

*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 BERLJAJA* AND STEFAN GÜTTEL*

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

Abstract. The RKFIT algorithm outlined in [M. BERLJAJA 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 common 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 [18, 24, 2, 21, 17]. Some other areas where rational approximants play an important role are analogue filter design [6], time-stepping methods [41], transparent boundary conditions [27], and iterative methods in numerical linear algebra (see, e.g., [31, 40, 16, 25, 32]). 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

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

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 [9, p. 109] is to take fixed $m \geq 1$ and $N > 2m$, and let

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

* School of Mathematics, The University of Manchester, Alan Turing Building, Oxford Road, M13 9PL Manchester, United Kingdom, mario.berljaja@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 $j \rightarrow \infty$, 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

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

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 $p_m \neq 0$. 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 [20, 19].

The aim of this paper is to present and analyse an extension of the RKFIT algorithm initially outlined in [5]. RKFIT is an iterative method for solving rational LS problems more general than (1.1). In its original form, for given matrices $\{A, F\} \subset \mathbb{C}^{N \times N}$ and a vector $\mathbf{b} \in \mathbb{C}^N$, RKFIT attempts to find a rational function r of type (m, m) such that

$$\|F\mathbf{b} - r(A)\mathbf{b}\|_2^2 \rightarrow \min. \quad (1.4)$$

Note that this problem contains (1.1) as a special case with $F = \text{diag}(f_i)$, $A = \text{diag}(\lambda_i)$, and $\mathbf{b} = [\sqrt{w_1} \ \dots \ \sqrt{w_N}]^T$. 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 $F \approx f(A)$.

Here we extend RKFIT to approximation problems involving a family of matrices $\{F^{[j]}\}_{j=1}^\ell \subset \mathbb{C}^{N \times N}$, a block of vectors $B = [\mathbf{b}_1 \ \dots \ \mathbf{b}_n] \in \mathbb{C}^{N \times n}$, and rational functions of type $(m+k, m)$ with k not necessarily being equal to zero. 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.,

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

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 applications of such approximation problems. 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):

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

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* [4] available online¹. 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 \mathbf{x}_i being defined in MATLAB, the user simply calls

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

to obtain a rational function r represented as a MATLAB object `ratfun`. The toolbox implements several methods for `ratfun`, for example, the evaluation of r at scalar arguments or as a matrix function; the commands `ratfun(z)` and `ratfun(A,B)` evaluate $r(z)$ and $r(A)B$, respectively (where A and B need not be the same matrices as used for the construction of `ratfun`). The conversion of a `ratfun` 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 [13].

Alongside with the extension of RKFIT from (1.4) to (1.5), another contribution of this paper is Theorem 3.1 and its Corollary 3.2, showing that RKFIT solves (1.5) exactly if the $F^{[j]}$ are rational matrix functions of type $(m + k, m)$. Further, in section 4, we propose an automated procedure for decreasing the degree parameters m and k , thereby reducing possible deficiencies in the rational approximants. In section 5 we develop a new approach for the efficient evaluation of the rational approximants (and their derivatives) produced by RKFIT. We also show how to compute the roots of these approximants, or 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 [24].

We note that initially we consider the case $n = 1$, i.e., $B = \mathbf{b} \in \mathbb{C}^N$ is a single vector. This is merely for simplicity of exposition, and the extension to the block-case $n \geq 1$ is given in section 3.5.

2. Preliminaries. Our derivation of RKFIT relies on rational Krylov spaces and in this section we review the required facts and introduce notation.

¹<http://guettel.com/rktoolbox/>

2.1. Rational Krylov spaces. Given a matrix $A \in \mathbb{C}^{N \times N}$, a (so-called) starting vector $\mathbf{b} \in \mathbb{C}^N$, 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* as

$$\mathcal{Q}_{m+1} \equiv \mathcal{Q}_{m+1}(A, \mathbf{b}, q_m) := q_m(A)^{-1} \mathcal{K}_{m+1}(A, \mathbf{b}),$$

where $\mathcal{K}_{m+1} \equiv \mathcal{K}_{m+1}(A, \mathbf{b}) := \text{span}\{\mathbf{b}, A\mathbf{b}, \dots, A^m \mathbf{b}\}$ is a *polynomial Krylov space*. There exists an integer $M(A, \mathbf{b}) \equiv M \leq N$, called the *invariance index* for (A, \mathbf{b}) , such that $\mathcal{K}_1 \subset \mathcal{K}_2 \subset \dots \subset \mathcal{K}_{M-1} \subset \mathcal{K}_M = \mathcal{K}_{M+1}$. Throughout this work we assume that $m < M$, in which case both $\mathcal{K}_{m+1}(A, \mathbf{b})$ and $\mathcal{Q}_{m+1}(A, \mathbf{b}, q_m)$ are of full dimension $m+1$. As a consequence, each vector $\mathbf{v} \in \mathcal{Q}_{m+1}(A, \mathbf{b}, q_m)$ corresponds to a unique rational function $r = p_m/q_m$ of type (m, m) such that $\mathbf{v} = r(A)\mathbf{b}$, and vice versa. We denote this correspondence by $r \equiv \mathbf{v}$.

The roots of q_m are called *poles* of the rational Krylov space and denoted by $\xi_1, \xi_2, \dots, \xi_m$. We often 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 .

2.2. The rational Krylov method. The rational Krylov method [34, 35, 36, 37] constructs an orthonormal basis $V_{m+1} \in \mathbb{C}^{N \times (m+1)}$ of $\mathcal{Q}_{m+1}(A, \mathbf{b}, q_m)$ by applying the Gram–Schmidt process to $\{\mathbf{b}, p_1(A)q_1(A)^{-1}\mathbf{b}, \dots, p_m(A)q_m(A)^{-1}\mathbf{b}\}$, with $q_j(z) = \prod_{\substack{i=1 \\ \xi_i \neq \infty}}^j (z - \xi_i)$ and some nonzero polynomials $p_j \in \mathcal{P}_j$. The method is listed in Algorithm 2.1. It computes a decomposition of the form

$$AV_{m+1}\underline{K}_m = V_{m+1}\underline{H}_m, \quad (2.1)$$

where $(\underline{H}_m, \underline{K}_m)$ is an $(m+1) \times m$ upper-Hessenberg pencil satisfying $|h_{j+1,j}| + |k_{j+1,j}^j| \neq 0$ and $\xi_j = h_{j+1,j}/k_{j+1,j}^j \notin \Lambda(A)$ for $j = 1, \dots, m$. Any such decomposition is called a *rational Arnoldi decomposition (RAD)*, see [5, Definition 2.3]. Further, the numbers $h_{j+1,j}/k_{j+1,j}^j$ are called the *poles* of the pencil $(\underline{H}_m, \underline{K}_m)$. With any rational Krylov space $\mathcal{Q}_{m+1} \equiv \mathcal{Q}_{m+1}(A, \mathbf{b}, q_m)$ we can associate a RAD (2.1) such that $\mathcal{R}(V_{m+1}) = \mathcal{Q}_{m+1}$, \mathbf{v}_1 is collinear to \mathbf{b} , and the poles of $(\underline{H}_m, \underline{K}_m)$ are the (formal) roots of q_m . Here, $\mathcal{R}(V_{m+1})$ denotes the range of V_{m+1} . In this sense we say that (2.1) is a RAD for \mathcal{Q}_{m+1} .

Algorithm 2.1 Rational Krylov method [34, 36, 37]. RK Toolbox [4]: `rat_krylov`

Input: $A \in \mathbb{C}^{N \times N}$, $\mathbf{b} \in \mathbb{C}^N$, and poles $\{\xi_j\}_{j=1}^m \subset \overline{\mathbb{C}} \setminus \Lambda(A)$ with $m < M(A, \mathbf{b})$.

Output: Decomposition (2.1).

1. For every $j = 1, \dots, m$, take any $(\nu_j, \mu_j, \rho_j, \eta_j)$ satisfying (2.3).
 2. Set $\mathbf{v}_1 := \mathbf{b}/\|\mathbf{b}\|_2$.
 3. **for** $j = 1, \dots, m$ **do**
 4. Choose nonzero $\mathbf{t}_j \in \mathbb{C}^j$. See discussion following (2.2).
 5. Compute $\mathbf{w} := (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)V_j \mathbf{t}_j$.
 6. Project $\mathbf{y}_j := V_j^* \mathbf{w}$ and compute $y_{j+1,j} := \|\mathbf{w} - V_j \mathbf{y}_j\|_2$.
 7. Compute $\mathbf{v}_{j+1} := (\mathbf{w} - V_j \mathbf{y}_j)/y_{j+1,j}$ orthogonal to $\mathbf{v}_1, \dots, \mathbf{v}_j$.
 8. Set $\underline{\mathbf{k}}_j := \nu_j \underline{\mathbf{y}}_j - \rho_j \underline{\mathbf{t}}_j$ and $\underline{\mathbf{h}}_j := \mu_j \underline{\mathbf{y}}_j - \eta_j \underline{\mathbf{t}}_j$, where $\underline{\mathbf{t}}_j = [\mathbf{t}_j^T \ 0]^T$. See (2.2).
 9. **end for**
-

Let us discuss briefly the choice of the parameters in Algorithm 2.1. From lines 5–8 we retrieve (for $j = 1, \dots, m$)

$$V_{j+1}\underline{\mathbf{y}}_j = (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)V_j \underline{\mathbf{t}}_j, \quad (2.2a)$$

$$AV_{j+1}(\nu_j \underline{\mathbf{y}}_j - \rho_j \underline{\mathbf{t}}_j) = V_{j+1}(\mu_j \underline{\mathbf{y}}_j - \eta_j \underline{\mathbf{t}}_j), \quad (2.2b)$$

$$AV_{j+1}\underline{\mathbf{k}}_j = V_{j+1}\underline{\mathbf{h}}_j. \quad (2.2c)$$

The vector $\underline{\mathbf{t}}_j \in \mathbb{C}^j$ is called *continuation combination* as $V_j \underline{\mathbf{t}}_j$ is used to expand the space. For $\underline{\mathbf{t}}_j$ we use a unit 2-norm left null vector of $\mu_j \underline{K}_{j-1} - \nu_j \underline{H}_{j-1}$ if $j > 1$, and $\underline{\mathbf{t}}_1 = \mathbf{e}_1$ if $j = 1$, as proposed in [37]. This left null vector is unique, up to scaling, as $\mu_j \underline{K}_{j-1} - \nu_j \underline{H}_{j-1}$ is of full column rank [5, Section 2].

To handle both finite and infinite poles $\{\xi_j\}_{j=1}^m$ in a unified way, we introduce for each j the quadruple $(\nu_j, \mu_j, \rho_j, \eta_j) \in \mathbb{C}^4$ such that

$$\mu_j/\nu_j = \xi_j \quad \text{and} \quad \rho_j \mu_j \neq \eta_j \nu_j. \quad (2.3)$$

The first property assures that we build a rational Krylov space having the poles $\{\xi_j\}_{j=1}^m$, cf. (2.2a), while the second forbids a deficient case where $(\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)$ is a scalar multiple of the identity matrix. Our choice is $(\nu_j, \mu_j) = (1, \xi_j)$ for finite ξ_j , and $(\nu_j, \mu_j) = (0, 1)$ when $\xi_j = \infty$, whilst $(\rho_j, \eta_j) = (0, 1)$ for $|\xi_j| < 1$ and $(\rho_j, \eta_j) = (1, 0)$ otherwise.

Collecting the vectors $\underline{\mathbf{k}}_j$, appropriately padded with zeros to be of size $m+1$, in a matrix \underline{K}_m and accordingly for \underline{H}_m , we arrive at the decomposition (2.1).

3. The RKFIT algorithm. The RKFIT algorithm aims to find rational functions $\{r^{[j]}\}_{j=1}^\ell$ of type $(m+k, m)$, all sharing a common denominator $q_m \in \mathcal{P}_m$, solving problem (1.5). As the denominator q_m is not known and hence (1.5) is nonlinear, RKFIT tries to iteratively improve a starting guess for q_m by solving a linearised problem at each iteration.

RKFIT is succinctly described in Algorithm 3.2. Different from the basic version presented in [5], our new description of RKFIT 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. By $P_{\mathcal{T}}$ we denote the orthogonal projection onto \mathcal{T} . The essence of Algorithm 3.2 is the relocation of poles in line 7. Since with any polynomial $\check{q}_m \in \mathcal{P}_m$ we can associate a vector $\mathbf{v} = \check{q}_m(A)q_m(A)^{-1}\mathbf{b} \in \mathcal{S}$, and the other way around, we may identify \check{q}_m , the improvement of q_m , by looking for the corresponding vector $\mathbf{v} \in \mathcal{S}$. This is indeed done in line 5 and further explained in Section 3.2. Corollary 3.2, a consequence of the following Theorem 3.1, provides insight into the RKFIT pole relocation, i.e., lines 5–7 in Algorithm 3.2. We use $\ell = 1$ and $D^{[1]} = I$ for simplicity only. The result easily extends to $\ell > 1$ and arbitrary nonsingular $D^{[j]}$.

THEOREM 3.1. *Let $q_m^*, \check{q}_m^* \in \mathcal{P}_m$ 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 $\mathbf{b} \in \mathbb{C}^N$ be such that $2m+k < M(A, \mathbf{b})$. Assume that $F = p_{m+k}^*(A)q_m^*(A)^{-1}$ for some $p_{m+k}^* \in \mathcal{P}_{m+k}$. Define \mathcal{S} and \mathcal{T} as in lines 3 and 4 of Algorithm 3.2, respectively, and let V_{m+1} be an orthonormal basis of \mathcal{S} . The matrix $(I - P_{\mathcal{T}})FV_{m+1}$ has a nullspace of dimension $\Delta m + 1$ if and only if Δm is the largest integer such that p_{m+k}^*/q_m^* is of type $(m+k-\Delta m, m-\Delta m)$.*

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

$$F\check{\mathbf{v}} = p_{m+k}^*(A)q_m^*(A)^{-1}\check{p}_m(A)q_m(A)^{-1}\mathbf{b} =: p_{2m+k}(A)q_m^*(A)^{-1}q_m(A)^{-1}\mathbf{b}$$

Algorithm 3.2 High-level description of RKFIT.	RK Toolbox [4]: rkfit
1. Take initial guess for q_m .	
2. repeat	
3. Set search space $\mathcal{S} := \mathcal{Q}_{m+1}(A, \mathbf{b}, q_m)$.	See section 2.
4. Set target space $\mathcal{T} := \mathcal{K}_{m+k+1}(A, q_m(A)^{-1}\mathbf{b})$.	See section 3.1.
5. Find $\mathbf{v} = \operatorname{argmin}_{\substack{\check{\mathbf{v}} \in \mathcal{S} \\ \ \check{\mathbf{v}}\ _2=1}} \sum_{j=1}^{\ell} \ D^{[j]}(I - P_{\mathcal{T}})F^{[j]}\check{\mathbf{v}}\ _2^2$.	See section 3.2.
6. Let $\check{q}_m \in \mathcal{P}_m$ be such that $\mathbf{v} = \check{q}_m(A)q_m(A)^{-1}\mathbf{b}$.	See section 3.2.
7. Set $q_m := \check{q}_m$.	See Corollary 3.2.
8. until stopping criteria is satisfied.	See sections 3.2–3.3.
9. Construct wanted approximants $\{r^{[j]}\}_{j=1}^{\ell}$.	See section 3.3.

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

$$\mathcal{N} := \left\{ p_{\Delta m}(A)q_{m-\Delta m}^*(A)q_m(A)^{-1}\mathbf{b} \mid p_{\Delta m} \in \mathcal{P}_{\Delta m} \right\} \subseteq \mathcal{S}, \quad (3.1)$$

whose elements $\check{\mathbf{v}}$ are such that $F\check{\mathbf{v}} \in \mathcal{T}$. Hence, $\Delta m + 1$ is the dimension of the nullspace of $(I - P_{\mathcal{T}})FV_{m+1}$. \square

COROLLARY 3.2. *Let $q_m, q_m^*, F, A, \mathbf{b}, m, k, \mathcal{S}$, and \mathcal{T} be as in Theorem 3.1. Then p_{m+k}^* and q_m^* are coprime and either $\deg(p_{m+k}^*) = m+k$ or $\deg(q_m^*) = m$ if and only if there is a unique, up to unimodular scaling, solution to $\|(I - P_{\mathcal{T}})F\check{\mathbf{v}}\|_2^2 \rightarrow \min$, over all $\check{\mathbf{v}} \in \mathcal{S}$ of unit 2-norm. This solution is given by $\mathbf{v} = \gamma q_m^*(A)q_m(A)^{-1}\mathbf{b}$ with some scaling factor $\gamma \in \mathbb{C}$.*

The corollary asserts that if $F\mathbf{b} = p_{m+k}(A)q_m^*(A)^{-1}\mathbf{b}$ and $\Delta m = 0$, then the “roots” of $\mathbf{v} = \gamma q_m^*(A)q_m(A)^{-1}\mathbf{b}$ match the unknown poles q_m^* and the next approximate poles become $q_m := q_m^*$. 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. Based on Theorem 3.1 we develop in section 4 an automatic procedure to reduce the denominator degree m by Δm , and adapt k . When dealing with noisy data (and roundoff) or when $F\mathbf{b}$ cannot be exactly represented as $r(A)\mathbf{b}$ for a rational function r of type $(m+k, m)$, the convergence of RKFIT is not understood.

In the remaining part of this section we show how the various parts of Algorithm 3.2 are realized using rational Krylov techniques.

3.1. Constructing the target space. We assume (2.1) has been constructed. Hence for the search space we have $\mathcal{S} = \mathcal{R}(V_{m+1})$. If $k = 0$, then $\mathcal{S} = \mathcal{T}$ and $P_{\mathcal{T}} = V_{m+1}V_{m+1}^*$. Otherwise \mathcal{S} either has to be expanded (if $k > 0$) or compressed (if $k < 0$) to get \mathcal{T} .

Let us first consider superdiagonal approximants, i.e., $k > 0$. In this case $\mathcal{T} = \mathcal{Q}_{m+k+1}(A, \mathbf{b}, q_m)$, 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 $\mathcal{Q}_{m+k+1}(A, \mathbf{b}, q_m)$, we expand (2.1) to $AV_{m+k+1}\overline{K_{m+k}} = V_{m+k+1}\overline{H_{m+k}}$ by performing k additional polynomial steps with the rational Krylov method. Let us,

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

We now consider the subdiagonal case $k < 0$. The target space \mathcal{T} is given by $\mathcal{T} = \mathcal{K}_{m+k+1}(A, q_m(A)^{-1} \mathbf{b})$. Note that $\mathcal{Q}_{m+1}(A, \mathbf{b}, q_m) = \mathcal{K}_{m+1}(A, q_m(A)^{-1} \mathbf{b})$. Therefore, we aim at transforming the RAD (2.1) for $\mathcal{Q}_{m+1}(A, \mathbf{b}, q_m)$ into a RAD

$$A \check{V}_{m+1} \check{K}_m = \check{V}_{m+1} \check{H}_m, \quad (3.2)$$

for $\mathcal{K}_{m+1}(A, q_m(A)^{-1} \mathbf{b})$. An orthonormal basis for \mathcal{T} is then given by truncating \check{V}_{m+1} to \check{V}_{m+k+1} , the first $m+k+1$ columns of \check{V}_{m+1} .

Using a sequence of Givens rotations in a QZ fashion (as explained in [38, p. 495]) we get unitary $Q_{m+1} \in \mathbb{C}^{(m+1) \times (m+1)}$ and $Z_m \in \mathbb{C}^{m \times m}$ such that $\check{K}_m = Q_{m+1}^* \underline{K}_m Z_m$ is upper-triangular and $\check{H}_m = Q_{m+1}^* \underline{H}_m Z_m$ is upper-Hessenberg. Fittingly, the poles $\check{h}_{j+1,j} / \check{k}_{j+1,j}$ of (3.2), where $\check{V}_{m+1} = V_{m+1} Q_{m+1}$, are all at infinity. Hence $\mathcal{R}(\check{V}_{j+1}) = \mathcal{K}_{j+1}(A, q_m(A)^{-1} \mathbf{b})$ for $j = 0, 1, \dots, m$, and we can set $P_{\mathcal{T}} = \check{V}_{m+k+1} \check{V}_{m+k+1}^*$.

3.2. Solving the linear problem and relocating poles. We consider the problem in line 5 of Algorithm 3.2, i.e., the problem of finding a unit 2-norm vector $\mathbf{v} \in \mathcal{S}$ such that $\sum_{j=1}^{\ell} \|D^{[j]}(I - P_{\mathcal{T}})F^{[j]}\mathbf{v}\|_2^2$ is minimal. We have $\mathcal{S} = \mathcal{R}(V_{m+1})$ and $\mathcal{T} = \mathcal{R}(\check{V}_{m+k+1})$ with both V_{m+1} and \check{V}_{m+k+1} being orthonormal.

Defining the matrix $S = [S_1^T \ \dots \ S_{\ell}^T]^T \in \mathbb{C}^{N \times (m+1)}$, where

$$S_j = D^{[j]} \left[F^{[j]} V_{m+1} - \check{V}_{m+k+1} \left(\check{V}_{m+k+1}^* F^{[j]} V_{m+1} \right) \right] \in \mathbb{C}^{N \times (m+1)}, \quad (3.3)$$

a solution is given by $\mathbf{v} = V_{m+1} \mathbf{c}$, where \mathbf{c} is a right singular vector of S corresponding to a smallest singular value σ_{\min} .

We now show how to determine the polynomial $\check{q}_m \in \mathcal{P}_m$, from line 6 of Algorithm 3.2, such that $\mathbf{v} = V_{m+1} \mathbf{c} = \check{q}_m(A) q_m(A)^{-1} \mathbf{b}$. As this part is independent of k , the procedure is the one presented in [5, Section 5], and we review it only briefly.

Let $Q_{m+1} \in \mathbb{C}^{(m+1) \times (m+1)}$ be unitary with first column $Q_{m+1} \mathbf{e}_1 = \mathbf{c}$. Using (2.1) as a RAD for \mathcal{S} , it follows from [5, Theorem 4.4] that the roots of \check{q}_m are the eigenvalues of the $m \times m$ pencil

$$([\mathbf{0} \ I_m] Q_{m+1}^* \underline{H}_m, [\mathbf{0} \ I_m] Q_{m+1}^* \underline{K}_m). \quad (3.4)$$

Note that if $\mathbf{c} = \mathbf{e}_1$, the first canonical unit vector, then \mathbf{v} is collinear with \mathbf{b} . In this case \check{q}_m and q_m share the same roots and the algorithm stagnates.

3.3. Constructing the approximants. The approximants $\{r^{[j]}\}_{j=1}^{\ell}$ of type $(m+k, m)$ are computed by LS approximation of $F^{[j]} \mathbf{b}$ from the rational Krylov space $\mathcal{Q}_{\hat{m}+1}(A, \mathbf{b})$, $\hat{m} = \max\{m, m+k\}$, with poles ξ_1, \dots, ξ_m identified by RKFIT and $\max\{0, k\}$ poles at ∞ . More precisely, let $V_{\hat{m}+1}$ be an orthonormal basis of $\mathcal{Q}_{\hat{m}+1}(A, \mathbf{b})$ with associated RAD

$$A V_{\hat{m}+1} \underline{K}_{\hat{m}} = V_{\hat{m}+1} \underline{H}_{\hat{m}}. \quad (3.5)$$

Each $r^{[j]}$ is represented by a coefficient vector $\mathbf{c}^{[j]} \in \mathbb{C}^{\hat{m}+1}$ such that $r^{[j]}(A) \mathbf{b} / \|\mathbf{b}\|_2 = V_{\hat{m}+1} \mathbf{c}^{[j]}$. The coefficient vectors are given by

$$\mathbf{c}^{[j]} = \begin{cases} V_{\hat{m}+1}^* (F^{[j]} \mathbf{b}) / \|\mathbf{b}\|_2 & \text{if } D^{[j]} = I_N, \\ (D^{[j]} V_{\hat{m}+1})^\dagger (D^{[j]} F^{[j]} \mathbf{b}) / \|\mathbf{b}\|_2 & \text{otherwise,} \end{cases} \quad (3.6)$$

with X^\dagger denoting the Moore–Penrose pseudoinverse of X .

Computing the coefficients $\mathbf{c}^{[j]}$ at each iteration does not significantly increase the computational complexity because the $D^{[j]}F^{[j]}\mathbf{b}$ need to be computed only once. The coefficients $\mathbf{c}^{[j]}$ enable the cheap evaluation of the relative misfit (1.5), which allows to stop the RKFIT iteration when a wanted tolerance $\varepsilon_{\tau_{\text{ol}}}$ is achieved.

3.4. Avoiding complex arithmetic. If $F^{[j]}$, A , and \mathbf{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 Algorithm 2.1, see [35]. The matrices S_j in (3.3) are guaranteed to be real-valued and the generalized eigenproblem (3.4) is real-valued as well. Apart from requiring less memory and making the computation faster, this also guarantees the relocated poles to appear in complex-conjugate pairs.

If $F^{[j]}$, A , and \mathbf{b} are not real-valued, but can be simultaneously transformed into real-valued form, complex arithmetic can be avoided too. We show how to achieve this for scalar data, assuming $\ell = 1$ for simplicity. Let the data set $\{(\lambda_i, f_i)\}_{i=1}^N$ be closed under complex conjugation, i.e., if (λ, f) is in the set, so is $(\bar{\lambda}, \bar{f})$. Without loss of generality, we assume that the pairs are ordered such that $\{(\lambda_i, f_i)\}_{i=1}^{N_{\mathbb{R}}} \subset \mathbb{R}^2$ and $\{(\lambda_i, f_i)\}_{i=N_{\mathbb{R}}+1}^N \subset \mathbb{C}^2 \setminus \mathbb{R}^2$, where the complex-conjugate pairs in the latter subset appear next to each other, and $0 \leq N_{\mathbb{R}} \leq N$ is a natural number. Define $A = \text{diag}(\lambda_1, \dots, \lambda_N)$, $F = \text{diag}(f_1, \dots, f_N)$, $\mathbf{b} = [1 \ \dots \ 1]^T \in \mathbb{R}^N$, and let $Q \in \mathbb{C}^{N \times N}$ be unitary. Then

$$\begin{aligned} \|F\mathbf{v} - r(A)\mathbf{b}\|_2 &= \|(QFQ^*)(Q\mathbf{b}) - Qr(A)Q^*(Q\mathbf{b})\|_2 \\ &= \|(QFQ^*)(Q\mathbf{b}) - r(QAQ^*)(Q\mathbf{b})\|_2. \end{aligned}$$

The first equality follows from the unitary invariance of the 2-norm, and the second from [26, Theorem 1.13]. With the particular choice

$$Q = \text{blkdiag} \left(I_{N_{\mathbb{R}}}, \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix}, \dots, \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix} \right) \in \mathbb{C}^{N \times N},$$

we have $F_{\mathbb{R}} = QFQ^* \in \mathbb{R}^{N \times N}$, $A_{\mathbb{R}} = QAQ^* \in \mathbb{R}^{N \times N}$ and $\mathbf{b}_{\mathbb{R}} = Q\mathbf{b} \in \mathbb{R}^N$. Precisely,

$$F_{\mathbb{R}} = \text{blkdiag} \left(f_1, \dots, f_{N_{\mathbb{R}}}, \begin{bmatrix} \Re(f_{i_1}) & -\Im(f_{i_1}) \\ \Im(f_{i_1}) & \Re(f_{i_1}) \end{bmatrix}, \dots, \begin{bmatrix} \Re(f_{i_{N_{\mathbb{C}}})} & -\Im(f_{i_{N_{\mathbb{C}}})} \\ \Im(f_{i_{N_{\mathbb{C}}})} & \Re(f_{i_{N_{\mathbb{C}}})} \end{bmatrix} \right),$$

where $N_{\mathbb{C}} = \frac{N - N_{\mathbb{R}}}{2}$ and $i_j = N_{\mathbb{R}} + 2j - 1$. For $A_{\mathbb{R}}$ we obtain an analogous expression, while $\mathbf{b}_{\mathbb{R}} = [1 \ \dots \ 1 \ \sqrt{2} \ 0 \ \dots \ \sqrt{2} \ 0]^T$, with $N_{\mathbb{R}}$ entries equal to 1, and $N_{\mathbb{C}}$ consecutive $[\sqrt{2} \ 0]^T$ copies.

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

$$\mathbf{D}^{[j]} = I_n \otimes D^{[j]}, \quad \mathbf{F}^{[j]} = I_n \otimes F^{[j]}, \quad \text{and} \quad \mathbf{A} = I_n \otimes A, \quad (3.7)$$

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

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

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

Our implementation [4] supports the case $n > 1$, and takes advantage of the structure present in (3.7) so that only $\{D^{[j]}, F^{[j]}\}_{j=1}^{\ell}$ and A are stored, while $\mathbf{D}^{[j]}, \mathbf{F}^{[j]}$, and \mathbf{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 $D^{[j]}\mathbf{x}, F^{[j]}\mathbf{x}, A\mathbf{x}$ for arbitrary $\mathbf{x} \in \mathbb{C}^N$, as well as the ability to solve shifted linear systems $(A - \xi I)\mathbf{x} = \mathbf{v}$.

4. 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.

For ease of notation we consider the case $\ell = 1$, $n = 1$, and $D = I$. We assume to have at hand an $(m+k, m)$ approximant r such that $\|F\mathbf{b} - r(A)\mathbf{b}\|_2 \leq \|F\mathbf{b}\|_2 \varepsilon_{\tau_01}$. 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 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. An illustration is given in Figure 4.1. In subsection 4.4 we discuss the case when F is not a rational matrix function.

4.1. Reducing the denominator degree m . Assume that F is a rational matrix function. Our reduction procedure for m is based on Theorem 3.1, which asserts that a defect $\Delta m + 1$ of the matrix $S = (I - P_{\mathcal{T}})FV_{m+1}$ 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 $\|F\mathbf{b}\|_2 \varepsilon_{\tau_01}$ (with, e.g., $\varepsilon_{\tau_01} = 10^{-15}$) is computed. More precisely, we reduce m by the largest integer $\Delta m \leq \min\{m, m+k\}$ such that

$$\sigma_{m+1-\Delta m} \leq \|F\mathbf{b}\|_2 \varepsilon_{\tau_01}, \quad (4.1)$$

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

4.2. 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 Theorem 3.1, the $\Delta m + 1$ linearly independent vectors spanning \mathcal{N} all share as the greatest common divisor (GCD) the polynomial $q_{m-\Delta m}^*$, 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 $(\underline{H}_m, \underline{K}_m)$ in (2.1).

THEOREM 4.1. *Let (2.1) be a RAD for $\mathcal{Q}_{m+1}(A, \mathbf{b}, q_m)$, with $m+1 < M(A, \mathbf{b})$, and let the $r_j \equiv V_{m+1}\mathbf{c}_j$ for $j = 1, \dots, \Delta m + 1$ 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 $X\mathbf{e}_j = \mathbf{c}_j$ for $j = 1, \dots, \Delta m + 1$. Introduce*

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

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

Proof. We transform the RAD (2.1) into $A\check{V}_{m+1}\check{K}_m = \check{V}_{m+1}\check{H}_m$, where $\check{V}_{m+1} = V_{m+1}X$, $\check{K}_m = X^{-1}\underline{K}_m Y$, and $\check{H}_m = X^{-1}\underline{H}_m Y$, and with $Y = \text{blkdiag}(I_{\Delta m}, K_{\star})^{-1}$.

Hence in scalar form, for all $z \in \mathbb{C}$ we have

$$z\mathbf{r}(z)\check{\underline{K}}_m = \mathbf{r}(z)\check{\underline{H}}_m \iff \mathbf{r}(z)(z\check{\underline{K}}_m - \check{\underline{H}}_m) = \mathbf{0}^T,$$

where $\mathbf{r}(z) = [r_1(z) \ \dots \ r_{\Delta m+1}(z) \ r_{\Delta m+2}(z) \ \dots \ r_{m+1}(z)]$. Introduce $\check{\underline{K}}_\star$ and $\check{\underline{H}}_\star$ as the lower-right $(m - \Delta m) \times (m - \Delta m)$ submatrices of $\check{\underline{K}}_m$ and $\check{\underline{H}}_m$, respectively. Since $\Lambda(H_\star, K_\star) = \Lambda(\check{\underline{H}}_\star, \check{\underline{K}}_\star)$, we need to show that the roots of the GCD are $\Lambda(\check{\underline{H}}_\star, \check{\underline{K}}_\star)$.

Let λ be a common root of $\{r_j\}_{j=1}^{\Delta m+1}$. Then the last $m - \Delta m$ columns of $\mathbf{r}(\lambda)(\lambda\check{\underline{K}}_m - \check{\underline{H}}_m) = \mathbf{0}^T$ assert that λ is a generalized eigenvalue of $(\check{\underline{H}}_\star, \check{\underline{K}}_\star)$ with left eigenvector $\mathbf{r}_\star(\lambda)^\ast = [r_{\Delta m+2}(\lambda) \ \dots \ r_{m+1}(\lambda)]^\ast \neq \mathbf{0}$. This handles simple roots.

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

$$\mathbf{r}'(\lambda)(\lambda\check{\underline{K}}_m - \check{\underline{H}}_m) + \mathbf{r}(\lambda)\check{\underline{K}}_m = \mathbf{0}^T \iff \mathbf{r}'(\lambda)(\lambda\check{\underline{K}}_m - \check{\underline{H}}_m) = -\mathbf{r}(\lambda)\check{\underline{K}}_m.$$

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

$$\mathbf{r}'_\star(\lambda)(\lambda\check{\underline{K}}_\star - \check{\underline{H}}_\star) = -\mathbf{r}_\star(\lambda)\check{\underline{K}}_\star = -\mathbf{r}_\star(\lambda) \neq \mathbf{0}^T.$$

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

REMARK 4.2. *The assumption that K_\star is nonsingular is used in the proof of Theorem 4.1 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 $m - \Delta m$.*

4.3. 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 $\|F\mathbf{b} - r(A)\mathbf{b}\|_2 \leq \|F\mathbf{b}\|_2 \varepsilon_{\text{tol}}$ has been found. Reducing the numerator degree is a linear problem and we can guarantee the misfit to stay below ε_{tol} after the reduction.

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

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

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

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

$$\|\mathbf{c}_{-i}\|_2 \leq \|F\mathbf{b}\|_2 \varepsilon_{\text{tol}} - \|F\mathbf{b} - r(A)\mathbf{b}\|_2, \quad (4.2)$$

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

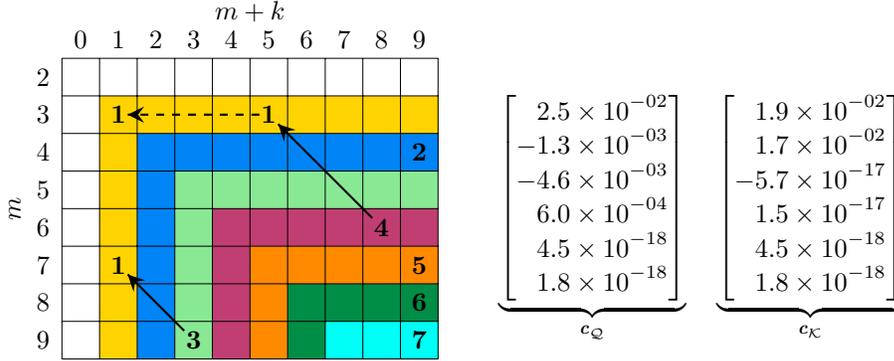


Fig. 4.1: Degree reduction when fitting $F\mathbf{b}$, where $F = A(A + I)^{-1}(A + 3I)^{-2}$, $A = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}$, and $\mathbf{b} = \mathbf{e}_1 \in \mathbb{R}^N$, with $N = 150$. Note that F is of type $(1, 3)$. The initial poles of the search space are all at infinity. The table on the left shows the number $\Delta m + 1$ of singular values of $(I - P_{\mathcal{T}})FV_{m+1}$ below the tolerance $\|F\mathbf{b}\|_2 \varepsilon_{\text{tol}} = 10^{-15}$, for different choices of m and k . For the choice $(m+k, m) = (3, 9)$ 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 $-3 \pm i2.32 \times 10^{-7}$. With these three poles, and other two at infinity, the type $(5, 3)$ approximant r produces a relative misfit 7.02×10^{-17} . The expansion coefficients $\mathbf{c}_{\mathcal{Q}}$ of r in the orthonormal rational basis are given 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 4.3, reveals that r is of type $(1, 3)$. The coefficients $\mathbf{c}_{\mathcal{K}}$ in this polynomial basis are also given.

4.4. General F . The following lemma extends Theorem 3.1 to the case when F is not necessarily a rational matrix function.

LEMMA 4.3. Let $q_m, A, \mathbf{b}, m, k, \mathcal{S}, \mathcal{T}$, and V_{m+1} be as in Theorem 3.1. 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 $\|F\mathbf{b} - r(A)\mathbf{b}\|_2 \leq \|F\mathbf{b}\|_2 \varepsilon_{\text{tol}}$. If the matrix $(I - P_{\mathcal{T}})FV_{m+1}$ has $\Delta m + 1$ singular values smaller than $\|F\mathbf{b}\|_2 \varepsilon_{\text{tol}}$, then there exists a $(\Delta m + 1)$ -dimensional subspace $\mathcal{N}_g \subseteq \mathcal{S}$, containing \mathbf{b} , such that

$$\min_{\mathbf{v} \in \mathcal{P}_{m+k}} \|F\mathbf{v} - p(A)q_m(A)^{-1}\mathbf{b}\|_2 \leq \|F\mathbf{b}\|_2 \varepsilon_{\text{tol}}$$

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

Proof. Consider a thin SVD of the matrix $(I - P_{\mathcal{T}})FV_{m+1} = U\Sigma W^*$, where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{m+1}) \in \mathbb{R}^{(m+1) \times (m+1)}$ and $\sigma_{m+1} \leq \dots \leq \sigma_{m-\Delta m} \leq \|F\mathbf{b}\|_2 \varepsilon_{\text{tol}}$ by assumption. Equivalently, $(I - P_{\mathcal{T}})FV_{m+1}W = U\Sigma$. Then the final $\Delta m + 1$ columns of $V_{m+1}W$ form a basis for \mathcal{N}_g . It follows from the assumption $\|F\mathbf{b} - r(A)\mathbf{b}\|_2 \leq \|F\mathbf{b}\|_2 \varepsilon_{\text{tol}}$ that $\mathbf{b} \in \mathcal{N}_g$. \square

Recall that if F is a rational matrix function, then the space \mathcal{N}_g defined in Lemma 4.3 corresponds to the exact nullspace $\mathcal{N} = \mathcal{K}_{\Delta m+1}(A, q_{m-\Delta m}^*(A)q_m(A)^{-1}\mathbf{b})$ defined in (3.1), where the (numerators of the) rational functions share as GCD the polynomial $q_{m-\Delta m}^*$. 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

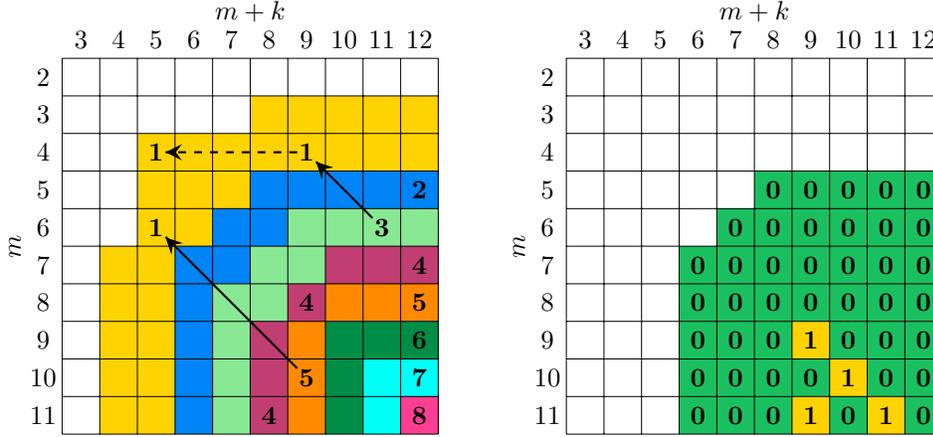


Fig. 4.2: Degree reduction when fitting $F\mathbf{b}$, where $F = \sqrt{A + A^2}$, $A = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}$, and $\mathbf{b} = \mathbf{e}_1 \in \mathbb{R}^N$, with $N = 150$. The poles of the search space are obtained after three RKFIT iterations with initial poles all at infinity. The table on the left shows the number $\Delta m + 1$ of singular values of $(I - P_T)FV_{m+1}$ below $\|F\mathbf{b}\|_{2\varepsilon_{\text{tol}}\varepsilon_{\text{safe}}} = 10^{-5}$ 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 $\varepsilon_{\text{tol}} = 10^{-4}$, using the approximate GCD strategy for selecting the poles to restart RKFIT with.

a common denominator. However, for every $\mathbf{v} = p_m(A)q_m(A)^{-1}\mathbf{b} \in \mathcal{N}_g$ the vector $Fp_m(A)q_m(A)^{-1}\mathbf{b}$ is well approximated in the 2-norm by some vector $p(A)q_m(A)^{-1}\mathbf{b}$, with $p \in \mathcal{P}_{m+k}$. This suggests that the polynomials p_m corresponding to vectors $\mathbf{v} \in \mathcal{N}_g$ share an *approximate* GCD (see, e.g., [7]) whose roots approximate the poles of a “good” rational approximation $r(A)\mathbf{b}$ for $F\mathbf{b}$. We therefore propose to use the same reduction procedure as suggested by Theorem 4.1.

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

$$\sigma_{m+1-\Delta m} \leq \|F\mathbf{b}\|_{2\varepsilon_{\text{tol}}\varepsilon_{\text{safe}}}, \quad (4.3)$$

where $\sigma_1 \leq \dots \leq \sigma_{m+1}$ are the singular values of S . By default we use $\varepsilon_{\text{safe}} = 0.1$. Figure 4.2 illustrates our reduction strategy for a non-rational function F .

5. Working with rational functions in pencil form. After the RKFIT algorithm has terminated, a rational function r of type $(m+k, m)$ is represented by the pencil $(\underline{H}_{\widehat{m}}, \underline{K}_{\widehat{m}})$, $\widehat{m} = \max\{m, m+k\}$, satisfying (3.5) and the coefficients \mathbf{c} in (3.6). We now show how to perform computations with this representation.

5.1. Evaluating the approximants and their derivatives. Our approach to evaluating a rational function r is based on the scalar form of (3.5), that is

$$z[r_1(z) \quad \dots \quad r_{\widehat{m}+1}(z)]\underline{K}_{\widehat{m}} = [r_1(z) \quad \dots \quad r_{\widehat{m}+1}(z)]\underline{H}_{\widehat{m}},$$

Algorithm 5.3 Rerunning the rational Krylov method. RK Toolbox [4]: `rat_krylov`

Input: $\widehat{A} \in \mathbb{C}^{\widehat{N} \times \widehat{N}}$, $\widehat{\mathbf{b}} \in \mathbb{C}^{\widehat{N}}$, and an upper-Hessenberg pencil $(\underline{H}_{\widehat{m}}, \underline{K}_{\widehat{m}})$.

Output: $W_{\widehat{m}+1} \in \mathbb{C}^{\widehat{N} \times (\widehat{m}+1)}$ such that $\widehat{A}W_{\widehat{m}+1}\underline{K}_{\widehat{m}} = W_{\widehat{m}+1}\underline{H}_{\widehat{m}}$ and $W_{\widehat{m}+1}\mathbf{e}_1 = \widehat{\mathbf{b}}$.

1. **for** $j = 1, \dots, \widehat{m}$ **do**
 2. Define $\xi_j := h_{j+1,j}/k_{j+1,j}$, and take any $(\nu_j, \mu_j, \rho_j, \eta_j)$ satisfying (2.3).
 3. Set $\mathbf{t}_j := \mu_j \mathbf{k}_j - \nu_j \mathbf{h}_j \in \mathbb{C}^j$, and $\underline{\mathbf{y}}_j := \eta_j \underline{\mathbf{k}}_j - \rho_j \underline{\mathbf{h}}_j \in \mathbb{C}^{j+1}$. See (2.2b), (2.2c).
 4. Compute $\mathbf{w} := (\nu_j \widehat{A} - \mu_j I)^{-1} (\rho_j \widehat{A} - \eta_j I) W_j \mathbf{t}_j$.
 5. Compute $\mathbf{w}_{j+1} := (\mathbf{w} - W_j \underline{\mathbf{y}}_j) / y_{j+1,j}$.
 6. **end for**
-

with the rational functions r_j satisfying $r_j(A)\mathbf{b}/\|\mathbf{b}\|_2 = \mathbf{v}_j$ for all columns \mathbf{v}_j of $V_{\widehat{m}+1}$.

Given a matrix $\widehat{A} \in \mathbb{C}^{\widehat{N} \times \widehat{N}}$ and a starting vector $\widehat{\mathbf{b}} = \mathbf{w}_1 \in \mathbb{C}^{\widehat{N}}$, one easily verifies that the vectors $\mathbf{w}_j = r_j(\widehat{A})\widehat{\mathbf{b}}$ satisfy the Arnoldi-type relation

$$\widehat{A}W_{\widehat{m}+1}\underline{K}_{\widehat{m}} = W_{\widehat{m}+1}\underline{H}_{\widehat{m}}, \quad (5.1)$$

provided that all $r_j(\widehat{A})$ are defined, i.e., $\Lambda(\widehat{A})$ does not contain any of the poles ξ_1, \dots, ξ_m . By rerunning the rational Krylov method, we can therefore evaluate each rational basis function $r_j(\widehat{A})\widehat{\mathbf{b}}$ for any matrix argument \widehat{A} and vector $\widehat{\mathbf{b}}$.

Recalling that $r(z) = \sum_{j=1}^{\widehat{m}+1} c_j r_j(z)$ with the coefficients $\mathbf{c} = [c_j]$ defined in (3.6), we readily evaluate $r(\widehat{A})\widehat{\mathbf{b}} = \sum_{j=1}^{\widehat{m}+1} c_j r_j(\widehat{A})\widehat{\mathbf{b}}$. In particular, choosing $\widehat{A} = \text{diag}(\lambda_1, \dots, \lambda_{\widehat{N}})$ and $\widehat{\mathbf{b}} = [1 \ \dots \ 1]^T$, we can evaluate r pointwise as $r(\widehat{A})\widehat{\mathbf{b}} = [r(\lambda_1) \ \dots \ r(\lambda_{\widehat{N}})]^T$. Derivatives of r can easily be evaluated by using a Jordan block for \widehat{A} . For example, if $\widehat{A} = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}$ and $\widehat{\mathbf{b}} = [0 \ 1]^T$, then $r(\widehat{A})\widehat{\mathbf{b}} = [r'(\lambda) \ r(\lambda)]^T$.

Our method for rerunning the rational Krylov method and thereby generating (5.1) is given in Algorithm 5.3 and implemented in the `rat_krylov` function of the Rational Krylov Toolbox [4].

5.2. Root-finding. For finding the roots of r we recall that $r(A)\mathbf{b}/\|\mathbf{b}\|_2 = V_{\widehat{m}+1}\mathbf{c} = p_{\widehat{m}}(A)q_m(A)^{-1}\mathbf{b}$. Let us assume that $\mathbf{c} \neq \mathbf{e}_1$, otherwise $r(A)\mathbf{b} = c_1\mathbf{b}$, i.e., r has no roots. Define $P = I_{m+1} - 2\mathbf{u}\mathbf{u}^*$, where $\mathbf{u} = (\gamma\mathbf{c} - \mathbf{e}_1)/\|\gamma\mathbf{c} - \mathbf{e}_1\|_2$ and $\gamma \in \mathbb{C}$ is a unimodular scalar such that $\gamma\mathbf{e}_1^*\mathbf{c}$ is real and nonnegative. It follows from [5, Theorem 4.4] that the roots of $p_{\widehat{m}}$ are the eigenvalues of the $\widehat{m} \times \widehat{m}$ pencil

$$([\mathbf{0} \ I_{\widehat{m}}] P \underline{H}_{\widehat{m}}, [\mathbf{0} \ I_{\widehat{m}}] P \underline{K}_{\widehat{m}}).$$

If $k < 0$, then among the \widehat{m} eigenvalues there are $-k$ infinite eigenvalues, or numerically, eigenvalues of large modulus. In our implementation `roots` of the Rational Krylov Toolbox [4] we hence sort the roots by their magnitudes and return only the $m + k$ smallest ones.

5.3. Conversion to partial fraction form. The conversion of a type $(m + k, m)$ rational function r into partial fraction form can be achieved by transforming the rational Arnoldi decomposition (3.5) in such a way that it reveals the residues. Here we only consider the case $k \leq 0$, i.e., $\widehat{m} = m$, and pairwise distinct finite poles

Algorithm 5.4 Conversion to partial fraction form. RK Toolbox [4]: **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 unitary $L_{m+1} = \text{blkdiag}(1, Q_m^*)$, where $[\mathbf{0} \quad I_m] \underline{H}_m Q_m = Q_m \text{diag}(\xi_1, \dots, \xi_m)$.
 3. Update $R_m := R_m Q_m$, $\underline{H}_m := L_{m+1} \underline{H}_m Q_m$, and $\underline{K}_m := L_{m+1} \underline{K}_m Q_m$.
 4. Introduce $D_{m+1} = [-e_1 \quad \underline{K}_m]$.
 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 $R_m := R_m D_m$, $\underline{H}_m := \underline{H}_m D_m$, $\underline{K}_m := \underline{K}_m D_m$, where $D_m = \text{diag}(1/h_{1j})$.
 7. Redefine $D_m := \text{diag}(1/k_{j+1,j})$, and $D_{m+1} := \text{blkdiag}(1, D_m)$.
 8. Update $L_{m+1} := D_{m+1} L_{m+1}$, $\underline{H}_m := D_{m+1} \underline{H}_m$, and $\underline{K}_m := D_{m+1} \underline{K}_m$.
-

ξ_1, \dots, ξ_m (generalizations are discussed in section 7). We aim to transform (3.5) into

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

where $W_{m+1} e_1 = v_1$. One then easily verifies that the columns of W_{m+1} satisfy $w_{j+1} = (A - \xi_j)^{-1} v_1$. This conversion is achieved via left- and right-multiplication of the pencil $(\underline{H}_m, \underline{K}_m)$ by invertible matrices as indicated in Algorithm 5.4.

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 $r(A)\mathbf{b} = \|\mathbf{b}\|_2 V_{m+1} \mathbf{c}_{m+1}$ in the basis V_{m+1} , we arrive at the partial fraction expansion

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

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

The transformation of V_{m+1} to the partial fraction basis W_{m+1} has condition number $\text{cond}(L_{m+1})$, which can be arbitrarily bad in particular if some of the poles ξ_j are close together. Our implementation **residue** in the Rational Krylov Toolbox [4] 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 Rational Krylov Toolbox [4], among other examples (including those in [5]).

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 [10]. 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 [10], appearing in complex-conjugate pairs on the range $\pm i[10^{-2}, 10^3]$. 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 3.4. 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 $\varepsilon_{\text{tol}} = 10^{-3}$ 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 $\Re(\xi) \leq 0$, 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 taken to be all infinite, and the misfit at iteration 0 corresponds to these initial poles. Both RKFIT variants achieve a misfit below ε_{tol} at iteration 4, after which the degree reduction discussed in section 4 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 error corresponding to the new 56 (respectively 54) poles corresponds to iteration 5; as it is still below ε_{tol} no further RKFIT iterations are required.

For the second experiment we compare RKFIT with the vector fitting code VFIT [12, 22, 24] 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 A.) 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 $-\xi/100 \pm i\xi$, 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 ε_{tol} so that no further iterations were required. This validates our approximate GCD strategy for choosing the poles after reducing m .

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

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

at several time points t_1, \dots, t_ℓ . 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 $\mathbf{u}(t_j)$ are given as $\mathbf{u}(t_j) = \exp(-t_j K^{-1} L) \mathbf{u}_0$, and a popular approach for approximating $\mathbf{u}(t_j)$ is to use rational functions $r^{[j]}$ of the form

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

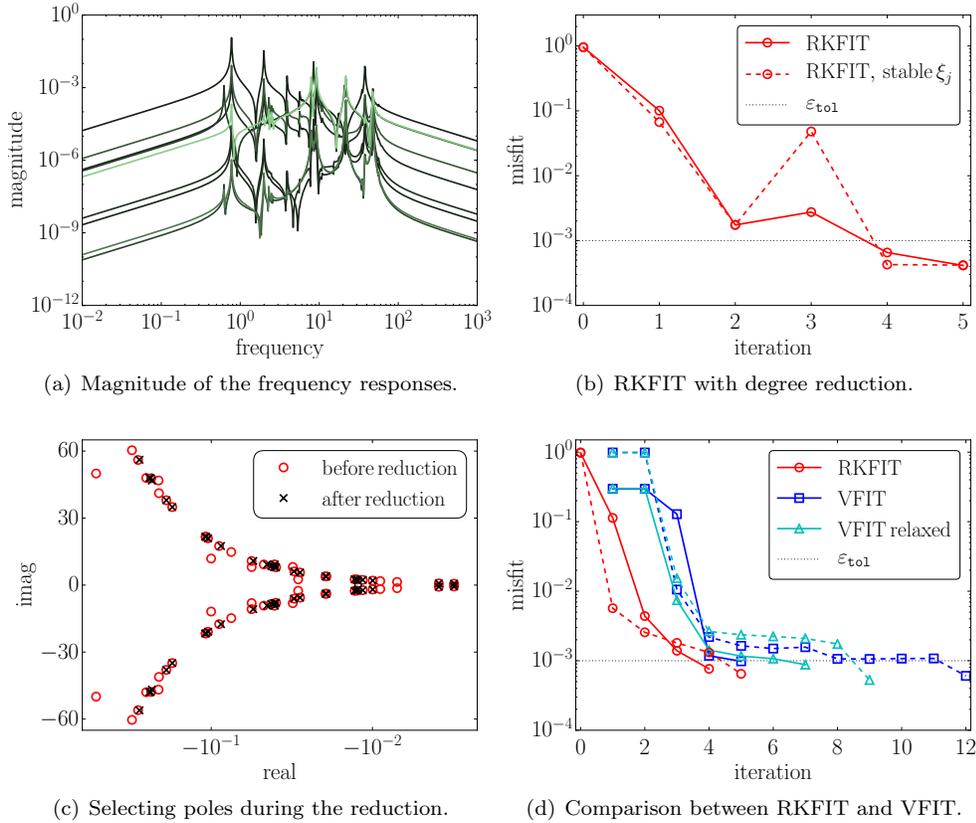


Fig. 6.1: Low-order model approximation to the MIMO system ISS from [10]. The frequency responses are plotted in figure (a). In (b) the progress of RKFIT is given for $m = 70$ infinite starting poles. At iteration 4 the degree reduction takes place. The 70 poles after convergence and 54 selected ones (for the case when stability of poles is 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 $r^{[j]}(K^{-1}L)\mathbf{u}_0 \approx \mathbf{u}(t_j)$. Note that the poles of $r^{[j]}$ do not depend on t_j and we have

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

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 [33, 31, 40, 16, 8]. Let us assume that L is Hermitian positive semi-definite, K is Hermitian positive definite, and define the vector K -norm

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

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

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

In order to use RKFIT for finding poles ξ_1, \dots, ξ_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 [8]. Let $A = \text{diag}(\lambda_1, \dots, \lambda_N)$ be a diagonal matrix with ‘‘sufficiently dense’’ eigenvalues on $\lambda \geq 0$. 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 $\mathbf{b} = [1 \ \dots \ 1]^T$ 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

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

is minimized. Note that

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

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 $\lambda \geq 0$ one can expect the upper error bound (6.1) to be small.

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 [40, Figure 6.1] by approximating $\mathbf{u}(t_j) = \exp(-t_j L) \mathbf{u}_0$ 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 \mathbf{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 $\|\mathbf{u}(t_j) - r^{[j]}(L) \mathbf{u}_0\|_2$ for each time point t_j (solid curve with circles), together with the approximate upper error bound $\|\exp(-t_j A) \mathbf{b} - r^{[j]}(A) \mathbf{b}\|_{\infty}$ (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 [40]. Applying an m -point quadrature rule to the Cauchy integral

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

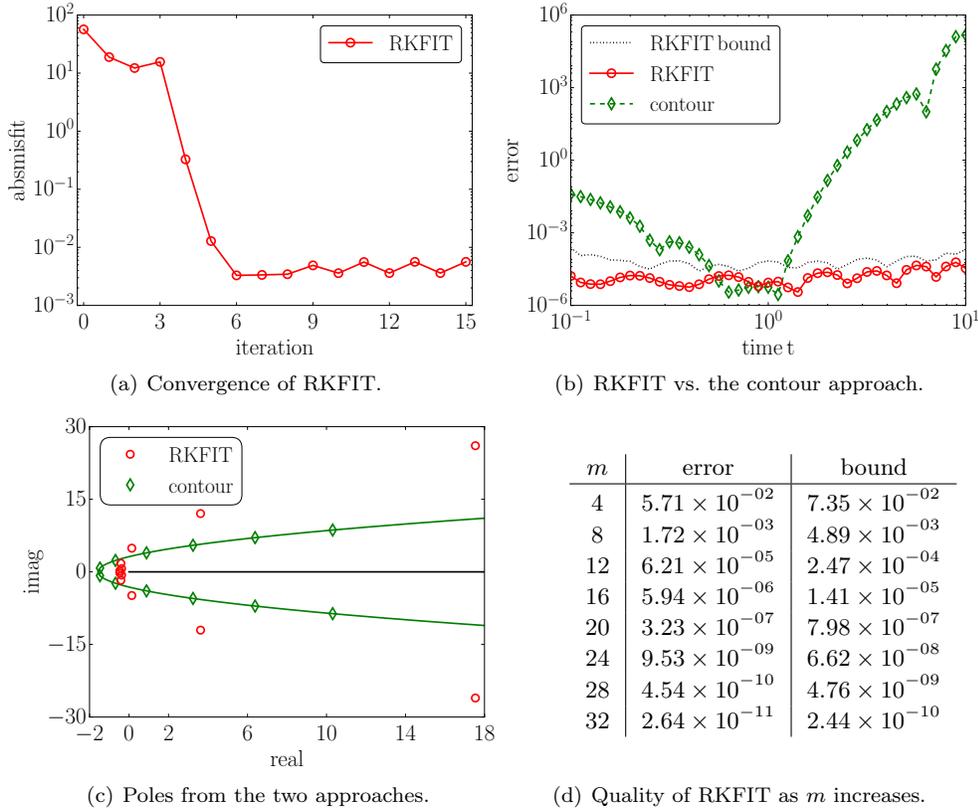


Fig. 6.2: Approximating $\exp(-tL)\mathbf{u}_0$ for a range of parameters t with rational approximants sharing common poles. The convergence behaviour of RKFIT, for approximants of type (11, 12), is shown in (a). In (b) we show the approximation error for $\ell = 41$ logspaced time points $t \in [0.1, 10]$ 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 poles of the two families of rational approximants. The table in (d) shows the maximal RKFIT error and bound, uniformly over all t_j , for various m .

on a contour Γ enclosing the positive real axis, one obtains a family of rational functions $\tilde{r}^{[j]}$ 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 [40], such quadrature-based approximants tend to be good only for a small range of parameters t_j . In Figure 6.2(b) we see that the error $\|\mathbf{u}(t_j) - \tilde{r}^{[j]}(L)\mathbf{u}_0\|_2$ 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 $\tilde{r}^{[j]}$, which all lie on a parabolic contour [40, 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., [16, 8]). 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 in time, 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 [7]. 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 [28].

The automated degree reduction opens the possibility for “Chebfun-like computing” [13] 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.

Another interesting problem is the extension of RKFIT to rational block-Krylov spaces, with the potential of solving tangential interpolation problems (see, e.g., [17]). Further potential applications include the construction of rational filter functions or the computation of perfectly matched layers.

Acknowledgments. We would like to thank Vladimir Druskin, Zlatko Drmač, Serkan Gugercin, and Marc Van Barel for stimulating discussions.

Appendix A. Relations to other rational approximation algorithms.

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

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

A popular approach for solving problems of this form introduced in [24] and designed to fit frequency response measurements of dynamical systems is vector fitting (VFIT).

As already observed in [5], 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 non-linear problem (A.1) was through linearisation [30]. 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 [30] 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 data 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 [39] suggested to replace the nonlinear problem (A.1) with a sequence of linear problems. Once the linearised problem $\sum_{i=1}^N |f_i q_m(\lambda_i) - p_{m-1}(\lambda_i)|^2 \rightarrow \min$ has been solved, one can set $\check{q}_m := q_m$ and solve a reweighed linear problem $\sum_{i=1}^N \frac{|f_i q_m(\lambda_i) - p_{m-1}(\lambda_i)|^2}{|\check{q}_m(\lambda_i)|^2} \rightarrow \min$. This process can be iterated until a satisfactory approximation has been obtained or a maximal number of iterations has been performed.

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; see below.

A.2. Vector fitting. 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 $\{\xi_j\}_{j=1}^m$, that is,

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

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

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

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

$$\begin{bmatrix} \frac{1}{\lambda_1 - \xi_1} & \cdots & \frac{1}{\lambda_1 - \xi_m} & \frac{-f_1}{\lambda_1 - \xi_1} & \cdots & \frac{-f_1}{\lambda_1 - \xi_m} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{1}{\lambda_N - \xi_1} & \cdots & \frac{1}{\lambda_N - \xi_m} & \frac{-f_N}{\lambda_N - \xi_1} & \cdots & \frac{-f_N}{\lambda_N - \xi_m} \end{bmatrix} \begin{bmatrix} \varphi \\ \psi \end{bmatrix} = \mathbf{f}, \quad (\text{A.4})$$

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 reweighing 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

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

and $\check{C}_m = C_{m+1} [\mathbf{0} \quad I_m]^T$. We now rewrite (A.4) in the equivalent form

$$[\check{C}_m \quad -FC_{m+1}] \begin{bmatrix} \varphi \\ \psi_0 \\ \psi \end{bmatrix} = \mathbf{0}, \quad (\text{A.5})$$

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

Therefore, when solving (A.4) in VFIT one gets $r = \frac{\check{p}_m/q_m}{\check{q}_m/q_m}$, where $\check{q}_m(z)/q_m(z) = 1 + \sum_{j=1}^m \frac{\psi_j}{z - \xi_j}$ and $\check{p}_m(z)/q_m(z) = \sum_{j=1}^m \frac{\varphi_j}{z - \xi_j}$ is the projection of $f\check{q}_m/q_m$, with f being defined on the discrete set of interpolation nodes as $f(\lambda_i) = f_i$, onto the target space, here represented by \check{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 is the constraints under which the LS problems are solved. The constraint in VFIT is for \check{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 interpolation nodes λ_i vary over a large scale of magnitudes. This was observed in [22], 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, $\|\mathbf{v}\|_2 = \|\check{q}(A)q(A)^{-1}\mathbf{b}\|_2 = 1$, cf. line 5 in Algorithm 3.2.

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, c.f. (A.4). *Orthonormal vector fitting* was proposed as a remedy in [11], where the basis of partial fractions was replaced by an orthonormal basis. Soon after it was claimed [23] that a numerically more careful implementation of VFIT is as good as the orthonormal VFIT variant proposed in [11], and hence the orthonormal VFIT never became a reality.

Numerical issues arising in VFIT have been recently discussed and tackled in [14, 15]. Our approach avoids these problems.

The problem with the orthonormal VFIT [11] 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.

So far we assumed the interpolation nodes λ_i to be given. If they can be chosen freely, one can choose them based on quadrature rules tailored to the application. This may improve both the numerical aspects as well as the approximation. This is suggested in [14, 15] for the discretized \mathcal{H}_2 approximation of transfer function measurements and carries over straightforwardly to RKFIT.

As to date, there are no comprehensive convergence analyses for VFIT and RKFIT. In [29, Section IV] an example of degree $m = 2$ was constructed where the VFIT fixed-point iteration is repellent and hence diverges, independently of the starting guess for the poles. Despite this one example, VFIT has been and is being successfully used by the IEEE community for various (large scale) problems. Both VFIT and RKFIT have the property that if a rational function r of sufficiently low degree and there are sufficiently many interpolation nodes, then in the absence of roundoff r is recovered exactly, see [29, Corollary III.1] and our Corollary 3.2.

REFERENCES

- [1] *Multiprecision Computing Toolbox*. Advanpix, Tokyo. <http://www.advanpix.com>.
- [2] A. C. ANTOLAS, 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. BERLJAJA 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://guettel.com/rktoolbox/>.
- [5] M. BERLJAJA AND S. GÜTTEL, *Generalized rational Krylov decompositions with an application to rational approximation*, SIAM J. Matrix Anal. Appl., (2015). To appear.
- [6] H. BLINCHIKOFF AND A. ZVEREV, *Filtering in the Time and Frequency Domains*, John Wiley & Sons Inc., New York, 1976.
- [7] P. BOITO, *Structured Matrix Based Methods for Approximate Polynomial GCD*, vol. 15, Springer Science & Business Media, 2012.
- [8] R.-U. BÖRNER, O. G. ERNST, AND S. GÜTTEL, *Three-dimensional transient electromagnetic modeling using rational Krylov methods*, Geophys. J. Int., (2015). To appear.
- [9] D. BRAESS, *Nonlinear Approximation Theory*, Berlin, Germany, 1986.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] T. A. DRISCOLL, N. HALE, AND L. N. TREFETHEN, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.
- [14] 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.
- [15] Z. DRMAČ, S. GUGERCIN, AND C. BEATTIE, *Vector fitting for matrix-valued rational approximation*, 2015. arXiv:1503.00411.
- [16] 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.
- [17] 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.
 - [18] K. GALLIVAN, E. GRIMME, AND P. VAN DOOREN, *A rational Lanczos algorithm for model reduction*, Numer. Algorithms, 12 (1996), pp. 33–63.
 - [19] P. GONNET, S. GÜTTEL, AND L. N. TREFETHEN, *Robust Padé approximation via SVD*, SIAM Rev., 55 (2013), pp. 101–117.
 - [20] P. GONNET, R. PACHÓN, AND L. N. TREFETHEN, *Robust rational interpolation and least-squares*, Electron. Trans. Numer. Anal., 38 (2011), pp. 146–167.
 - [21] S. GÜGERCIN, A. ANTOULAS, AND C. BEATTIE, *A rational Krylov iteration for optimal H2 model reduction*, in Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006, pp. 1665–1667.
 - [22] B. GUSTAVSEN, *Improving the pole relocating properties of vector fitting*, IEEE Trans. Power Del., 21 (2006), pp. 1587–1592.
 - [23] 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.
 - [24] B. GUSTAVSEN AND A. SEMLYEN, *Rational approximation of frequency domain responses by vector fitting*, IEEE Trans. Power Del., 14 (1999), pp. 1052–1061.
 - [25] S. GÜTTEL, *Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection*, GAMM-Mitt., 36 (2013), pp. 8–31.
 - [26] N. J. HIGHAM, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, PA, USA, 2008.
 - [27] 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.
 - [28] B. KÄGSTRÖM 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.
 - [29] S. LEFTERIU AND A. ANTOULAS, *On the convergence of the vector-fitting algorithm*, IEEE Trans. Microw. Theory Techn., 61 (2013), pp. 1435–1443.
 - [30] E. C. LEVY, *Complex-curve fitting*, IRE Trans. Autom. Control, AC-4 (1959), pp. 37–43.
 - [31] I. MORET AND P. NOVATI, *RD-rational approximations of the matrix exponential*, BIT, 44 (2004), pp. 595–615.
 - [32] 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.
 - [33] S. P. NØRSETT, *Restricted Padé approximations to the exponential function*, SIAM J. Numer. Anal., 15 (1978), pp. 1008–1029.
 - [34] A. RUHE, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.
 - [35] A. RUHE, *The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices*, BIT, 34 (1994), pp. 165–176.
 - [36] A. RUHE, *Rational Krylov algorithms for nonsymmetric eigenvalue problems. II. Matrix pairs*, Linear Algebra Appl., 198 (1994), pp. 283–295.
 - [37] A. RUHE, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551.
 - [38] 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.
 - [39] C. SANATHANAN AND J. KOERNER, *Transfer function synthesis as a ratio of two complex polynomials*, IEEE Trans. Automat. Control, 8 (1963), pp. 56–58.
 - [40] L. N. TREFETHEN, J. A. C. WEIDEMAN, AND T. SCHMELZER, *Talbot quadratures and rational approximations*, BIT, 46 (2006), pp. 653–670.
 - [41] G. WANNER, E. HAIRER, AND S. NØRSETT, *Order stars and stability theorems*, BIT, 18 (1978), pp. 475–489.