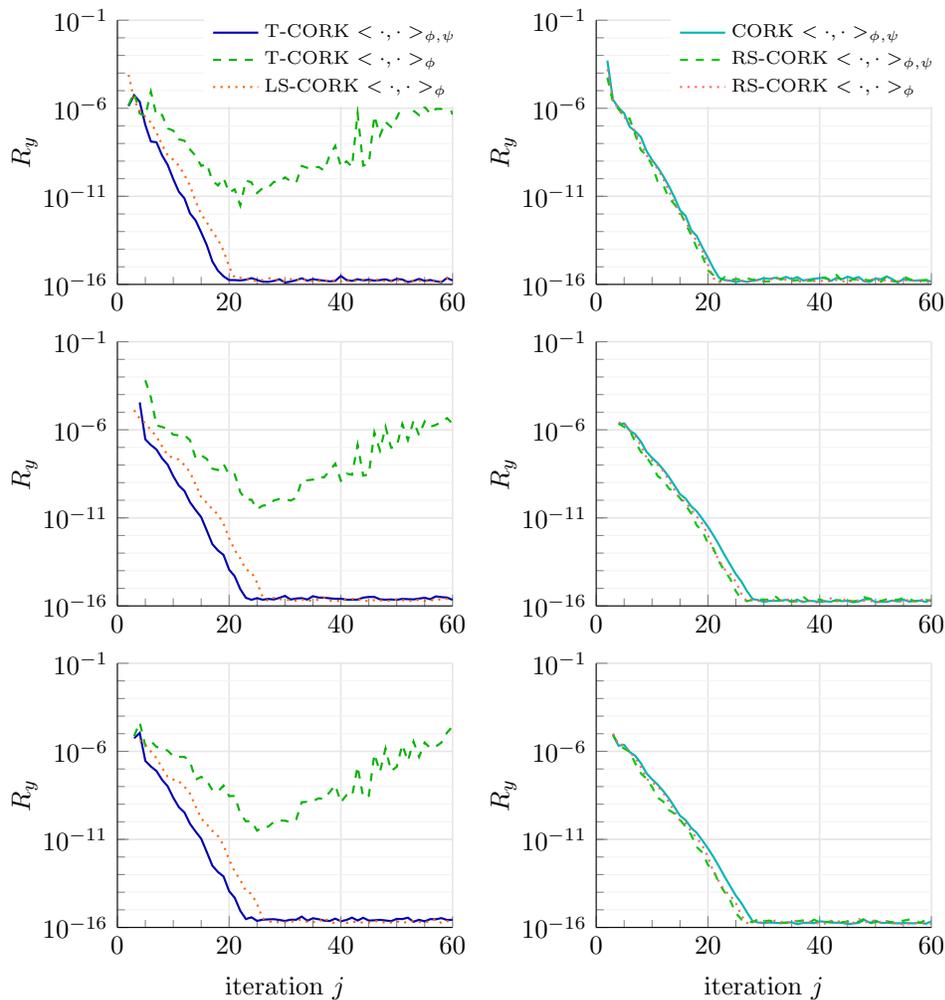
FIG. 4.5. *The residual norms R_x and R_y for the three fastest converging Ritz values. (Left) T-CORK and LS-CORK. (Right) CORK and RS-CORK.*

Finally, the left side of Figure 4.4 shows the computation time[1] as a function of the number of iterations for a direct computation, without exploitation of the structure and for RS-CORK and RS-CORK-R in the reduced format of section 3.5. Note the logarithmic scale. The first iterations are dominated by costly sparse LU-factorizations. This is especially true for the full case, where sparse factorizations of the size 209,076 pencil are required, compared to the compact CORK case where only LU factorizations of original size 9,956 are computed. For the remaining iterations, up to $j = 70$, the increase in computation time is considerably less, since the dominating costs are only the operations on the sparse matrices.

---

[1]Using MATLAB 2017b, averaged over 3 runs, computations performed on a machine with a 64-bit Intel processor, 4 cores, 2.5 Ghz processors, and 8 GB RAM.
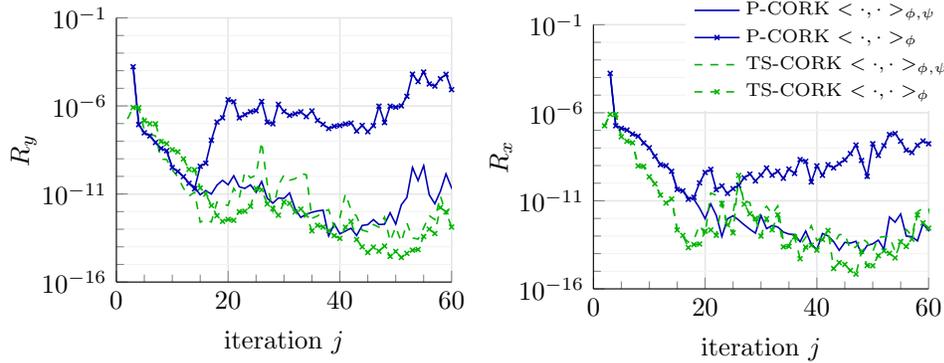
FIG. 4.6. *The residual norms R_x and R_y for the left and right Ritz vectors for the fastest converging Ritz values for P-CORK and TS-CORK without filtering. (Left) Residual norms for the left eigenvectors. (Right) Residual norms for the right eigenvectors.*

4.2. Time delay problem. We consider the partial differential equation with a delay term in the right-hand side, based on [34], which we refer to as the DDE-problem,

$$\frac{\partial^2 v(x,t)}{\partial t^2} = \frac{\partial^2 v(x,t)}{\partial x^2} + a_0(x)v(x,t) + a_1(x)v(\pi/2 - x, t - 1),$$

with $a_0(x) = -2\sin(x)$, $a_1(x) = 2\sin(x)$. We choose $x \in [0, \pi/2]$ so that the spatial discretization of the $a_1(x)v(\pi/2 - x, t - 1)$-term results in a nonsymmetric matrix. The boundary conditions are $v_x(0, t) = v_x(\pi/2, t) = 0$. After discretization with central differences with $n = 5000$, as in [12], the associated second order delay eigenvalue problem becomes

$$A(\lambda)x = (C_0 - \lambda^2 I_n + C_1 e^{-\lambda})x = 0,$$

where $C_0, C_1 \in \mathbb{C}^{5000 \times 5000}$. We use a Taylor polynomial around $\lambda = 0$ for approximating the exponential as in [18]:

$$A(\lambda) \approx P(\lambda) = (C_0 + C_1) + (C_0 - C_1)\lambda + \left(-I_n + \frac{1}{2}C_1\right)\lambda^2 + \sum_{j=3}^{d} \frac{1}{j!}(-1)^j C_1 \lambda^j.$$

We are interested in the interval $[-5, 5]$. For $d = 29$, the relative approximation error for the Taylor polynomial at $\lambda = -5$, given by $\frac{e^5 5^{30}}{30!\|A(-5)\|_1}$, is around machine precision. We now perform three experiments for the polynomial eigenvalue problem of degree 29.

First, we compare the CORK, T-CORK, RS-CORK, and LS-CORK methods. Figure 4.5 shows the residual norms of the left (for T-CORK and LS-CORK) and right Ritz vectors (for CORK and RS-CORK) as a function of the iteration count for the three fastest converging eigenvalues in these methods. The experiment leads to the same conclusions as for the `gun` problem. T-CORK performs well with the $\langle \cdot, \cdot \rangle_{\phi, \psi}$ inner product, but not with the $\langle \cdot, \cdot \rangle_\phi$ inner product. RS-CORK and LS-CORK perform well with both inner products.

Second, we use P-CORK and TS-CORK with the $\langle \cdot, \cdot \rangle_\phi$ and $\langle \cdot, \cdot \rangle_{\phi, \psi}$ inner products to compute the left and right eigenvectors simultaneously. Figure 4.6 shows the
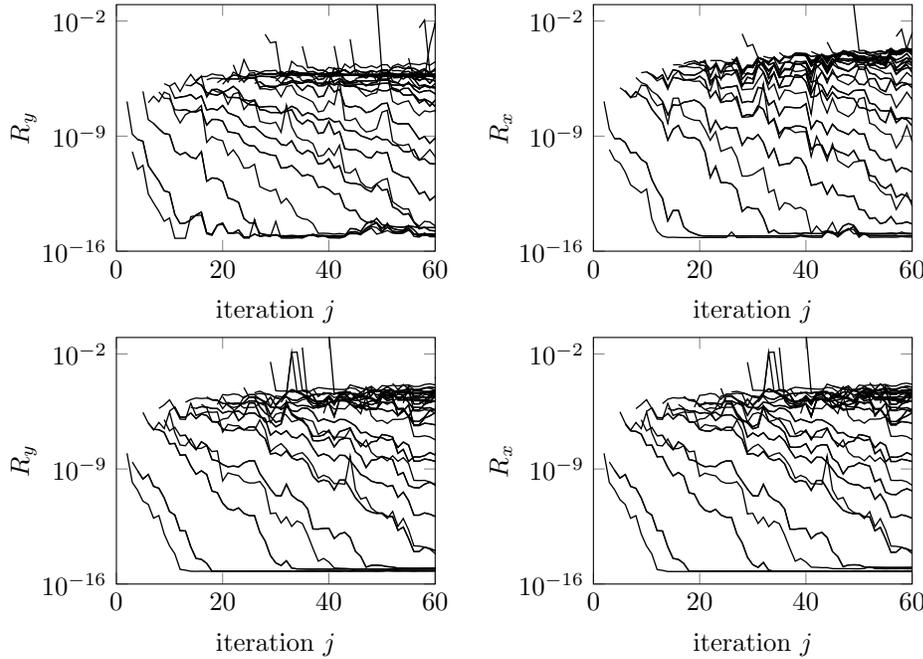
FIG. 4.8. *The residual norms of all Ritz values for the DDE problem. (Top) P-CORK with* $\langle \cdot, \cdot \rangle_{\phi,\psi}$ *inner product. (Bottom) TS-CORK with* $\langle \cdot, \cdot \rangle_{\phi}$ *inner product. (Left) Left eigenvectors. (Right) Right eigenvectors.*

nonlinear term in $A(\lambda)$, so that $m = 3$ in (1.2), which leads to a significant gain in memory use of LS-CORK-R, with iteration vectors $\mathbf{u}_j$ of size $43j^2$, compared to $407j^2$ for LS-CORK.

The computation time is shown in Figure 4.4. Since the cost of the solver with sparse matrices is relatively low for this example and the degree d is relatively high, the computational cost is dominated by the manipulation of the iteration vectors. As for the memory use, the difference in computation time increase between RS-CORK and RS-CORK-R is more pronounced for this example.

5. Conclusions. We presented a generalization of the compact rational Krylov method (CORK) that can be applied to the transpose of the CORK linearizations and the symmetrically structured linearizations of Robol, Vandebril, and Van Dooren [27]. This generalization allows us to efficiently implement a two-sided rational Krylov method with the aim of computing right and left eigenvectors. To achieve this, we introduced a more general pencil structure that included both CORK linearizations and symmetrically structured linearizations. We showed that the iteration vectors can be represented in a compact form as for CORK. This leads to an efficient expansion step of the Krylov space with the same efficiency as the CORK method.

There are, however, two downsides. First, the storage of the compact structure is considerably more expensive. We have shown that for our numerical applications, however, the cost can be dramatically reduced by exploiting the fact that the number of terms, m, in (1.2) is very small. Second, the inner product of vectors with this compact structure also requires matrix vector products with the coefficient matrices. We have shown two alternative inner products that do not require products with the

coefficient matrices. The inner product $\langle \cdot, \cdot \rangle_{\phi, \psi}$ appears to perform as well as the standard inner product for our numerical experiments with one-sided methods.

The two-sided (rational) Krylov method in its simplest form does not produce reliable results: the residual norms for both left and right Ritz vectors show erratic behavior. The application of implicit filtering for every ten iterations improves the method considerably. We did not observe significant differences in convergence behavior between P-CORK and TS-CORK. We found the methods to be sensitive to the choice of inner product. As for one-sided methods, the $\langle \cdot, \cdot \rangle_{\phi, \psi}$ inner product is more reliable for P-CORK. For TS-CORK, we found the behavior is problem dependent. We expected the symmetrically structured linearizations to have an advantage for two-sided methods, but our numerical results do not agree with this hypothesis. Since the construction of the polynomial expressions is more involved than for CORK linearizations and their transposes, we suggest using the latter instead (i.e., $\mathbf{L}_1$ and $\mathbf{L}_2$).

This paper did not show how to incorporate implicit restarting as for one-sided methods. The two-sided Krylov Schur restart from [36] can be applied using a procedure which is close to the implicit filtering of section 3.6.

Acknowledgments. We sincerely thank the reviewers for their extensive effort that helped considerably improve the quality of this paper.

REFERENCES

[1] A. Amiraslani, R. M. Corless, and P. Lancaster, *Linearization of matrix polynomials expressed in polynomial bases*, IMA J. Numer. Anal., 29 (2009), pp. 141–157.

[2] A. Amparan, F. Dopico, S. Marcaida, and I. Zaballa, *Strong Linearizations of Rational Matrices*, MIMS EPrint 2010.51, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2016.

[3] E. N. Antoniou and S. Vologiannidis, *A new family of companion forms of polynomial matrices*, Electron. J. Linear Algebra, 11 (2004), pp. 78–87.

[4] Z. Bai, D. Day, and Q. Ye, *ABLE: An adaptive block Lanczos method for non-Hermitian eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1060–1082, https://doi.org/10.1137/S0895479897317806.

[5] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, ACM Trans. Math. Software, 39 (2013), 7.

[6] R. M. Corless, *On a generalized companion matrix pencil for matrix polynomials expressed in the Lagrange basis*, Birkhäuser, Basel, 2007, pp. 1–15.

[7] J. Daniel, W. Gragg, L. Kaufman, and G. Stewart, *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[8] G. De Samblanx and A. Bultheel, *Nested Lanczos : Implicitly restarting a Lanczos algorithm*, Numer. Algorithms, 18 (1998), pp. 31–50.

[9] 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.

[10] C. Effenberger and D. Kressner, *Chebyshev interpolation for nonlinear eigenvalue problems*, BIT, 52 (2012), pp. 933–951.

[11] J. G. David Forney, Jr., *Minimal bases of rational vector spaces, with applications to multivariable linear systems*, SIAM J. Control, 13 (1975), pp. 493–520, https://doi.org/10.1137/0313029.

[12] S. W. Gaaf and E. Jarlebring, *The infinite bi-Lanczos method for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 39 (2017), pp. S898–S919, https://doi.org/10.1137/16M1084195.

[13] F. R. Gantmacher, *The Theory of Matrices*, Vol. 1, Chelsea Publishing, New York, 1959.

[14] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Classics Appl. Math. 58, SIAM, Philadelphia, 2009.

[15] L. Grammont, N. J. Higham, and F. Tisseur, *A framework for analyzing nonlinear eigenproblems and parametrized linear systems*, Linear Algebra Appl., 435 (2011), pp. 623–640.

[16] S. GÜTTEL AND F. TISSEUR, *The nonlinear eigenvalue problem*, Acta Numer., 26 (2017), pp. 1–94.

[17] S. GÜTTEL, R. VAN BEEUMEN, K. MEERBERGEN, AND W. MICHIELS, *NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 36 (2014), pp. A2842–A2864, https://doi.org/10.1137/130935045.

[18] E. JARLEBRING, W. MICHIELS, AND K. MEERBERGEN, *A linear eigenvalue algorithm for the nonlinear eigenvalue problem*, Numer. Math., 122 (2012), pp. 169–195.

[19] P. W. LAWRENCE AND J. PÉREZ, *Constructing strong linearizations of matrix polynomials expressed in Chebyshev bases*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 683–709, https://doi.org/10.1137/16M105839X.

[20] B.-S. LIAO, Z. BAI, L.-Q. LEE, AND K. KO, *Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems*, Taiwanese J. Math., 14 (2010), pp. 869–883.

[21] D. LU, Y. SU, AND Z. BAI, *Stability analysis of the two-level orthogonal Arnoldi procedure*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 195–214, https://doi.org/10.1137/151005142.

[22] K. MEERBERGEN, *The Lanczos method with semi-inner product*, BIT, 41 (2001), pp. 1069–1078.

[23] K. MEERBERGEN AND A. SPENCE, *Implicitly restarted Arnoldi and purification for the shift-invert transformation*, Math. Comp., 66 (1997), pp. 667–689.

[24] Y. NAKATSUKASA AND F. TISSEUR, *Eigenvector Error Bound and Perturbation for Polynomial and Rational Eigenvalue Problems*, Mathematical Engineering Technical Reports METR 2016-04, Department of Mathematical Informatics, The University of Tokyo, Japan, 2016.

[25] M. L. OVERTON, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120, https://doi.org/10.1137/0802007.

[26] J. PEETERS AND W. MICHIELS, *Computing the $H2$ norm of large-scale time-delay systems*, in Proceedings of the IFAC Joint Conference, vol. 11, Grenoble, 2013, pp. 114–119.

[27] L. ROBOL, R. VANDEBRIL, AND P. VAN DOOREN, *A framework for structured linearizations of matrix polynomials in various bases*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 188–216, https://doi.org/10.1137/16M106296X.

[28] A. RUHE, *The two-sided Arnoldi algorithm for nonsymmetric eigenvalue problems*, Lecture Notes in Math. 973, Springer, Berlin, Heidelberg, 1983, pp. 104–120.

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

[30] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Algorithms and Architectures for Advanced Scientific Computing, Manchester University Press, Manchester, UK, 1992.

[31] A. SHAPIRO AND M. K. H. FAN, *On eigenvalue optimization*, SIAM J. Optim., 5 (1995), pp. 552–569, https://doi.org/10.1137/0805028.

[32] F. TISSEUR, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.

[33] R. VAN BEEUMEN, K. MEERBERGEN, AND W. MICHIELS, *A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 35 (2013), pp. A327–A350, https://doi.org/10.1137/120877556.

[34] R. VAN BEEUMEN, K. MEERBERGEN, AND W. MICHIELS, *Compact rational Krylov methods for nonlinear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 820–838, https://doi.org/10.1137/140976698.

[35] R. VAN BEEUMEN, W. MICHIELS, AND K. MEERBERGEN, *Linearization of Lagrange and Hermite interpolating matrix polynomials*, IMA J. Numer. Anal., 35 (2015), pp. 909–930.

[36] I. N. ZWAAN AND M. E. HOCHSTENBACH, *Krylov–Schur-type restarts for the two-sided Arnoldi method*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 297–321, https://doi.org/10.1137/16M1078987.