Automatic rational approximation and linearization of nonlinear eigenvalue problems

PIETER LIETAERT †, KARL MEERBERGEN †, JAVIER PÉREZ ‡ AND BART VANDEREYCKEN §

We present a method for solving nonlinear eigenvalue problems using rational approximation. The method uses the AAA algorithm of Nakatsukasa, Sète, and Trefethen to approximate the nonlinear eigenvalue problem via a rational eigenvalue problem. A set-valued variant of the AAA algorithm is also presented for building low-degree rational approximations of nonlinear eigenvalue problems with a large number of nonlinear functions. The rational approximation is embedded in the state-space representation of a rational polynomial by Su and Bai. This procedure perfectly fits the framework of the compact rational Krylov methods (CORK and TS-CORK), allowing to efficiently solve large-scale nonlinear eigenvalue problems. One advantage of our method, compared to related techniques such as NLEIGS and infinite Arnoldi, is that it automatically selects the poles and zeros of the rational approximations. Numerical examples show that the presented framework is competitive with NLEIGS and usually produces smaller linearizations with the same accuracy but with less effort for the user.

Keywords: Nonlinear eigenvalue problem, rational interpolation, rational Krylov method

1. Introduction

The nonlinear eigenvalue problem (NEP) is the problem of finding scalars $\lambda \in \Sigma \subseteq \mathbb{C}$ and nonzero vectors $x, y \in \mathbb{C}^n$ such that

$$A(\lambda)x = 0 \quad \text{and} \quad y^* A(\lambda) = 0, \tag{1.1}$$

where $A\colon \Sigma \to \mathbb{C}^{n\times n}$ is a nonlinear matrix-valued function. The scalar $\lambda \in \mathbb{C}$ is called an eigenvalue and the vectors x and y are called, respectively, associated right and left eigenvectors. Usually, one is interested in the eigenvalues in a specific region $\Sigma \subset \mathbb{C}$. We assume that A in (1.1) is regular, i.e., there is at least one $\lambda \in \mathbb{C}$ for which $\det(A(\lambda)) \neq 0$. It is also standard to assume that A is holomorphic in Σ, as this avoids numerous difficulties. However, we want to mention that the algorithm proposed in this work can deal with some non-analytical situations (see the numerical experiment in Section 4.3). For a thorough treatment of nonlinear eigenvalue problems, we refer the reader to the recent survey (Güttel & Tisseur, 2017) for most of the concepts mentioned in this introduction.

Numerical methods for computing eigenvalues of generic nonlinear eigenvalue problems in a region Σ are based on approximation theory or Newton's method. There currently are two classes of methods based on approximation theory: those based on contour integrals (Beyn, 2012), and those based on rational and polynomial approximation, e.g. Effenberger & Kressner (2012), infinite Arnoldi (Jarlebring *et al.*, 2012) and NLEIGS (Güttel *et al.*, 2014). The methods based on contour integration rely on

[†] Department of Computer Science, KU Leuven, University of Leuven, 3001 Heverlee, Belgium. Supported by KU Leuven Research Council Grant OT/14/074 and BELSPO Grant IAP VII/19, initiated by the Belgian State Science Policy Office.

[‡] Department of Mathematical Sciences, University of Montana, MT, USA. The research of J. Pérez was partially supported by KU Leuven Research Council grant OT/14/074 and BELSPO Grant IAP VII/1.

[§] Section of Mathematics, University of Geneva, Rue du Lièvre 2-4, 1211 Geneva, Switzerland. Partially supported by SNF project 169115.

Keldysh' theorem, where the eigenvalues in Σ are found as the poles of a resolvent using contour integration. There are several variations of this approach; see Beyn (2012). Another example comes from Effenberger (2013), where invariant pairs of large-scale NEPs are computed by combining the Jacobi–Davidson method with contour integration. The second class of methods approximates A by a polynomial or rational function on Σ and solves the resulting polynomial or rational eigenvalue problem. Polynomial and rational eigenvalue problems can be solved by Krylov methods through a linearization. The prototype linearization is the companion pencil.

In this paper, we focus on methods that build a rational approximation, reformulate the resulting problem as a linear eigenvalue problem (by the process of linearization) and then use a Krylov method for solving the linearized problem. Roughly speaking, there exist three approaches to rational approximation. The NLEIGS method uses potential theory for the selection of poles and interpolation points, and embeds this within a rational polynomial expressed in a rational Newton basis (see Güttel *et al.*, 2014). The second is the infinite Arnoldi method (Jarlebring *et al.*, 2012) that uses the discretization of an infinite dimensional operator that is a linear representation of the nonlinear eigenvalue problem. The discretization of this operator leads to a finite dimensional linear problem that is solved by the Arnoldi method. The third approach expresses a Padé approximation in state-space form and applies a Krylov method to a linearization; see Su & Bai (2011).

The approach of this paper is inspired by Su & Bai (2011) and assumes that the matrix-valued function A can be written as

$$A(\lambda) = P(\lambda) + G(\lambda), \tag{1.2}$$

where $P(\lambda)$ is an $n \times n$ matrix polynomial or rational matrix, and G is an arbitrary matrix-valued function of the form

$$G(\lambda) = \sum_{i=1}^{s} (C_i - \lambda D_i) g_i(\lambda), \tag{1.3}$$

where C_i, D_i are constant $n \times n$ matrices and $g_i \colon \Sigma \to \mathbb{C}$ is a nonlinear function. Each g_i can be approximated by a different rational function with different poles and interpolation points, determined independently from each other. For efficiency, it is, therefore, assumed in this paper that the number of nonlinear terms, s, is modest. The NLEIGS method by Güttel *et al.* (2014) does not have this limitation since it uses the same poles and nodes for all terms.

The contribution of this paper is threefold. First, the rational approximations used in this work are obtained by employing the adaptive Antoulas–Anderson (AAA) algorithm introduced by Nakatsukasa *et al.* (2018). This approach presents two key advantages: the AAA algorithm is not domain-dependent and it works effectively even with sets consisting of disconnected regions of irregular shape; and once the approximation region has been fixed, it is the algorithm and not the user that chooses in an adaptive way the number of poles and zeros, and their values, of the rational approximation. Hence, unlike NLEIGS by Güttel *et al.* (2014), neither special knowledge of the nonlinear functions nor advanced knowledge of complex analysis is required from the user.

The second contribution is an automatic strategy that uses the same poles and interpolation points for (a subset of) all g_i, which leads to the same appealing properties as NLEIGS, when s is not so small. Our numerical experiments compare to rational approximations obtained using NLEIGS in Güttel *et al.* (2014). We mention that a similar idea to adapt the AAA algorithm for multiple nonlinear functions interpolation has already been independently investigated in Hochman (2017), using an updated Cholesky factor of a Gram matrix, where we use an updated QR factorization. An alternative to AAA, that is not considered in the paper, is Vector Fitting: it builds an approximation in barycentric form by iteratively determining support points and weights in order to minimize the approximation error (Gustavsen &

Semlyen, 1999), where AAA minimizes the linearized residual and chooses the support points using a greedy optimization method.

The third contribution of the paper lies in the observation that the linearization from Su & Bai (2011) fits perfectly in the framework of the CORK (Compact Rational Krylov) method by Van Beeumen *et al.* (2015) and the two-sided CORK method by Lietaert *et al.* (2018), which makes the linearization suitable for large scale problems and rational functions of high degree. There is no need to require C_i, D_i to be of low rank but if they are, it can be exploited.

We note that we are not the first ones to propose the use of the AAA algorithm and its appealing properties in the Nonlinear Eigenvalue Problem. In Elsworth & Güttel (2018), the authors propose to apply the AAA algorithm to $v^*A(\lambda)u$ for random vectors u, v, and then turn the barycentric formulation into a Newton representation, which is passed into the NLEIGS algorithm. However, we found that this approach leads to approximations of lower accuracy compared to the set valued approach from §2.2.

The rest of the paper is structured as follows. Section 2 presents the original AAA approximation of a nonlinear function g and its generalization to a set of nonlinear functions $g_1, \ldots, g_s$. In Section 3, we reformulate the linearization by Su & Bai (2011) as a CORK linearization and establish a relation between the eigenvalues and eigenvectors of the linearization and the rational matrix polynomial, including the case of low rank $C_i - \lambda D_i$, $i = 1, \ldots, s$. Section 4 illustrates the use of AAA for solving nonlinear eigenvalue problems and compares AAA to NLEIGS. Section 5 is reserved for the conclusions.

2. Scalar rational approximations by AAA

As explained in the introduction, we intend to use rational approximations for the NEP. As a first step, this requires approximating the scalar functions $g_i(\lambda)$ in (1.3), either separately or together, by rational functions. Our approach is based on the recently introduced algorithm by Nakatsukasa *et al.* (2018) that we review next.

2.1 *The AAA algorithm*

Let $g\colon \Sigma \to \mathbb{C}$ denote a generic nonlinear function that we would like to approximate on $\Sigma \subset \mathbb{C}$ by a rational function $r(\lambda)$. The adaptive Antoulas–Anderson (AAA) algorithm from Nakatsukasa *et al.* (2018), constructs $r(\lambda)$ in barycentric form:

$$r(\lambda) = \underbrace{\sum_{j=1}^{m} \frac{g(z_j)\,\omega_j}{\lambda - z_j}}_{=:n_m(\lambda)} \Bigg/ \underbrace{\sum_{j=1}^{m} \frac{\omega_j}{\lambda - z_j}}_{=:d_m(\lambda)}. \tag{2.1}$$

Here, $z_1, \ldots, z_m$ are a set of distinct *support points* and $\omega_1, \ldots, \omega_m$ are the *weights*. Note that, as long as $\omega_j \neq 0$, $\lim_{\lambda \to z_j} r(\lambda) = g(z_j)$. In other words, the rational function (2.1) interpolates the function $g(\lambda)$ at $z_1, \ldots, z_m$.

The AAA algorithm computes the support points and the weights iteratively by minimizing the linearized residual of the rational approximation on a sample set Z of M points. The set Z can be seen as a sufficiently fine discretization of the region Σ which means M is typically quite large. At the mth step of the algorithm, the next support point z_m is chosen so that the residual $g(\lambda) - n_{m-1}(\lambda)/d_{m-1}(\lambda)$

attains its maximum absolute value for $\lambda = z_m$. Then, denoting

$$\begin{aligned} &Z^{(m)} = \{Z_1^{(m)}, \dots, Z_{M-m}^{(m)}\} := Z/\{z_1, \dots, z_m\} \quad \text{and} \\ &G^{(m)} = \{G_1^{(m)}, \dots, G_{M-m}^{(m)}\} \text{ with } G_i^{(m)} = g(Z_i^{(m)}), \end{aligned}$$

it computes the vector of weights $\omega = \begin{bmatrix}\omega_1 & \cdots & \omega_m\end{bmatrix}^T$ with $\|\omega\|_2 = 1$ that minimizes the 2-norm of the linearized residual

$$\begin{bmatrix} g(Z_1^{(m)})d_m(Z_1^{(m)}) - n_m(Z_1^{(m)}) \\ \vdots \\ g(Z_{M-m}^{(m)})d_m(Z_{M-m}^{(m)}) - n_m(Z_{M-m}^{(m)}) \end{bmatrix} = \begin{bmatrix} \frac{G_1^{(m)}-g(z_1)}{Z_1^{(m)}-z_1} & \cdots & \frac{G_1^{(m)}-g(z_m)}{Z_1^{(m)}-z_m} \\ \vdots & \ddots & \vdots \\ \frac{G_{M-m}^{(m)}-g(z_1)}{Z_{M-m}^{(m)}-z_1} & \cdots & \frac{G_{M-m}^{(m)}-g(z_m)}{Z_{M-m}^{(m)}-z_m} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_m \end{bmatrix}. \tag{2.2}$$

This can be done by using the SVD on the Loewner matrix

$$\begin{bmatrix} \frac{G_1^{(m)}-g(z_1)}{Z_1^{(m)}-z_1} & \cdots & \frac{G_1^{(m)}-g(z_m)}{Z_1^{(m)}-z_m} \\ \vdots & \ddots & \vdots \\ \frac{G_{M-m}^{(m)}-g(z_1)}{Z_{M-m}^{(m)}-z_1} & \cdots & \frac{G_{M-m}^{(m)}-g(z_m)}{Z_{M-m}^{(m)}-z_m} \end{bmatrix}.$$

The procedure terminates when the norm of the residual (2.2) is less than a user defined tolerance, for example, 10^{-13}. For further details of the AAA algorithm, including the removal of numerical Froissart doublets, we refer to Nakatsukasa *et al.* (2018).

A key feature of AAA is its flexibility in selecting the domain of approximation (through the set Z), unlike other methods that are domain-dependent. Furthermore, the user only needs to supply this domain and a tolerance. Then, the poles and zeros of the rational interpolant (2.1) are found automatically by the algorithm. In many cases, AAA succeeds in computing a rational interpolant that is often not far from the optimal one. However, the algorithm can fail on difficult functions; see Filip *et al.* (2018) for examples. In the numerical experiments, however, we did not see such pathological behaviour and AAA performed adequately.

In Section 3, it will be convenient to write the rational functions from AAA that are in barycentric form into an equivalent state-space form. In the following proposition, we will present such a state-space representation.

PROPOSITION 2.1 The rational function (2.1) can be written as

$$r(\lambda) = \begin{bmatrix} g(z_1)\omega_1 & \cdots & g(z_m)\omega_m \end{bmatrix} \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_{m-1} & \omega_m \\ \lambda - z_1 & z_2 - \lambda & & & \\ & \lambda - z_2 & \ddots & & \\ & & \ddots & z_{m-1} - \lambda & \\ & & & \lambda - z_{m-1} & z_m - \lambda \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.3}$$

where the entries that are not depicted are equal to zero.

A slightly different state-space form for (2.1) can be found as Theorem 1.6 in Ionită (2013), where all basis functions are related to first basis function.

Proof. Let $d(\lambda) = \sum_{j=1}^{m} \omega_j (\lambda - z_j)^{-1}$ and $n(\lambda) = \sum_{j=1}^{m} g(z_j)\omega_j(\lambda - z_j)^{-1}$ denote, respectively, the denominator and numerator of $r(\lambda)$ in (3.4). Then, it is easily verified that the vector

$$\frac{1}{d(\lambda)} \begin{bmatrix} (\lambda - z_1)^{-1} \\ \vdots \\ (\lambda - z_m)^{-1} \end{bmatrix}$$

is the first column of

$$\begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_{m-1} & \omega_m \\ \lambda - z_1 & z_2 - \lambda & & & \\ & \lambda - z_2 & \ddots & & \\ & & \ddots & z_{m-1} - \lambda & \\ & & & \lambda - z_{m-1} & z_m - \lambda \end{bmatrix}^{-1}.$$

Thus, we obtain the desired result as

$$\frac{1}{d(\lambda)} \begin{bmatrix} g(z_1)\omega_1 & \cdots & g(z_m)\omega_m \end{bmatrix} \begin{bmatrix} (\lambda - z_1)^{-1} \\ \vdots \\ (\lambda - z_m)^{-1} \end{bmatrix} = \frac{n(\lambda)}{d(\lambda)} = r(\lambda).$$

□

2.2 *A set-valued AAA algorithm*

There are applications where a large number of nonlinear functions $g_1, \ldots, g_s$ need to be approximated on the same region Σ of the complex plane. One can of course use the AAA algorithm on each function separately but, as we will see in the numerical examples in Section 4, it is sometimes beneficial to find support points and poles that work for all the functions at the same time. The result is then a linearization with a smaller total degree compared to the linearization obtained from the separate applications of AAA. In this section, we show how to extend the AAA approach to accomplish this; see also Hochman (2017).

Let $g_1, \ldots, g_s$ be some nonlinear functions. In contrast to Hochman (2017), we assume that $g_1, \ldots, g_s$ are of the same scale over the sample set Z, that is, we assume

$$\max_{z \in Z} |g_i(z)| \approx \max_{z \in Z} |g_j(z)| \approx 1 \quad \text{for } i \neq j. \tag{2.4}$$

Observe that the condition (2.4) can be accomplished by simply scaling the functions $g_1, \ldots, g_s$ as $g_i(\lambda)/\max_{z \in Z} |g_i(z)|$. Our aim is to construct rational approximations to $g_1, \ldots, g_s$ of the form

$$g_i(\lambda) \approx \underbrace{\sum_{j=1}^{m} \frac{g_i(z_j)\,\omega_j}{\lambda - z_j}}_{=:n_{i,m}(\lambda)} \Big/ \underbrace{\sum_{j=1}^{m} \frac{\omega_j}{\lambda - z_j}}_{=:d_m(\lambda)}. \tag{2.5}$$

Note that all rational approximants share the same support points z_j and weights ω_j. In the spirit of the AAA algorithm, these support points and weights are computed iteratively. At the mth step, the next support point z_m is chosen where

$$\max_i \left| g_i(\lambda) - \frac{n_{i,m-1}(\lambda)}{d_{m-1}(\lambda)} \right|,$$

attains its maximum on Σ. Then, denoting

$$\begin{aligned} Z^{(m)} &= \{Z^{(m)}_1, \dots, Z^{(m)}_{M-m}\} := Z/\{z_1, \dots, z_m\} \quad \text{and} \\ G^{(m)}_i &= \{G^{(m)}_{i,1}, \dots, G^{(m)}_{i,M-m}\} \text{ with } G^{(m)}_{i,j} = g_i(Z^{(m)}_j), \end{aligned}$$

the residual vector (2.2) can be written to incorporate the different functions:

$$\begin{bmatrix} \frac{G^{(m)}_{1,1}-g_1(z_1)}{Z^{(m)}_1-z_1} & \cdots & \frac{G^{(m)}_{1,1}-g_1(z_m)}{Z^{(m)}_1-z_m} \\ \vdots & \ddots & \vdots \\ \frac{G^{(m)}_{1,M-m}-g_1(z_1)}{Z^{(m)}_{M-m}-z_1} & \cdots & \frac{G^{(m)}_{1,M-m}-g_1(z_m)}{Z^{(m)}_{M-m}-z_m} \\ \frac{G^{(m)}_{2,1}-g_2(z_1)}{Z^{(m)}_1-z_1} & \cdots & \frac{G^{(m)}_{2,1}-g_2(z_m)}{Z^{(m)}_1-z_m} \\ \vdots & \ddots & \vdots \\ \frac{G^{(m)}_{2,M-m}-g_2(z_1)}{Z^{(m)}_{M-m}-z_1} & \cdots & \frac{G^{(m)}_{2,M-m}-g_2(z_m)}{Z^{(m)}_{M-m}-z_m} \\ \vdots & \vdots & \vdots \\ \frac{G^{(m)}_{s,1}-g_s(z_1)}{Z^{(m)}_1-z_1} & \cdots & \frac{G^{(m)}_{s,1}-g_s(z_m)}{Z^{(m)}_1-z_m} \\ \vdots & \ddots & \vdots \\ \frac{G^{(m)}_{s,M-m}-g_s(z_1)}{Z^{(m)}_{M-m}-z_1} & \cdots & \frac{G^{(m)}_{s,M-m}-g_s(z_m)}{Z^{(m)}_{M-m}-z_m} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_m \end{bmatrix}, \tag{2.6}$$

The vector of weights $\omega = \begin{bmatrix} \omega_1 & \cdots & \omega_m \end{bmatrix}^T$ is computed as the vector minimizing the norm of (2.6) under the constraint $\|\omega\|_2 = 1$. We note that minimizing this norm is equivalent to minimizing the sum of squares of the norms of the residual vectors of the different functions.

The computational cost of the AAA algorithm and of our modified set-valued version might become an issue when a large number of nonlinear functions are approximated. To partly alleviate this extra cost, we can slightly reformulate how the AAA algorithm computes the minimizing vector ω in (2.2) or (2.6). This is outlined in the next section.

2.3 *Efficient solution of the least squares problem in AAA*

The original AAA algorithm requires the repeated singular value decomposition of the tall but skinny Loewner matrix in (2.2) to find its right singular vector corresponding to the smallest singular value. This can become costly for large matrices, which is certainly the case if we have do this for a large number of functions as in (2.6). Fortunately, we can exploit that the Loewner matrices in (2.2) or in (2.6) differ only in a few rows and columns throughout each iteration of the AAA algorithm. In particular, by computing economy-size QR decompositions of said matrices, we only need to obtain the right singular vectors of much smaller matrices. In turn, these QR decompositions can be computed using updating strategies.

Let L_m be the Loewner matrix in (2.2) or (2.6), of size $n \times m$. We recall that $n \gg m$. The matrix L_m can be stored as

$$L_m = QH,$$

where Q is an $n \times m$ matrix with orthonormal columns, and H is an $m \times m$ matrix. Note that the right singular vectors of the matrix L_m can be computed as the right singular vectors of the small matrix H. The column of the matrix Q can be iteratively found by adding one column of the Loewner matrix L_m in each step and applying Gram–Schmidt orthogonalization. Note, however, that in each step of AAA, the number of rows of Q is reduced by one for (2.2) and by s for (2.6) because the support points z_i are removed from the set Z that defines the residual. With the removal of these rows, Q is no longer orthogonal. However, we can reorthogonolize Q cheaply as follows. Let $Q_\mathrm{r} \in \mathbb{C}^{r \times m}$ be the matrix whose rows have been removed from Q in step m and let $\widetilde{Q}$ be the matrix obtained from Q after the removal of these rows. Note that the subscript in Q_r does not refer to the rth iteration of the AAA method, but is an abbreviation of the word removed. We then have, since Q is orthogonal,

$$\widetilde{Q}^* \widetilde{Q} = I_m - Q_\mathrm{r}^* Q_\mathrm{r}.$$

By taking the Cholesky decomposition,

$$I_m - Q_\mathrm{r}^* Q_\mathrm{r} = S^* S,$$

we have that matrix the $\widetilde{Q}S^{-1}$ is orthogonal, and we replace H by SH. We can further avoid the (costly) explicit multiplication $\widetilde{Q}S^{-1}$ by storing S as

$$S_m = \begin{bmatrix} S_{m-1} & \\ & 1 \end{bmatrix} S^{-1},$$

which is only used in matrix-vector and matrix-matrix multiplications with vectors and matrices of size $O(m)$, that is, of small size.

This procedure, including the set-valued AAA algorithm, is implemented in our MATLAB version of the AAA algorithm, see Section 4 on numerical examples. The main cost of the algorithm is reduced to the Gram–Schmidt orthogonalization process of the long vectors of the Loewner matrix.

REMARK 2.1 The idea of speeding up the original AAA algorithm by carefully updating a matrix decomposition of the Loewner matrix has also been investigate in Hochman (2017). In contrast to our approach, which uses a QR-like decomposition, in Hochman (2017), the right singular vectors of the Loewner matrix L_m are computed as the right singular vectors of the Cholesky factor of $L_m^* L_m$.

3. Rational approximations for NEPs using AAA

In this section, we show how the scalar rational functions, computed by AAA, can be used efficiently to obtain a rational approximation of the NEP. In particular, we will present linearizations that build on the CORK (Van Beeumen *et al.*, 2015) and the TS-CORK (Lietaert *et al.*, 2018) frameworks and exploit possible low-rank terms. These frameworks support efficient operations on linearizations for (rational) Krylov methods.

3.1 *The CORK framework*

The starting point of the compact rational Krylov (CORK) method in Van Beeumen *et al.* (2015) is a matrix-valued function of the form

$$P(\lambda) = \sum_{i=0}^{k-1} (A_i - \lambda B_i) f_i(\lambda), \quad \text{with} \quad A_i, B_i \in \mathbb{C}^{n\times n}, \tag{3.1}$$

where $f_i \colon \mathbb{C} \to \mathbb{C}$ are polynomial or rational functions satisfying the linear relation

$$(M - \lambda N) f(\lambda) = 0 \qquad \text{with } \operatorname{rank}(M - \lambda N) = k-1 \text{ for all } \lambda \in \mathbb{C}, \tag{3.2}$$

$N, M \in \mathbb{C}^{(k-1)\times k}$ and $f(\lambda) = \begin{bmatrix} f_0(\lambda) & \cdots & f_{k-1}(\lambda) \end{bmatrix}^T \neq 0$. Without much loss of generality, we further assume that $f_0(\lambda) \equiv 1$ has degree zero. This assumption is indeed not restrictive in practice, as it covers most of the important cases in applications including monomials, Chebyshev polynomials, orthogonal polynomials, Newton polynomials, rational Newton functions. For a more general setting and explicit examples of $M - \lambda N$, we refer to Van Beeumen *et al.* (2015).

Given a matrix-valued function (3.1) satisfying (3.2), the matrix pencil

$$\mathscr{L}_P(\lambda) = \left[\begin{array}{c} A_0 - \lambda B_0 \quad \cdots \quad A_{k-1} - \lambda B_{k-1} \\ \hline (M - \lambda N) \otimes I_n \end{array} \right], \tag{3.3}$$

is called the CORK linearization of $P(\lambda)$. When $P(\lambda)$ is a matrix polynomial, the pencil (3.3) is a linearization in the usual sense (Gohberg *et al.*, 1982), that is, there are matrix polynomials $U(\lambda)$ and $V(\lambda)$ with nonzero constant determinants such that

$$U(\lambda)\mathscr{L}_P(\lambda)V(\lambda) = \operatorname{diag}(P(\lambda), I_s),$$

for some identity matrix I_s; see Dopico *et al.* (2018); Robol *et al.* (2016). We recall that the definition of linearization implies that $P(\lambda)$ and $\mathscr{L}_P(\lambda)$ have the same eigenvalues with the same geometric and algebraic multiplicities. When $P(\lambda)$ is a rational matrix, it is not clear whether the pencil (3.3) is a linearization of $P(\lambda)$ in the sense of Amparan *et al.* (2018). In any case, for our purposes, we only need to use that $P(\lambda)$ and $\mathscr{L}_P(\lambda)$ have the same eigenvalues and that the eigenvectors of $P(\lambda)$ can be easily recovered from those of $\mathscr{L}_P(\lambda)$ (see Van Beeumen *et al.*, 2015, Corollary 2.4).

The CORK linearization (3.3) is of size $kn \times kn$ which can become quite large. Fortunately, its Kronecker structure can be exploited by Krylov methods; see, e.g., (Van Beeumen *et al.*, 2015, Algorithm 3) on how to efficiently use the rational Krylov method in this context.

3.2 *Extending CORK with AAA*

Let us now consider the NEP with $A(\lambda)$ as defined in (1.2), that is,

$$A(\lambda) = P(\lambda) + \sum_{i=1}^{s} (C_i - \lambda D_i) g_i(\lambda).$$

Using AAA, or its set-valued generalization, we can approximate each function $g_i(\lambda)$ on the region $\Sigma \subset \mathbb{C}$ as

$$g_i(\lambda) \approx r_i(\lambda) = \sum_{j=1}^{\ell_i} \frac{g_i(z_j^{(i)})\,\omega_j^{(i)}}{\lambda - z_j^{(i)}} \Bigg/ \sum_{j=1}^{\ell_i} \frac{\omega_j^{(i)}}{\lambda - z_j^{(i)}}, \tag{3.4}$$

where ℓ_i is the number of support points $z_j^{(i)}$ and weights $\omega_j^{(i)}$ for each $i = 1, \dots, s$. If some of the g_i are approximated together by the set-valued AAA algorithm, the $z_j^{(i)}$ and $\omega_j^{(i)}$ are the same for their corresponding indices i. For now, we ignore this property. In any case, we can use the rational approximations $r_i(\lambda)$ to obtain an approximation of the NEP on the same region Σ:

$$A(\lambda) \approx R(\lambda) = P(\lambda) + \sum_{i=1}^{s} (C_i - \lambda D_i) r_i(\lambda). \tag{3.5}$$

We now show how to obtain a CORK linearization of $R(\lambda)$. If we assume that $P(\lambda)$ satisfies (3.1), then by making use of Prop. 2.1, we can also write the rational part in (3.5) in state-space form as

$$R(\lambda) = \sum_{i=0}^{k-1} (A_i - \lambda B_i) f_i(\lambda) + \sum_{i=1}^{s} (C_i - \lambda D_i)\, a_i^T (E_i - \lambda F_i)^{-1} b_i \tag{3.6}$$

for some vectors $a_i, b_i \in \mathbb{C}^{\ell_i}$ and the $\ell_i \times \ell_i$ matrices

$$E_i = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_{\ell_{i-1}} & \omega_{\ell_i} \\ -z_1 & z_2 & & & \\ & -z_2 & \ddots & & \\ & & \ddots & z_{\ell_{i-1}} & \\ & & & -z_{\ell_{i-1}} & z_{\ell_i} \end{bmatrix} \quad \text{and} \quad F_i = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & -1 & & & \\ & 1 & \ddots & & \\ & & \ddots & -1 & \\ & & & 1 & -1 \end{bmatrix}.$$

Next, introduce for $i = 1, \dots, s$ the vector-valued function

$$R_i \colon \mathbb{C} \to \mathbb{C}^{\ell_i}, \qquad R_i(\lambda) = (E_i - \lambda F_i)^{-1} b_i. \tag{3.7}$$

Assuming that $P(\lambda)$ satisfies (3.1) with $f_0(\lambda) = 1$ and observing that $(E_i - \lambda F_i) R_i(\lambda) = b_i$ for all $i = 1, \dots, s$, we obtain the linear relation

$$\left[\begin{array}{c:ccc|ccc} \multicolumn{4}{c|}{M - \lambda N} & 0 & \cdots & 0 \\ \hline -b_1 & 0 & \cdots & 0 & E_1 - \lambda F_1 & & \\ \vdots & \vdots & & \vdots & & \ddots & \\ -b_s & 0 & \cdots & 0 & & & E_s - \lambda F_s \end{array}\right] \left[\begin{array}{c} f_0(\lambda) \\ \hdashline f_1(\lambda) \\ \vdots \\ f_{k-1}(\lambda) \\ \hline R_1(\lambda) \\ \vdots \\ R_s(\lambda) \end{array}\right] = 0.$$

Collecting the basis functions into the single vector

$$f(\lambda) = \begin{bmatrix} f_0(\lambda) \\ \vdots \\ f_{k-1}(\lambda) \end{bmatrix}, \quad \Psi(\lambda) = \left[\begin{array}{c} f(\lambda) \\ \hline R_1(\lambda) \\ \vdots \\ R_s(\lambda) \end{array}\right], \tag{3.8}$$

we arrive at the following result.

PROPOSITION 3.1 Let $\Psi(\lambda)$ be the vector-valued function (3.8) with $f_i(\lambda)$ scalar functions satisfying (3.2) such that $f_0(\lambda) = 1$ and $R_i(\lambda)$ satisfying (3.7). Then,

$$(\widehat{M} - \lambda\widehat{N})\Psi(\lambda) = 0 \qquad \text{with} \quad \widehat{M} - \lambda\widehat{N} = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{M - \lambda N} & 0 \\ \hline -b & 0 & E - \lambda F \end{array}\right], \tag{3.9}$$

where we used

$$b = \begin{bmatrix} b_1^T & \cdots & b_s^T \end{bmatrix}^T \qquad \text{and} \qquad E - \lambda F = \operatorname{diag}(E_1 - \lambda F_1, \ldots, E_s - \lambda F_s). \tag{3.10}$$

Furthermore, the pencil $\widehat{M} - \lambda\widehat{N}$ has full row rank for any $\lambda \in \mathbb{C}$ so that $E_i - \lambda F_i$ is invertible for all $i = 1, \ldots, s$.

Proof. The identity (3.9) was shown above since $f_0(\lambda) = 1$. The second result is immediate since $M - \lambda N$ has full row rank by assumption and $E - \lambda F$ is only singular when one of the $E_i - \lambda F_i$ is singular. □

In order to obtain a CORK linearization of (3.6), we first write it using (3.7) as

$$\begin{aligned} R(\lambda) &= \sum_{i=0}^{k-1}(A_i - \lambda B_i)(f_i(\lambda) \cdot I_n) + \sum_{i=1}^{s}(C_i - \lambda D_i)\,(a_i^T R_i(\lambda) \cdot I_n) \\ &= \sum_{i=0}^{k-1}(A_i - \lambda B_i)\,(f_i(\lambda) \otimes I_n) + \sum_{i=1}^{s}[a_i^T \otimes (C_i - \lambda D_i)]\,(R_i(\lambda) \otimes I_n). \end{aligned}$$

Observe that this is a trivial rewriting of scalar multiplications in terms of Kronecker products. However, using the vector $\Psi(\lambda)$ as defined in (3.8), it allows us to express the rational expression in (3.6) as

$$R(\lambda) = \left[\begin{array}{ccc|ccc} A_0 - \lambda B_0 & \cdots & A_{k-1} - \lambda B_{k-1} & a_1^T \otimes (C_1 - \lambda D_1) & \cdots & a_s^T \otimes (C_s - \lambda D_s) \end{array}\right] \times (\Psi(\lambda) \otimes I_n).$$

Together with Prop. 3.1, this suggests the following CORK linearization of (3.6).

DEFINITION 3.2 (CORK linearization for AAA rational approximation) Let $R(\lambda)$ be the rational approximation (3.6) obtained by using the AAA algorithm or the set-valued AAA algorithm. We define the pencil $\mathscr{L}_R(\lambda)$ as follows

$$\mathscr{L}_R(\lambda) = \left[\begin{array}{c} \begin{array}{ccc|ccc} A_0 - \lambda B_0 & \cdots & A_{k-1} - \lambda B_{k-1} & a_1^T \otimes (C_1 - \lambda D_1) & \cdots & a_s^T \otimes (C_s - \lambda D_s) \end{array} \\ \hline (\widehat{M} - \lambda\widehat{N}) \otimes I_n \end{array}\right], \tag{3.11}$$

where the pencil $\widehat{M} - \lambda\widehat{N}$ has been defined in (3.9).

The size of $\mathscr{L}_R(\lambda)$ is $(k + \sum_{i=1}^{s} \ell_i)n$. Fortunately, one can again exploit the Kronecker structure and show that the CORK algorithm can be applied to (3.11), as long as the shifts in the shift-and-invert steps of the rational Krylov method are not poles of the rational interpolants (3.4). Furthermore, as a special case of Theorem 3.4 in Section 3.3 with full-rank matrices, any $\lambda \in \mathbb{C}$ that is not a pole of any of the rational interpolants (3.4) is an eigenvalue of $R(\lambda)$ if and only if it is an eigenvalue of (3.11), and their associated right eigenvectors are easily related.

REMARK 3.1 In case the set-valued AAA algorithm is applied to all g_i together, we have that $E_i - \lambda F_i$ is the same for all i, as well as all b_i. As a result, one can use the CORK linearization

$$\mathscr{L}_R(\lambda) = \left[\begin{array}{ccc} A_0 - \lambda B_0 \quad \cdots \quad A_{k-1} - \lambda B_{k-1} & & \sum_{i=1}^{s} a_i^T \otimes (C_i - \lambda D_i) \\ \hline (M - \lambda N) \otimes I_n & & 0 \\ -b_1 \otimes I_n \;\; 0 & & (E_1 - \lambda F_1) \otimes I_n \end{array}\right],$$

which is of size $(k + \ell_1)n$.

Sumarizing, the CORK algorithm can be applied to (3.11) for computing eigenvalues of $R(\lambda)$ that are not poles of the rational interpolants (3.4) and their associated right eigenvectors. In practice, we have noticed that this assumption is not very restrictive, since the AAA algorithm tends to place the poles outside the region of interest.

3.3 *Low-rank exploitation*

In several applications, the matrix coefficients of the nonlinear valued function $G(\lambda)$ in (1.3) are of low rank. In this section, we show how the exploitation of these low ranks leads to a linearization of size smaller than that of $\mathscr{L}_R(\lambda)$. This linearization generalizes the one used in Dopico & González-Pizarro (2018); Su & Bai (2011), which is valid when $P(\lambda)$ in (3.1) is expressed using monomials, i.e., $f_i(\lambda) = \lambda^i$, to the more general setting used by CORK.

Suppose that the coefficients of the rational part in (3.6) admit the following structure (possibly after a rank-revealing decomposition of $C_i - \lambda D_i$)

$$C_i - \lambda D_i = (\widetilde{C}_i - \lambda \widetilde{D}_i)\widetilde{Z}_i^* \quad \text{with} \quad \widetilde{C}_i, \widetilde{D}_i, \widetilde{Z}_i \in \mathbb{C}^{n \times k_i}, \quad \text{and} \quad \widetilde{Z}_i^* \widetilde{Z}_i = I_{k_i}. \tag{3.12}$$

Observe that this holds trivially for $\widetilde{Z}_i = I_{k_i}$ but in many problems k_i is potentially much smaller than n. Introducing the matrices $\widetilde{Z}_i$ in the definition of $R(\lambda)$ in (3.6), we obtain

$$\begin{aligned} R(\lambda) &= \sum_{i=0}^{k-1} (A_i - \lambda B_i) f_i(\lambda) + \sum_{i=1}^{s} (a_i^T R_i(\lambda)) \cdot (\widetilde{C}_i - \lambda \widetilde{D}_i)\widetilde{Z}_i^* \\ &= \sum_{i=0}^{k-1} (A_i - \lambda B_i)\,(f_i(\lambda) I_n) + \sum_{i=1}^{s} [a_i^T \otimes (\widetilde{C}_i - \lambda \widetilde{D}_i)]\,[R_i(\lambda) \otimes I_{k_i}]\,\widetilde{Z}_i^*, \end{aligned}$$

where we recall that $R_i(\lambda) = (E_i - \lambda F_i)^{-1} b_i$. We can therefore write

$$R(\lambda) = \begin{bmatrix} \boldsymbol{A} - \lambda \boldsymbol{B} & \boldsymbol{C} - \lambda \boldsymbol{D} \end{bmatrix} \cdot \Psi(\lambda)$$

using the matrices

$$\begin{gathered} \boldsymbol{A} = \begin{bmatrix} A_0 & \cdots & A_{k-1} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} B_0 & \cdots & B_{k-1} \end{bmatrix}, \\ \boldsymbol{C} = \begin{bmatrix} a_1^T \otimes \widetilde{C}_1 & \cdots & a_s^T \otimes \widetilde{C}_s \end{bmatrix}, \quad \boldsymbol{D} = \begin{bmatrix} a_1^T \otimes \widetilde{D}_1 & \cdots & a_s^T \otimes \widetilde{D}_s \end{bmatrix}, \\ f(\lambda) = \begin{bmatrix} f_0(\lambda) \\ \vdots \\ f_{k-1}(\lambda) \end{bmatrix}, \quad \Psi(\lambda) = \begin{bmatrix} f(\lambda) \otimes I_n \\ (R_1(\lambda) \otimes I_{k_1})\widetilde{Z}_1^* \\ \vdots \\ (R_s(\lambda) \otimes I_{k_s})\widetilde{Z}_s^* \end{bmatrix}. \end{gathered}$$

Denoting by $\boldsymbol{O}$ a matrix of all zeros (of suitable size), and using

$$\boldsymbol{M} = M \otimes I_n, \quad \boldsymbol{N} = N \otimes I_n,$$

where the pencil $M - \lambda N$ is the one in (3.2), we obtain from $(M - \lambda N) f(\lambda) = 0$ the identity

$$\begin{bmatrix} \boldsymbol{M} - \lambda \boldsymbol{N} & \boldsymbol{O} \end{bmatrix} \cdot \Psi(\lambda) = \boldsymbol{O}.$$

As before $(E_i - \lambda F_i) R_i(\lambda) = b_i$, whence

$$[(E_i - \lambda F_i) \otimes I_{k_i}] \, [R_i(\lambda) \otimes I_{k_i}] \widetilde{Z}_i^* = [b_i \otimes I_{k_i}] \widetilde{Z}_i^*.$$

Therefore by assuming again that $f_0(\lambda) \equiv 1$ and introducing

$$\boldsymbol{E} = \operatorname{diag}(E_1 \otimes I_{k_1}, \ldots, E_s \otimes I_{k_s}), \quad \boldsymbol{F} = \operatorname{diag}(F_1 \otimes I_{k_1}, \ldots, F_s \otimes I_{k_s}),$$
$$e_1^T = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^k, \quad \boldsymbol{Z}^* = \begin{bmatrix} -(b_1 \otimes I_{k_1}) \widetilde{Z}_1^* \\ \vdots \\ -(b_s \otimes I_{k_s}) \widetilde{Z}_s^* \end{bmatrix} (e_1^T \otimes I_n),$$

we obtain the identity

$$\begin{bmatrix} \boldsymbol{Z}^* & \boldsymbol{E} - \lambda \boldsymbol{F} \end{bmatrix} \cdot \Psi(\lambda) = \boldsymbol{O}.$$

Putting all the identities from above together, we obtain the following square matrix of size $\widetilde{d} = nk + \sum_{i=1}^s \ell_i k_i$:

$$\widetilde{\mathscr{L}_R}(\lambda) \, \Psi(\lambda) = \begin{bmatrix} R(\lambda) \\ \boldsymbol{O} \end{bmatrix} \qquad \text{with } \widetilde{\mathscr{L}_R}(\lambda) = \begin{bmatrix} \boldsymbol{A} - \lambda \boldsymbol{B} & \boldsymbol{C} - \lambda \boldsymbol{D} \\ \boldsymbol{M} - \lambda \boldsymbol{N} & \boldsymbol{O} \\ \boldsymbol{Z}^* & \boldsymbol{E} - \lambda \boldsymbol{F} \end{bmatrix}. \tag{3.13}$$

In Theorem 3.4 below, we show that, as long as λ is not a pole of any of the rational functions $r_i(\lambda)$ in (3.5), $\widetilde{\mathscr{L}_R}(\lambda)$ is indeed a linearization for $R(\lambda)$ in the sense that we can use it to compute the eigenpairs of $R(\lambda)$. Observe that $\widetilde{d}$ is never larger than $d = n(k + \sum_{i=1}^s \ell_i)$, the size of $\mathscr{L}_R(\lambda)$. Hence, $\widetilde{\mathscr{L}_R}(\lambda)$ is a trimmed linearization that effectively exploits the low-rank terms in the rational part of $R(\lambda)$. It is also possible to exploit low-rank terms in $P(\lambda)$ as in Van Beeumen *et al.* (2015). However, as this would complicate notation and the gain in size is typically less significant, we do not pursue this here.

Together with Theorem 3.4, we also have in Theorem 3.3 an explicit block-UL factorization of $\widetilde{\mathscr{L}_R}(\lambda)$. The proof of both these results is fairly standard and is therefore demoted to the appendix—in particular, we refer to similar results in Su & Bai (2011) for rational terms $R_i(\lambda)$ with explicit state-space representation, in Van Beeumen *et al.* (2015) for $P(\lambda)$ in CORK form, and in Dopico & González-Pizarro (2018) for $P(\lambda)$ in companion form combined with explicit state space for $R_i(\lambda)$. However, a compact representation that is a combination of general $P(\lambda)$ in CORK form and rational terms stemming from AAA is new.

The theorems are stated for a certain permuted version of the columns of $\widetilde{\mathscr{L}_R}(\lambda)$. By assumption, the $(k-1) \times k$ pencil $M - \lambda N$ has rank $k - 1$. Hence, there exists a permutation $P_0 \in \mathbb{R}^{k \times k}$, possibly depending on λ, such that

$$(M - \lambda N) P_0 =: \begin{bmatrix} m_0 - \lambda n_0 & M_1 - \lambda N_1 \end{bmatrix} \qquad \text{with } M_1 - \lambda N_1 \text{ nonsingular.}$$

Denoting $\boldsymbol{P_0} = P_0 \otimes I_n$, we can also apply this permutation block-wise to the first nk columns of $\widetilde{\mathscr{L}}_R(\lambda)$. We then obtain

$$\left[\begin{array}{c|c} \boldsymbol{A} - \lambda \boldsymbol{B} & \boldsymbol{C} - \lambda \boldsymbol{D} \\ \boldsymbol{M} - \lambda \boldsymbol{N} & \boldsymbol{O} \\ \boldsymbol{Z}^* & \boldsymbol{E} - \lambda \boldsymbol{F} \end{array}\right] \left[\begin{array}{cc} \boldsymbol{P_0} & \\ \hline & I \end{array}\right] = \left[\begin{array}{cc|c} \boldsymbol{A}_0 - \lambda \boldsymbol{B}_0 & \boldsymbol{A}_1 - \lambda \boldsymbol{B}_1 & \boldsymbol{C} - \lambda \boldsymbol{D} \\ \boldsymbol{M}_0 - \lambda \boldsymbol{N}_0 & \boldsymbol{M}_1 - \lambda \boldsymbol{N}_1 & \boldsymbol{O} \\ \boldsymbol{Z}_0^* & \boldsymbol{Z}_1^* & \boldsymbol{E} - \lambda \boldsymbol{F} \end{array}\right], \tag{3.14}$$

where I denotes an identity matrix of suitable size, and with

$$\boldsymbol{M}_0 = m_0 \otimes I_n, \quad \boldsymbol{n}_0 = n_0 \otimes I_n, \quad \boldsymbol{M}_1 = M_1 \otimes I_n, \quad \boldsymbol{N}_1 = N_1 \otimes I_n.$$

The other block matrices are partitioned accordingly. This means $\boldsymbol{A}_0 = A_j$ and $\boldsymbol{B}_0 = B_j$ for some j that corresponds to the column that $\boldsymbol{P_0}$ has permuted to the first position.

As mentioned above, one of the results is a block-UL factorization. Amongst others, it is key for performing efficiently the shift-and-invert steps of the rational Krylov method when computing the eigenvalues of $\widetilde{\mathscr{L}}_R(\lambda)$.

THEOREM 3.3 Let $\widetilde{\mathscr{L}}_R(\lambda)$ be the pencil in (3.13) for the rational matrix $R(\lambda)$ in (3.6) with the low-rank structure (3.12). If $\mu \in \mathbb{C}$ is such that all $E_1 - \mu F_1, \dots, E_s - \mu F_s$ are nonsingular, then using the block matrices as defined in (3.14), the following block-UL decomposition holds:

$$\widetilde{\mathscr{L}}_R(\mu)\,\mathscr{P} = \mathscr{U}(\mu)\,\mathscr{L}(\mu),$$

where (empty blocks are zero and $\rho = \sum_{i=1}^{s} \ell_i k_i$)

$$\mathscr{P} = \begin{bmatrix} \boldsymbol{P_0} & \\ & I \end{bmatrix}$$

$$\mathscr{U}(\mu) = \begin{bmatrix} I_n & [\boldsymbol{A}_1 - \mu \boldsymbol{B}_1 - \boldsymbol{Z}_1^*(\boldsymbol{C} - \mu \boldsymbol{D})]\,[\boldsymbol{M}_1 - \mu \boldsymbol{N}_1]^{-1} & (\boldsymbol{C} - \mu \boldsymbol{D})(\boldsymbol{E} - \mu \boldsymbol{F})^{-1} \\ & I_{(k-1)n} & \\ & & I_\rho \end{bmatrix},$$

$$\mathscr{L}(\mu) = \begin{bmatrix} \alpha(\mu)^{-1} R(\mu) & & \\ \boldsymbol{M}_0 - \mu \boldsymbol{N}_0 & \boldsymbol{M}_1 - \mu \boldsymbol{N}_1 & \\ \boldsymbol{Z}_0^* & \boldsymbol{Z}_1^* & \boldsymbol{E} - \mu \boldsymbol{F} \end{bmatrix}, \quad \alpha(\mu) = e_1^T P_0^T f(\mu) \neq 0.$$

In addition,

$$\alpha(\mu)^n \det \widetilde{\mathscr{L}}_R(\mu) = \det R(\mu)\,(\det(M_1 - \mu N_1))^{k-1} \prod_{i=1}^{s} (\det(E_i - \mu F_i))^{\ell_i}. \tag{3.15}$$

Proof. See appendix A. □

Next, we have the main result for the linearization: the relation of the eigenvalues (and their algebraic and geometric multiplicities) and eigenvectors of the rational matrix $R(\lambda)$ with those of the matrix trimmed pencil $\widetilde{\mathscr{L}}_R(\lambda)$.

THEOREM 3.4 Let $\widetilde{\mathscr{L}}_R(\lambda)$ be the pencil in (3.13) for the rational matrix $R(\lambda)$ in (3.6) with $f_0(\lambda) \equiv 1$ and the low-rank structure (3.12). Let $\lambda_0 \in \mathbb{C}$ be such that all $E_1 - \lambda_0 F_1, \dots, E_s - \lambda_0 F_s$ are nonsingular. Denote $\rho = \sum_{i=1}^{s} \ell_i k_i$.

(a) If $x \in \mathbb{C}^n$ is an eigenvector of $R(\lambda)$ with eigenvalue λ_0, then $\Psi(\lambda_0)x \in \mathbb{C}^{kn+\rho}$ is an eigenvector of $\widetilde{\mathscr{L}_R}(\lambda)$ with eigenvalue λ_0.

(b) If $z \in \mathbb{C}^{kn+\rho}$ is an eigenvector of $\widetilde{\mathscr{L}_R}(\lambda)$ with eigenvalue λ_0, then $z = \Psi(\lambda_0)x$ for some eigenvector $x \in \mathbb{C}^n$ of $R(\lambda)$ with eigenvalue λ_0.

(c) The algebraic and geometric multiplicities of λ_0 as an eigenvalue of $\widetilde{\mathscr{L}_R}(\lambda)$ and as an eigenvalue of $R(\lambda)$ are the same.

Proof. See appendix A. □

REMARK 3.2 We observe that Theorem 3.4 might miss some of the eigenvalues of $R(\lambda)$, since there could be eigenvalues of $R(\lambda)$ for which at least one of the matrices $E_i - \lambda F_i$ is singular. However, in practice, this does not pose a problem given that AAA tends to place the poles of the rational interpolants, that is, the eigenvalues of the pencils $E_i - \lambda F_i$, outside the interpolation region.

REMARK 3.3 It should be noted that the rational approximation, and therefore the linearization of $A(\lambda)$, computes eigenvalues of a rational matrix $R(\lambda)$ that is a small perturbation of $A(\lambda)$. The level of the error of approximation does not guarantee that the same level of error is obtained for the eigenvalues of $A(\lambda)$.

4. Numerical examples

This section illustrates the theory with a number of applications. All examples are implemented in MATLAB[1]. The AAA algorithm is used to approximate different nonlinear matrix-valued functions and the accuracy of the resulting rational approximations is compared to the accuracy obtained with potential theory (NLEIGS). The approximations are compared in terms of accuracy and the number of poles (which is equal to the degree plus one) to achieve that accuracy. We used the rational Krylov method (more specifically, its CORK implementation by Van Beeumen *et al.* (2015)) to obtain eigenvalue and eigenvector estimates.

We compare the following three methods.

NLEIGS: This is the static variant from Güttel *et al.* (2014). The rational polynomial is expressed in a basis of rational Newton polynomials:

$$R(\lambda) = P(\lambda) + \sum_{i=0}^{s} R_i N_i(\lambda),$$

where $N_0, \dots, N_s$ are rational Newton polynomials of the form

$$N_i(\lambda) = \frac{1}{\beta_0} \prod_{k=1}^{i} \frac{1}{\beta_k} \frac{\lambda - \sigma_{k-1}}{1 - \lambda/\xi_k}$$

with $\sigma_0, \dots, \sigma_{s-1}$ Leja Bagby interpolation points and $1/\xi_1, \dots, 1/xi_s$ poles. The interpolation points and poles are determined recursively using the Leja Bagby algorithm, i.e., such that the

[1] Available at https://people.cs.kuleuven.be/karl.meerbergen/files/aaa/autoCORK.zip

poles are selected from the singularity set $\varXi$ of the nonlinear function and the nodes are selected from the region $\varSigma$, such that

$$\inf_{\lambda\in\varXi}|n_{i-1}(\lambda)| = |n_{i-1}(1/\xi_i)| \quad \text{and} \quad \max_{\lambda\in\varSigma}|n_i(\lambda)| = |n_i(\sigma_i)|$$

for $i = 1,\dots,s$.

AAA-EIGS: This is the rational Krylov method applied to linearization (3.3). The rational functions are determined by applying AAA to the s nonlinear functions from (1.3) separately.

SV-AAA-EIGS: This is the rational Krylov method applied to linearization (3.3). The rational functions are determined using the set-valued AAA approach explained in §2.2.

In the next section, we review the rational Krylov method that is used in the numerical experiments for finding eigenvalues of the linearizations. We do not use implicit restarting, i.e., the number of iterations corresponds to the dimension of the Krylov space.

4.1 *The rational Krylov method*

The rational Krylov method, sketched in Algorithm 1, is a generalization of the shift-and-invert Arnoldi method for solving large-scale generalized eigenvalue problems.

Algorithm 1 Rational Krylov method

```
Choose vector $v_1$, where $\|v\|_2 = 1$.
for $j = 1,2,\dots$ do
    Choose shift $\sigma_j$.
    Choose continuation vector $t_j$.
    Compute $\widehat{v}_j = (A-\sigma_j B)^{-1} B w_j$, where $w_j = V_j t_j$.
    Orthogonalize $\widetilde{v}_j = \widehat{v}_j - V_j h_j$, where $h_j = V_j^* \widehat{v}_j$.
    Get new vector $v_{j+1} = \widetilde{v}_j / h_{j+1,j}$, where $h_{j+1,j} = \|\widetilde{v}_j\|_2$.
    Set $V_{j+1} = \begin{bmatrix} V_j & v_{j+1} \end{bmatrix}$.
end for
```

At step j, Algorithm 1 computes a matrix V_{j+1} whose columns form an orthonormal basis for the rational Krylov subspace

$$\mathscr{K}_j(A,B,v_1) = \operatorname{span}\{v_1, \widehat{v}_1, \dots, \widehat{v}_j\},$$

where $\widehat{v}_i = (A-\sigma_i B)^{-1} B w_i$, for $i = 1,\dots,j$. Furthermore, the matrix V_{j+1} satisfies the rational Krylov recurrence relation

$$A V_{j+1} \underline{H}_j = B V_{j+1} \underline{K}_j,$$

where $\underline{H}_j \in \mathbb{C}^{(j+1)\times j}$ is an upper Hessenberg matrix whose nonzero entries are the Gram–Schmidt coefficients computed by Algorithm 1, and

$$\underline{K}_j = \underline{H}_j \operatorname{diag}(\sigma_1,\dots,\sigma_j) + \underline{T}_j \in \mathbb{C}^{(j+1)\times j},$$

where $\underline{T}_j$ is an upper triangular matrix whose ith column is the continuation vector t_i appended with some extra zeros. Note that we use $t_j = e_j$, where e_j is the j-th vector of I, i.e. we always choose w_j as the iteration vector of the previous step.

Approximations for the eigenvalues and right eigenvectors of the pencil $A - \lambda B$ are obtained by solving the small generalized eigenvalue problem

$$K_j s_i = \lambda_i H_j s_i,$$

where H_j and K_j are, respectively, the $j \times j$ upper part of $\underline{H}_j$ and $\underline{K}_j$. The pair $(\lambda_i, x_i = V_{j+1}\underline{H}_j s_i)$ is referred to as a Ritz pair of the pencil $A - \lambda B$.

REMARK 4.1 An important consideration that the reader should keep in mind is how the memory and orthogonalization costs of the CORK method depend on the number of poles needed by any of the three algorithms (NLEIGS, AAA-EIGS and SV-AAA-EIGS) considered in this section. Assuming the algorithm needs to use d poles in total, at step j of the CORK method the storage cost of the iteration vectors is of order $d + j$ vectors of size n, and the orthogonalization cost is of order $(d + j)j$ scalar products of vectors of size n; see Van Beeumen *et al.* (2015). Hence, the smaller the number of poles, the lower are the orthogonalization and memory costs. Nevertheless, we found, as for NLEIGS, the main cost lies in the LU factorization and the linear system solves with $R(\lambda)$. We compared timings of NLEIGS and SV-AAA-EIGS obtained by averaging three run on an Intel 2.9GHz processor with 16GB memory using Matlab R2018a. Both methods have comparable timings as can be seen in Table 1. Further analysis confirms the statements made in this remark.

4.2 *Gun problem*

The radio-frequency gun cavity problem from the NLEVP collection Betcke *et al.* (2013) is described by the following matrix-valued function in λ

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

where $K, M, W_1, W_2 \in \mathbb{R}^{9956 \times 9956}$ are symmetric positive semi-definite matrices, $\sigma_1 = 0$, and $\sigma_2 = 108.8774$.

We accurately approximate the nonlinear part in $A(\lambda)$ in a semi-circle Σ in the complex plane; see Figure 1a. The function is approximated by a rational function in two different ways. First, for NLEIGS, a rational polynomial with 31 poles is used, with poles picked on the branch cut of $\sqrt{\lambda - \sigma_2^2}$ on the open interval $(-\infty, \sigma_2]$. Second, for AAA-EIGS, the AAA test set consists of 500 random points in the half disk combined with 500 equally distributed points on the boundary. The resulting poles for both functions $f_j = \sqrt{(\lambda - \sigma_j^2)}$, $j = 1, 2$, are plotted in Figure 1a. Note that not all poles are shown, only the ones with a positive real part.

Table 1: Computation time [s] for all iterations for different problems.

	NLEIGS	SV-AAA-EIGS
Gun problem	10.9652	10.7188
Car cavity	51.5775	49.6916
Sandwich beam	0.562	0.3972

Figure 1b shows the approximation error as a function of the number of poles for NLEIGS, AAA-EIGS, and SV-AAA-EIGS. The approximation error is expressed as the relative summed error on the nonlinear functions:

$$E_f = \sqrt{\sum_i \left(\sum_j \frac{f_j(s_i)}{\sum_k f_j^2(s_k)} \right)^2}, \tag{4.1}$$

for $j = 1,2$, and as the error on the matrix functions:

$$E_m = \sqrt{\sum_{i=1}^{m} \frac{\|A(s_i) - R(s_i)\|_1^2}{\|A(s_i)\|_1^2}},$$

where the points $s_i,\ i = 1,\dots,1000$ are points of a test set consisting of random points on the boundary and the interior of Σ. AAA-EIGS leads to a small reduction in degree compared to NLEIGS. An important reduction, however, is achieved using the set-valued variant, SV-AAA-EIGS: 17 poles are sufficient for an accuracy of 10^{-13} for the AAA set, whereas NLEIGS requires 31 poles. Both errors E_f and E_m are very comparable.

We then determine eigenvalue and eigenvector estimates of the rational eigenvalue problems, around the central point $\mathrm{Re}(s) = 250^2$, using the rational Krylov method. We used the same shifts as in Güttel *et al.* (2014), i.e., three shifts equally spaced on the real axis and two inside the semi-circle. Figure 1c shows residual norms of the five fastest converging Ritz values, as a function of the iteration count, for NLEIGS and SV-AAA-EIGS. The residual norms are defined as

$$\rho_i = \frac{\|A(\lambda_i)x_i\|_2}{\|A(\lambda_i)\|_1 \|x_i\|_2}, \tag{4.2}$$

where λ_i is the ith eigenvalue estimate, or Ritz value, and x_i is an associated Ritz vector. In the figure, comparable convergence behaviour is observed for both approaches, with slightly less accurate results for NLEIGS.

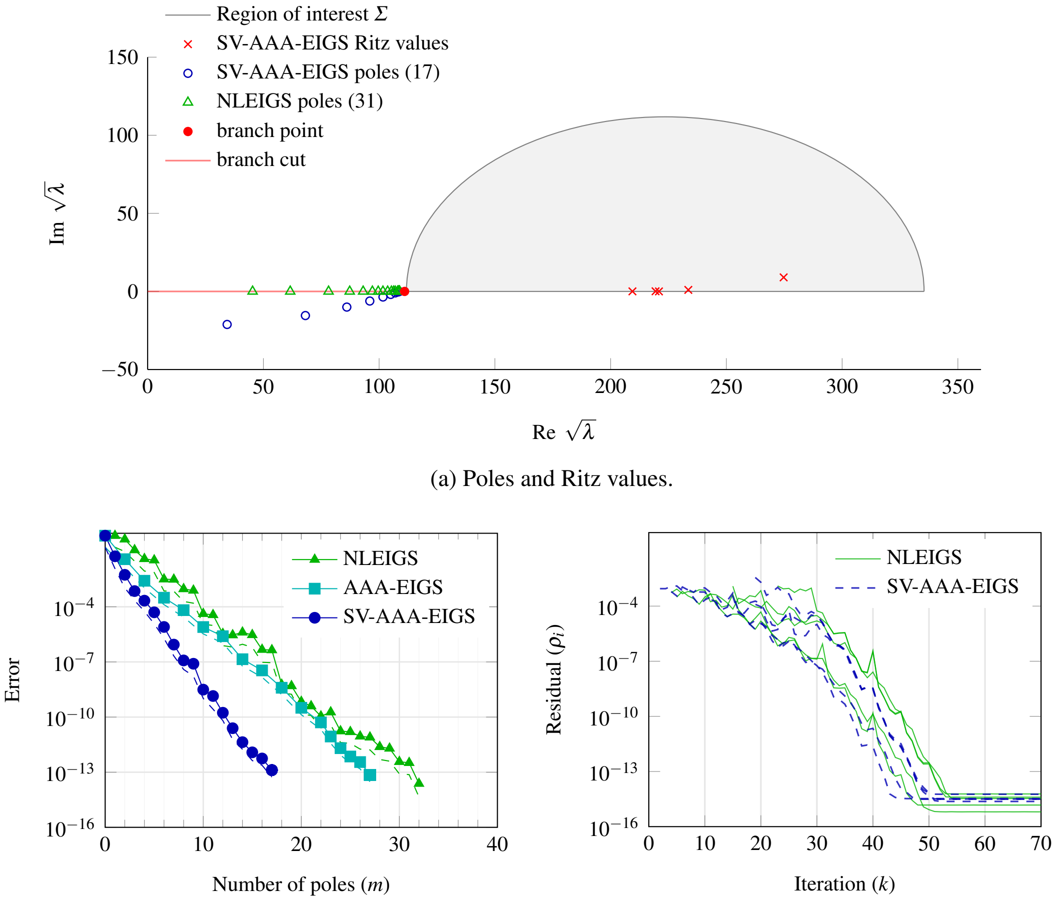
4.3 *Bound states in semiconductor devices*

Determining bound states of a semiconductor device requires the solution of the Schrödinger equation, that, after discretization, leads to a nonlinear eigenvalue problem with the matrix-valued function

$$A(\lambda) = H - \lambda I + \sum_{j=0}^{80} e^{i\sqrt{\lambda - \alpha_j}} S_j,$$

where $H, S_j \in \mathbb{R}^{16281 \times 16281}$; see Vandenberghe *et al.* (2014); Van Beeumen (2015). Matrix H is symmetric and matrices S_j have low rank. This function has 81 branch points on the real axis at $\lambda = \alpha_j,\ j = 0,\dots,80$, between -0.19 and 22.3, as can be seen in Figure 2. There is a branch cut running from $[-\infty, \alpha_0]$ and one between each branch point.

For approximating the nonlinear functions with Leja–Bagby points, in Vandenberghe *et al.* (2014), a transformation is used that removes the branch cut between two predetermined, subsequent branch points, i.e., for $\lambda \in [\alpha_{i-1}, \alpha_i]$. The interpolant based on these Leja–Bagby points is only valid for



(a) Poles and Ritz values.

(b) Error E_f (marked lines) and E_m (dashed lines).

(c) Residual norms (4.2).

FIG. 1: Results for the gun problem.

λ-values within this interval. For interval $[\alpha_0, \alpha_1]$, a rational approximation with 50 poles was used in Vandenberghe *et al.* (2014). This corresponds to the green triangular marker in Figure 2.

In contrast, using AAA and set-valued AAA, the 81 nonlinear functions are approximated on the real axis, over multiple branch points, without first transforming the problem. Figure 2 shows the resulting number of poles for approximating the nonlinear functions with an accuracy of 10^{-13} on a test set of 2000 equally spaced points, between α_0 and the following seven branch points, i.e., for $\lambda \in [\alpha_0, \alpha_i]$, $i = 1, \ldots, 7$. For example, the third bullet marker indicates the number of poles when the AAA test set runs from α_0 to α_3, so that it includes branchpoints $\alpha_0, \alpha_1, \alpha_2$ and α_3 and the branch cuts in between. We show the results for AAA and set-valued AAA, the second resulting in a significant reduction of the number of poles for the 81 functions in the example.

The two eigenvalues between α_0 and α_7 are situated in the first interval, $[\alpha_0, \alpha_1]$, as can be seen in Figure 2. We used the rational Krylov method with five equally spaced shifts from $\alpha_0 + \varepsilon$ to $\alpha_1 - \varepsilon$, where $\varepsilon = 10^{-2}$. The convergence behaviour, using a SV-AAA-EIGS with 141 poles for interval $[\alpha_0, \alpha_7]$, is shown in Figure 3. We used $\|A(\lambda)x\|_2$ as error measure, to be able to compare the results to those found in Van Beeumen (2015). The behaviour is comparable to that observed for the static variant of NLEIGS.

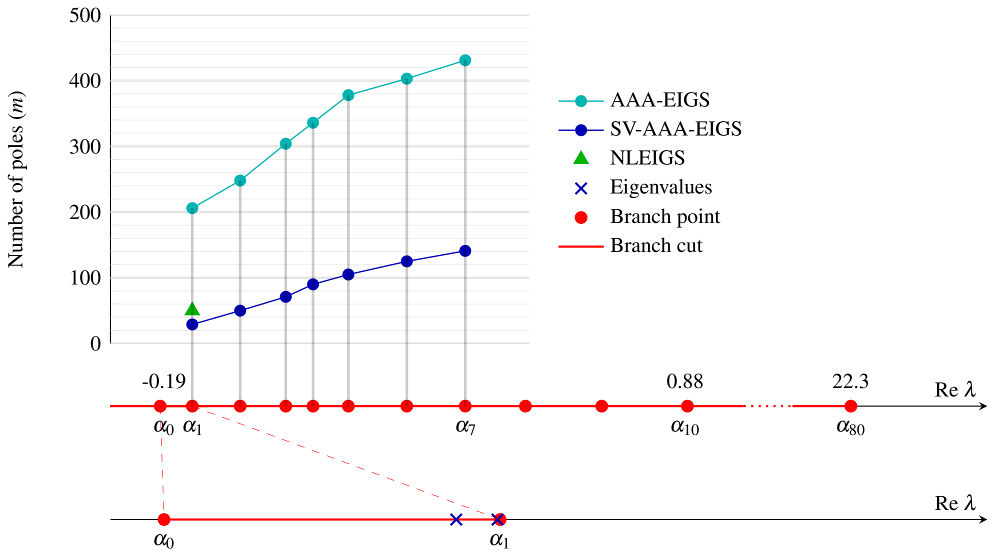


FIG. 2: Number of poles for the semiconductor problem when NLEIGS, AAA-EIGS, and SV-AAA-EIGS are applied to the seven intervals $[\alpha_0, \alpha_i]$ with $i = 1, \ldots, 7$.

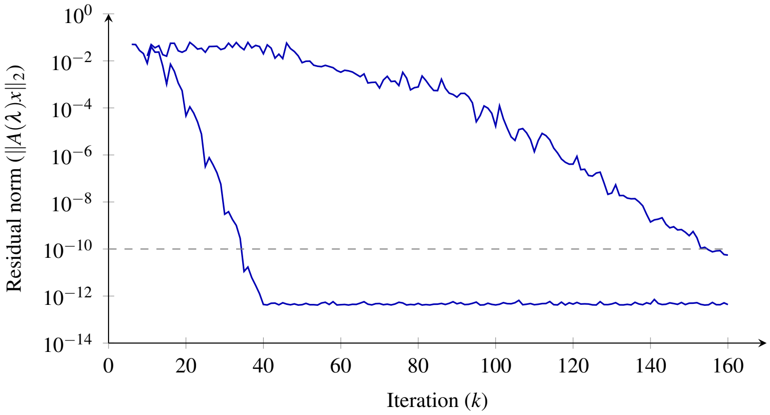


FIG. 3: Convergence behavior of the two Ritz values for set-valued AAA between α_0 and α_7, with 141 poles, for the semiconductor problem. The faster converging Ritz value corresponds to the smallest (leftmost) eigenvalue in Figure 2 near -0.14 and the slower converging value corresponds to the larger eigenvalue around -0.13.

4.4 *Sandwich beam*

A beam, consisting of two steel layers surrounding a damping layer, is modeled using the following matrix function

$$A(\lambda) = K - \lambda^2 M + \frac{G_0 + G_\infty (i\lambda\tau)^\alpha}{1 + (i\lambda\tau)^\alpha} C,$$

with $K, M, C \in \mathbb{R}^{168\times 168}$ symmetric positive semi-definite matrices; see Van Beeumen *et al.* (2013). Here, $G_0 = 350.4$kPa is the static shear modulus, $G_\infty = 3.062$MPa is the asymptotic shear modulus, $\tau = 8.23$ns is the relaxation time and $\alpha = 0.675$ a fractional parameter. Variable λ is the angular frequency and we are interested in eigenvalues in the range $\lambda \in [200, 30000]$.

We use the AAA algorithm to approximate the nonlinear term in $A(\lambda)$, where the sample set consists of 10^4 equidistant points within the frequency range, i.e., in the interval $[200, 30000]$. The algorithm converges with 11 poles. The location of the poles is shown in Figure 4 and the approximation error, on a random test set of 1000 points $\lambda \in [200, 30000]$, is shown in Figure 5a. Note that $\operatorname{Re}\lambda < -1$ and $\operatorname{Im}\lambda > 0$ for all poles, so that we can visualize the poles on a logarithmic axes, with on the negative real axis $-\log_{10}|\operatorname{Re}\lambda|$. Note also that we do not use set-valued AAA, as there is only one non-polynomial function. We similarly developed a rational approximation of degree 30 using NLEIGS in the rectangle with corners $200, 200+5000i, 30000, 30000+5000i$. The poles were chosen on the negative real axis. We also tried to use an NLEIGS approximation for the interval $[200, 30000]$, but the eigenvalues further away from the interval did not converge to eigenvalues of the nonlinear eigenvalue problem.

We used the rational Krylov method to obtain eigenvalue and eigenvector estimates of $A(\lambda)$. We used 10 shifts, $[2, 5, 10, 100, 200, 210, 220, 230, 240, 250] \cdot 100$, i.e., with more shifts located near the end of the interval. This results in the Ritz values also shown in Figure 4, on the same logarithmic axes, since $\operatorname{Re}\lambda > 1$ for all Ritz values. The residual norms (4.2) can be seen in Figure 5b. They are compared to the residual norms of NLEIGS. Note that because of the large norm of K, most residual norm values lie around 10^{-8} for $k = 1$. After 50 iterations, the residual norms decrease by a factor 10^{-4} for most Ritz values.

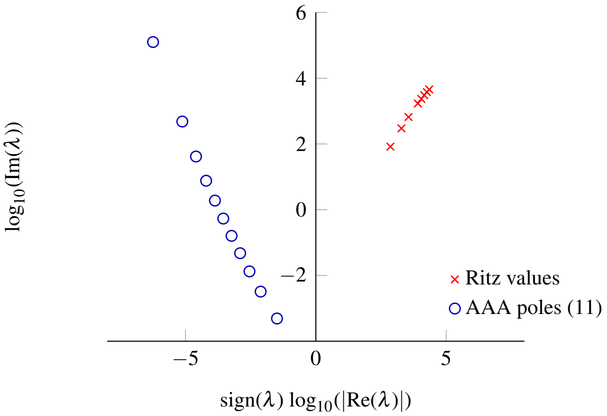


FIG. 4: AAA poles (○) and Ritz values (×) for the sandwich beam.

4.5 *Car cavity problem*

The following model was generated by Dr. Axel van de Walle from KU Leuven, using a mesh from a Siemens tutorial, and using poro-elastic material properties from Allard & Noureddine (2009) and Chazot *et al.* (2013). The following matrix-valued function describes the nonlinear behaviour of the sound pressure inside a car cavity with porous seats:

$$A(\lambda) = K_0 + h_K(\lambda)K_1 - \lambda^2(M_0 + h_M(\lambda)M_1),$$

where $K_0, K_1, M_0, M_1 \in \mathbb{R}^{15036\times15036}$ are symmetric positive semidefinite matrices and λ is the angular frequency $\lambda = 2P_0 f$. The nonlinear functions are given by

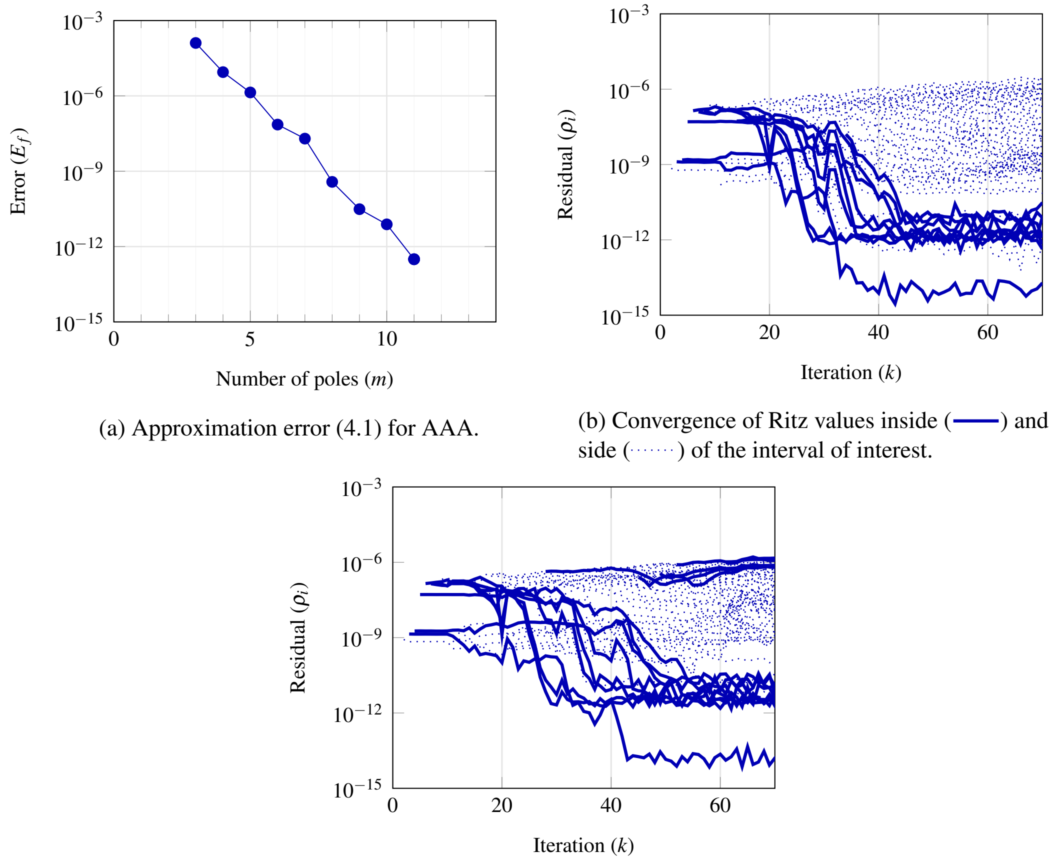
$$h_K(\lambda) = \frac{\phi}{\alpha(\lambda)}, \quad \alpha(\lambda) = \alpha_\infty + \frac{\sigma\phi}{i\lambda\rho_0}\sqrt{1 + i\lambda\rho_0\frac{4\alpha_\infty^2\eta}{\sigma^2\Lambda^2\phi^2}},$$

and

$$h_M(\lambda) = \phi\left(\gamma - \frac{\gamma-1}{\alpha'(\lambda)}\right), \quad \alpha'(\lambda) = 1 + \frac{8\eta}{i\lambda\rho_0\Lambda'^2P_r}\sqrt{1 + i\lambda\rho_0\frac{\Lambda'^2P_r}{16\eta}},$$

with the parameters defined in Table 2. Function h_K is unbounded around $\lambda = 514i$ and has a branch point around $\lambda = 619i$, with a branch cut on the imaginary axis. The second nonlinear function h_M is unbounded around $\lambda = 815i$ and has a branch point at $\lambda = 2089i$. These points and branch cuts are shown in Figure 6.

Since the physical model includes damping, we need to take into account that the eigenvalues have positive imaginary parts. We chose a test set with $5\cdot10^4$ real values, $\mathrm{Re}(\lambda)\in[1,300]$, and $5\cdot10^4$



(a) Approximation error (4.1) for AAA.

(b) Convergence of Ritz values inside (——) and outside (⋯⋯) of the interval of interest.

(c) Convergence of Ritz values inside (——) and outside (⋯⋯) of the interval of interest for NLEIGS.

FIG. 5: Approximation error and residuals for the sandwich beam.

random values in the rectangle with corner points 0 and $300+10^4 i$, i.e., with very large imaginary part. We can see from Figure 6 that AAA chooses some poles very close to the singularities of the nonlinear functions. The degree of the set valued rational approximation is 42 for a tolerance of 10^{-13}.

In order to compare with NLEIGS, we reduced the test set to the rectangle with corners 0 and $300+510i$. In this way, the set of points remains below the first singular point of the function around $514i$. The Leja–Bagby poles are picked on the imaginary axis, starting at $\mathrm{Im}(\lambda) = 514$. This is shown in Figure 7. The degree necessary to reach a relative accuracy of 10^{-12} on the border of the rectangle was 40. We found that AAA reaches the same required accuracy for a polynomial of degree 11. We compared the results obtained by the rational Krylov method for both approximations, where we used 10 equally spaced shifts on the real axis $\mathrm{Re}(\lambda)$, from 1 to 300. Figure 8 shows some of the Ritz values together with the number of Krylov iterations required to reach a residual norm (4.2) below 10^{-12}. These Ritz values have low imaginary parts, confirming our choice of test set, and, as for the gun problem,

Table 2: Constants of the car cavity model

α_∞	1.7	σ	$13500 kgm^{-3}s^{-1}$	ϕ	0.98
η	$1.839 \cdot 10^{-5}$	Λ	$80 \cdot 10^{-6} m$	Λ'	$160 \cdot 10^{-6} m$
γ	1.4	ρ_0	1.213	P_r	0.7217

convergence is comparable for both methods.

5. Conclusions

We have proposed a method for solving the nonlinear eigenvalue problem by first approximating the associated nonlinear matrix-valued function using the AAA algorithm. This approximation is embedded in a state space representation, which leads to a linearization that can be solved using the Compact Rational Krylov method. We presented two versions: one that approximates each function separately and then the set-valued version that approximates all functions together. The latter version is very competitive with NLEIGS in terms of degree of the rational approximation, and in all our tests, AAA requires less poles than an approximation using potential theory, even for a problem with eighty different functions. The main advantage of the method is the fully automatic procedure in the determination of the rational approximations. Contrary to NLEIGS, we expect that the construction of the AAA approximation may become prohibitively expensive when the number of nonlinear functions is extremely high, i.e, in the order of the size of the matrix.

Acknowledgment

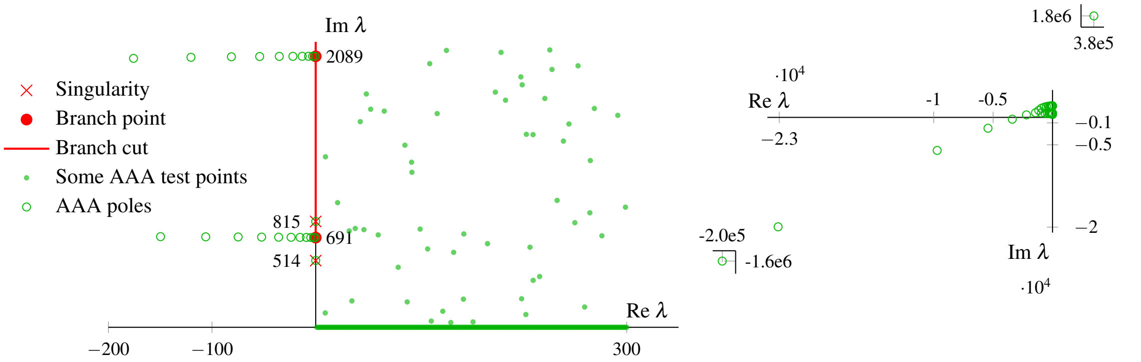
The authors thank Dr. Axel van de Walle for providing us with the car cavity model that he generated for us using a model from a Virtual Lab tutorial (formerly LMS International, currently Siemens Industry Software).

References

ALLARD, J. & NOUREDDINE, A. (2009) *Propagation of sound in porous media: modelling sound absorbing materials*, 2nd edn. John Wiley & Sons.

AMPARAN, A., DOPICO, F. M., MARCAIDA, S. & ZABALLA, I. (2018) Strong linearizations of rational matrices. *SIAM Journal on Matrix Analysis and Applications*, **39**, 1670–1700.

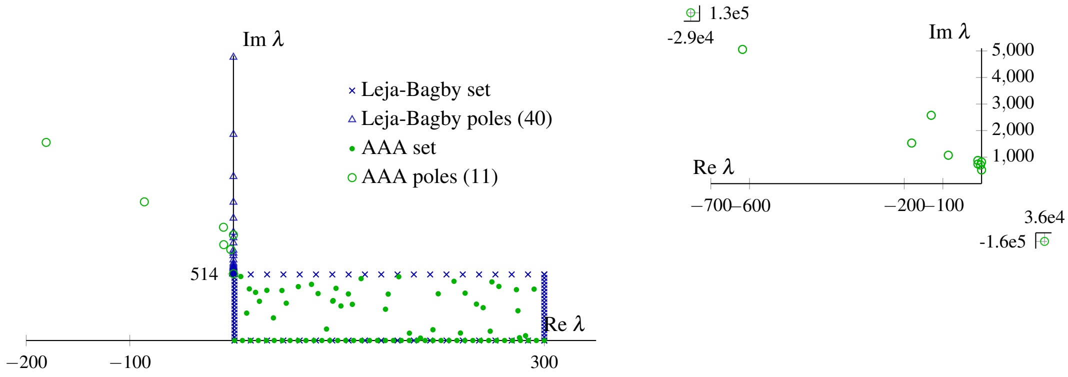
BETCKE, T., HIGHAM, N. J., MEHRMANN, V., SCHRÖDER, C. & TISSEUR, F. (2013) NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans. Math. Softw.*, **39**, 7:1–7:28.



(a) Branch points and cuts and values for $\mathrm{Re}(\lambda) \in [-200, 300]$ and the poles nearest the two singularities.

(b) All of the AAA poles.

FIG. 6: Some test points and poles of AAA for the car cavity problem for large imaginary values.



(a) Leja–Bagby and AAA set and poles.

(b) All of the AAA poles.

FIG. 7: Leja–Bagby and AAA sets for the car cavity problem, up to $514i$, and accompanying poles.

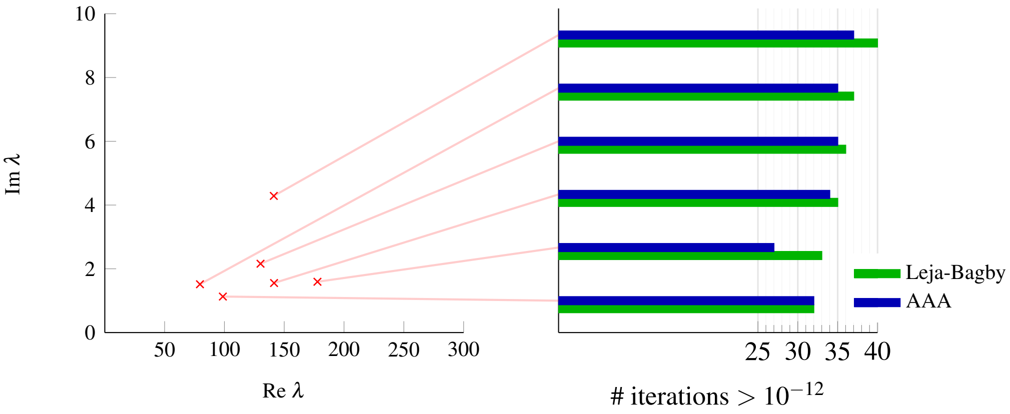


FIG. 8: Ritz values and number of iterations to reach tolerance 10^{-12} for the car cavity problem.

BEYN, W.-J. (2012) An integral method for solving nonlinear eigenvalue problems. *Linear Algebra and its Applications*, **436**, 3839–3863.

CHAZOT, J.-D., NENNIG, B. & PERREY-DEBAIN, E. (2013) Performances of the partition of unity finite element method for the analysis of two-dimensional interior sound fields with absorbing materials. *Journal of Sound and Vibration*, **332**, 1918–1929.

DOPICO, F. M., LAWRENCE, P. W., PÉREZ, J. & DOOREN, P. V. (2018) Block Kronecker linearizations of matrix polynomials and their backward errors. *Numerische Mathematik*, **140**, 373–426.

DOPICO, F. M. & GONZÁLEZ-PIZARRO, J. (2018) A compact rational Krylov method for large-scale rational eigenvalue problems. *Numerical Linear Algebra with Applications*.

EFFENBERGER, C. (2013) Robust successive computation of eigenpairs for nonlinear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, **34**, 1231–1256.

EFFENBERGER, C. & KRESSNER, D. (2012) Chebyshev interpolation for nonlinear eigenvalue problems. *BIT*, **52**, 933–951.

ELSWORTH, S. & GÜTTEL, S. (2018) Conversions between barycentric, RKFUN, and Newton representations of rational interpolants. *Linear Algebra and its Applications*.

FILIP, S., NAKATSUKASA, Y., TREFETHEN, L. & BECKERMANN, B. (2018) Rational minimax approximation via adaptive barycentric representations. *SIAM Journal on Scientific Computing*, **40**, A2427–A2455.

GOHBERG, I., LANCASTER, P. & RODMAN, L. (1982) *Matrix Polynomials*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.

GUSTAVSEN, B. & SEMLYEN, A. (1999) Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on Power Delivery*, **14**, 1052–1061.

GÜTTEL, S., VAN BEEUMEN, R., MEERBERGEN, K. & MICHIELS, W. (2014) NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems. *SIAM Journal on Scientific Computing*, **36**, A2842–A2864.

GÜTTEL, S. & TISSEUR, F. (2017) The nonlinear eigenvalue problem. *Acta Numerica*, **26**, 1–94.

HOCHMAN, A. (2017) FastAAA: A fast rational-function fitter. *2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS). 2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS).*, pp. 1–3.

IONITĂ, A. C. (2013) Lagrange rational interpolation and its applications to approximation of large-scale dynamical systems. *Ph.D. thesis*, Rice University.

JARLEBRING, E., MICHIELS, W. & MEERBERGEN, K. (2012) A linear eigenvalue algorithm for the nonlinear eigenvalue problem. *Numerische Mathematik*, **122**, 169–195.

LIETAERT, P., MEERBERGEN, K. & TISSEUR, F. (2018) Compact two-sided Krylov methods for nonlinear eigenvalue problems. *SIAM Journal on Scientific Computing*, **40**, A2801–A2829.

NAKATSUKASA, Y., SÈTE, O. & TREFETHEN, L. N. (2018) The AAA algorithm for rational approximation. *SIAM Journal on Scientific Computing*, **40**, A1494–A1522.

ROBOL, L., VANDEBRIL, R. & VAN DOOREN, P. (2016) A framework for structured linearizations of matrix polynomials in various bases. *SIAM Journal on Matrix Analysis and Applications*, **38**, 188–216.

SU, Y. & BAI, Z. (2011) Solving rational eigenvalue problems via linearization. *SIAM Journal on Matrix Analysis and Applications*, **32**, 201–216.

VAN BEEUMEN, R., MEERBERGEN, K. & MICHIELS, W. (2013) A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems. *SIAM Journal on Scientific Computing*, **35**, A327–A350.

VAN BEEUMEN, R., MEERBERGEN, K. & MICHIELS, W. (2015) Compact rational Krylov methods for nonlinear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, **36**, 820–838.

VAN BEEUMEN, R. (2015) *Rational Krylov Methods for Nonlinear Eigenvalue Problems*. Ph.D. thesis, KU Leuven.

VANDENBERGHE, W. G., FISCHETTI, M. V., VAN BEEUMEN, R., MEERBERGEN, K., MICHIELS, W. & EFFENBERGER, C. (2014) Determining bound states in a semiconductor device with contacts using a nonlinear eigenvalue solver. *Journal of Computational Electronics*, **13**, 753–762.

A. Proofs of Theorems 3.3 and 3.4

The following simple lemma turns out be useful.

LEMMA A.1 Let $G_{22} \in \mathbb{C}^{m\times m}$ and $X_1 \in \mathbb{C}^{n\times n}$ be invertible matrices that satisfy

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} Y_1 \\ O \end{bmatrix}, \tag{A.1}$$

then the following block-UL decomposition holds:

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} I_n & G_{12}G_{22}^{-1} \\ O & I_m \end{bmatrix} \begin{bmatrix} Y_1X_1^{-1} & O \\ G_{21} & G_{22} \end{bmatrix}. \tag{A.2}$$

Proof. After block elimination of the square matrix on the left-hand size of (A.2), we only need to show that its Schur complement $S = G_{11} - G_{12}G_{22}^{-1}G_{21}$ equals $Y_1X_1^{-1}$. But this follows directly from the identities $G_{11} = (Y_1 - G_{12}X_2)X_1^{-1}$ and $X_2 = -G_{22}^{-1}G_{21}X_1$ that are implied by (A.1). □

Proof of Theorem 3.3. Identity (3.15) follows from direct manipulation together with $\det(A \otimes I_n) = (\det(A))^n$. To show the block-UL decomposition, recall from (3.13) and (3.14) that

$$(\widetilde{\mathscr{L}_R}(\lambda)\mathscr{P})\,(\mathscr{P}^T\Psi(\lambda)) = \left[\begin{array}{c} R(\lambda) \\ \hline \boldsymbol{O} \end{array}\right] \text{ with } \widetilde{\mathscr{L}_R}(\lambda)\mathscr{P} = \left[\begin{array}{c|cc} \boldsymbol{A}_0 - \lambda\boldsymbol{B}_0 & \boldsymbol{A}_1 - \lambda\boldsymbol{B}_1 & \boldsymbol{C} - \lambda\boldsymbol{D} \\ \hline \boldsymbol{M}_0 - \lambda\boldsymbol{N}_0 & \boldsymbol{M}_1 - \lambda\boldsymbol{N}_1 & \boldsymbol{O} \\ \boldsymbol{Z}_0^* & \boldsymbol{Z}_1^* & \boldsymbol{E} - \lambda\boldsymbol{F} \end{array}\right].$$

The vertical and horizontal lines indicate compatible block partitioning. The corresponding partitioning for $\mathscr{P}^T\Psi(\lambda)$ satisfies

$$\mathscr{P}^T\Psi(\lambda) = \left[\begin{array}{c} p_1I_n \\ \hline p_2I_n \\ \vdots \\ p_kI_n \\ (R_1(\lambda)\otimes I_{k_1})\widetilde{Z}_1^* \\ \vdots \end{array}\right] \text{ with } p = \begin{bmatrix} p_1 \\ \vdots \\ p_k \end{bmatrix} := P_0^T f(\lambda) \in \mathbb{C}^k.$$

The required block-UL decomposition now follows from a direct calculation if we can apply Lemma A.1 to the partitioned matrix $\widetilde{\mathscr{L}_R}(\lambda)\mathscr{P}$. In order to be able to do this, we only have to establish that $p_1 = e_1^TP_0^Tf(\lambda) \neq 0$ since then $X_1 = p_1I_n$ is invertible (and we already have $Y_1 = R(\lambda)$). To this end, we use the definition of P_0 to obtain

$$(M - \lambda N)f(\lambda) = 0 \iff \begin{bmatrix} m_0 - \lambda n_0 & M_1 - \lambda N_1 \end{bmatrix} p = 0.$$

Since $M_1 - \lambda N_1$ is invertible and $\dim\operatorname{Ker}(M - \lambda N) = 1$, the null-vector p has to be of the form

$$p = \alpha \begin{bmatrix} 1 \\ -(m_0 - \lambda n_0)(M_1 - \lambda N_1)^{-1} \end{bmatrix}, \qquad \alpha \in \mathbb{C}.$$

Since $f(\lambda)$, and thus p, cannot ever be identically zero, we get as requested that $\alpha = p_1 \neq 0$. □

Proof of Theorem 3.4. (a) Let (λ_0, x) be an eigenpair of $R(\lambda)$, that is, $R(\lambda_0)x = 0$. Recalling that

$f(\lambda)^T = \begin{bmatrix} f_0(\lambda) & \cdots & f_{k-1}(\lambda) \end{bmatrix}$ with $f_0(\lambda) \equiv 1$, we obtain

$$y = \Psi(\lambda_0)x = \begin{bmatrix} x \\ f_1(\lambda_0)x \\ \vdots \\ f_{k-1}(\lambda_0)x \\ (R_1(\lambda_0) \otimes I_{k_1})\widetilde{Z}_1^* x \\ \vdots \\ (R_s(\lambda_0) \otimes I_{k_s})\widetilde{Z}_s^* x \end{bmatrix}$$

and thus also $y \neq 0$ due to $x \neq 0$. Using (3.13), we see that (λ_0, y) verifies the eigenpair equation

$$\widetilde{\mathscr{L}}(\lambda_0)y = \begin{bmatrix} R(\lambda_0) \\ \boldsymbol{O} \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

(b) Let (λ_0, y) be an eigenpair of $\widetilde{\mathscr{L}}(\lambda)$. Thanks to (3.13), it suffices to show that $y = \Psi(\lambda_0)x$ for some nonzero $x \in \mathbb{C}^n$ since that implies $R(\lambda_0)x = 0$. To this end, consider the eigenpair equation $\widetilde{\mathscr{L}}(\lambda_0)y = 0$ in partitioned form:

$$\left[\begin{array}{c|c} \boldsymbol{A} - \lambda_0 \boldsymbol{B} & \boldsymbol{C} - \lambda_0 \boldsymbol{D} \\ \boldsymbol{M} - \lambda_0 \boldsymbol{N} & \boldsymbol{O} \\ \boldsymbol{Z}^* & \boldsymbol{E} - \lambda_0 \boldsymbol{F} \end{array}\right] \left[\begin{array}{c} y_{AB} \\ \hline y_{CD} \end{array}\right] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The second block-row expresses that y_{AB} is a null vector of $\boldsymbol{M} - \lambda_0 \boldsymbol{N} = (M - \lambda_0 N) \otimes I_n$. By definition of $f(\lambda)$, we have $(M - \lambda N) f(\lambda) = 0$ from which $(\boldsymbol{M} - \lambda \boldsymbol{N})(f(\lambda) \otimes I_n) = 0$. Since $f_0(\lambda) \equiv 1$, the $kn \times n$ matrix $F(\lambda) := f(\lambda) \otimes I_n$ has rank n. In addition, by rank nullity, $\dim \operatorname{Ker}(\boldsymbol{M} - \lambda \boldsymbol{N}) = n$ and so $F(\lambda)$ is a basis for $\operatorname{Ker}(\boldsymbol{M} - \lambda \boldsymbol{N})$. Hence, there exists $x \in \mathbb{C}^n$ such that (recall $f_0(\lambda) \equiv 1$)

$$y_{AB} = F(\lambda_0)x = \begin{bmatrix} x \\ f_1(\lambda_0)x \\ \vdots \\ f_{k-1}(\lambda_0)x \end{bmatrix}. \tag{A.3}$$

The third block-row reads

$$(\boldsymbol{E} - \lambda_0 \boldsymbol{F}) y_{CD} = -\boldsymbol{Z}^* y_{AB}.$$

Together with the definitions of $\boldsymbol{Z}, \boldsymbol{E}, \boldsymbol{F}$ and (A.3), we also obtain

$$\begin{bmatrix} (E_1 - \lambda_0 F_1) \otimes I_{k_1} & & \\ & \ddots & \\ & & (E_s - \lambda_0 F_s) \otimes I_{k_s} \end{bmatrix} y_{CD} = \begin{bmatrix} (b_1 \otimes I_{k_1})\widetilde{Z}_1^* \\ \vdots \\ (b_s \otimes I_{k_s})\widetilde{Z}_s^* \end{bmatrix} (e_1^T f(\lambda_0) \otimes I_n)x.$$

Hence, isolating for y_{CD} and using $R_i(\lambda) = (E_i - \lambda F_i)^{-1} b_i$, we get

$$y_{CD} = \begin{bmatrix} ((E_1 - \lambda_0 F_1)^{-1} \otimes I_{k_1})(b_1 \otimes I_{k_1})\widetilde{Z}_1^* \\ \vdots \\ ((E_s - \lambda_0 F_s)^{-1} \otimes I_{k_s})(b_s \otimes I_{k_s})\widetilde{Z}_s^* \end{bmatrix} x = \begin{bmatrix} (R_1(\lambda_0) \otimes I_{k_1})\widetilde{Z}_1^* \\ \vdots \\ (R_s(\lambda_0) \otimes I_{k_s})\widetilde{Z}_s^* \end{bmatrix} x. \tag{A.4}$$

Now combining (A.3) and (A.4), we have indeed shown that there exists $x \in \mathbb{C}^n$ such that

$$y = \left[\begin{array}{c} y_{AB} \\ \hline y_{CD} \end{array}\right] = \Psi(\lambda_0)x.$$

Observe that $\Psi(\lambda)$ has full column rank thanks to $F(\lambda)$ being its upper block. Hence, from $y \neq 0$ it follows that $x \neq 0$ and we haven proven (b).

(c) The statement about the algebraic multiplicity of λ_0 follows directly from (3.15) since the matrices $M_1 - \lambda_0 N_1$ and $E_i - \lambda_0 F_i$ are invertible by construction of P_0 and by assumption, respectively, and $\alpha \neq 0$. Next, we show the equality of the geometric multiplicity of λ_0, that is, $\dim\ker R(\lambda_0) = \dim\ker \widetilde{\mathscr{L}_R}(\lambda_0)$. Let $\{x_1,\ldots,x_t\}$ be a basis for $\ker R(\lambda_0)$. Then by (a), $\Psi(\lambda_0)x_i \in \ker \widetilde{\mathscr{L}_R}(\lambda_0)$ for $i = 1,\ldots,t$. As argued above $\Psi(\lambda_0)$ has full column rank, hence the vectors $\Psi(\lambda_0)x_1,\ldots,\Psi(\lambda_0)x_t$ are linearly independent and so $\dim\ker R(\lambda_0) \leqslant \dim\ker \widetilde{\mathscr{L}_R}(\lambda_0)$. Similarly, if $\{y_1,\ldots,y_t\}$ is a basis for $\ker \widetilde{\mathscr{L}_R}(\lambda_0)$, then by (b), $y_i = \Psi(\lambda_0)x_i$ for some $x_i \in \mathbb{C}^n$. Again due to $\Psi(\lambda_0)$ having full column rank, the $x_1,\ldots,x_t$ are linearly independent. Hence, $\dim\ker R(\lambda_0) \geqslant \dim\ker \widetilde{\mathscr{L}_R}(\lambda_0)$, as we wanted to show. □