Rational Krylov Decompositions: Theory and Applications

Berljafa, Mario

2017

MIMS EPrint: **2017.6**

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

RATIONAL KRYLOV DECOMPOSITIONS: THEORY AND APPLICATIONS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science & Engineering

2017

Mario Berljafa
School of Mathematics

Contents

Word count 43500

List of Tables

List of Figures

List of Algorithms

List of RKToolbox Examples

The University of Manchester

Mario Berljafa
Doctor of Philosophy
Rational Krylov Decompositions: Theory and Applications
January 9, 2017

Numerical methods based on rational Krylov spaces have become an indispensable tool of scientific computing. In this thesis we study rational Krylov spaces by considering rational Krylov decompositions; matrix relations which, under certain conditions, are associated with these spaces. We investigate the algebraic properties of such decompositions and present an implicit Q theorem for rational Krylov spaces.

We derive standard and harmonic Ritz extraction strategies for approximating the eigenpairs of a matrix and for approximating the action of a matrix function onto a vector. While these topics have been considered previously, our approach does not require the last pole to be infinite, which makes the extraction procedure computationally more efficient.

Typically, the computationally most expensive component of the rational Arnoldi algorithm for computing a rational Krylov basis is the solution of a large linear system of equations at each iteration. We explore the option of solving several linear systems simultaneously, thus constructing the rational Krylov basis in parallel. If this is not done carefully, the basis being orthogonalized may become poorly conditioned, leading to numerical instabilities in the orthogonalization process. We introduce the new concept of continuation pairs which gives rise to a near-optimal parallelization strategy that allows to control the growth of the condition number of this nonorthogonal basis. As a consequence we obtain a more accurate and reliable parallel rational Arnoldi algorithm. The computational benefits are illustrated using our high performance C++ implementation.

We develop an iterative algorithm for solving nonlinear rational least squares problems. The difficulty is in finding the poles of a rational function. For this purpose, at each iteration a rational Krylov decomposition is constructed and a modified linear problem is solved in order to relocate the poles to new ones. Our numerical results indicate that the algorithm, called RKFIT, is well suited for model order reduction of linear time invariant dynamical systems and for optimisation problems related to exponential integration. Furthermore, we derive a strategy for the degree reduction of the approximant obtained by RKFIT. The rational function obtained by RKFIT is represented with the aid of a scalar rational Krylov decomposition and an additional coefficient vector. A function represented in this form is called an RKFUN. We develop efficient methods for the evaluation, pole and root finding, and for performing basic arithmetic operations with RKFUNs.

Lastly, we discuss RKToolbox, a rational Krylov toolbox for MATLAB, which implements all our algorithms and is freely available from `http://rktoolbox.org`. RKToolbox also features an extensive guide and a growing number of examples. In particular, most of our numerical experiments are easily reproducible by downloading the toolbox and running the corresponding example files in MATLAB.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Publications

- The material in Sections 2.2–2.3, Section 5.1, and part of the material in Sections 6.1–6.2 is based on the paper:

  [10] 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. (This paper won a 2016 SIAM Student paper prize.)

- Chapter 4 is based on the paper:

  [12] M. Berljafa and S. Güttel, *Parallelization of the rational Arnoldi algorithm*, MIMS EPrint 2016.32, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2016. Submitted for publication.

- Chapter 6 is based on the paper:

  [11] M. Berljafa and S. Güttel, *The RKFIT algorithm for nonlinear rational approximation*, MIMS EPrint 2015.38, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2015. Submitted for publication.

- The *RKToolbox corner* sections in Chapters 2, 5–7 are, in part, based on the technical report:

  [9] 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. (Last updated September 2015.)

- Sections 2.4–2.5, Chapter 3, Section 5.2 and Chapter 7 (review existing and) present results not included in the above list.

Acknowledgements

I sincerely thank my supervisor Stefan Güttel for his guidance and unrelenting patience through these 3 years. I also acknowledge Françoise Tisseur for her useful advice throughout the years. Finally, I wish to thank Massimiliano Fasi and Ana Šušnjara as well as the examiners Bernhard Beckermann and Nicholas Higham for their valuable comments and corrections which significantly improved this thesis.

1 Introduction & background

1.1 Introduction

Published in 1984, Axel Ruhe's paper "Rational Krylov sequence methods for eigenvalue computation" presents "a new class of algorithms which is based on rational functions of the matrix" [86]. In fact, what the author does is essentially to suggest replacing, within iterative eigenvalues algorithms, the space $\text{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}$, where $A \in \mathbb{C}^{N,N}$ and $\boldsymbol{b} \in \mathbb{C}^N$, with the more general space $\text{span}\{\psi_1(A)\boldsymbol{b}, \psi_2(A)\boldsymbol{b}, \ldots, \psi_m(A)\boldsymbol{b}\}$, where $\psi_1, \psi_2, \ldots, \psi_m$ are arbitrary functions. He soon realises that "besides polynomials, which we have treated, the only feasible choice computationally is rational functions." The paper obtains almost no attention in the following decade, and this lack of interest is probably due to two main factors. First, the paper reports no numerical experiments, and, thus, competitiveness and reach of the method remain unclear. Moreover, adequate guidance for choosing the *best* or at least *good* rational functions was not provided; this second problem remains an active area of current (and future) research. Fortunately, Ruhe himself reconsiders the method and his subsequent work [87, 88, 89] published in 1994 lay the foundation for the theory of rational Krylov methods as we know it today. His initial investigation of the topic converges in the 1998 paper [90], and by that time other researchers have started contributing to the theory and application of rational Krylov methods; see, e.g., [24, 37, 73].

Originally devised for the solution of large sparse eigenvalue problems, these methods have proved themselves a key tool for an increasing number of applications over the last two decades. Examples of rational Krylov applications can be found in model

order reduction [34, 37, 49, 51, 71], computation of the action of matrix functions on vectors [7, 31, 33, 35, 40, 56], solution of matrix equations [8, 32, 75], nonlinear eigenvalue problems [59, 67, 91, 109], and nonlinear rational least squares fitting [10, 11]. The use of rational functions is justified by their approximation properties, which are often superior to linear schemes such as polynomial interpolation, in particular when approximating functions near singularities or on unbounded regions of the complex plane; see, e.g., [18, 105].

Computationally, the most costly part of rational Krylov methods is the solution of shifted linear systems of the form $(A - \xi_j I)\boldsymbol{x}_j = \boldsymbol{b}_j$ for $\boldsymbol{x}_j$, for many indices j, where the matrix A, and vectors $\boldsymbol{b}_j$ are given (I denotes the identity matrix). The parameters $\xi_j \in \overline{\mathbb{C}}$ are called *poles* of the *rational Krylov space*, and the success of rational Krylov methods heavily depends on their choice. If *good* poles are available, using just a few of them may suffice to solve the problem at hand. Otherwise, the solution of a large number of shifted linear systems may be needed, rendering the process computationally unfeasible. Finding good pole parameters is highly non-trivial and problem-dependent. Despite the large number of applications, rational Krylov methods are not yet fully understood. One of our main contributions is the development of a new theory of rational Arnoldi decompositions, which provides a better understanding of rational Krylov spaces, and ultimately allows rational Krylov methods themselves to be used, in an inverse manner, to find *near-optimal* pole parameters in certain applications.

The rational Arnoldi algorithm used to construct an orthonormal basis for a rational Krylov space with a matrix A leads to a decomposition of the form

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

called *rational Arnoldi decomposition* (RAD). The range $\mathcal{R}(V_{m+1})$ of V_{m+1} spans the rational Krylov space in question. We provide a better understanding of rational Krylov spaces and the interplay of their defining parameters (starting vector $\boldsymbol{b}$ and poles ξ_j) by studying such, and related, decompositions. Specifically, in Chapter 2 we describe the complete set of RADs associated with rational Krylov spaces, and present a new *rational implicit Q theorem* about the uniqueness of RADs. In practice, the rational implicit Q theorem is useful as it allows for certain transformations of RADs to be performed at a reduced computational cost. Such transformations consist of two steps. First, the transformation is applied to the reduced pencil $(\underline{H_m}, \underline{K_m})$, instead

of the operator A, and second, the RAD structure is recovered and reinterpreted. Concrete examples and applications are discussed in Chapters 5–6. Furthermore, we consider the variant of the rational Arnoldi algorithm for real-valued matrices with complex-conjugate poles which constructs real-valued decompositions of a form similar to RADs [87]. The presentation of [87] is extended and formalised and an implicit Q theorem for the obtained *quasi-RADs* is proposed. Finally, we discuss decompositions of the form $AV_{m+1}\underline{K_m} = BV_{m+1}\underline{H_m}$ which correspond to rational Krylov spaces related to a matrix pencil (A, B) instead of a single matrix A. In particular, we show how to reduce them to RADs so that the established theory can be transferred directly. The use of nonstandard inner products is included in Chapter 2 as well.

In Chapter 3 we review known strategies, based on projections, for extracting information from RADs, and develop new ones, highlighting their potential benefit. Specifically, for an RAD of the form $AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}$, one can, for instance, approximate some of the eigenvalues of A with some of the eigenvalues of the smaller matrix $V_{m+1}^\dagger AV_{m+1}$, while $f(A)\boldsymbol{b}$ may be approximated by $V_{m+1}f(V_{m+1}^\dagger AV_{m+1})V_{m+1}^\dagger\boldsymbol{b}$, which requires the computation of the potentially much smaller matrix function $f(V_{m+1}^\dagger AV_{m+1})$. Forming the projected matrix $V_{m+1}^\dagger AV_{m+1}$ at each iteration m of the rational Arnoldi algorithm may, however, be computationally too expensive. If V_{m+1} is orthonormal and the mth pole $\xi_m = \infty$ is infinity, then the last row of $\underline{K_m}$ is zero. Consequently, the RAD reduces to $AV_mK_m = V_{m+1}\underline{H_m}$, and thus $V_m^\dagger AV_m = V_m^* AV_m = H_mK_m^{-1}$, which allows us to bypass the explicit projection $V_m^\dagger AV_m$. As this is applicable only when the last, mth, pole is infinite, the authors in [58] have considered adding and removing a temporary infinite pole after each iteration of the rational Arnoldi algorithm. We suggest new formulas that do not depend in such a manner on the poles. For instance, we show that $f(A)\boldsymbol{b}$ may be approximated as $(V_{m+1}\underline{K_m})f(\underline{K_m^\dagger}\underline{H_m})(V_{m+1}\underline{K_m})^\dagger\boldsymbol{b}$, independently of any of the poles or their order of appearance.

Rational functions can be decomposed into partial fractions, and this simple property makes rational Krylov methods highly parallelisable; several basis vectors spanning the rational Krylov space can be computed at once. Unfortunately, the basis constructed in this way may easily become ill-conditioned [98, 99]. Chapter 4 is devoted to the study of the influence of internal parameters when constructing an RAD in order to

monitor the condition number of the basis. We also provide a high performance C++ implementation which shows the benefits of the parallelisation.

Finally, in Chapter 6 we consider the problem of approximating, in a least squares sense, $f(A)\boldsymbol{b}$ as $r(A)\boldsymbol{b}$, where r is a rational function. This is a nonlinear optimisation problem, since the poles of r are unknown. We propose an iterative algorithm, called *rational Krylov fitting* (RKFIT) for its solution. At each iteration an RAD is constructed and a modified linear problem is solved in order to relocate the poles of r to new (hopefully better) ones. The relocation of poles itself is studied in Chapter 5, and it is based on the rational implicit Q theorem. This theoretical observations lead to the notion of *rational Krylov decompositions*, which are a more general class of decompositions than RADs, and, from a practical point of view, they allow us to monitor the various transformation arising in the RKFIT algorithm. A distinct feature of our RKFIT algorithm is the degree reduction strategy which allows for further fine tuning once a solution r is obtained. We test RKFIT for model order reduction and exponential integration problems and show that the new approach is superior to some existing methods. The rational function r obtained by RKFIT is represented with the aid of a *scalar RAD* and an additional coefficient vector. A function represented in this form is called a *rational Krylov function* (RKFUN). In Chapter 7 we show how to use RKFUNs in order to, for instance, evaluate $r(z)$ or perform basic arithmetic operations.

Alongside our theoretical contribution, we discuss *RKToolbox*, a rational Krylov toolbox for MATLAB, which implements all our algorithms and is freely available for download from `http://rktoolbox.org`; see also [9]. The main features of the toolbox are the `rat_krylov` and `rkfit` functions and the RKFUN class. The function `rat_krylov`, for instance, provides a flexible implementation of the rational Arnoldi algorithm. There are 18 different ways to call `rat_krylov`, and furthermore, several parameters can be adjusted. Typing `help rat_krylov` in MATLAB command line provides all the details. RKToolbox also features a large collection of utility functions, basic unit testing, an extensive guide and a growing number of examples. In particular, most of our numerical experiments are easily reproducible by downloading the toolbox and running the corresponding example files in MATLAB. The usage of the main features of the toolbox is explained in the *RKToolbox corner* sections which conclude

most of the forthcoming chapters.

In the remainder of the chapter we review standard results from (numerical) linear algebra needed for our developments. General results are considered in Section 1.2, while in Section 1.3 we focus on polynomial Krylov methods.

1.2 Background material

In this section we review some of the fundamental definitions and matrix properties that we use through the thesis, others are introduced when needed. We stress that this is a brief review, and refer the interested reader to [42, 60, 64, 100] for a thorough discussion on these topics.

Matrices and vectors. We shall often denote matrices with uppercase Latin letters while for their elements we shall use the corresponding lowercase Latin letters with indices indicating the row and column they reside in. For instance,

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \in \mathbb{C}^{N,N}.$$

With A^T we denote the *transpose* of A, i.e., the matrix whose element on the position (i, j) is the (j, i) element a_{ji} of A. Analogously, with $A^* = \overline{A}^T$ we denote the *conjugate transpose* of A, where $\overline{A}$ denotes element-wise conjugation. With

$$I_N = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \equiv \mathrm{diag}(1, 1, \dots, 1)$$

we denote the *identity matrix*. The subscript N may be removed if the dimension of the matrix is clear from the context. The kth column of I_N is denoted by $\boldsymbol{e}_k$, and referred to as a *canonical vector*. With 0 we shall denote a zero matrix of any size, while for vectors only we may also use $\mathbf{0}$.

We say that a square matrix $A \in \mathbb{C}^{N,N}$ is *upper* (*lower*) *triangular* if $a_{ij} = 0$ ($a_{ji} = 0$), whenever $i > j$. A triangular matrix A is called *strictly triangular* if $a_{jj} = 0$ for all j. If A is both upper and lower triangular, we say that it is a *diagonal matrix*.

On the other hand, we say that a rectangular matrix $A \in \mathbb{C}^{N,M}$ is *upper* (*lower*) *trapezoidal* if $a_{ij} = 0$ ($a_{ji} = 0$), whenever $i > j$.

Eigenvalues and eigenvectors. Let $A \in \mathbb{C}^{N,N}$. If $(\lambda, \mathbf{0} \neq \boldsymbol{x}) \in \mathbb{C} \times \mathbb{C}^N$ satisfies

$$A\boldsymbol{x} = \lambda\boldsymbol{x}, \tag{1.1}$$

then λ is called an *eigenvalue* of A and $\boldsymbol{x}$ its corresponding *eigenvector*. Any matrix $A \in \mathbb{C}^{N,N}$ has N eigenvalues, not necessarily mutually distinct, and they are the zeros of the *characteristic polynomial* $\chi_A(z) = \det(A - \lambda I)$ of A. Here, $\det : \mathbb{C}^{N,N} \to \mathbb{C}$ denotes the *determinant* of the matrix; see, e.g., [64, p. 8]. We denote the set containing all the eigenvalues of A by

$$\Lambda(A) = \{z \in \mathbb{C} | \det(A - \lambda I) = 0\}.$$

The matrix A can be expressed in the *Jordan canonical form*

$$\begin{aligned} &Z^{-1}AZ = J = \operatorname{diag}(J_1, J_2, \ldots, J_\ell), \quad \text{with} \\ &J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{n_k, n_k}, \end{aligned} \tag{1.2}$$

where Z is nonsingular, λ_k are the eigenvalues of A, and $n_1 + n_2 + \ldots + n_\ell = N$. The matrix J_k is called a *Jordan block*. The Jordan canonical form is typically useful from a theoretical viewpoint. Since the Jordan form is not continuous and is thus numerically unstable, when designing numerical algorithms one usually resorts to the so called *Schur form* $Q^*AQ = T$, where $Q \in \mathbb{C}^{N,N}$ is a *unitary matrix* and T is upper triangular. A matrix $Q \in \mathbb{C}^{N,N}$ is called unitary if $QQ^* = I$. Note that $\Lambda(A) = \{t_{jj}\}_{j=1}^N$, and Q can be chosen so that the elements on the diagonal of T appear in any order.

Generalised eigenvalues and eigenvectors. Let $A, B \in \mathbb{C}^{N,N}$. The pair (A, B) is called a *pencil*. If $(\lambda, \mathbf{0} \neq \boldsymbol{x}) \in \mathbb{C} \times \mathbb{C}^N$ satisfies the equation

$$A\boldsymbol{x} = \lambda B\boldsymbol{x}, \tag{1.3}$$

then λ is called a *generalised eigenvalue* of (A, B) and $\boldsymbol{x}$ its corresponding *generalised eigenvector*. The set of all generalised eigenvalues of (A, B) is denoted by

$$\Lambda(A, B) = \{z \in \mathbb{C} | \det(A - zB) = 0\}.$$

Clearly, $\Lambda(A, I) = \Lambda(A)$. The analogue of the Schur form for a matrix to pencils is the *generalised Schur form* $(T, S) = (Q^*AZ, Q^*BZ)$, where $Q, Z \in \mathbb{C}^{N,N}$ are unitary and $T, S \in \mathbb{C}^{N,N}$ are upper triangular. If for some j, t_{jj} and s_{jj} are both zero, then $\Lambda(A, B) = \mathbb{C}$. Otherwise, we have $\Lambda(A, B) = \{t_{jj}/s_{jj} | s_{jj} \neq 0\}$.

When $A, B \in \mathbb{R}^{N,N}$, the *generalised real Schur form* $(T, S) = (Q^*AZ, Q^*BZ)$, where $Q, Z \in \mathbb{R}^{N,N}$ are *orthogonal*, T is *upper quasi-triangular* and S is upper triangular, may be of interest instead of the generalised Schur form. A matrix $Q \in \mathbb{R}^{N,N}$ is said to be *orthogonal* if $QQ^T = I$, while $T = [T_{ij}] \in \mathbb{R}^{N,N}$ is said to be upper quasi-triangular if it is block upper triangular and T_{jj} are either of size 1-by-1 or of size 2-by-2.

Functions of matrices. Let $A \in \mathbb{C}^{N,N}$ have the Jordan canonical form (1.2). We say that the function f is *defined on the spectrum of* A if the values

$$f^{(j)}(\lambda_k), \quad j = 0, 1, \ldots, n_k - 1, \quad k = 1, 2, \ldots, \ell$$

exist. If f is defined on the spectrum of A, then

$$f(A) := Zf(J)Z^{-1} = Z \operatorname{diag}(f(J_1), f(J_2), \ldots, f(J_\ell))Z^{-1},$$

where

$$f(J_k) = \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \ldots & \frac{f^{(n_k-1)}(\lambda_k)}{(n_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix} \in \mathbb{C}^{n_k, n_k}.$$

There exist other, equivalent, definitions of $f(A)$. For our purposes, we state the definition of $f(A)$ related to *Hermite interpolation*. Note that the *minimal polynomial* of A is defined as the unique monic polynomial ψ of lowest degree such that $\psi(A) = 0$. By considering the Jordan canonical form (1.2) we can see that

$$\psi(z) = \prod_{j=1}^{s} (z - \lambda_j)^{\nu_j}, \tag{1.4}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_s$ are the distinct eigenvalues of A and ν_j is the dimension of the largest Jordan block where λ_j appears. Finally, if f is defined on the spectrum of A, and (1.4) is the minimal polynomial of A, then

$$f(A) := p(A),$$

where p is the unique polynomial of degree less then $\deg \psi$ such that

$$f^{(j)}(\lambda_k) = p^{(j)}(\lambda_k), \quad j = 0, 1, \ldots, \nu_k - 1, \quad k = 1, 2, \ldots, s.$$

The polynomial p is called the *Hermite interpolating polynomial*. For a proof of equivalence between the two definitions see, e.g., [60, Theorem 1.12].

LU factorisation. If zero is not an eigenvalue of $A \in \mathbb{C}^{N,N}$, then A is said to be *nonsingular* and there exists a unique matrix $A^{-1} \in \mathbb{C}^{N,N}$ such that $AA^{-1} = A^{-1}A = I$. The matrix A^{-1} is called the *inverse* of A. A common task in numerical linear algebra is to solve a *linear system of equation* $A\boldsymbol{x} = \boldsymbol{b}$, where A is nonsingular and $\boldsymbol{b} \in \mathbb{C}^N$ is a given vector. The sought-after vector $\boldsymbol{x} \in \mathbb{C}^N$ is given by $\boldsymbol{x} = A^{-1}\boldsymbol{b}$, and can be computed by forming the *LU factorisation* $A = LU$ of A, if it exists. Here, L is a *unit* lower triangular matrix, i.e., it is lower triangular with all diagonal elements being equal to one. The matrix U is upper triangular. In practise, the factorisation $PA = LU$, where P is a *permutation* matrix, that is, an orthogonal matrix with elements being equal to either zero or one, is more often used, since it always exists if A is nonsingular, and, moreover, it enjoys better numerical properties. If $PA = LU$, then $\boldsymbol{x} = U^{-1}[L^{-1}(P\boldsymbol{b})]$ can be formed by permuting the elements of $\boldsymbol{b}$, followed by *forward substitution* and then *back substitution*; see, e.g., [42, Section 3.1].

QR factorisation. Let $A \in \mathbb{C}^{N,M}$. The factorisation $A = QR$, with unitary $Q \in \mathbb{C}^{N,N}$ and upper trapezoidal $R \in \mathbb{C}^{N,M}$ is called the *QR factorisation* of A. If $N > M$ and $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ with $Q_1 \in \mathbb{C}^{N,M}$, then Q_1 is called an *orthonormal* matrix. If, furthermore, $R = \begin{bmatrix} R_1^T & 0 \end{bmatrix}^T$ with $R_1 \in \mathbb{C}^{M,M}$, then $A = Q_1 R_1$ is called the *thin QR factorisation* of A.

Singular value decomposition. Let $A \in \mathbb{C}^{N,M}$. The decomposition

$$A = U\Sigma V^*, \quad \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p) \in \mathbb{R}^{N,M}, \quad p = \min\{N, M\},$$

where U and V are unitary and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$, is called the *singular value decomposition* of A. The scalars σ_j are called the *singular values* of A. The columns of U and V are the *left and right singular vectors* of A, respectively. The *rank* of A is equal to the number r of nonzero singular values of A. The *pseudoinverse* $A^\dagger$ of A is defined as $A^\dagger = V \operatorname{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}, 0, 0, \ldots, 0)U^*$.

1.3 Polynomial Krylov methods

We now provide a brief overview of polynomial Krylov methods, the predecessor of rational Krylov methods. More detailed expositions can be found in, e.g., [76, 94, 95]. Let $A \in \mathbb{C}^{N,N}$ be a matrix and $\mathbf{0} \neq \boldsymbol{b} \in \mathbb{C}^N$ a nonzero starting vector. For any $m \in \mathbb{N}_0$, the *polynomial Krylov space of order* $m+1$ *for* $(A, \boldsymbol{b})$ is defined as

$$\mathcal{K}_{m+1}(A, \boldsymbol{b}) := \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \dots, A^m\boldsymbol{b}\}.$$

There exists a uniquely defined integer $1 \leq d \equiv d(A, \boldsymbol{b}) \leq N$ such that

$$\mathcal{K}_1(A, \boldsymbol{b}) \subset \mathcal{K}_2(A, \boldsymbol{b}) \subset \cdots \subset \mathcal{K}_{d-1}(A, \boldsymbol{b}) \subset \mathcal{K}_d(A, \boldsymbol{b}) = \mathcal{K}_{d+1}(A, \boldsymbol{b}).$$

We call $d(A, \boldsymbol{b})$ the *invariance index* for $(A, \boldsymbol{b})$. We shall typically assume that $m < d(A, \boldsymbol{b})$, so that $\mathcal{K}_{m+1}(A, \boldsymbol{b})$ is of full dimension $m+1$ and is isomorphic to $\mathcal{P}_m$, i.e., any $\boldsymbol{w} \in \mathcal{K}_{m+1}(A, \boldsymbol{b})$ corresponds to a polynomial $p \in \mathcal{P}_m$ satisfying $\boldsymbol{w} = p(A)\boldsymbol{b}$, and vice versa.

Polynomial Arnoldi algorithm. With the polynomial Arnoldi algorithm given in Algorithm 1.1, one can compute an orthonormal basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_{m+1}\}$ for $\mathcal{K}_{m+1}(A, \boldsymbol{b})$. The starting vector $\boldsymbol{b}$ is normalised to $\boldsymbol{v}_1$ in line 1, and then a new direction $A\boldsymbol{v}_j$ is added to the basis, cf. line 3. The Gram–Schmidt procedure is employed in lines 4–5 to orthonormalise the newly added vector. The process is repeated for $j = 1, 2, \dots, m$. By introducing

$$V_{m+1} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_{m+1} \end{bmatrix} \in \mathbb{C}^{N,m+1}, \quad \text{and} \quad \underline{H_m} = \begin{bmatrix} \underline{\boldsymbol{h}_1} & \underline{\boldsymbol{h}_2} & \dots & \underline{\boldsymbol{h}_m} \end{bmatrix} \in \mathbb{C}^{m+1,m},$$

where $\underline{\boldsymbol{h}_j} = \begin{bmatrix} \boldsymbol{h}_j^T & h_{j+1,j} & 0 & \dots & 0 \end{bmatrix}^T$, we obtain the decomposition $AV_m = V_{m+1}\underline{H_m}$. Here, V_{m+1} is orthonormal while $\underline{H_m}$ is an unreduced upper Hessenberg matrix. Recall that a matrix $\underline{H_m} \in \mathbb{C}^{m+1,m}$ is called *upper Hessenberg* if all the elements below the first subdiagonal are zero, i.e., if $i > j+1$ implies $h_{ij} = 0$. Further, we say that $\underline{H_m}$ is *unreduced* if none of the elements on the first subdiagonal are zero, i.e., $h_{j+1,j} \neq 0$.

Implicit Q theorem. Let us now recall the implicit Q theorem (see, e.g., [42, 102]) which plays an important role for the practical application of the polynomial Arnoldi algorithm.

Algorithm 1.1 Polynomial Arnoldi algorithm.

Input: $A \in \mathbb{C}^{N,N}$, $\boldsymbol{b} \in \mathbb{C}^N$, and $m < d(A, \boldsymbol{b})$.
Output: Decomposition $AV_m = V_{m+1}\underline{H_m}$, with $V_{m+1}^* V_{m+1} = I_{m+1}$.

1. Set $\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|_2$.
2. **for** $j = 1, 2, \ldots, m$ **do**
3. Compute $\boldsymbol{w}_{j+1} := A\boldsymbol{v}_j$.
4. Orthogonalize $\widehat{\boldsymbol{v}}_{j+1} := \boldsymbol{w}_{j+1} - V_j \boldsymbol{h}_j$, where $\boldsymbol{h}_j := V_j^* \boldsymbol{w}_{j+1}$.
5. Normalize $\boldsymbol{v}_{j+1} := \widehat{\boldsymbol{v}}_{j+1}/h_{j+1,j}$, where $h_{j+1,j} := \|\widehat{\boldsymbol{v}}_{j+1}\|_2$.
6. **end for**

Theorem 1.1. *Let $Q \in \mathbb{C}^{N,N}$ be a unitary matrix, and $Q^*AQ = H$ be an unreduced upper Hessenberg matrix. Then the first column of Q determines uniquely, up to unimodular scaling, the other columns of Q.*

One of the applications of the implicit Q theorem is the efficient implementation of the shifted QR iteration (see, e.g., [42, 102]) for the decomposition $AV_m = V_{m+1}\underline{H_m}$, which may accelerate the convergence of specific Ritz values. Instead of the shifted QR iteration for A, the theorem allows for the shifted QR iteration to be applied on the typically smaller matrix $\underline{H_m}$ in an implicit two-step process. First, we change the leading vector $V_{m+1}\boldsymbol{e}_1$ of V_{m+1} by applying a suitable transformation $V_{m+1}GG^{-1}\underline{H_m}$, and second, we recover the upper Hessenberg structure of $G^{-1}\underline{H_m}$ without affecting the leading column of $V_{m+1}G$. This is further discussed for the more general, rational, case at the end of Section 5.1.

Gram–Schmidt procedure. The Gram–Schmidt procedure used in Algorithm 1.1 is often referred to as *classical Gram–Schmidt*, and in finite precision arithmetic may cause numerical instabilities. A more robust approach is that of the *modified Gram–Schmidt* procedure, where, instead of line 4, we have:

for $k = 1, 2, \ldots, j$ **do**
Compute $h_{kj} = \boldsymbol{v}_k^* \boldsymbol{w}_{j+1}$, and update $\boldsymbol{w}_{j+1} := \boldsymbol{w}_{j+1} - h_{kj}\boldsymbol{v}_k$.
end for

In this case line 5 reduces to:

Orthogonalize $\boldsymbol{v}_{j+1} := \boldsymbol{w}_{j+1}/h_{j+1,j}$, where $h_{j+1,j} := \|\widehat{\boldsymbol{v}}_{j+1}\|_2$.

Furthermore, it is common to perform the orthogonalization, with both methods, *twice*. Interested discussions and analyses on this topic can be found in, e.g., [13, 14, 38, 39, 47].

In our forthcoming discussions we shall keep the presentation as in Algorithm 1.1, but one should be aware that a more sophisticated implementation is needed in practice.

Solving linear systems and eigenproblems. The polynomial Arnoldi algorithm may be used for solving large and sparse or structured linear systems of equations. If $AV_m = V_{m+1}\underline{H_m}$ and $H_m = \begin{bmatrix} I_m & \mathbf{0} \end{bmatrix} \underline{H_m}$, then $\boldsymbol{x}_m := V_m H_m^{-1} V_m^* \boldsymbol{b}$ provides an approximation to $A^{-1}\boldsymbol{b}$, provided that A and H_m are nonsingular. This procedure is known as the *full orthogonalization method* (*FOM*). An alternative is the *generalised minimal residual method* (*GMRES*), where $V_{m+1}\underline{H_m}^{\dagger} V_m^* \boldsymbol{b} \approx A^{-1}\boldsymbol{b}$ is used instead. Moreover, some of the eigenvalues of H_m may provide good approximations to eigenvalues of A. In applications, these are typically the eigenvalues having larger module. Therefore, by replacing A with $(A - \xi I)^{-1}$ in the polynomial Arnoldi method, one may obtain good approximations to eigenvalues of A close to any $\xi \in \mathbb{C}$. This is referred to as the *shift-and-invert Arnoldi algorithm*, and the rational Krylov method of Ruhe [86, 87, 88, 89, 90], which we cover in Chapter 2, generalises it by allowing the parameter ξ to change from one iteration to the next. Because of this connection to the polynomial Arnoldi algorithm, we shall refer to the *rational Krylov method* as the *rational Arnoldi algorithm*.

2 Rational Krylov spaces and related decompositions

In this chapter we study various algebraic properties of rational Krylov spaces, using as starting point a *rational Arnoldi decomposition*

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}, \tag{2.1}$$

where $A \in \mathbb{C}^{N,N}$ is a given matrix and the matrices $V_{m+1} \in \mathbb{C}^{N,m+1}$ and $\underline{K_m}, \underline{H_m} \in \mathbb{C}^{m+1,m}$ are of maximal column rank. The rational Arnoldi algorithm by Ruhe [89, 87, 90] naturally generates decompositions of the form (2.1) in which case it is known (by construction) that the columns of V_{m+1} are an (orthonormal) basis of a rational Krylov space. Different choices of the so called *continuation combinations* in the rational Arnoldi algorithm give rise to different decompositions, but all of them correspond to the same rational Krylov space. We answer the converse question of when a decomposition (2.1) is associated with a rational Krylov space, and, furthermore, discuss its uniqueness. The goal is to provide fundamental properties, important for the developments of forthcoming chapters, of decompositions (2.1) related to rational Krylov spaces.

The outline of this chapter is as follows: in Section 2.1 we review the rational Arnoldi algorithm and derive the related decomposition (2.1). The notion of a *rational Arnoldi decomposition* is formally introduced in Section 2.2. We relate these decompositions to the poles and the starting vector of a rational Krylov space and establish some of their properties. Section 2.3 provides a rational implicit Q theorem about the uniqueness of such decompositions, while Section 2.4 is devoted to a variant of (2.1) with all the matrices being real-valued. Rational Krylov spaces were initially proposed for the purpose of solving large sparse generalised eigenvalue problems [86, 89, 87, 90]; in

Section 2.5 we consider the possibility of working with a pencil (A, B), with $A, B \in \mathbb{C}^{N,N}$, instead of A only. This leads to decompositions of the form $AV_{m+1}\underline{K_m} = BV_{m+1}\underline{H_m}$, and naturally opens the question of considering nonstandard inner products. Finally, in Section 2.6 we show how to use the RKToolbox to construct decompositions of the form (2.1), highlighting the flexibility and freedom the RKToolbox provides yet still keeping the exposition concise.

2.1 The rational Arnoldi algorithm

Let $A \in \mathbb{C}^{N,N}$ be a matrix, $\mathbf{0} \neq \boldsymbol{b} \in \mathbb{C}^N$ a nonzero starting vector, and let $q_m \in \mathcal{P}_m$ be a nonzero polynomial that has no roots in $\Lambda(A)$, with $m \in \mathbb{N}_0$. The *rational Krylov space of order m for* $(A, \boldsymbol{b}, q_m)$ is defined as [86, 89]

$$\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) := q_m(A)^{-1}\mathcal{K}_{m+1}(A, \boldsymbol{b}). \tag{2.2}$$

The roots of q_m are the *poles* of $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$. Note that $q_m(A)$ is nonsingular since no root of q_m is an eigenvalue of A and therefore $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ is well defined. Clearly, $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ is independent of nonzero scaling of $\boldsymbol{b}$ and/or q_m. Further, the spaces $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ and $\mathcal{K}_{m+1}(A, \boldsymbol{b})$ are of the same dimension for all m. Therefore $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ is A-variant if and only if $m + 1 < d(A, \boldsymbol{b})$. We shall often denote the poles of a the rational Krylov space by $\{\xi_j\}_{j=1}^m$ and thus may also use the notation $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, \{\xi_j\}_{j=1}^m) = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$. If $\deg(q_m) < m$, then $m - \deg(q_m)$ of the poles are set to infinity. In this case we refer to infinity as a *formal* (multiple) root of q_m. To handle both finite and infinite poles in a unifying way we may also use the representation $\xi = \mu/\nu$, for an adequate choice of scalars $\mu, \nu \in \mathbb{C}$.

The rational Arnoldi algorithm [89, 90] constructs an orthonormal basis V_{m+1} for (2.2) in a Gram–Schmidt fashion as described in Algorithm 2.2. In line 1 we normalise the starting vector $\boldsymbol{b}$. The main part of the algorithm consists of lines 2–11 where an orthonormal basis for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ is constructed iteratively.

In line 3 we select a *continuation pair* $(\eta_j/\rho_j, \boldsymbol{t}_j)$ which is used in line 4 to expand the space $\mathcal{R}(V_j)$.

Definition 2.1. *We call* $(\eta_j/\rho_j, \boldsymbol{t}_j \neq \mathbf{0}) \in \overline{\mathbb{C}} \times \mathbb{C}^m$ *a* continuation pair of order j. *The value* η_j/ρ_j *is its* continuation root, *and* $\boldsymbol{t}_j$ *its* continuation vector.

Algorithm 2.2 Rational Arnoldi algorithm. RKToolbox: `rat_krylov`

Input: $A \in \mathbb{C}^{N,N}$, $\boldsymbol{b} \in \mathbb{C}^N$, poles $\{\mu_j/\nu_j\}_{j=1}^m \subset \overline{\mathbb{C}} \setminus \Lambda(A)$, with $m < d(A, \boldsymbol{b})$.
Output: Decomposition $AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}$, with $V_{m+1}^* V_{m+1} = I_{m+1}$.

1. Set $\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|_2$.
2. **for** $j = 1, 2, \ldots, m$ **do**
3. Choose an *admissible continuation pair* $(\eta_j/\rho_j, \boldsymbol{t}_j) \in \overline{\mathbb{C}} \times \mathbb{C}^j$.
4. Compute $\boldsymbol{w}_{j+1} := (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)V_j \boldsymbol{t}_j$.
5. Orthogonalize $\widehat{\boldsymbol{v}}_{j+1} := \boldsymbol{w}_{j+1} - V_j \boldsymbol{c}_j$, where $\boldsymbol{c}_j := V_j^* \boldsymbol{w}_{j+1}$.
6. Normalize $\boldsymbol{v}_{j+1} := \widehat{\boldsymbol{v}}_{j+1}/c_{j+1,j}$, where $c_{j+1,j} := \|\widehat{\boldsymbol{v}}_{j+1}\|_2$.
7. Set $\underline{\boldsymbol{k}_j} := \nu_j \underline{\boldsymbol{c}_j} - \rho_j \underline{\boldsymbol{t}_j}$ and $\underline{\boldsymbol{h}_j} := \mu_j \underline{\boldsymbol{c}_j} - \eta_j \underline{\boldsymbol{t}_j}$, where $\underline{\boldsymbol{t}_j} = \begin{bmatrix} \boldsymbol{t}_j \\ 0 \end{bmatrix}$, and $\underline{\boldsymbol{c}_j} = \begin{bmatrix} \boldsymbol{c}_j \\ c_{j+1,j} \end{bmatrix}$.
8. **end for**

The notion of continuation vector has already been used in the literature, though not consistently. For instance, in [90] the author refers to $V_j \boldsymbol{t}_j$ as the continuation vector, while in [73] the term is used to denote $(\rho_j A - \eta_j I)V_j \boldsymbol{t}_j$. The terminology of "continuation combinations" is adopted in [10, 109, 90] for the vectors $\boldsymbol{t}_j$. With the notion of continuation pair, we want to stress that the two components are equally important; see Chapter 4.

The Möbius transformation $(\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)$ with fixed pole μ_j/ν_j and the chosen (continuation) root $\eta_j/\rho_j \neq \mu_j/\nu_j$ is applied onto $V_j \boldsymbol{t}_j$ in order to produce $\boldsymbol{w}_{j+1}$. The continuation pair must be such that $\boldsymbol{w}_{j+1} \notin \mathcal{R}(V_j)$, as otherwise we cannot expand the space. Such *admissible* continuation pairs exist as long as $j < d(A, \boldsymbol{b})$; a thorough discussion on the selection of continuation pairs is included in Chapter 4. For now it is sufficient to add that (admissible) continuation pairs correspond to linear parameters and do not affect the space (in exact arithmetic, at least). Lines 5–6 correspond to the Gram–Schmidt process, where $\boldsymbol{w}_{j+1}$ is orthogonalised against $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_j$ to produce the unit 2-norm vector $\boldsymbol{v}_{j+1}$. From lines 4–6 we deduce

$$\boldsymbol{w}_{j+1} = V_{j+1}\underline{\boldsymbol{c}_j} = (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)V_j \boldsymbol{t}_j, \quad \text{and hence} \tag{2.3a}$$

$$(\nu_j A - \mu_j I)V_{j+1}\underline{\boldsymbol{c}_j} = (\rho_j A - \eta_j I)V_j \boldsymbol{t}_j. \tag{2.3b}$$

Rearranging the terms with and without A we obtain

$$AV_{j+1}(\nu_j \underline{\boldsymbol{c}_j} - \rho_j \underline{\boldsymbol{t}_j}) = V_{j+1}(\mu_j \underline{\boldsymbol{c}_j} - \eta_j \underline{\boldsymbol{t}_j}), \tag{2.4}$$

which justifies the notation

$$\underline{\boldsymbol{k}_j} := \nu_j \underline{\boldsymbol{c}_j} - \rho_j \underline{\boldsymbol{t}_j}, \quad \text{and} \quad \underline{\boldsymbol{h}_j} := \mu_j \underline{\boldsymbol{c}_j} - \eta_j \underline{\boldsymbol{t}_j}, \tag{2.5}$$

used in line 7. Note that $h_{j+1,j} = \mu_j c_{j+1,j}$ and $k_{j+1,j} = \nu_j c_{j+1,j}$, with $c_{j+1,j} \neq 0$. Hence, $h_{j+1,j}/k_{j+1,j}$ is equal to the jth pole. Concatenating (2.4) for $j = 1, 2, \ldots, m$ provides

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}, \tag{2.6}$$

with the jth column of $\underline{H_m} \in \mathbb{C}^{m+1,m}$ being $[\underline{\boldsymbol{h}_j}^T \quad \mathbf{0}^T]^T \in \mathbb{C}^{m+1}$, and analogously for the matrix $\underline{K_m}$. It is convenient to consider (2.6) even if $m = 0$, in which case one can think of the pencil $(\underline{H_m}, \underline{K_m})$ as being of size 1-by-0, and we only have the matrix A and the normalised starting vector $\boldsymbol{v}_1$. This corresponds to the initial stage of Algorithm 2.2, i.e., right after line 1.

The rational Arnoldi algorithm is a generalisation of the polynomial and shift-and-invert Arnoldi algorithms, and the latter two can be recovered with a specific choice of poles and continuation pairs, as the following two examples demonstrate.

Example 2.2. Let $\mu_j/\nu_j \equiv 1/0$, and $(\eta_j/\rho_j, \boldsymbol{t}_j) \equiv (0/-1, \boldsymbol{e}_j)$, for $j = 1, 2, \ldots, m$. Then in line 4 of Algorithm 2.2 we compute $\boldsymbol{w}_{j+1} = AV_j\boldsymbol{e}_j = A\boldsymbol{v}_j$. Furthermore, the formulas for $\underline{\boldsymbol{k}_j} = \underline{\boldsymbol{e}_j}$, and $\underline{\boldsymbol{h}_j} = \underline{\boldsymbol{c}_j}$ simplify. Overall, we retrieve $AV_{m+1}\underline{I_m} = AV_m = V_{m+1}\underline{H_m}$, same as with the polynomial Arnoldi algorithm; cf. Section 1.3.

Example 2.3. Recall the shift-and-invert Arnoldi decomposition $(A - \sigma I)^{-1}V_m = V_{m+1}\underline{C_m}$. By multiplying it from the left with $(A - \sigma I)$ and rearranging the terms we obtain (2.6) with $\underline{K_m} = \underline{C_m}$, and $\underline{H_m} = \sigma\underline{C_m} + \underline{I_m}$. This can be obtained with Algorithm 2.2 by setting $\mu_j/\nu_j \equiv \sigma/1$, and $(\eta_j/\rho_j, \boldsymbol{t}_j) \equiv (-1/0, \boldsymbol{e}_j)$, for all iterations $j = 1, 2, \ldots, m$.

The polynomial Arnoldi algorithm uses repeatedly a pole at infinity, while the shift-and-invert Arnoldi algorithm uses a finite pole σ at each iteration. The rational Arnoldi algorithm allows for poles to change from one iteration to the next. The success of rational Krylov methods heavily depends on these parameters. If *good* poles are available, only a few may suffice to solve the problem at hand. Otherwise, the solution of a large number of shifted linear systems may be needed to construct the space, thus rendering the process computationally unfeasible. Finding good pole parameters is highly non-trivial and problem-depend. We discuss the selection of poles in Chapters 4–6.

2.2 Rational Arnoldi decompositions

In the following we aim to establish a correspondence between rational Krylov spaces and matrix decompositions of the form (2.6). As a consequence, we are able to study the algebraic properties of rational Krylov spaces using these decompositions.

Definition 2.4. *Let* $\underline{K_m}, \underline{H_m} \in \mathbb{C}^{m+1,m}$ *be upper Hessenberg matrices. We say that the pencil* $(\underline{H_m}, \underline{K_m})$ *is an* unreduced upper Hessenberg pencil *if* $|h_{j+1,j}| + |k_{j+1,j}| \neq 0$ *for all* $j = 1, 2, \ldots, m$.

We are now ready to introduce the notion of a rational Arnoldi decomposition, which is a generalisation of decompositions generated by Ruhe's rational Arnoldi algorithm [89, 90]. Although these decompositions have been considered before, ours is the most general definition (cf. Theorem 2.10 below). Other approaches typically exclude the possibility to have poles at both zero and infinity, by requiring $\underline{H_m}$ to be unreduced; see, e.g., [24, 55, 90]. The introduction of unreduced pencils allows us to bypass this restriction.

Definition 2.5. *Let* $A \in \mathbb{C}^{N,N}$. *A relation of the form* (2.6) *is called a* rational Arnoldi decomposition (RAD) of order m *if* $V_{m+1} \in \mathbb{C}^{N,m+1}$ *is of full column rank,* $(\underline{H_m}, \underline{K_m})$ *is an unreduced upper Hessenberg pencil of size* $(m+1)$*-by-*m*, and none of the quotients* $\{h_{j+1,j}/k_{j+1,j}\}_{j=1}^{m}$*, called* poles *of the decomposition, is in* $\Lambda(A)$. *The columns of* V_{m+1} *are called the* basis *of the RAD and they span the* space *of the RAD. If* V_{m+1} *is orthonormal, we say that* (2.6) *is an* orthonormal RAD.

The terminology of basis and space of an RAD is inspired by [101, 103] where decompositions related to the polynomial Arnoldi algorithm are studied. It is noteworthy that both $\underline{H_m}$ and $\underline{K_m}$ in the RAD (2.6) are of full rank, which follows from the following lemma (for $\beta = 0$ and $\alpha = 0$, respectively).

Lemma 2.6. *Let* (2.6) *be an RAD, and let* $\alpha, \beta \in \mathbb{C}$ *be such that* $|\alpha| + |\beta| \neq 0$. *The matrix* $\alpha\underline{H_m} - \beta\underline{K_m}$ *is of full column rank* m.

Proof. Consider auxiliary scalars $\widehat{\alpha} = 1$ and any $\widehat{\beta} \in \mathbb{C}$ such that $\widehat{\alpha}h_{j+1,j} - \widehat{\beta}k_{j+1,j} \neq 0$ for $j = 1, 2, \ldots, m$. Multiplying the RAD (2.6) by $\widehat{\alpha}$ and subtracting $\widehat{\beta}V_{m+1}\underline{K_m}$ from both sides gives

$$(\widehat{\alpha}A - \widehat{\beta}I)V_{m+1}\underline{K_m} = V_{m+1}\big(\widehat{\alpha}\underline{H_m} - \widehat{\beta}\underline{K_m}\big). \tag{2.7}$$

The choice of $\widehat{\alpha}$ and $\widehat{\beta}$ is such that $\widehat{\alpha}\underline{H_m} - \widehat{\beta}\underline{K_m}$ is an unreduced upper Hessenberg

matrix, and as such of full column rank m. In particular, the right-hand side of (2.7) is of full column rank m. Thus, the left-hand side, and in particular $\underline{K_m}$, is of full column rank. This proves the statement for the case $\alpha = 0$. For the case $\alpha \neq 0$, consider $\widehat{\alpha} = \alpha$ and $\widehat{\beta} = \beta$ in (2.7). If $\alpha\underline{H_m} - \beta\underline{K_m}$ is unreduced, then it is of full column rank and the statement follows. If, however, $\alpha\underline{H_m} - \beta\underline{K_m}$ is not unreduced, then we have $\alpha h_{j+1,j} - \beta k_{j+1,j} = 0$ for at least one index $j \in \{1, 2, \ldots, m\}$. Equivalently, $\beta/\alpha = h_{j+1,j}/k_{j+1,j}$; that is, β/α equals the jth pole of (2.6) and hence $\alpha A - \beta I$ is nonsingular. Finally, since V_{m+1} and $\underline{K_m}$ are of full column rank, the left-hand side of (2.7) is of full column rank. It follows that $\alpha\underline{H_m} - \beta\underline{K_m}$ is of full column rank as well, and the proof is complete. □

Furthermore, any RAD (2.6) can be transformed into an orthonormal RAD using the thin QR factorization $V_{m+1} = Q_{m+1}R_{m+1}$. Setting $\widehat{V}_{m+1} = Q_{m+1}$, $\underline{\widehat{K}_m} = R_{m+1}\underline{K_m}$, and $\underline{\widehat{H}_m} = R_{m+1}\underline{H_m}$, we obtain the decomposition

$$A\widehat{V}_{m+1}\underline{\widehat{K}_m} = \widehat{V}_{m+1}\underline{\widehat{H}_m}, \tag{2.8}$$

satisfying $\mathcal{R}(\widehat{V}_{j+1}) = \mathcal{R}(V_{j+1})$, and $h_{j+1,j}/k_{j+1,j} = \widehat{h}_{j+1,j}/\widehat{k}_{j+1,j}$ for all $j = 1, 2, \ldots, m$.

Definition 2.7. *The RADs* (2.6) *and* (2.8) *are called* equivalent *if they span the same space and have the same poles.*

Note that we do not impose equal ordering of the poles for two RADs to be equivalent. Additionally, it follows from Lemma 2.8 below that equivalent RADs have the same starting vector, up to nonzero scaling. We shall often assume, for convenience, the RAD to be orthonormal. We now show that the poles of a rational Krylov space are uniquely determined by the starting vector and vice versa.

Lemma 2.8. *Let $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ be a given A-variant rational Krylov space. Then the poles of $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ are uniquely determined by $\mathcal{R}(\boldsymbol{b})$, or equivalently, the starting vector of $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ is uniquely, up to scaling, determined by the (formal) roots of the polynomial q_m.*

Proof. We first show that for a given A-variant polynomial Krylov space $\mathcal{K}_{m+1}(A, \boldsymbol{q})$, all vectors $\boldsymbol{w} \in \mathcal{K}_{m+1}(A, \boldsymbol{q})$ that satisfy $\mathcal{K}_{m+1}(A, \boldsymbol{q}) = \mathcal{K}_{m+1}(A, \boldsymbol{w})$ are of the form $\boldsymbol{w} = \alpha\boldsymbol{q}$, for a nonzero scalar $\alpha \in \mathbb{C}$. Assume, to the contrary, that there exists a polynomial p_j with $1 \leq \deg(p_j) = j \leq m$ such that $\boldsymbol{w} = p_j(A)\boldsymbol{q}$. Then $A^{m+1-j}\boldsymbol{w} \in \mathcal{K}_{m+1}(A, \boldsymbol{w})$, but for the same vector we have $A^{m+1-j}\boldsymbol{w} = A^{m+1-j}p_j(A)\boldsymbol{q} \notin \mathcal{K}_{m+1}(A, \boldsymbol{q})$. This is a contradiction to $\mathcal{K}_{m+1}(A, \boldsymbol{q}) = \mathcal{K}_{m+1}(A, \boldsymbol{w})$.

To show that the poles are uniquely determined by the starting vector $\boldsymbol{b}$, assume that $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, \widehat{q}_m)$. Using the definition of a rational Krylov space (2.2), this is equivalent to $\mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b}) = \mathcal{K}_{m+1}(A, \widehat{q}_m(A)^{-1}\boldsymbol{b})$. Multiplying the latter with $q_m(A)\widehat{q}_m(A) = \widehat{q}_m(A)q_m(A)$ from the left provides the equivalent $\mathcal{K}_{m+1}(A, \widehat{q}_m(A)\boldsymbol{b}) = \mathcal{K}_{m+1}(A, q_m(A)\boldsymbol{b})$. This space is A-variant, hence by the above argument we know that $q_m(A)\boldsymbol{b} = \alpha\widehat{q}_m(A)\boldsymbol{b}$, for a nonzero scalar $\alpha \in \mathbb{C}$. This vector is an element of $\mathcal{K}_{m+1}(A, \boldsymbol{b})$ which is isomorphic to $\mathcal{P}_m$. Therefore $q_m = \alpha\widehat{q}_m$ and hence q_m and $\widehat{q}_m$ have identical roots. Similarly one shows that if $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) = \mathcal{Q}_{m+1}(A, \widehat{\boldsymbol{b}}, q_m)$, then $\boldsymbol{b} = \alpha\widehat{\boldsymbol{b}}$ with $\alpha \neq 0$. ∎

The rational Arnoldi algorithm generates RADs of the form (2.6), in which case it is known (by construction) that $\mathcal{R}(V_{m+1})$ spans a rational Krylov space. In Theorem 2.10 below we show that the converse also holds; we show that for every rational Krylov space $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ there exists an RAD (2.6) spanning $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ and conversely, if such a decomposition exists it spans a rational Krylov space. In particular, this shows that our Definition 2.5 indeed describes the complete set of RADs associated with rational Krylov spaces. To proceed it is convenient to write the polynomial q_m in factored form, and to label separately all the leading factors

$$q_0(z) = 1, \quad \text{and} \quad q_j(z) = \prod_{\ell=1}^{j} \left(h_{\ell+1,\ell} - k_{\ell+1,\ell} z\right), \quad j = 1, 2, \ldots, m, \tag{2.9}$$

with some scalars $\{h_{\ell+1,\ell}, k_{\ell+1,\ell}\}_{\ell=1}^{m} \subset \mathbb{C}$ such that $\xi_\ell = h_{\ell+1,\ell}/k_{\ell+1,\ell}$. Since (2.2) is independent of the scaling of q_m any choice of the scalars $h_{\ell+1,\ell}$ and $k_{\ell+1,\ell}$ is valid as long as their ratio is ξ_ℓ. When we make use of (2.9) without specifying the order of appearance of the poles, we mean any order. The fact that $q_j \mid q_{j+1}$ gives rise to a sequence of nested rational Krylov spaces, as we now show.

Proposition 2.9. *Let $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ be a rational Krylov space of order m, and let (2.9) hold. Then*

$$\mathcal{Q}_1 \subset \mathcal{Q}_2 \subset \cdots \subset \mathcal{Q}_{m+1}, \tag{2.10}$$

where $\mathcal{Q}_{j+1} = \mathcal{Q}_{j+1}(A, \boldsymbol{b}, q_j)$ for $j = 0, 1, \ldots, m$.

Proof. Let $\ell \in \{0, 1, \ldots, m\}$. We need to show that $\mathcal{Q}_\ell \subset \mathcal{Q}_{\ell+1}$. Let $\boldsymbol{v} \in \mathcal{Q}_\ell$ be arbitrarily. By the definition of $\mathcal{Q}_\ell$, there exists a polynomial $p_\ell \in \mathcal{P}_\ell$ such that $\boldsymbol{v} = q_\ell(A)^{-1}p_\ell(A)\boldsymbol{b}$. Then, $p_{\ell+1} \in \mathcal{P}_{\ell+1}$ defined by $p_{\ell+1}(z) := \left(h_{\ell+1,\ell} - k_{\ell+1,\ell} z\right) p_\ell(z)$ is such that $\boldsymbol{v} = q_{\ell+1}(A)^{-1}p_{\ell+1}(A)\boldsymbol{b}$, which shows that $\boldsymbol{v} \in \mathcal{Q}_{\ell+1}$. ∎

Finally, we are ready to establish the announced relation between rational Krylov spaces and RADs.

Theorem 2.10. *Let $\mathcal{V}_{m+1}$ be a vector space of dimension $m+1$. Then $\mathcal{V}_{m+1}$ is a rational Krylov space with starting vector $\boldsymbol{b} \in \mathcal{V}_{m+1}$ and poles $\xi_1, \xi_2, \ldots, \xi_m$ if and only if there exists an RAD (2.6) with $\mathcal{R}(V_{m+1}) = \mathcal{V}_{m+1}$, $\boldsymbol{v}_1 = \boldsymbol{b}$, and poles $\xi_1, \xi_2, \ldots, \xi_m$.*

Proof. Let (2.6) hold and define the polynomials $\{q_j\}_{j=0}^{m}$ as in (2.9). These are nonzero polynomials since the pencil $(\underline{H_m}, \underline{K_m})$ is unreduced. We show by induction that

$$\mathcal{V}_{j+1} := \operatorname{span}\left\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{j+1}\right\} = q_j(A)^{-1}\mathcal{K}_{j+1}(A, \boldsymbol{b}), \tag{2.11}$$

for $j = 1, 2, \ldots, m$, and with $\boldsymbol{b} = \boldsymbol{v}_1$. In particular, for $j = m$ we obtain $\mathcal{V}_{m+1} = q_m(A)^{-1}\mathcal{K}_{m+1}(A, \boldsymbol{b})$. Consider $j = 1$. Reading (2.6) column-wise, first column only, and rearranging the terms yields

$$q_1(A)\boldsymbol{v}_2 = (h_{21}I - k_{21}A)\,\boldsymbol{v}_2 = (k_{11}A - h_{11}I)\,\boldsymbol{v}_1. \tag{2.12}$$

Therefore, $\boldsymbol{v}_2 = q_1(A)^{-1}\,(k_{11}A - h_{11}I)\,\boldsymbol{v}_1 \in q_1(A)^{-1}\mathcal{K}_2(A, \boldsymbol{b})$ which together with the fact $\boldsymbol{v}_1 \in q_1(A)^{-1}\mathcal{K}_2(A, \boldsymbol{b})$ proves (2.11) for $j = 1$. Let us assume that (2.11) holds for $j = 1, 2, \ldots, n-1 < m$. We now consider the case $j = n$. Comparing the nth column on the left- and the right-hand side in (2.6) and rearranging the terms gives

$$\left(h_{n+1,n}I - k_{n+1,n}A\right)\boldsymbol{v}_{n+1} = \sum_{\ell=1}^{n}(k_{\ell n}A - h_{\ell n}I)\,\boldsymbol{v}_\ell, \tag{2.13}$$

$$\text{and hence} \qquad q_n(A)\boldsymbol{v}_{n+1} = \sum_{\ell=1}^{n}(k_{\ell n}A - h_{\ell n}I)\,q_{n-1}(A)\boldsymbol{v}_\ell. \tag{2.14}$$

By the induction hypothesis $\boldsymbol{v}_\ell \in q_{n-1}(A)^{-1}\mathcal{K}_n(A, \boldsymbol{b})$, therefore

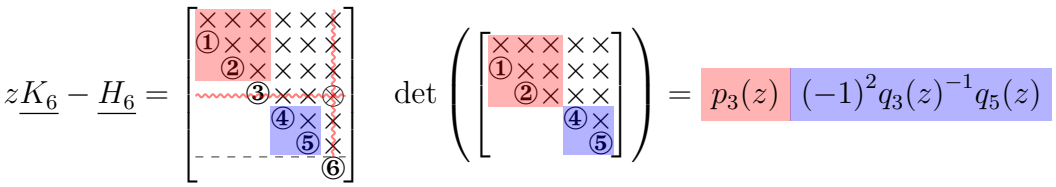
$$(k_{\ell n}A - h_{\ell n}I)\,q_{n-1}(A)\boldsymbol{v}_\ell \in \mathcal{K}_{n+1}(A, \boldsymbol{b}), \quad \ell = 1, 2, \ldots, n. \tag{2.15}$$

It follows from (2.14) and (2.15) that $\boldsymbol{v}_{n+1} \in q_n(A)^{-1}\mathcal{K}_{n+1}(A, \boldsymbol{b})$. The induction hypothesis asserts $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\} \subseteq q_n(A)^{-1}\mathcal{K}_{n+1}(A, \boldsymbol{b})$ which concludes this direction.

Alternatively, let $\mathcal{V}_{m+1} = q_m(A)^{-1}\mathcal{K}_{m+1}(A, \boldsymbol{b})$ be a rational Krylov space with a basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{m+1}\}$ satisfying (2.11). Thus for $n = 1, 2, \ldots, m$ there holds

$$\boldsymbol{v}_{n+1} \in q_n(A)^{-1}\mathcal{K}_{n+1}(A, \boldsymbol{b}) \Leftrightarrow \left(h_{n+1,n}I - k_{n+1,n}A\right)\boldsymbol{v}_{n+1} \in q_{n-1}(A)^{-1}\mathcal{K}_{n+1}(A, \boldsymbol{b}).$$

Since $\mathcal{K}_{n+1}(A, \boldsymbol{b}) = \mathcal{K}_n(A, \boldsymbol{b}) \cup A\mathcal{K}_n(A, \boldsymbol{b})$ we have $q_{n-1}(A)^{-1}\mathcal{K}_{n+1}(A, \boldsymbol{b}) = \mathcal{Q}_n \cup A\mathcal{Q}_n$. Consequently, there exist numbers $\{h_{\ell n}, k_{\ell n}\}_{\ell=1}^{n} \subset \mathbb{C}$ such that (2.13) holds. These



(a) Upper Hessenberg structure. (b) Contribution from a minor.

Figure 2.1: Sketch illustrating the proof of Theorem 2.12. Part (a) shows the upper-Hessenberg structure of the shifted pencil $z\underline{K_j} - \underline{H_j}$, for $j = 6$. The elements marked with numbers, like ① $= zk_{21} - h_{21}$, are those carrying the poles. The contribution of the element $\otimes = zk_{46} - h_{46}$ in the Laplace expansion of the determinant $\det(zK_6 - H_6)$ along the last column of the matrix is $(-1)^{4+6}(zk_{46} - h_{46})\det(M_\otimes)$. Here, $M_\otimes$ is the minor of $zK_6 - H_6$ resulting from the removal of the 4th row and the last column, and is shown in part (b).

relations can be merged into matrix form to get (2.6) with the pencil $(\underline{H_m}, \underline{K_m})$ being unreduced as a consequence of q_m being a nonzero polynomial. □

Clearly, an RAD related to a rational Krylov space $\mathcal{V}_{m+1}$ with a given starting vector and poles is not unique; not only can the poles be ordered arbitrarily, but also the scalars $\{h_{\ell n}, k_{\ell n}\}_{\ell=1}^{n+1}$ can be chosen in a nonunique way. The different RADs are, however, equivalent. We comment further on the uniqueness in Section 2.3. Let us now introduce the following terminology.

Definition 2.11. *We say that* (2.6) is an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ *if* (2.6) *is an RAD spanning* $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$*, with* $V_{m+1}\boldsymbol{e}_1$ *being collinear to* $\boldsymbol{b}$*, and if the poles of* (2.6) *coincide with the poles of* $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$*.*

Elaborating further on the proof of the previous theorem, we retrieve an explicit formula for the vectors $\boldsymbol{v}_j$, given in Theorem 2.12 below. This result appears to some extent in [86, 89], stated up to a normalization factor and given without proof. We stress that the result holds irrespectively of the RAD being orthonormal or not.

Theorem 2.12. *Let* (2.6) *be an RAD. Then*

$$\boldsymbol{v}_{j+1} = p_j(A)q_j(A)^{-1}\boldsymbol{v}_1, \quad j = 1, 2, \ldots, m, \tag{2.16}$$

where $p_j(z) = \det\left(zK_j - H_j\right)$*, and the polynomials* q_j *are given by* (2.9)*.*

Proof. The proof goes by induction on j. For $j = 1$, (2.16) follows from (2.12).

Assume (2.16) has been established for $j = 1, 2, \ldots, n < m$ and insert it into (2.14), giving rise to

$$q_n(A)\boldsymbol{v}_{n+1} = \sum_{\ell=1}^{n} (k_{\ell n}A - h_{\ell n}I)\, q_{n-1}(A)p_{\ell-1}(A)q_{\ell-1}(A)^{-1}\boldsymbol{v}_1. \tag{2.17}$$

We obtain (2.16) for $j = n+1$ by noticing that the right-hand side of (2.17) represents the Laplace expansion of $\det(zK_n - H_n)$ along the last column. Indeed

$$q_n(A)\boldsymbol{v}_{n+1} = \sum_{\ell=1}^{n} (-1)^{\ell+n} \left(k_{\ell n}A - h_{\ell n}I\right) p_{\ell-1}(A)(-1)^{n-\ell} q_{\ell-1}(A)^{-1} q_{n-1}(A)\boldsymbol{v}_1.$$

See also Figure 2.1 for an illustration. □

We remark that $p_j(z)$ is the determinant of the upper j-by-j submatrix of $z\underline{K_j} - \underline{H_j}$, whilst $(-1)^j q_j(z)$ is the determinant of its lower j-by-j submatrix. Clearly, the pencil $(\underline{H_m}, \underline{K_m})$ implicitly defines the *scalar* rational functions p_j/q_j, and they satisfy a *scalar RAD* (2.18) as we now show.

Theorem 2.13. *Let (2.6) be an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$. Define $r_j := p_j/q_j$ where the polynomials $p_j, q_j \in \mathcal{P}_j$ for $j = 1, 2, \ldots, m$ are as in (2.16), and $r_0 \equiv 1$. Then for any $z \in \mathbb{C}$ such that $q_m(z) \neq 0$ there holds*

$$z \begin{bmatrix} r_0(z) & r_1(z) & \ldots & r_m(z) \end{bmatrix} \underline{K_m} = \begin{bmatrix} r_0(z) & r_1(z) & \ldots & r_m(z) \end{bmatrix} \underline{H_m}. \tag{2.18}$$

Furthermore, for any $z \in \mathbb{C}$ there holds

$$z \begin{bmatrix} p_0^{[m]}(z) & p_1^{[m]}(z) & \ldots & p_m^{[m]}(z) \end{bmatrix} \underline{K_m} = \begin{bmatrix} p_0^{[m]}(z) & p_1^{[m]}(z) & \ldots & p_m^{[m]}(z) \end{bmatrix} \underline{H_m}, \tag{2.19}$$

where $p_j^{[m]} \in \mathcal{P}_m$ are polynomials formally defined as $p_j^{[m]} \equiv r_j q_m$, for $j = 0, 1, \ldots, m$.

Proof. Can be verified column-wise as in the proofs of Theorems 2.10 and 2.12. □

The scalar RAD indicates a way of evaluating the rational functions r_j at arbitrary points $z \in \mathbb{C}$, excluding the poles, using the information contained in the corresponding scalar RAD. We remark that the scalar variant of an RAD is well known in the polynomial case, see, e.g., [76, eq. (3.3.10)].

Theorem 2.14. *Let (2.6) be an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, and let the rational functions r_j for $j = 0, 1, \ldots, m$ be as in Theorem 2.13. Then for any $w \in \mathbb{C}$ such that $q_m(w) \neq 0$ there holds*

$$r_j(w) = \overline{\gamma}, \quad \text{where} \quad \gamma = \frac{\boldsymbol{q}_{m+1}^* \boldsymbol{e}_{j+1}}{\boldsymbol{q}_{m+1}^* \boldsymbol{e}_1},$$

with $\boldsymbol{q}_{m+1}$ being the last, i.e., $(m+1)$st, column of the Q factor of the full QR factorisation of $\underline{H_m} - w\underline{K_m}$.

Proof. The scalar RAD (2.18) can be shifted to

$$(z-w) \begin{bmatrix} r_0(z) & r_1(z) & \ldots & r_m(z) \end{bmatrix} \underline{K_m} = \begin{bmatrix} r_0(z) & r_1(z) & \ldots & r_m(z) \end{bmatrix} \left(\underline{H_m} - w\underline{K_m}\right),$$

which shows, by setting $z = w$, that indeed $\boldsymbol{q} := \begin{bmatrix} r_0(w) & r_1(w) & \dots & r_m(w) \end{bmatrix}^* \neq \mathbf{0}$ is a left null vector of $\underline{H_m} - w\underline{K_m}$. (The vector $\boldsymbol{q}$ is nonzero since $r_0(w) = 1$.) Since $\underline{H_m} - w\underline{K_m}$ is of full column rank m by Lemma 2.6, the null vector $\boldsymbol{q}$ is unique up to nonzero scaling. Therefore, $\boldsymbol{q} = \boldsymbol{q}_{m+1}/\boldsymbol{q}_{m+1}^* \boldsymbol{e}_1$, and the statement follows. ∎

In Chapter 7 we further discuss scalar RADs, based on which we propose a framework to work with (scalar) rational functions numerically. The evaluation of rational functions based on Theorem 2.14 requires the computation of a QR factorisation of a typically rather small matrix. However, this may become costly if the evaluation in many points is required, and in Chapter 7 we introduce, among others, a more efficient algorithm.

We remark that some of the roots of r_m can yield good approximations to some of the eigenvalues of A, and in Section 3.1.4 we discuss this in more detail. We shall now focus again on non-scalar RADs and in particular on the question of uniqueness.

2.3 A rational implicit Q theorem

We now return to the question of uniqueness of RADs for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$. We need to restrict ourselves to orthonormal RADs, and consider a fixed ordering of the poles. This allows us to establish a generalisation of the implicit Q theorem to the rational case. The theorem asserts that any two orthonormal RADs for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, with poles ordered in the same way, are *essentially equal*. Let us clarify what it means for two RADs to be essentially unique. Apart from the column scaling of V_{m+1}, in the rational case the decomposition (2.6) is also *invariant* (in the sense that it spans the same space, the poles remain unchanged, and the upper Hessenberg structure is preserved) under right-multiplication by upper triangular nonsingular matrices T_m. We make this precise.

Definition 2.15. *The orthonormal RADs* (2.6) *and* (2.8) *are called* essentially equal *if there exists a unitary diagonal matrix $D_{m+1} \in \mathbb{C}^{m+1,m+1}$ and an upper triangular nonsingular matrix $T_m \in \mathbb{C}^{m,m}$, such that $\widehat{V}_{m+1} = V_{m+1}D_{m+1}$, $\underline{\widehat{H}_m} = D_{m+1}^* \underline{H_m} T_m$, and $\underline{\widehat{K}_m} = D_{m+1}^* \underline{K_m} T_m$. Essentially equal orthonormal RADs form an equivalence class and we call any of its elements* essentially unique.

Note that two orthonormal RADs may be equivalent but not essentially equal, as the poles may be ordered differently. We are now ready to generalise the implicit Q

theorem to the rational case. The proof is partly analogous to that of the polynomial implicit Q theorem given in [102, pp. 116–117].

Theorem 2.16. *Let (2.6) be an orthonormal RAD with poles $\xi_j = h_{j+1,j}/k_{j+1,j}$. For every $j = 1, 2, \dots, m$ the orthonormal matrix V_{j+1} and the pencil $(\underline{H_j}, \underline{K_j})$ are essentially uniquely determined by $V_{j+1}\boldsymbol{e}_1$ and the poles $\xi_1, \xi_2, \dots, \xi_j$.*

Proof. Let (2.8) be an orthonormal RAD with $\widehat{V}_{m+1}\boldsymbol{e}_1 = V_{m+1}\boldsymbol{e}_1$ and $\widehat{h}_{j+1,j}/\widehat{k}_{j+1,j} = h_{j+1,j}/k_{j+1,j}$ for all $j = 1, 2, \dots, m$. We show by induction that (2.8) is essentially equal to (2.6). We assume, without loss of generality, that $h_{j+1,j} \neq 0$ for all $j = 1, 2, \dots, m$. Otherwise, if $h_{j+1,j} = 0$ for some j, then $0 = \xi_j \notin \Lambda(A)$ and we can consider $V_{m+1}\underline{K_m} = A^{-1}V_{m+1}\underline{H_m}$ at that step j, thus interchanging the roles of $\underline{H_m}$ and $\underline{K_m}$ and using A^{-1} instead of A. Since $(\underline{H_m}, \underline{K_m})$ is unreduced, $k_{j+1,j} \neq 0$ if $h_{j+1,j} = 0$. The relation (2.6) can be shifted for all $\xi \in \overline{\mathbb{C}}^* \setminus \Lambda(A)$ to provide

$$A^{(\xi)}V_{m+1}\underline{L_m^{(\xi)}} = V_{m+1}\underline{H_m}, \tag{2.20}$$

where $A^{(\xi)} := (I - A/\xi)^{-1}A$ and $\underline{L_m^{(\xi)}} := (\underline{K_m} - \underline{H_m}/\xi)$. We make frequent use of this relation, reading it column-wise. It is worth noticing that the jth column of $\underline{L_m^{(\xi_j)}}$ has all but eventually the leading j components equal to zero, and that $\underline{L_j^{(\xi)}}$ is of full rank for all j and ξ, by Lemma 2.6. Analogous results hold for (2.8). We are now ready to prove the statement.

Define $d_1 := 1$, so that $\widehat{\boldsymbol{v}}_1 = d_1\boldsymbol{v}_1$. The first column in (2.20) for $\xi = \xi_1$ yields

$$\ell_{11}^{(\xi_1)}A^{(\xi_1)}\boldsymbol{v}_1 = h_{11}\boldsymbol{v}_1 + h_{21}\boldsymbol{v}_2. \tag{2.21}$$

Since $\boldsymbol{v}_1^*\boldsymbol{v}_1 = 1$ and $\boldsymbol{v}_1^*\boldsymbol{v}_2 = 0$, we have

$$h_{11} = \ell_{11}^{(\xi_1)}\boldsymbol{v}_1^*A^{(\xi_1)}\boldsymbol{v}_1. \tag{2.22}$$

We then have

$$\begin{aligned} h_{21}\boldsymbol{v}_2 &= \ell_{11}^{(\xi_1)}A^{(\xi_1)}\boldsymbol{v}_1 - h_{11}\boldsymbol{v}_1, \\ \boldsymbol{v}_2 &= \ell_{11}^{(\xi_1)}\big[A^{(\xi_1)}\boldsymbol{v}_1 - \big(\boldsymbol{v}_1^*A^{(\xi_1)}\boldsymbol{v}_1\big)\boldsymbol{v}_1\big]/h_{21}. \end{aligned}$$

Since $\|\boldsymbol{v}_2\|_2 = 1$ and $h_{21} \neq 0$ by assumption, we have $\ell_{11}^{(\xi_1)} \neq 0$. Analogously

$$\widehat{h}_{11} = \widehat{\ell}_{11}^{(\xi_1)}\boldsymbol{v}_1^*A^{(\xi_1)}\boldsymbol{v}_1, \quad \widehat{\boldsymbol{v}}_2 = \widehat{\ell}_{11}^{(\xi_1)}\big[A^{(\xi_1)}\boldsymbol{v}_1 - \big(\boldsymbol{v}_1^*A^{(\xi_1)}\boldsymbol{v}_1\big)\boldsymbol{v}_1\big]/\widehat{h}_{21}, \text{ and } \widehat{\ell}_{11}^{(\xi_1)} \neq 0.$$

Obviously, $\boldsymbol{v}_2$ and $\widehat{\boldsymbol{v}}_2$ are collinear and since they are both of unit 2-norm, there exists a unimodular scalar $d_2 \in \mathbb{C}$ such that $\widehat{\boldsymbol{v}}_2 = d_2 \boldsymbol{v}_2$. Defining $t_1 := \widehat{\ell}_{11}^{(\xi_1)}/\ell_{11}^{(\xi_1)}$, and $D_2 := \operatorname{diag}(d_1, d_2)$, and making use of $A^{(\xi_1)}\boldsymbol{v}_1 - \left(\boldsymbol{v}_1^* A^{(\xi_1)}\boldsymbol{v}_1\right)\boldsymbol{v}_1 = \widehat{h}_{21}\widehat{\boldsymbol{v}}_2/\widehat{\ell}_{11}^{(\xi_1)} = h_{21}\boldsymbol{v}_2/\ell_{11}^{(\xi_1)}$, we obtain $\underline{\widehat{H}_1} = D_2^* \underline{H_1} T_1$. From $\underline{K_1} = \underline{L_1^{(\xi_1)}} + \underline{H_1}/\xi_1$ and $\underline{\widehat{K}_1} = \underline{\widehat{L}_1^{(\xi_1)}} + \underline{\widehat{H}_1}/\xi_1$ we see that indeed $\underline{\widehat{K}_1} = D_2^* \underline{K_1} T_1$. This proves the statement for $j = 1$.

Suppose that, for $j = 2, 3, \ldots, m$, we have $\widehat{V}_j = V_j D_j$, $\underline{\widehat{H}_{j-1}} = D_j^* \underline{H_{j-1}} T_{j-1}$, and $\underline{\widehat{K}_{j-1}} = D_j^* \underline{K_{j-1}} T_{j-1}$, for a diagonal unitary matrix $D_j = \operatorname{diag}(d_1, d_2, \ldots, d_j)$ and upper triangular nonsingular matrix T_{j-1}.

The jth column in (2.20) for $\xi = \xi_j$ gives

$$A^{(\xi_j)} V_j \boldsymbol{l}_j^{(\xi_j)} = V_{j+1} \underline{\boldsymbol{h}_j}. \tag{2.23}$$

Since $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{j+1}$ are orthonormal we have

$$\boldsymbol{h}_j = V_j^* A^{(\xi_j)} V_j \boldsymbol{l}_j^{(\xi_j)}. \tag{2.24}$$

Rearranging the two equations above we deduce

$$\begin{aligned} h_{j+1,j}\boldsymbol{v}_{j+1} &= A^{(\xi_j)} V_j \boldsymbol{l}_j^{(\xi_j)} - V_j \boldsymbol{h}_j \\ &= A^{(\xi_j)} V_j \boldsymbol{l}_j^{(\xi_j)} - V_j V_j^* A^{(\xi_j)} V_j \boldsymbol{l}_j^{(\xi_j)} \\ &= \left(I - V_j V_j^*\right) A^{(\xi_j)} V_j \boldsymbol{l}_j^{(\xi_j)}. \end{aligned}$$

Expanding $\boldsymbol{l}_j^{(\xi_j)}$ as $\boldsymbol{l}_j^{(\xi_j)} =: \underline{L_{j-1}^{(\xi_j)}} \boldsymbol{z}_{j-1} + \boldsymbol{q}_j$, where $\boldsymbol{q}_j^* \underline{L_{j-1}^{(\xi_j)}} = \boldsymbol{0}^*$, gives

$$\begin{aligned} h_{j+1,j}\boldsymbol{v}_{j+1} &= \left(I - V_j V_j^*\right) A^{(\xi_j)} V_j \left(\underline{L_{j-1}^{(\xi_j)}} \boldsymbol{z}_{j-1} + \boldsymbol{q}_j\right) \\ &= \left(I - V_j V_j^*\right) A^{(\xi_j)} V_j \underline{L_{j-1}^{(\xi_j)}} \boldsymbol{z}_{j-1} + \left(I - V_j V_j^*\right) A^{(\xi_j)} V_j \boldsymbol{q}_j \\ &= \left(I - V_j V_j^*\right) A^{(\xi_j)} V_j \boldsymbol{q}_j. \end{aligned} \tag{2.25}$$

To obtain the last equality we have used $A^{(\xi_j)} V_j \underline{L_{j-1}^{(\xi_j)}} = V_j \underline{H_{j-1}}$, which are the first $j-1$ columns in (2.20) with $\xi = \xi_j$. Note that since $h_{j+1,j} \neq 0$ the vector $\boldsymbol{q}_j$ is also nonzero. We label analogously $\widehat{\boldsymbol{l}}_j^{(\xi_j)} =: \underline{\widehat{L}_{j-1}^{(\xi_j)}} \widehat{\boldsymbol{z}}_{j-1} + \widehat{\boldsymbol{q}}_j$, where $\widehat{\boldsymbol{q}}_j^* \underline{\widehat{L}_{j-1}^{(\xi_j)}} = \boldsymbol{0}^*$, and obtain

$$\begin{aligned} \widehat{h}_{j+1,j}\widehat{\boldsymbol{v}}_{j+1} &= \left(I - \widehat{V}_j \widehat{V}_j^*\right) A^{(\xi_j)} \widehat{V}_j \widehat{\boldsymbol{q}}_j, & \widehat{\boldsymbol{q}}_j^* \underline{\widehat{L}_{j-1}^{(\xi_j)}} &= \boldsymbol{0}^*, \\ \widehat{h}_{j+1,j}\widehat{\boldsymbol{v}}_{j+1} &= \left(I - V_j V_j^*\right) A^{(\xi_j)} V_j D_j \widehat{\boldsymbol{q}}_j, & \widehat{\boldsymbol{q}}_j^* D_j^* \underline{L_{j-1}^{(\xi_j)}} &= \boldsymbol{0}^*, \end{aligned}$$

where in the last equality above we have applied post-multiplication by T_{j-1}^{-1}. Since $\underline{L_{j-1}^{(\xi_j)}} \in \mathbb{C}^{j,j-1}$ is of full column rank, $\boldsymbol{q}_j$ and $D_j \widehat{\boldsymbol{q}}_j$ are collinear, i.e., there exists a

nonzero scalar $0 \neq \gamma \in \mathbb{C}$ such that $D_j \widehat{\boldsymbol{q}}_j = \gamma \boldsymbol{q}_j$. As a consequence $\boldsymbol{v}_{j+1}$ and $\widehat{\boldsymbol{v}}_{j+1}$ are collinear as well. Furthermore, as $\|\boldsymbol{v}_{j+1}\|_2 = \|\widehat{\boldsymbol{v}}_{j+1}\|_2 = 1$, there exists a unimodular scalar $d_{j+1} \in \mathbb{C}$ such that $\widehat{\boldsymbol{v}}_{j+1} = d_{j+1}\boldsymbol{v}_{j+1}$. We also observe $\widehat{h}_{j+1,j} = d_{j+1}^* \gamma h_{j+1,j}$.

It remains to find such a $\boldsymbol{t}_j \in \mathbb{C}^j$ that $T_j = [\underline{T_{j-1}} \quad \boldsymbol{t}_j]$ is nonsingular and that additionally $\underline{\widehat{H}_j} = D_{j+1}^* \underline{H_j} T_j$ and $\underline{\widehat{K}_j} = D_{j+1}^* \underline{K_j} T_j$. From $D_j \widehat{\boldsymbol{q}}_j = \gamma \boldsymbol{q}_j$ we infer

$$D_j \left(\widehat{\boldsymbol{l}}_j^{(\xi_j)} - \underline{\widehat{L}_{j-1}^{(\xi_j)}} \widehat{\boldsymbol{z}}_{j-1} \right) = \gamma \left(\boldsymbol{l}_j^{(\xi_j)} - \underline{L_{j-1}^{(\xi_j)}} \boldsymbol{z}_{j-1} \right),$$
$$\widehat{\boldsymbol{l}}_j^{(\xi_j)} = D_j^* \underline{L_{j-1}^{(\xi_j)}} \left(T_{j-1} \widehat{\boldsymbol{z}}_{j-1} - \gamma \boldsymbol{z}_{j-1} \right) + \gamma D_j^* \boldsymbol{l}_j^{(\xi_j)} = D_j^* L_j^{(\xi_j)} \boldsymbol{t}_j,$$

where $\boldsymbol{t}_j = \left[\begin{smallmatrix} T_{j-1}\widehat{\boldsymbol{z}}_{j-1} - \gamma \boldsymbol{z}_{j-1} \\ \gamma \end{smallmatrix} \right]$. Finally, using the equation above, the relation $\widehat{\boldsymbol{h}}_j = \widehat{V}_j^* A^{(\xi_j)} \widehat{V}_j \widehat{\boldsymbol{l}}_j^{(\xi_j)}$, and again $A^{(\xi_j)} V_j \underline{L_{j-1}^{(\xi_j)}} = V_j \underline{H_{j-1}}$, we derive $\widehat{\boldsymbol{h}}_j = D_j^* H_j \boldsymbol{t}_j$. With $\widehat{h}_{j+1,j} = d_{j+1}^* \gamma h_{j+1,j}$ we get $\underline{\widehat{H}_j} = D_{j+1}^* \underline{H_j} T_j$. We can consider $\underline{\widehat{K}_j}$ similarly. ∎

A further comment for the case $m = N-1$ is required. For the polynomial case, i.e., $\underline{K_{N-1}} = \underline{I_{N-1}}$, we have $AV_{N-1} = V_N \underline{H_{N-1}}$. The vector $\boldsymbol{h}_N = V_N^* A V_N \boldsymbol{e}_N$ is uniquely defined by the starting vector and A and $AV_N = V_N H_N$ holds. This last decomposition is usually stated as the (polynomial) implicit Q theorem and essential uniqueness of H_N is claimed. Let us consider a more general RAD, namely, $AV_N \underline{K_{N-1}} = V_N \underline{H_{N-1}}$. Defining $\boldsymbol{h}_N := V_N^* A V_N \boldsymbol{k}_N$ for an arbitrary $\boldsymbol{k}_N \in \mathbb{C}^N$ we see that $AV_N K_N = V_N H_N$. Therefore we cannot say that $\left(H_N, K_N\right)$ is essentially unique. In fact, essential uniqueness is related to both V_{m+1} and the pencil $\left(\underline{H_m}, \underline{K_m}\right)$ concurrently.

In practice, the (rational) implicit Q theorem is useful as it allows for certain transformations of RADs to be performed at a reduced computational cost. Such transformations consist of two steps. First, the transformation is applied to the reduced pencil (instead of the operator A), and second, the RAD structure is recovered and reinterpreted using our Theorem 2.16.

As already mentioned, a polynomial Krylov space $\mathcal{K}_{m+1}(A, \boldsymbol{b})$ with orthonormal basis V_{m+1} is related to a decomposition of the form

$$AV_m = V_{m+1} \underline{H_m} = V_m H_m + h_{m+1,m} \boldsymbol{v}_{m+1} \boldsymbol{e}_m^T, \tag{2.26}$$

where H_m is upper Hessenberg. For a rational Krylov space we have an RAD (2.6) with an upper Hessenberg pencil $(\underline{H_m}, \underline{K_m})$ rather than a single upper Hessenberg matrix $\underline{H_m}$. It has been shown, for example in [36, 77, 107, 111], that decompositions of the form (2.26) with H_m being *semiseparable plus diagonal* (have a particular rank

structure, see the aforementioned references) are related to rational Krylov spaces in the same way as RADs are. Corresponding implicit Q theorems have been developed. The semiseparable structure can be used to develop (short) recurrences related to rational Krylov spaces, see for instance [66, 83]. In this thesis we do not dwell upon these considerations here.

2.4 Complex poles for real-valued matrices

When working with a real-valued matrix $A \in \mathbb{R}^{N,N}$ and a real-valued starting vector $\boldsymbol{b} \in \mathbb{R}^N$, it may be beneficial to consider real-valued RADs. For instance, if $\lambda \in \mathbb{C} \setminus \mathbb{R}$ is an eigenvalue of A, then so is $\overline{\lambda}$. By enforcing (2.6) to be real-valued, we preserve this structure and can obtain approximate eigenvalues from (2.6) that also appear in complex-conjugate pairs. We discuss the extraction of approximate eigenpairs from (2.6) in Section 3.1. Clearly, if $\mu_j, \nu_j, \eta_j, \rho_j \in \mathbb{R}$, and $\boldsymbol{t}_j \in \mathbb{R}^j$, then Algorithm 2.2 produces a real-valued RAD. However, even with complex-valued poles it is possible to obtain real-valued decomposition of the form (2.6), provided that the poles appear in complex-conjugate pairs. This was introduced in [87], and we now review it. Our approach is, however, slightly more general; it incorporates continuation roots η_j/ρ_j, and further, we allow for complex-valued continuation vectors $\boldsymbol{t}_j$, and not only real-valued as in [87].

2.4.1. Real-valued rational Arnoldi algorithm. We wish to extend a real-valued decomposition of the form

$$AV_j\underline{K_{j-1}} = V_j\underline{H_{j-1}}, \tag{2.27}$$

with $j-1 < d(A, \boldsymbol{b}) - 1$, with the pole $\xi_j := z \in \overline{\mathbb{C}} \setminus \Lambda(A)$. If ξ_j is actually real-valued or infinity, we can proceed as in Algorithm 2.2 in order to obtain a decomposition of order j. Otherwise, we simultaneously extend (2.27) from order $j-1$ to order $j+1$ adding the pole ξ_j and its complex-conjugate $\xi_{j+1} := \overline{z}$, as we now explain. To this end, let

$$\mu_j, \eta_j \in \mathbb{C}, \nu_j, \rho_j \in \mathbb{R} \quad \text{satisfy} \quad \mu_j\rho_j \neq \nu_j\eta_j, \quad \text{and} \quad \mu_j/\nu_j = \xi_j \notin \Lambda(A), \tag{2.28}$$

and let $\boldsymbol{t}_j \in \mathbb{C}^j$. Define the vector

$$\boldsymbol{w}_{j+1} := (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)V_j\boldsymbol{t}_j, \tag{2.29}$$

and note that $(\boldsymbol{w}_{j+1}^T)^* = (\nu_j A - \overline{\mu_j} I)^{-1}(\rho_j A - \overline{\eta_j} I) V_j (\boldsymbol{t}_j^T)^*$. One can verify that

$$\Re(\boldsymbol{w}_{j+1}) = \frac{1}{2}\left[\boldsymbol{w}_{j+1} + (\boldsymbol{w}_{j+1}^T)^*\right], \quad \text{and} \quad \Im(\boldsymbol{w}_{j+1}) = \frac{1}{2\mathrm{i}}\left[\boldsymbol{w}_{j+1} - (\boldsymbol{w}_{j+1}^T)^*\right], \tag{2.30}$$

and hence $\operatorname{span}\{\boldsymbol{w}_{j+1}, (\boldsymbol{w}_{j+1}^T)^*\} = \operatorname{span}\{\Re(\boldsymbol{w}_{j+1}), \Im(\boldsymbol{w}_{j+1})\}$. Therefore, we can add the real-valued vectors $\Re(\boldsymbol{w}_{j+1})$ and $\Im(\boldsymbol{w}_{j+1})$ to the basis V_j, instead of the complex-valued $\boldsymbol{w}_{j+1}$ and $(\boldsymbol{w}_{j+1}^T)^*$. Consequently, let $\Re(\boldsymbol{w}_{j+1}) =: V_{j+1}\underline{\boldsymbol{c}_j}$, and $\Im(\boldsymbol{w}_{j+1}) =: V_{j+2}\underline{\boldsymbol{c}_{j+1}}$, with $V_{j+2}^* V_{j+2} = I_{j+2}$. From (2.29) we arrive at

$$(\nu_j A - \mu_j I) V_{j+2}\big(\underline{\underline{\boldsymbol{c}_j}} + \mathrm{i}\underline{\boldsymbol{c}_{j+1}}\big) = (\rho_j A - \eta_j I) V_j \boldsymbol{t}_j, \tag{2.31}$$

where $\underline{\underline{\boldsymbol{c}_j}} := [\underline{\boldsymbol{c}_j}^T \;\; 0]^T$. Rearanging the terms with and without A we have

$$A V_{j+2}\big(\nu_j \underline{\underline{\boldsymbol{c}_j}} + \mathrm{i}\nu_j \underline{\boldsymbol{c}_{j+1}} - \rho_j \underline{\underline{\boldsymbol{t}_j}}\big) = V_{j+2}\big(\mu_j \underline{\underline{\boldsymbol{c}_j}} + \mathrm{i}\mu_j \underline{\boldsymbol{c}_{j+1}} - \eta_j \underline{\underline{\boldsymbol{t}_j}}\big), \tag{2.32}$$

where $\underline{\underline{\boldsymbol{t}_j}} := [\underline{\boldsymbol{t}_j}^T \;\; 0 \;\; 0]^T$. Finally, we add separately the real part and the imaginary part of (2.32) as two real-valued columns to (2.27):

$$A V_{j+2}\begin{bmatrix}\underline{\underline{\boldsymbol{k}_j}} & \underline{\boldsymbol{k}_{j+1}}\end{bmatrix} = V_{j+2}\begin{bmatrix}\underline{\underline{\boldsymbol{h}_j}} & \underline{\boldsymbol{h}_{j+1}}\end{bmatrix}, \tag{2.33}$$

where, under the abbreviations $x^{\Re} \equiv \Re(x)$ and $x^{\Im} \equiv \Im(x)$,

$$\begin{bmatrix}\underline{\underline{\boldsymbol{k}_j}} & \underline{\boldsymbol{k}_{j+1}}\end{bmatrix} := \begin{bmatrix}\underline{\underline{\boldsymbol{c}_j}} & \underline{\boldsymbol{c}_{j+1}}\end{bmatrix}\begin{bmatrix}\nu_j & \\ & \nu_j\end{bmatrix} - \begin{bmatrix}\underline{\underline{\boldsymbol{t}_j^{\Re}}} & \underline{\underline{\boldsymbol{t}_j^{\Im}}}\end{bmatrix}\begin{bmatrix}\rho_j & \\ & \rho_j\end{bmatrix}, \quad \text{and} \tag{2.34}$$

$$\begin{bmatrix}\underline{\underline{\boldsymbol{h}_j}} & \underline{\boldsymbol{h}_{j+1}}\end{bmatrix} := \begin{bmatrix}\underline{\underline{\boldsymbol{c}_j}} & \underline{\boldsymbol{c}_{j+1}}\end{bmatrix}\begin{bmatrix}\mu_j^{\Re} & \mu_j^{\Im} \\ -\mu_j^{\Im} & \mu_j^{\Re}\end{bmatrix} - \begin{bmatrix}\underline{\underline{\boldsymbol{t}_j^{\Re}}} & \underline{\underline{\boldsymbol{t}_j^{\Im}}}\end{bmatrix}\begin{bmatrix}\eta_j^{\Re} & \eta_j^{\Im} \\ -\eta_j^{\Im} & \eta_j^{\Re}\end{bmatrix}. \tag{2.35}$$

Here $\underline{\underline{\boldsymbol{h}_j}} := [\underline{\boldsymbol{h}_j}^T \;\; 0]^T$, and analogously for $\underline{\underline{\boldsymbol{k}_j}}$. It follows from (2.34)–(2.35) that $\left(\left[\begin{smallmatrix} h_{j+1,j} & h_{j+1,j+1} \\ h_{j+2,j} & h_{j+2,j+1}\end{smallmatrix}\right], \left[\begin{smallmatrix} k_{j+1,j} & k_{j+1,j+1} \\ & k_{j+2,j+1}\end{smallmatrix}\right]\right)$ equals

$$\left(\begin{bmatrix} c_{j+1,j} & c_{j+1,j+1} \\ & c_{j+2,j+1}\end{bmatrix}\begin{bmatrix}\mu_j^{\Re} & \mu_j^{\Im} \\ -\mu_j^{\Im} & \mu_j^{\Re}\end{bmatrix}, \begin{bmatrix} c_{j+1,j} & c_{j+1,j+1} \\ & c_{j+2,j+1}\end{bmatrix}\begin{bmatrix}\nu_j & \\ & \nu_j\end{bmatrix}\right), \tag{2.36}$$

with $\left[\begin{smallmatrix} c_{j+1,j} & c_{j+1,j+1} \\ & c_{j+2,j+1}\end{smallmatrix}\right]$ being nonsingular if $c_{\ell+1,\ell} \neq 0$ for $\ell = j, j+1$, which is indeed the case if no breakdown occurs during the rational Arnoldi algorithm. Consequently, the eigenvalues of the 2-by-2 pencil $\left(\left[\begin{smallmatrix} h_{j+1,j} & h_{j+1,j+1} \\ h_{j+2,j} & h_{j+2,j+1}\end{smallmatrix}\right], \left[\begin{smallmatrix} k_{j+1,j} & k_{j+1,j+1} \\ & k_{j+2,j+1}\end{smallmatrix}\right]\right)$ are ξ_j and $\overline{\xi_j}$.

The rational Arnoldi algorithm with possibly complex-valued poles for real-valued matrices is presented in Algorithm 2.3.

Algorithm 2.3 Real-valued rational Arnoldi algorithm. RKToolbox: `rat_krylov`

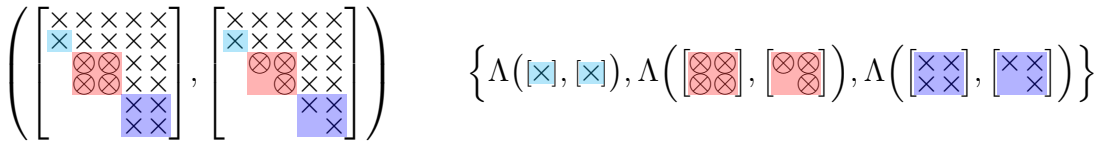
Input: $A \in \mathbb{R}^{N,N}$, $\boldsymbol{b} \in \mathbb{R}^N$, a set of poles $\{\mu_j/\nu_j\}_{j=1}^m \subset \overline{\mathbb{C}} \setminus \Lambda(A)$ closed under complex-conjugation, with complex-conjugate pairs of poles labelled with successive indices, and such that $\{\nu_j\}_{j=1}^m \subset \mathbb{R}$, and $\mu_j \in \mathbb{R}$ if $\nu_j = 0$, for $j = 1, 2, \ldots, m < d(A, \boldsymbol{b})$.
Output: Decomposition $AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}$, with $V_{m+1}^* V_{m+1} = I_{m+1}$.

1. Set $\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|_2$, and let $j = 1$.
2. **while** $j \leq m$ **do**
3. **if** $\mu_j \in \mathbb{R}$ **then** Choose admissible $(\eta_j/\rho_j, \boldsymbol{t}_j) \in \overline{\mathbb{R}} \times \mathbb{R}^j$, with $\rho_j \in \mathbb{R}$.
4. **else** Choose admissible $(\eta_j/\rho_j, \boldsymbol{t}_j) \in \overline{\mathbb{C}} \times \mathbb{C}^j$, with $\rho_j \in \mathbb{R}$. **end if**
5. Compute $\boldsymbol{w}_{j+1} := (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)V_j \boldsymbol{t}_j$.
6. **if** $\mu_j \in \mathbb{R}$ **then**
7. Orthogonalize $\widehat{\boldsymbol{v}}_{j+1} := \boldsymbol{w}_{j+1} - V_j \boldsymbol{c}_j$, where $\boldsymbol{c}_j := V_j^* \boldsymbol{w}_{j+1}$.
8. Normalize $\boldsymbol{v}_{j+1} := \widehat{\boldsymbol{v}}_{j+1}/c_{j+1,j}$, where $c_{j+1,j} := \|\widehat{\boldsymbol{v}}_{j+1}\|_2$.
9. Set $\underline{\boldsymbol{k}_j} := \nu_j \underline{\boldsymbol{c}_j} - \rho_j \underline{\boldsymbol{t}_j}$ and $\underline{\boldsymbol{h}_j} := \mu_j \underline{\boldsymbol{c}_j} - \eta_j \underline{\boldsymbol{t}_j}$, where $\underline{\boldsymbol{t}_j} = \begin{bmatrix} \boldsymbol{t}_j \\ 0 \end{bmatrix}$, $\underline{\boldsymbol{c}_j} = \begin{bmatrix} \boldsymbol{c}_j \\ c_{j+1,j} \end{bmatrix}$.
10. Update $j := j + 1$.
11. **else**
12. Orthogonalize $\widehat{\boldsymbol{v}}_{j+1} := \Re(\boldsymbol{w}_{j+1}) - V_j \boldsymbol{c}_j$, where $\boldsymbol{c}_j := V_j^* \Re(\boldsymbol{w}_{j+1})$.
13. Normalize $\boldsymbol{v}_{j+1} := \widehat{\boldsymbol{v}}_{j+1}/c_{j+1,j}$, where $c_{j+1,j} := \|\widehat{\boldsymbol{v}}_{j+1}\|_2$.
14. Orthogonalize $\widehat{\boldsymbol{v}}_{j+2} := \Im(\boldsymbol{w}_{j+1}) - V_{j+1} \boldsymbol{c}_{j+1}$, where $\boldsymbol{c}_{j+1} := V_{j+1}^* \Im(\boldsymbol{w}_{j+1})$.
15. Normalize $\boldsymbol{v}_{j+2} := \widehat{\boldsymbol{v}}_{j+2}/c_{j+2,j+1}$, where $c_{j+2,j+1} := \|\widehat{\boldsymbol{v}}_{j+2}\|_2$.
16. Define $\underline{\boldsymbol{k}_\ell}$, and $\underline{\boldsymbol{h}_\ell}$, for $\ell = j, j+1$, as in (2.34)–(2.35), where $\underline{\boldsymbol{c}_\ell} = \begin{bmatrix} \boldsymbol{c}_\ell \\ c_{\ell+1,\ell} \end{bmatrix}$.
17. Update $j := j + 2$.
18. **end if**
19. **end while**

2.4.2. Quasi-RADs and the related implicit Q theorem. The real-valued decomposition (2.6) Algorithm 2.3 produces has a specific structure; it is such that the pencil $(\underline{H_{-m}}, \underline{K_{-m}})$ is in generalised real Schur form, that is, $\underline{H_{-m}} \in \mathbb{R}^{m,m}$ is upper quasi-triangular, while $\underline{K_{-m}} \in \mathbb{R}^{m,m}$ is upper triangular. This is why we insist on $\nu_j \in \mathbb{R}$, since if ν_j were complex-valued, $\underline{K_{-m}}$ would be upper quasi-triangular as well. Imposing this canonical form allows us to use well established algorithms from the literature. Hence, if we denote

$$\underline{H_m} = \begin{bmatrix} H_{11} & H_{12} & \ldots & H_{1\check{\ell}} & H_{1\ell} \\ H_{21} & H_{22} & \ldots & H_{2\check{\ell}} & H_{2\ell} \\ & H_{32} & \ldots & H_{3\check{\ell}} & H_{3\ell} \\ & & \ddots & \vdots & \vdots \\ & & & H_{\ell\check{\ell}} & H_{\ell\ell} \\ & & & & H_{\hat{\ell}\ell} \end{bmatrix}, \quad \underline{K_m} = \begin{bmatrix} K_{11} & K_{12} & \ldots & K_{1\check{\ell}} & K_{1\ell} \\ K_{21} & K_{22} & \ldots & K_{2\check{\ell}} & K_{2\ell} \\ & K_{32} & \ldots & K_{3\check{\ell}} & K_{3\ell} \\ & & \ddots & \vdots & \vdots \\ & & & K_{\ell\check{\ell}} & K_{\ell\ell} \\ & & & & K_{\hat{\ell}\ell} \end{bmatrix}, \tag{2.37}$$

with $\check{\ell} := \ell - 1$, and $\hat{\ell} := \ell + 1$, then the real-valued pencils $(H_{\hat{\jmath}\jmath}, K_{\hat{\jmath}\jmath})$, with $\hat{\jmath} := j + 1$, are either 1-by-1 or 2-by-2, while all the blocks H_{1j} and K_{1j} have one row only. We call RKDs having this structure *quasi-RADs*.



(a) Nonzero pattern of $(\underline{H}_m, \underline{K}_m)$. (b) Poles of the quasi-RAD.

Figure 2.2: Nonzero pattern of $(\underline{H}_m, \underline{K}_m)$ from a quasi-RAD, with $m = 5$. The quasi-RAD has one real-valued and four complex-valued poles (two complex-conjugate pairs). The poles are the generalized eigenvalues of the 1-by-1 or 2-by-2 matrix pencils from the (block) subdiagonal of $(\underline{H}_m, \underline{K}_m)$.

Definition 2.17. *Let $A \in \mathbb{R}^{N,N}$, and let (2.6) hold. If $V_{m+1}, \underline{K}_m$ and $\underline{H}_m$ are real-valued, and $(\underline{H}_{-m}, \underline{K}_{-m})$ is a regular pencil in generalised real Schur form, then we call (2.6) a* quasi-RAD of order m.

Note that requiring for the pencil $(\underline{H}_{-m}, \underline{K}_{-m})$ to be regular is the natural generalisation of $(\underline{H}_m, \underline{K}_m)$ being unreduced in case of RADs. Indeed, if the pencil $(\underline{H}_{-m}, \underline{K}_{-m})$ is regular and in generalised Schur form, then and only then is $(\underline{H}_m, \underline{K}_m)$ unreduced. The notions of (*orthonormal*) *basis, space* and *equivalent decompositions* are analogous to those for RADs. A possible structure of the reduced pencil $(\underline{H}_m, \underline{K}_m)$ is depicted graphically for $m = 5$ in Figure 2.2. The following generalisation of Lemma 2.6 holds.

Corollary 2.18. *Let (2.6) be a (quasi-)RAD, and let $\alpha, \beta \in \mathbb{C}$ be such that $|\alpha|+|\beta| \neq 0$. The matrix $\alpha\underline{H}_m - \beta\underline{K}_m$ is of full column rank m.*

Proof. Follows from Corollary 5.5 which we discuss later. □

When referring to the block structure of a quasi-RAD (2.6), we mean the block structure (2.37) of $(\underline{H}_m, \underline{K}_m)$, $\underline{H}_m$, or $\underline{H}_{-m}$, depending on the context. Let us now generalise the notion of *essential uniqueness*.

Definition 2.19. *Let (2.6) and (2.8) be two orthonormal quasi-RADs with the same block structure. The two quasi-RADs are called* essentially equal *if there exist an orthogonal block-diagonal matrix $D_{m+1} = \operatorname{blkdiag}(d_1, D_m) \in \mathbb{R}^{m+1,m+1}$, with $D_m \in \mathbb{R}^{m,m}$ having the same block structure as (2.6), and an upper quasi-triangular nonsingular matrix $T_m \in \mathbb{R}^{m,m}$ with the same block structure as (2.6), such that $\widehat{V}_{m+1} = V_{m+1}D_{m+1}$, $\underline{\widehat{H}}_m = D_{m+1}^* \underline{H}_m T_m$, and $\underline{\widehat{K}}_m = D_{m+1}^* \underline{K}_m T_m$. Essentially equal orthonormal quasi-RADs form an equivalence class and we call any of its elements* essentially unique.

Let us clarify the block ordering of the poles of a quasi-RAD (2.6) with block structure (2.37). We refer to $\Lambda(H_{\hat{\jmath}\hat{\jmath}}, K_{\hat{\jmath}\hat{\jmath}})$ as the jth block pole of (2.6), for $j = 1, 2, \ldots, \ell$. We can now formulate the corresponding rational implicit Q theorem.

Theorem 2.20. *Let* (2.6) *be an orthonormal quasi-RAD. The orthonormal matrix* V_{m+1} *and the pencil* $(\underline{H_m}, \underline{K_m})$ *are essentially uniquely determined by* $V_{m+1}\boldsymbol{e}_1$ *and the block ordering of the poles of* (2.6).

Proof. Let (2.8) be an orthonormal quasi-RAD with equal block structure (2.37) and equal block ordering of the poles as (2.6), and with starting vector colinear to that of (2.6). Let unitary $Q_{\hat{\jmath}\hat{\jmath}}, Z_{\hat{\jmath}\hat{\jmath}}, \widehat{Q}_{\hat{\jmath}\hat{\jmath}}$, and $\widehat{Z}_{\hat{\jmath}\hat{\jmath}}$ be such that $(Q_{\hat{\jmath}\hat{\jmath}}^* H_{\hat{\jmath}\hat{\jmath}} Z_{\hat{\jmath}\hat{\jmath}}, Q_{\hat{\jmath}\hat{\jmath}}^* K_{\hat{\jmath}\hat{\jmath}} Z_{\hat{\jmath}\hat{\jmath}})$ and $(\widehat{Q}_{\hat{\jmath}\hat{\jmath}}^* \widehat{H}_{\hat{\jmath}\hat{\jmath}} \widehat{Z}_{\hat{\jmath}\hat{\jmath}}, \widehat{Q}_{\hat{\jmath}\hat{\jmath}}^* \widehat{K}_{\hat{\jmath}\hat{\jmath}} \widehat{Z}_{\hat{\jmath}\hat{\jmath}})$ are in generalised Schur form, with generalised eigenvalues ordered in the same way, for $j = 1, 2, \ldots, \ell$. Define $D_{m+1} := \mathrm{blkdiag}(1, Q_{21}, Q_{32}, \ldots, Q_{\hat{\ell}\ell})$, and $T_m := \mathrm{blkdiag}(Z_{21}, Z_{32}, \ldots, Z_{\hat{\ell}\ell})$. Let $\widehat{D}_{m+1}$ and $\widehat{T}_m$ be defined analogously. By Theorem 2.16 the complex-valued orthonormal RAD

$$A\big(V_{m+1}D_{m+1}\big)\big(D_{m+1}^*\underline{K_m}T_m\big) = \big(V_{m+1}D_{m+1}\big)\big(D_{m+1}^*\underline{H_m}T_m\big)$$

is essentially equal to the complex-valued orthonormal RAD

$$A\big(\widehat{V}_{m+1}\widehat{D}_{m+1}\big)\big(\widehat{D}_{m+1}^*\underline{\widehat{K}_m}\widehat{T}_m\big) = \big(\widehat{V}_{m+1}\widehat{D}_{m+1}\big)\big(\widehat{D}_{m+1}^*\underline{\widehat{H}_m}\widehat{T}_m\big).$$

Therefore, there exist unitary $D \in \mathbb{C}^{m+1,m+1}$ and nonsingular upper triangular $T \in \mathbb{C}^{m,m}$ such that $\widehat{V}_{m+1}\widehat{D}_{m+1} = V_{m+1}D_{m+1}D$, $\widehat{D}_{m+1}^*\underline{\widehat{H}_m}\widehat{T}_m = D^*D_{m+1}^*\underline{H_m}T_mT$, and analogously for $\underline{\widehat{K}_m}$. Consequently, $\widehat{V}_{m+1} = V_{m+1}D_{m+1}D\widehat{D}_{m+1}^*$,

$$\underline{\widehat{H}_m} = \widehat{D}_{m+1}D^*D_{m+1}^*\underline{H_m}T_mT\widehat{T}_m^{-1}, \quad \text{and} \quad \underline{\widehat{K}_m} = \widehat{D}_{m+1}D^*D_{m+1}^*\underline{K_m}T_mT\widehat{T}_m^{-1}.$$

In particular, $D_{m+1}D\widehat{D}_{m+1}^* = V_{m+1}^*\widehat{V}_{m+1} \in \mathbb{R}^{m+1,m+1}$ has the required block structure. Furthermore, since $\underline{\widehat{K}_m}$ and $\underline{K_m}$ are both real-valued, we have that $T_mT\widehat{T}_m^{-1} \in \mathbb{R}^{m,m}$, and it is upper quasi-triangular with the desired block structure. ∎

2.5 Matrix pencils and nonstandard inner products

So far we have been working with a single matrix A, however, rational Krylov spaces were initially proposed for the purpose of solving large sparse generalised eigenvalue problems $A\boldsymbol{x} = \lambda B\boldsymbol{x}$, with $A, B \in \mathbb{C}^{N,N}$; see [86, 89, 87, 90]. We now comment on the possibility of using a pencil (A, B), instead of A only. In applications, a pencil (A, B) may arise, for instance, after the discretisation of certain partial differential equations, in which case it may also be convenient to use a nonstandard inner product;

see [40, Section 6.3] for an example. Our goal is to provide a simple way to reinterpret the related RADs with (A, B) as RADs of the form (2.6), which in turn allows to use the theory developed so far. This viewpoint, presented in Section 2.5.1, as well as the neat definition, cf. (2.40) below, of the corresponding rational Krylov space $\mathcal{Q}_{m+1}(A, B, \boldsymbol{b}, q_m)$ is new. In Section 2.5.2 we focus on nonstandard inner products.

2.5.1. Matrix pencils. The rational Arnoldi algorithm can easily be adapted to handle N-by-N matrix pencils (A, B) instead of a single matrix A. It is enough to replace $(\nu_j A-\mu_j I)^{-1}(\rho_j A-\eta_j I)$ at line 4 of Algorithm 2.2, or at line 5 of Algorithm 2.3, with $(\nu_j A - \mu_j B)^{-1}(\rho_j A - \eta_j B)$, where the poles μ_j/ν_j satisfy $\mu_j/\nu_j \notin \Lambda(A, B)$. The (quasi-)RAD (2.6) takes the form

$$AV_{m+1}\underline{K_m} = BV_{m+1}\underline{H_m}, \tag{2.38}$$

and we say that is is a (quasi-)RAD of order m, provided that the requirements analogous to those for (2.6) are fulfilled. Furthermore, the notion of (*orthonormal*) *basis, space, equivalent* and *essentially unique decomposition* is the same as for RADs and quasi-RADs of the form (2.6). Let

$$\mu, \nu, \eta, \rho \in \mathbb{C} \quad \text{satisfy} \quad \mu\rho \neq \eta\nu, \quad \text{and} \quad \mu/\nu \notin \Lambda(A, B). \tag{2.39}$$

Introduce formally $\mathcal{M}(\alpha, \beta) = (\nu\alpha - \mu\beta)^{-1}(\rho\alpha - \eta\beta)$. Thus, for matrices $A, B \in \mathbb{C}^{N,N}$ we have $\mathcal{M}(A, B) = (\nu A - \mu B)^{-1}(\rho A - \eta B)$. We can now define the rational Krylov space $\mathcal{R}(V_{m+1})$ spanned by (2.38) as

$$\mathcal{Q}_{m+1}(A, B, \boldsymbol{b}, q_m) := \mathcal{Q}_{m+1}\left(\mathcal{M}(A, B), \boldsymbol{b}, \{\mathcal{M}(\mu_j, \nu_j)\}_{j=1}^{m}\right), \tag{2.40}$$

where $\{\mu_j/\nu_j\}_{j=1}^{m}$ are the formal roots of q_m. To show that $\mathcal{Q}_{m+1}(A, B, \boldsymbol{b}, q_m)$ is well defined, we need to show that the right-hand side of (2.40) is well defined, and that it is independent of the choice for $\mathcal{M}$. To this end, from (2.38) we obtain

$$(\rho A - \eta B)V_{m+1}(\nu\underline{H_m} - \mu\underline{K_m}) = (\nu A - \mu B)V_{m+1}(\rho\underline{H_m} - \eta\underline{K_m}), \tag{2.41}$$

and since $\nu A - \mu B$ is nonsingular, as $\mu/\nu \notin \Lambda(A, B)$, we have

$$\mathcal{M}(A, B)V_{m+1}(\nu\underline{H_m} - \mu\underline{K_m}) = V_{m+1}(\rho\underline{H_m} - \eta\underline{K_m}). \tag{2.42}$$

Note that (2.42) is an RAD for $\mathcal{Q}_{m+1}(\mathcal{M}(A,B),\boldsymbol{b},\{\mathcal{M}(\mu_j,\nu_j)\}_{j=1}^m)$, if (2.38) is an RAD, and not a quasi-RAD (we return to quasi-RADs later).

Indeed, V_{m+1} is of full column rank, and $(\rho\underline{H_m}-\eta\underline{K_m},\nu\underline{H_m}-\mu\underline{K_m})$ is an unreduced upper Hessenberg pencil, since $(\underline{H_m},\underline{K_m})$ is, and $\mu\rho\neq\eta\nu$. To show the latter, assume, to the contrary, that $|\rho h_{j+1,j}-\eta k_{j+1,j}|+|\nu h_{j+1,j}-\mu k_{j+1,j}|=0$ for at least one index $j\in\{1,2,\ldots,m\}$. Then $\rho h_{j+1,j}-\eta k_{j+1,j}=\nu h_{j+1,j}-\mu k_{j+1,j}=0$, and in particular $\nu h_{j+1,j}=\mu k_{j+1,j}$. Using the latter in, firstly, $\rho h_{j+1,j}-\eta k_{j+1,j}=0$ multiplied by ν, and, secondly, $\rho h_{j+1,j}-\eta k_{j+1,j}=0$ multiplied by μ, gives $k_{j+1,j}(\rho\mu-\nu\eta)=0$, and $h_{j+1,j}(\rho\mu-\nu\eta)=0$, respectively. Since $\mu\rho\neq\eta\nu$ we have $k_{j+1,j}=0$ and $h_{j+1,j}=0$, which is in contradiction with $(\underline{H_m},\underline{K_m})$ being unreduced.

Lastly, since $\mu_j/\nu_j\in\Lambda(A,B)$ if and only if $\mathcal{M}(\mu_j,\nu_j)\in\Lambda(\mathcal{M}(A,B))$, the poles of (2.42) are allowed. Further, (2.42) spans the same space as (2.38) independently of $\mathcal{M}$. The terminology of *is an RAD for* transfers analogously from RADs of the form (2.6) to those of the form (2.38). Lastly, we can define $d(A,B,\boldsymbol{b}):=d(\mathcal{M}(A,B),\boldsymbol{b})$. Let us summarise these observations.

Proposition 2.21. *Let $A,B\in\mathbb{C}^{N,N}$ be complex-valued matrices, and $\mu,\nu,\eta,\rho\in\mathbb{C}$ scalars such that $\mu\rho\neq\eta\nu$, and $\mu/\nu\notin\Lambda(A,B)$. Let $\mathcal{M}(\alpha,\beta)\equiv(\rho\alpha-\eta\beta)/(\nu\alpha-\mu\beta)$. The decomposition (2.38) is an RAD for $\mathcal{Q}_{m+1}\left(A,B,\boldsymbol{b},\{\mu_j/\nu_j\}_{j=1}^m\right)$ if and only if (2.42) is an RAD for $\mathcal{Q}_{m+1}\left(\mathcal{M}(A,B),\boldsymbol{b},\{\mathcal{M}(\mu_j,\nu_j)\}_{j=1}^m\right)$.*

Let us now return to quasi-RADs (2.38). The corresponding (2.42) may fail to be a quasi-RAD if the decomposition is not real-valued, or if $\nu\underline{H_m}-\mu\underline{K_m}$ has 2-by-2 blocks on the subdiagonal. The first problem is easily solved by considering only $\mu,\nu,\eta,\rho\in\mathbb{R}$. Regarding the structure, it can be recovered by bringing $(\underline{H_{-m}},\underline{K_{-m}})$ into the generalised real Schur form. Thus (2.42) can be transformed into the equivalent quasi-RAD

$$\mathcal{M}(A,B)\breve{V}_{m+1}\underline{\breve{K}_m}=\breve{V}_{m+1}\underline{\breve{H}_m},\tag{2.43}$$

where

$$\begin{aligned}\breve{V}_{m+1}&:=V_{m+1}Q_{m+1}, & Q_{m+1}&:=\operatorname{blkdiag}(1,Q_m),\\ \underline{\breve{K}_m}&:=Q_{m+1}^*(\nu\underline{H_m}-\mu\underline{K_m})Z_m, \quad\text{and}\quad & \underline{\breve{H}_m}&:=Q_{m+1}^*(\rho\underline{H_m}-\eta\underline{K_m})Z_m,\end{aligned}\tag{2.44}$$

with orthogonal matrices $Q_m,Z_m\in\mathbb{R}^{m,m}$ being such that $(\underline{\breve{H}_{-m}},\underline{\breve{K}_{-m}})$ is in generalised real Schur form. The analogue to Proposition 2.21 can now be easily established.

Proposition 2.22. *Let $A, B \in \mathbb{R}^{N,N}$ be real-valued matrices, and $\mu, \nu, \eta, \rho \in \mathbb{R}$ scalars such that $\mu\rho \neq \eta\nu$, and $\mu/\nu \notin \Lambda(A, B)$. Let $\mathcal{M}(\alpha, \beta) \equiv (\rho\alpha - \eta\beta)/(\nu\alpha - \mu\beta)$. The decomposition (2.38) is a quasi-RAD for $\mathcal{Q}_{m+1}\left(A, B, \boldsymbol{b}, \{\mu_j/\nu_j\}_{j=1}^m\right)$ if and only if (2.43) is a quasi-RAD for $\mathcal{Q}_{m+1}\left(\mathcal{M}(A, B), \boldsymbol{b}, \{\mathcal{M}(\mu_j, \nu_j)\}_{j=1}^m\right)$ such that (2.44) holds.*

2.5.2. Nonstandard inner products. A nonstandard inner product $\langle\cdot|\cdot\rangle : \mathbb{C}^N \times \mathbb{C}^N \to \mathbb{C}$ may easily be employed for the Gram–Schmidt process. For convenience, we introduce the notation $\langle\cdot,\cdot\rangle : \mathbb{C}^{N,k} \times \mathbb{C}^{N,n} \to \mathbb{C}^{n,k}$, for varying $k, n \in \mathbb{N}$, by defining $\boldsymbol{e}_i^* \langle X_k, Y_n\rangle \boldsymbol{e}_j := \langle X_k \boldsymbol{e}_j | Y_n \boldsymbol{e}_i\rangle$, for $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, k$. Simply replacing $\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|_2$ by $\boldsymbol{v}_1 := \boldsymbol{b}/\sqrt{\langle \boldsymbol{b}, \boldsymbol{b}\rangle}$ in line 1 of Algorithm 2.2, $\boldsymbol{c}_j := V_j^* \boldsymbol{w}_{j+1}$ by $\boldsymbol{c}_j := \langle \boldsymbol{w}_{j+1}, V_j\rangle$ in line 5, and $c_{j+1,j} := \|\boldsymbol{v}_{j+1}\|_2$ by $c_{j+1,j} := \sqrt{\langle \boldsymbol{w}_{j+1}, \boldsymbol{w}_{j+1}\rangle}$ in line 6, incorporates the desired inner product within the rational Arnoldi algorithm. We shall call $\langle\cdot,\cdot\rangle$ an inner product, even though strictly speaking only $\langle\cdot|\cdot\rangle$ is an inner product. We may refer to the RAD (2.38) with V_{m+1} satisfying $\langle V_{m+1}, V_{m+1}\rangle = I_{m+1}$ as an $\langle\cdot,\cdot\rangle$-*orthonormal*, while V_{m+1} is said to be $\langle\cdot,\cdot\rangle$-*orthonormal*.

The rational implicit Q theorem holds for (quasi-)RADs in this more general form, and the proof is analogous to that of Theorem 2.16. Apart from working with the pencil (A, B) instead of just A, we additionally have to account for the inner product $\langle\cdot,\cdot\rangle$. Specifically, we have to take the inner product $\langle\cdot, \boldsymbol{v}_1\rangle$ in (2.21) to obtain $h_{11} = \langle \ell_{11}^{(\xi_1)} A^{(\xi_1)} \boldsymbol{v}_1, \boldsymbol{v}_1\rangle$ instead of (2.22), and analogously with (2.23)–(2.24).

Theorem 2.23. *Let (2.38) be an $\langle\cdot,\cdot\rangle$-orthonormal (quasi-)RAD. The $\langle\cdot,\cdot\rangle$-orthonormal matrix V_{m+1} and the pencil $(\underline{H_m}, \underline{K_m})$ are essentially uniquely determined by $V_{m+1}\boldsymbol{e}_1$ and the (block) ordering of the poles of (2.38).*

For simplicity, we shall mainly work with a single matrix A instead of a pencil (A, B). Propositions 2.21–2.22 and Theorem 2.23 can be used to transfer results from (quasi-)RADs with $B = I$ to those with $B \neq I$.

2.6 RKToolbox corner

We conclude the chapter with a few short MATLAB code examples showing how to use the RKToolbox for the algorithms discussed so far.

The rational Arnoldi algorithm (Algorithms 2.2 and 2.3) is implemented as `rat_krylov`,

```
1  [V, K, H] = rat_krylov(A, b, xi);
2
3  C.multiply= @(eta, rho, x) rho*A*x - eta*x;
4  C.solve   = @(mu,  nu,  x) (nu*A - mu*speye(size(A)))\x;
5  [V, K, H] = rat_krylov(C, b, xi);
6
7  [V, K, H] = rat_krylov(A, B, b, xi, 'real');
8
9  param.orth    = 'CGS';
10 param.reorth = 1;
11 param.inner_product = @(x, y) y'*(B*x);
12 [V, K, H] = rat_krylov(A, b, xi, param);
```

RKToolbox Example 2.1: Constructing RADs.

```
[V, K, H] = rat_krylov(A, b, xi1);
[V, K, H] = rat_krylov(A, V, K, H, xi2);
```

RKToolbox Example 2.2: Generating and extending an RAD.

and RKToolbox Example 2.1 shows four possible calls to `rat_krylov` in order to generate a (quasi-)RAD. Currently, there are 18 different ways to call `rat_krylov`, and typing `help rat_krylov` in MATLAB command line provides all the details. Here we provide a brief overview of some of them to give a flavour of the supported features and show the flexibility the function provides. Perhaps the most basic way of calling the function is the one given at line 1, where it is assumed that `A` is an N-by-N matrix, `b` an N-by-1 vector, and `xi` a 1-by-m row-vector of poles. The matrix (or pencil) can also be passed implicitly by providing a structure with fields `multiply` and `solve`, which are handles to functions implementing $(\eta, \rho, \boldsymbol{x}) \mapsto (\rho A - \eta I)\boldsymbol{x}$ and $(\mu, \nu, \boldsymbol{x}) \mapsto (\nu A - \mu I)^{-1}\boldsymbol{x}$, respectively. Thus, the call on line 5 is equivalent to the one on line 1. That on line 7 shows how to construct a quasi-RAD for the real-valued pencil (A, B) and real-valued starting vector $\boldsymbol{b}$. In this case the poles have to be either real-valued or appear in complex-conjugate pairs. Finally, on line 12 we show the usage of the `param` structure, which provides several options. We specify a non-standard inner product (in this case `param.inner_product` would be an inner product, if B is a symmetric positive definite matrix), and run the rational Arnoldi algorithm using the classical Gram–Schmidt algorithm with reorthogonalization.

RKToolbox Example 2.2 shows another feature of `rat_krylov`, namely, that an RAD can be extended. The two calls are equivalent to just calling `[V, K, H] =`

```
1 A = gallery('wathen', 10, 12); b = ones(length(A), 1);
2 [V, K, H] = rat_krylov(A, b, inf(1, 4));
3 H2 = H/K(1:end-1, :);
4
5 disp(norm(A*V(:, 1:end-1)-V*H2))
5    2.9684e-14
```

RKToolbox Example 2.3: Polynomial Arnoldi algorithm.

`rat_krylov(A, b, [xi1 xi2])` if the second set of poles `xi2` is known right from the start, which is not necessarily the case.

We now consider RKToolbox Example 2.3, which consists of two blocks; the first contains a fragment of MATLAB code, while the second contains the corresponding output. Through the thesis the computations were performed using the double-precision data type according to the IEEE standard. The lines in the output are labelled according to the related input line that produces the result. By setting all the m poles `xi` to infinity, as is done in RKToolbox Example 2.3, line 2, we compute an orthonormal basis `V` of a polynomial Krylov space using `rat_krylov`. In this case `K` is upper trapezoidal (because of the poles) and its upper m-by-m submatrix `K(1:end-1, :)` is nonsingular, by Lemma 2.6. Therefore, `H2=H/K(1:end-1, :)` is an unreduced upper Hessenberg matrix, and we obtain a decomposition as with the polynomial Arnoldi algorithm, cf. line 5. For this reason, the RKToolbox does not contain a dedicated function for Algorithm 1.1.

3 Rational Krylov subspace extraction

In this chapter we focus on extracting information out of a given $(m+1)$-dimensional rational Krylov space $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, related to a corresponding (quasi-)RAD

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m} \tag{3.1}$$

of order m. We start by considering the approximation of a few of the eigenvalues of A in Section 3.1. Applications include, e.g., structural engineering [48], the computation of specific eigenvalues, also referred to as dominant poles (of the transfer function), of the state space matrix of a dynamical system [84, 85], and the stability analysis of dynamical systems, where computing the eigenvalues with largest real part is of interest [78]. The general strategy [6, 24, 42, 73, 90, 102] is to obtain a surrogate matrix $A_\ell \in \mathbb{C}^{\ell,\ell}$ with $\ell \ll N$, say $\ell = m$ or $\ell = m+1$, such that $\Lambda(A_\ell)$, at least partly, provides a good approximation to some of the eigenvalues of A.

These developments form the basis for tackling the problem of approximating $f(A)\boldsymbol{b}$; the action of a function of a matrix $f(A)$ onto a vector $\boldsymbol{b}$, which we cover in Section 3.2. Computing or approximating $f(A)\boldsymbol{b}$ for large and sparse or structured matrices A arises, e.g., with $f(z) = \exp(z)$ and related functions within exponential integrators for solving differential equations [17, 40, 61, 62], while the function $f(z) = \sqrt{z}$ is of interest for stochastic differential equations [2]. The difficulty is that computing $f(A)$ for large A may be computationally too expensive or even unfeasible, due to memory requirements, since $f(A)$ typically does not preserve the (sparsity) structure of A. Approximating $f(A)\boldsymbol{b}$ directly is, however, possible [31, 55, 56, 58, 93].

The main goal of the chapter is to show how the rational Arnoldi algorithm can be used for the aforementioned tasks. We review some known strategies and propose new

ones, highlighting their potential benefit and deriving some of their basic properties. Our theoretical developments are complemented by small scale examples for which we can compute the exact solution for testing purposes. More advanced techniques, like *implicit filtering* for eigenvalue approximations, are discussed in Chapter 5. Further numerical examples, including large-scale problems from applications, are included in Chapters 4–6.

3.1 Approximate eigenpairs

We now review some of the most common approaches for approximating eigenpairs from a Krylov space available in the literature (see, e.g., [23, 24, 59, 73, 89, 90, 108, 109]), and provide new insights. These references consider RADs only (typically just a particular subset of RADs), but the results hold for, or can be extended to, general RADs as well as quasi-RADs, and we consider only the most general case. Four different extraction strategies are discussed: *explicit projection*, the *standard Ritz* approach, the *harmonic Ritz* approach and its generalisations, and finally, an extraction procedure based on *roots of orthogonal rational functions*. The section is concluded with a few (numerical) examples.

3.1.1. Explicit projection. We first address the situation when we can obtain *exact* eigenpairs of A from (3.1). The following result asserts that if there is one eigenvector of A in $\mathcal{R}(V_{m+1})$, then the whole space $\mathcal{R}(V_{m+1})$ is A-invariant. This is a consequence of the special structure of the space. The reverse, clearly, holds as well.

Proposition 3.1. *Let* (3.1) *be an orthonormal* (*quasi-*)*RAD. The space* $\mathcal{R}(V_{m+1})$ *is* A*-invariant if and only if there exists an eigenpair of* A *of the form* $(\vartheta, V_{m+1}\boldsymbol{y})$.

Proof. Let us assume that $AV_{m+1}\boldsymbol{y} = \vartheta V_{m+1}\boldsymbol{y}$, with $\boldsymbol{y} \neq \boldsymbol{0}$. We can extend (3.1) into

$$AV_{m+1}\begin{bmatrix}\underline{K_m} & \boldsymbol{y}\end{bmatrix} = V_{m+1}\begin{bmatrix}\underline{H_m} & \vartheta\boldsymbol{y}\end{bmatrix}. \tag{3.2}$$

Under the assumption that $[\underline{K_m} \quad \boldsymbol{y}]$ is nonsingular, equation (3.2) implies $AV_{m+1} = V_{m+1}[\underline{H_m} \quad \vartheta\boldsymbol{y}][\underline{K_m} \quad \boldsymbol{y}]^{-1}$, which shows that $\mathcal{R}(V_{m+1})$ is A-invariant.

It follows from Corollary 2.18 that $\underline{K_m}$ is of full column rank m. Therefore, it suffices to show that $\boldsymbol{y} \notin \mathcal{R}(\underline{K_m})$. Let us assume, to the contrary, that $\boldsymbol{y} = \underline{K_m}\boldsymbol{z}$, for

some nonzero vector $\boldsymbol{z} \in \mathbb{C}^m$. As $(\vartheta, V_{m+1}\boldsymbol{y})$ is an eigenpair of A, using (3.1) we obtain

$$AV_{m+1}\underline{K_m}\boldsymbol{z} = \vartheta V_{m+1}\underline{K_m}\boldsymbol{z} = V_{m+1}\underline{H_m}\boldsymbol{z},$$

and hence $\vartheta\underline{K_m}\boldsymbol{z} = \underline{H_m}\boldsymbol{z}$. This implies that $\underline{H_m} - \vartheta\underline{K_m}$ is not of full column rank, which is in contradiction with Corollary 2.18. □

Note that $\mathcal{R}(V_{m+1})$ is A-invariant if and only if $m = d(A, \boldsymbol{b}) - 1$, and $d(A, \boldsymbol{b})$ may be as large as N. It may thus be impractical to construct RADs of order $d(A, \boldsymbol{b}) - 1$, and, indeed, that is typically not done in practice. However, we might obtain *good* approximations to some of the eigenpairs of A from (3.1), even if $\mathcal{R}(V_{m+1})$ is not A-invariant. A standard approach is to project A onto $\mathcal{R}(V_{m+1})$, that is, to form

$$A_{m+1} := V_{m+1}^{\dagger} A V_{m+1} \in \mathbb{C}^{m+1,m+1}, \tag{3.3}$$

and then consider the approximate eigenpairs

$$(\vartheta, \boldsymbol{x} = V_{m+1}\boldsymbol{y}), \quad \text{where} \quad A_{m+1}\boldsymbol{y} = \vartheta\boldsymbol{y}. \tag{3.4}$$

In this regard the problem is reduced to the (typically) lower dimensional problem of finding the eigenpairs of A_{m+1}, which can be done with a direct method. We remark that $V_{m+1}^{\dagger} = V_{m+1}^{*}$ if (3.1) is orthonormal. The *relative residual norm*

$$\frac{\|A\boldsymbol{x} - \vartheta\boldsymbol{x}\|_2}{\|A\|_2\|\boldsymbol{x}\|_2 + |\vartheta|\|\boldsymbol{x}\|_2} \tag{3.5}$$

can be used to assess the quality of the (approximate) eigenpair $(\vartheta, \boldsymbol{x})$. We report, for the interested reader, that the convergence of $\Lambda(A_{m+1})$ towards eigenvalues of A is studied for Hermitian matrices A in [6], within a potential-theoretic setup. We do not dwell upon these considerations here.

An undesirable property of the approach just described is that both computing the reduced matrix A_{m+1}, and forming the residuals (3.5), involve computations with vectors of size N. This problem can be overcome by considering approximate eigenpairs from an m-dimensional subspace of $\mathcal{R}(V_{m+1})$. The remaining dimension can be used to efficiently estimate the quality of the approximation.

3.1.2. Standard Ritz pairs. We say that a pair $(\vartheta, \boldsymbol{x} \neq \boldsymbol{0}) \in \mathbb{C} \times \mathcal{V}$, where $\mathcal{V}$ is a subspace of $\mathbb{C}^N$, is a (*standard*) *Ritz pair for A with respect to $\mathcal{V}$* if the condition

$A\boldsymbol{x} - \vartheta\boldsymbol{x} \perp \mathcal{V}$ is satisfied; see, e.g., [42, 102]. If, for instance, $(\vartheta, \boldsymbol{x})$ is an eigenpair of A, then it clearly is a Ritz pair, since $A\boldsymbol{x} - \vartheta\boldsymbol{x} = \boldsymbol{0} \perp \mathcal{V}$. We can consider Ritz pairs with respect to any space $\mathcal{V}$. For instance, the approximate eigenpairs related to (3.3)–(3.4) are standard Ritz pairs for A with respect to $\mathcal{R}(V_{m+1})$. We now specialise this general definition to (quasi-)RADs.

Definition 3.2. *Let* (3.1) *be a (quasi-)RAD. An approximate eigenpair* $(\vartheta, \boldsymbol{x} = V_{m+1}\underline{K_m}\boldsymbol{y} \neq \boldsymbol{0})$ *for* A *is called a* (standard) Ritz pair for A with respect to $\mathcal{R}(V_{m+1}\underline{K_m})$ *if the condition* $A\boldsymbol{x} - \vartheta\boldsymbol{x} \perp \mathcal{R}(V_{m+1}\underline{K_m})$ *is satisfied. We call* ϑ *a* Ritz value, *and* $\boldsymbol{x}$ *a corresponding* Ritz vector.

The choice $\mathcal{R}(V_{m+1}\underline{K_m})$ is natural when working with (3.1), as it allows to construct efficiently (without need of additional explicit projection) a reduced eigenproblem of order m, as we show in Proposition 3.4 below. It is interesting that $\mathcal{R}(V_{m+1}\underline{K_m})$ equals the space $\mathcal{K}_m(A, q_m(A)^{-1}\boldsymbol{b})$. Indeed, by Corollary 2.18, $\underline{K_m}$ is of full column rank m, and, thus, so is $V_{m+1}\underline{K_m}$. Consequently, $\mathcal{R}(V_{m+1}\underline{K_m})$ is an m-dimensional subspace of $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, as is $A\mathcal{R}(V_{m+1}\underline{K_m})$, by (3.1) and again Corollary 2.18. It follows, by (3.1), that $\mathcal{R}(V_{m+1}\underline{K_m})$ is the space of all vectors $r_{m-1,m}(A)\boldsymbol{b}$, where $r_{m-1,m} = p_{m-1}/q_m \in \mathcal{P}_{m-1}/q_m$ is a rational function of type at most $(m-1, m)$, with fixed denominator q_m. A more general statement follows.

Proposition 3.3. *Let* (3.1) *be a (quasi-)RAD for* $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, *and let* $\alpha, \beta \in \mathbb{C}$ *be such that* $|\alpha| + |\beta| \neq 0$. *Then*

$$\mathcal{R}\left(\alpha V_{m+1}\underline{H_m} - \beta V_{m+1}\underline{K_m}\right) = \left\{ r(A)\boldsymbol{b} \middle| r(z) = \frac{(\alpha z - \beta)p_{m-1}(z)}{q_m(z)}, p_{m-1} \in \mathcal{P}_{m-1} \right\},$$

is the m*-dimensional subspace of* $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ *containing all vectors* $r(A)\boldsymbol{b}$ *with rational functions* r *of type at most* (m, m), *having a fixed denominator* q_m, *and a fixed* (*formal*) *root* β/α.

Proof. Follows from Corollary 2.18 and (2.7) with $\widehat{\alpha} = \alpha$, and $\widehat{\beta} = \beta$. ∎

For the case $\alpha = 0$ in Proposition 3.3 we get $\mathcal{R}(V_{m+1}\underline{K_m})$, and the rational function r of type at most (m, m) is said to have a *formal* root fixed at *infinity*, i.e., it is of type at most $(m-1, m)$. For the space $\mathcal{R}(V_{m+1}\underline{H_m})$, for example, the fixed root is at zero. Part (i) of the following result can be found in [23, Lemma 2.4], or [73, Theorem 2.1]. Analogous results to (ii)–(iv) are derived for the explicit projection (3.3) in [55, Lemma 4.5]. (The choice of the notation χ_m^∞ that we employ becomes clear in the following subsection.)

Proposition 3.4. *Let (3.1) be an orthonormal (quasi-)RAD, and let χ_m^∞ be the characteristic polynomial of $\underline{K_m^\dagger}\underline{H_m}$. Then the following holds.*

(i) *The pair $(\vartheta, V_{m+1}\underline{K_m}\boldsymbol{y})$ is a Ritz pair of A with respect to $\mathcal{R}(V_{m+1}\underline{K_m})$ if and only if $(\vartheta, \boldsymbol{y})$ is an eigenpair of the m-by-m matrix $\underline{K_m^\dagger}\underline{H_m}$.*

(ii) *The matrix $\underline{K_m^\dagger}\underline{H_m}$ is nonderogatory.*

(iii) $\chi_m^\infty(A)q_m(A)^{-1}\boldsymbol{b} \perp \mathcal{K}_m(A, q_m(A)^{-1}\boldsymbol{b}) = \mathcal{R}(V_{m+1}\underline{K_m})$.

(iv) *The characteristic polynomial χ_m^∞ minimizes $\|p_m(A)q_m(A)^{-1}\boldsymbol{b}\|_2$ over all polynomials $p_m \in \mathcal{P}_m$ of the form $p_m(z) = z^m + p_{m-1}(z)$, with $p_{m-1} \in \mathcal{P}_{m-1}$.*

Proof. Let us consider (i). The pair $(\vartheta, V_{m+1}\underline{K_m}\boldsymbol{y})$ is a Ritz pair for A with respect to $\mathcal{R}(V_{m+1}\underline{K_m})$ if

$$\underline{K_m}^* V_{m+1}^* A V_{m+1}\underline{K_m}\boldsymbol{y} - \vartheta \underline{K_m}^* V_{m+1}^* V_{m+1}\underline{K_m}\boldsymbol{y} = \boldsymbol{0}.$$

Using (3.1) and $V_{m+1}^* V_{m+1} = I_{m+1}$, we arrive at $\underline{K_m}^*\underline{H_m}\boldsymbol{y} - \vartheta\underline{K_m}^*\underline{K_m}\boldsymbol{y} = \boldsymbol{0}$. Finally, as $\underline{K_m}$ is of full rank by Corollary 2.18, the matrix $\underline{K_m}^*\underline{K_m}$ is nonsingular. This gives $\left(\underline{K_m}^*\underline{K_m}\right)^{-1}\underline{K_m}^*\underline{H_m}\boldsymbol{y} = \vartheta\boldsymbol{y}$, or, equivalently, $\underline{K_m^\dagger}\underline{H_m}\boldsymbol{y} = \vartheta\boldsymbol{y}$.

To show statement (ii), introduce $\boldsymbol{q} := q_m(A)^{-1}\boldsymbol{b}$, and $W_m := \begin{bmatrix}\boldsymbol{q} & A\boldsymbol{q} & \dots & A^{m-1}\boldsymbol{q}\end{bmatrix}$. Note that the *companion matrix* (see, e.g., [42, Section 7.4.6])

$$W_m^\dagger A W_m = \begin{bmatrix} & & & & \alpha_0 \\ 1 & & & & \alpha_1 \\ & 1 & & & \alpha_2 \\ & & \ddots & & \vdots \\ & & & 1 & \alpha_{m-2} \\ & & & 1 & \alpha_{m-1} \end{bmatrix}, \tag{3.6}$$

for some $\{\alpha_{j-1}\}_{j=1}^m \subset \mathbb{C}$, is an unreduced upper Hessenberg matrix, and, therefore, nonderogatory (see, e.g., [60, Problem 13.3]). It follows from Proposition 3.3 that the matrix $\underline{K_m^\dagger}\underline{H_m}$ is similar to $W_m^\dagger A W_m$, since $\mathcal{R}(V_{m+1}\underline{K_m}) = \mathcal{R}(W_m)$, which concludes the proof.

We now consider the statement (iii). It follows from (3.6) that $\chi_m^\infty(z) = z^m - \sum_{j=0}^{m-1}\alpha_j z^j$. Therefore,

$$\begin{aligned} W_m W_m^\dagger \chi_m^\infty(A)\boldsymbol{q} &= W_m W_m^\dagger A^m \boldsymbol{q} - \sum_{j=0}^{m-1} W_m W_m^\dagger \alpha_j A^j \boldsymbol{q} \\ &= W_m W_m^\dagger A^m \boldsymbol{q} - \sum_{j=0}^{m-1} \alpha_j A^j \boldsymbol{q} = \boldsymbol{0}. \end{aligned}$$

Finally, let us consider (iv). Let $\widehat{\chi}_{m-1}^{\infty}(z) := \chi_m^{\infty}(z) - z^m \in \mathcal{P}_{m-1}$, and $p_m(z) = z^m + p_{m-1}(z)$. Note that

$$\begin{aligned}\|p_m(A)q_m(A)^{-1}\boldsymbol{b}\|_2^2 &= \|\chi_m^{\infty}(A)\boldsymbol{q} - \chi_m^{\infty}(A)\boldsymbol{q} + p_m(A)\boldsymbol{q}\|_2^2 \\ &= \|\chi_m^{\infty}(A)\boldsymbol{q} - \widehat{\chi}_{m-1}^{\infty}(A)\boldsymbol{q} + p_{m-1}(A)\boldsymbol{q}\|_2^2 \\ &= \|\chi_m^{\infty}(A)\boldsymbol{q}\|_2^2 + \|p_{m-1}(A)\boldsymbol{q} - \widehat{\chi}_{m-1}^{\infty}(A)\boldsymbol{q}\|_2^2,\end{aligned}$$

where the last equality follows from (iii). Clearly, $\|p_m(A)q_m(A)^{-1}\boldsymbol{b}\|_2^2$ is minimised by taking $p_{m-1} = \widehat{\chi}_{m-1}^{\infty}$. ∎

We note that $\underline{K_m}^{\dagger}\underline{H_m}$ can be constructed without explicitly forming $\underline{K_m}^{\dagger}$, by considering m least squares problems $\underline{K_m}\boldsymbol{x}_j = \underline{H_m}\boldsymbol{e}_j$ for $\boldsymbol{x}_j$, $j = 1, 2, \ldots, m$. The rational implicit Q theorems (cf. Theorem 2.16 and Theorem 2.20) state that orthonormal RADs and quasi-RADs are essentially uniquely determined by the starting vector and the ordering of the poles. The following remark shows that this implies they have the same standard Ritz pairs.

Remark 3.5. Essentially equal (quasi-)RADs have the same set of Ritz pairs. Indeed, let $A\widehat{V}_{m+1}\underline{\widehat{K}_m} = \widehat{V}_{m+1}\underline{\widehat{H}_m}$, with $\widehat{V}_{m+1} = V_{m+1}D$, $\underline{\widehat{K}_m} = D^{-1}\underline{K_m}T$, and $\underline{\widehat{H}_m} = D^{-1}\underline{H_m}T$, be essentially equal to (3.1). Then, $\underline{\widehat{K}_m}^{\dagger}\underline{\widehat{H}_m} = T^{-1}\underline{K_m}^{\dagger}\underline{H_m}T$ is similar to $\underline{K_m}^{\dagger}\underline{H_m}$, and if $(\vartheta, \boldsymbol{x})$ is an eigenpair of $\underline{K_m}^{\dagger}\underline{H_m}$, then $(\vartheta, T^{-1}\boldsymbol{x})$ is an eigenpair of $\underline{\widehat{K}_m}^{\dagger}\underline{\widehat{H}_m}$. The corresponding Ritz pairs $(\vartheta, \widehat{V}_{m+1}\underline{\widehat{K}_m}T^{-1}\boldsymbol{x}) = (\vartheta, V_{m+1}DD^{-1}\underline{K_m}TT^{-1}\boldsymbol{x}) = (\vartheta, V_{m+1}\underline{K_m}\boldsymbol{x})$ coincide.

Rational implicit Q theorems thus guarantee that, for instance, independent on how we choose the admissible continuation pairs in Algorithm 2.2, we obtain the same eigenvalue approximations. Also, we can transform the (quasi-)RAD obtained by Algorithm 2.2 into an essentially equal (quasi-)RAD which might have some advantageous properties; see, e.g., Remark 3.6 below, which appears to be new.

Remark 3.6 (orthonormal $\underline{K_m}$). If A is Hermitian (symmetric), it is desired for $\underline{K_m}^{\dagger}\underline{H_m}$ to be Hermitian (symmetric) as well. Let $\underline{K_m} =: \underline{Q_m}R_m$ be a thin QR factorisation of $\underline{K_m}$. Then (3.1) can be replaced with $AV_{m+1}\underline{Q_m}R_m = V_{m+1}\underline{H_m}$, or, equivalently,

$$AV_{m+1}\underline{Q_m} = V_{m+1}\underline{H_m}R_m^{-1}. \tag{3.7}$$

In this case $\underline{K_m}^{\dagger}\underline{H_m}$ is replaced by $\underline{Q_m}^{*}\underline{H_m}R_m^{-1}$, which is Hermitian (symmetric) if A is. Indeed, multiplying (3.7) with $(V_{m+1}\underline{Q_m})^{*}$ from the left we have $\underline{Q_m}^{*}\underline{H_m}R_m^{-1} =$

$\underline{Q_m}^* V_{m+1}^* A V_{m+1} \underline{Q_m}$, and the observation follows. Hence, if needed, we may assume, without loss of generality, that $\underline{K_m}$ in the orthonormal (quasi-)RAD (3.1) is orthonormal, as otherwise (3.1) may be replaced by the equivalent (3.7).

Let $(\vartheta, V_{m+1}\underline{K_m}\boldsymbol{y})$ be a Ritz pair for A. From (3.1) we have $AV_{m+1}\underline{K_m}\boldsymbol{y} - \vartheta V_{m+1}\underline{K_m}\boldsymbol{y} = V_{m+1}\left(\underline{H_m} - \vartheta\underline{K_m}\right)\boldsymbol{y}$. Therefore, the norm

$$\|\underline{H_m}\boldsymbol{y} - \vartheta\underline{K_m}\boldsymbol{y}\|_2 \tag{3.8}$$

provides a cheap estimate of the accuracy for the Ritz pair. If this norm is small compared to $\|A\|_2$, the Ritz pair $(\vartheta, V_{m+1}\underline{K_m}\boldsymbol{y})$ is an eigenpair of a nearby matrix [102].

3.1.3. Harmonic Ritz pairs. Let us now discuss an alternative extraction process, considered also in [23, 90] for the rational Arnoldi algorithm.

Definition 3.7. *Let (3.1) be a (quasi-)RAD. An approximate eigenpair $(\vartheta, \boldsymbol{x} = V_{m+1}\underline{K_m}\boldsymbol{y} \neq \boldsymbol{0})$ for A is called a* harmonic Ritz pair for A with respect to $\mathcal{R}(V_{m+1}\underline{K_m})$ *if the condition $A\boldsymbol{x} - \vartheta\boldsymbol{x} \perp \mathcal{R}(AV_{m+1}\underline{K_m}) = \mathcal{R}(V_{m+1}\underline{H_m})$ is satisfied. We call ϑ a* harmonic Ritz value, *and $\boldsymbol{x}$ a corresponding* harmonic Ritz vector.

Same as with standard Ritz pairs, an eigenpair is a harmonic Ritz pair. Part (i) of the following result can be found in [23, Lemma 2.4].

Proposition 3.8. *Let (3.1) be an orthonormal (quasi-)RAD, with A and $\underline{H_m}^\dagger\underline{K_m}$ being nonsingular, and let χ_m^0 be the characteristic polynomial of $\underline{H_m}^\dagger\underline{K_m}$. Then the following holds.*

(i) *The pair $(\vartheta^{-1}, V_{m+1}\underline{K_m}\boldsymbol{y})$ is a harmonic Ritz pairs for A with respect to $\mathcal{R}(V_{m+1}\underline{K_m})$ if and only if $(\vartheta, \boldsymbol{y})$ is an eigenpair of the matrix $\underline{H_m}^\dagger\underline{K_m}$.*

(ii) *The matrix $\underline{H_m}^\dagger\underline{K_m}$ is nonderogatory.*

(iii) $\chi_m^0(A)q_m(A)^{-1}\boldsymbol{b} \perp \mathcal{K}_m(A, Aq_m(A)^{-1}\boldsymbol{b}) = \mathcal{R}(V_{m+1}\underline{H_m})$.

(iv) *The rescaled characteristic polynomial $\gamma\chi_m^0$, where $\gamma \in \mathbb{C}$ is such that for some $\widehat{\chi}_{m-1}^0 \in \mathcal{P}_{m-1}$ we have $\gamma\chi_m^0(z) = z\widehat{\chi}_{m-1}^0(z) - 1$, minimizes $\|p_m(A)q_m(A)^{-1}\boldsymbol{b}\|_2$ over all polynomials $p_m \in \mathcal{P}_m$ of the form $p_m(z) = zp_{m-1}(z) - 1$, with $p_{m-1} \in \mathcal{P}_{m-1}$.*

Proof. The derivation for $\underline{H_m}^\dagger\underline{K_m}$ is analogous to that in Proposition 3.4. To show that $\underline{H_m}^\dagger\underline{K_m}$ is nonderogatory, we can consider the RKD $A^{-1}V_{m+1}\underline{H_m} = V_{m+1}\underline{K_m}$, as we considered (3.1) in Proposition 3.4; see also the discussion after Proposition 2.21. The remaining two properties can be established analogously to those in Proposition 3.4.

We only remark that χ_m^0 can be rescaled to obtain the form $\gamma\chi_m^0(z) = z\widehat{\chi}_{m-1}^0(z) - 1$, as $\underline{H_m^\dagger K_m}$ is nonsingular and hence zero is not a root of χ_m^0. □

Analogous remarks to those for $\underline{K_m^\dagger H_m}$ hold for $\underline{H_m^\dagger K_m}$; the matrix $\underline{H_m^\dagger K_m}$ can be constructed without explicitly forming $\underline{H_m^\dagger}$. Essentially equal (quasi-)RADs have the same set of harmonic Ritz pairs, and it might be beneficial to consider orthonormal $\underline{H_m}$, in which case $\underline{K_m}$ cannot, in general, be orthonormal as well.

Remark 3.9 (orthonormal $\underline{H_m}$). If needed, we may assume, without loss of generality, that $\underline{H_m}$ in the orthonormal (quasi-)RAD (3.1), is orthonormal, as otherwise (3.1) may be replaced by the equivalent $AV_{m+1}\underline{K_m}R_m^{-1} = V_{m+1}\underline{Q_m}$, where $\underline{H_m} =: \underline{Q_m}R_m$ is a thin QR factorisation of $\underline{H_m}$. If $\underline{H_m}$ is orthonormal and A is Hermitian then $\underline{H_m^\dagger K_m}$ is Hermitian as well.

We state for reference that the residual analogous to (3.8) for the harmonic Ritz pair $(\vartheta^{-1}, V_{m+1}\underline{K_m}\boldsymbol{y})$ is

$$\|\underline{H_m}\boldsymbol{y} - \vartheta^{-1}\underline{K_m}\boldsymbol{y}\|_2. \tag{3.9}$$

Harmonic Ritz pairs are sometimes referred to as harmonic Ritz pairs with *target* $\tau = 0$, and any other finite $\tau \in \mathbb{C} \setminus \Lambda(A)$ may be considered; see [102]. We shall use the notion τ-harmonic instead.

Definition 3.10. *Let* (3.1) *be a (quasi-)RAD, and let* $\tau \in \mathbb{C} \setminus \Lambda(A)$ *be a scalar. An approximate eigenpair* $(\vartheta+\tau, \boldsymbol{x} = V_{m+1}\underline{K_m}\boldsymbol{y} \neq \boldsymbol{0})$ *for* A *is called a* τ-harmonic Ritz pair for A with respect to $\mathcal{R}(V_{m+1}\underline{K_m})$ *if the condition* $A\boldsymbol{x} - (\vartheta+\tau)\boldsymbol{x} \perp \mathcal{R}([A-\tau I]V_{m+1}\underline{K_m})$ *is satisfied. We call* ϑ *a* τ-harmonic Ritz value, *and* $\boldsymbol{x}$ *a corresponding* τ-harmonic Ritz vector.

Equivalently, τ-harmonic Ritz pairs for A can be defined through harmonic Ritz pairs for $A - \tau I$, as the following lemma shows.

Lemma 3.11. *Let* $A \in \mathbb{C}^{N,N}$, $\tau \in \mathbb{C} \setminus \Lambda(A)$, *and let* $\mathcal{V}$ *be a subspace of* $\mathbb{C}^N$. *The pair* $(\vartheta + \tau, \boldsymbol{x}) \in \mathbb{C} \times \mathcal{V}$ *is a* τ-harmonic Ritz pair for A with respect to $\mathcal{V}$ *if and only if* $(\vartheta, \boldsymbol{x})$ *is a harmonic Ritz pair for* $A - \tau I$ *with respect to* $\mathcal{V}$.

Proof. The statement follows from the definition of τ-harmonic Ritz pairs. □

If (3.1) is an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, \{\xi_j\}_{j=1}^m)$, by Proposition 2.21,

$$(A - \tau I)V_{m+1}\underline{K_m} = V_{m+1}(\underline{H_m} - \tau\underline{K_m}) \tag{3.10}$$

is an RAD for $\mathcal{Q}_{m+1}(A-\tau I, \boldsymbol{b}, \{\xi_j-\tau\}_{j=1}^m)$. If $\tau \in \mathbb{R}$, by Proposition 2.22, this also holds for quasi-RADs. In these cases, Proposition 3.8 is applicable to (3.10), and by Lemma 3.11 it indicates how to obtain τ-harmonic Ritz pairs for A. We adopt the notation χ_m^τ for the characteristic polynomial of $(\underline{H_m}-\tau\underline{K_m})^\dagger \underline{K_m}$, whose roots $\{\vartheta_j\}_{j=1}^m$ provide τ-harmonic Ritz values $\{\vartheta_j^{-1}+\tau\}_{j=1}^m$, provided that all $\vartheta_j \neq 0$.

We can also use the representation $\tau = \beta/\alpha$, in which case $\underline{H_m}-\tau\underline{K_m}$ is replaced by $\alpha\underline{H_m}-\beta\underline{K_m}$. From here we see, by continuity, that standard Ritz values may be considered as ∞-harmonic Ritz values.

3.1.4. Roots of orthogonal rational functions. Instead of standard or harmonic Ritz pairs, in, e.g., [24, 59, 89, 108, 109], approximate eigenpairs related to roots of orthogonal rational functions are used (and are nonetheless often referred to as Ritz pairs). Let us clarify what is meant by roots of orthogonal rational functions. By Theorem 2.12 we know that the basis vectors $\boldsymbol{v}_{j+1}$ can be expressed as $\boldsymbol{v}_{j+1} = r_j(A)\boldsymbol{v}_1$, where $r_j = p_j/q_j$ is a rational function. We refer to the (formal) roots of $p_j \in \mathcal{P}_j$ as roots of orthogonal rational functions; *orthogonal* because typically the corresponding RAD (3.1) is orthonormal. Since p_j is of degree at most j it has at most j roots. If $\deg(p_j) < j$, then we say that p_j has $j-\deg(p_j)$ *formal* roots at infinity. Therefore, we can say that p_j has (formally) j roots. We may refer to these roots as the *roots of* $\boldsymbol{v}_{j+1}$, for $j = 1, 2, \ldots, m$. Hence, by Theorem 2.12, the generalized eigenpairs of the m-by-m pencil (H_m, K_m) are used to construct approximate eigenpairs

$$(\vartheta, V_{m+1}\underline{K_m}\boldsymbol{y}), \quad \text{where} \quad H_m\boldsymbol{y} = \vartheta K_m\boldsymbol{y}. \tag{3.11}$$

The corresponding residual for an orthonormal (quasi-)RAD is derived from

$$\begin{aligned} AV_{m+1}\underline{K_m}\boldsymbol{y} - \vartheta\underline{K_m}\boldsymbol{y} &= V_{m+1}(\underline{H_m}\boldsymbol{y} - \vartheta\underline{K_m}\boldsymbol{y}) \\ &= [h_{m+1,m-1}(\boldsymbol{e}_{m-1}^T\boldsymbol{y}) + (h_{m+1,m} - \vartheta k_{m+1,m})(\boldsymbol{e}_m^T\boldsymbol{y})]\boldsymbol{v}_{m+1}, \end{aligned}$$

with the right-hand side of the last equation being nonzero unless V_{m+1} is A-invariant; cf. Proposition 3.1. Therefore, for RADs, where $h_{m+1,m-1} = 0$, the quantity

$$|(h_{m+1,m} - \vartheta k_{m+1,m})(\boldsymbol{e}_m^T\boldsymbol{y})|, \tag{3.12}$$

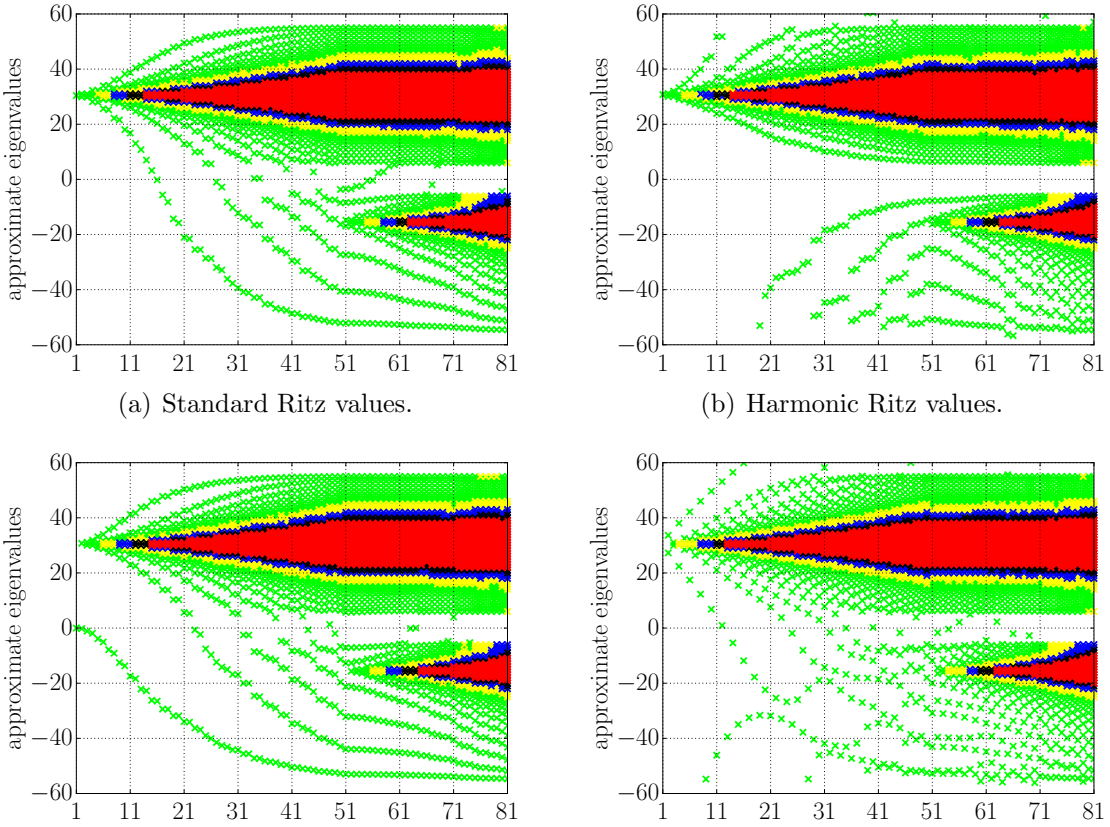
provides a cheap estimate of the quality of the eigenpair. For quasi-RADs (3.12) can be replaced by

$$|h_{m+1,m-1}(\boldsymbol{e}_{m-1}^T\boldsymbol{y}) + (h_{m+1,m} - \vartheta k_{m+1,m})(\boldsymbol{e}_m^T\boldsymbol{y})|. \tag{3.13}$$

The choice (3.11) appears to be motivated by the polynomial Arnoldi algorithm, where $\underline{K_m} = \underline{I_m}$, and standard Ritz values with respect to $\mathcal{R}(V_{m+1}\underline{K_m}) = \mathcal{R}(V_m)$ are computed from H_m, the leading m-by-m submatrix of $\underline{H_m}$. Thus, for the polynomial Arnoldi algorithm, standard Ritz values coincide with roots of orthogonal polynomials, but in the rational case this is not necessarily true. Standard Ritz values of order m coincide with the m roots of the $(m+1)$st orthogonal basis function if the mth pole $\xi_m = \infty$ is at infinity (which is always true in the polynomial case) and, hence, produces $\underline{K_m}$ with last row being $\mathbf{0}^T$. In the rational case in general, however, standard and harmonic Ritz values, roots of orthogonal rational functions, and explicit projection all provide different sets of approximants.

Harmonic Ritz values appear to be better suited for approximating interior eigenvalues than standard Ritz pairs, although this is still not yet fully understood. Interesting discussions can be found in [90] for rational Krylov spaces, and in [45, 82] for polynomial Krylov spaces. An insightful discussion is contained in [102, pp. 292–294] for general spaces, and further extensions to *rational harmonic Ritz values* are investigated in [63]. Standard and harmonic Ritz values provide, in a precise sense, optimal bounds on the eigenvalues of a Hermitian positive definite matrix; see, e.g., [5]. In the polynomial case this transfers over to roots of orthogonal polynomials, as they coincide with standard Ritz values, but in the general rational case this is not guaranteed, as the following examples show.

3.1.5. Numerical example. We consider $A = \mathrm{diag}(5, 6, \ldots, 55, -5, -6, \ldots, -55)$, which is clearly symmetric. For $\boldsymbol{b}$ we use the vector with all components equal to 1. The two distinct poles $\xi_1 = \xi_2 = \ldots = \xi_{50} = 30.5$ and $\xi_{51} = \xi_{52} = \ldots = \xi_{81} = -15.5$ are employed multiple times. The purpose of this example is to highlight some of the differences of various extraction procedures. In Figure 3.1 we plot the approximate eigenvalues per iteration with the four distinct extraction strategies discussed in Section 3.1. It follows from the *interlacing property* (see, e.g., [42, Theorem 8.1.7]) that standard Ritz values are guaranteed to be contained in the spectral interval of A. On the other hand, harmonic Ritz values approximate the eigenvalues from the outside (and some are outside the plotted range), but any interval containing zero and free of eigenvalues of A is guaranteed not to contain harmonic Ritz values [82]. Interestingly,



(a) Standard Ritz values.

(b) Harmonic Ritz values.

(c) Standard Ritz values by explicit projection.

(d) Roots of orthogonal rational functions.

Figure 3.1: We visualise approximate eigenvalues for the example from Section 3.1.5, extracted from an RAD as the iteration progresses, according to the extraction strategies discussed in Section 3.1. At iteration j there are j approximate eigenvalues and they are represented with a small $\times$ symbol. Different colours are used to indicate the quality of the corresponding eigenpair. Green is used for eigenpairs with relative residual (3.5) above 10^{-3}; yellow if (3.5) is between 10^{-6} and 10^{-3}; blue for the range between 10^{-9} and 10^{-6}; black for the range between 10^{-12} and 10^{-9}; and red for eigenpairs with relative residual below 10^{-12}. As expected, the eigenvalues closest to the two poles, 30.5 and -15.5, used repeatedly, converged first.

unlike the polynomial case, in the rational case some of the roots of orthogonal rational functions are also outside the spectral interval of A. For instance, for $j = 1$, the vector $\boldsymbol{v}_2$ is collinear with

$$\widehat{\boldsymbol{v}}_2 := (A-\xi_1 I)^{-1}\boldsymbol{v}_1 - \gamma \boldsymbol{v}_1, \quad \text{where} \quad \boldsymbol{v}_1 = \boldsymbol{b}/10, \quad \text{and} \quad \gamma := \boldsymbol{v}_1^*(A-\xi_1 I)^{-1}\boldsymbol{v}_1. \tag{3.14}$$

If $\gamma \neq 0$, then $\widehat{\boldsymbol{v}}_2 = -\gamma(A-\xi_1 I)^{-1}[(A-\xi_1 I)\boldsymbol{v}_1 - \gamma^{-1}\boldsymbol{v}_1]$, and the corresponding root of $\boldsymbol{v}_2$ is $\xi_1 + \gamma^{-1}$. If $\gamma = 0$, then $\widehat{\boldsymbol{v}}_2 = (A-\xi_1 I)^{-1}\boldsymbol{v}_1$ and the corresponding *formal* root is infinite. In our case $\gamma = \sum_{n=6}^{55} \frac{0.01}{n-\xi_1} + \frac{0.01}{-n-\xi_1}$, yielding the root $\xi_1 + \gamma^{-1} \approx -84.3$, which is not included in Figure 3.1 as it does not fit in the range. In the next example we show that one can indeed obtain $\gamma = 0$. We consider a symmetric positive definite

matrix A, and a pole within the spectral interval of A which, is a reasonable choice when approximating eigenvalues.

Example 3.12. Let $A = \mathrm{diag}(1, 2, 4, 5)$, $\xi_1 = 3$, and $\boldsymbol{b} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$. Then $(A - \xi_1 I)^{-1} = \mathrm{diag}(-2^{-1}, -1, 1, 2^{-1})$ and, therefore, the scalar γ from (3.14) equals zero. This means that the vector $\widehat{\boldsymbol{v}}_2 = (A - \xi_1 I)^{-1}\boldsymbol{v}_1$ is already orthogonal to $\boldsymbol{b}$ and the (formal) root of $\boldsymbol{v}_2$ is infinity. Clearly,

$$\boldsymbol{c}_2 := \begin{bmatrix} \gamma \\ \|\widehat{\boldsymbol{v}}_2\|_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\sqrt{5}}{2\sqrt{2}} \end{bmatrix}.$$

Using $\nu_j = 1, \mu_j = \xi_1, \rho_j = 0, \eta_j = -1$ and $\underline{\boldsymbol{t}_j} = \boldsymbol{e}_1 \in \mathbb{R}^2$ with $j = 1$ in (2.5) we obtain

$$\underline{K_1} = \begin{bmatrix} 0 \\ \frac{\sqrt{5}}{2\sqrt{2}} \end{bmatrix}, \quad \text{and} \quad \underline{H_1} = \begin{bmatrix} 1 \\ 3\frac{\sqrt{5}}{2\sqrt{2}} \end{bmatrix},$$

and indeed $\Lambda(H_1, K_1) = \{\infty\}$. Interestingly, since $\underline{K_1^\dagger} = \begin{bmatrix} 0 & \frac{2\sqrt{2}}{\sqrt{5}} \end{bmatrix}$, we have $\underline{K_1^\dagger}\underline{H_1} = 3$, thus, the standard Ritz value equals the pole ξ_1 that is used! For completeness, the harmonic Ritz value is approximately 3.533.

Example 3.12 is of theoretical interest and one should not be discouraged by it, but, rather, aware of it. In practice all extraction strategies can provide good approximations, in particular for larger m.

3.2 Functions of matrices times a vector

We now consider the problem of approximating $f(A)\boldsymbol{b}$. The approach related to the explicit projection of A onto the rational Krylov space $\mathcal{R}(V_{m+1})$ is briefly discussed in Section 3.2.1; more details can be found, e.g., in [7, 31, 55, 56, 58, 93]. As we explain, the efficient evaluation of such an approximation relies on the fact that the mth pole is infinity. This restriction may be disadvantageous when m is not know a priori, and the quality of the approximation needs to be assessed as the iteration progresses. For this purpose, we introduce new approximants, related to standard and harmonic Ritz values, in Section 3.2.2 and Section 3.2.3, respectively. We establish some basic properties of these new approximants analogously to what has been done for polynomial and other rational Arnoldi approaches in, e.g., [7, 31, 55, 56, 58, 93]. To conclude the section we report the results of a few numerical experiments, comparing the various approximation

strategies and showing the potential benefit of using general rational Krylov spaces over polynomial Krylov spaces.

3.2.1. Rational Arnoldi approximation to $f(A)\boldsymbol{b}$ by explicit projection. Let us consider an orthonormal (quasi-)RAD, say $A\widehat{V}_m\underline{\widehat{K}_{m-1}} = \widehat{V}_m\underline{\widehat{H}_{m-1}}$, for the moment of order $m-1$ instead of m. The corresponding rational Arnoldi approximation to $f(A)\boldsymbol{b}$ is defined as

$$\boldsymbol{f}_m := \widehat{V}_m f(\widehat{V}_m^* A \widehat{V}_m)\widehat{V}_m^*\boldsymbol{b}, \tag{3.15}$$

provided that $f(\widehat{V}_m^* A\widehat{V}_m)$ is defined [7, 55, 56, 58]. In order to avoid the explicit projection $\widehat{V}_m^* A\widehat{V}_m$, we can extend the (quasi-)RAD of order $m-1$ to order m with a single infinite pole $\xi_m = \infty$, thus obtaining $A\widehat{V}_{m+1}\underline{\widehat{K}_m} = A\widehat{V}_m\widehat{K}_m = \widehat{V}_{m+1}\underline{\widehat{H}_m}$, which provides $\widehat{V}_m^* A\widehat{V}_m = \widehat{H}_m\widehat{K}_m^{-1}$. The authors in [58] develop a version of the rational Arnoldi algorithm for the approximation of $f(A)\boldsymbol{b}$ as indicated by (3.15) and at every iteration the pole at infinity is added temporarily to obtain the approximate $\boldsymbol{f}_m$. A more efficient, but also more complicated, strategy which uses an additional vector that is being temporarily added and removed from the RAD is proposed in [55, Section 6.1]. We propose a simple solution for avoiding the explicit projection: the use of the matrices $\underline{\widehat{K}_m^\dagger}\underline{\widehat{H}_m}$, whose eigenvalues are the standard Ritz values, or $\left(\underline{\widehat{H}_m^\dagger}\underline{\widehat{K}_m}\right)^{-1}$, whose eigenvalues are the harmonic Ritz values. These are the strategies that we now formally introduce.

3.2.2. Standard rational Arnoldi approximation to $f(A)\boldsymbol{b}$. We start by considering rational Arnoldi approximations to $f(A)\boldsymbol{b}$ related to standard Ritz values.

Definition 3.13. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, and f a function such that $f(A)$ is defined. If $f(\underline{K_m^\dagger}\underline{H_m})$ is defined, we call*

$$\boldsymbol{f}_m^\infty := (V_{m+1}\underline{K_m})f(\underline{K_m^\dagger}\underline{H_m})(V_{m+1}\underline{K_m})^\dagger\boldsymbol{b}, \tag{3.16}$$

the (standard) rational Arnoldi approximation to $f(A)\boldsymbol{b}$ with respect to (3.1).

If (3.1) is orthonormal, then $(V_{m+1}\underline{K_m})^\dagger = \underline{K_m^\dagger}V_{m+1}^*$ since V_{m+1} is orthonormal and, therefore, $(V_{m+1}\underline{K_m})^\dagger\boldsymbol{b} = \beta\underline{K_m^\dagger}\boldsymbol{e}_1$, where the scalar $\beta = \boldsymbol{v}_1^*\boldsymbol{b}$ satisfies $|\beta| = \|\boldsymbol{b}\|_2$. Hence, (3.16) reads

$$\boldsymbol{f}_m^\infty = (V_{m+1}\underline{K_m})f(\underline{K_m^\dagger}\underline{H_m})\underline{K_m^\dagger}(\beta\boldsymbol{e}_1), \quad \beta = \boldsymbol{v}_1^*\boldsymbol{b}. \tag{3.17}$$

In the following we identify functions f for which the rational Arnoldi approximation is exact in general. The results and, partly, the proofs are analogous to those in [55, 93] for approximations of the form (3.15). We start with a technical lemma.

Lemma 3.14. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$. If $q_m(\underline{K_m^\dagger H_m})$ is nonsingular, then $q_m(A)^{-1}\boldsymbol{b} = (V_{m+1}\underline{K_m})q_m(\underline{K_m^\dagger H_m})^{-1}(V_{m+1}\underline{K_m})^\dagger \boldsymbol{b}$.*

Proof. To simplify the notation, let us label $W_m := V_{m+1}\underline{K_m}$, and $\boldsymbol{q} := q_m(A)^{-1}\boldsymbol{b}$. From (3.1) we have $\underline{K_m^\dagger H_m} = W_m^\dagger A W_m$, and by Proposition 3.3 we have $\mathcal{R}(W_m) = \mathcal{K}_m(A, \boldsymbol{q})$. We first show (by induction) that

$$W_m W_m^\dagger A^j \boldsymbol{q} = W_m(\underline{K_m^\dagger H_m})^j W_m^\dagger \boldsymbol{q}, \quad j = 0, 1, \ldots, m. \tag{3.18}$$

If $j = 0$, eq. (3.18) holds true. Assume that (3.18) holds for $j = 0, 1, \ldots, k < m$. Then

$$\begin{aligned} W_m W_m^\dagger A^{j+1}\boldsymbol{q} &= W_m W_m^\dagger A W_m W_m^\dagger A^j \boldsymbol{q} \\ &= W_m W_m^\dagger A W_m (\underline{K_m^\dagger H_m})^j W_m^\dagger \boldsymbol{q} \\ &= W_m (\underline{K_m^\dagger H_m})^{j+1} W_m^\dagger \boldsymbol{q}. \end{aligned}$$

By linearity we have, for $q_m \in \mathcal{P}_m$, that

$$W_m W_m^\dagger \boldsymbol{b} = W_m W_m^\dagger q_m(A)\boldsymbol{q} = W_m q_m(\underline{K_m^\dagger H_m}) W_m^\dagger \boldsymbol{q}.$$

Multiplying with $W_m q_m(\underline{K_m^\dagger H_m})^{-1} W_m^\dagger$ from the left we obtain

$$W_m q_m(\underline{K_m^\dagger H_m})^{-1} W_m^\dagger \boldsymbol{b} = W_m W_m^\dagger \boldsymbol{q}.$$

Finally, as $W_m W_m^\dagger \boldsymbol{q} = \boldsymbol{q} = q_m(A)^{-1}\boldsymbol{b}$ we obtain the result. ∎

Theorem 3.15. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ such that $q_m(\underline{K_m^\dagger H_m})$ is nonsingular. If $r_{m-1,m} \in \mathcal{P}_{m-1}/q_m$, and $r_{m,m} \in \mathcal{P}_m/q_m$, then*

(i) $r_{m-1,m}(A)\boldsymbol{b} = (V_{m+1}\underline{K_m}) r_{m-1,m}(\underline{K_m^\dagger H_m})(V_{m+1}\underline{K_m})^\dagger \boldsymbol{b}$, *and*

(ii) $(V_{m+1}\underline{K_m})(V_{m+1}\underline{K_m})^\dagger r_{m,m}(A)\boldsymbol{b} = (V_{m+1}\underline{K_m}) r_{m,m}(\underline{K_m^\dagger H_m})(V_{m+1}\underline{K_m})^\dagger \boldsymbol{b}$.

Proof. We consider the exactness (i) of the representation for $r_{m-1,m}(A)\boldsymbol{b}$ first, and then that of the projection (ii) of $r_{m,m}(A)\boldsymbol{b}$ onto $\mathcal{R}(V_{m+1}\underline{K_m})$. Let thus W_m and $\boldsymbol{q}$ be as in the proof of Lemma 3.14. It is enough to show that

$$A^j q_m(A)^{-1}\boldsymbol{b} = W_m(\underline{K_m^\dagger H_m})^j q_m(\underline{K_m^\dagger H_m})^{-1} W_m^\dagger \boldsymbol{b}, \tag{3.19}$$

for $j = 0, 1, \ldots, m-1$. If $j = 0$, eq. (3.19) reduces to Lemma 3.14. Let us now assume that (3.19) holds for $j = 0, 1, \ldots, k < m-1$. For $j+1$ we have

$$\begin{aligned} A^{j+1} q_m(A)^{-1} \boldsymbol{b} &= W_m W_m^\dagger A^{j+1} q_m(A)^{-1} \boldsymbol{b} \\ &= W_m W_m^\dagger A W_m (\underline{K_m^\dagger H_m})^j q_m(\underline{K_m^\dagger H_m})^{-1} W_m^\dagger \boldsymbol{b} \\ &= W_m (\underline{K_m^\dagger H_m})^{j+1} q_m(\underline{K_m^\dagger H_m})^{-1} W_m^\dagger \boldsymbol{b}, \end{aligned}$$

where the first equality follows from $A^{j+1} q_m(A)^{-1} \boldsymbol{b} \in \mathcal{R}(W_m) = \mathcal{K}_m(A, q_m(A)^{-1}\boldsymbol{b})$, and in the second we use the inductive hypothesis. To prove (ii), it only remains to show that

$$W_m W_m^\dagger A^m \boldsymbol{q} = W_m (\underline{K_m^\dagger H_m})^m q_m(\underline{K_m^\dagger H_m})^{-1} W_m^\dagger \boldsymbol{b},$$

which follows by rewriting $A^m = A A^{m-1}$, and using $A^{m-1}\boldsymbol{q} = W_m W_m^\dagger A^{m-1} \boldsymbol{q}$. □

Therefore, the standard rational Arnoldi approximation to $f(A)\boldsymbol{b}$ with respect to (3.1) is exact for all rational functions $r_{m-1,m}$ such that $r_{m-1,m}(A)\boldsymbol{b} \in \mathcal{R}(V_{m+1}\underline{K_m})$ resides in the space the approximation is extracted from. The same holds for the approximation (3.15); see, e.g., [55, 56]. In fact, as both approximants are independent from the choice of the basis, but depend solely on the space, the two approximants coincide if $\mathcal{R}(V_{m+1}\underline{K_m}) = \mathcal{R}(\widehat{V}_m)$. By Proposition 3.3, these two spaces are equal if $\mathcal{R}(\widehat{V}_m) = \mathcal{Q}_m(A, \boldsymbol{b}, \widehat{q}_{m-1})$ and $\mathcal{R}(V_{m+1}) = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, \widehat{q}_{m-1})$, that is, if they have the same poles, with $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, \widehat{q}_{m-1})$ having an additional pole at infinity. The position of the infinite pole in the pole sequence is, however, irrelevant; it may well be included as the first pole and left once and for all! The following result provides a characterisation for general functions through standard Ritz values.

Theorem 3.16. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ such that $q_m(\underline{K_m^\dagger H_m})$ is nonsingular, and $f(\underline{K_m^\dagger H_m})$ is defined. Then*

$$(V_{m+1}\underline{K_m}) f(\underline{K_m^\dagger H_m}) (V_{m+1}\underline{K_m})^\dagger \boldsymbol{b} = r_{m-1,m}(A)\boldsymbol{b},$$

where $r_{m-1,m} = p_{m-1}/q_m \in \mathcal{P}_{m-1}/q_m$ is the unique rational function of type at most $(m-1, m)$, with fixed denominator q_m, such that $r_{m-1,m}$ interpolates f in rational Hermite sense, i.e., p_{m-1} interpolates $f q_m$ in Hermite sense, on $\Lambda(\underline{K_m^\dagger H_m})$.

Proof. By Proposition 3.4, the matrix $\underline{K_m^\dagger H_m}$ is nonderogatory. Consequently, its minimal polynomial is its characteristic polynomial and has degree m, which implies the existence and uniqueness of p_{m-1} such that $(f q_m)(\underline{K_m^\dagger H_m}) = p_{m-1}(\underline{K_m^\dagger H_m})$.

Multiplying the last relation with $q_m(\underline{K_m^\dagger H_m})^{-1}$ on the right, gives

$$(fq_m)(\underline{K_m^\dagger H_m})q_m(\underline{K_m^\dagger H_m})^{-1} = f(\underline{K_m^\dagger H_m}) = r_{m-1,m}(\underline{K_m^\dagger H_m}),$$

where the first equality follows from [60, Theorem 1.15]. Therefore,

$$\begin{aligned}(V_{m+1}\underline{K_m})f(\underline{K_m^\dagger H_m})(V_{m+1}\underline{K_m})^\dagger \boldsymbol{b} &= (V_{m+1}\underline{K_m})r_{m-1,m}(\underline{K_m^\dagger H_m})(V_{m+1}\underline{K_m})^\dagger \boldsymbol{b}\\ &= r_{m-1,m}(A)\boldsymbol{b},\end{aligned}$$

where the last equality follows from Theorem 3.15. ∎

Let us return to Example 3.12, and let us assume that $f(\underline{K_1^\dagger H_1})$ is defined. Then, since $\boldsymbol{b} \perp \mathcal{R}(V_2\underline{K_1})$, we have $\boldsymbol{f}_1^\infty = \boldsymbol{0}$. In this case it is advantageous to replace the search space $\mathcal{R}(V_{m+1}\underline{K_m})$ with another space of the form $\mathcal{R}\big(\alpha V_{m+1}\underline{H_m} - \beta V_{m+1}\underline{K_m}\big)$, with $|\alpha|+|\beta| \neq 0$. It is clear from Proposition 3.3 that any two distinct such subspaces of $\mathcal{R}(V_{m+1}) = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ provide $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ in their union, which implies that $\boldsymbol{b} \neq \boldsymbol{0}$ can be orthogonal to one of them only. Furthermore, it follows from Proposition 3.3 that safe options in this regard are $\alpha, \beta \in \mathbb{C}$ such that $\beta/\alpha = \xi_j$ for some $j = 1, 2, \ldots, m$. Indeed if β/α is a pole of $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, then $\mathcal{R}\big(\alpha V_{m+1}\underline{H_m} - \beta V_{m+1}\underline{K_m}\big) = \mathcal{Q}_m(A, \boldsymbol{b}, \widehat{q}_{m-1})$, where $\widehat{q}_{m-1} \in \mathcal{P}_{m-1}$ is such that, formally, $\widehat{q}_{m-1}(z) = q_m(z)/(\alpha z - \beta)$, and, clearly, $\boldsymbol{b} \in \mathcal{Q}_m(A, \boldsymbol{b}, \widehat{q}_{m-1})$. In particular, if infinity is a pole, then the standard rational Arnoldi approximation does not exhibit this problem. These considerations bring us back to τ-harmonic Ritz values.

3.2.3. Harmonic rational Arnoldi approximation to $f(A)\boldsymbol{b}$. We now introduce the harmonic approximants similarly to the way we introduced the standard ones.

Definition 3.17. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, and f a function such that $f(A)$ is defined. If $f([\underline{H_m^\dagger K_m}]^{-1})$ is defined, we call*

$$\boldsymbol{f}_m^0 := (V_{m+1}\underline{H_m})f([\underline{H_m^\dagger K_m}]^{-1})(V_{m+1}\underline{H_m})^\dagger \boldsymbol{b}, \tag{3.20}$$

the harmonic rational Arnoldi approximation to $f(A)\boldsymbol{b}$ with respect to (3.1).

If (3.1) is orthonormal, then $(V_{m+1}\underline{H_m})^\dagger \boldsymbol{b} = \beta \underline{H_m^\dagger}\boldsymbol{e}_1$, where the scalar $\beta = \boldsymbol{v}_1^*\boldsymbol{b}$ satisfies $|\beta| = \|\boldsymbol{b}\|_2$, and hence (3.20) reads

$$\boldsymbol{f}_m^0 = (V_{m+1}\underline{H_m})f([\underline{H_m^\dagger K_m}]^{-1})\underline{H_m^\dagger}(\beta\boldsymbol{e}_1), \quad \beta = \boldsymbol{v}_1^*\boldsymbol{b}. \tag{3.21}$$

If A is nonsingular, then the harmonic Arnoldi approximation to $f(A)\boldsymbol{b}$ equals the standard Arnoldi approximation to $[f \circ (z \mapsto \frac{1}{z})](A^{-1})\boldsymbol{b}$. This fact is key to transferring results from the standard to the harmonic case.

Theorem 3.18. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ such that A, $\underline{H_m^\dagger K_m}$ and $q_m([\underline{H_m^\dagger K_m}]^{-1})$ are nonsingular. If $\widehat{r}_{m-1,m}(z) = zr_{m-1,m}(z)$, with $r_{m-1,m} \in \mathcal{P}_{m-1}/q_m$, and $r_{m,m} \in \mathcal{P}_m/q_m$, then*

(i) $\widehat{r}_{m-1,m}(A)\boldsymbol{b} = (V_{m+1}\underline{H_m})\widehat{r}_{m-1,m}([\underline{H_m^\dagger K_m}]^{-1})(V_{m+1}\underline{H_m})^\dagger \boldsymbol{b}$, *and*

(ii) $(V_{m+1}\underline{H_m})(V_{m+1}\underline{H_m})^\dagger r_{m,m}(A)\boldsymbol{b} = (V_{m+1}\underline{H_m})r_{m,m}([\underline{H_m^\dagger K_m}]^{-1})(V_{m+1}\underline{H_m})^\dagger \boldsymbol{b}$.

Proof. Since $\widehat{r}_{m-1,m}(z)\big|_{z=A} = \widehat{r}_{m-1,m}(z^{-1})\big|_{z=A^{-1}}$, it is useful to introduce $\breve{r}_{m-1,m}(z) := \widehat{r}_{m-1,m}(z^{-1})$, so that $\widehat{r}_{m-1,m}(A)\boldsymbol{b} = \breve{r}_{m-1,m}(A^{-1})\boldsymbol{b}$. It can be shown, for instance by considering the partial fraction form of $\widehat{r}_{m-1,m}$, that $\breve{r}_{m-1,m} \in \mathcal{P}_{m-1}/\breve{q}_m$, where the formal roots of $\breve{q}_m$ are $\{\xi_j^{-1}\}_{j=1}^m$, that is, the inverses of the formal roots $\{\xi_j\}_{j=1}^m$ of q_m. Multiplying (3.1) from the left with A^{-1} we arrive at $A^{-1}V_{m+1}\underline{H_m} = V_{m+1}\underline{K_m}$, which is an RAD for $\mathcal{Q}_{m+1}(A^{-1}, \boldsymbol{b}, \breve{q}_m)$; see also the discussion following (2.42). The standard Ritz approximation with this RAD is exact for $\breve{r}_{m-1,m}$ by Theorem 3.15, hence

$$\begin{aligned}\widehat{r}_{m-1,m}(A)\boldsymbol{b} &= \breve{r}_{m-1,m}(A^{-1})\boldsymbol{b} \\ &= (V_{m+1}\underline{H_m})\breve{r}_{m-1,m}(\underline{H_m^\dagger K_m})(V_{m+1}\underline{H_m})^\dagger \boldsymbol{b} \\ &= (V_{m+1}\underline{H_m})\widehat{r}_{m-1,m}([\underline{H_m^\dagger K_m}]^{-1})(V_{m+1}\underline{H_m})^\dagger \boldsymbol{b},\end{aligned}$$

and (i) follows. Statement (ii) follows analogously. ∎

Finally, we establish the connection of the harmonic rational Arnoldi approximation with rational interpolation.

Theorem 3.19. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ such that A, $\underline{H_m^\dagger K_m}$ and $q_m([\underline{H_m^\dagger K_m}]^{-1})$ are nonsingular, and $f([\underline{H_m^\dagger K_m}]^{-1})$ is defined. Then*

$$(V_{m+1}\underline{H_m})f([\underline{H_m^\dagger K_m}]^{-1})(V_{m+1}\underline{H_m})^\dagger \boldsymbol{b} = \widehat{r}_{m-1,m}(A)\boldsymbol{b},$$

where $\widehat{r}_{m-1,m}(z) = zp_{m-1}(z)/q_m(z)$, with $p_{m-1} \in \mathcal{P}_{m-1}$, is the unique rational function of type at most (m, m), with fixed denominator q_m, and a fixed root at zero, such that p_{m-1} interpolates $z \mapsto f(z)q_m(z)/z$, in Hermite sense, on $\Lambda([\underline{H_m^\dagger K_m}]^{-1})$.

Proof. By Proposition 3.8, the matrix $\underline{H_m^\dagger K_m}$ is nonderogatory. Since for every Jordan block $J(\lambda)$ in $\underline{H_m^\dagger K_m}$ there is a Jordan block $J(\lambda^{-1})$ of the same size in $(\underline{H_m^\dagger K_m})^{-1}$,

cf. [60, Theorem 1.36 with $f(z) = z^{-1}$]), the matrix $(\underline{H_m^\dagger K_m})^{-1}$ is nonderogatory as well. The uniqueness of $p_{m-1} \in \mathcal{P}_{m-1}$ such that $(z \mapsto f(z)q_m(z)/z)([\underline{H_m^\dagger K_m}]^{-1}) = p_{m-1}([\underline{H_m^\dagger K_m}]^{-1})$ follows. The later equality provides the relation

$$f([\underline{H_m^\dagger K_m}]^{-1}) = \widehat{r}_{m-1,m}([\underline{H_m^\dagger K_m}]^{-1}),$$

which finalizes the proof with the use of Theorem 3.18. □

Theorem 3.19 is stated as is in order to highlight the fixed target $\tau = 0$. It, however, does imply that $z \mapsto zp_{m-1}(z)$ interpolates $z \mapsto f(z)q_m(z)$, and, therefore, that $\widehat{r}_{m-1,m}$ interpolates f. The standard and harmonic rational Arnoldi approximations are generalisations, of polynomial Arnoldi approximations with $f(z) = z^{-1}$ to rational Arnoldi for "any" function f, of FOM and GMRES, respectively.

Remark 3.20. Let $AV_m = V_{m+1}\underline{H_m}$ be a polynomial RAD with A nonsingular, and let $\boldsymbol{b} = V_{m+1}\boldsymbol{e}_1$. Then, the standard rational Arnoldi approximation to $A^{-1}\boldsymbol{b}$ is $V_m H_m^{-1}\boldsymbol{e}_1$, which is the FOM approximation. Similarly, the harmonic rational Arnoldi approximation to $A^{-1}\boldsymbol{b}$ equals the GMRES approximation. An interesting discussion on the convergence of FOM and GMRES based on standard and harmonic Ritz values can be found in [45].

From (3.10) we see how to define the more general τ-harmonic approximants, for which generalisations of Theorems 3.18–3.19 may be established; see also Lemma 3.11.

Definition 3.21. *Let (3.1) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, and f a function such that $f(A)$ is defined, and let $\tau \in \mathbb{C}\backslash\Lambda(A)$. If $f([\underline{L_m^\dagger K_m}]^{-1} + \tau I_m)$, with $\underline{L_m} := \underline{H_m} - \tau\underline{K_m}$, is defined, we call*

$$\boldsymbol{f}_m^\tau := (V_{m+1}\underline{L_m})f([\underline{L_m^\dagger K_m}]^{-1} + \tau I_m)(V_{m+1}\underline{L_m})^\dagger \boldsymbol{b}, \tag{3.22}$$

the τ-harmonic rational Arnoldi approximation to $f(A)\boldsymbol{b}$ with respect to (3.1).

As discussed at the end of Section 3.2.2, attractive choices for τ include the poles of the rational Krylov space in question. We conclude this section with an example similar to [56, Example 3.5].

3.2.4. Rational Arnoldi approximation of $f(A)\boldsymbol{b}$ for Markov functions f. We now briefly summarise the automated pole selection strategy designed in [57, 58] for the rational Arnoldi approximation of $f(A)\boldsymbol{b}$ for Markov functions f, i.e., functions of

Algorithm 3.4 Rational Arnoldi with automated pole selection for $f(A)\boldsymbol{b}$. [57, 58]

Input: $A \in \mathbb{C}^{N,N}$, $\boldsymbol{b} \in \mathbb{C}^N$, a set $\Xi \subset \overline{\mathbb{C}}$ for selecting the poles to approximate $f(A)\boldsymbol{b}$, and the maximal number of iterations $m < d(A, \boldsymbol{b})$.

Output: Orthonormal RAD $AV_{\ell+1}\underline{K_\ell} = V_{\ell+1}\underline{H_\ell}$ of order $\ell \leq m$.

1. Set $\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|_2$, and $\xi_1 = \infty$.
2. **for** $j = 1, 2, \ldots, m$ **do**
3. Choose scalars $\mu_j, \nu_j \in \mathbb{C}$ such that $\mu_j/\nu_j = \xi_j$.
4. Choose an admissible continuation pair $(\eta_j/\rho_j, \boldsymbol{t}_j) \in \overline{\mathbb{C}} \times \mathbb{C}^j$.
5. Compute $\boldsymbol{w}_{j+1} := (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)V_j \boldsymbol{t}_j$.
6. Orthogonalize $\widehat{\boldsymbol{v}}_{j+1} := \boldsymbol{w}_{j+1} - V_j \boldsymbol{c}_j$, where $\boldsymbol{c}_j := V_j^* \boldsymbol{w}_{j+1}$.
7. Normalize $\boldsymbol{v}_{j+1} := \widehat{\boldsymbol{v}}_{j+1}/c_{j+1,j}$, where $c_{j+1,j} := \|\widehat{\boldsymbol{v}}_{j+1}\|_2$.
8. Set $\underline{\boldsymbol{k}_j} := \nu_j \underline{\boldsymbol{c}_j} - \rho_j \underline{\boldsymbol{t}_j}$ and $\underline{\boldsymbol{h}_j} := \mu_j \underline{\boldsymbol{c}_j} - \eta_j \underline{\boldsymbol{t}_j}$, where $\underline{\boldsymbol{t}_j} = \begin{bmatrix} \boldsymbol{t}_j \\ 0 \end{bmatrix}$, and $\underline{\boldsymbol{c}_j} = \begin{bmatrix} \boldsymbol{c}_j \\ c_{j+1,j} \end{bmatrix}$.
9. Compute $\boldsymbol{f}_j^\infty \approx f(A)\boldsymbol{b}$, and stop if a good approximation is obtained.
10. Let $\xi_{j+1} = \operatorname{argmin}_{\xi \in \Xi} |s_j(\xi)|$, where $s_j = \chi_j^\infty / q_j$.
11. **end for**

the form $f(z) = \int_\Gamma \frac{\mathrm{d}\gamma(x)}{z-x}$, with a (complex) measure γ supported on a prescribed closed set $\Gamma \subset \mathbb{C}$. For examples and applications we refer to [57, 58].

To obtain a rational Arnoldi approximation to $f(A)\boldsymbol{b}$, we need to construct a rational Krylov space and then extract the approximation, perhaps using an extraction procedure just discussed. For the latter part let us focus on standard rational Arnoldi approximants with one pole at infinity. In this way we obtain the same extraction procedure as in [57, 58], although implemented differently. In [57, 58] the authors use explicit projection by adding and removing a pole at infinity at every iteration of the rational Arnoldi algorithm, as we already discussed in Section 3.2.1. Regarding the construction of the rational Krylov space, the main question is how to determine the (other) poles. For the first pole we use $\xi_1 = \infty$. Let us now assume that we have computed an RAD $AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}$ of order m for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, and we wish to extend it to an RAD of order $m+1$. To this end, let χ_m^∞ be the characteristic polynomial of $\underline{K_m}^\dagger \underline{H_m}$, same as in Proposition 3.4, and recall that Theorem 3.16 asserts that the standard rational Arnoldi approximation $\boldsymbol{f}_m^\infty = r(A)\boldsymbol{b}$ for $f(A)\boldsymbol{b}$ is such that r interpolates f in rational Hermite sense on $\Lambda(\underline{K_m}^\dagger \underline{H_m})$. Finally, it is suggested in [57, 58] to use ξ_{m+1} such that

$$|s_m(\xi_{m+1})| = \min_{z \in \Gamma} |s_m(z)|,$$

where

$$s_m(z) = \frac{\chi_m^\infty(z)}{q_m(z)}, \tag{3.23}$$

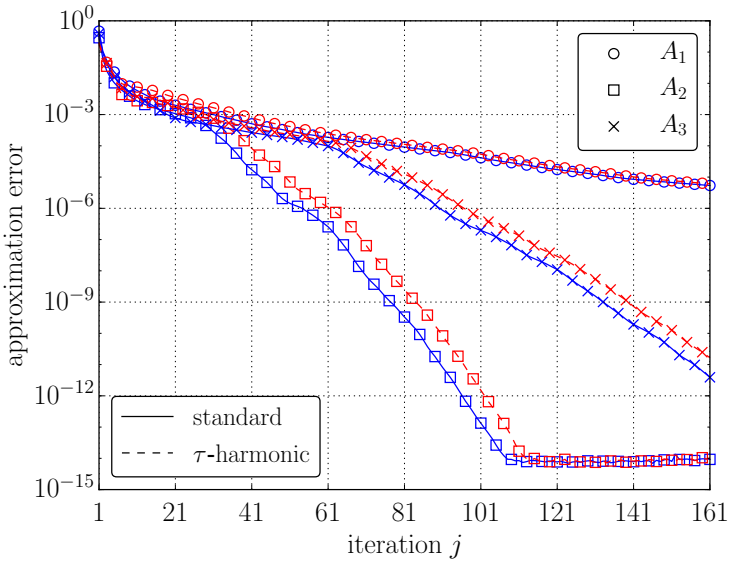


Figure 3.2: Relative approximation error $\|\boldsymbol{f}_{\ell,j}^{\tau} - A_\ell^{\frac{1}{2}}\boldsymbol{b}\|_2/\|A_\ell^{\frac{1}{2}}\boldsymbol{b}\|_2$ of the standard (blue solid lines) and τ-harmonic (red dashed lines) polynomial Arnoldi approximations $\boldsymbol{f}_{\ell,j}^{\tau}$ to $A_\ell^{\frac{1}{2}}\boldsymbol{b}$, with $\tau = -1$ and the three matrices A_ℓ for $\ell = 1, 2, 3$ specified in Section 3.2.5.

as the next pole. For more details see [58, Section 3]. The pseudoce of the algorithm is given in Algorithm 3.4. A discussion of possible stopping criteria is contained in [58, Section 4]. In RKToolbox Example 7.3 we provide a simple RKToolbox implemenation of Algorithm 3.4.

3.2.5. Numerical example. We compare the various approximation strategies with three different symmetric positive definite matrices $A_\ell \in \mathbb{R}^{N,N}$, with $N = 729$, a unit 2-norm random starting vector $\boldsymbol{b}$, and with the principal square root function $f(z) = z^{\frac{1}{2}}$. Let

$$T_n = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n,n},$$

and, further, $\widehat{A}_1 = T_N$, $\widehat{A}_2 = T_{29} \oplus T_{29}$. The matrices A_1 and A_2 are obtained by shifting and scaling $\widehat{A}_1$ and $\widehat{A}_2$, respectively, so that their spectral interval becomes $[0.01, 100]$. For A_3 we take a diagonal matrix with eigenvalues equispaced in the same interval. We consider the approximation from a polynomial Krylov space (i.e., a rational Krylov space with all poles at infinity), a rational Krylov space with the pole $\xi = -1$ used repeatedly, and the adaptive approach from Section 3.2.4. In Figures 3.2–3.4 we report the relative error $\frac{\|\boldsymbol{f}_{\ell,j}^{\tau} - \sqrt{A_\ell}\boldsymbol{b}\|_2}{\|\sqrt{A_\ell}\boldsymbol{b}\|_2}$ of the τ-harmonic rational Arnoldi approximations

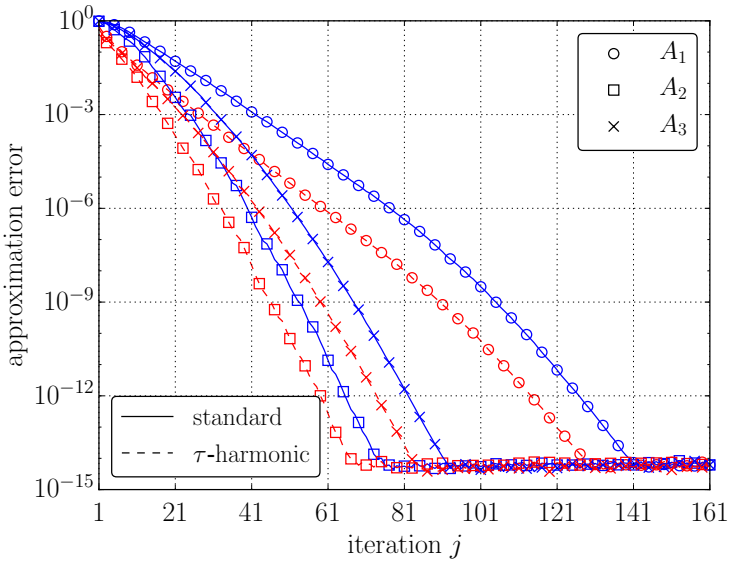


Figure 3.3: Relative approximation error $\|\boldsymbol{f}_{\ell,j}^{\tau} - A_\ell^{\frac{1}{2}}\boldsymbol{b}\|_2 / \|A_\ell^{\frac{1}{2}}\boldsymbol{b}\|_2$ of the standard (blue solid lines) and τ-harmonic (red dashed lines) rational Arnoldi approximations $\boldsymbol{f}_{\ell,j}^{\tau}$ to $A_\ell^{\frac{1}{2}}\boldsymbol{b}$, with $\tau = -1$ and the matrices A_ℓ for $\ell = 1, 2, 3$ specified in Section 3.2.5

to $\sqrt{A_\ell}\boldsymbol{b}$ as the iteration j progresses. We consider $\tau = \infty$, yielding the standard rational Arnoldi approximations, and $\tau = \xi$. While both extraction procedures struggle with the polynomial Arnoldi space, cf. Figure 3.2, the rational Krylov space approach converges sooner; see Figure 3.3. Interestingly, the harmonic approximants substantially outperform the standard ones in the rational case for this example. Furthermore, we can observe that the Algorithm 3.4 converges (much) faster than when poles as in Figures 3.2–3.3 are used. Furthermore, harmonic rational Arnoldi approximants outperform the standard rational Arnoldi approximants for all three matrices A_ℓ.

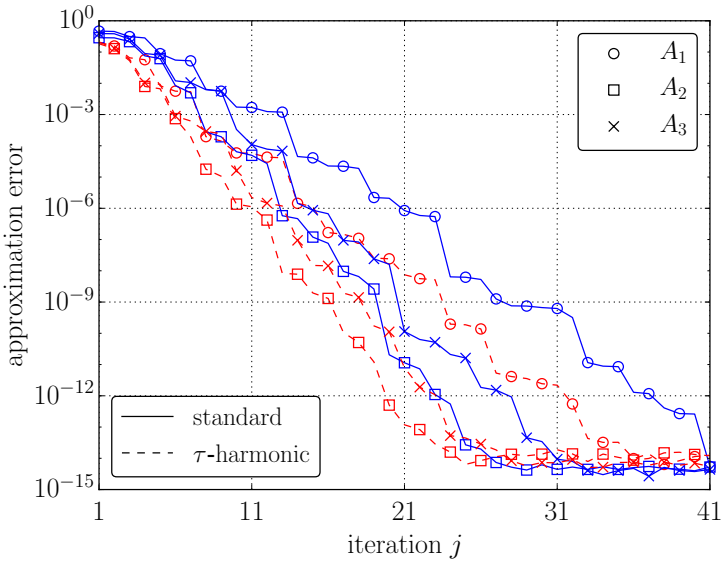


Figure 3.4: Relative approximation error $\|\boldsymbol{f}_{\ell,j} - A_\ell^{\frac{1}{2}}\boldsymbol{b}\|_2 / \|A_\ell^{\frac{1}{2}}\boldsymbol{b}\|_2$ of the standard (blue solid lines) and harmonic (red dashed lines) adaptive rational Arnoldi approximations $\boldsymbol{f}_{\ell,j}$ to $A_\ell^{\frac{1}{2}}\boldsymbol{b}$, with the matrices A_ℓ for $\ell = 1, 2, 3$ specified in Section 3.2.5.

4 Continuation pairs and parallelisation

An interesting feature of the rational Arnoldi algorithm is that, under certain conditions, basis vectors can be computed in parallel. Take, for example, m distinct poles $\xi_1, \xi_2, \ldots, \xi_m \in \mathbb{C} \setminus \Lambda(A)$. Then

$$\operatorname{span}\{\boldsymbol{b}, (A-\xi_1 I)^{-1}\boldsymbol{b}, (A-\xi_2 I)^{-1}\boldsymbol{b}, \ldots, (A-\xi_m I)^{-1}\boldsymbol{b}\}$$

is the rational Krylov space $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ with $q_m(z) = \prod_{j=1}^{m}(z-\xi_j)$, and clearly all the basis vectors can be computed simultaneously from $\boldsymbol{b}$. This is particularly attractive in the typical case when solving the linear systems $(A-\xi_j I)\boldsymbol{x}_j = \boldsymbol{b}$ is the dominant computational cost in the rational Arnoldi algorithm.

Unfortunately, this naive parallelisation approach may quickly lead to numerical instabilities. An instructive example is that of a diagonal matrix $A = \operatorname{diag}(\lambda_i)_{i=1}^{N}$, for which the basis vectors $\boldsymbol{x}_j = (A-\xi_j I)^{-1}\boldsymbol{b}$ are the columns of a Cauchy-like matrix

$$X = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_m \end{bmatrix} \in \mathbb{C}^{N,m} \quad \text{with} \quad x_{ij} := \boldsymbol{e}_i^T X \boldsymbol{e}_j = \frac{\boldsymbol{e}_j^T \boldsymbol{b}}{\lambda_i - \xi_j}.$$

Hence, X satisfies the Sylvester equation $AX - XB = C$ with rank-1 right-hand side $C = \boldsymbol{b}\begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}$ and $B = \operatorname{diag}(\xi_j)_{j=1}^{m}$. If the eigenvalues of A and B are well separated, e.g., by a straight line, the singular values of X decay exponentially as m increases (see [46]). Thus the matrix X will be exponentially ill-conditioned, which may cause problems during the Gram–Schmidt orthogonalization process. Available rounding error analyses of the modified Gram–Schmidt procedure (with reorthogonalization) typically assume that the basis X to be orthogonalized is numerically nonsingular, i.e., $g(m)\kappa(X)\varepsilon < 1$, where $\kappa(X)$ is a condition number of X, ε is the machine precision, and g is a slowly growing function in m (see, e.g., [38, 47]). Without this condition

being satisfied, as in our case, there is no guarantee that the Gram–Schmidt procedure is backward stable, i.e., that it computes the exact QR factorization of a nearby matrix $X + E$, with E being of small norm relative to X.

The potential for exponential growth in the condition number of a rational Krylov basis seems to discourage any attempt to parallelise the rational Arnoldi algorithm, and indeed only very few authors have considered this problem up to date. Most notably, Skoogh [98, 99] presents and compares, mostly from an algorithmic point of view, two (distributed memory) parallel variants. He notes that "generally the parallel rational Krylov programs get fewer converged eigenvalues than the corresponding sequential program" and that potential numerical instabilities may arise during the orthogonalization phases. Further remarks are contained in [49, Section 7] and [55, Section 6.5]. However, practical recommendations on how to best parallelise the rational Arnoldi algorithm seem to be lacking. The main goal of this chapter is to fill this gap.

To this end, in Section 4.1 we formally introduce the notion of *continuation pairs*, which represent the free parameters to be chosen during the rational Arnoldi algorithm, and link them to the condition number of the nonorthogonal rational Krylov basis. In Section 4.2 we propose and analyze a framework for constructing *near-optimal* continuation pairs; near-optimal in the sense of minimising the growth of the condition number of this nonorthogonal basis. These considerations are related to the sequential variant of the rational Arnoldi algorithm presented in Algorithm 2.2, and easily extend to the variant presented in Algorithm 2.3; see Remark 4.9. Although these considerations are interesting in their own right, in Sections 4.3–4.4 we finally exploit them to obtain several parallel variants of Algorithm 2.2. Specifically, in Section 4.3 we discuss a generic parallel variant of the rational Arnoldi algorithm, list some canonical choices for continuation pairs, and adapt the previously developed near-optimal strategy to the parallel case. In Section 4.4 we provide a range of numerical experiments, comparing different continuation strategies and high-performance (parallel) implementations.

4.1 Continuation pairs

We assume to have the matrix A, the starting vector $\boldsymbol{b}$, the poles $\{\mu_j/\nu_j\}_{j=1}^m$, and now discuss the roles of the "internal" parameters ρ_j, η_j, and $\boldsymbol{t}_j$, a problem that can be

illustrated graphically as follows:

$$AV_m\underline{K_{m-1}} = V_m\underline{H_{m-1}} \quad \xrightarrow[\mu_m/\nu_m]{(\eta_m/\rho_m,\boldsymbol{t}_m)} \quad AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}. \tag{4.1}$$

To be precise, we study the influence of $(\eta_m/\rho_m, \boldsymbol{t}_m) \in \overline{\mathbb{C}} \times \mathbb{C}^m$ for the extension of an order $m-1$ RAD for $\mathcal{Q}_m(A, \boldsymbol{b}, q_{m-1})$, namely,

$$AV_m\underline{K_{m-1}} = V_m\underline{H_{m-1}}, \tag{4.2}$$

with the pole $\mu_m/\nu_m \in \overline{\mathbb{C}} \setminus \Lambda(A)$, into an RAD (2.6) for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ of order m.

Let us now find an admissible continuation pair for (4.1). For any $\eta/\rho \neq \mu/\nu$, the RAD (4.2) can be transformed into (see, e.g., Section 2.5.1)

$$(\nu A - \mu I)^{-1}(\rho A - \eta I)V_m\big(\nu\underline{H_{m-1}} - \mu\underline{K_{m-1}}\big) = V_m\big(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}}\big). \tag{4.3}$$

Set $\mu/\nu \equiv \mu_m/\nu_m$. If $\boldsymbol{t}_m \in \mathcal{R}(\nu_m\underline{H_{m-1}} - \mu_m\underline{K_{m-1}})$ then there exists a vector $\boldsymbol{z}_{m-1} \in \mathbb{C}^{m-1}$ such that $\boldsymbol{t}_m = \nu_m\underline{H_{m-1}}\boldsymbol{z}_{m-1} - \mu_m\underline{K_{m-1}}\boldsymbol{z}_{m-1}$. Specifically,

$$\boldsymbol{w}_{m+1} = (\nu_m A - \mu_m I)^{-1}(\rho A - \eta I)V_m\boldsymbol{t}_m = V_m\big(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}}\big)\boldsymbol{z}_{m-1} \in \mathcal{R}(V_m),$$

showing that a continuation pair, independently of the continuation root, is not admissible if $\boldsymbol{t}_m \in \mathcal{R}(\nu_m\underline{H_{m-1}} - \mu_m\underline{K_{m-1}})$. This was first observed in [90] and led the author to suggest a nonzero left null vector $\boldsymbol{q}_m$ of $\nu_m\underline{H_{m-1}} - \mu_m\underline{K_{m-1}}$ as a continuation vector.

Currently, the choices $\boldsymbol{t}_m = \boldsymbol{e}_m$ and $\boldsymbol{t}_m = \boldsymbol{q}_m$ appear to be dominant in the literature; see, e.g., [90, 109]. Note that $\boldsymbol{t}_m = \boldsymbol{e}_m$ may (with probability zero) be not admissible, i.e., we would not be able to expand the space with the obtained $\boldsymbol{w}_{m+1}$ even though the space is not yet A-invariant. Such a situation is called *unlucky breakdown*. Nevertheless, these two choices do appear to work well in practice, but, as we shall see, this is not always the case for the parallel variant. Moreover, continuation roots are frequently ignored. Typical choices, adopted without justification, are zero and infinity. An exception to this is [73], where a choice for $(\vartheta, \boldsymbol{t}_m)$ is recommended in a way such that $(\vartheta, V_m\boldsymbol{t}_m)$ is a *rough* approximation to an eigenpair of A.

We will now show that for the *sequential* rational Arnoldi algorithm, Algorithm 2.2, there exist continuation pairs which yield $\boldsymbol{w}_{m+1}$ such that $\boldsymbol{w}_{m+1} \perp \mathcal{R}(V_m)$. We refer to such continuation pairs as *optimal*, as we are mainly concerned with the condition number of the basis being orthogonalised.

Definition 4.1. *An admissible continuation pair* $(\eta_m/\rho_m, \boldsymbol{t}_m)$ *is called* optimal *for* (4.1) *if the condition* $(\nu_m A - \mu_m I)^{-1}(\rho_m A - \eta_m I)V_m \boldsymbol{t}_m \perp \mathcal{R}(V_m)$ *is satisfied.*

Thus, the *optimality* is related to the angle $\angle(\boldsymbol{w}_{m+1}, V_m)$ between the vector $\boldsymbol{w}_{m+1} = (\nu_m A - \mu_m I)^{-1}(\rho_m A - \eta_m I)V_m \boldsymbol{t}_m$ and the space $\mathcal{R}(V_m)$. For our purposes, the closer the angle is to $\frac{\pi}{2}$, the better. Equivalently, if the two RADs appearing in (4.1) are orthonormal, the continuation pair $(\eta_m/\rho_m, \boldsymbol{t}_m)$ is optimal for (4.1) if $(\nu_m A - \mu_m I)^{-1}(\rho_m A - \eta_m I)V_m \boldsymbol{t}_m$ is a scalar multiple of $\boldsymbol{v}_{m+1}$. The key observation is thus triggered by Theorem 2.16, which asserts that the new direction $\boldsymbol{v}_{m+1}$ we are interested in is predetermined by the parameters $(A, \boldsymbol{b}, q_m)$ defining $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$.

Recall that by Theorem 2.12 we have $\boldsymbol{v}_{m+1} = p_m(A)q_m(A)^{-1}\boldsymbol{v}_1$. Denote by η_m/ρ_m any root of p_m, and label with $\mu_m/\nu_m \equiv h_{m+1,m}/k_{m+1,m}$ the last pole of (2.6). Let $p_m(z) =: (\rho_m z - \eta_m)\breve{p}_{m-1}(z)$ and $q_m(z) = (\nu_m z - \mu_m)q_{m-1}(z)$ hold. We clearly have

$$\begin{aligned} \boldsymbol{v}_{m+1} &= (\nu_m A - \mu_m I)^{-1}(\rho_m A - \eta_m I)\breve{p}_{m-1}(A)q_{m-1}(A)^{-1}\boldsymbol{v}_1 \\ &= (\nu_m A - \mu_m I)^{-1}(\rho_m A - \eta_m I)V_m \boldsymbol{t}_m =: \mathcal{M}(A)V_m \boldsymbol{t}_m, \end{aligned} \tag{4.4}$$

where $\boldsymbol{t}_m$ satisfies $V_m \boldsymbol{t}_m = \breve{p}_{m-1}(A)q_{m-1}(A)^{-1}\boldsymbol{v}_1 \in \mathcal{Q}_{m-1}(A, \boldsymbol{b}, q_{m-1}) = \mathcal{R}(V_m)$. Now if (2.6) is an orthonormal RAD, and hence $\boldsymbol{v}_{m+1} \perp \mathcal{R}(V_m)$, we have just verified that $(\eta_m/\rho_m, \boldsymbol{t}_m)$ is an optimal continuation pair.

It proves useful to derive a closed formula for the optimal continuation vector $\boldsymbol{t}_m$. To this end, let $\boldsymbol{x}_m$ be a right generalized eigenvector of (H_m, K_m) corresponding to the eigenvalue η_m/ρ_m; i.e., $(\rho_m H_m - \eta_m K_m)\boldsymbol{x}_m = \boldsymbol{0}$. Right-multiplying the RAD of the form (4.3), with $\mu/\nu \equiv \mu_m/\nu_m$ and $\eta/\rho \equiv \eta_m/\rho_m$, but of order m, by $\boldsymbol{x}_m$ yields

$$\mathcal{M}(A)V_m\big(\nu_m H_m - \mu_m K_m\big)\boldsymbol{x}_m = (\rho_m h_{m+1,m} - \eta_m k_{m+1,m})(\boldsymbol{e}_m^T \boldsymbol{x}_m)\boldsymbol{v}_{m+1}. \tag{4.5}$$

This gives the optimal continuation vector provided that $\gamma_m = (\boldsymbol{e}_m^T \boldsymbol{x}_m)(\rho_m h_{m+1,m} - \eta_m k_{m+1,m}) \neq 0$, which holds true under the assumption that (2.6) is an RAD. Indeed, if $\gamma_m = 0$, then $V_m(\nu_m H_m - \mu_m K_m)\boldsymbol{x}_m$ is an eigenvector of A with eigenvalue η_m/ρ_m, which implies the non-existence of an RAD of order m with starting vector $\boldsymbol{v}_1$, as Proposition 3.1 shows. Let us now summarise our findings.

Proposition 4.2. *Let* (2.6) *be an orthonormal RAD, and let* $(\eta/\rho, \boldsymbol{x})$ *be an eigenpair of* (H_m, K_m)*. The continuation pair*

$$(\eta_m/\rho_m, \boldsymbol{t}_m) \equiv \big(\eta/\rho, \gamma^{-1}[\nu_m H_m - \mu_m K_m]\boldsymbol{x}\big), \tag{4.6}$$

with $\gamma = x_m(\rho_m h_{m+1,m} - \eta_m k_{m+1,m})$*, is optimal for* (4.1)*. Alternatively, any optimal continuation pair for* (4.1) *is, up to nonzero scaling of* $\boldsymbol{t}_m$*, of this form.*

4.2 Near-optimal continuation pairs

While Proposition 4.2 characterizes optimal continuation pairs precisely, it requires the last column of $(\underline{H_m}, \underline{K_m})$, which is not available without computing $\boldsymbol{v}_{m+1}$ in the first place. Our idea is to employ a rough approximation $(\underline{\widehat{H}_m}, \underline{\widehat{K}_m}) \approx (\underline{H_m}, \underline{K_m})$ to obtain a *near-optimal continuation pair*. We then quantify the approximation accuracy that is required in order to generate a well-conditioned rational Krylov basis.

4.2.1. The framework. Assume we are given an RAD of order $j-1$, namely,

$$AV_j\underline{K_{j-1}} = V_j\underline{H_{j-1}}. \tag{4.7}$$

We seek a near-optimal continuation pair $(\eta_j/\rho_j, \boldsymbol{t}_j)$ for expanding (4.7) into

$$AV_{j+1}\underline{K_j} = V_{j+1}\underline{H_j}, \tag{4.8}$$

using the pole $\xi_j = \mu_j/\nu_j$. To this end we make use of an *auxiliary* continuation pair $(\widehat{\eta}_j/\widehat{\rho}_j, \widehat{\boldsymbol{t}}_j)$, whose only requirement is being admissible. For example, it could be the one proposed by Ruhe [90]. Let us consider the associated linear system

$$(\nu_j A - \mu_j I)\boldsymbol{w} = (\widehat{\rho}_j A - \widehat{\eta}_j I)V_j\widehat{\boldsymbol{t}}_j. \tag{4.9}$$

The solution $\boldsymbol{w}$ could be used to expand the rational Krylov space we are constructing. However, to obtain a near-optimal continuation pair we instead suggest to *approximate* the solution $\boldsymbol{w} \approx \widehat{\boldsymbol{w}}_{j+1}$. (The solution to (4.9) is labeled $\boldsymbol{w}$, and not $\boldsymbol{w}_{j+1}$, since $\boldsymbol{w}_{j+1}$ is reserved for $(\nu_j A - \mu_j I)\boldsymbol{w}_{j+1} = (\rho_j A - \eta_j I)V_j\boldsymbol{t}_j$.) To make the whole process computationally feasible, obtaining this approximation should be inexpensive; see Remark 4.7. The pencil $(\underline{\widehat{H}_j}, \underline{\widehat{K}_j})$ is then constructed as usual in the rational Arnoldi algorithm, as if $\widehat{\boldsymbol{w}}_{j+1}$ was the true solution. As a result, we obtain the Hessenberg matrices

$$\underline{\widehat{K}_j} = \begin{bmatrix} \underline{K_{j-1}} & \widehat{\boldsymbol{k}}_j \\ \boldsymbol{0}^T & \widehat{k}_{j+1,j} \end{bmatrix} \quad \text{and} \quad \underline{\widehat{H}_j} = \begin{bmatrix} \underline{H_{j-1}} & \widehat{\boldsymbol{h}}_j \\ \boldsymbol{0}^T & \widehat{h}_{j+1,j} \end{bmatrix}, \tag{4.10}$$

where

$$\underline{\widehat{\boldsymbol{k}}_j} = \nu_j\underline{\widehat{\boldsymbol{c}}_j} - \widehat{\rho}_j\underline{\widehat{\boldsymbol{t}}_j},\ \underline{\widehat{\boldsymbol{h}}_j} = \mu_j\underline{\widehat{\boldsymbol{c}}_j} - \widehat{\eta}_j\underline{\widehat{\boldsymbol{t}}_j},\ \widehat{\boldsymbol{c}}_j = V_j^*\widehat{\boldsymbol{w}}_{j+1},\ \text{and}\ \widehat{c}_{j+1} = \|\widehat{\boldsymbol{w}}_{j+1} - V_j\widehat{\boldsymbol{c}}_j\|_2. \tag{4.11}$$

Assume that $(\widehat{\eta}/\widehat{\rho}, \widehat{\boldsymbol{x}})$ is an eigenpair of $(\widehat{H}_j, \widehat{K}_j)$ such that

$$\widehat{\rho}\widehat{H}_j\widehat{\boldsymbol{x}} - \widehat{\eta}\widehat{K}_j\widehat{\boldsymbol{x}} = \mathbf{0} \quad \text{and} \quad \widehat{\gamma}_j := \widehat{x}_j(\widehat{\rho}\widehat{h}_{j+1,j} - \widehat{\eta}\widehat{k}_{j+1,j}) \neq 0. \tag{4.12}$$

Then a near-optimal continuation pair is given by

$$\eta_j/\rho_j \equiv \widehat{\eta}/\widehat{\rho} \quad \text{and} \quad \boldsymbol{t}_j = \widehat{\gamma}_j^{-1}\big(\nu_j\widehat{H}_j - \mu_j\widehat{K}_j\big)\widehat{\boldsymbol{x}}. \tag{4.13}$$

Our goal in the next section is to evaluate the quality of these near-optimal continuation pairs within the rational Arnoldi algorithm. In particular, we provide an upper bound on the condition number of the basis being orthonormalized, based on the error $\|\boldsymbol{v}_{j+1} - \widehat{\boldsymbol{v}}_{j+1}\|_2$.

4.2.2. Inexact RADs. We proceed by introducing the residual

$$\widehat{\boldsymbol{s}}_{j+1} = (\nu_j A - \mu_j I)\widehat{\boldsymbol{w}}_{j+1} - (\widehat{\rho}_j A - \widehat{\eta}_j I)V_j\widehat{\boldsymbol{t}}_j. \tag{4.14}$$

By (4.7), (4.10), and (4.14) we have an *inexact rational Arnoldi decomposition* (IRAD)

$$A\widehat{V}_{j+1}\underline{\widehat{K}_j} = \widehat{V}_{j+1}\underline{\widehat{H}_j} + \widehat{\boldsymbol{s}}_{j+1}\boldsymbol{e}_j^T, \tag{4.15}$$

where $\widehat{V}_{j+1} = [V_j \quad \widehat{\boldsymbol{v}}_{j+1}]$ is orthonormal and

$$\widehat{\boldsymbol{w}}_{j+1} = V_j\widehat{\boldsymbol{c}}_j + \widehat{c}_{j+1}\widehat{\boldsymbol{v}}_{j+1} \quad \text{with} \quad \widehat{c}_{j+1} \neq 0. \tag{4.16}$$

Multiplying (4.15) by ν_j and then subtracting $\mu_j\widehat{V}_{j+1}\underline{\widehat{K}_j}$ from both sides provides

$$(\nu_j A - \mu_j I)\breve{V}_{j+1}\underline{\widehat{K}_j} = \breve{V}_{j+1}\big(\nu_j\underline{\widehat{H}_j} - \mu_j\underline{\widehat{K}_j}\big), \tag{4.17}$$

where

$$\begin{aligned} \breve{V}_{j+1} &= \Big[V_j \quad \widehat{\boldsymbol{v}}_{j+1} + \widehat{\boldsymbol{f}}_{j+1}\Big], \quad \text{and} \\ \widehat{\boldsymbol{f}}_{j+1} &= -\widehat{k}_{j+1,j}^{-1}\nu_j(\nu_j A - \mu_j I)^{-1}\widehat{\boldsymbol{s}}_{j+1} = -\widehat{c}_{j+1}^{-1}(\nu_j A - \mu_j I)^{-1}\widehat{\boldsymbol{s}}_{j+1}. \end{aligned} \tag{4.18}$$

Eq. (4.17) holds since the last row of $\nu_j\underline{\widehat{H}_j} - \mu_j\underline{\widehat{K}_j}$ is zero. We can also "add back" $\mu_j\breve{V}_{j+1}\underline{\widehat{K}_j}$ to both sides of (4.17), and rescale by ν_j^{-1} to get

$$A\breve{V}_{j+1}\underline{\widehat{K}_j} = \breve{V}_{j+1}\underline{\widehat{H}_j}. \tag{4.19}$$

Finally, under the assumption that $\breve{V}_{j+1}$ is of full rank, (4.19) is a non-orthonormal RAD, equivalent to the IRAD (4.15). Theorem 2.12 applied to (4.19) asserts that the eigenvalues of $\big(\widehat{H}_j, \widehat{K}_j\big)$ are the roots of the rational function corresponding to the vector $\widehat{\boldsymbol{v}}_{j+1} + \widehat{\boldsymbol{f}}_{j+1}$. This discussion is summarised in the following theorem.

Theorem 4.3. *Let the orthonormal RAD (4.7) and the auxiliary continuation pair $(\widehat{\eta}_j/\widehat{\rho}_j, \widehat{\boldsymbol{t}}_j)$ be given. Denote by $\widehat{\boldsymbol{w}}_{j+1} \notin \mathcal{R}(V_j)$ an approximate solution to (4.9). If (4.10)–(4.12), (4.16) and (4.18) hold, and (4.19) is an RAD, then choosing the continuation pair (4.13) in line 3 of Algorithm 2.2 provides*

$$\boldsymbol{w}_{j+1} = \widehat{\boldsymbol{v}}_{j+1} + \widehat{\boldsymbol{f}}_{j+1} \tag{4.20}$$

in line 4 of Algorithm 2.2.

The vector $\boldsymbol{w}_{j+1}$ is not necessarily orthogonal to $\mathcal{R}(V_j)$, but if $\|\widehat{\boldsymbol{f}}_{j+1}\|_2$ is "small enough" it almost is, since the vector $\widehat{\boldsymbol{v}}_{j+1}$ is orthogonal to $\mathcal{R}(V_j)$. We make this more precise in the following corollary.

Corollary 4.4. *Let the assumptions of Theorem 4.3 hold. If $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 = 0$, then $\angle(\boldsymbol{w}_{j+1}, V_j) = \frac{\pi}{2}$. If $0 < \|\widehat{\boldsymbol{f}}_{j+1}\|_2 < 1$, then*

$$\angle(\boldsymbol{w}_{j+1}, V_j) \geq \arctan \frac{1 - \|\widehat{\boldsymbol{f}}_{j+1}\|_2}{\|\widehat{\boldsymbol{f}}_{j+1}\|_2}. \tag{4.21}$$

Proof. By Theorem 4.3 we have $\boldsymbol{w}_{j+1} = \widehat{\boldsymbol{v}}_{j+1} + \widehat{\boldsymbol{f}}_{j+1}$, with $V_j V_j^* \widehat{\boldsymbol{v}}_{j+1} = \boldsymbol{0}$. If $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 = 0$, then $\boldsymbol{w}_{j+1} = \widehat{\boldsymbol{v}}_{j+1}$ is orthogonal to $\mathcal{R}(V_j)$. If $0 < \|\widehat{\boldsymbol{f}}_{j+1}\|_2 < 1$, then

$$\angle(\boldsymbol{w}_{j+1}, V_j) = \arctan \frac{\|\boldsymbol{w}_{j+1} - V_j V_j^* \boldsymbol{w}_{j+1}\|_2}{\|V_j V_j^* \boldsymbol{w}_{j+1}\|_2} = \arctan \frac{\|\widehat{\boldsymbol{v}}_{j+1} + \widehat{\boldsymbol{f}}_{j+1} - V_j V_j^* \widehat{\boldsymbol{f}}_{j+1}\|_2}{\|V_j V_j^* \widehat{\boldsymbol{f}}_{j+1}\|_2}.$$

Eq. (4.21) now follows from the reverse triangle inequality and the monotonicity of arctan, using the relation $\|\widehat{\boldsymbol{f}}_{j+1} - V_j V_j^* \widehat{\boldsymbol{f}}_{j+1}\|_2 = \|(I - V_j V_j^*)\widehat{\boldsymbol{f}}_{j+1}\|_2 \leq \|\widehat{\boldsymbol{f}}_{j+1}\|_2$. ∎

Note that Corollary 4.4 can be formulated even if $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 \geq 1$, but in this case would provide no useful information. Before continuing with the analysis of our near-optimal continuation strategy, let us remark on the choice of $(\widehat{\eta}_j/\widehat{\rho}_j, \widehat{\boldsymbol{t}}_j)$.

Remark 4.5 (auxiliary continuation pairs). The authors in [73] consider the rational Arnoldi algorithm with inexact solves, and suggest to use continuation pairs $(\eta_j/\rho_j, \boldsymbol{t}_j)$ such that $(\eta_j/\rho_j, V_j \boldsymbol{t}_j)$ is an approximate eigenpair of A close to convergence. As inexact solves are used within our framework to get a near-optimal continuation pair, this observation also applies to the auxiliary continuation pair $(\widehat{\eta}_j/\widehat{\rho}_j, \widehat{\boldsymbol{t}}_j)$.

4.2.3. Condition number of the Arnoldi basis. As $\breve{V}_{j+1} = \widehat{V}_{j+1} + \widehat{\boldsymbol{f}}_{j+1} \boldsymbol{e}_{j+1}^T$, with $\widehat{V}_{j+1}^* \widehat{V}_{j+1} = I_{j+1}$, from [42, Corollary 2.4.4] we obtain the following bounds

$$\sigma_{\max}(\breve{V}_{j+1}) \leq 1 + \|\widehat{\boldsymbol{f}}_{j+1}\|_2 \quad \text{and} \quad \sigma_{\min}(\breve{V}_{j+1}) \geq 1 - \|\widehat{\boldsymbol{f}}_{j+1}\|_2, \tag{4.22}$$

for the largest singular value $\sigma_{\max}(\breve{V}_{j+1})$ and for the smallest singular value $\sigma_{\min}(\breve{V}_{j+1})$ of $\breve{V}_{j+1}$. Composing these bounds for all indices j we are able to provide an upper bound on the condition number $\kappa(W_{m+1})$ of the basis

$$W_{j+1} := \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_{j+1} \end{bmatrix} \quad \text{with} \quad \boldsymbol{w}_1 = \boldsymbol{b}, \quad j = 1, 2, \dots, m,$$

which is constructed iteratively by Algorithm 2.2. The Gram–Schmidt orthogonalization process is mathematically equivalent to computing the thin QR factorisation

$$W_{j+1} = V_{j+1} \begin{bmatrix} \|\boldsymbol{b}\|_2 \boldsymbol{e}_1 & \underline{K_j} \operatorname{diag}(\eta_\ell)_{\ell=1}^{j} - \underline{H_j} \operatorname{diag}(\rho_\ell)_{\ell=1}^{j} \end{bmatrix} =: V_{j+1} R_{j+1}, \tag{4.23}$$

where the first equality follows from (2.3). As already discussed in the introduction of the chapter, numerical instability may occur if $\kappa(W_{j+1})$ is too large.

Theorem 4.6. *Let the assumptions of Theorem 4.3 hold for $j = 1, 2, \dots, m$, and let the orthonormal RAD (2.6) be constructed with Algorithm 2.2 using near-optimal continuation pairs $(\eta_j/\rho_j, \boldsymbol{t}_j)$ given by (4.12)–(4.13). Let $R_1 = I_1$, and R_{j+1} be as in (4.23). Assume that the scaled error $\widehat{\boldsymbol{f}}_{j+1}$ at iteration j satisfies $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 < 1$. Then for all $j = 1, 2, \dots, m$ we have*

$$\begin{aligned} \sigma_{\max}\left(W_{j+1}\right) &\le \prod_{i=1}^{j} \left(1 + \|\widehat{\boldsymbol{f}}_{i+1}\|_2\right) =: \sigma_{j+1}^{u}, \quad \textit{and} \\ \sigma_{\min}\left(W_{j+1}\right) &\ge \prod_{i=1}^{j} \left(1 - \|\widehat{\boldsymbol{f}}_{i+1}\|_2\right) =: \sigma_{j+1}^{l}. \end{aligned} \tag{4.24}$$

In particular, $\kappa(W_{m+1}) \le \sigma_{m+1}^{u}/\sigma_{m+1}^{l}$.

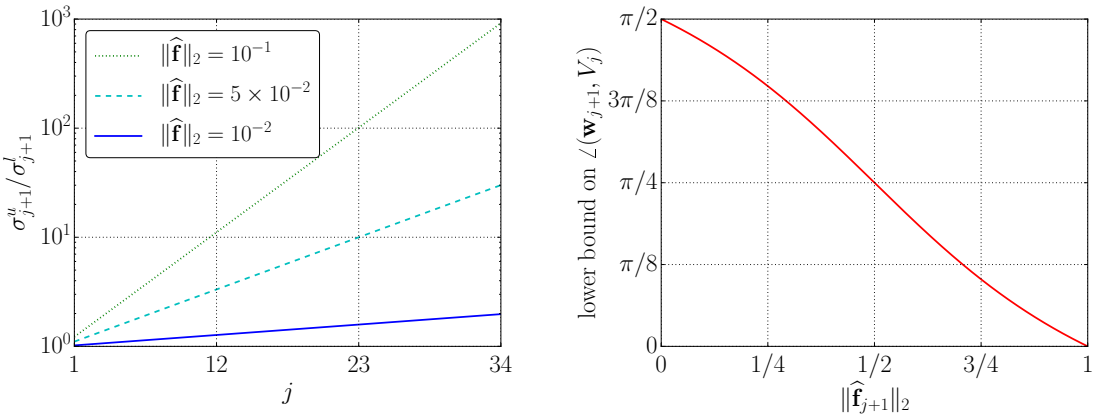
Proof. For any $j = 1, 2, \dots, m$ we have

$$W_{j+1} = V_{j+1} R_{j+1} = \begin{bmatrix} V_j R_j & \widehat{\boldsymbol{v}}_{j+1} + \widehat{\boldsymbol{f}}_{j+1} \end{bmatrix} = \begin{bmatrix} V_j & \widehat{\boldsymbol{v}}_{j+1} + \widehat{\boldsymbol{f}}_{j+1} \end{bmatrix} \begin{bmatrix} \underline{R_j} & \boldsymbol{e}_{j+1} \end{bmatrix}, \tag{4.25}$$

with $V_j^* \widehat{\boldsymbol{v}}_{j+1} = \boldsymbol{0}$, and $\|\widehat{\boldsymbol{v}}_{j+1}\|_2 = 1$. The proof goes by induction on j. For $j = 1$ the statement follows from (4.25), (4.22), and the fact that $[\underline{R_1} \quad \boldsymbol{e}_2] = I_2$.

Let us assume that (4.24) holds for $j = 1, 2, \dots, \ell < m$. For the induction step we consider the case $j = \ell + 1$, and use the fact that, for any two conformable matrices X and Y of full rank, there holds $\sigma_{\max}(XY) \le \sigma_{\max}(X)\sigma_{\max}(Y)$ and $\sigma_{\min}(XY) \ge \sigma_{\min}(X)\sigma_{\min}(Y)$. Hence, (4.24) for $j = \ell + 1$ follows from (4.25), from the bound (4.22) for $[V_{\ell+1} \quad \widehat{\boldsymbol{v}}_{\ell+2} + \widehat{\boldsymbol{f}}_{\ell+2}]$, from the fact that the singular values of $R_{\ell+1}$ coincide with those of $W_{\ell+1}$, and from the observation

$$\sigma_{\max}\left(\begin{bmatrix} \underline{R_{\ell+1}} & \boldsymbol{e}_{\ell+2} \end{bmatrix}\right) \le \sigma_{\ell+1}^{u}, \quad \text{and} \quad \sigma_{\min}\left(\begin{bmatrix} \underline{R_{\ell+1}} & \boldsymbol{e}_{\ell+2} \end{bmatrix}\right) \ge \sigma_{\ell+1}^{l}.$$



(a) Upper bounds on $\kappa(W_{m+1})$.

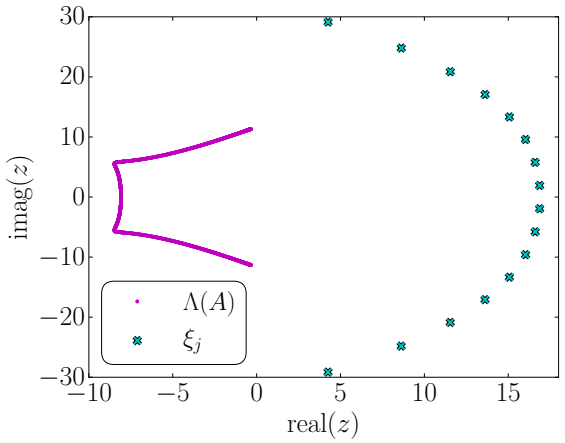
(b) Lower bound on the angle.

Figure 4.1: Evaluating the quality of the near-optimal continuation strategy. Left: The function $j \mapsto \left(\frac{1+\|\widehat{\boldsymbol{f}}\|_2}{1-\|\widehat{\boldsymbol{f}}\|_2}\right)^j$ for three different values of $\|\widehat{\boldsymbol{f}}\|_2$. Theorem 4.6 asserts that these are upper bounds on $\kappa(W_{m+1})$, provided that for all js there holds $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 \leq \|\widehat{\boldsymbol{f}}\|_2$. Right: The function $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 \mapsto \arctan \frac{1-\|\widehat{\boldsymbol{f}}_{j+1}\|_2}{\|\widehat{\boldsymbol{f}}_{j+1}\|_2}$, which provides a lower bound on $\angle(\boldsymbol{w}_{j+1}, V_j)$.

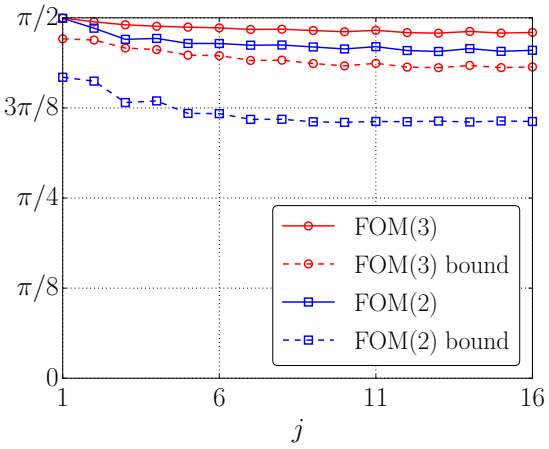
This last relation holds since the singular values of $[\underline{R_{\ell+1}} \quad \boldsymbol{e}_{\ell+2}]$ are those of $\underline{R_{\ell+1}}$ with the addition of the singular value $1 \in [\sigma^l_{\ell+1}, \sigma^u_{\ell+1}]$. □

We now briefly comment on the results established in Theorem 4.6, the assumptions of which we assume to hold. If, for instance, $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 = 0.5$, then $\sigma^u_{j+1}/\sigma^l_{j+1} \leq 3\sigma^u_j/\sigma^l_j$. That is, the bound $\sigma^u_{j+1}/\sigma^l_{j+1}$ on the condition number $\kappa(W_{j+1})$ grows by at most a factor of 3 compared to σ^u_j/σ^l_j, which does not necessarily imply $\kappa(W_{j+1}) \leq 3\kappa(W_j)$. It would imply that, if $\sigma_{\min}(W_j) \leq 1 \leq \sigma_{\max}(W_j)$ holds true (this observation is clear from the proof of Theorem 4.6). In Figure 4.1(a) we illustrate the upper bounds given by Theorem 4.6 for some particular values of $\|\widehat{\boldsymbol{f}}_{j+1}\|_2$. Figure 4.1(b) visualizes the lower bound, provided by Corollary 4.4, on the angle $\angle(\boldsymbol{w}_{j+1}, V_j)$. For example, for a rough approximation $\widehat{\boldsymbol{w}}_{j+1}$ that gives $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 = 0.5$, we have $\angle(\boldsymbol{w}_{j+1}, V_j) \geq \frac{\pi}{4}$.

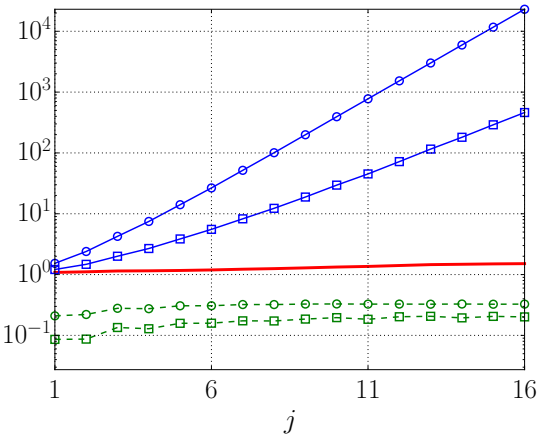
Remark 4.7. If the poles of the rational Krylov space are fairly well separated from the spectral region of A, a good approximate solution to (4.9) may be obtained with a few iterations of a cheap polynomial Krylov method, like unpreconditioned FOM or GMRES, or with a cycle of multigrid [94]. Computational examples of this situation are given in Section 4.2.4 and Section 4.4.1. When direct solvers are used within the rational Arnoldi algorithm, it may even be worth solving (4.9) to full accuracy, as the most costly computation is the analysis and factorization of each shifted linear system,



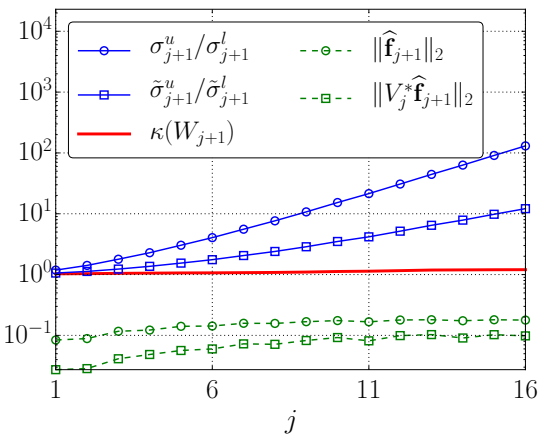
(a) Spectrum of A and location of poles.



(b) Angles $\angle(\boldsymbol{w}_{j+1}, V_j)$ and lower bounds.



(c) Approximating $\widehat{\boldsymbol{w}}_{j+1}$ with FOM(2).



(d) Approximating $\widehat{\boldsymbol{w}}_{j+1}$ with FOM(3).

Figure 4.2: Near-optimal continuation strategy on a nonnormal matrix A; see Section 4.2.4.

which needs to be done at most once per pole. An example of this situation is given in Section 4.4.2.

4.2.4. Numerical illustration. In Figure 4.2 we illustrate the effectiveness of our near-optimal continuation framework. The matrix A is of size $N = 1000$, and it is generated in MATLAB with `A=-5*gallery('grcar',N,3)`. This is a nonnormal matrix and its eigenvalues are shown in Figure 4.2(a), together with the $m = 16$ poles used in this example. The poles are obtained using the RKFIT algorithm which we discuss in Chapter 6, and they are optimized for approximating $\exp(A)\boldsymbol{b}$, where the starting vector $\boldsymbol{b}$ has all its entries equal to 1. (A similar example is considered in Section 6.2.3.) Two experiments are performed with this data, and they differ in the way the approximants $\widehat{\boldsymbol{w}}_{j+1}$, used to obtain the near-optimal continuation pairs, are

computed. Since the poles are far away from $\Lambda(A)$, we expect a few iterations of FOM to provide good approximants $\widehat{\boldsymbol{w}}_{j+1}$. To obtain each $\widehat{\boldsymbol{w}}_{j+1}$ we hence use a fixed number k of FOM iterations; this is referred to as FOM(k). In Figure 4.2(b) we plot the angles $\angle(\boldsymbol{w}_{j+1}, V_j)$ and the lower bound (4.21) at each iteration $j = 1, 2, \ldots, m$. Both FOM(2) and FOM(3) are giving satisfactory results, with FOM(3) performing slightly better.

Figures 4.2(c)–4.2(d) show the condition numbers $\kappa(W_{m+1})$ of the bases as well as the upper bounds from Theorem 4.6. Additionally, we provide a refined upper-bound on $\kappa(W_{m+1})$. The refined bound can be derived in the same manner as the one in Theorem 4.6, but using (4.26) below instead of (4.22). We remark that (4.26) imposes a slightly more stringent condition on $\widehat{\boldsymbol{f}}_{j+1}$. We start by introducing the projection $\widehat{\boldsymbol{e}}_{j+1} := \widehat{V}_{j+1}^* \widehat{\boldsymbol{f}}_{j+1}$ and noting that

$$\breve{V}_{j+1}^* \breve{V}_{j+1} = I_{j+1} + \|\widehat{\boldsymbol{f}}_{j+1}\|_2^2 \boldsymbol{e}_{j+1}\boldsymbol{e}_{j+1}^T + \boldsymbol{e}_{j+1}\widehat{\boldsymbol{e}}_{j+1}^* + \widehat{\boldsymbol{e}}_{j+1}\boldsymbol{e}_{j+1}^T =: I_{j+1} + E_{j+1}.$$

Directly from the definition of E_{j+1} we have $\|E_{j+1}\|_2 \leq 2\|\widehat{\boldsymbol{e}}_{j+1}\|_2 + \|\widehat{\boldsymbol{f}}_{j+1}\|_2^2$. Finally, under the assumption that $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 < \sqrt{2} - 1$, we deduce

$$\begin{aligned} \sigma_{\max}(\breve{V}_{j+1}) &\leq \sqrt{1 + 2\|\widehat{\boldsymbol{e}}_{j+1}\|_2 + \|\widehat{\boldsymbol{f}}_{j+1}\|_2^2} =: \tilde{\sigma}_{j+1}^u / \tilde{\sigma}_j^u, \quad \text{and} \\ \sigma_{\min}(\breve{V}_{j+1}) &\geq \sqrt{1 - 2\|\widehat{\boldsymbol{e}}_{j+1}\|_2} =: \tilde{\sigma}_{j+1}^l / \tilde{\sigma}_j^l, \end{aligned} \tag{4.26}$$

with $\tilde{\sigma}_{j+1}^u$ and $\tilde{\sigma}_{j+1}^l$ being defined recursively, with initial values $\tilde{\sigma}_0^u = \tilde{\sigma}_0^l = 1$.

In both Figures 4.2(c)–4.2(d) we include the norms $\|\widehat{\boldsymbol{f}}_{j+1}\|_2$ and $\|\widehat{\boldsymbol{e}}_{j+1}\|_2$ for reference. With FOM(2), we have $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 \approx 0.30$ on average (geometric mean), and the overall upper bound on $\kappa(W_{m+1}) \approx 1.51$ is $\sigma_{m+1}^u / \sigma_{m+1}^l \approx 2.30 \times 10^4$. The refined upper bound that makes use of the projections $\widehat{\boldsymbol{e}}_{j+1}$ gives $\tilde{\sigma}_{m+1}^u / \tilde{\sigma}_{m+1}^l \approx 461$, which is about two orders of magnitude sharper. Using FOM(3) produces on average $\|\widehat{\boldsymbol{f}}_{j+1}\|_2 \approx 0.15$. The condition number of the basis being orthogonalised is $\kappa(W_{m+1}) \approx 1.20$, while the upper bound provided by Theorem 4.6 is $\sigma_{m+1}^u / \sigma_{m+1}^l \approx 130$. The refined upper bound based on (4.26) yields $\tilde{\sigma}_{m+1}^u / \tilde{\sigma}_{m+1}^l \approx 12$. We observe that the bounds get sharper as the error $\|\widehat{\boldsymbol{f}}_{j+1}\|_2$ gets smaller, and also that the two bases W_{m+1}, computed using both the FOM(2) and FOM(3) near-optimal continuation strategy, becomes better conditioned as the approximations $\widehat{\boldsymbol{w}}_{j+1}$ get more accurate. In both examples, the nonorthogonal bases are in fact remarkably well-conditioned.

Note that computing or estimating $\|\widehat{\boldsymbol{f}}_{j+1}\|_2$ may be too costly in practice. However, the main message of Theorem 4.6 and this numerical illustration is that rather poor

approximations $\widehat{\boldsymbol{w}}_{j+1}$ are sufficient to limit the growth of $\kappa(W_{m+1})$ considerably. See also Figures 4.1–4.2.

Before passing on to the parallel rational Arnoldi algorithm we briefly discuss a possible alternative for obtaining near-optimal continuation pairs.

4.2.5. A different viewpoint for Hermitian matrices. The following observation suggests an alternative approach for Hermitian problems with real-valued poles.

Proposition 4.8. *Let* (2.6) *and* (4.2) *be orthonormal RADs with Hermitian A, and let real-valued scalars $\eta, \rho, \mu, \nu \in \mathbb{R}$ satisfy $\nu\eta \neq \mu\rho$. Assume that η/ρ is a (formal) root of $p_m(z) = \det(H_m - zK_m)$. The continuation pair $(\eta/\rho, \boldsymbol{t})$ is optimal for* (4.1) *if and only if the vector $\boldsymbol{t} \neq \boldsymbol{0}$ is orthogonal to $\mathcal{R}(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}})$.*

Proof. Let $(\eta/\rho, \boldsymbol{x} \neq \boldsymbol{0})$ be such that $(\rho H_m - \eta K_m)\boldsymbol{x} = \boldsymbol{0}$. Then and only then we have that $(\eta/\rho, \nu H_m\boldsymbol{x} - \mu K_m\boldsymbol{x})$ is an optimal continuation pair; cf. Proposition 4.2. It follows from Lemma 2.6 that $\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}}$ is of full column rank, and therefore, the orthogonal complement of $\mathcal{R}(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}})$ is a one-dimensional space. Consequently, to prove the statement it is enough to show that $(\nu H_m - \mu K_m)\boldsymbol{x} \perp \mathcal{R}(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}})$.

Multiplying (2.6) by $\underline{K_m^*}V_{m+1}^*$ from the left we obtain $\underline{K_m^*}V_{m+1}^*AV_{m+1}\underline{K_m} = \underline{K_m^*}\underline{H_m}$, which is Hermitian, and hence $\underline{K_m^*}\underline{H_m} = \underline{H_m^*}\underline{K_m}$. In particular,

$$K_m^*\underline{H_{m-1}} = H_m^*\underline{K_{m-1}}. \tag{4.27}$$

Further, conjugate-transposing $(\rho H_m - \eta K_m)\boldsymbol{x} = \boldsymbol{0}$, and then multiplying by $\underline{H_{m-1}}$ from the right gives (4.28) below. In an analogous manner we obtain (4.29):

$$\rho\boldsymbol{x}^*H_m^*\underline{H_{m-1}} = \eta\boldsymbol{x}^*K_m^*\underline{H_{m-1}}; \tag{4.28}$$

$$\eta\boldsymbol{x}^*K_m^*\underline{K_{m-1}} = \rho\boldsymbol{x}^*H_m^*\underline{K_{m-1}}. \tag{4.29}$$

Exploiting (4.28) and (4.29), we obtain the relation

$$\boldsymbol{x}^*(\nu H_m^* - \mu K_m^*)(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}}) = (\nu\eta - \mu\rho)\boldsymbol{x}^*(K_m^*\underline{H_{m-1}} - H_m^*\underline{K_{m-1}}) = \boldsymbol{0}^*,$$

where the last equality follows from (4.27). ∎

A near-optimal continuation strategy inspired by Proposition 4.8 would require us to approximate one root of $p_m(z)$, and then use it to compute the corresponding continuation vector. If $\widehat{\eta}/\widehat{\rho}$ is such continuation root, then as a continuation vector one

could use any vector spanning $\mathcal{R}(\widehat{\rho}\underline{H_{m-1}} - \widehat{\eta}\underline{K_{m-1}})^{\perp}$. We shall not elaborate on this further. Instead, we briefly relate it to Ruhe's considerations in [90, Sec. 3.2].

Throughout [90], infinity is consistently used as continuation root, but the choice of continuation vector gained more attention. Specifically, the choice $\boldsymbol{t} \neq \mathbf{0}$ such that $\boldsymbol{t} \perp \mathcal{R}(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}})$ is suggested. For the parameter η/ρ the possibilities ξ_{m-1}, ∞ and ξ_m are discussed. Further, it is argued that if η/ρ is a generalised eigenvalue of (H_{m-1}, K_{m-1}), i.e., a (formal) root of p_{m-1}, then $\boldsymbol{e}_m^T\boldsymbol{t} = 0$. As discussed in Section 4.1, the choice ξ_m is interesting in that it provides an admissible continuation vector. However, in light of Proposition 4.8 and the (easy to show) fact that $\boldsymbol{t} \perp \mathcal{R}(\rho\underline{H_{m-1}} - \eta\underline{K_{m-1}})$ and $\boldsymbol{t} \perp \mathcal{R}(\nu\underline{H_{m-1}} - \mu\underline{K_{m-1}})$ for $\nu\eta \neq \mu\rho$ implies $\boldsymbol{t} = \mathbf{0}$, we know that it will not provide an optimal one, at least for Hermitian A and the poles being real-valued, since p_m cannot have as root the mth pole ξ_m; cf. Theorem 2.12.

Remark 4.9. Optimal continuation pairs for quasi-RADs may be obtained similarly as for RADs; the vector $\boldsymbol{w}_{j+1}$ in (2.29) may be computed using complex arithmetic and orthogonalized against the real-valued basis V_j to provide an optimal continuation pair which may be complex-valued. With this continuation pair one obtains the complex-valued vector $\boldsymbol{v}_{j+1}$ orthogonal to $\mathcal{R}(V_j)$. In particular, both $\Re(\boldsymbol{v}_{j+1})$ and $\Im(\boldsymbol{v}_{j+1})$ are orthogonal to $\mathcal{R}(V_j)$. It remains to note that instead of adding $\begin{bmatrix}\Re(\boldsymbol{v}_{j+1}) & \Im(\boldsymbol{v}_{j+1})\end{bmatrix}$ to the quasi-RAD (2.27) one can add any orthonormal basis of the corresponding two-dimensional space in order to expand (2.27).

4.3 Parallel rational Arnoldi algorithm

In this section we introduce a new parallel variant of the rational Arnoldi algorithm based on near-optimal continuation pairs. The parallelism we consider comes from generating more than one of the basis vectors concurrently. Another possibility is to parallelise the involved linear algebra operations, thought that might scale less favorably as the number of parallel processes increases. Combining both parallelisation approaches is also viable, and our implementation supports this. Further comments about the implementation and numerical examples are given in Section 4.4.

4.3.1. High-level description of a parallel rational Arnoldi algorithm. The aim of the parallel rational Arnoldi algorithm, outlined in Algorithm 4.5 and in the discussion below, is to construct an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ using $p > 1$ parallel processes. The basis is constructed iteratively, but unlike the sequential version, at each iteration $p > 1$ vectors are computed simultaneously, one per parallel process (with a possible exception of the last iteration if m is not a multiple of p). The poles assigned to distinct parallel processes have to be mutually distinct, as otherwise we would not obtain p linearly independent vectors to expand the rational Krylov space.

We assume a copy of the matrix A, the starting vector $\boldsymbol{b}$, and the poles $\{\xi_j\}_{j=1}^m$ to be available to each parallel process. After the orthogonal basis vectors have been constructed, they are broadcasted to the other parallel processes for use in the following parallel iterations. This means that a copy of the basis V_{j+1} is available to every parallel process. We now go through Algorithm 4.5 line by line.

In line 2 of Algorithm 4.5 the starting vector $\boldsymbol{b}$ is normalised (on every parallel process), providing the first basis vector $\boldsymbol{v}_1 \equiv \boldsymbol{v}_1^{[\ell]}$. We use the superscript notation $(\cdot)^{[\ell]}$ to denote that the quantity $(\cdot)$ belongs to the parallel process ℓ. If a quantity is sent to another parallel process $\hat{\ell} \neq \ell$, a copy $(\cdot)^{[\hat{\ell}]} = (\cdot)^{[\ell]}$ becomes available to $\hat{\ell}$. The main part of the algorithm is the j-loop spanning lines 3–27, where the remaining m vectors of the orthonormal basis are generated by p parallel processes, which requires $\lceil \frac{m}{p} \rceil$ iterations. The variable s represents the order of the RAD $AV_{s+1}\underline{K_s} = V_{s+1}\underline{H_s}$ constructed so far, and every parallel process has its own copy of it. The variable $\underline{p} \leq p$ equals p for all iterations j, except perhaps the last one where $\underline{p} = m - s$ represents the number of remaining basis vectors to be constructed. Parallel processes with labels greater than $\underline{p}$ do not perform the remaining part of the last iteration of the j-loop.

The selection of continuation pairs in line 9 is discussed in subsection 4.3.2. We shall only stress that the continuation pairs are of order $s + 1$ for all ℓ, and that we assume the choice to be such that unlucky breakdown is avoided. Once the continuation pairs have been computed, a new direction $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ is computed the same way as in the sequential rational Arnoldi algorithm; cf. line 10. The orthogonalization part, however, is more involved and consists of two parts.

The first part of the orthogonalization process corresponds to lines 11–12, where the newly computed vector $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ is orthogonalized against $\mathcal{R}(V_{s+1}^{[\ell]})$. The second part of

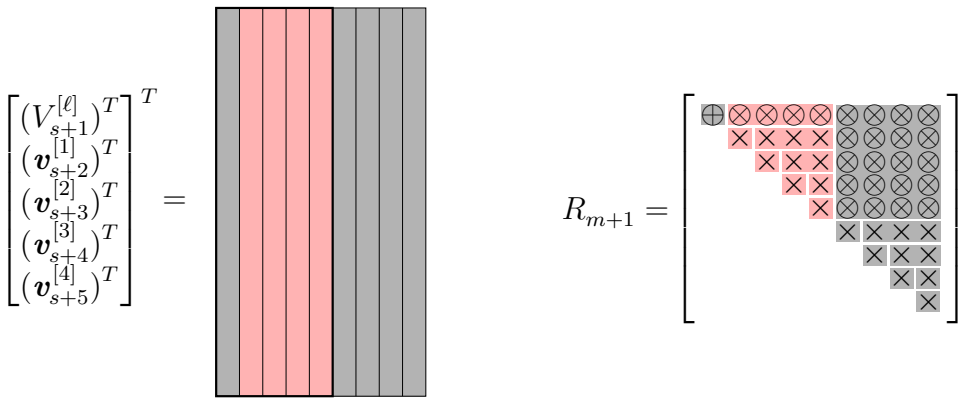
Algorithm 4.5 Parallel rational Arnoldi for distributed memory architectures.

Input: $A \in \mathbb{C}^{N,N}$, $\boldsymbol{b} \in \mathbb{C}^N$, poles $\{\mu_j/\nu_j\}_{j=1}^m \subset \overline{\mathbb{C}} \setminus \Lambda(A)$, with $m < M$, and such that the partitions $\{\mu_{kp+\ell}/\nu_{kp+\ell}\}_{\ell=1}^p$, for $k = 0, 1, \ldots, \lfloor \frac{m}{p} \rfloor - 1$, and $\{\mu_j/\nu_j\}_{j=p\lfloor \frac{m}{p} \rfloor+1}^m$, where p is the number of parallel processes, consist of pairwise distinct poles.

Output: The RAD $AV_{m+1}^{[1]}\underline{K_m^{[1]}} = V_{m+1}^{[1]}\underline{H_m^{[1]}}$.

1. Let the p parallel processes be labelled by $\ell = 1, 2, \ldots, p$.
2. Set $\boldsymbol{v}_1^{[\ell]} := \boldsymbol{b}/\|\boldsymbol{b}\|_2$.
3. **for** $j = 1, \ldots, \lceil \frac{m}{p} \rceil$ **do**
4. Set $s := (j-1)p$. ▷ The RAD $AV_{s+1}^{[\ell]}\underline{K_s^{[\ell]}} = V_{s+1}^{[\ell]}\underline{H_s^{[\ell]}}$ holds.
5. Let $\underline{p} := \min\{p, m-s\}$.
6. **if** $\ell > \underline{p}$ **then**
7. Mark processor ℓ as inactive. ▷ Applies to the case $j = \lceil \frac{m}{p} \rceil$ if $p \nmid m$.
8. **end if**
9. Choose continuation pair $(\eta_{s+\ell}^{[\ell]}/\rho_{s+\ell}^{[\ell]}, \boldsymbol{t}_{s+\ell}^{[\ell]}) \in \overline{\mathbb{C}} \times \mathbb{C}^{s+1}$.
10. Compute $\boldsymbol{w}_{s+\ell+1}^{[\ell]} := (\nu_{s+\ell}A - \mu_{s+\ell}I)^{-1}(\rho_{s+\ell}^{[\ell]}A - \eta_{s+\ell}^{[\ell]}I)V_{s+1}^{[\ell]}\boldsymbol{t}_{s+\ell}^{[\ell]}$.
11. Project $\boldsymbol{c}_{s+1}^{[\ell]} := (V_{s+1}^{[\ell]})^*\boldsymbol{w}_{s+\ell+1}^{[\ell]}$.
12. Update $\boldsymbol{w}_{s+\ell+1}^{[\ell]} := \boldsymbol{w}_{s+\ell+1}^{[\ell]} - V_{s+1}^{[\ell]}\boldsymbol{c}_{s+1}^{[\ell]}$.
13. **for** $k = 1, \ldots, \underline{p}$ **do**
14. **if** $\ell = k$ **then**
15. Compute $c_{s+\ell+1}^{[\ell]} := \|\boldsymbol{w}_{s+\ell+1}^{[\ell]}\|_2$, and set $\boldsymbol{v}_{s+\ell+1}^{[\ell]} := \boldsymbol{w}_{s+\ell+1}^{[\ell]}/c_{s+\ell+1}^{[\ell]}$.
16. **end if**
17. Broadcast $\boldsymbol{v}_{s+k+1}^{[\ell]}$ from parallel process k.
18. **if** $\ell = k$ **then**
19. Let $\boldsymbol{c}_{s+\ell+1}^{[\ell]} := \left[\, (\boldsymbol{c}_{s+1}^{[\ell]})^T \; c_{s+2}^{[\ell]} \; \cdots \; c_{s+\ell+1}^{[\ell]} \,\right]^T$, and $\underline{\boldsymbol{t}_{s+\ell}^{[\ell]}} := \left[\, (\boldsymbol{t}_{s+\ell}^{[\ell]})^T \; \boldsymbol{0}^T \,\right]^T \in \mathbb{C}^{s+\ell+1}$.
20. Form $\underline{\boldsymbol{k}_{s+\ell}^{[\ell]}} := \nu_{s+\ell}\boldsymbol{c}_{s+\ell+1}^{[\ell]} - \rho_{s+\ell}^{[\ell]}\underline{\boldsymbol{t}_{s+\ell}^{[\ell]}}$, and $\underline{\boldsymbol{h}_{s+\ell}^{[\ell]}} := \mu_{s+\ell}\boldsymbol{c}_{s+\ell+1}^{[\ell]} - \eta_{s+\ell}^{[\ell]}\underline{\boldsymbol{t}_{s+\ell}^{[\ell]}}$.
21. **else if** $\ell > k$ **then**
22. Project $c_{s+k+1}^{[\ell]} := (\boldsymbol{v}_{s+k+1}^{[\ell]})^*\boldsymbol{w}_{s+\ell+1}^{[\ell]}$.
23. Update $\boldsymbol{w}_{s+\ell+1}^{[\ell]} := \boldsymbol{w}_{s+\ell+1}^{[\ell]} - c_{s+k+1}^{[\ell]}\boldsymbol{v}_{s+k+1}^{[\ell]}$.
24. **end if**
25. Broadcast $\underline{\boldsymbol{k}_{s+k}^{[\ell]}}$ and $\underline{\boldsymbol{h}_{s+k}^{[\ell]}}$ from parallel process k.
26. **end for** (orthogonalization loop)
27. **end for** (main loop)

the orthogonalization corresponds to the loop in lines 13–26, and involves communication between the parallel processes. In this part the (partially) orthogonalized vectors $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ are gradually being orthonormalised against each other. As soon as $\boldsymbol{w}_{s+k+1}^{[k]}$ is normalised to $\boldsymbol{v}_{s+k+1}^{[k]}$ in line 15, it is broadcasted to the remaining active parallel processes in line 17. At this stage the parallel process k updates the RAD from order $s+k-1$ to order $s+k$ (lines 19–20), while the active parallel processes $\ell > k$ orthonormalize $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ against the just received $\boldsymbol{v}_{s+k+1}^{[\ell]}$; lines 22–23. See also Figure 4.3 for an example. The final part, line 25, is to broadcast the update for the reduced upper-Hessenberg pencil



(a) Constructing the basis V_{m+1}. (b) Gram–Schmidt coefficients.

Figure 4.3: Executing Algorithm 4.5 with $m = 8$ and $p = 4$, which requires 2 iterations of the main loop. In 4.3(a) we show which parallel process is responsible for constructing a particular column of V_{m+1}. The columns of R_{m+1} shown in 4.3(b) are partitioned in three sets. The first consists of the first column alone and represents line 2 (of Algorithm 4.5). The second consists of columns 2–5, and the third of columns 6–9. The partitioning of elements within these last two block-columns represents the two parts of the orthogonalization. The elements marked with $\otimes$ are computed in line 11, while those marked $\times$ are computed either in line 15 or 22, depending on whether they are on the diagonal of R_{m+1} or not, respectively. Elements contained in the same rectangle can be computed simultaneously, after (all) the elements within rectangles to the left and above it have been computed.

from parallel process k to the remaining active ones. The communication between the p parallel processes involves $\mathcal{O}(p^2)$ messages, which is not prohibitive in our case as p is typically moderate (not exceeding $p = 8$ in our experiments in section 4.4).

Alternative implementation options. Depending on possible memory constraints, one may consider distributing the basis, instead of having copies on every parallel process. In this case the p vectors $V_{s+1}^{[\ell]} \boldsymbol{t}_{s+\ell}^{[\ell]}$ could be formed jointly by all the parallel processes, and once all have been formed and distributed, the vectors $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ may be constructed independently. The Gram–Schmidt process can be adapted accordingly.

A shared memory implementation may follow the same guidelines of Algorithm 4.5, excluding the broadcast statements. Also, the second part of the orthogonalization may be performed jointly by assigning an (almost) equal amount of work to each thread. (The index notation adopted in Algorithm 4.5 guarantees that different threads do not overwrite "each others" data.)

Lastly, the version of the parallel rational Arnoldi algorithm presented, tacitly assumes that solving the various shifted linear system takes roughly the same time,

which is the case if one uses direct solvers, for example. If the time to solution of the shifted linear system varies substantially depending on the poles, then it may be advantageous not to construct p vectors at a time, but to consider an asynchronous approach.

4.3.2. Locally near-optimal continuation pairs. We now discuss the choice of continuation pairs for Algorithm 4.5. To this end we use the *continuation matrix* $T_m := [\boldsymbol{t}_1 \quad \boldsymbol{t}_2 \quad \dots \quad \boldsymbol{t}_m] \in \mathbb{C}^{m,m}$, which collects the continuation vectors (padded with zeros) of order $j = 1, 2, \dots, m$ as they are being used in the sequential rational Arnoldi algorithm. Consequently, T_m is an upper triangular matrix. For the parallel rational Arnoldi algorithm, we order the continuation vectors $\boldsymbol{t}_{s+\ell}^{[\ell]}$ increasingly by their indices $s+\ell$, obtaining again an upper triangular matrix. In the parallel case there are, however, further restrictions on the nonzero pattern of T_m, as can be observed in Figure 4.4. There we display three *canonical* choices for $\boldsymbol{t}_{s+\ell}^{[\ell]}$.

Perhaps the two most canonical choices for continuation vectors are

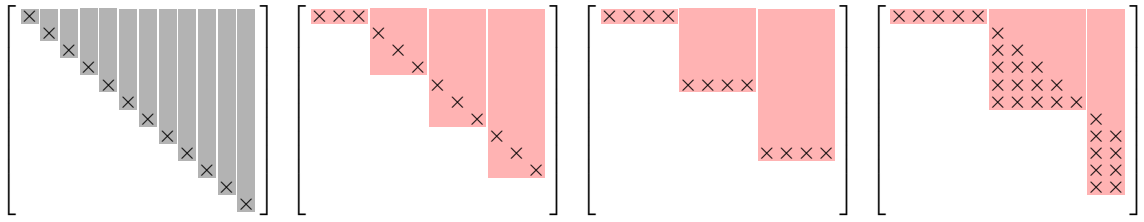
$$\boldsymbol{t}_{s+\ell}^{[\ell]} = \boldsymbol{e}_{\max\{1,s+1-p+\ell\}}, \quad \text{and} \tag{4.30}$$

$$\boldsymbol{t}_{s+\ell}^{[\ell]} = \boldsymbol{e}_{s+1}. \tag{4.31}$$

With continuation vectors given by (4.30), each parallel process ℓ applies the transformation $(\nu_{s+\ell}A - \mu_{s+\ell}I)^{-1}(\rho_{s+\ell}^{[\ell]}A - \eta_{s+\ell}^{[\ell]}I)$ to either the rescaled starting vector $\boldsymbol{v}_1^{[\ell]}$ (for $s = 0$) or to the vector $\boldsymbol{v}_{s+1-p+\ell}^{[\ell]}$ (for $s > 0$). On the other hand, with continuation vectors as in (4.31) the same vector $\boldsymbol{v}_{s+1}^{[\ell]}$ is used for all ℓ. These two choices are illustrated with the aid of the corresponding continuation matrices in Figures 4.4(b)–4.4(c) for the case $m = 12$ and two distinct choices for p. The choice (4.30) was used in [99] with infinity as the corresponding continuation root, while (4.31) has been introduced in [55, section 6.5]. Another possibility for continuation vectors is to use Ruhe's strategy [90] locally on each parallel process;

$$\boldsymbol{t}_{s+\ell}^{[\ell]} = Q_{s+1}^{[\ell]}\boldsymbol{e}_{s+1}, \quad \text{where} \quad \nu_{s+\ell}\underline{H_s^{[\ell]}} - \mu_{s+\ell}\underline{K_s^{[\ell]}} =: Q_{s+1}^{[\ell]}\underline{R_s^{[\ell]}} \tag{4.32}$$

is a full QR factorization of $\nu_{s+\ell}\underline{H_s^{[\ell]}} - \mu_{s+\ell}\underline{K_s^{[\ell]}}$, i.e., $Q_{s+1}^{[\ell]} \in \mathbb{C}^{s+1,s+1}$ is unitary and $\underline{R_s^{[\ell]}} \in \mathbb{C}^{s+1,s}$ is upper triangular with last row being $\mathbf{0}^T$. The corresponding continuation matrix T_m is shown in Figure 4.4(d) for the case when the poles on each parallel process



(a) Sequential ($p = 1$). (b) Parallel with $p = 3$. (c) Parallel with $p = 4$. (d) Parallel with $p = 5$.

Figure 4.4: Canonical continuation matrices T_{12} for the sequential, 4.4(a), and parallel, 4.4(b)–4.4(d), rational Arnoldi algorithm. The shaded area in the upper triangles of the continuation matrices represents the allowed nonzero pattern, while the elements marked with $\times$ represent a particular choice of nonzeros. For instance, the sequential continuation strategy in 4.4(a) corresponds to $\boldsymbol{t}_j = \gamma_j \boldsymbol{e}_j \neq \boldsymbol{0}$. Each of the three parallel variants corresponds to a canonical choice described in section 4.3.2, with a varying number of parallel processes $p > 1$.

are being used repeatedly, which generates this curious nonzero pattern. If the poles were not repeated cyclically, T_m would generally be populated with nonzero elements in the allowed (shaded) region. These canonical choices for the continuation vectors may be supplemented with continuation roots being either zero or infinity, for example.

Let us now move to the admissibility and optimality conditions on continuation pairs in the parallel case. By assumption, the $\underline{p}$ active parallel processes at a given iteration j have mutually distinct poles $\{\mu_{s+\ell}/\nu_{s+\ell}\}_{\ell=1}^{\underline{p}}$, where $s = (j-1)p$. It is easy to show that if for every $\ell = 1, 2, \ldots, \underline{p}$ the continuation pair $(\eta_{s+\ell}^{[\ell]}/\rho_{s+\ell}^{[\ell]}, \boldsymbol{t}_{s+\ell}^{[\ell]})$ is admissible for $AV_{s+1}^{[\ell]}\underline{K_s^{[\ell]}} = V_{s+1}^{[\ell]}\underline{H_s^{[\ell]}}$, that is, if it is *locally admissible* for each parallel process, then no unlucky breakdown occurs during iteration j overall, assuming exact arithmetic and $s + \underline{p} < M$. Hence, an example of admissible continuation pairs for Algorithm 4.5 is $(\eta_{s+\ell}^{[\ell]}/\rho_{s+\ell}^{[\ell]} \not\equiv \mu_{s+\ell}/\nu_{s+\ell}, \boldsymbol{t}_{s+\ell}^{[\ell]})$, with $\boldsymbol{t}_{s+\ell}^{[\ell]}$ provided by (4.32). Unfortunately, obtaining $p > 1$ optimal continuation pairs concurrently is almost always impossible.

Proposition 4.10. *Let $AV_{s+1}^{[\ell]}\underline{K_s^{[\ell]}} = V_{s+1}^{[\ell]}\underline{H_s^{[\ell]}}$ be mutually equal RADs for all $\ell = 1, \ldots, \underline{p}$, and $\{\mu_{s+\ell}/\nu_{s+\ell}\}_{\ell=1}^{\underline{p}}$ be mutually distinct poles, with $\underline{p} > 1$ and $s + \underline{p} < M$. In general, there are no continuation pairs $(\eta_{s+\ell}^{[\ell]}/\rho_{s+\ell}^{[\ell]}, \boldsymbol{t}_{s+\ell}^{[\ell]})$ of order $s+1$ such that $[V_{s+1}^{[\ell]} \quad \boldsymbol{w}_{s+2}^{[1]} \quad \ldots \quad \boldsymbol{w}_{s+\underline{p}+1}^{[\underline{p}]}]$, with the vectors $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ given by line 10 of Algorithm 4.5, is orthonormal.*

Proof. The rational implicit Q theorem implies that the RADs $AV_{s+1}^{[\ell]}\underline{K_s^{[\ell]}} = V_{s+1}^{[\ell]}\underline{H_s^{[\ell]}}$ can be expanded essentially uniquely to $AV_{m+1}^{[\ell]}\underline{K_m^{[\ell]}} = V_{m+1}^{[\ell]}\underline{H_m^{[\ell]}}$, with $m = s + \underline{p}$. Theorem 2.12 provides the representation

$$\boldsymbol{v}_{s+\ell+1}^{[\ell]} = p_{s+\ell}(A)q_{s+\ell}(A)^{-1}\boldsymbol{b}, \quad \text{with} \quad p_{s+\ell}(z) = \det(zK_{s+\ell}^{[\ell]} - H_{s+\ell}^{[\ell]}).$$

Hence, the essentially unique basis vectors $\boldsymbol{v}_{s+\ell+1}^{[\ell]}$ which are mutually orthogonal to each other and to $\mathcal{R}(V_{s+1}^{[\ell]})$ are represented by rational functions $p_{s+\ell}q_{s+\ell}^{-1}$ of type at most $(s+\ell, s+\ell)$. (The type of a rational function is the ordered pair of its numerator and denominator polynomial degrees.) For any $(\eta_{s+\ell}^{[\ell]}/\rho_{s+\ell}^{[\ell]}, \boldsymbol{t}_{s+\ell}^{[\ell]})$, the vectors $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ are rational functions in A times the starting vector $\boldsymbol{b}$ of type at most $(s+1, s+1)$ for all ℓ, which does not match the type $(s+\ell, s+\ell)$ when $\ell > 1$. The only possibility to obtain, e.g., $\boldsymbol{w}_{s+\ell+1}^{[\ell]} = \boldsymbol{v}_{s+\ell+1}^{[\ell]}$ for $\ell > 1$ would be if, by chance, $\ell - 1$ of the (formal) roots of $p_{s+\ell}$ canceled with $\ell - 1$ poles of $q_{s+\ell}$. By remarking that this may never happen, for instance, for Hermitian A with real-valued poles outside the spectral interval of $\Lambda(A)$ and the last pole being infinite, as the roots of the last vector are contained in the aforementioned spectral interval (which is easy to show), we conclude the proof. ∎

For the sequential version we have just enough degrees of freedom to be able to find an optimal continuation pair. For $p > 1$ there is a lack of degrees of freedom, which gets more pronounced as p increases. This can also be interpreted visually in Figure 4.4, where the shaded area decreases with increasing p.

Our proposal is thus to apply the near-optimal framework from section 4.2 locally on each parallel process ℓ, i.e.,

$$\eta_{s+\ell}^{[\ell]}/\rho_{s+\ell}^{[\ell]} \equiv \widehat{\eta}^{[\ell]}/\widehat{\rho}^{[\ell]}, \quad \boldsymbol{t}_{s+\ell}^{[\ell]} = \widehat{\gamma}_{s+\ell}^{-1}\big(\nu_{s+\ell}\widehat{H}_{s+1}^{[\ell]} - \mu_{s+\ell}\widehat{K}_{s+1}^{[\ell]}\big)\widehat{\boldsymbol{x}}^{[\ell]}, \tag{4.33}$$

where $(\widehat{H}_{s+1}^{[\ell]}, \widehat{K}_{s+1}^{[\ell]})$ approximates the pencil $(\underline{H_{s+1}^{[\ell]}}, \underline{K_{s+1}^{[\ell]}})$, that is, the extension of $(\underline{H_{s}^{[\ell]}}, \underline{K_{s}^{[\ell]}})$ with the pole $\mu_{s+\ell}/\nu_{s+\ell}$, and where $(\widehat{\eta}^{[\ell]}/\widehat{\rho}^{[\ell]}, \widehat{\boldsymbol{x}}^{[\ell]})$ is an eigenpair of $(\widehat{H}_{s+1}^{[\ell]}, \widehat{K}_{s+1}^{[\ell]})$ such that $\widehat{\gamma}_{s+\ell} = \widehat{x}_{s+1}^{[\ell]}(\widehat{\rho}^{[\ell]}\widehat{h}_{s+2,s+1}^{[\ell]} - \widehat{\eta}^{[\ell]}\widehat{k}_{s+2,s+1}^{[\ell]}) \neq 0$. This should yield vectors $\boldsymbol{w}_{s+\ell+1}^{[\ell]}$ close to orthogonal to $\mathcal{R}(V_{s+1}^{[\ell]})$, thought nothing can be said a priori about their mutual angles.

With such an approach we expect the condition number of the basis W_{m+1} undergoing the Gram–Schmidt orthogonalization process to increase compared to the sequential case. However, as the growth of $\kappa(W_{m+1})$ gets substantially suppressed with our near-optimal strategy (if the $\|\widehat{\boldsymbol{f}}_{j+1}\|_2$ are small enough), we hope that the growth due to the parallelisation is not prohibitive. Our numerical experiments in section 4.4 confirm that our approach based on near-optimal continuation pairs is more robust than the canonical continuation strategies. We end this section with a few practical considerations.

Real-valued near-optimal continuation pairs. Recall that a near-optimal continuation pair is formed from an eigenpair of (H_j, K_j); cf. (4.6). Even if A, $\boldsymbol{b}$ and the poles are real-valued, a near-optimal continuation pair may hence be complex, which may be undesirable. This problem can be resolved easily: in particular, if j is odd, there is at least one real-valued eigenpair of (H_j, K_j), and it can be used to construct a real-valued continuation pair $(\eta_j/\rho_j, \boldsymbol{t}_j)$. Thus, for the parallel algorithm with p being even, we have that $s+1 = (j-1)p+1$ is odd and hence a real-valued near-optimal continuation pair exists. In our implementation for real-valued data we hence construct near-optimal continuation pairs as in (4.6), but with $(\eta/\rho, \boldsymbol{x})$ replaced by $(\Re(\eta/\rho), \Re(\boldsymbol{x}))$, where $(\eta/\rho, \boldsymbol{x})$ is an eigenpair of (H_j, K_j) such that $\Im(\eta/\rho) = 0$ or otherwise $\Im(\eta/\rho)/\Re(\eta/\rho) \to \min$. Therefore, for odd j or odd $s+1$, we obtain $(\Re(\eta/\rho), \Re(\boldsymbol{x})) = (\eta/\rho, \boldsymbol{x})$. One could also consider the constrained problem of finding a best real-valued $(\eta_j/\rho_j, \boldsymbol{t}_j)$, but we have not done this as the problem with complex continuation pairs disappears with common (even) values of p.

Reordering poles. The ordering of the poles $\{\mu_j/\nu_j\}_{j=1}^m$ is likely to influence the condition number $\kappa(W_{m+1})$ of the basis W_{m+1} being orthogonalized in the Gram–Schmidt process. By (approximately) maximizing the distance between any two distinct poles from $\{\mu_{s+\ell}/\nu_{s+\ell}\}_{\ell=1}^p$ used simultaneously, one may obtain a better conditioned basis W_{m+1}. We have not yet analyzed the numerical influence of the pole ordering and leave this for future work.

4.4 Numerical experiments

In this section we report on two numerical experiments from different application areas, each illustrating another aspect of our parallel algorithms. The algorithms are implemented[1] in C++, using Intel MKL LAPACK and BLAS for dense linear algebra operations, SparseBLAS for sparse matrix-vector multiplications, and PARDISO for the linear system solves (Intel MKL version 10.0.1). Sparse matrices are stored in the CSR format. All tests are run on an Intel Xeon CPU E56-2640, with 6 cores (12 threads), running at 2.5 GHz. We have 64 GB of RAM at our disposal. The

[1] The implementation is available for download from http://www.maths.manchester.ac.uk/~berljafa/RAIN.zip.

code is compiled with the Intel `icpc` compiler (version 12.0.5) using the `-O3` flag. The implementation of the MPI standard is Open MPI (version 1.6). Both the sequential and the parallel rational Arnoldi algorithm are linked with either the sequential or multi-threaded version of Intel MKL, giving raise to the following four configurations:

variant	algorithm	Intel MKL
1×1	Algorithm 2.2	sequential
$1 \times p$	Algorithm 2.2	multi-threaded
$p \times 1$	Algorithm 4.5	sequential
$p \times \hat{p}$	Algorithm 4.5	multi-threaded

Given a computed rational Arnoldi decomposition $AV_{m+1}\underline{K_m} = BV_{m+1}\underline{H_m}$, where B may but does not have to be the identity matrix, we assess various continuation strategies using the following quantities:

`orth` Departure from orthonormality $\|I_{m+1} - \langle V_{m+1}, V_{m+1}\rangle\|_2$. Here, the notation $\langle \cdot, \cdot \rangle : \mathbb{C}^{N,k} \times \mathbb{C}^{N,n} \to \mathbb{C}^{n,k}$ denotes the employed inner product and is application dependent.

`cond` The condition number $\kappa(W_{m+1}D) = \sqrt{\kappa_2(\langle W_{m+1}D, W_{m+1}D\rangle)}$ of the rescaled basis $W_{m+1}D$, with respect to the inner product used. We have used MATLAB `fminsearch` to determine a diagonal matrix D such that $\kappa(W_{m+1}D)$ is (approximately) minimized. This is because the stability of the Gram–Schmidt procedure applied to W_{m+1} is unaffected by column scaling of W_{m+1}.

`space` The space $\mathcal{R}(V_{m+1})$ is a (rational) Krylov space for $B^{-1}A$, where we assume B to be nonsingular, if and only if $S = B^{-1}AV_{m+1} - V_{m+1}\langle B^{-1}AV_{m+1}, V_{m+1}\rangle$ has rank at most one; see [104, Cor. 3.3]. We therefore look at the ratio σ_2/σ_1 of the second largest and the largest singular values of S. The smaller the ratio is, the least $\mathcal{R}(V_{m+1})$ deviates from a (rational) Krylov space.

In all our experiments the relative backward error

$$\|AV_{m+1}\underline{K_m} - BV_{m+1}\underline{H_m}\|_2/(\|A\|_2\|V_{m+1}\|_2\|\underline{K_m}\|_2 + \|B\|_2\|V_{m+1}\|_2\|\underline{H_m}\|_2)$$

of the computed RAD was close to machine precision, and is hence not reported. When reporting the total CPU time, we provide a breakdown made up of four components. The component `mv+orth` measures the elapsed time for lines 2, 11–26, and the computation of

$(\rho_{s+\ell}^{[\ell]}A-\eta_{s+\ell}^{[\ell]}I)V_{s+1}^{[\ell]}\boldsymbol{t}_{s+\ell}^{[\ell]}$ in line 10 of Algorithm 4.5. The component `solve` measures the solution phases consisting of backward and forward substitutions for the linear systems in line 10, while `factorise` measures the initial symbolic and numerical factorisations. Finally, `continuation` measures the time spent in line 9 of Algorithm 4.5. An analogous breakdown is given for Algorithm 2.2.

4.4.1. Exponential integration. Our first example relates to the modeling of a transient electromagnetic field in a geophysical application [17]. We are given a symmetric positive semidefinite matrix $A \in \mathbb{R}^{N,N}$ and a symmetric positive definite matrix $B \in \mathbb{R}^{N,N}$, and the task is to solve $B\boldsymbol{e}'(t) + A\boldsymbol{e}(t) = \boldsymbol{0}$, $\boldsymbol{e}(0) = \boldsymbol{b}$, for the electric field $\boldsymbol{e}(t)$. The time parameters of interest are $t \in T = [10^{-6}, 10^{-3}]$.

The approach suggested in [17] is to build a B-orthonormal (that is, the inner product defined by $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{y}^*B\boldsymbol{x}$ is used) rational Krylov basis $V_{m+1} \in \mathbb{R}^{N,m+1}$ of $\mathcal{Q}_{m+1}(A, B, \boldsymbol{b}, q_m)$, and to extract Arnoldi approximants

$$\boldsymbol{f}_m(t) = \|\boldsymbol{b}\|_B V_{m+1} \exp(-tA_{m+1})\boldsymbol{e}_1, \quad A_{m+1} = V_{m+1}^T A V_{m+1}$$

for all desired time parameters $t \in T$. Here $\|\boldsymbol{b}\|_B = (\boldsymbol{b}^T B\boldsymbol{b})^{1/2}$. The two test problems in [17, Section 5.1] are of sizes $N = 27623$ and $N = 152078$, and they correspond to discretizations of a layered half space using Nédélec elements of orders 1 and 2, respectively. Following [17, Table 1] we set $p = 4$, with mutually distinct poles

$$\{-2.76 \times 10^4, -4.08 \times 10^4, -2.45 \times 10^6, -6.51 \times 10^6\},$$

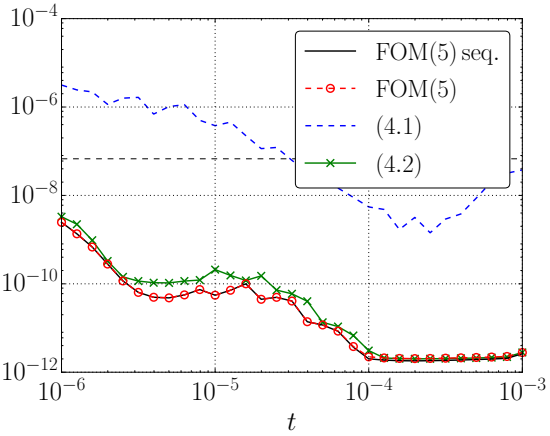
each repeated cyclically 9 times, resulting in a rational Krylov space of order $m = 36$, and guaranteeing Arnoldi approximants with (absolute) errors $\|\boldsymbol{e}(t)-\boldsymbol{f}_m(t)\|_B \le 6.74\times 10^{-8}$ for all $t \in T$, independent of the spectral interval of (A, B).

We test various parallel continuation strategies for computing the basis V_{m+1}, namely, our near-optimal FOM(5) continuation strategy and the two canonical variants specified by (4.30) and (4.31). We also compare to the sequential approach using again the FOM(5) strategy for predicting the next basis vector. The numerical results are shown in Table 4.1 and Figure 4.5. We highlight that all these tests have been run using classical Gram–Schmidt *without reorthogonalization*. The reasoning behind this choice is that our FOM(5) continuation strategy tries to choose continuation pairs which lead to very well-conditioned basis vectors and hence the Gram–Schmidt procedure works

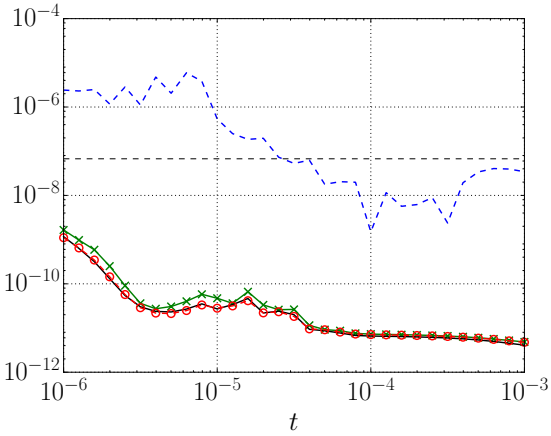
Table 4.1: Numerical quantities associated with the transient electromagnetics problems from Section 4.4.1 solved by various (parallel) rational Arnoldi variants.

strategy	GEOPHYS_{27623}			GEOPHYS_{152078}		
	`cond`	`orth`	`space`	`cond`	`orth`	`space`
$p=1$, (4.13)	7.5×10^{0}	2.2×10^{-14}	1.7×10^{-15}	3.6×10^{1}	3.1×10^{-13}	5.7×10^{-15}
$p=4$, (4.13)	9.1×10^{2}	4.2×10^{-5}	3.5×10^{-14}	7.5×10^{3}	3.2×10^{-1}	6.3×10^{-13}
$p=4$, (4.30)	9.6×10^{9}	1.9×10^{1}	2.1×10^{-7}	6.7×10^{9}	1.8×10^{1}	8.5×10^{-7}
$p=4$, (4.31)	1.9×10^{3}	9.4×10^{-1}	2.8×10^{-14}	1.8×10^{3}	1.1×10^{0}	1.2×10^{-13}

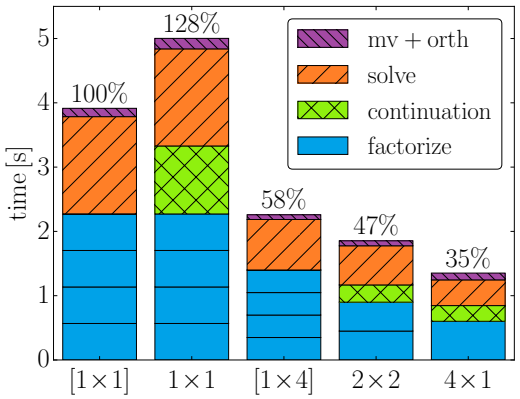
fine without reorthogonalization. This is justified by the condition number `cond` and the orthogonality measure `orth` in Table 4.1, which are both much better with the parallel FOM(5) strategy than they are with the canonical variants (4.30) and (4.31).



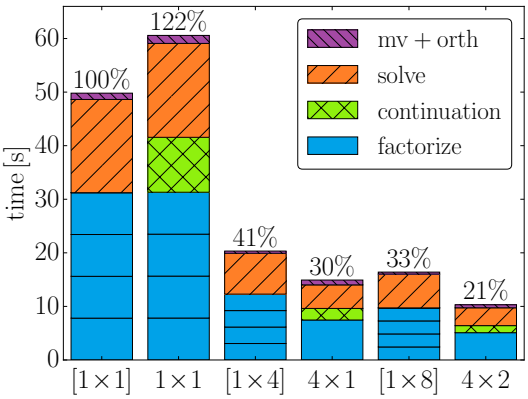
(a) GEOPHYS_{27623}: Error $\|\boldsymbol{e}(t)-\boldsymbol{f}_m(t)\|_B$.



(b) GEOPHYS_{152078}: Error $\|\boldsymbol{e}(t)-\boldsymbol{f}_m(t)\|_B$.



(c) GEOPHYS_{27623}: CPU timings.



(d) GEOPHYS_{152078}: CPU timings.

Figure 4.5: Numerical results for the transient electromagnetics examples from Section 4.4.1.

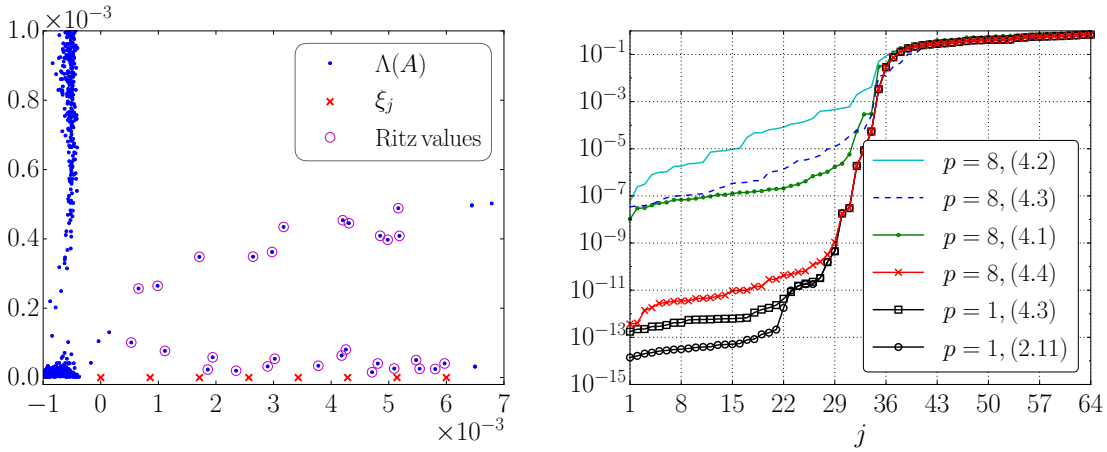
We observe that for variant (4.30) the space $\mathcal{R}(V_{m+1})$ deviates significantly from a rational Krylov space (column `space`). This instability in computing the rational Krylov basis affects the accuracy of the extracted Arnoldi approximants, as can be seen

in Figures 4.5(a)–4.5(b), where we plot the B-norm errors $\|\boldsymbol{e}(t) - \boldsymbol{f}_m(t)\|_B$ as a function of the time parameter t. While our parallel FOM(5) strategy yields approximants of approximately the same accuracy as the sequential FOM(5) variant, the errors of the approximants computed with (4.31) and in particular (4.30) are larger.

In Figures 4.5(c)–4.5(d) we report the CPU timings (averaged over 50 runs) for our C++ implementations. The first bar labeled $[1 \times 1]$ corresponds to the sequential algorithm run using the continuation strategy (4.31). We find that the computationally most expensive parts are the four matrix factorisations (one factorisation for each of the four distinct poles), and the solution phases consisting of the 4×9 backward and forward substitutions. Some speedup is achieved by using four threads to factorise and solve with each system one after the other; note the reduction in computation time when going from $[1 \times 1]$ to $[1 \times 4]$ in our notation. However, it is apparent that even more speedup is obtained by using a single core to factor and solve, but to do this with four matrices *simultaneously* (this corresponds to the 4×1 case), even though our near-optimal FOM(5) continuation strategy adds significant computational cost.

Further reduction in computation time is achieved by combining both levels of parallelism, i.e., factorising and solving with all four matrices simultaneously using two threads in each case (the 4×2 case which we have only timed for the larger example). Let us also point out that the `mv+orth` portion is slightly bigger for the 4×1 case compared to the $[1 \times 4]$ case. This is mainly due to the added communication between the parallel MPI processes within the 4×1 variant. However, the difference is not large and indicates that communication costs are negligible here.

4.4.2. A complex non-Hermitian eigenvalue problem. In this example we consider a finite element discretisation matrix of a three-dimensional waveguide (`waveguide3D`) from The University of Florida Sparse Matrix Collection [22]. This non-Hermitian matrix A is of size $N = 21036$ and has complex entries. Our aim is to compute a few of the propagating wave modes associated with eigenvalues of A close to the interval $[0, 6 \times 10^{-3}]$. To this end we place $p = 8$ equidistant poles on this interval and repeat them cyclically for eight times, thus building a rational Krylov space of order $m = 64$, using modified Gram–Schmidt with reorthogonalization. From the computed rational Krylov decomposition $AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}$ we extract τ-harmonic



(a) Spectrum $\Lambda(A)$ and location of poles. (b) Relative residuals of approximate eigenpairs.

Figure 4.6: Left: "Exact" eigenvalues of the waveguide problem and harmonic Ritz approximations with relative residual norms below 10^{-8} extracted from a rational Krylov space of order $m = 64$ with eight cyclically repeated poles, computed using the near-optimal parallel strategy with $p = 8$ processors. Right: Residual norms of all $m = 64$ harmonic Ritz pairs computed using different (parallel) strategies to compute the rational Krylov basis.

Table 4.2: Numerical quantities for the 3D waveguide example from Section 4.4.2.

strategy	`cond`	`orth`	`space`
$p = 1$, (4.32)	1.6×10^{3}	9.8×10^{-16}	1.8×10^{-13}
$p = 1$, (4.6)	1.1×10^{0}	1.2×10^{-15}	5.3×10^{-15}
$p = 8$, (4.30)	2.5×10^{15}	8.9×10^{-16}	9.1×10^{-2}
$p = 8$, (4.31)	6.8×10^{8}	8.4×10^{-16}	3.0×10^{-8}
$p = 8$, (4.32)	5.0×10^{8}	9.9×10^{-16}	1.1×10^{-8}
$p = 8$, (4.33)	2.1×10^{4}	1.0×10^{-15}	7.3×10^{-12}

Ritz pairs with $\tau = 3 \times 10^{-3}$; see Section 3.1.3 for more details. Figure 4.6(a) visualises the eigenvalues, poles, and harmonic Ritz values.

As is typical for eigenvalue problems, the poles of the rational Krylov space are close to the eigenvalues of A, and hence a continuation prediction using FOM is likely to be unsuccessful. We therefore use the direct solver itself to predict the continuation vectors, which doubles the number of linear system solves but the factorisations are computed only once (per distinct pole). Figure 4.6(b) shows the harmonic Ritz residuals with various continuation strategies discussed in this paper for $p = 8$ processors, including our near-optimal continuation pair (4.33), which becomes optimal for $p = 1$ (i.e., the predicted basis vectors are already orthogonal to the previous vectors). The timings are reported in Figure 4.7. We notice that due to the rather expensive construction of near-optimal continuation pairs the $[1 \times p]$ version is faster than the $p \times 1$ for $p = 2$, but already for $p = 4$ the situation is reversed as then $p \times 1$ scales better.

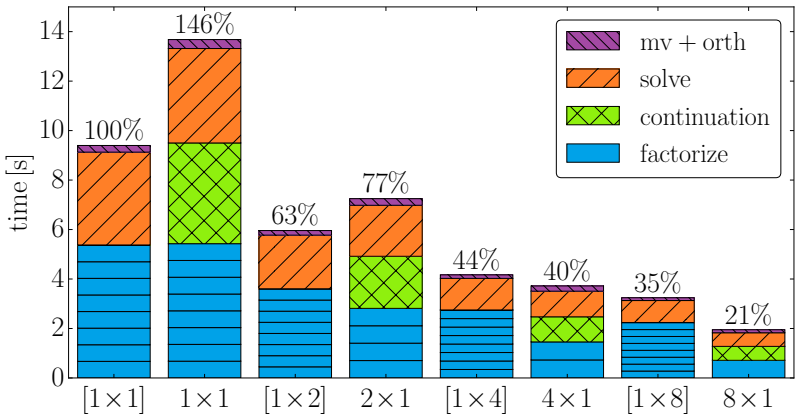


Figure 4.7: CPU timings for the 3D waveguide example from Section 4.4.2.

We note that better results with all variants may be obtained by explicitly projecting the eigenvalue problem with respect to the computed rational Krylov basis, instead of using the quantities from the RAD. Such an explicit projection is, however, undesirable because of the increase in computational cost.

5 Generalised rational Krylov decompositions

In this chapter we study more advanced properties of RADs and quasi-RADs. Let us review some of the main points from Chapter 2. For ease of reference, we review RADs only. For a fixed rational Krylov space $\mathcal{Q}_{m+1} = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ the poles are uniquely defined by the starting vector $\boldsymbol{b}$ and, up to nonzero scaling of $\boldsymbol{b}$, the reverse is true; see Lemma 2.8. Further, by Theorem 2.10, there exists an orthonormal RAD

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m} \tag{5.1}$$

for $\mathcal{Q}_{m+1}$. Upon fixing the order of the poles, Theorem 2.16 guarantees the RAD to be essentially unique.

Interestingly, the rational Krylov space $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ can also be interpreted as a polynomial Krylov space with a different starting vector; $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) = \mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b})$. This follows directly from the definition of a rational Krylov space since $q_m(A)^{-1}$ commutes with A. An interesting characteristic of $\mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b})$ is that the subspaces $\mathcal{K}_{j+1}(A, q_m(A)^{-1}\boldsymbol{b})$ contain all vectors $r_{jm}(A)\boldsymbol{b}$ with $r_{jm} = p_j/q_m \in \mathcal{P}_j/q_m$ being rational functions of type at most (j, m) having the fixed denominator q_m, for $j = 0, 1, \ldots, m$. In fact, $\mathcal{Q}_{m+1}$ can be interpreted as a rational Krylov space with starting vector being almost any vector from $\mathcal{Q}_{m+1}$. Indeed, let a nonzero polynomial $\breve{q}_m \in \mathcal{P}_m$ have roots disjoint from $\Lambda(A)$, then

$$\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) = \mathcal{Q}_{m+1}(A, \breve{q}_m(A)q_m(A)^{-1}\boldsymbol{b}, \breve{q}_m). \tag{5.2}$$

A natural task to consider is that of transforming a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ into one for $\mathcal{Q}_{m+1}(A, \breve{q}_m(A)q_m(A)^{-1}\boldsymbol{b}, \breve{q}_m)$. In Section 5.1 we develop two algorithms for this task, and in Section 5.2 we devise another one specifically tailored to the important

case of a polynomial Krylov space. These considerations are linked to implicit filtering for eigenvalue approximations, and further applications are considered in Chapters 6–7.

5.1 Rational Krylov decompositions

We develop two algorithms for obtaining an RAD for $\mathcal{Q}_{m+1}(A, \breve{q}_m(A)q_m(A)^{-1}\boldsymbol{b}, \breve{q}_m)$ from an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$. The first algorithm relocates the poles by changing the starting vector to a new one, while the second algorithm relocates poles to explicitly given new ones. The relocation is, in both cases, conducted by applying transformations to an initial RAD. These transformations destroy the RAD structure, which then needs to be appropriately restored. While the first algorithm is applicable to both RADs and quasi-RADs, the second algorithm is applicable to RADs only.

5.1.1. Moving the poles implicitly. Let (2.6) be a (quasi-)RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, and let $\breve{\boldsymbol{b}} = V_{m+1}\boldsymbol{c} \in \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ be a nonzero vector. Take any nonsingular matrix $P \in \mathbb{C}^{m+1,m+1}$ such that $P\boldsymbol{e}_1 = \boldsymbol{c}$. (Of course, for quasi-RADs we want $P \in \mathbb{R}^{m+1,m+1}$, which also means that we consider only real-valued vectors $\boldsymbol{c}$.) Then

$$A\breve{V}_{m+1}\underline{\breve{K}_m} = \breve{V}_{m+1}\underline{\breve{H}_m} \tag{5.3}$$

holds with $\breve{V}_{m+1} := V_{m+1}P$, $\underline{\breve{H}_m} := P^{-1}\underline{H_m}$ and $\underline{\breve{K}_m} := P^{-1}\underline{K_m}$. This construction guarantees the first column $\breve{\boldsymbol{v}}_1$ of $\breve{V}_{m+1}$ to be equal to $\breve{\boldsymbol{b}}$, however, the pencil $(\underline{\breve{H}_m}, \underline{\breve{K}_m})$ may loose the upper Hessenberg structure. In the following we aim at recovering this structure in (5.3) without affecting $\breve{\boldsymbol{v}}_1$. For that purpose we generalise the notion of RADs by first giving a technical definition. For a matrix $\underline{X_m} \in \mathbb{C}^{m+1,m}$ the notation $\underline{X_{-m}} := \begin{bmatrix}\boldsymbol{0} & I_m\end{bmatrix}\underline{X_m}$ is used to denote its lower m-by-m submatrix.

Definition 5.1. *Let $\underline{\breve{K}_m}, \underline{\breve{H}_m} \in \mathbb{C}^{m+1,m}$. We say that the pencil $(\underline{\breve{H}_m}, \underline{\breve{K}_m})$ is* regular *if the lower m-by-m subpencil $(\underline{\breve{H}_{-m}}, \underline{\breve{K}_{-m}})$ is regular, i.e., $\breve{q}_m(z) = \det\left(z\underline{\breve{K}_{-m}} - \underline{\breve{H}_{-m}}\right)$ is not identically equal to zero.*

Note that an upper Hessenberg pencil of size $(m+1)$-by-m is unreduced if and only if it is regular. We are now ready to introduce decompositions of the form (5.3).

Definition 5.2. *Let $A \in \mathbb{C}^{N,N}$. A relation of the form (5.3) where $\breve{V}_{m+1} \in \mathbb{C}^{N,m+1}$ is of full column rank and the $(m+1)$-by-m pencil $(\underline{\breve{H}_m}, \underline{\breve{K}_m})$ is regular is called a* generalised rational Krylov decomposition of order m. *The generalised eigenvalues of*

$(\breve{\underline{H}}_{-m}, \breve{\underline{K}}_{-m})$ *are called* poles *of the decomposition. If no pole of* (5.3) *is in* $\Lambda(A)$*, then* (5.3) *is called a* rational Krylov decomposition (RKD).

The notion of (*orthonormal*) *basis, space* and *equivalent decompositions* are the same as for RADs. We call a generalised RKD with an upper Hessenberg pencil a *generalised RAD*. The two definitions above let us speculate that the unique poles associated with $\breve{\boldsymbol{b}}$ are the eigenvalues of $(\breve{\underline{H}}_{-m}, \breve{\underline{K}}_{-m})$. The justification follows from Theorem 2.10 (or, alternatively, Theorem 2.16) and the following result.

Theorem 5.3. *Any* (*generalised*) *RKD is equivalent to a* (*generalised*) *RAD.*

Proof. Let (5.3) be a generalised RKD. We need to bring both $\breve{\underline{H}}_m$ and $\breve{\underline{K}}_m$ into upper Hessenberg form. To achieve this it suffices to bring $\big(\breve{\underline{H}}_{-m}, \breve{\underline{K}}_{-m}\big)$ into generalised Schur form. The existence of unitary $Q_m, Z_m \in \mathbb{C}^{m,m}$ such that $Q_m^* \breve{\underline{H}}_{-m} Z_m$ and $Q_m^* \breve{\underline{K}}_{-m} Z_m$ are both upper triangular follows from, e.g., [42, Theorem 7.7.1]. Multiplying (5.3) from the right with Z_m and "inserting" $I_{m+1} = Q_{m+1} Q_{m+1}^*$, where $Q_{m+1} = \mathrm{blkdiag}(1, Q_m)$, we obtain the generalised RAD

$$A\big(\breve{V}_{m+1} Q_{m+1}\big) Q_{m+1}^* \breve{\underline{K}}_m Z_m = \big(\breve{V}_{m+1} Q_{m+1}\big) Q_{m+1}^* \breve{\underline{H}}_m Z_m.$$

Label $V_{m+1} := \breve{V}_{m+1} Q_{m+1}$, $\underline{H_m} := Q_{m+1}^* \breve{\underline{H}}_m Z_m$ and $\underline{K_m} := Q_{m+1}^* \breve{\underline{K}}_m Z_m$. Remarking that $\mathcal{R}(\breve{V}_{m+1}) = \mathcal{R}(V_{m+1})$, and that $(\underline{H_m}, \underline{K_m})$ is an upper Hessenberg pencil with poles being identical to those of $(\breve{\underline{H}}_m, \breve{\underline{K}}_m)$, we conclude the proof. ∎

The proof of Theorem 5.3 is constructive and provides an algorithm for transforming a generalised RKD into a generalised RAD; see Algorithm 5.6. This discussion is summarised in Algorithm 5.7, which replaces the starting vector $\boldsymbol{b}$ with $\breve{\boldsymbol{b}} = V_{m+1} \boldsymbol{c}$. The matrix P having its first column (a multiple of) the vector $\boldsymbol{c}$ is the (possibly complex-valued) Householder reflector P_{m+1} (see, e.g., [42, Section 5.1.13]) generated in line 1.

Note that there is no guarantee that by transforming an RAD the resulting decomposition is also an RAD, i.e., some poles may be moved to eigenvalues of A. We prove later (cf. Theorem 5.4) that if $\breve{\boldsymbol{b}} = V_{m+1} \boldsymbol{c} = \breve{p}_m(A) q_m(A)^{-1} \boldsymbol{b}$ for some $\breve{p}_m \in \mathcal{P}_m$, then the poles of the decomposition are always the roots of $\breve{p}_m$, even if they coincide with eigenvalues of A.

5.1.2. Moving the poles explicitly. If the vector $\breve{\boldsymbol{b}}$ is not given as a linear combination of the basis vectors V_{m+1} but rather by specifying the new poles $\breve{q}_m$ one

Algorithm 5.6 RAD structure recovery. RKToolbox: `util_recover_rad`

Input: Generalised RKD (5.3) and a flag `quasi`.
Output: Generalised RAD (2.6) equivalent to (5.3).

1. **if** Flag `quasi` is set to `true` and (5.3) is real-valued. **then**
2. Find orthogonal matrices $Q_m, Z_m \in \mathbb{R}^{m,m}$ such that $(Q_m^* \breve{\underline{H}}_{-m} Z_m, Q_m^* \breve{\underline{K}}_{-m} Z_m)$ is in generalised real Schur form, and let $Q_{m+1} := \operatorname{blkdiag}(1, Q_m)$.
3. **else**
4. Find unitary matrices $Q_m, Z_m \in \mathbb{C}^{m,m}$ such that $(Q_m^* \breve{\underline{H}}_{-m} Z_m, Q_m^* \breve{\underline{K}}_{-m} Z_m)$ is in generalised Schur form, and let $Q_{m+1} := \operatorname{blkdiag}(1, Q_m)$.
5. **end if**
6. Define $V_{m+1} := \breve{V}_{m+1} Q_{m+1}$, $\underline{H_m} := Q_{m+1}^* \breve{\underline{H}}_m Z_m$ and $\underline{K_m} := Q_{m+1}^* \breve{\underline{K}}_m Z_m$.

Algorithm 5.7 Implicit pole placement. RKToolbox: `move_poles_impl`

Input: Generalised RAD (5.3) and unit 2-norm $\boldsymbol{e}_1 \neq \boldsymbol{c} \in \mathbb{C}^{m+1}$.
Output: Generalised RAD (2.6) spanning $\mathcal{R}(\breve{V}_{m+1})$ with $\boldsymbol{v}_1 = \breve{V}_{m+1}\boldsymbol{c}$.

1. Define $P_{m+1} := I_{m+1} - 2\boldsymbol{u}\boldsymbol{u}^*$, where $\boldsymbol{u} := (\boldsymbol{c} - \boldsymbol{e}_1)/\|\boldsymbol{c} - \boldsymbol{e}_1\|_2$.
2. Update $\breve{V}_{m+1} := V_{m+1} P_{m+1}$, $\breve{\underline{H}}_m := P_{m+1}^* \breve{\underline{H}}_m$, and $\breve{\underline{K}}_m := P_{m+1}^* \breve{\underline{K}}_m$.
3. Apply Algorithm 5.6 to the updated (5.3) to produce (2.6).

can compute $\boldsymbol{c} = V_{m+1}^* \breve{\boldsymbol{b}}$, where $\breve{\boldsymbol{b}} = \breve{q}_m(A) q_m(A)^{-1} \boldsymbol{b}$, and still use Algorithm 5.7 to recover the new decomposition. In the following we present an approach that works directly with the pencil $(\underline{H_m}, \underline{K_m})$, changing the poles iteratively one after the other.

Moving the first pole. The poles are the ratios of the subdiagonal elements of $(\underline{H_m}, \underline{K_m})$; see the discussion following (2.5). Applying a Givens rotation G acting on planes $(1, 2)$ from the left of the pencil preserves the upper Hessenberg structure and, as we show, can move the first pole anywhere. We now derive the formulas for $s = e^{\mathrm{i}\phi} \sin\vartheta$ and $c = \cos\vartheta$ satisfying $c^2 + |s|^2 = 1$ and such that the Givens rotation

$$G = \operatorname{blkdiag}\left(\begin{bmatrix} c & -s \\ \overline{s} & c \end{bmatrix}, I_{m-1} \right) \tag{5.4}$$

replaces the pole $\xi_1 =: \mu_1/\nu_1$ with $\breve{\xi}_1 =: \breve{\mu}_1/\breve{\nu}_1$ when applied to the pencil from the left. Define $\breve{\underline{H}}_m = G\underline{H_m}$ and $\breve{\underline{K}}_m = G\underline{K_m}$. This gives

$$\begin{aligned} \breve{h}_{11} &= c h_{11} - s h_{21}, \quad \breve{k}_{11} = c k_{11} - s k_{21}, \\ \breve{h}_{21} &= \overline{s} h_{11} + c h_{21}, \quad \breve{k}_{21} = \overline{s} k_{11} + c k_{21}. \end{aligned} \tag{5.5}$$

Additionally, G is chosen so that $\breve{\mu}_1/\breve{\nu}_1 = \breve{h}_{21}/\breve{k}_{21}$. Using $t := \overline{s}/c$, we derive

$$t = \frac{\breve{\nu}_\ell h_{21} - \breve{\mu}_\ell k_{21}}{\breve{\mu}_\ell k_{11} - \breve{\nu}_\ell h_{11}}, \tag{5.6}$$

where $\ell = 1$. With the help of standard trigonometric relations we then arrive at

$$\begin{aligned} &c = (1+|t|^2)^{-1/2}, \quad s = \overline{t}c, \qquad \text{if } t \neq \infty, \text{ and otherwise,} \\ &c = 0, \qquad\qquad\quad s = 1. \end{aligned} \tag{5.7}$$

Formula (5.2) asserts (with the roots of $\breve{q}_m$ being $\breve{\xi}_1, \xi_2, \xi_3, \ldots, \xi_m$) that this process replaces the starting vector $\boldsymbol{v}_1$ with a multiple of $(\breve{\nu}_1 A - \breve{\mu}_1 I)(\nu_1 A - \mu_1 I)^{-1}\boldsymbol{v}_1$. Let us verify that. Define $\breve{V}_{n+1} = V_{n+1}G^*$. In particular,

$$\breve{\boldsymbol{v}}_1 = c\boldsymbol{v}_1 - \overline{s}\boldsymbol{v}_2. \tag{5.8}$$

Recall that (2.12) reads $(h_{21}I - k_{21}A)\boldsymbol{v}_2 = (k_{11}A - h_{11}I)\boldsymbol{v}_1$. Hence, using the relation (2.12) within (5.8) together with (5.5) provides

$$\big(h_{21}I - k_{21}A\big)\breve{\boldsymbol{v}}_1 = \big[c\big(h_{21}I - k_{21}A\big) - \overline{s}\big(k_{11}A - h_{11}I\big)\big]\,\boldsymbol{v}_1 = \big(\breve{h}_{21}I - \breve{k}_{21}A\big)\boldsymbol{v}_1. \tag{5.9}$$

Note that (2.12) holds even if $h_{21}/k_{21} = \xi_1 \in \Lambda(A)$ as long as the generalised RAD (2.6) exists. As we impose no constraints on $\breve{\xi}_1$, we conclude that (5.9) holds even if $\breve{\xi}_1 \in \Lambda(A)$ and/or $\xi_1 \in \Lambda(A)$. If however $\xi_1 \notin \Lambda(A)$ we can further write

$$\breve{\boldsymbol{v}}_1 = \big(\breve{h}_{21}I - \breve{k}_{21}A\big)\big(h_{21}I - k_{21}A\big)^{-1}\boldsymbol{v}_1.$$

Moving all poles. Changing the other ratios with Givens rotations results in the loss of the upper Hessenberg structure. However, the poles are the eigenvalues of the pencil $(\underline{H_{-m}}, \underline{K_{-m}})$ which is (already) in generalised Schur form. After changing the first pole, using the Givens rotation approach just described, the poles can be reordered (see for instance [68, 70]) with the aim of bringing an unchanged pole to the front of the decomposition so that it can be changed using a Givens rotation. Theoretically, reordering the poles of an RAD is equivalent to using the poles within the rational Arnoldi algorithm in a different order. This process is formalized in Algorithms 5.8–5.9 and an illustration is presented in Figure 5.1.

Let us now consider Algorithm 5.9 when $k = m$. As we have shown in (5.9), after applying the first Givens rotation the starting vector $\boldsymbol{v}_1$ gets replaced with $\boldsymbol{v}_1^{[1]}$ satisfying

$$\big(\nu_1 A - \mu_1 I\big)\boldsymbol{v}_1^{[1]} = \gamma_1\big(\breve{\nu}_m A - \breve{\mu}_m I\big)\boldsymbol{v}_1, \tag{5.10}$$

where $0 \neq \gamma_1 \in \mathbb{C}$ is a scaling factor. By reordering the poles we do not affect the "new starting vector" $\boldsymbol{v}_1^{[1]}$ and bring ξ_2 to the leading positions, i.e., second row, first column,

Algorithm 5.8 RAD poles reordering. RKToolbox: `util_reorder_poles`

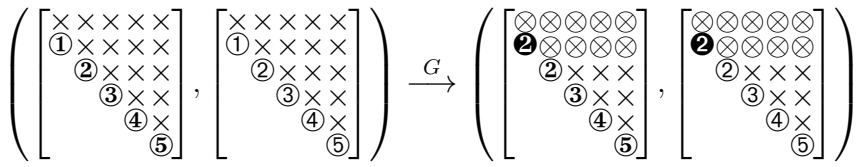
Input: Generalised RAD (2.6) and permutation $\pi \in S_\ell$, with $\ell \in \{2, 3, \ldots, m\}$.
Output: Equivalent generalised RAD (2.6) with reordered poles.

1. Find unitary matrices $Q_\ell, Z_\ell \in \mathbb{C}^{\ell,\ell}$ such that $(Q_\ell^* \underline{H_{-\ell}} Z_\ell, Q_\ell^* \underline{K_{-\ell}} Z_\ell)$ is in generalised Schur form with the jth generalised eigenvalue being $h_{\pi(j)+1,\pi(j)}/k_{\pi(j)+1,\pi(j)}$, for $j = 1, 2, \ldots, \ell$, and let $Q_{m+1} := \mathrm{blkdiag}(1, Q_\ell, I_{m-\ell})$ and $Z_m := \mathrm{blkdiag}(Z_\ell, I_{m-\ell})$.
2. Update $V_{m+1} := V_{m+1} Q_{m+1}$, $\underline{H_m} := Q_{m+1}^* \underline{H_m} Z_m$, and $\underline{K_m} := Q_{m+1}^* \underline{K_m} Z_m$.

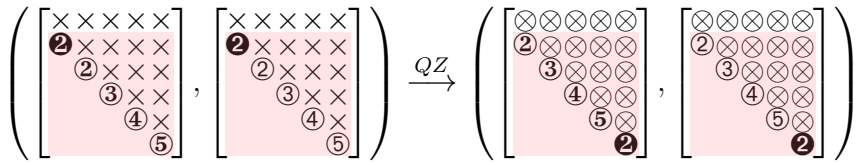
Algorithm 5.9 Explicit pole placement. RKToolbox: `move_poles_expl`

Input: Generalised RAD (2.6) and $\{\breve{\xi}_j\}_{j=1}^k \subset \overline{\mathbb{C}}$, with $k \in \{1, 2, \ldots, m\}$.
Output: Updated generalised RAD (2.6) with the first k poles replaced by $\{\breve{\xi}_j\}_{j=1}^k$.

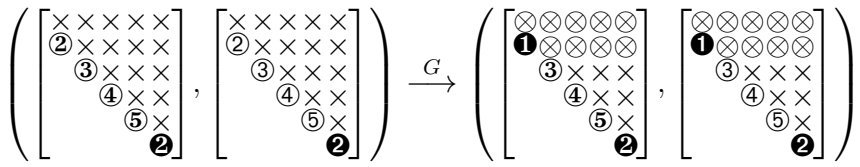
1. **for** $j = 1, 2, \ldots, k$ **do**
2. Let $\ell = k + 1 - j$, and introduce any $\breve{\mu}_\ell, \breve{\nu}_\ell \in \mathbb{C}$ such that $\breve{\mu}_\ell / \breve{\nu}_\ell = \breve{\xi}_\ell$.
3. Define G as in (5.4), with c and s given by (5.7), and t by (5.6).
4. Update $V_{m+1} := V_{m+1} G^*$, $\underline{H_m} := G \underline{H_m}$, and $\underline{K_m} := G \underline{K_m}$.
5. Update (2.6) with Algorithm 5.8 using $\pi = (2, 3, \ldots, \ell, 1) \in S_\ell$, if $\ell > 1$.
6. **end for**



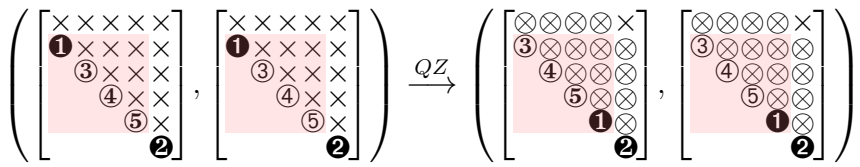
(a) Applying the first Givens rotation to replace $\xi_1 = ①/①$ with $\widehat{\xi}_2 = ❷/❷$.



(b) Reordering the generalised Schur form of the lower 5×5 part.



(c) Applying the second Givens rotation to replace $\xi_2 = ②/②$ with $\widehat{\xi}_1 = ❶/❶$.



(d) Reordering the generalised Schur form of size 4.

Figure 5.1: Looking at the 6-by-5 upper-Hessenberg pencil while Algorithm 5.9 is applied on the corresponding RAD with $k = 2$. The original poles are the ratios ①/①,②/②,..., ⑤/⑤. The first two poles are replaced with ❷/❷ and ❶/❶. The transition from $\times$ to $\otimes$ symbolizes that the element potentially changes.

where the next Givens rotation acts. Thus, for $j = 2$ the Givens rotation replaces $\boldsymbol{v}_1^{[1]}$ with $\boldsymbol{v}_1^{[2]}$ satisfying $(\nu_2 A - \mu_2 I)\boldsymbol{v}_1^{[2]} = \gamma_2(\breve{\nu}_{m-1}A - \breve{\mu}_{m-1}I)\boldsymbol{v}_1^{[1]}$, where $0 \neq \gamma_2 \in \mathbb{C}$ is a scaling factor. Using (5.10) we obtain

$$(\nu_1 A - \mu_1 I)(\nu_2 A - \mu_2 I)\boldsymbol{v}_1^{[2]} = \gamma_1\gamma_2(\breve{\nu}_{m-1}A - \breve{\mu}_{m-1}I)(\breve{\nu}_m A - \breve{\mu}_m I)\boldsymbol{v}_1.$$

Reasoning inductively we deduce

$$q_m(A)\breve{\boldsymbol{v}}_1 = \gamma\breve{q}_m(A)\boldsymbol{v}_1, \tag{5.11}$$

where $0 \neq \gamma \in \mathbb{C}$ is a scaling factor, $\breve{\boldsymbol{v}}_1 = \boldsymbol{v}_1^{[m]}$, q_m is given by (2.9), and $\breve{q}_m$ is defined in an analogous manner. The above discussion is the gist of the following result.

Theorem 5.4. *Let $\mathcal{Q}_{m+1} = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ be A-variant. If the generalised RKD (5.3) with poles $\breve{q}_m$ spans $\mathcal{Q}_{m+1}$ then $\breve{\boldsymbol{v}}_1 = \gamma\breve{q}_m(A)q_m(A)^{-1}\boldsymbol{b}$ with a scalar $0 \neq \gamma \in \mathbb{C}$. Alternatively, if $\breve{\boldsymbol{v}}_1 = \breve{q}_m(A)q_m(A)^{-1}\boldsymbol{b}$ then there exists a generalised RKD with poles $\breve{q}_m$ spanning $\mathcal{Q}_{m+1}$.*

Proof. If (5.3) spans $\mathcal{Q}_{m+1}$ we can transform it into an equivalent generalised RAD (cf. Theorem 5.3) and then, using Algorithm 5.9, into (2.6), having poles q_m and still spanning $\mathcal{Q}_{m+1}$. According to Lemma 2.8, $\boldsymbol{v}_1$ is collinear with $\boldsymbol{b}$. Therefore, it follows from (5.11) that $\breve{\boldsymbol{v}}_1 = \gamma\breve{q}_m(A)q_m(A)^{-1}\boldsymbol{b}$ for some scalar $0 \neq \gamma \in \mathbb{C}$. The other direction follows from Theorem 2.10 and (5.11) after using Algorithm 5.9. ∎

Corollary 5.5. *Let (5.3) be a generalised RKD, and let $\alpha, \beta \in \mathbb{C}$ be such that $|\alpha|+|\beta| \neq 0$. The matrix $\alpha\underline{\breve{H}_m} - \beta\underline{\breve{K}_m}$ is of full column rank m.*

Proof. The poles of (5.3) can be moved anywhere outside $\Lambda(A)$, and the corresponding RKD can be transformed, cf. Theorem 5.3, into an an RAD (2.6) so that overall $\underline{H_m} = Q_{m+1}^*\underline{\breve{H}_m}Z_m$, and $\underline{K_m} = Q_{m+1}^*\underline{\breve{K}_m}Z_m$, where $Q_{m+1} \in \mathbb{C}^{m+1,m+1}$ and $Z_m \in \mathbb{C}^{m,m}$ are unitary. By Lemma 2.6, $\alpha\underline{H_m} - \beta\underline{K_m}$ is of full column rank m, and hence $Q_{m+1}(\alpha\underline{H_m} - \beta\underline{K_m})Z_m^*$ is as well. ∎

Theorem 5.4 shows that Algorithm 5.7 and Algorithm 5.9 are equivalent, provided that equivalent input data are given. It also shows, together with Theorem 2.10 and Theorem 5.3, that an $(m+1)$-dimensional space $\mathcal{V}_{m+1}$ is a rational Krylov space if and only if there exist a generalised RKD spanning $\mathcal{V}_{m+1}$.

Implicitly restarted rational Arnoldi algorithm. Implicit filtering, or restarting, aims at compressing the space $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ into $\mathcal{Q}_{m+1-k}(A, p_k(A)q_k(A)^{-1}\boldsymbol{b}, \breve{q}_{m-k})$,

where $k \in \{1, 2, \ldots, m\}$, $q_m = q_k \cdot \breve{q}_{m-k}$, and $p_k \in \mathcal{P}_k$ is a polynomial with (formal) roots (infinity allowed) in the region we want to filter out. In applications this technique is usually used to deal with large memory requirements or orthogonalization costs for V_{m+1}, or to purge unwanted or spurious eigenvalues (see, e.g., [19, 23, 24] and the references given therein). Implicit filtering for RADs was first introduced in [24] and further studied in [23]. Algorithm 5.9 can easily be used for implicit filtering. In fact, applying Algorithm 5.9 with the k poles $\breve{\xi}_j$ being the roots of p_k implicitly applies the filter $p_k(A)q_k(A)^{-1}$ to the RAD. The k "new" poles correspond to the rightmost k columns in $\breve{V}_{m+1}$, $\underline{\breve{K}_m}$ and $\underline{\breve{H}_m}$, cf. Figure 5.1. Hence, truncating the decomposition to the leading $m + 1 - k$ columns completes the process. The derivation and algorithms in [23, 24] are different, and it would perhaps be interesting to compare them. This is, however, not done here. Pertinent ideas for polynomial Krylov methods have recently appeared in [19] where the authors relate implicit filtering in the Krylov–Schur algorithm [101, 103] with partial eigenvalue assignment.

5.2 Connection with polynomial Krylov spaces

For the particular case $\breve{q}_m(z) = 1$ we have $\mathcal{Q}_{m+1}(A, \boldsymbol{v}, q_m) = \mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{v})$, and we can recover a polynomial Arnoldi decomposition for $\mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{v})$ from an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{v}, q_m)$ using Algorithm 5.9 with all poles $\breve{\xi}_j$ set to infinity. In this case, a simpler method is to bring the pencil $\left(\underline{H_m}, \underline{K_m}\right)$ from upper-(quasi-)Hessenberg–upper-Hessenberg to upper-Hessenberg–upper-triangular form. We shall refer to the so obtained RAD as a *polynomial RAD*. The algorithm for RADs is outlined in [92, p. 495], and we now review it. Afterwards, we generalise the algorithm to handle quasi-RADs as well. The algorithm uses Givens rotation in a QZ-like fashion and to facilitate the discussion, we use a graphical representation of the pencil before and after a particular transformation, like, for instance,

$$\left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}\right) \longrightarrow \left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \\ & & & \end{bmatrix}\right),$$

which depicts the starting structure of $(\underline{H_m}, \underline{K_m})$ on the left-hand side, and the sought after structure on the right-hand side. The goal is thus to annihilate the subdiagonal of $\underline{K_m}$. We achieve this by applying a sequence of Givens rotations to the pencil $(\underline{H_m}, \underline{K_m})$

from both sides. To this end we denote with $Q_{\ell,\ell-1} \in \mathbb{C}^{m+1,m+1}$ a Givens rotation acting in planes $(\ell-1,\ell)$, i.e.,

$$\widehat{Q}_{\ell,m}^* Q_{\ell,\ell-1} \widehat{Q}_{\ell,m} = I_{m-1}, \quad \text{and} \quad \begin{bmatrix} \boldsymbol{e}_{\ell-1} & \boldsymbol{e}_\ell \end{bmatrix}^* Q_{\ell,\ell-1} \begin{bmatrix} \boldsymbol{e}_{\ell-1} & \boldsymbol{e}_\ell \end{bmatrix} = \begin{bmatrix} c & -s \\ \overline{s} & c \end{bmatrix}, \tag{5.12}$$

where $\widehat{Q}_{\ell,\ell-1} = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_{\ell-2} & \boldsymbol{e}_{\ell+1} & \boldsymbol{e}_{\ell+2} & \dots & \boldsymbol{e}_{m+1} \end{bmatrix}$, and c and s satisfy

$$c^2 + |s|^2 = 1, \quad \text{and} \quad \begin{bmatrix} c & -s \\ \overline{s} & c \end{bmatrix} \begin{bmatrix} k_{\ell-1,\ell-1} \\ k_{\ell,\ell-1} \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}. \tag{5.13}$$

With $Z_{k,\ell-1} \in \mathbb{C}^{m,m}$ we denote a Givens rotation acting in planes $(\ell-1,\ell)$, i.e.,

$$\widehat{Z}_{\ell,m}^* Z_{k,\ell-1} \widehat{Z}_{\ell,m} = I_{m-2}, \quad \text{and} \quad \begin{bmatrix} \boldsymbol{e}_{\ell-1} & \boldsymbol{e}_\ell \end{bmatrix}^* Z_{k,\ell-1} \begin{bmatrix} \boldsymbol{e}_{\ell-1} & \boldsymbol{e}_\ell \end{bmatrix} = \begin{bmatrix} c & -s \\ \overline{s} & c \end{bmatrix}, \tag{5.14}$$

where $\widehat{Z}_{\ell,m} = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_{\ell-2} & \boldsymbol{e}_{\ell+1} & \boldsymbol{e}_{\ell+2} & \dots & \boldsymbol{e}_m \end{bmatrix}$, and c and s satisfy

$$c^2 + |s|^2 = 1, \quad \text{and} \quad \begin{bmatrix} h_{k,\ell-1} & h_{k,\ell} \end{bmatrix} \begin{bmatrix} c & -s \\ \overline{s} & c \end{bmatrix} = \begin{bmatrix} 0 & \times \end{bmatrix}. \tag{5.15}$$

For the graphical illustrations we denote by $\oplus$ the element we are about to annihilate, and by $\otimes$ the ellement we annihilate with. The first step is thus

$$\left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \otimes & \times & \times & \times \\ \oplus & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \right) \xrightarrow{Q_{21}} \left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \\ & & & \end{bmatrix} \right),$$

and the RAD (3.1) is updated as

$$V_{m+1} := V_{m+1} Q_{\ell,\ell-1}^*, \quad \underline{H_m} := Q_{\ell,\ell-1} \underline{H_m}, \quad \text{and} \quad \underline{K_m} := Q_{\ell,\ell-1} \underline{K_m}, \tag{5.16}$$

with $\ell = 2$. We can continue with

$$\left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \times & \times & \times & \times \\ & \otimes & \times & \times \\ & \oplus & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \right) \xrightarrow{Q_{32}} \left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \right)$$

and perform the update (5.16) with $\ell = 3$. This may destroy the upper Hessenberg structure of $\underline{H_m}$, as h_{32} may be nonzero. The structure can be recovered by applying Z_{31} from the right:

$$\left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \oplus & \otimes & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \right) \xrightarrow{Z_{31}} \left(\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}, \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \right),$$

and hence, the updates

$$\underline{H_m} := \underline{H_m} Z_{k,\ell-1}, \quad \text{and} \quad \underline{K_m} := \underline{K_m} Z_{k,\ell-1}, \tag{5.17}$$

Algorithm 5.10 (Quasi-)RAD to polynomial RAD. RKToolbox: `util_hh2th`

Input: (Quasi-)RAD (3.1) for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$.
Output: Polynomial RAD (3.1) for $\mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b})$.

1. Initialize $j := 1$.
2. **while** $j \leq m$ **do**
3. **if** $j+1 \leq m$ & $h_{j+2,j} \neq 0$ **then** ▷ Annihilating a 2×2 block.
4. Perform (5.16), where $Q_{\ell,\ell-1}$ satisfies (5.12)–(5.13); $\ell = j+1$.
5. Perform (5.16), where $Q_{\ell,\ell-1}$ satisfies (5.12)–(5.13); $\ell = j+2$.
6. **for** $i = j+2, j+1, \ldots, 4$ **do** ▷ Bulge chasing.
7. Perform (5.17), where $Z_{k,\ell-1}$ satisfies (5.14)–(5.15); $k = i$, and $\ell = i-2$.
8. Perform (5.17), where $Z_{k,\ell-1}$ satisfies (5.14)–(5.15); $k = i$, and $\ell = i-1$.
9. Perform (5.16), where $Q_{\ell,\ell-1}$ satisfies (5.12)–(5.13); $\ell = i-2$.
10. Perform (5.16), where $Q_{\ell,\ell-1}$ satisfies (5.12)–(5.13); $\ell = i-1$.
11. **end for**
12. Perform (5.17), where $Z_{k,\ell-1}$ satisfies (5.14)–(5.15); $k = 3$, and $\ell = 2$.
13. Perform (5.16), where $Q_{\ell,\ell-1}$ satisfies (5.12)–(5.13); $\ell = 2$.
14. Increase $j := j+1$.
15. **else** ▷ Annihilating a single element.
16. Perform (5.16), where $Q_{\ell,\ell-1}$ satisfies (5.12)–(5.13); $\ell = j+1$.
17. **for** $i = j+1, j, \ldots, 3$ **do** ▷ Bulge chasing.
18. Perform (5.17), where $Z_{k,\ell-1}$ satisfies (5.14)–(5.15); $k = i$, and $\ell = i-1$.
19. Perform (5.16), where $Q_{\ell,\ell-1}$ satisfies (5.12)–(5.13); $\ell = i-1$.
20. **end for**
21. **end if**
22. Increase $j := j+1$.
23. **end while**

with $k = 3$ and $\ell = 2$ are in order. The element k_{21} may now be nonzero, and the structure in $\underline{K_m}$ is recovered by another Givens rotation of the form Q_{21}:

$$\left(\begin{bmatrix}\times&\times&\times&\times\\ \times&\times&\times&\times\\ &\times&\times&\times\\ &&\times&\times\\ &&&\times\end{bmatrix}, \begin{bmatrix}\otimes&\times&\times&\times\\ \oplus&\times&\times&\times\\ &&\times&\times\\ &&\times&\times\\ &&&\times\end{bmatrix}\right) \xrightarrow{Q_{21}} \left(\begin{bmatrix}\times&\times&\times&\times\\ \times&\times&\times&\times\\ &\times&\times&\times\\ &&\times&\times\\ &&&\times\end{bmatrix}, \begin{bmatrix}\times&\times&\times&\times\\ &\times&\times&\times\\ &&\times&\times\\ &&\times&\times\\ &&&\times\end{bmatrix}\right).$$

The process analogously continues until $\underline{K_m}$ becomes upper triangular, as formalised in Algorithm 5.10. The algorithm is stated for RADs and quasi-RADs together. The difference for quasi-RADs is that for a 2-by-2 block on the subdiagonal we need to use successively two Givens rotations of the form $Q_{\ell,\ell-1}$, and two of the form $Z_{k,\ell-1}$, each acting on different planes. Furthermore, as a quasi-RAD is real-valued, we can chose both $Q_{\ell,\ell-1}$ and $Z_{k,\ell-1}$ to be real-valued as well. The process of transforming a quasi-RAD into a polynomial RAD is illustrated in Figure 5.2.

Remark 5.6. Algorithm 5.10 can be used to move all the poles of a (quasi-)RAD also

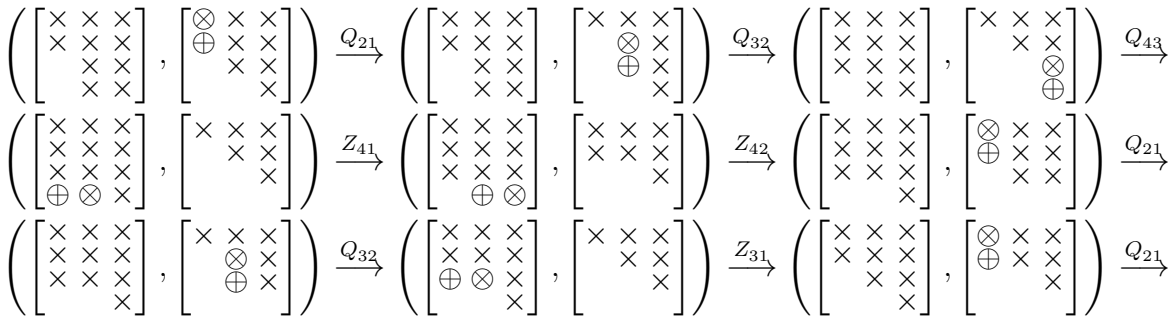


Figure 5.2: Transforming a quasi-RAD of order 3, with one real-valued and a pair of complex-conjugate poles, into a polynomial RAD. The element marked with $\oplus$ on the left-hand side of $\xrightarrow{G}$ is annihilated with the element marked $\otimes$ using the Givens rotation G, and the resulting nonzero structure is shown on the right-hand side of $\xrightarrow{G}$. In the end we obtain the desired structure $\left(\begin{bmatrix}\times&\times&\times\\ \times&\times&\times\\ &\times&\times\\ &&\times\end{bmatrix}, \begin{bmatrix}\times&\times&\times\\ &\times&\times\\ &&\times\end{bmatrix}\right)$.

to any another finite pole $\mu/\nu \notin \Lambda(A)$. Eq. (2.41) with $B = I$ reads

$$(\rho A - \eta I)V_{m+1}(\nu \underline{H_m} - \mu \underline{K_m}) = (\nu A - \mu I)V_{m+1}(\rho \underline{H_m} - \eta \underline{K_m}). \tag{5.18}$$

If we apply Algorithm 5.10 to the RAD (5.18), transforming $(\rho \underline{H_m} - \eta \underline{K_m}, \nu \underline{H_m} - \mu \underline{K_m})$ into $(\underline{\widehat{H}_m}, \underline{\widehat{K}_m})$, and V_{m+1} into $\widehat{V}_{m+1}$, we obtain $(\rho A - \eta I)\widehat{V}_{m+1}\underline{\widehat{K}_m} = (\nu A - \mu I)\widehat{V}_{m+1}\underline{\widehat{H}_m}$, where $\underline{\widehat{K}_m}$ is upper triangular. Rearranging the terms with and without A we have

$$A\widehat{V}_{m+1}(\rho \underline{\widehat{K}_m} - \nu \underline{\widehat{H}_m}) = \widehat{V}_{m+1}(\eta \underline{\widehat{K}_m} - \mu \underline{\widehat{H}_m}), \tag{5.19}$$

with $(\eta \widehat{k}_{j+1,j} - \mu \widehat{h}_{j+1,j})/(\rho \widehat{k}_{j+1,j} - \nu \widehat{h}_{j+1,j}) = \mu/\nu$, for $j = 1, 2, \ldots, m$, as desired. For quasi-RADs (3.1), the scalars μ and ν should be real-valued in order to preserve the structure. Furthermore, (5.18) may still fail to be a quasi-RAD, as $\nu \underline{H_m} - \mu \underline{K_m}$ may have 2-by-2 blocks on the subdiagonal. The quasi-RAD structure can, however, be recovered by Algorithm 5.6. Afterwards, Algorithm 5.10 can be used as explained.

5.3 RKToolbox corner

RKToolbox Example 5.1 is devoted to Algorithm 5.7, Theorem 2.12, and Theorem 5.4. An RAD (2.6) is constructed in line 2. The poles of the RAD are the ratios of the subdiagonal elements of the pencil $(\underline{H_m}, \underline{K_m})$, and this is verified in line 3. Next, in line 5, we change the starting vector $\boldsymbol{b}$ with $\boldsymbol{v}_3 = V_5 \boldsymbol{e}_3$. Theorem 2.12 asserts $\boldsymbol{v}_3 = p_2(A)q_2(A)^{-1}\boldsymbol{b}$, where the roots of p_2 are the generalised eigenvalues of (H_2, K_2).

```
1 A = gallery('tridiag', 100); b = ones(100, 1); xi = -(1:4);
2 [V, K, H] = rat_krylov (A, b, xi);
3 poles = diag(H, -1)./diag(K, -1); disp(poles.')
4
5 [Khat, Hhat] = move_poles_impl(K, H, [0 0 1 0 0]');
6 disp(eig(H(1:2, 1:2), K(1:2, 1:2)).')
7 disp(util_pencil_poles(Khat, Hhat))
```

```
3     -1     -2     -3     -4
6     0.0058    1.0217
7     0.0058    1.0217   -3.0000   -4.0000
```

RKToolbox Example 5.1: Moving poles implicitely and roots of orthogonal rational functions.

```
1 A = gallery('tridiag', 27) ; b = eye(27, 1); xi = -(1:3);
2 [V, K, H] = rat_krylov(A, b, xi);
3
4 [K, H, Q] = move_poles_expl(K, H, [8, 10, 1989]); V = V*Q';
5 xi_hat = util_pencil_poles(K, H); disp(xi_hat)
6
7 V_hat = rat_krylov(A, V(:, 1), xi_hat);
8 disp(V_hat'*V) % Should be unitary and diagonal.
```

```
5   8.0000e+00   1.0000e+01   1.9890e+03
8   1.0000e+00   1.8175e-17   1.0630e-16   4.7399e-17
8   1.0479e-16  -1.0000e+00   2.7472e-16  -1.2991e-16
8  -5.8781e-17   3.0426e-17  -1.0000e+00   3.9259e-16
8   2.7751e-17  -1.6025e-16   4.4440e-16   1.0000e+00
```

RKToolbox Example 5.2: Moving poles explicitly (to my birth date).

According to Theorem 5.4, by replacing the starting vector $\boldsymbol{b}$ with $\boldsymbol{v}_3$, the first two poles are replaces with the roots of p_2 (while the last two remain unchanged). This is verified by lines 6–7. On line 7 we use the function `util_pencil_poles` provided by the RKToolbox to check the new poles. Alternatively, we could look at the ratios of the subdiagonal elements of $(\underline{\widehat{H}_m}, \underline{\widehat{K}_m})$. The advantage of `util_pencil_poles` is that it works for quasi-RADs as well (see Figure 2.2).

In line 5 of RKToolbox Example 5.2 we show the usage of `move_poles_expl`. One can notice that, contrary to Algorithm 5.9, the basis V_{m+1} is not transformed by the routine to $\widehat{V}_{m+1}$, but one can do it afterwards if required as the unitary matrix $Q \in \mathbb{C}^{m+1,m+1}$ is returned, as is $Z \in \mathbb{C}^{m,m}$. The same applies to `move_poles_impl`. According to the rational implicit Q theorem, the obtained RAD is essentially equal to the RAD obtained by running the rational Arnoldi algorithm with the new starting

```
1 A = diag(1:100); b = ones(100, 1);
2 [V, K, H] = rat_krylov(A, b, [19+9i 19-9i 2016], 'real');
3 disp(H)
4
5 [K, H] = util_hh2th(K, H);
6 disp(H)
```

```
3    14.0515   20.9495   46.2427
3    14.7806    4.2148   17.2506
3    -5.3519   11.2984  -33.0145
3          0         0   28.5537
6    17.0348   -6.2405   -0.1937
6     6.4442  -21.4397  -22.0285
6          0  -14.7299  -55.3649
6          0         0   27.7892
```

RKToolbox Example 5.3: Moving poles implicitly to infinity.

vector, and the new poles. This is in part verified by lines 7–8.

Finally, in RKToolbox Example 5.3, line 5, we demonstrate the usage of `util_hh2th` on a small example. The upper quasi-Hessenberg structure of `H` before the call to `util_hh2th`, and its transformed, upper Hessenberg, structure afterwards, are displayed.

6 Rational Krylov fitting

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 [3, 34, 37, 50, 54]. Some other areas where rational approximants play an important role are analogue filter design [15], time-stepping methods [112], transparent boundary conditions [65], and iterative methods in numerical linear algebra; see, e.g., [33, 56, 79, 80, 106]. 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

$$\sum_{i=1}^{N} w_i |f_i - r(\lambda_i)|^2 \to \min. \tag{6.1}$$

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 (6.1), rational LS problems are challenging. Finding a rational function $r = p_m/q_m$ in (6.1) 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 not exist. An illustrating example inspired by

Braess [18, p. 109] is to fix $m \geq 1$ and $N > 2m$, and let

$$\lambda_i = \frac{i-1}{N}, \quad \text{and} \quad f_i = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{if } 2 \leq i \leq N. \end{cases} \tag{6.2}$$

Then the sequence of rational functions $r_j(z) = 1/(1 + jz)$ makes the misfit for (6.1) arbitrarily small as $j \to \infty$, but the f_i's 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* an LS solution to (6.1).

A very common approach for solving (6.1) approximately is linearisation. Consider again the data (6.2) and the problem of finding polynomials p_m and q_m of degree at most m such that

$$\sum_{i=1}^{N} w_i |f_i q_m(\lambda_i) - p_m(\lambda_i)|^2 \to \min. \tag{6.3}$$

This problem has a trivial solution with $q_m \equiv 0$, which we exclude by imposing, for instance, the "point-wise" normalisation condition $q_m(0) = 1$. Under this assumption, the linear problem (6.3) 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 (6.3) 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 (6.3) may yield an arbitrarily large (or even infinite) misfit for the nonlinear problem (6.1).

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 [4]. Robust solution methods for the linearised problem using regularised SVD are discussed in [43, 44].

The aim of this chapter is to present and analyse *Rational Krylov Fitting* (*RKFIT*), an iterative algorithm for solving rational LS problems more general than (6.1). For given matrices $A, F \in \mathbb{C}^{N,N}$, a vector $\boldsymbol{b} \in \mathbb{C}^N$, and $k \in \mathbb{Z}$ such that $k \geq -m$, RKFIT

attempts to find a rational function r of type $(m+k, m)$, such that the *relative misfit*

$$\text{misfit} = \frac{\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2}{\|F\boldsymbol{b}\|_2} \to \min, \tag{6.4}$$

is minimal. Note that this problem contains (6.1) as a special case if $A = \text{diag}(\lambda_i)$, $F = \text{diag}(f_i)$, $\boldsymbol{b} = \begin{bmatrix}\sqrt{w_1} & \sqrt{w_2} & \dots & \sqrt{w_N}\end{bmatrix}$, and $k = 0$. For RKFIT, however, the matrices A and F are not required to be diagonal. The matrix F may, for instance, be a function of the matrix A, i.e., $F = f(A)$. The benefit in obtaining the rational approximation r, for instance, is that it can thereafter be evaluated for different arguments A; see Section 6.2 and Section 6.6.

In Section 6.1 we show how rational Krylov techniques can be used to tackle problems of the form (6.4) by introducing the RKFIT algorithm. In Section 6.2 we report and discuss a few numerical experiments. Section 6.3 relates RKFIT to other rational approximation algorithms, in particular to the popular *vector fitting* algorithm [54, 52]. For simplicity, this discussion is concentrated to scalar rational approximations problems like (6.1). We continue with Section 6.4 where we propose an automated procedure for decreasing the degree parameters m and k, thereby reducing possible deficiencies in the rational approximants.

In Section 6.5 we extend RKFIT in order to incorporate multiple matrices $F^{[j]}$, as well as additional weighting. Specifically, for a given matrix $A \in \mathbb{C}^{N,N}$, two families of matrices $\{F^{[j]}\}_{j=1}^{\ell} \subset \mathbb{C}^{N,N}$ and $\{D^{[j]}\}_{j=1}^{\ell} \subset \mathbb{C}^{N,N}$, a vector $\boldsymbol{b} \in \mathbb{C}^N$, and $k \in \mathbb{Z}$ such that $k \geq -m$, we now want to find a family of rational functions $\{r^{[j]}\}_{j=1}^{\ell} \subset \mathcal{P}_{m+k}/q_m$, all sharing a common denominator $q_m \in \mathcal{P}_m$, such that the relative misfit

$$\text{misfit} = \sqrt{\frac{\sum_{j=1}^{\ell} \|D^{[j]}[F^{[j]}\boldsymbol{b} - r^{[j]}(A)\boldsymbol{b}]\|_F^2}{\sum_{j=1}^{\ell} \|D^{[j]}F^{[j]}\boldsymbol{b}\|_F^2}} \to \min, \tag{6.5}$$

is minimal. This problem contains (6.4) as a special case if $\ell = 1$ and $D^{[1]} = I$. The section is concluded with the pseudocode of the complete algorithm. Numerical examples for the case $\ell > 1$ are given in Section 6.6. In the first example we consider the fitting of a multiple-input and multiple-output (MIMO) dynamical system and in the second we propose a new pole optimization approach for exponential integration. Finally, Section 6.7 details the usage of the `rkfit` implementation within the RKToolbox.

Algorithm 6.11 High-level description of 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 $\mathcal{T} := \mathcal{K}_{m+k+1}(A, q_m(A)^{-1}\boldsymbol{b})$. ▷ See Section 6.1.1.
5. Find $\boldsymbol{v} = \operatorname{argmin}_{\breve{\boldsymbol{v}} \in \mathcal{S}, \|\breve{\boldsymbol{v}}\|_2=1} \| (I - P_{\mathcal{T}}) F \breve{\boldsymbol{v}} \|_2$. ▷ See Section 6.1.2.
6. Let $\breve{q}_m \in \mathcal{P}_m$ be such that $\boldsymbol{v} = \breve{q}_m(A) q_m(A)^{-1}\boldsymbol{b}$. ▷ See Section 6.1.2.
7. Set $q_m := \breve{q}_m$. ▷ See Corollary 6.2.
8. **until** stopping criteria is satisfied. ▷ See Sections 6.1.2–6.1.3.
9. Construct wanted approximant r. ▷ See Section 6.1.3.

6.1 The RKFIT algorithm

The RKFIT algorithm aims at finding a rational function $r = p_{m+k}/q_m$ of type $(m+k, m)$, solving (6.4). Since the denominator q_m is not known and hence (6.4) 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 6.11. In the description we use 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. Simply put, in the search space we look for poles that would provide a better LS approximation to $F\boldsymbol{b}$ from the target space. By $P_{\mathcal{T}}$ we denote the orthogonal projection onto $\mathcal{T}$. The essence of Algorithm 6.11 is the relocation of poles in lines 5–7. Since to any polynomial $\breve{q}_m \in \mathcal{P}_m$ we can associate a vector $\boldsymbol{v} = \breve{q}_m(A) q_m(A)^{-1}\boldsymbol{b} \in \mathcal{S}$, and vice versa, we may identify $\breve{q}_m$, the improvement of q_m, by looking for the corresponding vector $\boldsymbol{v} \in \mathcal{S}$. This is indeed done in line 5 and further explained in Section 6.1.2. Corollary 6.2, a consequence of the following Theorem 6.1, provides insight into the RKFIT pole relocation, i.e., lines 5–7 in Algorithm 6.11.

Theorem 6.1. *Let $q_m, q_m^\star \in \mathcal{P}_m$ be nonzero polynomials with roots disjoint from the spectrum of $A \in \mathbb{C}^{N,N}$. Fix $-m \le k \in \mathbb{Z}$, and let $\boldsymbol{b} \in \mathbb{C}^N$ be such that $2m+k < d(A, \boldsymbol{b})$. 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 6.11, respectively, and let V_{m+1} be an orthonormal basis of $\mathcal{S}$. Then, 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 $p_{m+k}^\star/q_m^\star$ is of type $(m+k-\Delta m, m-\Delta m)$.*

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

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

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

$$\mathcal{N} := \left\{ p_{\Delta m}(A)q^\star_{m-\Delta m}(A)q_m(A)^{-1}\boldsymbol{b} \,\middle|\, p_{\Delta m} \in \mathcal{P}_{\Delta m} \right\} \subseteq \mathcal{S}, \tag{6.6}$$

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

Corollary 6.2. *Let $q_m, q^\star_m, F, A, \boldsymbol{b}, m, k, \mathcal{S}$, and $\mathcal{T}$ be as in Theorem 6.1. Then $p^\star_{m+k}$ and $q^\star_m$ are coprime and either $\deg(p^\star_{m+k}) = m+k$ or $\deg(q^\star_m) = m$ if and only if there is a unique, up to unimodular scaling, solution to $\| (I - P_{\mathcal{T}})\, F\breve{\boldsymbol{v}}\|_2^2 \to \min$, over all $\breve{\boldsymbol{v}} \in \mathcal{S}$ of unit 2-norm. This solution is given by $\boldsymbol{v} = \gamma q^\star_m(A)q_m(A)^{-1}\boldsymbol{b}$ with some scaling factor $\gamma \in \mathbb{C}$.*

The corollary asserts that if $F\boldsymbol{b} = p_{m+k}(A)q^\star_m(A)^{-1}\boldsymbol{b}$ and $\Delta m = 0$, then the roots of $\boldsymbol{v} = \gamma q^\star_m(A)q_m(A)^{-1}\boldsymbol{b}$ match the unknown poles $q^\star_m$ and the next approximate poles become $q_m := q^\star_m$. 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 6.1 we develop in Section 6.4 an automatic procedure to reduce the denominator degree m by Δm, and adapt k. When dealing with noisy data (and roundoff), or when $F\boldsymbol{b}$ cannot be exactly represented as $r(A)\boldsymbol{b}$ for a rational function r of type $(m+k, m)$, the convergence of RKFIT is not yet clear. In the remaining part of this section, we show how the various parts of Algorithm 6.11 can be realized using rational Krylov techniques.

6.1.1. Constructing the target space. We assume that an orthonormal RAD

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m} \tag{6.7}$$

for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ has been constructed, for instance, by means of Algorithm 2.2. 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., the case $k > 0$. In this case $\mathcal{T} = \mathcal{Q}_{m+k+1}(A, \boldsymbol{b}, q_m)$, 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 $\mathcal{Q}_{m+k+1}(A, \boldsymbol{b}, q_m)$, we expand (6.7) to

$$A\widehat{V}_{m+k+1}\underline{\widehat{K}_{m+k}} = \widehat{V}_{m+k+1}\underline{\widehat{H}_{m+k}} \tag{6.8}$$

by performing k additional polynomial steps with the rational Arnoldi algorithm. Thus, we have $P_{\mathcal{T}} = \widehat{V}_{m+k+1}\widehat{V}_{m+k+1}^*$. We shall use this notation even if $k = 0$, i.e., if $k = 0$, then (6.8) coincides with (6.7), so that $\widehat{V}_{m+1}$ is defined.

We now consider the subdiagonal case $k < 0$. The target space $\mathcal{T}$ is given by $\mathcal{T} = \mathcal{K}_{m+k+1}(A, q_m(A)^{-1}\boldsymbol{b})$. Recall that $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m) = \mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b})$, and (6.7) can be transformed into a polynomial RAD

$$A\widehat{V}_{m+1}\underline{\widehat{K}_{m}} = \widehat{V}_{m+1}\underline{\widehat{H}_{m}}, \tag{6.9}$$

for $\mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{b})$ by Algorithm 5.10. An orthonormal basis for $\mathcal{T}$ is then given by truncating $\widehat{V}_{m+1}$ to $\widehat{V}_{m+k+1}$, the first $m + k + 1$ columns of $\widehat{V}_{m+1}$.

6.1.2. Solving the linear problem and relocating poles. We consider the problem in line 5 of Algorithm 6.11, i.e., the problem of finding a unit 2-norm vector $\boldsymbol{v} \in \mathcal{S}$ such that $\|(I - P_{\mathcal{T}})F\boldsymbol{v}\|_2^2$ is minimal. We have $\mathcal{S} = \mathcal{R}(V_{m+1})$ and $\mathcal{T} = \mathcal{R}(\widehat{V}_{m+k+1})$, with both V_{m+1} and $\widehat{V}_{m+k+1}$ being orthonormal. Defining the matrix

$$S = FV_{m+1} - \widehat{V}_{m+k+1}\big(\widehat{V}_{m+k+1}^* FV_{m+1}\big) \in \mathbb{C}^{N,m+1}, \tag{6.10}$$

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

We now discuss how to determine the polynomial $\breve{q}_m \in \mathcal{P}_m$, from line 6 of Algorithm 6.11, such that $\boldsymbol{v} = V_{m+1}\widehat{\boldsymbol{c}} = \breve{q}_m(A)q_m(A)^{-1}\boldsymbol{b}$. Let $Q_{m+1} \in \mathbb{C}^{m+1,m+1}$ be unitary with first column $Q_{m+1}\boldsymbol{e}_1 = \widehat{\boldsymbol{c}}$. Using (6.7) as an RAD for $\mathcal{S}$, it follows from

Theorem 5.4 that the roots of $\breve{q}_m$ are the eigenvalues of the m-by-m pencil

$$\left(\begin{bmatrix}\mathbf{0} & I_m\end{bmatrix} Q_{m+1}^* \underline{H_m}, \begin{bmatrix}\mathbf{0} & I_m\end{bmatrix} Q_{m+1}^* \underline{K_m}\right). \tag{6.11}$$

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

6.1.3. Constructing the approximants. 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 (6.8) is an RAD for $\mathcal{T}$, then the approximant r is represented by a coefficient vector $\boldsymbol{c} \in \mathbb{C}^{m+k+1}$ such that $r(A)\boldsymbol{b} = \|\boldsymbol{b}\|_2 \widehat{V}_{m+k+1}\boldsymbol{c}$. The coefficient vector is given by

$$\boldsymbol{c} = \widehat{V}_{m+k+1}^*\big(F\boldsymbol{b}\big)/\|\boldsymbol{b}\|_2. \tag{6.12}$$

Computing the coefficient vector $\boldsymbol{c}$ at each iteration does not significantly increase the computational complexity because the vector $F\boldsymbol{b}$ needs to be computed only once. The vector $\boldsymbol{c}$ enables the cheap evaluation of the relative misfit (6.4), which allows to stop the RKFIT iteration when a desired tolerance $\varepsilon_{\texttt{tol}}$ is achieved.

6.1.4. Avoiding complex arithmetic. If F, A, and $\boldsymbol{b}$ are real-valued and the set of starting poles $\{\xi_j\}_{j=1}^m$ is closed under complex conjugation, we can use Algorithm 2.3 instead of Algorithm 2.2 and work with quasi-RADs instead of RADs. The matrix S in (6.10) is guaranteed to be real-valued and the generalized eigenproblem (6.11) is real-valued as well. This guarantees that the relocated poles appear in complex-conjugate pairs as well. If F, A, and $\boldsymbol{b}$ 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.

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 $(\overline{\lambda}, \overline{f})$. Without loss of generality, we assume that the pairs are ordered so 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 $0 \le N_\mathbb{R} \le N$ is a natural number. Define $A = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N), F = \operatorname{diag}(f_1, f_2, \ldots, f_N)$, $\boldsymbol{b} = [1 \quad 1 \quad \ldots \quad 1]^T \in \mathbb{R}^N$,

and let $Q \in \mathbb{C}^{N,N}$ be unitary. Then

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

The first equality follows from the unitary invariance of the 2-norm, and the second from [60, Theorem 1.13]. With the particular choice

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

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

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

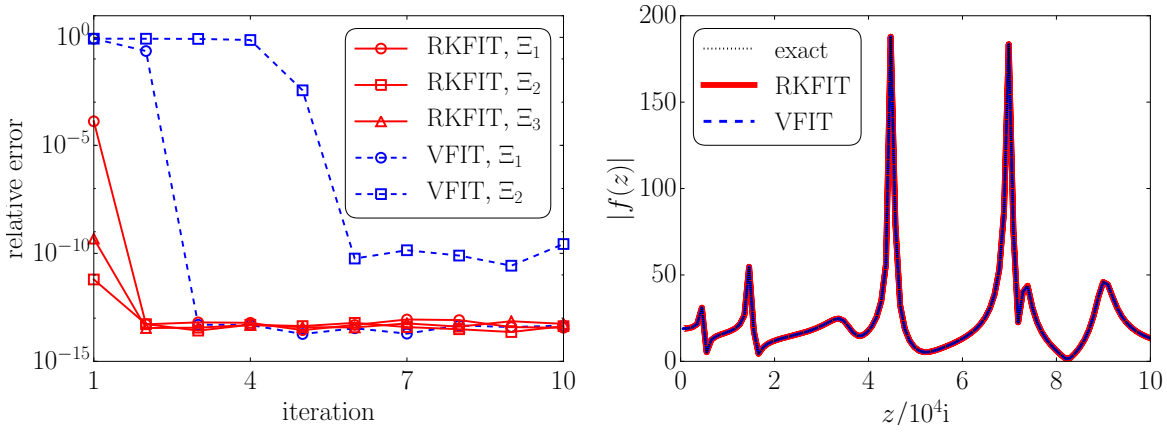
where $N_{\mathbb{C}} = \frac{N - N_{\mathbb{R}}}{2}$ and $i_j = N_{\mathbb{R}} + 2j - 1$. For $A_{\mathbb{R}}$ we obtain an analogous expression, while $\boldsymbol{b}_{\mathbb{R}} = [1 \ \ldots \ 1 \ \sqrt{2} \ \ 0 \ \ldots \ \sqrt{2} \ \ 0]^T$, with $N_{\mathbb{R}}$ entries equal to 1, and $N_{\mathbb{C}}$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

6.2 Numerical experiments (with $\ell = 1$)

Before studying and extending RKFIT further, we provide a some comments relating the RKFIT approximation to those considered in Chapter 3, and then examine a few examples.

In general, RKFIT approximations differ from those considered in Chapter 3. To see this, let us considered the simplest case of $F = A^{-1}$ with $m = k = 1$. Then, by Corollary 6.2, RKFIT finds the minimizer after one reallocation independently from the used starting pole. On the other hand, rational Arnoldi approximations are based on the interpolation on Ritz values and thus the rational Arnoldi approximant may be incorrect even if the solution $A^{-1}\boldsymbol{b}$ is contained in the corresponding rational Krylov space. The main reason for this difference is that RKFIT uses instead the LS approximation from the rational Krylov space. This, however, makes the extrapolation step more costly. We now consider three numerical examples.

6.2.1. Experiment 1: Fitting an artificial frequency response. We first consider a diagonal matrix $A \in \mathbb{C}^{N,N}$ with $N = 200$ linearly spaced eigenvalues in the



(a) Convergence of RKFIT and VFIT.

(b) Magnitude of the frequency responses.

Figure 6.1: Least-squares approximation of a rational function f of type $(19, 18)$ using RKFIT and the vector fitting code VFIT. Left: Relative error $\|f(A)\boldsymbol{b} - r(A)\boldsymbol{b}\|_2/\|f(A)\boldsymbol{b}\|_2$ after each iteration of RKFIT (solid red) and VFIT (dashed blue). The two convergence curves for each method correspond to different choices for the initial poles Ξ_1 (circles), Ξ_2 (squares), and Ξ_3 (triangles), respectively. Right: Plot of $|f|$ over an interval on the imaginary axis overlaid with the approximants $|r|$ obtained after 10 iterations of RKFIT and VFIT with initial poles Ξ_1 (the curves are visually indistinguishable).

interval $[10^{-5}\mathrm{i}, 10^{5}\mathrm{i}]$. The matrix $F = f(A)$ is a rational matrix function of type $(19, 18)$ given in partial fraction form in [54, Section 4.1], and $\boldsymbol{b} = [1 \quad 1 \quad \ldots \quad 1]^T \in \mathbb{R}^N$. We compare RKFIT to the vector fitting algorithm (VFIT) [54, 52] which is available online.[1] We review VFIT and relate it to RKFIT in the following Section 6.3. In this example we use $k = -1$, and we consider three different sets of starting poles, namely

- Ξ_1: 9 log-spaced poles in $[10^{3}\mathrm{i}, 10^{5}\mathrm{i}]$ and their complex conjugates;
- Ξ_2: 12 log-spaced poles in $[10^{6}\mathrm{i}, 10^{9}\mathrm{i}]$ and their complex conjugates;
- Ξ_3: 18 infinite poles (applicable to RKFIT only);

and run 10 iterations of RKFIT and VFIT, respectively.

The numerical results are shown in Figure 6.1. Figure 6.1(a) shows the relative error $\|f(A)\boldsymbol{b} - r(A)\boldsymbol{b}\|_2/\|f(A)\boldsymbol{b}\|_2$ after each iteration. We observe that RKFIT converges within the first 2 iterations for all three sets of initial poles Ξ_1, Ξ_2, and Ξ_3. VFIT requires 3 iterations starting with Ξ_1 and it fails to converge within 10 iterations when being initialised with the poles Ξ_2. In the later case, the warnings that MATLAB raises about solves of close-to-singular linear systems seem to indicate that VFIT relies on ill-conditioned linear algebra. The choice of infinite initial poles Ξ_3 is interesting in that

[1] http://www.sintef.no/Projectweb/VECTFIT/Downloads/VFUT3/ as of November 2014.

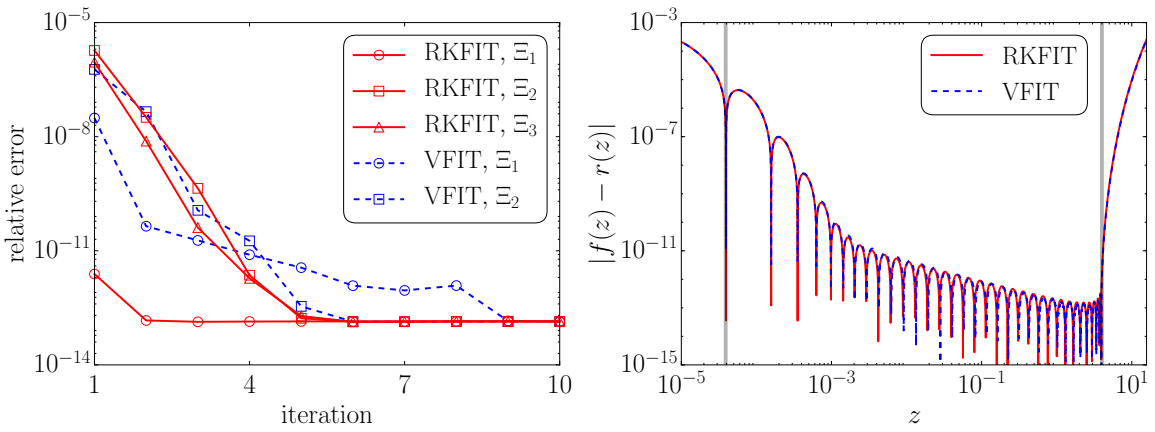
it requires no a priori knowledge of the pole location (choosing all poles to be infinite is not possible in the available VFIT code). Figure 6.1(b) shows the plot of $|f(z)|$ over an interval on the imaginary axis, together with the RKFIT and VFIT approximants $|r(z)|$. The evaluation of the scalar function r may be based on Theorem 2.14. Indeed, the vector $\boldsymbol{c}$ from (6.12) collects the coefficients of the approximant $r(A)\boldsymbol{b}$ in the rational Krylov basis $\widehat{V}_{m+k+1}$, i.e., $r(A)\boldsymbol{b} = \|\boldsymbol{b}\|_2 \widehat{V}_{m+k+1}\boldsymbol{c}$. Using Theorem 2.14 we find that $r(z)$ can be evaluated for any point $z \in \mathbb{C}$, excluding the poles, by computing a full QR factorisation of $z\underline{K_m} - \underline{H_m}$ and forming an inner product of $\|\boldsymbol{b}\|_2\boldsymbol{c}$ with the last column $\boldsymbol{q}_{m+1}^{(z)}$ of the Q factor scaled by its first entry, i.e., $r(z) = \|\boldsymbol{b}\|_2 \frac{(\boldsymbol{q}_{m+1}^{(z)})^*\boldsymbol{c}}{(\boldsymbol{q}_{m+1}^{(z)})^*\boldsymbol{e}_1}$. (In Chapter 7 we introduce a more efficient evaluation algorithm.) Figure 6.1(b) essentially coincides with [54, Figure 1] (it does not exactly coincide as apparently the figure in that paper has been produced with a smaller number of sampling points, causing some "spikes" to be missed or reduced).

6.2.2. Experiment 2: Square root of a symmetric matrix. We consider the approximation of $F\boldsymbol{b}$ with the matrix square root $F = A^{1/2}$, $A = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{100,100}$, $\boldsymbol{b} = \boldsymbol{e}_1$, and $k = 0$. Again, we test different sets of initial poles, namely

- Ξ_1: 16 log-spaced poles in $[-10^8, -10^{-8}]$;
- Ξ_2: 16 linearly spaced poles in $[0, 4]$;
- Ξ_3: 16 infinite poles (applicable to RKFIT only).

Note that the initial poles Ξ_1 are placed on the branch cut of $z^{1/2}$, which is a reasonable initial guess for the poles of r. Some of the poles Ξ_2 are located very close to the eigenvalues of A whose spectral interval is approximately $[0, 4]$. The convergence, i.e, the relative error per iteration of RKFIT and VFIT is shown on Figure 6.2(a). In order to use VFIT for this problem, we have diagonalized A and provided the code with weights corresponding to the components of $\boldsymbol{b}$ in the eigenvector basis of A. All tests converge within at most 9 iterations, with the fastest convergence achieved by RKFIT with initial guess Ξ_1. In Figure 6.2(b) we show the error $|z^{1/2} - r(z)|$ over an interval containing the spectrum of A.

6.2.3. Experiment 3: Exponential of a nonnormal matrix. We consider the approximation of $F\boldsymbol{b}$ with the matrix exponential $F = \exp(A)$ of a Grcar-like matrix



(a) Convergence of RKFIT and VFIT.

(b) Absolute error of the approximation.

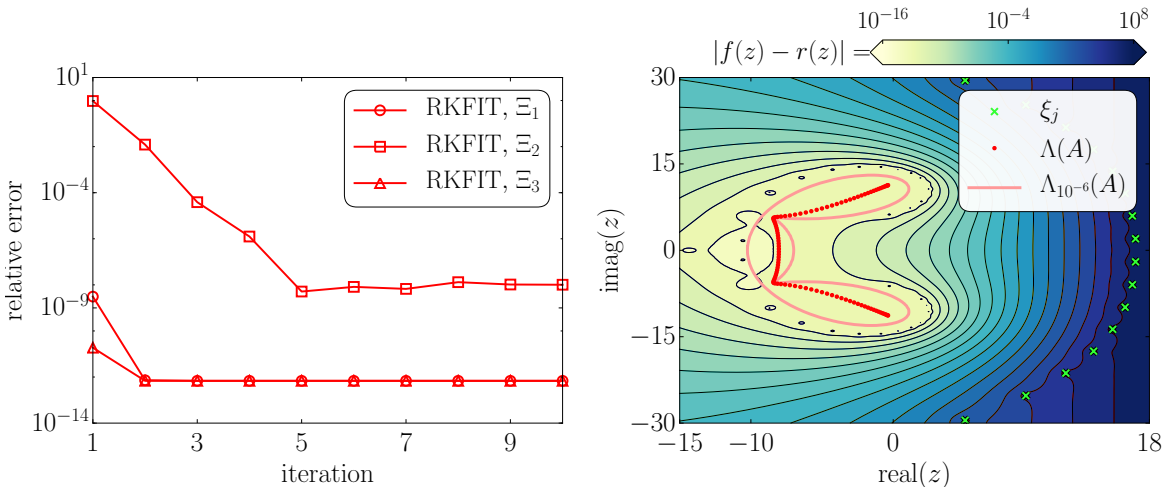
Figure 6.2: Least-squares approximation of the function $f(z) = z^{1/2}$ using RKFIT and the vector fitting code VFIT. Left: This plot shows the relative approximation error $\|f(A)\boldsymbol{b} - r(A)\boldsymbol{b}\|_2/\|f(A)\boldsymbol{b}\|_2$ after each iteration of RKFIT (solid red) and VFIT (dashed blue). The convergence curves for each method correspond to different choices for the initial poles Ξ_1 (circles), Ξ_2 (squares), and Ξ_3 (triangles), respectively. Right: This is the plot of $|f - r|$ over an interval on the positive real axis obtained after 10 iterations of RKFIT and VFIT with initial poles Ξ_1. The vertical lines indicate the spectral interval of A.

A of size $N = 100$ generated in MATLAB via `A = -5*gallery('grcar',N,3)`. The eigenvalues and the boundary of the 10^{-6}-pseudospectrum of A are shown on the right of Figure 6.3. The vector is $\boldsymbol{b} = [1 \quad 1 \quad \dots \quad 1]^T \in \mathbb{R}^{100}$ and we use $m = 16$ with $k = 0$. The different sets of initial poles for RKFIT we compare are:

- Ξ_1: 16 poles equal to 0;
- Ξ_2: 16 poles equal to -10;
- Ξ_3: 16 infinite poles.

Note that A is not diagonalizable and therefore VFIT cannot be applied as in the previous two experiments. In Figure 6.3(a) we observe excellent convergence of RKFIT within 2 iterations starting with the initial poles Ξ_1 and Ξ_3.

With the initial poles Ξ_2 the error stagnates on a higher level, possibly trapped nearby a non-global minimum. As is the case with any nonlinear iteration, RKFIT is not guaranteed to converge to a global minimum (even when it exists). We currently do not have a good explanation of why the initial guess Ξ_2 is bad, but we have verified that $\xi = -10$ lies in the 10^{-6}-pseudospectrum of A, and hence the initial rational Krylov space may have too large components in just a few eigendirections of A. In Figure 6.3(b) we have included a contour plot for the scalar error $|f - r|$ over a region in the complex



(a) Convergence of RKFIT.

(b) Absolute error of the approximation.

Figure 6.3: Least-squares approximation of the function $f(z) = \exp(z)$ using RKFIT. Left: This plot shows the relative approximation error $\|f(A)\boldsymbol{b} - r(A)\boldsymbol{b}\|_2/\|f(A)\boldsymbol{b}\|_2$ after each iteration of RKFIT (solid red) for different choices of initial poles Ξ_1, Ξ_2, and Ξ_3, respectively. Right: A plot of $|f - r|$ over a region in the complex plane together with the poles of r (green crosses), where r is the rational least squares approximant obtained after 10 iterations of RKFIT with initial poles Ξ_1. The eigenvalues of the Grcar matrix and the boundary of the 10^{-6}-pseudospectrum are also shown.

plane together with the poles of r, and the aforementioned pseudospectrum.

6.3 Other rational approximation algorithms

Here we consider scalar rational approximation problems, like the one encountered in the introduction of the chapter. In our discussion we refrain from using weights, and fix the type of the rational approximant to be $(m-1, m)$, 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

$$\sum_{i=1}^{N} |f_i - r(\lambda_i)|^2 \to \min. \tag{6.13}$$

A popular approach, introduced in [54], for solving problems of this form and designed to fit frequency response measurements of dynamical systems is *vector fitting* (VFIT).

As already observed in Section 6.2, numerical experiments indicate that RKFIT performs more robustly than VFIT. The main goal of this section is to clarify the differences and similarities between the two methods. In Section 6.3.1, we briefly

review the predecessors of VFIT, followed by a derivation of VFIT in Section 6.3.2. In Section 6.3.3 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 6.3.4.

6.3.1. Iteratively reweighted linearisation. The first attempt to solve the nonlinear problem (6.13) was through linearisation [74]. 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 [74] to replace (6.13) 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 (6.13).

As a remedy, Sanathanan and Koerner [96] suggest to replace the nonlinear problem (6.13) 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 $\breve{q}_m := q_m$ and solve a reweighted linear problem $\sum_{i=1}^{N} \frac{|f_i q_m(\lambda_i) - p_{m-1}(\lambda_i)|^2}{|\breve{q}_m(\lambda_i)|^2} \to \min$. This process can be iterated until a satisfactory approximation has been obtained or a maximal number of iterations has been performed.

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.

6.3.2. 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,

$$r(z) = \frac{p_{m-1}(z)}{q_m(z)} = \frac{p_{m-1}/\breve{q}_m(z)}{q_m(z)/\breve{q}_m(z)} = \frac{\sum_{j=1}^{m} \frac{\varphi_j}{z-\xi_j}}{1 + \sum_{j=1}^{m} \frac{\psi_j}{z-\xi_j}}. \tag{6.14}$$

Algorithm 6.12 Vector fitting [54].

Input: Interpolation nodes $\{\lambda_i\}_{i=1}^N$ and data $\{f_i\}_{i=1}^N$, and $m \leq N$.
Output: Rational function r of type $(m-1, m)$ such that $r(\lambda_i) \approx f_i$, for $i = 1, 2, \ldots, N$.

1. Select pairwise distinct finite $\{\xi_j\}_{j=1}^m \cap \{\lambda_i\}_{i=1}^N = \emptyset$.
2. **repeat**
3. Solve (6.16) for $\begin{bmatrix}\boldsymbol{\varphi}^T & \boldsymbol{\psi}^T\end{bmatrix}^T$.
4. Update $\{\xi_j\}_{j=1}^m := \Lambda(\operatorname{diag}(\xi_j) - \boldsymbol{\psi}\boldsymbol{e}^T)$.
5. **until** $\boldsymbol{\psi} \not\approx \mathbf{0}$ ▷ See Section 6.3.4.
6. Solve (6.16) for $\boldsymbol{\varphi}$ only, i.e., remove last m columns of the system matrix.
7. Set $r(z) = \sum_{j=1}^m \frac{\varphi_j}{z-\xi_j}$.

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 ultimately converge to the poles of an (approximate) minimizer r.

The linearised version of (6.14) reads

$$r(z)\left(1 + \sum_{j=1}^m \frac{\psi_j}{z-\xi_j}\right) = \sum_{j=1}^m \frac{\varphi_j}{z-\xi_j}. \tag{6.15}$$

Inserting $z = \lambda_i$ and replacing $r(\lambda_i)$ with f_i in (6.15) for $i = 1, 2, \ldots, N$ gives a linear system of equations

$$\begin{bmatrix} \frac{1}{\lambda_1-\xi_1} & \cdots & \frac{1}{\lambda_1-\xi_m} & \frac{-f_1}{\lambda_1-\xi_1} & \cdots & \frac{-f_1}{\lambda_1-\xi_m} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{1}{\lambda_N-\xi_1} & \cdots & \frac{1}{\lambda_N-\xi_m} & \frac{-f_N}{\lambda_N-\xi_1} & \cdots & \frac{-f_N}{\lambda_N-\xi_m} \end{bmatrix} \begin{bmatrix}\boldsymbol{\varphi} \\ \boldsymbol{\psi}\end{bmatrix} = \boldsymbol{f}, \tag{6.16}$$

which is solved in the LS sense. Afterwards, the poles $\{\xi_j\}_{j=1}^m$ are replaced by the roots of the denominator $1 + \sum_{j=1}^m \frac{\psi_j}{z-\xi_j}$, i.e., by the eigenvalues of $\operatorname{diag}(\xi_j) - \boldsymbol{\psi}\boldsymbol{e}^T$, where $\boldsymbol{e} = \begin{bmatrix}1 & 1 & \ldots & 1\end{bmatrix}^T \in \mathbb{C}^m$; see [20, 41]. It is assumed that those will again be pairwise distinct and disjoint from $\{\lambda_i\}_{i=1}^N$. Iterating this process gives the VFIT algorithm; see Algorithm 6.12. The reweighting as in the Sanathanan–Koerner algorithm is implicitly achieved in VFIT through the change of interpolation nodes for the barycentric representation.

6.3.3. 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 $\breve{C}_m = C_{m+1}\begin{bmatrix}\mathbf{0} & I_m\end{bmatrix}^T$. We now rewrite (6.16) in the equivalent form

$$\begin{bmatrix}\breve{C}_m & -FC_{m+1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\varphi} \\ \psi_0 \\ \boldsymbol{\psi}\end{bmatrix} = \mathbf{0}, \tag{6.17}$$

with $\psi_0 = 1$. For any fixed $\boldsymbol{\psi} \in \mathbb{C}^m$, solving (6.17) for $\boldsymbol{\varphi} \in \mathbb{C}^m$ in the LS sense, subject to $\psi_0 = 1$, is equivalent to solving $\breve{C}_m\boldsymbol{\varphi} = FC_{m+1}[1 \quad \boldsymbol{\psi}^T]^T$ in the LS sense. Under the (reasonable) assumption that $\breve{C}_m \in \mathbb{C}^{N,m}$ is of full column rank with $m \leq N$, the unique solution is given by $\boldsymbol{\varphi} = \breve{C}_m^{\dagger}FC_{m+1}[1 \quad \boldsymbol{\psi}^T]^T$.

Therefore, when solving (6.16) in VFIT one gets $r = \dfrac{\breve{p}_m/q_m}{\breve{q}_m/q_m}$, where $\breve{q}_m(z)/q_m(z) = 1 + \sum_{j=1}^m \frac{\psi_j}{z-\xi_j}$ and $\breve{p}_m(z)/q_m(z) = \sum_{j=1}^m \frac{\varphi_j}{z-\xi_j}$ is the projection of $f\breve{q}_m/q_m$, with f being defined on the discrete set of interpolation nodes as $f(\lambda_i) = f_i$, onto the target space, here represented by $\breve{C}_m$.

Both VFIT and RKFIT solve an 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 $\breve{q}/q$ 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 [52], 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 (6.16). This modification to a global normalization condition avoids the problems with point-wise normalization conditions exemplified in the introduction of the chapter. VFIT with this additional constrain is known as relaxed VFIT. The normalization condition in RKFIT is also of global nature, $\|\boldsymbol{v}\|_2 = \|\breve{q}(A)q(A)^{-1}\boldsymbol{b}\|_2 = 1$, cf. line 5 in Algorithm 6.11.

6.3.4. 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. (6.16). *Orthonormal vector fitting* was proposed as a remedy in [25], where the basis of partial fractions was replaced by an orthonormal basis. Soon after it was claimed [53] that a numerically more careful implementation of VFIT is as good as the orthonormal VFIT variant proposed in [25], and hence the orthonormal VFIT never became a reality.

The issue with the orthonormal VFIT [25] 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 [28, 29, 30]. Our approach avoids these problems altogether, and RKFIT is more general.

So far we have 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 the numerical properties as well as the approximation. This is suggested in [29, 30] for the discretized $\mathcal{H}_2$ approximation of transfer function measurements and carries over straightforwardly to RKFIT.

To date, there are no comprehensive convergence analyses for VFIT and RKFIT. In [72, Section IV] an example of degree $m = 2$ was constructed where the VFIT fixed-point is repellent and hence the iteration diverges, independently of the starting guess for the poles. Despite this one example, VFIT has been and is being successfully used by the engineering community for various (large scale) problems. Both VFIT and RKFIT have the property that if a rational function r is of sufficiently low degree and there are sufficiently many interpolation nodes, then in the absence of roundoff r is recovered exactly; see [72, Corollary III.1] and our Corollary 6.2.

6.4 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.

We assume to have an $(m+k, m)$ approximant r such that $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \leq \|F\boldsymbol{b}\|_2 \varepsilon_{\texttt{tol}}$, and 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 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 when F is a rational matrix function. An illustration is given in Figure 6.4. In Section 6.4.4 we discuss the case when F is not a rational matrix function.

6.4.1. Reducing the denominator degree m. Assume that F is a rational matrix function. Our reduction procedure for m is based on Theorem 6.1, 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 $\|F\boldsymbol{b}\|_2 \varepsilon_{\texttt{tol}}$ (with, e.g., $\varepsilon_{\texttt{tol}} = 10^{-15}$) is computed. More precisely, we reduce m by the largest integer $\Delta m \leq \min\{m, m+k\}$ such that

$$\sigma_{m+1-\Delta m} \leq \|F\boldsymbol{b}\|_2 \varepsilon_{\texttt{tol}}, \tag{6.18}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{m+1}$ are the singular values of S.

6.4.2. 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 6.1, 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 $(\underline{H_m}, \underline{K_m})$ in (6.7).

Theorem 6.3. *Let (2.6) be an RAD for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, with $m+1 < d(A, \boldsymbol{b})$, and let the $\widehat{r}_j \equiv V_{m+1}\widehat{\boldsymbol{c}}_j$ for $j = 1, 2, \ldots, \Delta m + 1$ be linearly independent. Assume that the numerators of $\widehat{r}_j$ share as GCD a polynomial of degree $m - \Delta m$ with no repeated roots. Let $X \in \mathbb{C}^{m+1,m+1}$ be a nonsingular matrix with $X\boldsymbol{e}_j = \widehat{\boldsymbol{c}}_j$ for $j = 1, 2, \ldots, \Delta m + 1$. Introduce*

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

Then the roots of the GCD are the eigenvalues of the $(m-\Delta m)$-by-$(m-\Delta m)$ generalized eigenproblem $(H_\star, K_\star)$.

Proof. We transform the RAD (6.7) into (6.9) where $\widehat{V}_{m+1} = q_m(A)V_{m+1}X$, $\underline{\widehat{K}_m} = X^{-1}\underline{K_m}$, and $\underline{\widehat{H}_m} = X^{-1}\underline{H_m}$. Hence, in scalar form (see Theorem 2.13) we have

$$z\boldsymbol{p}(z)\underline{\widehat{K}_m} = \boldsymbol{p}(z)\underline{\widehat{H}_m} \iff \boldsymbol{p}(z)\big(z\underline{\widehat{K}_m} - \underline{\widehat{H}_m}\big) = \boldsymbol{0}^T,$$

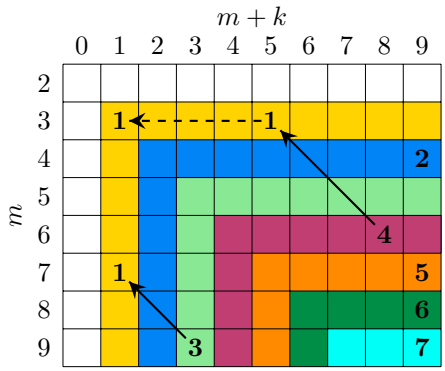
where $\boldsymbol{p}(z) = [p_1(z) \quad p_2(z) \quad \ldots \quad p_{m+1}(z)]$ with, formally, $p_j = q_m\widehat{r}_j \in \mathcal{P}_m$. Introduce $\widehat{K}_\star$ and $\widehat{H}_\star$ as the lower-right $(m - \Delta m)$-by-$(m - \Delta m)$ submatrices of $\underline{\widehat{K}_m}$ and $\underline{\widehat{H}_m}$, respectively. Since $(H_\star, K_\star) = (\widehat{H}_\star, \widehat{K}_\star)$, we need to show that the roots of the GCD are $\Lambda(\widehat{H}_\star, \widehat{K}_\star)$.

Let λ be a common root of $\{p_j\}_{j=1}^{\Delta m+1}$. Note that this is then also a common root of $\{r_j\}_{j=1}^{\Delta m+1}$. Then the last $m - \Delta m$ columns of $\boldsymbol{p}(\lambda)(\lambda\underline{\widehat{K}_m} - \underline{\widehat{H}_m}) = \boldsymbol{0}^T$ assert that λ is a generalized eigenvalue of $(\widehat{H}_\star, \widehat{K}_\star)$ with left eigenvector $\boldsymbol{p}_\star(\lambda)^* = [p_{\Delta m+2}(\lambda) \quad \ldots \quad p_{m+1}(\lambda)]^* \neq \boldsymbol{0}$. ∎

Remark 6.4. We conjecture that Theorem 6.3 holds also if the GCD has repeated roots. This is proved in [11, Theorem 4.1] under the additional assumption that $K_\star$ is nonsingular (which is conjectured to be superfluous [11, Remark 4.2]).

6.4.3. 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 $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \leq \|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}}$ has been found. Reducing the numerator degree is a linear problem and we can guarantee the misfit to stay below $\varepsilon_{\texttt{tol}}$ 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 $\widehat{V}_j$ be an orthonormal basis for $\mathcal{K}_j(A, q_m(A)^{-1}\boldsymbol{b})$ for $j = 1, 2, \ldots, m + k + 1$.



$$\underbrace{\begin{bmatrix} 2.5\times10^{-02} \\ -1.3\times10^{-03} \\ -4.6\times10^{-03} \\ 6.0\times10^{-04} \\ 4.5\times10^{-18} \\ 1.8\times10^{-18} \end{bmatrix}}_{\boldsymbol{c}_{\mathcal{Q}}} \quad \underbrace{\begin{bmatrix} 1.9\times10^{-02} \\ 1.7\times10^{-02} \\ -5.7\times10^{-17} \\ 1.5\times10^{-17} \\ 4.5\times10^{-18} \\ 1.8\times10^{-18} \end{bmatrix}}_{\boldsymbol{c}_{\mathcal{K}}}$$

Figure 6.4: Degree reduction when fitting $F\boldsymbol{b}$, where $F = A(A+I)^{-1}(A+3I)^{-2}$, $A = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N,N}$, 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 $(I - P_{\mathcal{T}})FV_{m+1}$ below the tolerance $\|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}} = 10^{-15}$, 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 $-3 \pm \text{i}2.32 \times 10^{-7}$. 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 $\boldsymbol{c}_{\mathcal{Q}}$ 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 6.4.3, reveals that r is of type $(1, 3)$. The coefficients $\boldsymbol{c}_{\mathcal{K}}$ in this polynomial basis are also given.

As the vectors in $\widehat{V}_j$ 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 $r(A)\boldsymbol{b}/\|\boldsymbol{b}\|_2 = \widehat{V}_{m+k+1}\boldsymbol{c}$.

Introduce $\boldsymbol{c}_{-i} = [O \quad I_i]\boldsymbol{c} \in \mathbb{C}^i$ for $i = 1, 2, \ldots, m + k$. By the triangle inequality,

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

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

$$\|\boldsymbol{c}_{-i}\|_2 \le \|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}} - \|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2, \tag{6.19}$$

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

6.4.4. General F. The following lemma extends Theorem 6.1 to the case when F is not necessarily a rational matrix function.

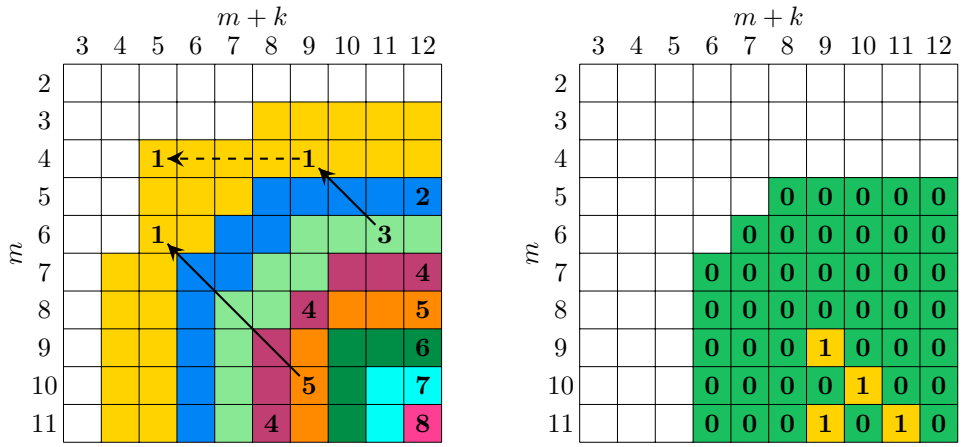


Figure 6.5: Degree reduction when fitting $F\boldsymbol{b}$, where $F = (A+A^2)^{\frac{1}{2}}$, $A = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N,N}$, and $\boldsymbol{b} = \boldsymbol{e}_1 \in \mathbb{R}^N$, with $N = 150$. The poles of the search space are obtained after three RKFIT iterations with initial poles all at infinity. The table on the left shows the number $\Delta m + 1$ of singular values of $(I - P_{\mathcal{T}})FV_{m+1}$ 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.

Lemma 6.5. *Let $q_m, A, \boldsymbol{b}, m, k, \mathcal{S}, \mathcal{T}$, and V_{m+1} be as in Theorem 6.1. Assume that for $F \in \mathbb{C}^{N,N}$ we have a rational approximant $r = p_{m+k}/q_m$ of type $(m+k, m)$ such that $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \le \|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}}$. If the matrix $(I - P_{\mathcal{T}})FV_{m+1}$ has $\Delta m + 1$ singular values smaller than $\|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}}$, then there exists a $(\Delta m + 1)$-dimensional subspace $\mathcal{N}_g \subseteq \mathcal{S}$, containing $\boldsymbol{b}$, such that*

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

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

Proof. Consider a thin SVD of the matrix $(I - P_{\mathcal{T}})FV_{m+1} = U\Sigma W^*$, where $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{m+1}) \in \mathbb{R}^{m+1,m+1}$ and $\sigma_{m+1} \le \sigma_m \le \ldots \le \sigma_{m-\Delta m} \le \|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}}$ 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 $\|F\boldsymbol{b} - r(A)\boldsymbol{b}\|_2 \le \|F\boldsymbol{b}\|_2\varepsilon_{\texttt{tol}}$ 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 6.5 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 (6.6), 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 $\boldsymbol{v} = p_m(A)q_m(A)^{-1}\boldsymbol{b} \in \mathcal{N}_g$ the vector $Fp_m(A)q_m(A)^{-1}\boldsymbol{b}$ 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 p_m corresponding to vectors $\boldsymbol{v} \in \mathcal{N}_g$ share an *approximate* GCD (see, e.g., [16]) 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 6.3.

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

$$\sigma_{m+1-\Delta m} \leq \|F\boldsymbol{b}\|_2 \varepsilon_{\texttt{tol}} \varepsilon_{\texttt{safe}}, \tag{6.20}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{m+1}$ are the singular values of S. By default we use $\varepsilon_{\texttt{safe}} = 0.1$. Figure 6.5 illustrates our reduction strategy for a non-rational function F. The table on the left shows the number $\Delta m + 1$ of singular values of $(I - P_{\mathcal{T}})FV_{m+1}$ below the threshold, for different choices of m and k. It can be observed that when F is not a rational matrix function the table loses the regular structure like the one in Figure 6.4.

6.5 Extensions and complete algorithm

In order to tackle the more general problem given by (6.5) we only need to modify line 5 in Algorithm 6.11 into:

$$\text{Find } \boldsymbol{v} = \underset{\substack{\breve{\boldsymbol{v}} \in \mathcal{S} \\ \|\breve{\boldsymbol{v}}\|_2 = 1}}{\operatorname{argmin}} \sum_{j=1}^{\ell} \left\| D^{[j]} \left(I - P_{\mathcal{T}}\right) F^{[j]} \breve{\boldsymbol{v}} \right\|_2.$$

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

$$S = [S_1^T \quad S_2^T \quad \ldots \quad S_\ell^T]^T \in \mathbb{C}^{N\ell, m+1}, \qquad \text{where} \tag{6.21}$$

$$S_j = D^{[j]} \left[F^{[j]} V_{m+1} - \breve{V}_{m+k+1} \left(\breve{V}_{m+k+1}^* F^{[j]} V_{m+1} \right) \right] \in \mathbb{C}^{N, m+1}. \tag{6.22}$$

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

$$\boldsymbol{c}^{[j]} = \left(D^{[j]} \widehat{V}_{m+k+1} \right)^{\dagger} \left(D^{[j]} F^{[j]} \boldsymbol{b} \right) / \|\boldsymbol{b}\|_2, \tag{6.23}$$

Algorithm 6.13 Rational Krylov Fitting (RKFIT). RKToolbox: `rkfit`

Input: Matrix $A \in \mathbb{C}^{N,N}$, a family of matrices $\{F^{[j]}\}_{j=1}^{\ell} \subset \mathbb{C}^{N,N}$, a vector $\boldsymbol{b} \in \mathbb{C}^N$, and a starting guess $q_m \in \mathcal{P}_m$ for the denominator.

Optional input: A family of matrices $\{D^{[j]}\}_{j=1}^{\ell} \subset \mathbb{C}^{N,N}$, integer $k \in \mathbb{Z}$ such that $k \geq -m$, tolerances $\varepsilon_{\texttt{tol}}, \varepsilon_{\texttt{safe}} \in \mathbb{R}_0^+$, flags `real` and `reduction`, and maximal number `maxit` of relocation iterations to perform.

Output: RAD (6.8) and vectors $\{\boldsymbol{c}^{[j]}\}_{j=1}^{\ell}$ representing the approximants $\{r^{[j]}\}_{j=1}^{\ell}$.

1. Initialise missing optional input parameters as indicated by Table 6.1.
2. Set `real` to false if any of $A, \{F^{[j]}, D^{[j]}\}_{j=1}^{\ell}, \boldsymbol{b}$ or q_m is not real-valued.
3. Compute $\boldsymbol{f}^{[j]} = D^{[j]}F^{[j]}\boldsymbol{b}$ and $f^{[j]} = \|\boldsymbol{f}^{[j]}\|_2$ for $j = 1, 2, \ldots, \ell$.
4. **for** `it` $= 1, 2, \ldots,$ `maxit` **do**
5. Compute RAD (6.7) for $\mathcal{S} = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ by Algorithm 2.2/2.3.
6. **if** $k \geq 0$ **then**
7. Extend RAD (6.7) to (6.8) by adding k infinite poles using Algorithm 2.2/2.3.
8. **else**
9. Transform RAD (6.7) to (6.9) by Algorithm 5.10 and truncate it to (6.8).
10. **end if**
11. Form $\{\boldsymbol{c}^{[j]}\}_{j=1}^{\ell}$ as in (6.23).
12. Compute misfit as in (6.5) (exploiting the previously computed $\boldsymbol{f}^{[j]}$ and $f^{[j]}$).
13. **if** misfit $\leq \varepsilon_{\texttt{tol}}$ **then**
14. **if** `reduction` **then**
15. Form S, with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{m+1}$, as in (6.21)–(6.22).
16. Let $\Delta m \leq \min\{m, m+k\}$ be the largest integer for which (6.18) holds.
17. Compute $m - \Delta m$ new poles following Theorem 6.3 with $\widehat{\boldsymbol{c}}_j$ being the right singular vectors corresponding to the smallest singular values of S.
18. Update $m := m - \Delta m$. Set `reduction2` to true and `reduction` to false.
19. **else**
20. **return**
21. **end if**
22. **end if**
23. Compute S (if needed) and a right singular vector $\widehat{\boldsymbol{c}}$ corresponding to σ_{m+1}.
24. Replace the poles q_m with the generalised eigenvalues of (6.11), where Q_{m+1} is a unitary/orthogonal matrix with $Q_{m+1}\boldsymbol{e}_1 = \widehat{\boldsymbol{c}}$.
25. **end for**
26. **if** misfit $\leq \varepsilon_{\texttt{tol}}$ and `reduction2` (defined) **then**
27. Update (6.8) using Algorithm 5.10 to obtain degree revealing basis (if needed).
28. Update accordingly $\{\boldsymbol{c}^{[j]}\}_{j=1}^{\ell}$ to get representation in the new basis.
29. Truncate close to zero components at the rear of $\boldsymbol{c}^{[j]}$ following Section 6.4.3.
30. **end if**

Table 6.1: Default RKFIT parameters.

parameter	value	parameter	value	parameter	value	parameter	value
D^j	I	$\varepsilon_{\texttt{tol}}$	10^{-15}	`real`	false	`maxit`	10
k	0	$\varepsilon_{\texttt{safe}}$	10^{-1}	`reduction`	false		

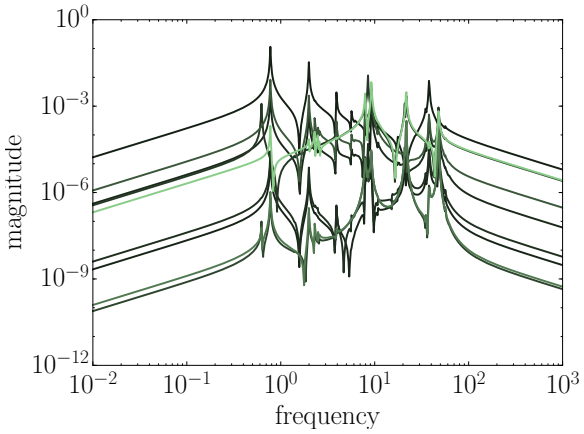
which reduces to $\boldsymbol{c}^{[j]} = \widehat{V}^*_{m+k+1}\big(F^{[j]}\boldsymbol{b}\big)/\|\boldsymbol{b}\|_2$ if $D^{[j]} = I_N$. The remaining parts of RKFIT, with the exception of the degree reducing strategy, are unaffected. In order to make sure that all of $\{r^{[j]}\}_{j=1}^{\ell}$ have the same denominator the reduction of m should be based on the singular values of S, and not the individual S_j. For the reduction of k, one can either reduce k by the smallest acceptable reduction k_j for $r^{[j]}$, or make potentially different reductions for each $r^{[j]}$ locally. The complete algorithm is summarised in Algorithm 6.13.

6.6 Numerical experiments (with $\ell > 1$)

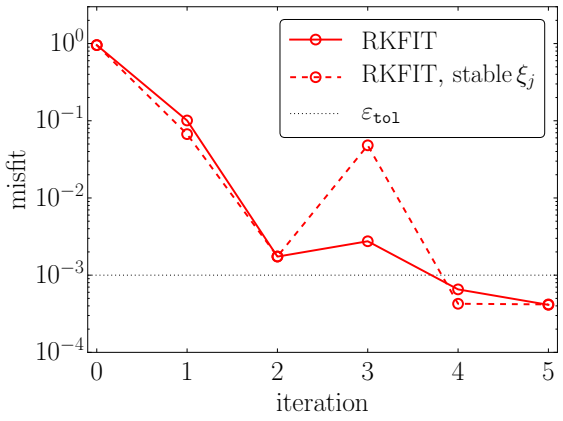
We now show the performance of RKFIT when $\ell > 1$ for two different applications.

6.6.1. MIMO dynamical system. We consider a model for the transfer function of the MIMO system ISS 1R [21]. 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 [21], 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 6.1.4. The magnitudes of the $\ell = 9$ transfer functions to be fitted are plotted in Figure 6.6(a).

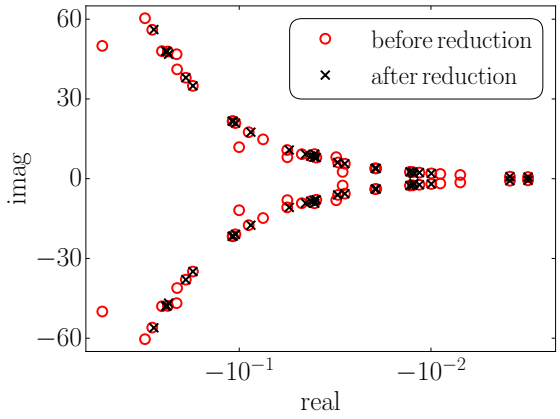
For the first experiment, we try to find rational functions of type $(70, 70)$, and then reducing their degrees. A tolerance of $\varepsilon_{\texttt{tol}} = 10^{-3}$ is used. In Figure 6.6(b) two convergence curves are shown, one for RKFIT as described in the previous sections (solid line), and the 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 $\varepsilon_{\texttt{tol}}$ at iteration 4, after which the degree reduction discussed in Section 6.4 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 6.6(c). The



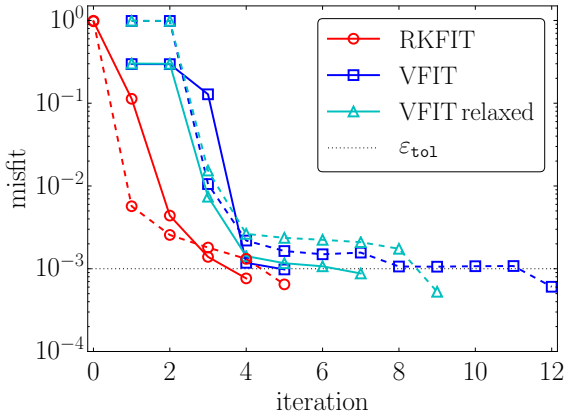
(a) Magnitude of the frequency responses.



(b) RKFIT with degree reduction.



(c) Selecting poles during the reduction.



(d) Comparison between RKFIT and VFIT.

Figure 6.6: Low-order model approximation to the MIMO system ISS from [21]. 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 6.6.1.

error corresponding to the new 56 (respectively 54) poles corresponds to iteration 5; as it is still below $\varepsilon_{\texttt{tol}}$ no further RKFIT iterations are required.

For the second experiment we compare RKFIT with the vector fitting code VFIT [26, 52, 54] for two different choices of initial poles, and with different normalisation conditions for VFIT. The results are reported in Figure 6.6(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 [52, 54]. The dashed curves result when using as initial poles the eigenvalues of a real-valued random

matrix. In both cases RKFIT outperforms VFIT, independently of the normalisation condition used by VFIT. Depending on the 56 initial poles, RKFIT requires either 4 or 5 iterations. This has to be compared to Figure 6.6(b), where the 56 poles selected by our reduction strategy immediately give a misfit below $\varepsilon_{\texttt{tol}}$ so that no further iteration is required. This provides further evidence that our approximate GCD strategy for choosing the poles after reducing m works well in practice.

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

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

at several time points $t_1, t_2, \ldots, t_\ell$. 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 nonsingular, the exact solutions $\boldsymbol{u}(t_j)$ are given by $\boldsymbol{u}(t_j) = \exp(-t_j K^{-1}L)\boldsymbol{u}_0$, and a popular approach for approximating $\boldsymbol{u}(t_j)$ is to use rational functions $r^{[j]}$ of the form

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

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 factorisation 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 [17, 33, 79, 81, 106]. Let us assume that L is Hermitian positive semi-definite, K is Hermitian positive definite, and introduce $\|\boldsymbol{v}\|_K := \sqrt{\boldsymbol{v}^* K \boldsymbol{v}}$. Then

$$\begin{aligned} \|\exp(-t_j K^{-1}L)\boldsymbol{b} - r^{[j]}(K^{-1}L)\boldsymbol{b}\|_K &\leq \|\boldsymbol{b}\|_K \max_{\lambda \in \Lambda(L,K)} |e^{-t_j\lambda} - r^{[j]}(\lambda)| \\ &\leq \|\boldsymbol{b}\|_K \max_{\lambda \geq 0} |e^{-t_j\lambda} - r^{[j]}(\lambda)|. \end{aligned} \tag{6.24}$$

In order to use RKFIT for finding poles $\xi_1, \xi_2, \ldots, \xi_m$ of the rational functions $r^{[j]}$ such that the right-hand side (6.24) of the inequality is small for all $j = 1, 2, \ldots, \ell$, we propose a surrogate approach similar to that in [17]. Let $A = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ 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 $\boldsymbol{b} = [1 \quad 1 \quad \ldots \quad 1]^T$ 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

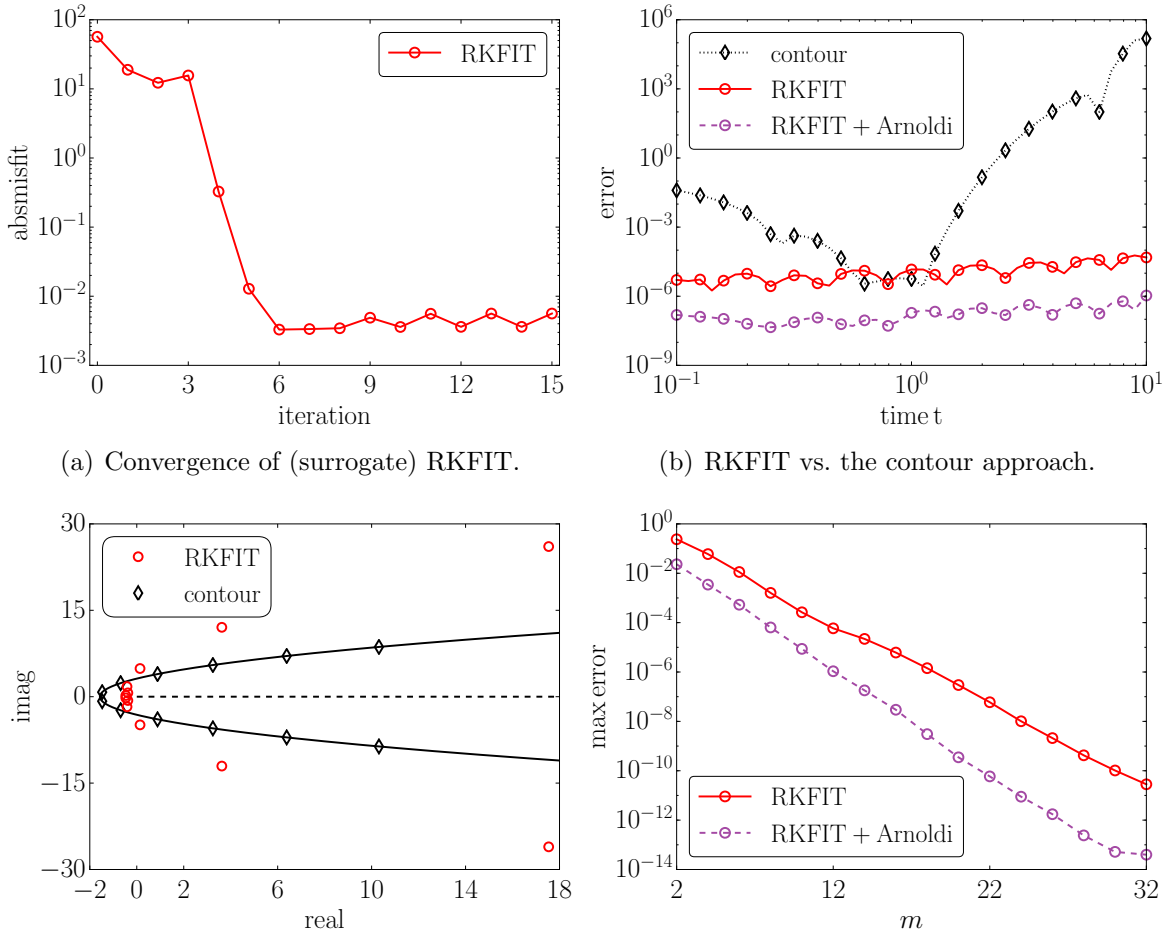
$$\text{absmisfit} = \sum_{j=1}^{\ell} \|F^{[j]}\boldsymbol{b} - r^{[j]}(A)\boldsymbol{b}\|_2^2$$

is minimised. Note that

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

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 $\lambda \geq 0$ one can expect the upper error bound (6.24) to be small.

Figure 6.7(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 [106, Figure 6.1] by approximating $\boldsymbol{u}(t_j) = \exp(-t_j L)\boldsymbol{u}_0$ and comparing the result with MATLAB `expm`. Here, $L \in \mathbb{R}^{2401,2401}$ 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 6.7(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). We see that the error is approximately uniform and smaller than 6.21×10^{-5} over the whole time interval $[10^{-1}, 10^{1}]$. An alternative is to approximate $\boldsymbol{u}(t_j)$ with an extraction procedure introduced in Section 3.2. For instance, the standard rational Arnoldi approximation corresponding to a quasi-RAD for $\mathcal{Q}_{m+1}(L, \boldsymbol{v}, \{\xi_j\}_{j=1}^m)$ yields rational approximations of type $(m-1, m)$, same as the



(a) Convergence of (surrogate) RKFIT.

(b) RKFIT vs. the contour approach.

(c) Poles from the two approaches.

(d) Quality of RKFIT as m increases.

Figure 6.7: Approximating $\exp(-tL)\boldsymbol{u}_0$ for a range of parameters t with rational approximants sharing common poles. The convergence behaviour of RKFIT with the surrogate problem, 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 the contour-based approach (dotted curve with diamonds), RKFIT (solid curve with circles) and the standard rational Arnoldi approximation with poles obtained from RKFIT (dashed curve with circles). In (c) we show the poles of the two families of rational approximants, while in (d) we show the maximal error for the two RKFIT-based approximations, uniformly over all $t_j \in [10^{-1}, 10^{1}]$, for various m.

type of the approximants $r^{[j]}$. Such approximations clearly have the same poles as the $r^{[j]}$, however, the numerators are based on L rather then the surrogate matrix A and thus may be better. This is indeed observed in Figure 6.7(b), were we denote the standard rational Arnoldi approximation as "RKFIT + Arnoldi" (dashed curve with circles). The $m = 12$ poles of the rational functions $r^{[j]}$ are shown in Figure 6.7(c) (circles).

Another approach for obtaining a family of rational approximants is to use contour integration [106]. Applying an m-point quadrature rule to the Cauchy integral 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 it has already been pointed out in [106], such quadrature-based approximants tend be good only for a small range of parameters t_j. In Figure 6.7(b) we see that the error $\|\boldsymbol{u}(t_j) - \widetilde{r}^{[j]}(L)\boldsymbol{u}_0\|_2$ increases 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 [106, eq. (3.1)], are shown in Figure 6.7(c) (diamonds).

6.7 RKToolbox corner

In RKToolbox Example 6.1 we list the possible calls to our `rkfit` implementation of Algorithm 6.13 contained in the RKToolbox. Let us first comment on the list of input parameters. `F` may be a matrix of size N-by-N, a function handle representing matrix-matrix multiplication (such as, for instance, `F = @(X)A*X`, where `A` is a matrix), or a cell array of `ell` such objects `F{1}`, `F{2}`, ..., `F{ell}`. The variables `A`, `b` and `xi` are the same as for `rat_krylov`; see Section 2.6. On input, the 1-by-m row vector `xi` contains the initial m poles.

As can be noted in lines 3–5, for the maximal number of iterations `rkfit` can perform, we can provide the variable `maxit`, while for the tolerance $\varepsilon_{\texttt{tol}}$ (see Section 6.1.3) we can optionally provide `tol`. If the initial poles `xi` are closed under complex conjugation and the remaining data are real-valued, we can take advantage of the structure and use only real arithmetic, as shown in line 5. If some of this optional parameters are not provided, then the default ones, as listed in Table 6.1, are used. On the other hand, if one wants to specify the remaining optional parameters, the call in line 7 needs to be employed. Therein, `param` is a structure with fields, for instance, `maxit`, `tol` and `real`, with the same meaning as the variables in lines 3–5. Other fields include the flag `reduction` which indicates whether the degree reduction from Section 6.4 needs to be applied upon convergence, and `k` for specifying the type $(m + k, m)$. A complete list of `param`'s fields may be obtained by typing `help rkfit` in MATLAB command line.

Finally, let us comment on the output list `[xi, r, misfit, out]`. The vector `xi` contains the poles of the rational function(s) obtained by `rkfit`, `misfit` is a

```
[xi, r, misfit, out] = rkfit(F, A, b, xi);

[xi, r, misfit, out] = rkfit(F, A, b, xi, maxit);
[xi, r, misfit, out] = rkfit(F, A, b, xi, maxit, tol);
[xi, r, misfit, out] = rkfit(F, A, b, xi, maxit, tol, 'real');

[xi, r, misfit, out] = rkfit(F, A, b, xi, param);
```

RKToolbox Example 6.1: Using RKFIT.

vector containing the misfit (6.5) after each `rkfit` iteration, including the starting one corresponding to the provided initial poles, and `out` is a structure that collects various intermediate data, such as, for instance, the poles before and after reduction. The variable `r` is either an instance of the MATLAB class *RKFUN*, or a cell array of RKFUN instances, depending on ℓ. The class RKFUN is used to represent a rational function numerically, and is the topic of the following chapter.

7 Working with rational functions in pencil format

In this chapter we develop a system for working numerically with rational functions. The system is based on the (scalar) RADs introduced in Section 2.2. It follows from Theorem 2.13 that the unreduced pencil $(\underline{H_m}, \underline{K_m})$ from an RAD

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m} \tag{7.1}$$

for $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ encodes a basis $\{r_j\}_{j=0}^{m}$ for the set of all rational functions of type at most (m, m) with the fixed denominator q_m. A particular rational function r is then specified by a vector $\boldsymbol{c} \in \mathbb{C}^{m+1}$ containing the unique expansion coefficients $c_j \in \mathbb{C}$ of r in that basis, i.e., such that $r = \sum_{j=0}^{m} c_{j+1} r_j$.

In Section 7.1 we show how to use the triplet $(\underline{H_m}, \underline{K_m}, \boldsymbol{c}) \equiv r$ for the numerical evaluation of r, as well as for pole and root finding. The evaluation is considered both for scalars $z \in \mathbb{C}$ and for matrices $A \in \mathbb{C}^{N,N}$ times a vector $\boldsymbol{b} \in \mathbb{C}^N$, i.e., we show how to evaluate $r(z)$ and $r(A)\boldsymbol{b}$. In Section 7.2 we show how to perform basic arithmetic operations (addition, subtraction, multiplication and division) with objects in the new format, and in Section 7.3 we consider more advanced operations. Specifically, we consider changing the basis r is represented in, into the *partial fraction basis* that reveals the *residues* of each pole. Finally, in Section 7.4 we discuss our MATLAB implementation within the RKToolbox and we showcase the algorithms. Rational functions are represented as instances of a MATLAB object of the class *RKFUN* (which stands for *Rational Krylov FUNction*). The use of MATLAB object-oriented programming capabilities for these purposes is inspired by the Chebfun system [27].

7.1 Evaluation, pole and root finding

Let $(\underline{H}_m, \underline{K}_m)$ be a regular upper Hessenberg pencil of $(m+1)$-by-m complex-valued matrices. Define $\{r_j = p_j/q_j\}_{j=0}^m$ as in Theorem 2.13. Then, (2.18) still holds for any $z \in \mathbb{C}$ such that $q_m(z) \neq 0$, even if $(\underline{H}_m, \underline{K}_m)$ does not correspond to an RAD. The proof is the same as for Theorem 2.13. This is a simple consequence of the fact that the rational functions r_j are defined recursively through $(\underline{H}_m, \underline{K}_m)$; see also (2.17). For instance, (2.18) holds even when $\underline{H}_m = \underline{K}_m$, which is, by Lemma 2.6, not possible for RADs. In this case $(\underline{H}_m, \underline{K}_m)$ does not encode a basis but a *generating system*, that is, a superset of a basis, $\{r_j = p_j/q_j\}_{j=0}^m$ of rational functions. Nevertheless, the *RKFUN representation*

$$r \equiv (\underline{H}_m, \underline{K}_m, \boldsymbol{c}), \tag{7.2}$$

discussed in the introduction of the chapter, can be used to represent a rational function r. Furthermore, we can use the real-valued version with an upper quasi-Hessenberg matrix $\underline{H}_m$ together with a real-valued coefficient vector $\boldsymbol{c} \in \mathbb{R}^{m+1}$, if we want to avoid complex arithmetic, when possible. However, for simplicity, we shall not discuss the details for avoiding complex arithmetic, but the implementations in the RKToolbox support this option.

7.1.1. Evaluation of an RKFUN. Let r be as in (7.2). Based on Theorem 2.14 we can evaluate r at a scalar $z \in \mathbb{C}$ by considering the QR factorisation of $\underline{H}_m - z\underline{K}_m$, as was also discussed in Section 6.2. There is, however, a more efficient approach which, unlike the QR approach, is applicable to matrices as well. Furthermore, the evaluation of r at a scalar $z \in \mathbb{C}$ such that $q_m(z) \neq 0$ can be interpreted as the (matrix-valued) evaluation $r(A)\boldsymbol{b}$ with $A = \begin{bmatrix} z \end{bmatrix} \in \mathbb{C}^{1,1}$, and $\boldsymbol{b} = \begin{bmatrix} 1 \end{bmatrix} \in \mathbb{C}^1$. Therefore, we discuss directly the evaluation $r(A)\boldsymbol{b}$.

Let $A \in \mathbb{C}^{N,N}$ be a matrix whose eigenvalues are not poles of r, and let $\boldsymbol{b} \in \mathbb{C}^N$ be a nonzero vector. Note that in this chapter there are no restrictions on N. In order to form $r(A)\boldsymbol{b}$ we proceed in two steps. First, we compute the generating system $\{r_j(A)\boldsymbol{b}\}_{j=0}^m$, and, second, we form $r(A)\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b} & r_1(A)\boldsymbol{b} & r_2(A)\boldsymbol{b} & \ldots & r_m(A)\boldsymbol{b} \end{bmatrix} \boldsymbol{c}$. The generating system is computed recursively, essentially by rerunning the rational Arnoldi algorithm with the given pencil $(\underline{H}_m, \underline{K}_m)$. These two-step procedure is given

Algorithm 7.14 Evaluating an RKFUN. RKToolbox: `rkfun.feval`

Input: $A \in \mathbb{C}^{N,N}$, $\boldsymbol{b} \in \mathbb{C}^N$, and $r \equiv (\underline{H_m}, \underline{K_m}, \boldsymbol{c})$ with poles outside $\Lambda(A)$ and an unreduced upper-Hessenberg pencil $(\underline{H_m}, \underline{K_m})$ of size $(m+1)$-by-m.
Output: Vector $\boldsymbol{r} = r(A)\boldsymbol{b}$.

1. Let $\boldsymbol{w}_1 = \boldsymbol{b}$.
2. **for** $j = 1, 2, \ldots, m$ **do**
3. Let $\mu_j = h_{j+1,j}, \nu_j = k_{j+1,j}$, and take any $\rho_j, \eta_j, \in \mathbb{C}$ such that $\mu_j\rho_j \neq \nu_j\eta_j$.
4. Set $\boldsymbol{t}_j := \mu_j\boldsymbol{k}_j - \nu_j\boldsymbol{h}_j \in \mathbb{C}^j$, and $\underline{\boldsymbol{y}_j} := \eta_j\underline{\boldsymbol{k}_j} - \rho_j\underline{\boldsymbol{h}_j} \in \mathbb{C}^{j+1}$. ▷ See (2.4), (2.5).
5. Compute $\boldsymbol{w} := (\nu_j A - \mu_j I)^{-1}(\rho_j A - \eta_j I)W_j\boldsymbol{t}_j$.
6. Compute $\boldsymbol{w}_{j+1} := (\boldsymbol{w} - W_j\boldsymbol{y}_j)/y_{j+1,j}$. ▷ We now have $AW_{j+1}\underline{K_j} = W_{j+1}\underline{H_j}$.
7. **end for**
8. Compute $\boldsymbol{r} = W_{m+1}\boldsymbol{c}$.

in Algorithm 7.14. In the first part, lines 1–7, we form an RAD-like decomposition

$$AW_{m+1}\underline{K_m} = W_{m+1}\underline{H_m}, \tag{7.3}$$

where $W_{m+1} := \begin{bmatrix}\boldsymbol{b} & r_1(A)\boldsymbol{b} & r_2(A)\boldsymbol{b} & \ldots & r_m(A)\boldsymbol{b}\end{bmatrix}$ is not necessarily of full column rank (hence, RAD-*like*). Finally, in line 8 we form $r(A)\boldsymbol{b}$.

7.1.2. Pole and root finding. For completeness, let us remark that the roots of an RKFUN (7.2) can be computed as in (6.11), where Q_{m+1} is defined via $\boldsymbol{c}$ instead of the notation $\widehat{\boldsymbol{c}}$ used in the paragraph containing (6.11). Of course, the poles of (7.2) are the eigenvalues of the lower m-by-m subpencil of $(\underline{H_m}, \underline{K_m})$.

7.2 Basic arithmetic operations

We now consider performing basic arithmetic operations with RKFUNs. Specifically, we consider two RKFUNs, say, (7.2) and

$$\widehat{r} \equiv (\underline{\widehat{H}_\ell}, \underline{\widehat{K}_\ell}, \widehat{\boldsymbol{c}}), \tag{7.4}$$

and want to obtain an RKFUN representation for $r \pm \widehat{r}$, $r\widehat{r}$, and $r/\widehat{r}$.

7.2.1. Sum of RKFUNs. Let the scalar RADs corresponding to r and $\widehat{r}$ be

$$z\begin{bmatrix}r_0(z) & r_1(z) & \ldots & r_m(z)\end{bmatrix}\underline{K_m} = \begin{bmatrix}r_0(z) & r_1(z) & \ldots & r_m(z)\end{bmatrix}\underline{H_m}, \quad \text{and} \tag{7.5}$$

$$z\begin{bmatrix}\widehat{r}_0(z) & \widehat{r}_1(z) & \ldots & \widehat{r}_\ell(z)\end{bmatrix}\underline{\widehat{K}_\ell} = \begin{bmatrix}\widehat{r}_0(z) & \widehat{r}_1(z) & \ldots & \widehat{r}_\ell(z)\end{bmatrix}\underline{\widehat{H}_\ell}, \tag{7.6}$$

respectively. Recall that $r_0 = \widehat{r}_0 \equiv 1$. Hence, $r = \sum_{j=0}^{m} c_{j+1} r_j$, $\widehat{r} = \sum_{j=0}^{\ell} \widehat{c}_{j+1} \widehat{r}_j$, and

$$r + \widehat{r} = (c_1 + \widehat{c}_1) + \sum_{j=1}^{m} c_{j+1} r_j + \sum_{j=1}^{\ell} \widehat{c}_{j+1} \widehat{r}_j. \tag{7.7}$$

This suggests using the pencil $(\underline{H}^{\oplus}_{m+\ell}, \underline{K}^{\oplus}_{m+\ell})$ defined by

$$\underline{H}^{\oplus}_{m+\ell} := \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,m} & \widehat{h}_{1,1} & \widehat{h}_{1,2} & \dots & \widehat{h}_{1,\ell} \\ h_{2,1} & h_{2,2} & \dots & h_{2,m} & & & & \\ & \ddots & \ddots & \vdots & & & & \\ & & \ddots & h_{m,m} & & & & \\ & & & h_{m+1,m} & & & & \\ & & & & \widehat{h}_{2,1} & \widehat{h}_{2,2} & \dots & \widehat{h}_{2,\ell} \\ & & & & & \ddots & \ddots & \vdots \\ & & & & & & \ddots & \widehat{h}_{\ell,\ell} \\ & & & & & & & \widehat{h}_{\ell+1,\ell} \end{bmatrix} \in \mathbb{C}^{m+\ell+1,m+\ell}, \tag{7.8}$$

and analogously for $\underline{K}^{\oplus}_{m+\ell}$. Note that, for instance,

$$p^{\oplus}_{m+1}(z) := \det(zK^{\oplus}_{m+1} - H^{\oplus}_{m+1}) = \det(z\widehat{k}_{1,1} - \widehat{h}_{1,1}) q_m(z),$$

with q_m as in (2.9). Furthermore,

$$q^{\oplus}_{m+1}(z) := \det(z[\mathbf{0} \quad I_{m+1}]\underline{K}^{\oplus}_{m+1} - [\mathbf{0} \quad I_{m+1}]\underline{H}^{\oplus}_{m+1}) = \det(z\widehat{k}_{2,1} - \widehat{h}_{2,1}) q_m(z),$$

and thus $p^{\oplus}_{m+1}/q^{\oplus}_{m+1} = \widehat{r}_1$. Similarly we find $p^{\oplus}_{m+j}/q^{\oplus}_{m+j} = \widehat{r}_j$ for the remaining $j = 2, 3, \dots, \ell$. Therefore,

$$r + \widehat{r} \equiv (\underline{H}^{\oplus}_{m+\ell}, \underline{K}^{\oplus}_{m+\ell}, \boldsymbol{c}^{\oplus}), \tag{7.9}$$

where $\underline{H}^{\oplus}_{m+\ell}$ is given by (7.8), $\underline{K}^{\oplus}_{m+\ell}$ analogously, and, based on (7.7),

$$\boldsymbol{c}^{\oplus} = \begin{bmatrix} c_1 + \widehat{c}_1 & c_2 & c_3 & \dots & c_{m+1} & \widehat{c}_2 & \widehat{c}_3 & \dots & \widehat{c}_{\ell+1} \end{bmatrix}^T \in \mathbb{C}^{m+\ell+1}. \tag{7.10}$$

7.2.2. Difference of RKFUNs. If $\widehat{r} \equiv (\underline{\widehat{H}}_{\ell}, \underline{\widehat{K}}_{\ell}, \widehat{\boldsymbol{c}})$, then $-\widehat{r} \equiv (\underline{\widehat{H}}_{\ell}, \underline{\widehat{K}}_{\ell}, -\widehat{\boldsymbol{c}})$, and therefore $r - \widehat{r} = r + (-\widehat{r})$ can be formed as explained in Section 7.2.1. The two algorithms are implemented in the RKToolbox as `rkfun.plus` and `rkfun.minus`.

7.2.3. Product of RKFUNs. Here we assume that $c_{m+1} \neq 0$. If $c_{m+1} = 0$, then it can be removed and the last columns of $\underline{H}_m$ and $\underline{K}_m$ can be truncated. The idea is to

merge the two generating systems for r and $\widehat{r}$, but first (7.5) is transformed so that the function r_m is replaced by r. For instance, if $C_{m+1} = \begin{bmatrix} \underline{I_m} & \boldsymbol{c} \end{bmatrix}$, then

$$\begin{bmatrix} r_0(z) & r_1(z) & \dots & r_m(z) \end{bmatrix} C_{m+1} = \begin{bmatrix} r_0(z) & r_1(z) & \dots & r_{m-1}(z) & r(z) \end{bmatrix},$$

and the pencil $(\underline{H_m}, \underline{K_m})$ can be transformed accordingly to

$$(\underline{\widetilde{H}_m}, \underline{\widetilde{K}_m}) = (C_{m+1}^{-1}\underline{H_m}, C_{m+1}^{-1}\underline{K_m}). \tag{7.11}$$

It is now easy to show that

$$r\widehat{r} \equiv (\underline{H_{m+\ell}^{\otimes}}, \underline{K_{m+\ell}^{\otimes}}, \boldsymbol{c}^{\otimes}), \tag{7.12}$$

where

$$\underline{H_{m+\ell}^{\otimes}} := \begin{bmatrix} \widetilde{h}_{1,1} & \widetilde{h}_{1,2} & \dots & \widetilde{h}_{1,m} & & & & \\ \widetilde{h}_{2,1} & \widetilde{h}_{2,2} & \dots & \widetilde{h}_{2,m} & & & & \\ & \ddots & \ddots & \vdots & & & & \\ & & \ddots & \widetilde{h}_{m,m} & & & & \\ & & & \widetilde{h}_{m+1,m} & \widehat{h}_{1,1} & \widehat{h}_{1,2} & \dots & \widehat{h}_{1,\ell} \\ & & & & \widehat{h}_{2,1} & \widehat{h}_{2,2} & \dots & \widehat{h}_{2,\ell} \\ & & & & & \ddots & \ddots & \vdots \\ & & & & & & \ddots & \widehat{h}_{\ell,\ell} \\ & & & & & & & \widehat{h}_{\ell+1,\ell} \end{bmatrix} \in \mathbb{C}^{m+\ell+1,m+\ell}, \tag{7.13}$$

analogously $\underline{K_{m+\ell}^{\otimes}}$, and with

$$\boldsymbol{c}^{\otimes} = \begin{bmatrix} 0 & 0 & \dots & 0 & \widehat{c}_1 & \widehat{c}_2 & \dots & \widehat{c}_{\ell+1} \end{bmatrix}^T \in \mathbb{C}^{m+\ell+1}. \tag{7.14}$$

Indeed, using the Laplace expansion of the involved determinants we find that

$$\frac{p_{m+j}^{\otimes}(z)}{q_{m+j}^{\otimes}(z)} \equiv \frac{\det(zK_{m+j}^{\otimes} - H_{m+j}^{\otimes})}{\det(z[\mathbf{0} \quad I_{m+j}]\underline{K_{m+j}^{\otimes}} - [\mathbf{0} \quad I_{m+j}]\underline{H_{m+j}^{\otimes}})} = r(z)\widehat{r}_j(z),$$

for $j = 0, 1, \dots, \ell$, which justifies (7.12).

7.2.4. Quotient of RKFUNs. The division $r/\widehat{r}$ is performed as the multiplication $r(\widehat{r})^{-1}$. Hence, we proceed with the inversion of an RKFUN $\widehat{r}$. We use Algorithm 5.7 to relocate the poles of $\widehat{r}$ to its roots (only transformations with the upper Hessenberg pencil are performed). Label with $(\underline{H_\ell^{[-1]}}, \underline{K_\ell^{[-1]}}) = (Q_{m+1}^*\underline{\widehat{H}_\ell}Z_m, Q_{m+1}^*\underline{\widehat{K}_\ell}Z_m)$ the

transformed pencil. The newly obtained pencil represents a generating system for $\widehat{r}^{-1}$, since its poles coincide with those of $\widehat{r}^{-1}$. Therefore

$$\widehat{r}^{-1} \equiv (\underline{H_\ell^{[-1]}}, \underline{K_\ell^{[-1]}}, \boldsymbol{c}^{[-1]}), \tag{7.15}$$

for some $\boldsymbol{c}^{[-1]} \in \mathbb{C}^{\ell+1}$. It follows from Theorem 5.4 that $\boldsymbol{c}^{[-1]} = \gamma Q_{m+1}\boldsymbol{e}_1$, where $\gamma \in \mathbb{C}$ is a scaling factor which can be obtained by enforcing $\widehat{r}(\lambda)\widehat{r}^{-1}(\lambda) = 1$, for any $\lambda \in \mathbb{C}$ such that $\widehat{r}(\lambda) \neq 0$ is defined.

The two algorithms, multiplication and division of RKFUNs, are implemented in the RKToolbox as `rkfun.times` and `rkfun.rdivide`, respectively. Furthermore, with `rkfun.power` one can compute r^k, for $k \in \mathbb{Z}$, by repeated multiplication.

7.3 Obtaining the partial fraction basis

We consider an RKFUN r as in (7.2) with pairwise distinct finite poles $\xi_1, \xi_2, \ldots, \xi_m$. We comment on extensions at the end of the section. Our goal is to find these poles and the corresponding *residues* $\widehat{c}_j$. In other words, we wish to obtain the parameters of the partial fraction expansion

$$r(z) = \widehat{c}_0 + \sum_{j=1}^{m} \frac{\widehat{c}_j}{z - \xi_j} \tag{7.16}$$

of r. We achieve this by transforming the scalar RAD (2.18) into

$$z\widehat{\boldsymbol{r}}(z)^T \begin{bmatrix} 0 & & & \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} = \widehat{\boldsymbol{r}}(z)^T \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_1 & & & \\ & \xi_2 & & \\ & & \ddots & \\ & & & \xi_m \end{bmatrix}, \tag{7.17}$$

where $\widehat{\boldsymbol{r}}(z)^T = \begin{bmatrix} \widehat{r}_0(z) & \widehat{r}_1(z) & \ldots & \widehat{r}_m(z) \end{bmatrix}$, with $\widehat{r}_0 = r_0 \equiv 1$. This transformation is achieved via left- and right-multiplication of the pencil $(\underline{H_m}, \underline{K_m})$ by nonsingular matrices $L_{m+1} \in \mathbb{C}^{m+1,m+1}$ and $R_m \in \mathbb{C}^{m,m}$ built as explained in Algorithm 7.15 below, thus producing $(L_{m+1}\underline{H_m}R_m, L_{m+1}\underline{K_m}R_m)$. The basis $\boldsymbol{r}(z)^T = \begin{bmatrix} r_0(z) & r_1(z) & \ldots & r_m(z) \end{bmatrix}$ of (2.18) is transformed accordingly to $\widehat{\boldsymbol{r}}(z)^T := \boldsymbol{r}(z)^T L_{m+1}^{-1}$. The jth column of (7.17) guarantees $z\widehat{r}_j(z) = 1 + \xi_j \widehat{r}_j(z)$, or equivalently, $\widehat{r}_j(z) = \frac{1}{z-\xi_j}$, for $j = 1, 2, \ldots, m$. Therefore, $\widehat{\boldsymbol{r}}(z)^T$ is indeed the partial fraction basis. Finally, $r = \boldsymbol{r}(z)^T\boldsymbol{c} = \widehat{\boldsymbol{r}}(z)^T L_{m+1}\boldsymbol{c}$ and, thus, $\widehat{\boldsymbol{c}} := L_{m+1}\boldsymbol{c}$ contains the residues of the partial fraction expansion.

Algorithm 7.15 Conversion to partial fraction form. RKToolbox: `rkfun.residue`

Input: RKFUN $r \equiv (\underline{H}_m, \underline{K}_m, \boldsymbol{c})$ with pairwise distinct finite poles.
Output: RKFUN $r \equiv (\underline{\widehat{H}}_m, \underline{\widehat{K}}_m, \widehat{\boldsymbol{c}})$, with $(\underline{\widehat{H}}_m, \underline{\widehat{K}}_m)$ as in (7.17).

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 $[\mathbf{0} \quad I_m]\underline{H}_m Q_m = Q_m \,\mathrm{diag}(\xi_1, \xi_2, \ldots, \xi_m)$.
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 $D_{m+1} = [-\boldsymbol{e}_1 \quad \underline{K}_m]$.
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 $R_m := R_m D_m$, $\underline{H}_m := \underline{H}_m D_m$, $\underline{K}_m := \underline{K}_m D_m$, where $D_m = \mathrm{diag}(1/h_{1j})$.
7. Redefine $D_m := \mathrm{diag}(1/k_{j+1,j})$, and $D_{m+1} := \mathrm{blkdiag}(1, D_m)$.
8. Update $L_{m+1} := D_{m+1}L_{m+1}$, $\underline{\widehat{H}}_m := D_{m+1}\underline{H}_m$, and $\underline{\widehat{K}}_m := D_{m+1}\underline{K}_m$.
9. Define $\widehat{\boldsymbol{c}} := L_{m+1}\boldsymbol{c}$.

The complete algorithm consists of four parts and gradually builds the matrices L_{m+1} and R_m. The first part corresponds to lines 1–3 in Algorithm 7.15, and it transforms the pencil so that the lower m-by-m part matches that of (7.17). The matrix $[\mathbf{0} \quad I_m]\underline{K}_m$ is nonsingular 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 eigenvector matrix Q_m is nonsingular since the eigenvalues $\xi_1, \xi_2, \ldots, \xi_m$ of $[\mathbf{0} \quad I_m]\underline{H}_m$ are all distinct. 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, and form $\widehat{\boldsymbol{c}}$ in line 9.

The transformation of $\boldsymbol{r}^T$ to the partial fraction basis $\widehat{\boldsymbol{r}}^T$ has condition number $\mathrm{cond}(L_{m+1})$, which can be arbitrarily bad in particular if some of the poles ξ_j are close to one another. Our implementation `rkfun.residue` in the RKToolbox therefore supports the use of MATLAB variable precision arithmetic as well as the use of the Advanpix Multiprecision Toolbox [1].

The partial fraction conversion can be extended to the case of repeated poles, both finite and infinite, which then amounts to bringing the lower m-by-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. As a starting point one could consider [69].

7.4 RKToolbox corner

Computing with RKFUNs. In RKToolbox Example 7.1 we show some basic usage of the RKFUN class. In line 1 we generate the RKFUN triplet for the rational function $r_1(z) = \frac{(z+1)(z-2)}{(z-3)^2}$ using the `rkfun.nodes2rkfun` function. The function allows to generate RKFUNs by specifying the numerator and denominator as monic nodal polynomials, i.e., by providing the roots and poles. Additional scaling may be applied afterwards, as we shall see in RKToolbox Example 7.2. In line 2 we display the triplet $(\underline{H}_2, \underline{K}_2, \boldsymbol{c}_3) \equiv r_1$ as a 3-by-7 matrix with two `NaN` columns, one separating $\underline{H}_2$ from $\underline{K}_2$, and the other separating $\underline{K}_2$ from $\boldsymbol{c}_3$. Line 4 shows how to evaluate $r_1(7) = \frac{40}{16} = 2.5$, while in the following two lines we calculate the roots and poles of r_1, respectively.

A new rational function r_2 is initialised in line 8 using `rkfun.nodes2rkfun` again. The rational function has three roots and two poles which means that an additional pole at infinity is included in the RKFUN triplet. In line 10 we compute $r = r_1 + r_2$, having the RKFUN triplet $(\underline{H}_5^{\oplus}, \underline{K}_5^{\oplus}, \boldsymbol{c}_6^{\oplus})$. Only $\underline{H}_5^{\oplus}$ and $\boldsymbol{c}_6^{\oplus}$ are displayed in line 11, again separated by a column of `NaN`s. The structure (7.8) can be observed in $\underline{H}_5^{\oplus}$. In line 13 we have verified that $r(7) = r_1(7) + r_2(7)$.

Finally, in line 15 we form $r := r_1 r_2$ using the MATLAB notation `.*` for scalar multiplication. The function `rkfun.times` which implements the algorithm discussed in Section 7.2.3 is invoked, and in the following line we show the corresponding $\underline{H}_5^{\otimes}$ and $\boldsymbol{c}_6^{\otimes}$. In this case r_1 already correspond to the last basis function and, consequently, $\underline{\widetilde{H}}_2 = \underline{H}_2$, cf. (7.11), can be spotted in the top-left corner of $\underline{H}_5^{\otimes}$.

Chapter heading. Further possibilities for computing with RKFUNs are considered in RKToolbox Example 7.2. In line 1 we define `x`, representing the identity map $x \mapsto x$, while in line 2 we construct `cheby`, representing the Chebyshev polynomial T_8 of degree 8. By typing `help rkfun.gallery`, one can obtain a complete list of supported RKFUN constructors. In line 3 we show that basic arithmetic operations can be performed between RKFUNs and scalars. The result is an RKFUN representing the "expected" rational function. The fragment `cheby(1./x)` produces an RKFUN for the composition $T_8 \circ r$, where $r(x) = \frac{1}{x}$. Composition $r_1 \circ r_2$ between two RKFUNs r_1 and r_2 is currently supported for rational functions r_2 of type $(0,0)$, $(1,0)$, $(0,1)$ and $(1,1)$.

```
r1 = rkfun.nodes2rkfun([-1, 2], [3, 3]);
disp([r1.H NaN(3, 1) r1.K NaN(3, 1)  r1.coeffs])

disp(r1(7))
disp(roots(r1).')
disp(poles(r1).')

r2 = rkfun.nodes2rkfun([1, -2, 0], [-4, 5]);

r = r1 + r2;                % calls rkfun.plus
[r.H NaN(6, 1) r.coeffs]

disp(r1(7)+r2(7) - r(7))

r = r1 .* r2;               % calls rkfun.times
[r.H NaN(6, 1) r.coeffs]
```

```
     1     0   NaN    -1     0   NaN     0
     3    -2   NaN     1    -1   NaN     0
     0     3   NaN     0     1   NaN     1
    2.5000
    -1     2
     3     3
     1     0     2     0     0   NaN     0
     3    -2     0     0     0   NaN     0
     0     3     0     0     0   NaN     1
     0     0    -4     0     0   NaN     0
     0     0     0     5     1   NaN     0
     0     0     0     0     1   NaN     1
     0
     1     0     0     0     0   NaN     0
     3    -2     0     0     0   NaN     0
     0     3     2     0     0   NaN     0
     0     0    -4     0     0   NaN     0
     0     0     0     5     1   NaN     0
     0     0     0     0     1   NaN     1
```

RKToolbox Example 7.1: Cumputing with RKFUNs.

```
x = rkfun();
cheby  = rkfun('cheby', 8);
cheby2 = 1./(1 + 1./(0.1*cheby(1./x).^2));
```

RKToolbox Example 7.2: Chapter heading.

The final function `cheby2` is the Chebyshev type 2 filter plotted in Figure 7.1. The filter is used, for instance, in signal processing in order to suppress unwanted or enhance wanted frequency components from a signal [15, 110], and we use it for the design of the chapter headings of this thesis.

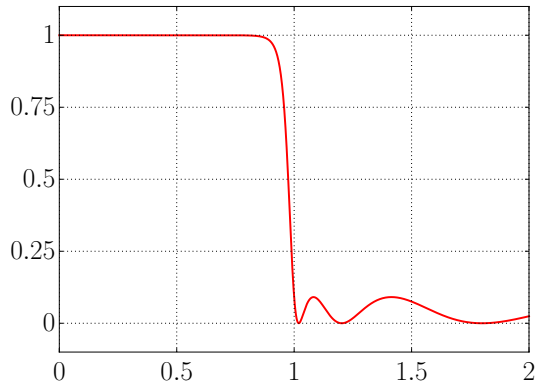


Figure 7.1: Chebyshev type 2 filter.

Degree reduction. When performing basic arithmetic operations with RKFUNs, we combine the triplets to form a new, bigger, one. For instance, if we subtract an RKFUN of type (m, m) from itself we obtain a new RKFUN of type $(2m, 2m)$, instead one of type $(0, 0)$. This is because the sum (or difference) of two rational functions of type (m, m) is of type $(2m, 2m)$ in the worst case, and we currently do not perform any degree reduction on the resulting sum (or difference). Similar problems may be encountered with multiplication and division. As a consequence, the resulting RKFUN may have degrees higher than necessary, which may lead to an unnecessarily fast growth of parameters, and might also cause problems when evaluating the RKFUN nearby spurious poles.

It might be possible to design a degree reduction similar to the degree reduction from Section 6.4 for RKFIT. In fact, the arithmetic operations may be performed by sampling the resulting function on a large enough set of interpolation nodes, and then fitting them with RKFIT, which produces an RKFUN with optionally reduced degrees. However, the need for providing interpolation nodes is a disadvantage, and, more importantly, the poles after reduction with the algorithm from Section 6.4 are not necessarily a subset of the original poles.

Rational Arnoldi approximation of $f(A)\boldsymbol{b}$ for Markov functions f. We present a simple implementation, RKToolbox Example 7.3, of Algorithm 3.4, built on top of the RKToolbox, RKFUNs, and other results from the thesis. The algorithm itself is explained in Section 3.2.4. The implemented function is called `util_markovfunmv` and it takes as arguments the matrix A, starting vector $\boldsymbol{b}$, (maximal) number of iterations

```
1  function [V, K, H, vf] = util_markovfunmv(A, b, m, fm, Gamma)
2    V = [b];
3    K = zeros(1, 0);
4    H = zeros(1, 0);
5
6    j  = 1;
7    xi = inf;
8
9    while j <= m
10     [V, K, H] = rat_krylov(A, V, K, H, xi); % Extend space.
11
12     [Q, R] = qr(K);                         % Find next pole.
13     s = rkfun(K, H, Q(:, end));
14     [~, index] = min(abs(s(Gamma)));
15     xi = Gamma(index);
16
17     Am = K\H;
18     vf = V*(K*fm(Am)*(K\(V'*b)));           % New approximant.
19
20     % If vf is good enough, then stop...
21
22     j = j+1;
23   end
24 end
```

RKToolbox Example 7.3: MATLAB implementation of Algorithm 3.4.

m, a function handle `fm` for f, and a discretisation `Gamma` of the region Γ. In lines 2–7 we initialise the data for storing the RAD parameters, including the first, infinite, pole. The main part is the `while` loop spanning lines 9–23. The RAD is extended in line 10, and the next pole is found in lines 12–15, where an RKFUN `s`, representing s_m defined in (3.23), is constructed. Here we can note that an RKFUN can be constructed by explicitly specifying the corresponding triplet. It is easy to show that the roots of `s` defined in line 13 are indeed the roots of s_m, assuming exact arithmetic. The scaling is irrelevant for finding the poles. The standard rational Arnoldi approximation if formed in line 18. This may be followed by a check for convergence, which we do not include. The interested reader is referred to [58, Section 4] for such considerations.

8 Conclusions

We introduced the notion of rational Krylov decompositions and identified necessary and sufficient conditions under which these decompositions correspond to rational Krylov spaces. Particular attention was given to rational Arnoldi decompositions (RADs) since they have a more intimate relation with a given basis of a rational Krylov spaces. An RAD specifies the starting vector (up to scaling) and poles defining the corresponding rational Krylov space $\mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$. We derived basic properties of RADs, generalised the implicit Q theorem to the rational case, and reconsidered the usage of rational Krylov methods within known applications from the literature as well as within new ones.

A common assumption in the rational Krylov literature is that the last pole has to be infinite in order to cheaply, that is, without the need of additional explicit projection, extract information from an RAD. We derived extraction strategies that do not impose such a requirement, by considering implicit projections on specific subspaces of the rational Krylov space at hand. Some of our numerical results indicate that the harmonic Ritz extraction for the $f(A)\boldsymbol{b}$ problem may be better then standard approaches.

By studying the internal parameters of the rational Arnoldi algorithm, called continuation pairs, we developed a more robust parallel implementation compared to previously available work. Eigenvalue applications, where the poles of a rational Krylov space are close to eigenvalues of A, appear to be more sensitive to the choice of continuation pairs in the parallel variant, compared to, for instance, applications in model order reduction or the approximation of $f(A)\boldsymbol{b}$.

Studying the relocation of poles within an RAD led to RKFIT, an iterative algorithm for nonlinear rational least squares approximation. Different algorithms for *scalar*

nonlinear rational least squares have been considered in the past. One of the most popular methods in the engineering community is *vector fitting*. However, vector fitting and similar approaches typically lead to ill-conditioned numerical linear algebra problems. By using orthonormal RADs we employed a well-conditioned basis for the underlying space, and, ultimately, obtained a more robust and faster convergent algorithm. Furthermore, by restating the problem in matrix form, we obtained a more general algorithm. As a consequence, we believe that RKFIT may become a valuable tool for finding good pole parameters for rational Krylov methods. For instance, in applications where a small surrogate matrix $\widehat{A}$, sharing similar spectral properties as A, e.g., stemming from a coarsened finite element discretisation of a PDE, is available. As an example we considered a problem of exponential integration.

Finally, we developed a system for working numerically with rational functions represented in the so called RKFUN format, which is based on scalar RADs. For instance, we demonstrated how to evaluate an RKFUN and how to perform basic arithmetic operations with RKFUNs.

An interesting topic for future work regarding the rational Arnoldi algorithm is backward error analysis. Recently the shift-and-invert Arnoldi was considered in [97], but a generalisation of the analysis to the rational Arnoldi algorithm appears to be nontrivial as the poles in the rational Arnoldi algorithm may change from one iteration to the other. An additional difficulty appears to be the appearance of the reduced pencil $(\underline{H_m}, \underline{K_m})$ instead of a single upper Hessenberg matrix $\underline{H_m}$. From a computational point of view, an asynchronous high performance and parallel implementation of the rational Arnoldi algorithm may be of interest, as well as an implementation of the compact rational Arnoldi algorithm [109] where the basis has a particular Kronecker structure stemming from the structure of the pencil (A, B). Regarding RKFIT, the convergence and the degree reduction process may be further studied, perhaps to obtain a better understanding of the approximate GCD. From a practical point of view, it may be interesting to compare RKFIT with other rational approximation algorithms and within distinct applications. Furthermore, devising different reduction procedures may be of interest, for instance, when working with RKFUNs, where one may desire a subset selection type of reduction, i.e., a reduction where the new poles are a subset of the original ones.

Bibliography

[1] Advanpix LLC, *Multiprecision Computing Toolbox for MATLAB*, ver 3.8.3.8882, Tokyo, Japan, 2015. `http://www.advanpix.com/`. (Cited on p. 159.)

[2] E. Allen, J. Baglama, and S. Boyd, *Numerical approximation of the product of the square root of a matrix with a vector*, Linear Algebra Appl., 310 (2000), pp. 167–181. (Cited on p. 61.)

[3] A. Antoulas, D. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems*, Contemp. Math., 280 (2001), pp. 193–220. (Cited on p. 123.)

[4] I. Barrodale and J. Mason, *Two simple algorithms for discrete rational approximation*, Math. Comp., 24 (1970), pp. 877–891. (Cited on p. 124.)

[5] C. Beattie, *Harmonic Ritz and Lehmann bounds*, Electron. Trans. Numer. Anal., 7 (1998), pp. 18–39. (Cited on p. 70.)

[6] B. Beckermann, S. Güttel, and R. Vandebril, *On the convergence of rational Ritz values*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1740–1774. (Cited on pp. 61, 63.)

[7] B. Beckermann and L. Reichel, *Error estimation and evaluation of matrix functions via the Faber transform*, SIAM J. Numer. Anal., 47 (2009), pp. 3849–3883. (Cited on pp. 26, 72, 73.)

[8] P. Benner and J. Saak, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM Mitteilungen, 36 (2013), pp. 32–52. (Cited on p. 26.)

[9] 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. Last updated September 2015. (Cited on pp. 21, 28.)

[10] 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. (Cited on pp. 21, 26, 39.)

[11] M. Berljafa and S. Güttel, *The RKFIT algorithm for nonlinear rational approximation*, MIMS EPrint 2015.38, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2015. (Cited on pp. 21, 26, 140.)

[12] M. Berljafa and S. Güttel, *Parallelization of the rational Arnoldi algorithm*, MIMS EPrint 2016.32, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2016. (Cited on p. 21.)

[13] Å. Björck, *Solving linear least squares problems by Gram–Schmidt orthogonalization*, BIT, 7 (1967), pp. 1–21. (Cited on p. 34.)

[14] Å. Björck and C. C. Paige, *Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 176–190. (Cited on p. 34.)

[15] H. Blinchikoff and A. Zverev, *Filtering in the Time and Frequency Domains*, John Wiley & Sons Inc., New York, 1976. (Cited on pp. 123, 161.)

[16] P. Boito, *Structured Matrix Based Methods for Approximate Polynomial GCD*, vol. 15, Springer Science & Business Media, 2012. (Cited on p. 143.)

[17] R.-U. Börner, O. G. Ernst, and S. Güttel, *Three-dimensional transient electromagnetic modelling using rational Krylov methods*, Geophys. J. Int., 202 (2015), pp. 2025–2043. (Cited on pp. 61, 104, 147, 148.)

[18] D. Braess, *Nonlinear Approximation Theory*, Springer-Verlag, Berlin Heidelberg, 1986. (Cited on pp. 26, 124.)

[19] Z. Bujanović and Z. Drmač, *A new framework for implicit restarting of the Krylov–Schur algorithm*, Numer. Linear Algebra Appl., 22 (2015), pp. 220–232. (Cited on p. 116.)

[20] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48. (Cited on p. 136.)

[21] 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. (Cited on pp. 145, 146.)

[22] T. Davis and Y. Hu, *The University of Florida Sparse Matrix Collection*, ACM Trans. Math. Software, 38 (2011), pp. 1–25. (Cited on p. 106.)

[23] G. De Samblanx and A. Bultheel, *Using implicitly filtered RKS for generalised eigenvalue problems*, J. Comput. Appl. Math., 107 (1999), pp. 195–218. (Cited on pp. 62, 64, 67, 116.)

[24] G. De Samblanx, K. Meerbergen, and A. Bultheel, *The implicit application of a rational filter in the RKS method*, BIT, 37 (1997), pp. 925–947. (Cited on pp. 25, 41, 61, 62, 69, 116.)

[25] 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. (Cited on p. 138.)

[26] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, *Macromodeling of multiple systems using a fast implementation of the vector fitting method*, IEEE Microwave and Wireless Components Letters, 18 (2008), pp. 383–385. (Cited on p. 146.)

[27] T. A. Driscoll, N. Hale, and L. N. Trefethen, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014. (Cited on p. 153.)

[28] Z. Drmač, *SVD of Hankel matrices in Vandermonde-Cauchy product form*, Electron. Trans. Numer. Anal., 44 (2015), pp. 593–623. (Cited on p. 138.)

[29] 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. (Cited on p. 138.)

[30] Z. Drmač, S. Gugercin, and C. Beattie, *Vector fitting for matrix-valued rational approximation*, SIAM J. Sci. Comput., 37 (2015), pp. A2346–A2379. (Cited on p. 138.)

[31] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 755–771. (Cited on pp. 26, 61, 72.)

[32] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898. (Cited on p. 26.)

[33] 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. (Cited on pp. 26, 123, 147.)

[34] 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. (Cited on pp. 26, 123.)

[35] J. Eshof and M. Hochbruck, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., 27 (2006), pp. 1438–1457. (Cited on p. 26.)

[36] D. Fasino, *Rational Krylov matrices and QR steps on Hermitian diagonal-plus-semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 743–754. (Cited on p. 50.)

[37] K. Gallivan, E. Grimme, and P. Van Dooren, *A rational Lanczos algorithm for model reduction*, Numer. Algorithms, 12 (1996), pp. 33–63. (Cited on pp. 25, 26, 123.)

[38] L. Giraud and J. Langou, *When modified Gram–Schmidt generates a well-conditioned set of vectors*, IMA J. Numer. Anal., 22 (2002), pp. 521–528. (Cited on pp. 34, 83.)

[39] L. Giraud, J. Langou, and M. Rozložník, *The loss of orthogonality in the Gram–Schmidt orthogonalization process*, Computers Math. Applic., 50 (2005), pp. 1069 – 1075. (Cited on p. 34.)

[40] T. Göckler and V. Grimm, *Uniform approximation of φ-functions in exponential integrators by a rational Krylov subspace method with simple poles*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1467–1489. (Cited on pp. 26, 56, 61.)

[41] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334. (Cited on p. 136.)

[42] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, USA, 4th ed., 2013. (Cited on pp. 29, 32, 33, 34, 61, 64, 65, 70, 89, 111.)

[43] P. Gonnet, S. Güttel, and L. N. Trefethen, *Robust Padé approximation via SVD*, SIAM Rev., 55 (2013), pp. 101–117. (Cited on p. 124.)

[44] P. Gonnet, R. Pachón, and L. N. Trefethen, *Robust rational interpolation and least-squares*, Electron. Trans. Numer. Anal., 38 (2011), pp. 146–167. (Cited on p. 124.)

[45] S. Goossens and D. Roose, *Ritz and harmonic Ritz values and the convergence of FOM and GMRES*, Numer. Linear Algebra Appl., 6 (1999), pp. 281–293. (Cited on pp. 70, 78.)

[46] L. Grasedyck, *Existence of a low rank or $\mathcal{H}$-matrix approximant to the solution of a Sylvester equation*, Numer. Linear Algebra Appl., 11 (2004), pp. 371–389. (Cited on p. 83.)

[47] A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical behaviour of the modified Gram–Schmidt GMRES implementation*, BIT, 37 (1997), pp. 706–719. (Cited on pp. 34, 83.)

[48] R. G. Grimes, J. G. Lewis, and H. D. Simon, *Eigenvalue problems and algorithms in structural engineering*, in Large Scale Eigenvalue Problems, J. Cullum and R. A. Willoughby, eds., vol. 127 of North-Holland Mathematics Studies, North-Holland, 1986, pp. 81–93. (Cited on p. 61.)

[49] E. Grimme, *Krylov Projection Methods for Model Reduction*, PhD thesis, University of Illinois at Urbana-Champaign, 1997. (Cited on pp. 26, 84.)

[50] S. Gugercin, 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. (Cited on p. 123.)

[51] S. Gugercin, A. C. Antoulas, and C. Beattie, $\mathcal{H}_2$ *model reduction for large-scale linear dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638. (Cited on p. 26.)

[52] B. Gustavsen, *Improving the pole relocating properties of vector fitting*, IEEE Trans. Power Del., 21 (2006), pp. 1587–1592. (Cited on pp. 125, 131, 137, 146.)

[53] 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. (Cited on p. 138.)

[54] B. Gustavsen and A. Semlyen, *Rational approximation of frequency domain responses by vector fitting*, IEEE Trans. Power Del., 14 (1999), pp. 1052–1061. (Cited on pp. 123, 125, 131, 132, 134, 136, 146.)

[55] S. Güttel, *Rational Krylov Methods for Operator Functions*, PhD thesis, Institut für Numerische Mathematik und Optimierung der Technischen Universität Bergakademie Freiberg, Freiberg, Germany, 2010. Available online at `http://nbn-resolving.de/urn:nbn:de:bsz:105-qucosa-27645`. (Cited on pp. 41, 61, 64, 72, 73, 74, 75, 84, 99.)

[56] S. Güttel, *Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection*, GAMM Mitteilungen, 36 (2013), pp. 8–31. (Cited on pp. 26, 61, 72, 73, 75, 78, 123.)

[57] S. Güttel and L. Knizhnerman, *Automated parameter selection for rational Arnoldi approximation of Markov functions*, Proc. Appl. Math. Mech., 11 (2011), pp. 15–18. (Cited on pp. 78, 79.)

[58] S. Güttel and L. Knizhnerman, *A black-box rational Arnoldi variant for Cauchy–Stieltjes matrix functions*, BIT, 53 (2013), pp. 595–616. (Cited on pp. 27, 61, 72, 73, 78, 79, 80, 163.)

[59] S. Güttel, R. Van Beeumen, K. Meerbergen, and W. Michiels, *NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 36 (2014), pp. A2842–A2864. (Cited on pp. 26, 62, 69.)

[60] N. J. Higham, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. (Cited on pp. 29, 32, 65, 76, 78, 130.)

[61] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925. (Cited on p. 61.)

[62] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numerica, 19 (2010), pp. 209–286. (Cited on p. 61.)

[63] M. E. Hochstenbach, *Generalizations of harmonic and refined Rayleigh–Ritz*, Electron. Trans. Numer. Anal., 20 (2005), pp. 235–252. (Cited on p. 70.)

[64] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2nd ed., 2013. (Cited on pp. 29, 30.)

[65] 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. (Cited on p. 123.)

[66] C. Jagels and L. Reichel, *Recursion relations for the extended krylov subspace method*, Linear Algebra Appl., 434 (2011), pp. 1716–1732. (Cited on p. 51.)

[67] E. Jarlebring and H. Voss, *Rational Krylov for nonlinear eigenproblems, an iterative projection method*, Appl. Math., 50 (2005), pp. 543–554. (Cited on p. 26.)

[68] B. Kågström and P. Poromaa, *Computing eigenspaces with specified eigenvalues of a regular matrix pair (A, B) and condition estimation: theory, algorithms and software*, Numer. Algorithms, 12 (1996), pp. 369–407. (Cited on p. 113.)

[69] 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. (Cited on p. 159.)

[70] D. Kressner, *Block algorithms for reordering standard and generalized Schur forms*, ACM Trans. Math. Software, 32 (2006), pp. 521–532. (Cited on p. 113.)

[71] G. Lassaux and K. Willcox, *Model reduction for active control design using multiple-point Arnoldi methods*, AIAA Paper, 616 (2003), pp. 1–11. (Cited on p. 26.)

[72] S. Lefteriu and A. Antoulas, *On the convergence of the vector-fitting algorithm*, IEEE Trans. Microw. Theory Techn., 61 (2013), pp. 1435–1443. (Cited on p. 138.)

[73] R. B. Lehoucq and K. Meerbergen, *Using generalized Cayley transformations within an inexact rational Krylov sequence method*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 131–148. (Cited on pp. 25, 39, 61, 62, 64, 85, 89.)

[74] E. C. Levy, *Complex-curve fitting*, IRE Trans. Autom. Control, AC-4 (1959), pp. 37–43. (Cited on p. 135.)

[75] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280. (Cited on p. 26.)

[76] J. Liesen and Z. Strakoš, *Krylov Subspace Methods: Principles and Analysis*, Oxford University Press, New York, 2013. (Cited on pp. 33, 46.)

[77] T. Mach, M. Pranić, and R. Vandebril, *Computing approximate (block) rational Krylov subspaces without explicit inversion with extensions to symmetric matrices*, Electron. Trans. Numer. Anal., 43 (2014), pp. 100–124. (Cited on p. 50.)

[78] K. Meerbergen, *An implicitly restarted rational Krylov strategy for Lyapunov inverse iteration*, IMA J. Numer. Anal., 36 (2016), pp. 655–674. (Cited on p. 61.)

[79] I. Moret and P. Novati, *RD-rational approximations of the matrix exponential*, BIT, 44 (2004), pp. 595–615. (Cited on pp. 123, 147.)

[80] Y. Nakatsukasa and R. W. Freund, *Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of Zolotarev's functions*, SIAM Rev., 58 (2016), pp. 461–493. (Cited on p. 123.)

[81] S. P. Nørsett, *Restricted Padé approximations to the exponential function*, SIAM J. Numer. Anal., 15 (1978), pp. 1008–1029. (Cited on p. 147.)

[82] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–133. (Cited on p. 70.)

[83] M. Pranić and L. Reichel, *Rational Gauss quadrature*, SIAM J. Numer. Anal., 52 (2014), pp. 832–851. (Cited on p. 51.)

[84] J. Rommes and N. Martins, *Efficient computation of multivariable transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Syst., 21 (2006), pp. 1471–1483. (Cited on p. 61.)

[85] J. Rommes and N. Martins, *Efficient computation of transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Syst., 21 (2006), pp. 1218–1226. (Cited on p. 61.)

[86] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405. (Cited on pp. 25, 35, 37, 38, 45, 55.)

[87] A. Ruhe, *The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices*, BIT, 34 (1994), pp. 165–176. (Cited on pp. 25, 27, 35, 37, 51, 55.)

[88] A. Ruhe, *Rational Krylov algorithms for nonsymmetric eigenvalue problems*, in Recent Advances in Iterative Methods, G. Golub, M. Luskin, and A. Greenbaum, eds., vol. 60, New York, 1994, Springer-Verlag, pp. 149–164. (Cited on pp. 25, 35.)

[89] A. Ruhe, *Rational Krylov algorithms for nonsymmetric eigenvalue problems. II. Matrix pairs*, Linear Algebra Appl., 198 (1994), pp. 283–295. (Cited on pp. 25, 35, 37, 38, 41, 45, 55, 62, 69.)

[90] A. Ruhe, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551. (Cited on pp. 25, 35, 37, 38, 39, 41, 55, 61, 62, 67, 70, 85, 87, 95, 99.)

[91] A. Ruhe, *The rational Krylov algorithm for nonlinear matrix eigenvalue problems*, J. Math. Sci. (N. Y.), 114 (2003), pp. 1854–1856. (Cited on p. 26.)

[92] 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. (Cited on p. 116.)

[93] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228. (Cited on pp. 61, 72, 74.)

[94] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd ed., 2003. (Cited on pp. 33, 91.)

[95] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2011. (Cited on p. 33.)

[96] C. Sanathanan and J. Koerner, *Transfer function synthesis as a ratio of two complex polynomials*, IEEE Trans. Automat. Control, 8 (1963), pp. 56–58. (Cited on p. 135.)

[97] C. Schröder and L. Taslaman, *Backward error analysis of the shift-and-invert Arnoldi algorithm*, Numer. Math., 133 (2016), pp. 819–843. (Cited on p. 166.)

[98] D. Skoogh, *An Implementation of a Parallel Rational Krylov Algorithm*, Licentiate thesis, Chalmers University of Technology, Göteborg, Sweden, 1996. (Cited on pp. 27, 84.)

[99] D. Skoogh, *A parallel rational Krylov algorithm for eigenvalue computations*, 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. 521–526. (Cited on pp. 27, 84, 99.)

[100] G. W. Stewart, *Matrix Algorithms, Volume I: Basic Decompositions*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998. (Cited on p. 29.)

[101] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614. (Cited on pp. 41, 116.)

[102] G. W. Stewart, *Matrix Algorithms, Volume II: Eigensystems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. (Cited on pp. 33, 34, 48, 61, 64, 67, 68, 70.)

[103] G. W. Stewart, *Addendum to "A Krylov–Schur algorithm for large eigenproblems"*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 599–601. (Cited on pp. 41, 116.)

[104] G. W. Stewart, *Backward error bounds for approximate Krylov subspaces*, Linear Algebra Appl., 340 (2002), pp. 81–86. (Cited on p. 103.)

[105] L. N. Trefethen, *Approximation Theory and Approximation Practice*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2013. (Cited on p. 26.)

[106] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer, *Talbot quadratures and rational approximations*, BIT, 46 (2006), pp. 653–670. (Cited on pp. 123, 147, 148, 149, 150.)

[107] M. Van Barel, D. Fasino, L. Gemignani, and N. Mastronardi, *Orthogonal rational functions and structured matrices*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 810–829. (Cited on p. 50.)

[108] R. Van Beeumen, K. Meerbergen, and W. Michiels, *A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 35 (2013), pp. A327–A350. (Cited on pp. 62, 69.)

[109] R. Van Beeumen, K. Meerbergen, and W. Michiels, *Compact rational Krylov methods for nonlinear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 820–838. (Cited on pp. 26, 39, 62, 69, 85, 166.)

[110] M. Van Valkenburg, *Analog Filter Design*, Holt, Rinehart, and Winston, 1982. (Cited on p. 161.)

[111] R. Vandebril and D. S. Watkins, *A generalization of the multishift QR-algorithm*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 759–779. (Cited on p. 50.)

[112] G. Wanner, E. Hairer, and S. Nørsett, *Order stars and stability theorems*, BIT, 18 (1978), pp. 475–489. (Cited on p. 123.)