The RKFIT algorithm for nonlinear rational approximation

Berljafa, Mario and Güttel, Stefan

2015

MIMS EPrint: **2015.38**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

THE RKFIT ALGORITHM FOR NONLINEAR RATIONAL APPROXIMATION

MARIO BERLJAFA* AND STEFAN GÜTTEL*

Dedicated to the memory of Axel Ruhe (1942–2015)

Abstract. The RKFIT algorithm outlined in [M. Berljafa and S. Güttel, *Generalized rational Krylov decompositions with an application to rational approximation,* SIAM J. Matrix Anal. Appl., 2015] is a Krylov-based approach for solving nonlinear rational least squares problems. This paper puts RKFIT into a general framework, allowing for its extension to nondiagonal rational approximants and a family of approximants sharing a common denominator. Furthermore, we derive a strategy for the degree reduction of the approximants, as well as methods for their conversion to partial fraction form, for the efficient evaluation, and root-finding. We also discuss commons and differences of RKFIT and the popular vector fitting algorithm. A MATLAB implementation of RKFIT is provided and numerical experiments, including the fitting of a MIMO dynamical system and an optimization problem related to exponential integration, demonstrate its applicability.

Key words. nonlinear rational approximation, least squares, rational Krylov method

AMS subject classifications. 15A22, 65F15, 65F18, 30E10

1. Introduction.

Rational approximation problems arise in many areas of engineering and scientific computing. A prominent example is that of system identification and model order reduction, where calculated or measured frequency responses of dynamical systems are approximated by (low-order) rational functions [20, 26, 2, 23, 19]. Some other areas where rational approximants play an important role are analogue filter design [7], time stepping methods [43], transparent boundary conditions [28, 17], and iterative methods in numerical linear algebra (see, e.g., [32, 42, 18, 27, 33]). Here we focus on discrete rational approximation in the least squares (LS) sense.

In its simplest form the weighted rational LS problem is the following: given data pairs $\{(\lambda_i, f_i)\}_{i=1}^N$ with pairwise distinct λ_i, and positive weights $\{w_i\}_{i=1}^N$, find a rational function r of type (m, m), that is, numerator and denominator of degree at most m, such that

$$\sum_{i=1}^{N} w_i |f_i - r(\lambda_i)|^2 \to \min. \tag{1.1}$$

The weights can be used to assign varying relevance to the data points. For example, when the function values f_i are known to be perturbed by white Gaussian noise, then the w_i can be chosen inversely proportional to the variance.

Even in their simplest form (1.1), rational LS problems are challenging. Finding a rational function $r = p_m/q_m$ in (1.1) corresponds to a nonlinear minimization problem as the denominator q_m is generally unknown, and solutions may depend discontinuously on the data, be non-unique, or even non-existent. An illustrating example inspired by Braess [10, p. 109] is to take fixed $m \geq 1$ and $N > 2m$, and let

$$\lambda_i = \frac{i-1}{N}, \quad \text{and} \quad f_i = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{if } 2 \leq i \leq N. \end{cases} \tag{1.2}$$

* School of Mathematics, The University of Manchester, Alan Turing Building, Oxford Road, M13 9PL Manchester, United Kingdom, mario.berljafa@manchester.ac.uk, stefan.guettel@manchester.ac.uk

Then the sequence of rational functions $r_j(z) = 1/(1+jz)$ makes the misfit for (1.1) arbitrarily small as $j \to \infty$, but the f_i do not correspond to values of a type (m,m) rational function (there are too many roots). Hence a rational LS solution does not exist. If, however, the data f_i are slightly perturbed to $\widehat{f}_i = r_j(\lambda_i)$ for an arbitrarily large j, then of course r_j itself *is* a LS solution to (1.1).

A very common approach for solving (1.1) approximately is linearisation. Consider again the data (1.2) and the problem of finding polynomials p_m and q_m of degree at most m such that

$$\sum_{i=1}^{N} w_i |f_i q_m(\lambda_i) - p_m(\lambda_i)|^2 \to \min. \tag{1.3}$$

This problem has a trivial solution with $q_m \equiv 0$ and to exclude it we need some normalisation like, for example, a "point-wise" condition $q_m(0) = 1$. Under this condition the linear problem (1.3) is guaranteed to have a nontrivial solution (p_m, q_m), but the solution is clearly not unique; since $f_i = 0$ for $i \geq 2$, any admissible denominator polynomial q_m with $q_m(0) = 1$ corresponds to a minimal solution with $p_m \not\equiv 0$. On the other hand, for the normalisation condition $q_m(1) = 1$, the polynomials $q_m(z) = z$ and $p_m \equiv 0$ solve (1.3) with zero misfit. This example shows that linearised rational LS problems can have non-unique solutions, and these may depend on normalisation conditions. With both normalisation conditions, the rational function $r = p_m/q_m$ with (p_m, q_m) obtained from solving the linearised problem (1.3) may yield an arbitrarily large (or even infinite) misfit for the nonlinear problem (1.1).

The pitfalls of nonlinear and linearised rational approximation problems have not prevented the development of algorithms for their solution. An interesting overview of algorithms for the nonlinear problem based on repeated linearisation, such as Wittmeyer's algorithm, is given in [3]. Robust solution methods for the linearised problem using regularised SVD are discussed in [22, 21].

The aim of this paper is to present and analyse an extension of the RKFIT algorithm initially outlined in [6]. RKFIT is an iterative method for solving rational LS problems more general than (1.1). For given matrices $\{A, F\} \subset \mathbb{C}^{N\times N}$ and a vector $\boldsymbol{b} \in \mathbb{C}^N$, RKFIT attempts to find a rational function r such that

$$\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2^2 \to \min. \tag{1.4}$$

Note that this problem contains (1.1) as a special case with $F = \operatorname{diag}(f_i)$, $A = \operatorname{diag}(\lambda_i)$, $\boldsymbol{b} = [\sqrt{w_1} \quad \dots \quad \sqrt{w_N}]^T$. For RKFIT the matrices A and F are not required to be diagonal. In many applications F is a matrix function of A or an approximation thereof, i.e., $F = f(A)$ or $F \approx f(A)$.

A main contribution of this work compared to [6] is the extension of RKFIT to *nondiagonal* approximants, i.e., allowing to compute rational functions of the general type $(m+k, m)$ with $k \geq -m$. Further, we extend RKFIT to rational approximation problems involving a family of matrices $\{F^{[j]}\}_{j=1}^{\ell} \subset \mathbb{C}^{N\times N}$, and a block of vectors $B = [\boldsymbol{b}_1 \quad \dots \quad \boldsymbol{b}_n] \in \mathbb{C}^{N\times n}$. More precisely, we seek a family of rational functions $\{r^{[j]}\}_{j=1}^{\ell}$ of type $(m+k, m)$, all sharing a common denominator q_m, such that the *relative misfit* is minimal, i.e.,

$$\text{misfit} = \sqrt{\frac{\sum_{j=1}^{\ell} \|D^{[j]}[F^{[j]}B - r^{[j]}(A)B]\|_F^2}{\sum_{j=1}^{\ell} \|D^{[j]}F^{[j]}B\|_F^2}} \to \min. \tag{1.5}$$

The matrices $F^{[j]}$ may, for instance, correspond to values of a parameter-dependent matrix function like $F^{[j]} = \exp(-t_j A)$, and in section 6 we consider an application of such a problem. The matrices $D^{[j]}$ act as *element-wise weights*, whereas the vectors in B can be viewed as *spectral weights* relative to the eigenpairs of A.

To summarize our terminology, here is a list of the data in problem (1.5):

$$\begin{aligned} A: &\quad \text{interpolation node matrix of size } N \times N, \\ F^{[j]}: &\quad \text{interpolation data matrices of size } N \times N, \\ D^{[j]}: &\quad \text{element-wise weight matrices of size } N \times N, \\ B: &\quad \text{block of spectral weight vectors, an } N \times n \text{ matrix}, \\ r^{[j]}: &\quad \text{rational functions sharing the same denominator } q_m, \\ (m+k, m): &\quad \text{type of the rational functions } r^{[j]} \text{ with } k \geq -m. \end{aligned}$$

We show how rational Krylov techniques can be used to tackle problems of the form (1.5). The outgrowth of this work is a new MATLAB implementation of RKFIT, which is part of the *Rational Krylov Toolbox* [5] available online[1]. One particularity of RKFIT is its ease of use. For example, with $\ell = 1$ and the matrices `A`, `F`, `B` and a vector of initial poles `xi` being defined in MATLAB, the user simply calls

```
[xi, r, misfit] = rkfit(F, A, B, xi)
```

to obtain a rational function `r` represented as a MATLAB object of class RKFUN, which stands for *rational Krylov function*. The toolbox implements several RKFUN methods, for example, the evaluation of r at scalar arguments or as a matrix function; the commands `r(z)` and `r(A,B)` evaluate $r(z)$ and $r(A)B$, respectively (where A and B can be different from the matrices used for the construction of `r`). The conversion of an RKFUN to partial fraction form (the `residue` command), root-finding (`roots`), or easy-to-use plotting (`ezplot`) are provided as well. The use of MATLAB's object-oriented programming capabilities for these purposes is inspired by the Chebfun system [14].

Alongside the extension of RKFIT to nondiagonal approximants in section 2, another contribution of this paper is Theorem 2.2 which shows that RKFIT solves (1.4) exactly if F is a rational matrix function of type $(m+k, m)$. In section 3 we propose a procedure for automatically decreasing the degree parameters m and k, thereby reducing possible deficiencies in the rational approximants. That section also contains Theorem 3.1, which relates the roots of a rational Krylov function to the eigenvalues of a matrix pencil. Based on this theorem, we present a new procedure to obtain good starting guesses for RKFIT after a degree reduction has been performed.

We point out that initially, in sections 2 and 3, we only consider problem (1.4), which is a special case of (1.5) for a single rational function ($\ell = 1$) and a single vector $B = \boldsymbol{b}$ ($n = 1$). The generalization to the full problem (1.5) is discussed in section 4. In section 5 we develop a new approach for the efficient evaluation of the RKFUNs produced by RKFIT. We also show how to compute the roots of RKFUNs and how to convert them into partial fraction form. Numerical examples are given in section 6, including the fitting of a MIMO dynamical system and a new pole optimization approach for exponential integration. An appendix discusses the connections of RKFIT and other approximation algorithms, in particular, the popular vector fitting method [26].

[1] `http://rktoolbox.org`

Algorithm 2.1 High-level description of RKFIT. RKToolbox [5]: `rkfit`

1. Take initial guess for q_m.
2. **repeat**
3. Set search space $\mathcal{S} := \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$.
4. Set target space $\mathcal{T} := \mathcal{K}_{m+k+1}(A, q_m(A)^{-1}\boldsymbol{b})$.
5. Find $\widehat{\boldsymbol{v}} = \operatorname{argmin}_{\substack{\boldsymbol{v}\in\mathcal{S} \\ \|\boldsymbol{v}\|_2=1}} \|(I - P_{\mathcal{T}})F\boldsymbol{v}\|_2$.
6. Let $\widehat{q}_m \in \mathcal{P}_m$ be such that $\widehat{\boldsymbol{v}} = \widehat{q}_m(A)q_m(A)^{-1}\boldsymbol{b}$.
7. Set $q_m := \widehat{q}_m$.
8. **until** stopping criteria is satisfied.
9. Construct wanted rational approximant r.

2. The RKFIT algorithm. The nondiagonal version of the RKFIT algorithm considered here aims to find a rational function $r = p_{m+k}/q_m$ of type $(m+k, m)$ which solves problem (1.4). As the denominator q_m is not known and (1.4) depends nonlinearly on it, RKFIT tries to iteratively improve a starting guess for q_m by solving a linearised problem at each iteration. Once a satisfactory q_m is obtained, the linear part p_{m+k} is easily found.

The method is succinctly described in Algorithm 2.1. Different from the basic version presented in [6], it makes use of *two* linear spaces in $\mathbb{C}^N$, a search space $\mathcal{S}$ and a target space $\mathcal{T}$, both of which are *(rational) Krylov spaces*. Given a matrix $A \in \mathbb{C}^{N\times N}$, a (so-called) starting vector $\boldsymbol{b} \in \mathbb{C}^N$, an integer $m \geq 0$, and a nonzero polynomial $q_m \in \mathcal{P}_m$ with roots disjoint from the spectrum of A, we define the associated *rational Krylov space of order* m as

$$\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) := \{p_m(A)q_m(A)^{-1}\boldsymbol{b} \,:\, p_m \in \mathcal{P}_m\}.$$

The roots of q_m are called *poles* of the rational Krylov space and they are denoted by $\xi_1, \xi_2, \ldots, \xi_m$. For convenience, we sometimes refer to q_m itself as *poles* of the rational Krylov space. If $\deg(q_m) < m$, then $m - \deg(q_m)$ of the poles are set to ∞, and we refer to them as formal (multiple) roots of q_m. If all poles are set to ∞, we obtain the *(polynomial) Krylov space* $\mathcal{K}_{m+1}(A, \boldsymbol{b}) := \{p_m(A)\boldsymbol{b} \,:\, p_m \in \mathcal{P}_m\}$ as a special case of a rational Krylov space.

By $P_{\mathcal{T}}$ in line 5 of Algorithm 2.1 we denote the orthogonal projection onto $\mathcal{T}$. The essence of Algorithm 2.1 is the relocation of poles in line 7. Since with any polynomial $\widehat{q}_m \in \mathcal{P}_m$ we can associate a vector $\widehat{\boldsymbol{v}} = \widehat{q}_m(A)q_m(A)^{-1}\boldsymbol{b} \in \mathcal{S}$, and the other way around, we may identify $\widehat{q}_m$, the improvement of q_m, by looking for the corresponding vector $\widehat{\boldsymbol{v}} \in \mathcal{S}$. Theorem 2.2 below, a consequence of the following Lemma 2.1, provides insight into the RKFIT pole relocation, i.e., lines 5–7 of Algorithm 2.1.

LEMMA 2.1. *Let $q_m, q_m^\star \in \mathcal{P}_m$ be nonzero polynomials with roots disjoint from the spectrum of $A \in \mathbb{C}^{N\times N}$. Fix $-m \leq k \in \mathbb{Z}$, and let $\boldsymbol{b} \in \mathbb{C}^N$ be such that $2m + k < M(A, \boldsymbol{b})$. Assume that $F = p_{m+k}^\star(A)q_m^\star(A)^{-1}$ for some $p_{m+k}^\star \in \mathcal{P}_{m+k}$. Define $\mathcal{S}$ and $\mathcal{T}$ as in lines 3 and 4 of Algorithm 2.1, respectively, and let $\widehat{V}_{m+1}$ be an orthonormal basis of $\mathcal{S}$. Then the matrix $(I-P_{\mathcal{T}})F\widehat{V}_{m+1}$ has a nullspace of dimension $\Delta m+1$ if and only if Δm is the largest integer such that $p_{m+k}^\star/q_m^\star$ is of type $(m+k-\Delta m, m-\Delta m)$.*

Proof. Let $\widehat{\boldsymbol{v}} = \widehat{p}_m(A)q_m(A)^{-1}\boldsymbol{b} \in \mathcal{S}$, with $\widehat{p}_m \in \mathcal{P}_m$ being arbitrary. Then

$$F\widehat{\boldsymbol{v}} = p_{m+k}^\star(A)q_m^\star(A)^{-1}\widehat{p}_m(A)q_m(A)^{-1}\boldsymbol{b} =: p_{2m+k}(A)q_m^\star(A)^{-1}q_m(A)^{-1}\boldsymbol{b}$$

has a unique representation in terms of $p_{2m+k}/(q_m^\star q_m)$ since $2m+k < M$. Assume that $F\widehat{\boldsymbol{v}} \in \mathcal{T}$. In this case we also have the representation $F\widehat{\boldsymbol{v}} = \widetilde{p}_{m+k}(A)q_m(A)^{-1}\boldsymbol{b}$, with a uniquely determined $\widetilde{p}_{m+k} \in \mathcal{P}_{m+k}$. By the uniqueness of the rational representations we conclude that $p_{2m+k}/(q_m^\star q_m) = \widetilde{p}_{m+k}/q_m$, or equivalently, $q_m^\star$ divides $p_{2m+k} = p_{m+k}^\star \widehat{p}_m$. Hence, the poles of $p_{m+k-\Delta m}^\star/q_{m-\Delta m}^\star \equiv p_{m+k}^\star/q_m^\star$ must be roots of $\widehat{p}_m$. The other Δm roots of $\widehat{p}_m$ can be chosen freely, giving rise to the $(\Delta m+1)$-dimensional subspace

$$\mathcal{N} := \left\{ p_{\Delta m}(A)q_{m-\Delta m}^\star(A)q_m(A)^{-1}\boldsymbol{b} \,\middle|\, p_{\Delta m} \in \mathcal{P}_{\Delta m} \right\} \subseteq \mathcal{S}, \tag{2.1}$$

whose elements $\widehat{\boldsymbol{v}}$ are such that $F\widehat{\boldsymbol{v}} \in \mathcal{T}$. Hence, $\Delta m + 1$ is the dimension of the nullspace of $(I - P_{\mathcal{T}})F\widehat{V}_{m+1}$. ∎

THEOREM 2.2. *Let $q_m, q_m^\star, F, A, \boldsymbol{b}, m, k, \mathcal{S}$, and $\mathcal{T}$ be as in Theorem 2.1. Then $p_{m+k}^\star$ and $q_m^\star$ are coprime and either $\deg(p_{m+k}^\star) = m+k$ or $\deg(q_m^\star) = m$ if and only if $F\boldsymbol{v} \in \mathcal{T}$ is solved uniquely (up to scaling) by $\boldsymbol{v} \in \mathcal{S}$. This solution is given by $\boldsymbol{v}^\star = \gamma q_m^\star(A)q_m(A)^{-1}\boldsymbol{b}$ with some nonzero scaling factor $\gamma \in \mathbb{C}$.*

The theorem asserts that if $F\boldsymbol{b} = p_{m+k}(A)q_m^\star(A)^{-1}\boldsymbol{b}$ and $\Delta m = 0$, then the "roots" of $\boldsymbol{v}^\star = \gamma q_m^\star(A)q_m(A)^{-1}\boldsymbol{b}$ match the unknown poles $q_m^\star$ and the next approximate poles become $q_m := q_m^\star$. Hence RKFIT identifies the exact poles within one iteration independently of the starting guess q_m. If $\Delta m > 0$ the exact $m - \Delta m$ poles are also found, but additional Δm superfluous poles at arbitrary locations are present as well. In section 3 we develop a procedure for automatically reducing the denominator degree m by Δm and adapting k. Comments regarding the convergence of RKFIT when dealing with noisy data (and roundoff) or when $F\boldsymbol{b}$ cannot be exactly represented as $r(A)\boldsymbol{b}$ for a rational function r of type $(m+k, m)$ are included in section A.5 of the appendix.

In the remaining part of this section we discuss line-by-line how Algorithm 2.1 can be implemented using rational Krylov techniques. These considerations are also important for developments in the forthcoming sections.

- **Line 3:** An orthonormal basis $\widehat{V}_{m+1} \in \mathbb{C}^{N\times(m+1)}$ for the search space $\mathcal{S} = \mathcal{R}(\widehat{V}_{m+1})$ can be obtained with the rational Arnoldi algorithm which, given $A, \boldsymbol{b}$ and q_m, constructs a decomposition of the form

$$A\widehat{V}_{m+1}\underline{\widehat{K}_m} = \widehat{V}_{m+1}\underline{\widehat{H}_m}, \tag{2.2}$$

  where $(\underline{\widehat{H}_m}, \underline{\widehat{K}_m})$ is an $(m+1)\times m$ upper-Hessenberg pencil satisfying $|\widehat{h}_{j+1,j}| + |\widehat{k}_{j+1,j}| \neq 0$ for $j = 1, \ldots, m$ and with $\{\widehat{h}_{j+1,j}/\widehat{k}_{j+1,j}\}_{j=1}^m$ being the (formal) roots of q_m, i.e., the *poles* of the rational Krylov space $\mathcal{S}$. A decomposition of the form (2.2) is called a *rational Arnoldi decomposition (RAD)*. For details of the rational Arnoldi algorithm and properties of RADs we refer to [4, 6, 35, 37, 38].
- **Line 4:** Since $\mathcal{T} = \mathcal{Q}_{m+k+1}(A, \boldsymbol{b}, q_m)$, we can compute an orthonormal basis V_{m+k+1} for $\mathcal{T}$ using once again the rational Arnoldi algorithm. A computationally more economic alternative is to reuse (2.2). Indeed, if $k = 0$, we simply have $\mathcal{T} = \mathcal{S}$. Otherwise, $\mathcal{S}$ either has to be expanded (if $k > 0$) or compressed (if $k < 0$) to get $\mathcal{T}$:
  - In the case of superdiagonal approximants ($k > 0$), $\mathcal{T} = \mathcal{Q}_{m+k+1}(A, \boldsymbol{b}, q_m)$ is the rational Krylov space of dimension $m+k+1$ with m poles being the roots of q_m and additional k poles at infinity. In order to get an orthonormal basis for $\mathcal{Q}_{m+k+1}(A, \boldsymbol{b}, q_m)$, we expand (2.2) into $AV_{m+k+1}\underline{K_{m+k}} = V_{m+k+1}\underline{H_{m+k}}$ by performing k additional polynomial steps with the rational Krylov algorithm.

Let us, for convenience, label by $V_{m+k+1} := \widehat{V}_{m+k+1}$ the orthonormal basis for $\mathcal{T}$ when $k \geq 0$. Thus, $P_{\mathcal{T}} = V_{m+k+1}V_{m+k+1}^*$.

– In the subdiagonal case ($k < 0$), the target space is $\mathcal{T} = \mathcal{K}_{m+k+1}(A, q_m(A)^{-1}\boldsymbol{b})$. Note that $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) = \mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b})$. Therefore, we aim at transforming the RAD (2.2) for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ into an RAD

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m} \tag{2.3}$$

for $\mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b})$. An orthonormal basis for $\mathcal{T}$ is then given by truncating V_{m+1} to V_{m+k+1}, the first $m+k+1$ columns of V_{m+1}. Using a sequence of Givens rotations in a QZ fashion (as explained in [39, p. 495] or [4, Section 5.2]) we get unitary matrices Q_{m+1} and Z_m such that $\underline{K_m} = Q_{m+1}^*\underline{\widehat{K}_m}Z_m$ is upper-triangular and $\underline{H_m} = Q_{m+1}^*\underline{\widehat{H}_m}Z_m$ is upper-Hessenberg. Fittingly, the poles $h_{j+1,j}/k_{j+1,j}$ of (2.3) with $V_{m+1} = \widehat{V}_{m+1}Q_{m+1}$ are all at infinity. Hence $\mathcal{R}(V_{j+1}) = \mathcal{K}_{j+1}(A, q_m(A)^{-1}\boldsymbol{b})$ for $j = 0, 1, \ldots, m$, and we can set $P_{\mathcal{T}} = V_{m+k+1}V_{m+k+1}^*$.

- **Line 5:** Defining the matrix

$$S = F\widehat{V}_{m+1} - V_{m+k+1}\left(V_{m+k+1}^*F\widehat{V}_{m+1}\right) \in \mathbb{C}^{N\times(m+1)}, \tag{2.4}$$

a solution is given by $\widehat{\boldsymbol{v}} = \widehat{V}_{m+1}\widehat{\boldsymbol{c}}$, where $\widehat{\boldsymbol{c}}$ is a right singular vector of S corresponding to a smallest singular value $\sigma_{\min}$.

- **Lines 6–7:** What we need in line 3 as input for the rational Arnoldi algorithm are the poles of the rational Krylov space that is being constructed, that is, the roots of $\widehat{q}_m$. Let Q_{m+1} be unitary with first column $Q_{m+1}\boldsymbol{e}_1 = \widehat{\boldsymbol{c}}$, then the roots of $\widehat{q}_m$ are the eigenvalues of the $m \times m$ pencil

$$\left(\begin{bmatrix}\boldsymbol{0} & I_m\end{bmatrix} Q_{m+1}^*\underline{\widehat{H}_m}, \begin{bmatrix}\boldsymbol{0} & I_m\end{bmatrix} Q_{m+1}^*\underline{\widehat{K}_m}\right); \tag{2.5}$$

see [6, Section 5] for details.

- **Line 9:** The approximant r of type $(m+k, m)$ is computed by LS approximation of $F\boldsymbol{b}$ from the target rational Krylov space $\mathcal{T}$. More precisely, if V_{m+k+1} is an orthonormal basis for $\mathcal{T}$, then the approximant r is represented by a coefficient vector $\boldsymbol{c} \in \mathbb{C}^{m+k+1}$ such that $r(A)\boldsymbol{b} = \|\boldsymbol{b}\|_2 V_{m+k+1}\boldsymbol{c}$. The coefficient vector is given by

$$\boldsymbol{c} = V_{m+k+1}^*\big(F\boldsymbol{b}\big)/\|\boldsymbol{b}\|_2. \tag{2.6}$$

Computing the coefficient vector $\boldsymbol{c}$ at each iteration does not significantly increase the computational complexity because $F\boldsymbol{b}$ needs to be computed only once. The availability of $\boldsymbol{c}$ also enables the cheap evaluation of the relative misfit (1.5), which allows to stop the RKFIT iteration when a desired tolerance $\varepsilon_{\texttt{tol}}$ is achieved.

3. Tuning degree parameters m and k. In some applications, one may want to construct a rational function of sufficiently small misfit without knowing the required degree parameters m and k in advance. In such situations one can try to fit the data with high enough (for instance maximal one is willing to use) degree parameters and then, after RKFIT has found a sufficiently good approximant, reduce m and k without deteriorating much the approximation accuracy. In this section we present a strategy for performing this reduction.

We assume to have at hand an $(m+k,m)$ approximant r such that $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \leq \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$. We then propose the following three-step procedure. (1) Reduce m to $m-\Delta m \geq 0$, with Δm such that $m-\Delta m+k \geq 0$. (2) Find a lower-degree approximant of type $(m-\Delta m+k, m-\Delta m)$. (3) Reduce k if required. These steps are discussed in the following three subsections for the case that F is a rational matrix function, while in subsection 3.4 we provide a numerical illustration. In subsection 3.5 we discuss the case when F is not a rational matrix function. This is followed by another numerical illustration in subsection 3.6.

3.1. Reducing the denominator degree m. Assume that F is a rational matrix function. Our reduction procedure for m is based on Lemma 2.1, which asserts that a defect $\Delta m+1$ of the matrix $S = (I-P_{\mathcal{T}})F\widehat{V}_{m+1}$ corresponds to F being of type $(m-\Delta m+k, m-\Delta m)$. Due to numerical roundoff, the numerical rank of S related to a given tolerance $\|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$ (with, e.g., $\varepsilon_{\mathtt{tol}} = 10^{-15}$) is computed. More precisely, we reduce m by the largest integer $\Delta m \leq \min\{m, m+k\}$ such that

$$\sigma_{m+1-\Delta m} \leq \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}, \tag{3.1}$$

where $\sigma_1 \geq \ldots \geq \sigma_{m+1}$ are the singular values of S.

3.2. Finding a lower-degree approximant. If $\Delta m \geq 1$, then m needs to be reduced and a new approximant of lower numerator and denominator degree is required. As seen in the proof of Lemma 2.1, the $\Delta m+1$ linearly independent vectors spanning $\mathcal{N}$ all share as the greatest common divisor (GCD) the polynomial $q^{\star}_{m-\Delta m}$, and its roots should be used as poles of the reduced-degree rational approximant. The following theorem shows how these roots can be obtained from the pencil $(\underline{\widehat{H}_m}, \underline{\widehat{K}_m})$ in (2.2).

THEOREM 3.1. *Let (2.2) be an RAD for $\mathcal{Q}_{m+1}(A,\boldsymbol{b},q_m)$ with $m+1 < M(A,\boldsymbol{b})$, and let the $r_j \equiv \widehat{V}_{m+1}\boldsymbol{c}_j$ for $j=1,\ldots,\Delta m+1$ be linearly independent. Assume that the numerators of r_j share as GCD a polynomial of degree $m-\Delta m$. Let $X \in \mathbb{C}^{(m+1)\times(m+1)}$ be a nonsingular matrix with $X\boldsymbol{e}_j = \boldsymbol{c}_j$ for $j=1,\ldots,\Delta m+1$. Introduce*

$$\widehat{K}_\star = \begin{bmatrix} O & I_{m-\Delta m}\end{bmatrix} X^{-1}\underline{\widehat{K}_m}\begin{bmatrix} O \\ I_{m-\Delta m}\end{bmatrix}, \quad \widehat{H}_\star = \begin{bmatrix} O & I_{m-\Delta m}\end{bmatrix} X^{-1}\underline{\widehat{H}_m}\begin{bmatrix} O \\ I_{m-\Delta m}\end{bmatrix}.$$

Assume further that $\widehat{K}_\star$ is nonsingular. Then the roots of the GCD are the eigenvalues of the $(m-\Delta m)\times(m-\Delta m)$ generalized eigenproblem $(\widehat{H}_\star, \widehat{K}_\star)$.

Proof. We transform the RAD (2.2) into $AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}$, where $V_{m+1} = \widehat{V}_{m+1}X$, $\underline{K_m} = X^{-1}\underline{\widehat{K}_m}Y$, and $\underline{H_m} = X^{-1}\underline{\widehat{H}_m}Y$, and with $Y = \mathrm{blkdiag}(I_{\Delta m}, K_\star)^{-1}$. Written in scalar form, we hence have for all $z \in \mathbb{C}$ the relation

$$z\boldsymbol{r}(z)\underline{K_m} = \boldsymbol{r}(z)\underline{H_m} \iff \boldsymbol{r}(z)\big(z\underline{K_m} - \underline{H_m}\big) = \boldsymbol{0}^T,$$

where $\boldsymbol{r}(z) = [r_1(z) \quad \ldots \quad r_{\Delta m+1}(z) \quad r_{\Delta m+2}(z) \quad \ldots \quad r_{m+1}(z)]$. Introduce $K_\star$ and $H_\star$ as the bottom-right $(m-\Delta m)\times(m-\Delta m)$ submatrices of $\underline{K_m}$ and $\underline{H_m}$, respectively. Since $\Lambda(\widehat{H}_\star, \widehat{K}_\star) = \Lambda(H_\star, K_\star)$, we only need to show that $\Lambda(H_\star, K_\star)$ are the roots of the GCD.

Let λ be a common root of $\{r_j\}_{j=1}^{\Delta m+1}$. Then the last $m-\Delta m$ columns of $\boldsymbol{r}(\lambda)(\lambda\underline{K_m} - \underline{H_m}) = \boldsymbol{0}^T$ assert that λ is a generalized eigenvalue of $(H_\star, K_\star)$ with left eigenvector $\boldsymbol{r}_\star(\lambda)^* = [r_{\Delta m+2}(\lambda) \quad \ldots \quad r_{m+1}(\lambda)]^* \neq \boldsymbol{0}$. This handles simple roots.

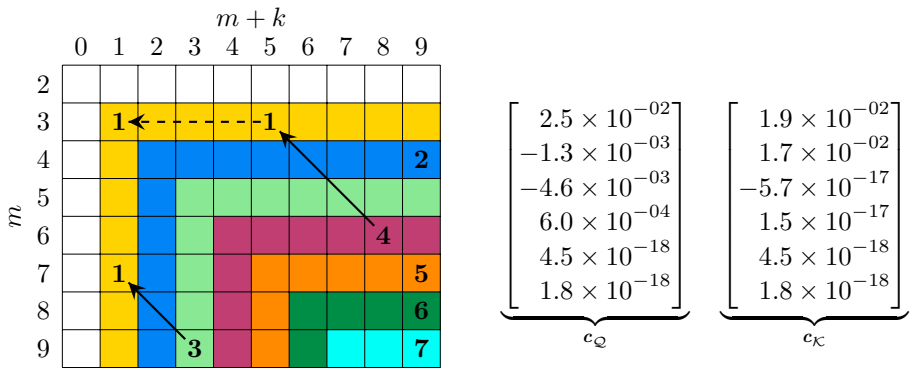


Fig. 3.1: Degree reduction when fitting a rational matrix function; see section 3.4.

Let us now assume that λ is a root of multiplicity 2. Note that $K_\star = I_{m-\Delta m}$. Differentiating the scalar RAD with respect to λ gives

$$\boldsymbol{r}'(\lambda)\big(\lambda\underline{K_m} - \underline{H_m}\big) + \boldsymbol{r}(\lambda)\underline{K_m} = \mathbf{0}^T \iff \boldsymbol{r}'(\lambda)\big(\lambda\underline{K_m} - \underline{H_m}\big) = -\boldsymbol{r}(\lambda)\underline{K_m}.$$

The last $m - \Delta m$ columns in the latter relation give

$$\boldsymbol{r}_\star'(\lambda)\big(\lambda K_\star - H_\star\big) = -\boldsymbol{r}_\star(\lambda)K_\star = -\boldsymbol{r}_\star(\lambda) \neq \mathbf{0}^T.$$

In particular $\boldsymbol{r}_\star'(\lambda) \neq \mathbf{0}^T$. Further $\boldsymbol{r}_\star'(\lambda)\big(\lambda K_\star - H_\star\big)^2 = -\boldsymbol{r}_\star(\lambda)\big(\lambda K_\star - H_\star\big) = \mathbf{0}^T$. Hence $\boldsymbol{r}_\star'(\lambda)$ is a generalized eigenvector for the eigenvalue λ of $(H_\star, K_\star)$, which is hence of multiplicity two or greater. The proof for roots of higher multiplicity follows the same argument. ∎

REMARK 3.2. *The assumption that $K_\star$ is nonsingular is used in the proof of Theorem 3.1 for the case of repeated roots only. We conjecture that this assumption can be removed also when there are multiple roots, and that it follows from the fact that the numerators of $\{r_j\}_{j=1}^{\Delta m+1}$ have as GCD a polynomial of degree $m - \Delta m$.*

3.3. Numerator degree revealing basis. We now assume that the denominator degree $m := m - \Delta m$ has already been reduced and a new approximant r of type $(m+k, m)$ such that $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \leq \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$ has been found. Reducing the numerator degree is a linear problem and we can guarantee the misfit to stay below $\varepsilon_{\mathtt{tol}}$ after the reduction.

Let $\mathcal{T} = \mathcal{K}_{m+k+1}(A, q_m(A)^{-1}\boldsymbol{b})$ be the final target space such that $r(A)\boldsymbol{b} \in \mathcal{T}$, and let V_j be an orthonormal basis for $\mathcal{K}_j(A, q_m(A)^{-1}\boldsymbol{b})$ for $j = 1, \ldots, m+k+1$. As the vectors in V_j have ascending numerator degree, this basis reveals the degree of $r(A)\boldsymbol{b}$ by looking at the trailing expansion coefficients $\boldsymbol{c} \in \mathbb{C}^{m+k+1}$ satisfying $r(A)\boldsymbol{b}/\|\boldsymbol{b}\|_2 = V_{m+k+1}\boldsymbol{c}$.

Introduce $\boldsymbol{c}_{-i} = [O \quad I_i]\boldsymbol{c} \in \mathbb{C}^i$ for $i = 1, \ldots, m+k$. By the triangle inequality,

$$\left\| F\boldsymbol{b}/\|\boldsymbol{b}\|_2 - V_{m+k+1}\boldsymbol{c} + V_{m+k+1}\begin{bmatrix}\mathbf{0}\\ \boldsymbol{c}_{-i}\end{bmatrix}\right\|_2 \leq \left\| F\boldsymbol{b}/\|\boldsymbol{b}\|_2 - V_{m+k+1}\boldsymbol{c}\right\|_2 + \left\|\begin{bmatrix}\mathbf{0}\\ \boldsymbol{c}_{-i}\end{bmatrix}\right\|_2.$$

The degree of the numerator of r can therefore be reduced to $m + k - \Delta k$, where Δk is the maximal integer $1 \leq i \leq m + k$ such that

$$\|\boldsymbol{c}_{-i}\|_2 \leq \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}} - \|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2, \tag{3.2}$$

or $\Delta k = 0$ if such an integer i does not exist. The last Δk components of $\boldsymbol{c}$ may hence be truncated, giving $\boldsymbol{c}_\Delta \in \mathbb{C}^{m+k-\Delta k+1}$ such that $r_\Delta \equiv V_{m+k-\Delta k+1}\boldsymbol{c}_\Delta$ still satisfies $\|F\boldsymbol{b} - r_\Delta(A)\boldsymbol{b}\|_2 \le \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$.

3.4. Example: Degree reduction for a rational matrix function. In Figure 3.1 we report some results for the degree reduction procedure when fitting $F\boldsymbol{b}$, where $F = A(A+I)^{-1}(A+3I)^{-2}$, $A = \mathrm{tridiag}(-1, 2, -1) \in \mathbb{R}^{N\times N}$, and $\boldsymbol{b} = \boldsymbol{e}_1 \in \mathbb{R}^N$, with $N = 150$. Note that F is of type $(1, 3)$. The initial poles of the search space are all at infinity.

The table on the left shows the number $\Delta m+1$ of singular values of $(I-P_{\mathcal{T}})F\widehat{V}_{m+1}$ below the tolerance $\|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}} = 10^{-15}$, for different choices of m and k. For the choice $(m + k, m) = (3, 9)$, for instance, we obtain $\Delta m = 2$, and hence the reduced type is $(1, 7)$. In this case m is not fully reduced because k was chosen too small. For the choice $(m + k, m) = (8, 6)$ we obtain $\Delta m = 3$, giving the reduced type $(5, 3)$. The roots of the GCD are -1 and $-3 \pm \mathrm{i}2.32 \times 10^{-7}$. With these three finite poles and another two poles at infinity, the type $(5, 3)$ approximant r produces a relative misfit 7.02×10^{-17}. The expansion coefficients $\boldsymbol{c}_{\mathcal{Q}}$ of r in the orthonormal rational basis are listed to the right of the table. They indicate that the last two poles at infinity are actually superfluous, and r is of type at most $(3, 3)$. Only the expansion of r in the orthonormal polynomial basis, as explained in subsection 3.3, reveals that r is of type $(1, 3)$. The coefficients $\boldsymbol{c}_{\mathcal{K}}$ in this polynomial basis are also given.

3.5. General F. The following lemma extends Lemma 2.1 to the case when F is not necessarily a rational matrix function.

LEMMA 3.3. *Let $q_m, A, \boldsymbol{b}, m, k, \mathcal{S}, \mathcal{T}$, and $\widehat{V}_{m+1}$ be as in Lemma 2.1. Assume that for $F \in \mathbb{C}^{N\times N}$ we have found a rational approximant $r = p_{m+k}/q_m$ of type $(m+k, m)$ such that $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \le \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$. If the matrix $(I - P_{\mathcal{T}})F\widehat{V}_{m+1}$ has $\Delta m + 1$ singular values smaller than $\|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$, then there exists a $(\Delta m + 1)$-dimensional subspace $\mathcal{N}_g \subseteq \mathcal{S}$, containing $\boldsymbol{b}$, such that*

$$\min_{p\in\mathcal{P}_{m+k}} \left\|F\widehat{\boldsymbol{v}} - p(A)q_m(A)^{-1}\boldsymbol{b}\right\|_2 \le \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$$

for all $\widehat{\boldsymbol{v}} \in \mathcal{N}_g$, $\|\boldsymbol{v}\|_2 = 1$.

Proof. Consider a thin SVD of the matrix $(I - P_{\mathcal{T}})F\widehat{V}_{m+1} = U\Sigma W^*$, where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{m+1}) \in \mathbb{R}^{(m+1)\times(m+1)}$ and $\sigma_{m+1} \le \cdots \le \sigma_{m-\Delta m} \le \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$ by assumption. Equivalently, $(I - P_{\mathcal{T}})F\widehat{V}_{m+1}W = U\Sigma$. Then the final $\Delta m + 1$ columns of $\widehat{V}_{m+1}W$ form a basis for $\mathcal{N}_g$. It follows from the assumption $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \le \|F\boldsymbol{b}\|_2\varepsilon_{\mathtt{tol}}$ that $\boldsymbol{b} \in \mathcal{N}_g$. ∎

Recall that if F is a rational matrix function, then the space $\mathcal{N}_g$ defined in Lemma 3.3 corresponds to the exact nullspace $\mathcal{N} = \mathcal{K}_{\Delta m+1}(A, q^\star_{m-\Delta m}(A)q_m(A)^{-1}\boldsymbol{b})$ defined in (2.1), where the (numerators of the) rational functions share as GCD the polynomial $q^\star_{m-\Delta m}$. In the general case $\mathcal{N}_g$ is only a subspace of the larger rational Krylov space $\mathcal{S}$, and the rational functions present in $\mathcal{N}_g$ do not necessarily share a common denominator. However, for every $\widehat{\boldsymbol{v}} = \widehat{p}_m(A)q_m(A)^{-1}\boldsymbol{b} \in \mathcal{N}_g$ the vector $F\widehat{p}_m(A)q_m(A)^{-1}\boldsymbol{b}$ is well approximated in the 2-norm by some vector $p(A)q_m(A)^{-1}\boldsymbol{b}$, with $p \in \mathcal{P}_{m+k}$. This suggests that the polynomials $\widehat{p}_m$ corresponding to vectors $\widehat{\boldsymbol{v}} \in \mathcal{N}_g$ share an *approximate* GCD (see, e.g., [8]) whose roots approximate the poles of a "good" rational approximation $r(A)\boldsymbol{b}$ for $F\boldsymbol{b}$. We therefore propose to use the same reduction procedure as suggested by Theorem 3.1.

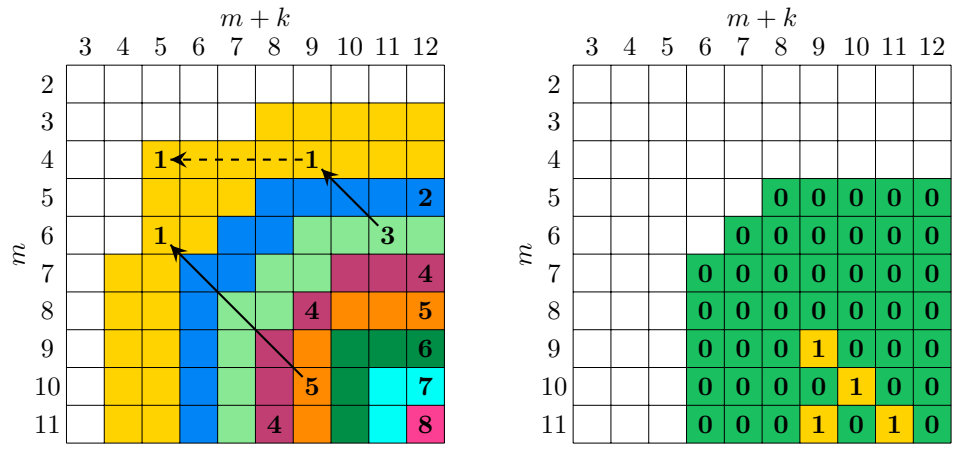


Fig. 3.2: Degree reduction when fitting a non-rational matrix function; see section 3.6.

As there is no guarantee that after reduction RKFIT will be able to find an approximant of relative misfit below $\varepsilon_{\texttt{tol}}$, the use of a safety parameter $\varepsilon_{\texttt{safe}}$ is recommended. More precisely, we reduce m by the largest integer $\Delta m \leq \min\{m, m+k\}$ such that

$$\sigma_{m+1-\Delta m} \leq \|F\boldsymbol{b}\|_2 \varepsilon_{\texttt{tol}} \varepsilon_{\texttt{safe}}, \tag{3.3}$$

where $\sigma_1 \geq \ldots \geq \sigma_{m+1}$ are the singular values of S. By default we use $\varepsilon_{\texttt{safe}} = 0.1$.

3.6. Example: Degree reduction for a non-rational matrix function. Figure 3.2 illustrates our reduction strategy for the function $F = \sqrt{A + A^2}$, where $A = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N\times N}$ and $N = 150$. The vector $\boldsymbol{b}$ is chosen as $\boldsymbol{b} = \boldsymbol{e}_1$. The poles of the search space are obtained after three RKFIT iterations with all initial poles at infinity.

The table on the left shows the number $\Delta m+1$ of singular values of $(I-P_{\mathcal{T}})F\widehat{V}_{m+1}$ below $\|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}}\varepsilon_{\texttt{safe}} = 10^{-5}$ for different choices of m and k. For the choice $(m+k, m) = (9, 10)$ we obtain $\Delta m = 4$, implying the reduced type $(5, 6)$. The choice $(m+k, m) = (11, 6)$ is reduced down to $(9, 4)$ as $\Delta m = 2$. Representing this new approximant in the numerator degree-revealing basis allows for a further reduction to type $(5, 4)$. The table on the right visualizes how many RKFIT iterations are required after reduction to reobtain an approximant of misfit below $\varepsilon_{\texttt{tol}} = 10^{-4}$, using the approximate GCD strategy for selecting the poles to restart RKFIT with. Note that in most instances the misfit remains acceptable after the reduction, while in the other only one RKFIT iteration is needed to obtain an acceptable approximation. This shows the benefit of the developed reduction strategy. (Another example is given in section 6.1.)

4. Extensions. In this section we discuss extensions of RKFIT for solving problems of the more general form (1.5).

4.1. Family of rational functions. In order to tackle the more general problem (1.5) of finding a family $\{r^{[j]}\}_{j=1}^{\ell}$ of rational functions with a common denominator we only need to modify line 5 in Algorithm 2.1 to

5. Find $\widehat{\boldsymbol{v}} = \operatorname{argmin}_{\substack{\boldsymbol{v}\in\mathcal{S}\\ \|\boldsymbol{v}\|_2=1}} \sum_{j=1}^{\ell} \|D^{[j]}\,(I - P_{\mathcal{T}})\,F^{[j]}\boldsymbol{v}\|_2^2$.

Once again, a solution is $\widehat{\boldsymbol{v}} = \widehat{V}_{m+1}\widehat{\boldsymbol{c}}$, where $\widehat{\boldsymbol{c}}$ is a right singular vector corresponding to a smallest singular value of the matrix

$$S = [S_1^T \quad S_2^T \quad \dots \quad S_\ell^T]^T \in \mathbb{C}^{N\ell,m+1}, \qquad \text{where} \tag{4.1}$$

$$S_j = D^{[j]}\left[F^{[j]}\widehat{V}_{m+1} - V_{m+k+1}\left(V_{m+k+1}^* F^{[j]}\widehat{V}_{m+1}\right)\right] \in \mathbb{C}^{N,m+1}. \tag{4.2}$$

The ℓ rational approximants $\{r^{[j]}\}_{j=1}^{\ell}$ may be represented by the coefficient vectors

$$\boldsymbol{c}^{[j]} = \left(D^{[j]}V_{m+k+1}\right)^{\dagger}\left(D^{[j]}F^{[j]}\boldsymbol{b}\right)/\|\boldsymbol{b}\|_2, \tag{4.3}$$

which reduces to $\boldsymbol{c}^{[j]} = V_{m+k+1}^*\left(F^{[j]}\boldsymbol{b}\right)/\|\boldsymbol{b}\|_2$ if $D^{[j]} = I_N$. The remaining parts of RKFIT, with the exception of the degree reducing strategy, are unaffected. In order to make sure that all of $\{r^{[j]}\}_{j=1}^{\ell}$ share the same denominator, the reduction of m should be based on the singular values of S, and not the individual S_j. The numerator reduction can be performed for each $r^{[j]}$ individually.

4.2. Block case. Let us consider the case $B = [\boldsymbol{b}_1 \quad \dots \quad \boldsymbol{b}_n] \in \mathbb{C}^{N\times n}$ with $n > 1$. Introduce the $Nn \times Nn$ matrices

$$\boldsymbol{D}^{[j]} = I_n \otimes D^{[j]}, \quad \boldsymbol{F}^{[j]} = I_n \otimes F^{[j]}, \quad \text{and} \quad \boldsymbol{A} = I_n \otimes A, \tag{4.4}$$

where $I_n \otimes X = \operatorname{blkdiag}(X, \dots, X)$. Since

$$\|D^{[j]}[F^{[j]}B - r^{[j]}(A)B]\|_F^2 = \|\boldsymbol{D}^{[j]}[\boldsymbol{F}^{[j]}\operatorname{vec}(B) - r^{[j]}(\boldsymbol{A})\operatorname{vec}(B)]\|_2^2$$

we recover the single-column case $n = 1$ considered so far, with $\boldsymbol{b} = \operatorname{vec}(B)$.

Our implementation [5] supports the case $n > 1$, and takes advantage of the structure present in (4.4) so that only $\{D^{[j]}, F^{[j]}\}_{j=1}^{\ell}$ and A are stored, while $\boldsymbol{D}^{[j]}$, $\boldsymbol{F}^{[j]}$, and $\boldsymbol{A}$ are never constructed explicitly. In fact $D^{[j]}, F^{[j]}$, and A are not explicitly needed either, as all that is required is the ability to compute $D^{[j]}\boldsymbol{x}, F^{[j]}\boldsymbol{x}, A\boldsymbol{x}$ for arbitrary $\boldsymbol{x} \in \mathbb{C}^N$, as well as the ability to solve shifted linear systems $(A - \xi I)\boldsymbol{x} = \boldsymbol{v}$.

4.3. Avoiding complex arithmetic. If $\{D^{[j]}, F^{[j]}\}_{j=1}^{\ell}$, A, and B are real-valued and the set of starting poles $\{\xi_j\}_{j=1}^{m}$ is closed under complex conjugation, we can use the "real version" of the rational Arnoldi algorithm and avoid complex arithmetic; see [36]. The matrix S in (4.1) is guaranteed to be real-valued and the generalized eigenproblem (2.5) is real-valued as well. This guarantees the relocated poles to appear in complex-conjugate pairs. For more details we refer to [4, Section 6.1.4].

5. Working with rational functions. After the RKFIT algorithm has terminated, a rational function r of type $(m + k, m)$ is represented by the pencil $(\underline{H_d}, \underline{K_d})$, satisfying $AV_{d+1}\underline{K_d} = V_{d+1}\underline{H_d}$ with $d := \max\{m, m + k\}$, and with the coefficients $\boldsymbol{c} = V_{d+1}^* F\boldsymbol{b}/\|\boldsymbol{b}\|_2$. We now show how to perform computations with such an RKFUN representation $r \equiv (\underline{H_d}, \underline{K_d}, \boldsymbol{c})$.

5.1. Evaluation. We consider the evaluation $r(\widehat{A})\widehat{\boldsymbol{b}}$ where $\widehat{A} \in \mathbb{C}^{\widehat{N}\times\widehat{N}}$ and $\widehat{\boldsymbol{b}} \in \mathbb{C}^{\widehat{N}}$. For this we require $\Lambda(\widehat{A})$ not to contain any of the poles $\xi_1, \dots, \xi_m$ of r. Note that $\widehat{A}$ and $\widehat{\boldsymbol{b}}$ may be different from A and $\boldsymbol{b}$ used to obtain r. Indeed, they may be of

different dimensions as well. For example, if $\widehat{N} = 1$ and $\widehat{\boldsymbol{b}} = 1$, we retrieve the scalar evaluation $r(z)$. Derivatives of r can be evaluated by using a Jordan block for $\widehat{A}$. For example, if $\widehat{A} = \left[\begin{smallmatrix} \lambda & 1 \\ 0 & \lambda \end{smallmatrix}\right]$ and $\widehat{\boldsymbol{b}} = [0 \quad 1]^T$, then $r(\widehat{A})\widehat{\boldsymbol{b}} = [r'(\lambda) \quad r(\lambda)]^T$.

The pencil $(\underline{H_d}, \underline{K_d})$ encodes recurrence relations for orthogonal rational functions $r_1, r_2, \ldots, r_{d+1}$ such that $r_j(A)\boldsymbol{b}/\|\boldsymbol{b}\|_2 = \boldsymbol{v}_j$, the jth column of V_{d+1}; see [6]. In this notation, we have $r = \sum_{j=1}^{d+1} c_j r_j$, where $\boldsymbol{c} = [c_1 \quad c_2 \quad \ldots \quad c_{d+1}]^T$. This suggests a two-step procedure for computing $r(\widehat{A})\widehat{\boldsymbol{b}}$. First, we construct $W_{d+1} \in \mathbb{C}^{N\times(d+1)}$ so that $r_j(\widehat{A})\widehat{\boldsymbol{b}} = W_{d+1}\boldsymbol{e}_j$, and second, we form $r(\widehat{A})\widehat{\boldsymbol{b}} = W_{d+1}\boldsymbol{c}$.

Let us elaborate on the first part. We need to form an RAD-like decomposition

$$\widehat{A}W_{d+1}\underline{K_d} = W_{d+1}\underline{H_d} \tag{5.1}$$

by rerunning the rational Arnoldi algorithm with the starting vector $W_{d+1}\boldsymbol{e}_1 = \widehat{\boldsymbol{b}}$. Note that (5.1) is equivalent to

$$(\rho\widehat{A} - \eta I)W_{d+1}(\nu\underline{H_d} - \mu\underline{K_d}) = (\nu\widehat{A} - \mu I)W_{d+1}(\rho\underline{H_d} - \eta\underline{K_d}),$$

for any scalars $\mu, \nu, \rho, \eta \in \mathbb{C}$ such that $\mu\rho \neq \nu\eta$. By taking $\mu/\nu \equiv h_{j+1,j}/k_{j+1,j}$ we can compute

$$W_{d+1}\boldsymbol{e}_{j+1} \equiv \boldsymbol{w}_{j+1} = \gamma_j^{-1}\big[(\nu\widehat{A} - \mu I)^{-1}(\rho\widehat{A} - \eta I)W_j(\nu\boldsymbol{h}_j - \mu\boldsymbol{k}_j) - W_j(\rho\boldsymbol{h}_j - \eta\boldsymbol{k}_j)\big],$$

where $\gamma_j = \rho h_{j+1,j} - \eta k_{j+1,j}$ for $j = 1, 2, \ldots, d$.

We have overloaded the `feval` function in MATLAB for RKFUN objects to implement this evaluation procedure. The function can be invoked by typing either `feval(r, A, b)` or `r(A, b)`.

5.2. Root-finding. For finding the roots of r we recall that $r(A)\boldsymbol{b}/\|\boldsymbol{b}\|_2 = V_{d+1}\boldsymbol{c} = p_d(A)q_m(A)^{-1}\boldsymbol{b}$. Let us assume that $\boldsymbol{c} \neq \boldsymbol{e}_1$, otherwise $r(A)\boldsymbol{b} = c_1\boldsymbol{b}$, i.e., r has no roots. Define $P = I_{m+1} - 2\boldsymbol{u}\boldsymbol{u}^*$, where $\boldsymbol{u} = (\gamma\boldsymbol{c} - \boldsymbol{e}_1)/\|\gamma\boldsymbol{c} - \boldsymbol{e}_1\|_2$ and $\gamma \in \mathbb{C}$ is a unimodular scalar such that $\gamma\boldsymbol{e}_1^*\boldsymbol{c}$ is real and nonnegative. It follows from [6, Theorem 4.4] that the roots of p_d are the eigenvalues of the $d \times d$ pencil

$$\left(\begin{bmatrix}\boldsymbol{0} & I_d\end{bmatrix} P\underline{H_d}, \begin{bmatrix}\boldsymbol{0} & I_d\end{bmatrix} P\underline{K_d}\right).$$

If $k < 0$, then among the d eigenvalues there are $-k$ infinite eigenvalues, or numerically, eigenvalues of large modulus. In our implementation `roots` of the RKToolbox [5] we hence sort the roots by their magnitudes and return only the $m + k$ smallest ones.

5.3. Conversion to partial fraction form. Here we only consider the case $k \leq 0$, i.e., $d = m$, and pairwise distinct finite poles $\xi_1, \ldots, \xi_m$; generalizations are discussed in section 7. The conversion of a type $(m + k, m)$ rational function r into partial fraction form can be achieved by transforming the rational Arnoldi decomposition $AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}$ in such a way that it reveals the residues. We aim to transform the latter RAD into

$$AW_{m+1}\begin{bmatrix} 0 & & & \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} = W_{m+1}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_1 & & & \\ & \xi_2 & & \\ & & \ddots & \\ & & & \xi_m \end{bmatrix}, \tag{5.2}$$

Algorithm 5.2 Conversion to partial fraction form. RKToolbox [5]: `residue`

Input: Upper-Hessenberg pencil $(\underline{H_m}, \underline{K_m})$ with finite distinct poles.
Output: Invertible matrices L_{m+1} and R_m representing the conversion.

1. Set $R_m = ([\mathbf{0} \quad I_m]\underline{K_m})^{-1}$, $\underline{H_m} := \underline{H_m}R_m$, and $\underline{K_m} := \underline{K_m}R_m$.
2. Set $L_{m+1} = \mathrm{blkdiag}(1, Q_m^{-1})$, where $[\mathbf{0} \quad I_m]\underline{H_m}Q_m = Q_m\mathrm{diag}(\xi_1, \ldots, \xi_m)$.
3. Update $R_m := R_mQ_m$, $\underline{H_m} := L_{m+1}\underline{H_m}Q_m$, and $\underline{K_m} := L_{m+1}\underline{K_m}Q_m$.
4. Introduce $D_{m+1} = [-\boldsymbol{e}_1 \quad \underline{K_m}]$.
5. Update $L_{m+1} := D_{m+1}L_{m+1}$, $\underline{H_m} := D_{m+1}\underline{H_m}$, and $\underline{K_m} := D_{m+1}\underline{K_m}$.
6. Update $R_m := R_mD_m$, $\underline{H_m} := \underline{H_m}D_m$, $\underline{K_m} := \underline{K_m}D_m$, where $D_m = \mathrm{diag}(1/h_{1j})$.
7. Redefine $D_m := \mathrm{diag}(1/\underline{k_{j+1,j}})$, and $D_{m+1} := \mathrm{blkdiag}(1, D_m)$.
8. Update $L_{m+1} := D_{m+1}L_{m+1}$, $\underline{H_m} := D_{m+1}\underline{H_m}$, and $\underline{K_m} := D_{m+1}\underline{K_m}$.

where $W_{m+1}\boldsymbol{e}_1 = \boldsymbol{v}_1$. One then easily verifies that the columns of W_{m+1} satisfy $\boldsymbol{w}_{j+1} = (A - \xi_j)^{-1}\boldsymbol{v}_1$. This conversion is achieved via left- and right-multiplication of the pencil $(\underline{H_m}, \underline{K_m})$ by invertible matrices given in Algorithm 5.2.

The algorithm consists of four parts. The first corresponds to lines 1–3, and it transforms the pencil so that the lower $m \times m$ part matches that of (5.2). The matrix $[\mathbf{0} \quad I_m]\underline{K_m}$ is invertible since it is upper-triangular with no zero elements on the diagonal (there are no infinite poles), and hence R_m is well defined in line 1. The second part corresponds to lines 4–5, and it zeroes the first row in $\underline{K_m}$. The third part, line 6, takes care of the first row in $\underline{H_m}$, setting all its elements to one. After this transformation, as the fourth part, we rescale $[\mathbf{0} \quad I_m]\underline{K_m}$ in lines 7–8, to recover I_m.

The process corresponds to transforming the original $\underline{H_m}$ and $\underline{K_m}$ as $\underline{H_m} := L_{m+1}\underline{H_m}R_m$ and $\underline{K_m} := L_{m+1}\underline{K_m}R_m$, and the rational Krylov basis V_{m+1} is transformed accordingly as $W_{m+1} = V_{m+1}L_{m+1}^{-1}$. Given a coefficient representation $r(A)\boldsymbol{b} = \|\boldsymbol{b}\|_2 V_{m+1}\boldsymbol{c}_{m+1}$ in the basis V_{m+1}, we arrive at the partial fraction expansion

$$r(A)\boldsymbol{b} = \|\boldsymbol{b}\|W_{m+1}\boldsymbol{d}_{m+1} = d_0\boldsymbol{b} + \sum_{j=1}^{m} d_j(A - \xi_j I)^{-1}\boldsymbol{b},$$

with residues $\boldsymbol{d}_{m+1} = L_{m+1}\boldsymbol{c}_{m+1} = [d_0 \quad d_1 \quad \ldots \quad d_m]^T$.

The transformation of V_{m+1} into the partial fraction basis W_{m+1} has condition number $\mathrm{cond}(L_{m+1})$, which can be arbitrarily bad in particular if some of the poles ξ_j are close together. Our implementation `residue` in the RKToolbox [5] therefore supports the use of MATLAB's variable precision arithmetic as well as the use of the Advanpix Multiprecision Toolbox [1].

6. Numerical experiments. In the following we demonstrate RKFIT with numerical experiments. MATLAB files for reproducing these experiments are part of the RKToolbox [5], among other examples (including those in [6]). Additionally, an RKFIT-based method for computing perfectly matched layers for Helmholtz problems on nonhomogeneous media has been developed and tested in [17].

6.1. MIMO dynamical system. We consider a model for the transfer function of the multiple-input/multiple-output (MIMO) system ISS 1R taken from [11]. There are 3 input and 3 output channels, giving $\ell = 9$ functions to be fitted. We use $N = 2 \times 561$ sampling points λ_j given in [11], appearing in complex-conjugate pairs

on the range $\pm\mathrm{i}[10^{-2}, 10^{3}]$. The data are closed under complex conjugation, and hence we can work with block-diagonal real-valued matrices A and $\{F^{[j]}\}_{j=1}^{\ell}$ as explained in section 4.3. The magnitudes of the $\ell = 9$ transfer functions to be fitted are plotted in Figure 6.1(a).

For the first experiment we try to find rational functions of type $(70, 70)$, and then reducing their degrees. A tolerance of $\varepsilon_{\mathtt{tol}} = 10^{-3}$ is used. In Figure 6.1(b) two convergence curves are shown, one for RKFIT as described in the previous sections (solid line), and the other for an RKFIT variant that enforces the poles to be stable (dashed line). A pole $\xi \in \mathbb{C}$ is stable if its real part is nonpositive, $\Re(\xi) \leq 0$, and this is enforced in the pole relocation step by simply flipping the real parts of the poles if necessary. At convergence the poles happen to be stable in both cases. The initial poles were all placed at infinity and the misfit at iteration 0 corresponds to these initial poles. Both RKFIT variants achieve a misfit below $\varepsilon_{\mathtt{tol}}$ at iteration 4, after which the degree reduction discussed in section 3 takes place. The denominator degree $m = 70$ is reduced to $m - \Delta m = 56$ without stability enforcement, and to $m - \Delta m_s = 54$ with stability enforcement. For the latter case, the 70 poles obtained after the fourth iteration and the 54 poles corresponding to the approximate GCD are plotted in Figure 6.1(c). The misfit achieved to the new 56 (respectively 54) poles corresponds to iteration 5. As this misfit is still below $\varepsilon_{\mathtt{tol}}$ no further RKFIT iterations are required.

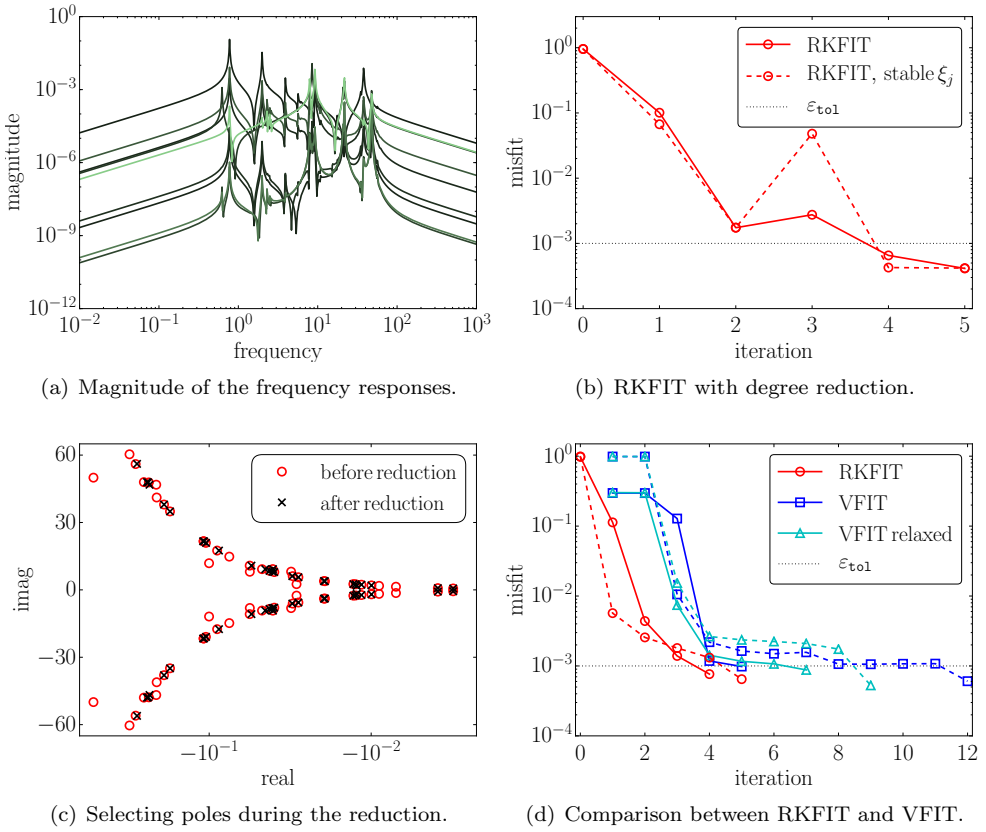
For the second experiment we compare RKFIT with the vector fitting code VFIT [13, 24, 26] for two different choices of initial poles, and with different normalization conditions for VFIT. (We briefly review VFIT in subsection A.2 of the appendix.) The results are reported in Figure 6.1(d). Here we search for type $(m-1, m)$ approximants with $m = 56$, do not enforce the poles to be stable, and do not perform any further degree reductions. The solid convergence curves are obtained with initial poles of the form $-\xi/100 \pm \mathrm{i}\xi$, with the ξ being logarithmically spaced on $[10^{-2}, 10^{3}]$. This is regarded as a good initial guess in the literature. The dashed curves result when using as initial poles the eigenvalues of a real random matrix. In both cases RKFIT outperforms VFIT, independently of the normalization condition used by VFIT. Depending on the 56 initial poles, RKFIT requires either 4 or 6 iterations. This has to be compared to Figure 6.1(b), where the 56 poles selected by our reduction strategy immediately gave a misfit below $\varepsilon_{\mathtt{tol}}$ so that no further iterations were required. This validates our approximate GCD strategy for choosing the poles after degree reduction.

6.2. Pole optimization for exponential integration. Let us consider the problem of solving a linear constant-coefficient initial-value problem

$$K\boldsymbol{u}'(t) + L\boldsymbol{u}(t) = \boldsymbol{0}, \quad \boldsymbol{u}(0) = \boldsymbol{u}_0,$$

at several time points $t_1, \ldots, t_\ell$. Problems like this arise, for example, after space-discretization of parabolic PDEs via finite differences or finite elements, in which case K and L are large sparse matrices. Assuming that K is invertible, the exact solutions $\boldsymbol{u}(t_j)$ are given as $\boldsymbol{u}(t_j) = \exp(-t_j K^{-1} L)\boldsymbol{u}_0$, and a popular approach for approximating $\boldsymbol{u}(t_j)$ is to use rational functions $r^{[j]}$ of the form

$$r^{[j]}(z) = \frac{\sigma_1^{[j]}}{\xi_1 - z} + \frac{\sigma_2^{[j]}}{\xi_2 - z} + \cdots + \frac{\sigma_m^{[j]}}{\xi_m - z},$$



(a) Magnitude of the frequency responses.

(b) RKFIT with degree reduction.

(c) Selecting poles during the reduction.

(d) Comparison between RKFIT and VFIT.

Fig. 6.1: Low-order model approximation to the MIMO system ISS from [11]. The frequency responses are plotted in figure (a)*. In* (b) *the progress of RKFIT is given for* $m = 70$ *infinite starting poles. At iteration* 4 *the degree reduction takes place. The* 70 *poles after convergence and* 54 *selected ones (for the case when stability of poles in enforced) are illustrated in figure* (c)*. Figure* (d) *presents a comparison with VFIT, when searching for* $(55, 56)$ *approximants, and using two different starting guesses. More details are given in section 6.1.*

constructed so that $r^{[j]}(K^{-1}L)\boldsymbol{u}_0 \approx \boldsymbol{u}(t_j)$. Note that the poles of $r^{[j]}$ do not depend on t_j and we have

$$r^{[j]}(K^{-1}L)\boldsymbol{u}_0 = \sum_{i=1}^{m} \sigma_i^{[j]}(\xi_i K - L)^{-1} K \boldsymbol{u}_0,$$

the evaluation of which amounts to the solution of m decoupled linear systems. Such fixed-pole approximants have great computational advantage, in particular in combination with direct solvers (the LU factorization of $\xi_i K - L$ can be used for all t_j) and on parallel computers.

The correct design of the pole-residue pairs $(\xi_i, \sigma_i^{[j]})$ is closely related to the scalar rational approximation of e^{-tz}, a problem which has received considerable attention in the literature [34, 32, 42, 18, 9]. Let us assume that L is Hermitian positive semi-definite, K is Hermitian positive definite, and define the vector K-norm

as $\|\boldsymbol{v}\|_K = \sqrt{\boldsymbol{v}^* K \boldsymbol{v}}$. Then

$$\begin{aligned}\|\exp(-t_j K^{-1}L)\boldsymbol{b} - r^{[j]}(K^{-1}L)\boldsymbol{b}\|_K &\le \|\boldsymbol{b}\|_K \max_{\lambda\in\Lambda(L,K)} |e^{-t_j\lambda} - r^{[j]}(\lambda)| \\ &\le \|\boldsymbol{b}\|_K \max_{\lambda\ge 0} |e^{-t_j\lambda} - r^{[j]}(\lambda)|, \end{aligned} \tag{6.1}$$

with $\Lambda(L,K)$ denoting the set of generalized eigenvalues of (L,K).

In order to use RKFIT for finding poles $\xi_1,\dots,\xi_m$ of the rational functions $r^{[j]}$ such that the right-hand side (6.1) of the inequality is small for all $j = 1,\dots,\ell$, we propose a surrogate approach similar to that in [9]. Let $A = \operatorname{diag}(\lambda_1,\dots,\lambda_N)$ be a diagonal matrix with "sufficiently dense" eigenvalues on $\lambda \ge 0$. In this example we take $N = 500$ logspaced eigenvalues on the interval $[10^{-6}, 10^6]$. Further, we define $\ell = 41$ logspaced time points t_j on the interval $[10^{-1}, 10^1]$, and the matrices $F^{[j]} = \exp(-t_j A)$. We also define $\boldsymbol{b} = [1 \quad \dots \quad 1]^T$ to assign equal weight to each eigenvalue of A and then run RKFIT for finding a family of type $(m-1, m)$ rational functions $r^{[j]}$ with $m = 12$ so that

$$\text{absmisfit} = \sum_{j=1}^{\ell} \|F^{[j]}\boldsymbol{b} - r^{[j]}(A)\boldsymbol{b}\|_2^2$$

is minimized. Note that

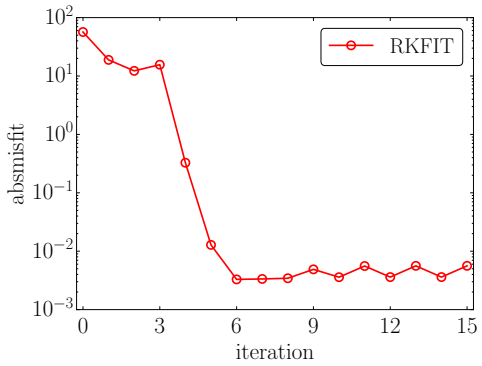
$$\text{absmisfit} \ge \sum_{j=1}^{\ell} \|F^{[j]}\boldsymbol{b} - r^{[j]}(A)\boldsymbol{b}\|_\infty^2 = \sum_{j=1}^{\ell} \Big(\max_{\lambda\in\Lambda(A)} |e^{-t_j\lambda} - r^{[j]}(\lambda)| \Big)^2,$$

and hence a small misfit implies that all $r^{[j]}$ are accurate uniform approximants for $e^{-t_j\lambda}$ on the eigenvalues $\Lambda(A)$. If these eigenvalues are dense enough on $\lambda \ge 0$ one can expect the upper error bound (6.1) to be tight.

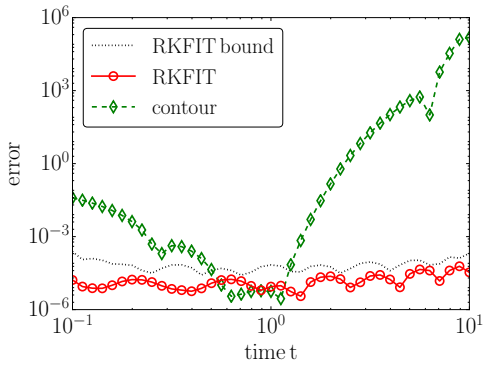
Figure 6.2(a) shows the convergence of RKFIT, starting from an initial guess of $m = 12$ poles at infinity (iteration 0 corresponds to the absolute misfit of the linearised rational approximation problem). We find that RKFIT attains its smallest absolute misfit of $\approx 3.44 \times 10^{-3}$ after 6 iterations. From iteration 7 onwards the misfit slightly oscillates about the stagnation level. To evaluate the quality of the common-pole rational approximants for all $\ell = 41$ time points t_j, we perform an experiment similar to that in [42, Figure 6.1] by approximating $\boldsymbol{u}(t_j) = \exp(-t_j L)\boldsymbol{u}_0$ and comparing the result to MATLAB's `expm`. Here, $L \in \mathbb{R}^{2401\times 2401}$ is a finite-difference discretization of the scaled 2D Laplace operator -0.02Δ on the domain $[-1,1]^2$ with homogeneous Dirichlet boundary condition, and $\boldsymbol{u}_0$ corresponds to the discretization of $u_0(x,y) = (1-x^2)(1-y^2)e^x$ on that domain. Figure 6.2(b) shows the error $\|\boldsymbol{u}(t_j) - r^{[j]}(L)\boldsymbol{u}_0\|_2$ for each time point t_j (solid curve with circles), together with the approximate upper error bound $\|\exp(-t_j A)\boldsymbol{b} - r^{[j]}(A)\boldsymbol{b}\|_\infty$ (dotted curve). We see that the error is approximately uniform and smaller than 6.21×10^{-5} over the whole time interval $[10^{-1}, 10^1]$. The $m = 12$ poles of the rational functions $r^{[j]}$ are shown in Figure 6.2(c) (circles).

Another approach for obtaining a family of rational approximants is to use contour integration [42]. Applying an m-point quadrature rule to the Cauchy integral

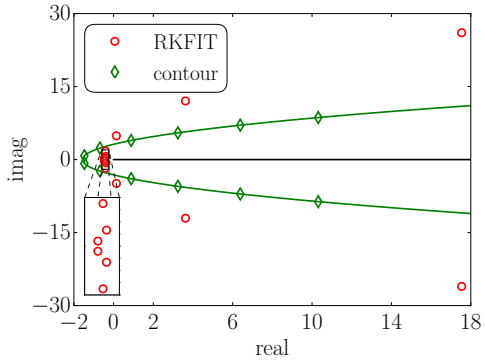
$$e^{-t_j z} = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{e^{-t_j\xi}}{\xi - z}\,\mathrm{d}\xi \approx \sum_{i=1}^{m} \frac{\sigma_i^{[j]}}{\xi_i - z} =: \widetilde{r}^{[j]}(z)$$



(a) Convergence of RKFIT.



(b) RKFIT vs. the contour approach.



(c) Poles from the two approaches.

m	error	bound
4	5.71×10^{-02}	7.35×10^{-02}
8	1.72×10^{-03}	4.89×10^{-03}
12	6.21×10^{-05}	2.47×10^{-04}
16	5.94×10^{-06}	1.41×10^{-05}
20	3.23×10^{-07}	7.98×10^{-07}
24	9.53×10^{-09}	6.62×10^{-08}
28	4.54×10^{-10}	4.76×10^{-09}
32	2.64×10^{-11}	2.44×10^{-10}

(d) Quality of RKFIT as m increases.

Fig. 6.2: Approximating $\exp(-tL)\boldsymbol{u}_0$ *for a range of parameters* t *with rational approximants sharing common poles. The convergence behaviour of RKFIT, for approximants of type* $(11, 12)$*, is shown in* (a)*. In* (b) *we show the approximation error for* $\ell = 41$ *logspaced time points* $t \in [0.1, 10]$ *for RKFIT (solid curve with circles) and the contour-based approach (dashed curve with diamonds). The errors of the RKFIT surrogate approximants are also indicated (these are approximate upper error bound for the RKFIT approximants). In* (c) *we show the pole locations of the two families of rational approximants in the complex plane. The small rectangle shows a five-fold magnification of the RKFIT poles near the origin. The table in* (d) *shows the maximal RKFIT error and the approximate upper error bound, uniformly over all time points* $t_j \in [10^{-1}, 10^{1}]$*, for various degrees* m*.*

on a contour Γ enclosing the positive real axis, one obtains a family of rational functions $\widetilde{r}^{[j]}$ whose poles are the quadrature points $\xi_i \in \Gamma$ and whose residuals $\sigma_i^{[j]}$ depend on t_j. As has already been pointed out in [42], such quadrature-based approximants tend be good only for a small range of parameters t_j. In Figure 6.2(b) we see that the error $\|\boldsymbol{u}(t_j) - \widetilde{r}^{[j]}(L)\boldsymbol{u}_0\|_2$ increases very rapidly away from $t = 1$ (dashed curve with diamonds). We have used the same degree parameter $m = 12$ as above and the poles of the $\widetilde{r}^{[j]}$, which all lie on a parabolic contour [42, formula (3.1)], are shown in Figure 6.2(c) (diamonds).

We believe that RKFIT may be a valuable tool for designing efficient exponential integrators based on partial fractions or rational Krylov techniques (see, e.g., [18, 9]). The table in Figure 6.2(d) shows that very high accuracies can be achieved

with a relatively small degree parameter m. It is also straightforward to incorporate weight matrices $D^{[j]}$ depending on t_j, which may be useful for minimizing the *relative* approximation error uniformly over a time interval, instead of the absolute error as in this example.

7. Summary and future work. We have presented an extension of the RKFIT algorithm to more general rational approximation problems, alongside with other improvements concerning the evaluation and transformation of the underlying rational functions, as well as root-finding. A main feature of the new RKFIT implementation is its automated degree reduction.

In future work we plan to investigate closer the relation of our degree reduction procedure to the problem of finding an approximate polynomial GCD [8]. We would also like to extend the partial fraction conversion to the case of repeated poles (both finite and infinite), which then amounts to bringing the lower $m \times m$ part of the pencil to Jordan canonical form instead of diagonal form. Such transformation raises the problem of deciding when nearby poles should be treated as a single Jordan block. A stable algorithm for computing a "numerical Jordan form" has been discussed in [29].

The automated degree reduction opens the possibility for "Chebfun-like computing" [14] with rational functions, e.g., allowing for summation, multiplication, or differentiation of rational functions, followed by a degree truncation of the resulting rational function. However, rational functions are generally more difficult to deal with than polynomials as, for example, integration is not a closed operation: the integral of a rational function may contain logarithmic terms.

Other interesting problems include the extension of RKFIT to rational block-Krylov spaces, with the potential of solving tangential interpolation problems (see, e.g., [19]), and the application of RKFIT for constructing rational filter functions.

Acknowledgments. We would like to thank Vladimir Druskin, Zlatko Drmač, Serkan Gugercin, and Marc Van Barel for stimulating discussions. We are also grateful to the anonymous referees who provided many useful suggestions.

Appendix A. Relations to iterative reweighting and vector fitting.

Here we consider scalar rational approximation problems, like the one encountered in the introduction. In our discussion we refrain from using weights, set $\ell = 1$, and fix the type of the rational approximant to $(m-1, m)$, for the sake of simplicity only. Hence, we consider the following problem: given data $\{(\lambda_i, f_i)\}_{i=1}^N$ with pairwise distinct λ_i, find a rational function r of type $(m-1, m)$ such that

$$\sum_{i=1}^{N} |f_i - r(\lambda_i)|^2 \to \min. \tag{A.1}$$

A popular approach for solving problems of this form introduced in [26] and designed to fit frequency response measurements of dynamical systems is vector fitting (VFIT).

As already observed in [6], numerical experiments indicate that RKFIT performs more robustly than VFIT. The main goal of this section is to clarify the differences and commons between the two methods. In section A.1 we briefly review the predecessors of VFIT, followed by a derivation of VFIT in section A.2. In section A.3 we reformulate VFIT in the spirit of RKFIT in order to compare the two methods. Other aspects of VFIT, applicable to RKFIT as well, are discussed in section A.4.

A.1. Iteratively reweighted linearisation. The first attempt to solve the nonlinear problem (A.1) was through linearisation [31]. Let us write $r = p_{m-1}/q_m$ with $p_{m-1} \in \mathcal{P}_{m-1}$ and $q_m \in \mathcal{P}_m$. Then the relation

$$\sum_{i=1}^{N} |f_i - r(\lambda_i)|^2 = \sum_{i=1}^{N} \frac{|f_i q_m(\lambda_i) - p_{m-1}(\lambda_i)|^2}{|q_m(\lambda_i)|^2}$$

inspired Levy [31] to replace (A.1) with the problem of finding $p_{m-1}(z) = \sum_{j=0}^{m-1} \alpha_j z^j$ and $q_m(z) = 1 + \sum_{j=1}^{m} \beta_j z^j$ such that $\sum_{i=1}^{N} |f_i q_m(\lambda_i) - p_{m-1}(\lambda_i)|^2$ is minimal. The latter problem is linear in the unknowns $\{\alpha_{j-1}, \beta_j\}_{j=1}^{m}$ and hence straightforward to solve. However, as q_m may vary substantially in magnitude over the nodes λ_i, the solution $r = p_{m-1}/q_m$ may be a poor approximation to a solution of (A.1).

As a remedy, Sanathanan and Koerner [40] suggested to replace the nonlinear problem (A.1) with a sequence of linear problems. Once the linearised problem $\sum_{i=1}^{N} |f_i q_m(\lambda_i) - p_{m-1}(\lambda_i)|^2 \to \min$ has been solved, one can set $\widehat{q}_m := q_m$ and solve a reweighted linear problem $\sum_{i=1}^{N} \frac{|f_i q_m(\lambda_i) - p_{m-1}(\lambda_i)|^2}{|\widehat{q}_m(\lambda_i)|^2} \to \min$. This process can be iterated until a satisfactory approximation has been obtained or a maximal number of iterations has been performed.

A.2. Vector fitting. Vector fitting is a reformulation of the Sanathanan–Koerner algorithm, where the polynomials p_{m-1} and q_m are not expanded in the monomial basis, but in a Lagrange basis written in barycentric form. Similarly to RKFIT, in VFIT one starts with an initial guess q_m of degree m for the denominator, but here with pairwise distinct finite roots $\{\xi_j\}_{j=1}^{m} \cap \{\lambda_i\}_{i=1}^{N} = \emptyset$, and iteratively tries to improve it as follows. Write again $r = p_{m-1}/q_m$ with p_{m-1} and q_m being unknown. Then r can be represented in barycentric form with interpolation nodes $\{\xi_j\}_{j=1}^{m}$,

$$r(z) = \frac{p_{m-1}(z)}{q_m(z)} = \frac{p_{m-1}/\widehat{q}_m(z)}{q_m(z)/\widehat{q}_m(z)} = \frac{\sum_{j=1}^{m} \frac{\varphi_j}{z-\xi_j}}{1 + \sum_{j=1}^{m} \frac{\psi_j}{z-\xi_j}}. \tag{A.2}$$

The coefficients φ_j and ψ_j are the unknowns to be determined. Once found, we use them to detect better interpolation nodes for the barycentric representation, and it is hoped that, by iterating the process, those will ultimately converge to the poles of an (approximate) minimizer r.

The linearised version of (A.2) reads

$$r(z)\left(1 + \sum_{j=1}^{m} \frac{\psi_j}{z-\xi_j}\right) = \sum_{j=1}^{m} \frac{\varphi_j}{z-\xi_j}. \tag{A.3}$$

Inserting $z = \lambda_i$ and replacing $r(\lambda_i)$ with f_i in (A.3) for $i = 1, \ldots, N$ gives a linear system of equations

$$\begin{bmatrix} \frac{1}{\lambda_1-\xi_1} & \cdots & \frac{1}{\lambda_1-\xi_m} & \frac{-f_1}{\lambda_1-\xi_1} & \cdots & \frac{-f_1}{\lambda_1-\xi_m} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{1}{\lambda_N-\xi_1} & \cdots & \frac{1}{\lambda_N-\xi_m} & \frac{-f_N}{\lambda_N-\xi_1} & \cdots & \frac{-f_N}{\lambda_N-\xi_m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi} \\ \boldsymbol{\psi} \end{bmatrix} = \boldsymbol{f}, \tag{A.4}$$

which is solved in the LS sense. Afterwards, the poles $\{\xi_j\}_{j=1}^{m}$ are replaced by the roots of the denominator $1 + \sum_{j=1}^{m} \frac{\psi_j}{z-\xi_j}$. Iterating this process gives the VFIT algorithm. The reweighting as in the Sanathanan–Koerner algorithm is implicitly achieved in VFIT through the change of interpolation nodes for the barycentric representation.

A.3. On the normalization condition. Although different approaches are used, both mathematically and numerically, RKFIT and VFIT are similar. However, there is a considerable difference in the way the poles are relocated. Let us introduce

$$C_{m+1} = \begin{bmatrix} 1 & \frac{1}{\lambda_1-\xi_1} & \cdots & \frac{1}{\lambda_1-\xi_m} \\ \vdots & \vdots & & \vdots \\ 1 & \frac{1}{\lambda_N-\xi_1} & \cdots & \frac{1}{\lambda_N-\xi_m} \end{bmatrix}, \qquad F = \begin{bmatrix} f_1 & & \\ & \ddots & \\ & & f_N \end{bmatrix},$$

and $\widehat{C}_m = C_{m+1} \begin{bmatrix} \mathbf{0} & I_m \end{bmatrix}^T$. We now rewrite (A.4) in the equivalent form

$$\begin{bmatrix} \widehat{C}_m & -FC_{m+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi} \\ \psi_0 \\ \boldsymbol{\psi} \end{bmatrix} = \mathbf{0}, \tag{A.5}$$

with $\psi_0 = 1$. For any fixed $\boldsymbol{\psi} \in \mathbb{C}^m$, solving (A.5) for $\boldsymbol{\varphi} \in \mathbb{C}^m$ subject to $\psi_0 = 1$ in the LS sense is equivalent to solving $\widehat{C}_m \boldsymbol{\varphi} = FC_{m+1}[1 \quad \boldsymbol{\psi}^T]^T$ in the LS sense. Under the (reasonable) assumption that $\widehat{C}_m \in \mathbb{C}^{N\times m}$ is of full column rank with $m \leq N$, the unique solution is given by $\boldsymbol{\varphi} = \widehat{C}_m^{\dagger} FC_{m+1}[1 \quad \boldsymbol{\psi}^T]^T$.

Therefore, when solving (A.4) in VFIT one gets $r = \dfrac{\widehat{p}_m/q_m}{\widehat{q}_m/q_m}$, where $\widehat{q}_m(z)/q_m(z) = 1 + \sum_{j=1}^m \frac{\psi_j}{z-\xi_j}$ and $\widehat{p}_m(z)/q_m(z) = \sum_{j=1}^m \frac{\varphi_j}{z-\xi_j}$ is the projection of $f\widehat{q}_m/q_m$ onto the target space, with f being defined on the discrete set of interpolation nodes as $f(\lambda_i) = f_i$ and the target space being represented by $\widehat{C}_m$.

Both VFIT and RKFIT solve a LS problem at each iteration, with the projection space represented in the partial fraction basis (VFIT) or via discrete-orthogonal rational functions (RKFIT). Apart from the potential ill-conditioning of the partial fraction basis, the main difference between VFIT and RKFIT are the constraints under which the LS problems are solved. The constraint in VFIT is for $\widehat{q}/q$ to have a unit absolute term, $\psi_0 = 1$. This asymptotic requirement degrades the convergence properties of VFIT, especially when the approximate poles ξ_j are far from those of a true minimizer and the nodes λ_i vary over a large scale of magnitudes. This was observed in [24], and as a fix it was proposed to use instead the condition $\Re\left\{\sum_{i=1}^N \left(\sum_{j=1}^m \frac{\psi_j}{\lambda_i-\xi_j} + \psi_0\right)\right\} = \Re\left\{N\psi_0 + \sum_{j=1}^m \left(\sum_{i=1}^N \frac{1}{\lambda_i-\xi_j}\right)\psi_j\right\} = N$, incorporated as an additional equation in (A.4). This modification to a global normalization condition avoids the problems with point-wise normalization conditions exemplified in the introduction. VFIT with this additional constrained is known as relaxed VFIT. The normalization condition in RKFIT is also of global nature, $\|\boldsymbol{v}\|_2 = \|\widehat{q}(A)q(A)^{-1}\boldsymbol{b}\|_2 = 1$; cf. line 5 in Algorithm 2.1.

A.4. On the choice of basis. In VFIT the approximant is expanded in the basis of partial fractions which may lead to ill-conditioned linear algebra problems, as can be anticipated by the appearance of Cauchy-like matrices; cf. (A.4). *Orthonormal vector fitting* was proposed as a remedy in [12], where the basis of partial fractions was replaced by an orthonormal basis. Soon after it was claimed [25] that a numerically more careful implementation of VFIT is as good as the orthonormal VFIT variant proposed in [12], and hence the orthonormal VFIT never became a reality.

The problem with the orthonormal VFIT [12] is that the orthonormal basis is computed by a Gram–Schmidt procedure applied to partial fractions, i.e., an ill-conditioned basis is transformed into an orthonormal one, hence ill-conditioned linear

algebra is not avoided. The orthonormal basis in RKFIT is obtained from successively applying a single partial fraction to the last basis vector, which amounts to the orthogonalisation of a basis with typically lower condition number.

Numerical issues arising in VFIT have been recently discussed and mitigated in [15, 16]. Our approach avoids these problems altogether.

So far we assumed the interpolation nodes λ_i to be given. If they can be chosen freely, one can choose them as nodes of certain quadrature rules tailored to the application in the hope to improve both the numerical stability as well as the approximation quality. This idea is suggested in [15, 16] for the discretized $\mathcal{H}_2$ approximation of transfer function measurements and it carries over straightforwardly to RKFIT.

A.5. Convergence. As to date, there are no complete convergence analyses for VFIT and RKFIT available. Both algorithms have the property that if a rational function is fitted with sufficiently many nodes, then in the absence of rounding errors this function is recovered exactly; see [30, Corollary III.1] and our Theorem 2.2. Some further work is available for VFIT. In [30, Section IV], and subsequently in [41], a degree $m = 2$ example is constructed where the VFIT fixed-point iteration is repellent and hence diverges, independently of the starting guess for the poles. Furthermore, it is known that VFIT does not necessarily satisfy first-order optimality conditions for the nonlinear LS problem upon convergence to a fixed point [41]. In our numerical experiments we typically observe that RKFIT reduces the fitting error more efficiently than VFIT, however, oscillations around a stagnation level may still occur; see, e.g., Figure 6.2(a). Furthermore, we observed that for the example specified in [41, Table I], RKFIT exhibits an oscillatory behavior similar to VFIT.

Despite a few constructed examples of nonconvergence, VFIT has been used successfully by the IEEE community for various (large-scale) rational fitting problems. We have argued and demonstrated with (scalar) examples that RKFIT is more robust and typically faster convergent than VFIT. Additionally, unlike VFIT, RKFIT is equipped with an automated degree reduction procedure. Therefore, we believe that RKFIT may be a useful algorithm for the IEEE community. For nonscalar approximation problems where A and F are not necessarily diagonalizable, we are currently not aware of an algorithm similar to RKFIT.

REFERENCES

[1] Advanpix LLC., *Multiprecision Computing Toolbox for MATLAB*, ver 3.8.3.8882, Tokyo, Japan, 2015. `http://www.advanpix.com/`.

[2] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems*, Contemp. Math., 280 (2001), pp. 193–220.

[3] I. Barrodale and J. Mason, *Two simple algorithms for discrete rational approximation*, Math. Comp., 24 (1970), pp. 877–891.

[4] M. Berljafa, *Rational Krylov Decompositions: Theory and Applications*, PhD thesis, The University of Manchester, Manchester, UK, 2017. Available as MIMS EPrint 2017.6 at `http://eprints.ma.man.ac.uk/2529/`.

[5] M. Berljafa and S. Güttel, *A Rational Krylov Toolbox for MATLAB*, MIMS EPrint 2014.56, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2014. Available for download at `http://rktoolbox.org`.

[6] M. Berljafa and S. Güttel, *Generalized rational Krylov decompositions with an application to rational approximation*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 894–916.

[7] H. Blinchikoff and A. Zverev, *Filtering in the Time and Frequency Domains*, John Wiley & Sons Inc., New York, 1976.

[8] P. Boito, *Structured Matrix Based Methods for Approximate Polynomial GCD*, vol. 15, Springer Science & Business Media, 2012.

[9] R.-U. Börner, O. G. Ernst, and S. Güttel, *Three-dimensional transient electromagnetic modeling using rational Krylov methods*, Geophys. J. Int., 202 (2015), pp. 2025–2043.
[10] D. Braess, *Nonlinear Approximation Theory*, Berlin, Germany, 1986.
[11] Y. Chahlaoui and P. Van Dooren, *A collection of benchmark examples for model reduction of linear time invariant dynamical systems*, MIMS EPrint 2008.22, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2008.
[12] D. Deschrijver, B. Haegeman, and T. Dhaene, *Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses*, IEEE Trans. Adv. Packag., 30 (2007), pp. 216–225.
[13] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, *Macromodeling of multiple systems using a fast implementation of the vector fitting method*, IEEE Microw. Compon. Lett., 18 (2008), pp. 383–385.
[14] T. A. Driscoll, N. Hale, and L. N. Trefethen, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.
[15] Z. Drmač, S. Gugercin, and C. Beattie, *Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation*, SIAM J. Sci. Comput., 37 (2015), pp. A625–A652.
[16] Z. Drmač, S. Gugercin, and C. Beattie, *Vector fitting for matrix-valued rational approximation*, SIAM J. Sci. Comput., 37 (2015), pp. A2346–A2379.
[17] V. Druskin, S. Güttel, and L. Knizhnerman, *Compressing variable-coefficient exterior Helmholtz problems via RKFIT*, MIMS EPrint 2016.53, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2016.
[18] V. Druskin, L. Knizhnerman, and M. Zaslavsky, *Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts*, SIAM J. Sci. Comput., 31 (2009), pp. 3760–3780.
[19] V. Druskin, V. Simoncini, and M. Zaslavsky, *Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 476–498.
[20] K. Gallivan, E. Grimme, and P. Van Dooren, *A rational Lanczos algorithm for model reduction*, Numer. Algorithms, 12 (1996), pp. 33–63.
[21] P. Gonnet, S. Güttel, and L. N. Trefethen, *Robust Padé approximation via SVD*, SIAM Rev., 55 (2013), pp. 101–117.
[22] P. Gonnet, R. Pachón, and L. N. Trefethen, *Robust rational interpolation and least-squares*, Electron. Trans. Numer. Anal., 38 (2011), pp. 146–167.
[23] S. Gugercin, A. Antoulas, and C. Beattie, *A rational Krylov iteration for optimal $\mathcal{H}_2$ model reduction*, in Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006, pp. 1665–1667.
[24] B. Gustavsen, *Improving the pole relocating properties of vector fitting*, IEEE Trans. Power Del., 21 (2006), pp. 1587–1592.
[25] B. Gustavsen, *Comments on "A comparative study of vector fitting and orthonormal vector fitting techniques for EMC applications"*, in Proceedings of the 18th International Zurich Symposium on Electromagnetic Compatibility, Zurich, Switzerland, 2007, pp. 131–134.
[26] B. Gustavsen and A. Semlyen, *Rational approximation of frequency domain responses by vector fitting*, IEEE Trans. Power Del., 14 (1999), pp. 1052–1061.
[27] S. Güttel, *Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection*, GAMM-Mitt., 36 (2013), pp. 8–31.
[28] D. Ingerman, V. Druskin, and L. Knizhnerman, *Optimal finite difference grids and rational approximations of the square root I. Elliptic problems*, Comm. Pure Appl. Math., 53 (2000), pp. 1039–1066.
[29] B. Kågström and A. Ruhe, *An algorithm for numerical computation of the Jordan normal form of a complex matrix*, ACM Trans. Math. Software, 6 (1980), pp. 398–419.
[30] S. Lefteriu and A. Antoulas, *On the convergence of the vector-fitting algorithm*, IEEE Trans. Microw. Theory Techn., 61 (2013), pp. 1435–1443.
[31] E. C. Levy, *Complex-curve fitting*, IRE Trans. Autom. Control, AC-4 (1959), pp. 37–43.
[32] I. Moret and P. Novati, *RD-rational approximations of the matrix exponential*, BIT, 44 (2004), pp. 595–615.
[33] Y. Nakatsukasa and R. W. Freund, *Using Zolotarev's rational approximation for computing the polar, symmetric eigenvalue, and singular value decompositions*, Tech. Report METR 2014–35, The University of Tokyo, 2014.
[34] S. P. Nørsett, *Restricted Padé approximations to the exponential function*, SIAM J. Numer. Anal., 15 (1978), pp. 1008–1029.
[35] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.

[36] A. Ruhe, *The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices*, BIT, 34 (1994), pp. 165–176.
[37] A. Ruhe, *Rational Krylov algorithms for nonsymmetric eigenvalue problems. II. Matrix pairs*, Linear Algebra Appl., 198 (1994), pp. 283–295.
[38] A. Ruhe, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551.
[39] A. Ruhe and D. Skoogh, *Rational Krylov algorithms for eigenvalue computation and model reduction*, in Applied Parallel Computing Large Scale Scientific and Industrial Problems, B. Kågström, J. Dongarra, E. Elmroth, and J. Waśniewski, eds., vol. 1541 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1998, pp. 491–502.
[40] C. Sanathanan and J. Koerner, *Transfer function synthesis as a ratio of two complex polynomials*, IEEE Trans. Automat. Control, 8 (1963), pp. 56–58.
[41] G. Shi, *On the nonconvergence of the vector fitting algorithm*, IEEE Trans. Circuits Syst. II, Exp. Briefs, 63 (2016), pp. 718–722.
[42] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer, *Talbot quadratures and rational approximations*, BIT, 46 (2006), pp. 653–670.
[43] G. Wanner, E. Hairer, and S. Nørsett, *Order stars and stability theorems*, BIT, 18 (1978), pp. 475–489.