KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Pole swapping methods for the eigenvalue problem

Rational QR algorithms

Daan Camps

Supervisors:
Prof. dr. R. Vandebril
Prof. dr. ir. K. Meerbergen

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Computer Science

September 2019

Pole swapping methods for the eigenvalue problem

Rational QR algorithms

Daan CAMPS

Examination committee:
Prof. dr. ir. H. Hens, chair
Prof. dr. R. Vandebril, supervisor
Prof. dr. ir. K. Meerbergen, supervisor
Prof. dr. ir. L. De Lathauwer
Prof. dr. ir. S. Vandewalle
Prof. dr. ir. P. Van Dooren
Prof. dr. B. Beckermann
(Université de Lille)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Computer Science

September 2019

Ter nagedachtenis aan Piet.
Grootvader, wiskundeleraar, levensgenieter.

Preface

This thesis is the outcome of my research conducted over the past four years under the supervision of Raf Vandebril and Karl Meerbergen. Reflecting on this period, I can't be anything but grateful for the opportunities I received and would like to take a moment to express my gratitude.

I first contacted Raf in April 2015 to inquire about the possibility of starting a PhD. At the time, I had already left university for nearly two years and had previously never followed a course taught by Raf. Nevertheless, Raf still invited me for a meeting and it rapidly became clear to me that this was something I'd really like to do. I started in September of the same year and ever since then Raf has been an enormous driving force behind my work. Raf, thank you for allowing me to pursue a PhD and for your advice, help, and support. It is wonderful how readily and frequently available you were for me. As my research project revolved around rational Krylov methods, Karl was involved from the start as my co-supervisor. His expertise in this area helped me tremendously over the years. Karl, thank you for your help and all your support.

One of the perks of being a researcher is the possibility to present your work at conferences, often organized at exotic locations. It was at such an occasion that I first met Paul Van Dooren at the NASCA Conference in Kalamata in July 2018. Thank you for taking an interest in my work, for your help, and never-ending stream of ideas. I would also like to thank Thomas Mach, David Watkins, and Nicola Mastronardi for the fruitful collaborations.

I thank the members of the examination committee, Bernd Beckermann, Lieven De Lathauwer, Stefan Vandewalle, and Paul Van Dooren, for reading my thesis text, providing me with valuable feedback, and challenging me during the preliminary defence. Thanks to Hugo Hens for chairing the defences.

Working towards your PhD can sometimes be a rather lonely process. Luckily I could enjoy the company of many colleagues to overcome this issue. Thanks to my office mates, initially Micol and Benjamin, later on Dries and Vincent,

for spending all this time together, although often in silence as we had our work to do. Thanks to Adrian, Andreas, Andrew, Bert, Bruno, Elien, Emil, Deesh, Dong, Francesco, Hannes, Koen, Joris, Luca, Matthias, Niel, Nick, Peter, Philippe, Pieter, Pieterjan, Roel, Sahar, Simon, Ward, and Yuya for the coffee and lunch breaks during which there were often animated discussions about a whole spectrum of topics. The ping-pong breaks often provided a welcome time-out, but unfortunately I was not able to take the *title* from Dries very often. I also loved the occasional social activities outside of the regular office hours: the dinners, the visits to the *oude markt*, the PhDays, the *afterworks*, etc.

I have fond memories of the conferences and workshops I attended and of many people that I had the pleasure to meet. The one-week workshop in April 2016 on low-rank tensors organized by the Hausdorff school in Bonn was a great start. Thanks Martijn, Otto, and Ben. My first *international* scientific conference, the ILAS in July 2016, was held in the city center of Leuven, just a stone's throw from our department. I must admit however that it was convenient to return home at the end of the day. Thanks goes to Simon for learning me everything about the appropriate dress code at conferences during that week. I also participated in the 2017 edition of ILAS in Ames, Iowa. It was definitely not a metropolitan environment, but we did found out about that one bar Ames had to offer, and how else would I ever have experienced living history in Iowa? I had a splendid time over all. The SIAM conference on applied linear algebra in May 2018 truly was a personal highlight. It was of great scientific (ánd touristic) interest. Hong Kong turned out to be a fascinating city. Later that year, in July 2018, I had a sun-drenched week during the NASCA conference in Kalamata (Greece). Thanks Koen for joining me in cheering for Belgium in the world cup. In my final year I attended the ETNA25 conference in Santa Margherita di Pula (Italy) immediately after finishing the first draft of the text that you're currently reading. I was very glad to ultimately get to know David Watkins a bit better. It was also the first time I went to a conference abroad together with both Raf and Niel simultaneously. In the words of Oasis: "Thank you for the good times"! The ICIAM in Valencia was the final conference during my PhD term and it was by far the largest scientific gathering that I attended. It was a lot of fun to be in the company of a large delegation from Leuven. I'd like to thank Stefan Güttel for the interesting discussion and for letting me speak in your mini-symposium. I commit to finally finishing the research that is overdue because of me spending nearly all my time on my thesis. The evening out in the karaoke bar with Alessandra, Roel, and Simon, singing some songs in English in between all these Spanish hits, was super entertaining.

This thesis would not have happened without Ben. Thank you for letting me know that Raf was interested in taking on a new student. I promise that I will

not follow you in your footsteps now, at least not immediately. Also thanks to Dries, Karel and Tuur for the fun nights out.

Ik sluit graag af door de mensen te bedanken die het dichtst bij me staan. Mijn familie heeft me altijd gesteund in al mijn (acadamische) doelen. Vooral mijn ouders zijn hierin onmisbaar geweest. Mama en Papa, dank u voor de hulp bij het maken van enkele moeilijke beslissingen zoals het kiezen van een gepaste opleiding en universiteit. Dank u om me de mogelijkheid te bieden om nog een diploma te behalen na de klassieke periode van vijf jaar. Dank u voor alle zaken, klein of groot, die ik nu vergeet op te sommen. Maren, Astrid, Laurids: bedankt! Ook mijn grootouders hebben me altijd warm omringd. Bedankt Oma, Driek en Ette. Een speciaal woord van dank gaat uit naar mijn grootvader Piet aan wie ik deze thesis opdraag. Mijn interesse in wetenschap is tijdens mijn jeugd voortdurend door je gestimuleerd. Je bijkomende wiskunde oefeningen waren vaak uitdagend en hebben me op menig examen uitstekend voorbereid. En natuurlijk Ine, dank je om er voor me te zijn wanneer nodig, voor je liefde, geluk en geduld. Ik kan niet wachten om erachter te komen welke avonturen ons nog te wachten staan!

Thank you all,

Daan Camps.

August 2019

Abstract

The matrix eigenvalue problem is often encountered in scientific computing applications. Although it has an uncomplicated problem formulation, the best numerical algorithms devised to solve it are far from obvious.

Computing all eigenvalues of a small to medium-sized matrix is nowadays a routine task for an algorithm of implicit QR-type using a bulge chasing technique. On the other hand projection methods are often used to compute a subset of the eigenvalues of sparse, large-scale eigenproblems. Krylov subspace methods are probably among the most used methods within this class.

The convergence of the classical implicit QR and Krylov subspace methods is determined by polynomials. The lion's share of this thesis is concerned with QR-type methods whose convergence is governed by the more general class of rational functions.

The first numerical scheme we present is the rational QZ method for the generalized eigenvalue problem. It uses a pole swapping technique on Hessenberg pencils. We provide an implicit Q theorem for Hessenberg pencils which motivates the pole swapping approach as it shows that the rational QZ iterates are unique. Rational Krylov theory allows us to prove that the rational QZ method implicitly performs subspace iteration accelerated by rational functions. An exactness result is included which shows that, in exact arithmetic, a pole swapping algorithm deflates a perfect shift in a single iteration. Numerical experiments exemplify that novel rational shifting strategies significantly reduce the computational cost compared to their polynomial counterparts. Furthermore, we propose a novel reduction algorithm to Hessenberg form and show that premature middle deflations can already be induced during the reduction phase provided a good choice of poles is made. Finally, a new swapping algorithm is introduced and an error analysis is provided which shows that the algorithm is backward stable.

In the subsequent chapter recent developments for polynomial QR-type methods

are adapted to the rational QZ method. This results in a multishift, multipole rational QZ method with tightly-packed shifts and poles. Aggressive early deflation is included to detect converged eigenvalues before classical deflation criteria are able to do so. Our implementation is made publicly available and numerical experiments demonstrate that it outperforms LAPACK in terms of accuracy, speed and empiric time complexity.

A rational QR method is proposed as a special case of the rational QZ method which treats the standard eigenvalue problem in an efficient manner. This method applies a pole swapping algorithm on Hessenberg, unitary Hessenberg pencils where only $O(n)$ storage space is required for the unitary matrix. Consequently, the storage requirements and computational cost is approximately half of the rational QZ method. Numerical experiments show that the pole swapping algorithm can outperform a bulge chasing method from LAPACK by reducing the CPU time by more than 30%.

Two-sided pole swapping algorithms for tridiagonal pencils are studied in the penultimate chapter. These result in the rational LR algorithm for unsymmetric pencils and the rational TT^{T} algorithm for symmetric, diagonalizable pencils. These algorithms have a reduced computational cost of $O(n^2)$ thanks to the tridiagonal structure but employ non-unitary equivalence transformations such that numerical stability is no longer guaranteed. We provide optimality results for the non-unitary transformations. Numerical experiments show promising results but also show that numerical stability is nontrivial.

The last chapter of the thesis considers the well-known rational Krylov method for the solution of large-scale eigenproblems. We show how eigenvalue estimates obtained with the rational Krylov method can be computed with the rational QZ method. Furthermore, we test the pole swapping technique to efficiently filter and restart the rational Krylov method.

Beknopte samenvatting

Het matrix eigenwaardeprobleem is vaak voorkomend in wetenschappelijk rekenen. Alhoewel de probleemstelling eenvoudig is, zijn de beste numerieke algoritmes verre van voor de hand liggend.

Het berekenen van alle eigenwaarden van een kleine tot middelgrote matrix is tegenwoordig een routine taak voor een impliciet QR algoritme dat gebruik maakt van een *bulge chasing* techniek. Voor grote, ijle eigenwaardeproblemen worden vaak projectiemethodes gebruikt om een deelverzameling van de eigenwaarden te berekenen. Krylov deelruimtemethodes zijn waarschijnlijk bij de meest gebruikte methodes binnen deze klasse.

Het convergentiegedrag van zowel het klassieke impliciet QR algoritme als van Krylov deelruimtemethodes wordt bepaald door veeltermen. Het leeuwendeel van deze thesis houdt zich bezig met QR-type methodes waarvoor het convergentiegedrag bepaald wordt door meer algemene rationale functies.

Het eerste numerieke schema dat we voorstellen is de rationale QZ methode voor veralgemeende eigenwaardeproblemen. De methode maakt gebruik van een poolpermutatietechniek voor Hessenberg matrix paren. We formuleren een impliciete Q stelling voor Hessenberg matrix paren die een motivering geeft voor de poolpermutatie aanpak vermits ze aantoont dat de rationale QZ iteraties uniek zijn. Rationale Krylov theorie stelt ons in staat om te bewijzen dat de rationale QZ methode impliciet een deelruimte-iteratie uitvoert die versneld wordt door rationale functies. Een nauwkeurigheidsstelling toont aan dat, in oneindige precisie, poolpermutatiemethodes met een perfecte *shift* in een enkele iteratie tot een deflatie leiden. Numerieke experimenten tonen aan dat de rationale QZ methode leidt tot een significante vermindering van het rekenwerk in vergelijking met de klassieke QZ methode. Verder formuleren we ook een nieuw algoritme om een matrix paar te reduceren tot Hessenbergvorm en tonen we dat een goede keuze van polen kan leiden tot voortijdige middendeflaties tijdens de reductiefase. Ten slotte stellen we een nieuw permutatie algoritme voor

en voorzien we een foutenanalyse die aantoont dat deze methode achterwaarts stabiel is.

Het daaropvolgende hoofdstuk past recente ontwikkelingen voor het QR algoritme aan voor de rationale QZ methode. Dit resulteert in een *multishift, multipool* rationale QZ methode met dicht op elkaar gepakte *shifts* en polen. Bovendien maken we gebruik van aggressieve voortijdige deflatie om geconvergeerde eigenwaarden te detecteren vooraleer klassieke deflatie criteria ze kunnen detecteren. Onze implementatie van het algoritme is publiek beschikbaar gemaakt en de numerieke testen tonen aan dat het LAPACK kan overtreffen op vlak van nauwkeurigheid, snelheid en empirische tijdscomplexiteit.

Verder stellen we een rationale QR methode voor als een specificatie van de rationale QZ methode die het standaard eigenwaardeprobleem op efficiënte wijze behandelt. Deze methode past een poolpermutatie-algoritme toe op Hessenberg, unitaire Hessenberg matrix paren en vereist slechts $O(n)$ opslagruimte voor de unitaire matrix. Dit halveert de opslagruimte en de rekenkost in vergelijking met de rationale QZ methode. Numerieke testen demonstreren dat het poolpermutatie-algoritme een *bulge chasing* methode van LAPACK kan overtreffen door de rekentijd met meer dan 30% te reduceren.

Tweezijdige poolpermutatie-algoritmes voor tridiagonale matrix paren worden bestudeerd in het voorlaatste hoofdstuk. Dit resulteert in het rationale LR algoritme voor niet-symmetrische paren en het rationale TT^{T} voor symmetrische, diagonaliseerbare paren. Deze algoritmes hebben een verminderde rekenkost van $O(n^2)$ dankzij de tridiagonale structuur maar ze maken gebruik van niet-unitaire equivalentie transformaties waardoor numerieke stabiliteit niet gegarandeerd kan worden. We voorzien optimaliteitsvoorwaarden voor de niet-unitaire transformaties. Numerieke experimenten geven veelbelovende resultaten maar tonen ook dat numerieke stabiliteit in dit geval niet triviaal is.

Het laatste hoofdstuk van de thesis gaat over de welbekende rationale Krylov methode for grootschalige eigenwaardeproblemen. We tonen aan hoe benaderende eigenwaarden, verkregen met behulp van de rationale Krylov methode, met de rationale QZ methode berekend kunnen worden. Verder testen we ook de poolpermutatietechniek om op een efficiënte manier de rationale Krylov methode te filteren en herstarten.

List of Symbols

Mathematical notation

$\alpha, \beta, \ldots$	Scalars
$\boldsymbol{a}, \boldsymbol{b}, \ldots$	Vectors
$A, B, \ldots$	Matrices
$\underline{A}, \underline{B}, \ldots$	Matrices with one more row than columns
$\mathcal{A}, \mathcal{B}, \ldots$	Subspaces
$a_{i,j}$	Element at position (i, j) of A
$\boldsymbol{a}_i$	ith column of A
A_i	First i columns of A
$A(i{:}j, k{:}\ell)$	Submatrix of A from rows i to j, and columns k to ℓ
$\mathcal{R}(\boldsymbol{a}_1, \ldots \boldsymbol{a}_n)$	Subspace generated by vectors $\boldsymbol{a}_1$ to $\boldsymbol{a}_n$
$(\alpha_1, \ldots, \alpha_n)$	Tuple of size n
$\{\alpha_1, \ldots, \alpha_n\}$	Multiset of size n
$\bar{\cdot}$	Complex conjugate

Special constants

$I, I_n, I_{n\times m}$	Identity matrix of unspecified size, of size $n\times n$, and of size $n\times m$
$\boldsymbol{e}_i$	ith canonical basis vector, ith column of I
$\mathcal{E}_i$	Subspace generated by the first i canonical basis vectors

$\mathbb{R}, \mathbb{C}, \mathbb{F}$	Field of real numbers, field of complex numbers, either real or complex number field
$\mathbb{R}^n, \mathbb{C}^n, \mathbb{F}^n$	n-dimensional vector space over $\mathbb{R}, \mathbb{C}, \mathbb{F}$
$\mathbb{R}^{n\times m}, \mathbb{C}^{n\times m}, \mathbb{F}^{n\times m}$	Set of matrices of dimension $n\times m$ over $\mathbb{R}, \mathbb{C}, \mathbb{F}$
$\bar{\mathbb{C}}$	Complex plane extended with the point at infinity
$\mathcal{P}_m$	Vector space of polynomials of degree $\leq m$
$\imath$	Imaginary unit
ϵ_m	Machine precision

Matrix operations and properties

A^{-1}	Inverse of A
$A^{\dagger}$	Moore-Penrose pseudoinverse of A
A^T	Transpose of A
A^*	Conjugate transpose of A
$\det(A)$	Determinant of A
$\operatorname{rank}(A)$	Rank of A
$\operatorname{diag}(\alpha_1, \ldots, \alpha_n)$	Diagonal matrix with diagonal entries $\alpha_1, \ldots, \alpha_n$
$\operatorname{diag}(A_1, \ldots, A_m)$	Block diagonal matrix with diagonal blocks $A_1, \ldots, A_m$

Other Symbols

$O(\cdot)$	Big-O notation

Contents

Chapter 1

Introduction

The matrix eigenvalue problem plays a central role in many computational problems encountered in science and engineering. The scalar λ is called an *eigenvalue* of the square matrix A if there exists a nonzero vector $\boldsymbol{x}$ such that,

$$A\boldsymbol{x} = \lambda\boldsymbol{x}.$$

The vector $\boldsymbol{x}$ is correspondingly called an *eigenvector* of A.

The numerical solution of the matrix eigenvalue problem has been an active area of research in numerical linear algebra, at least since the mid-twentieth century. Figure 1.1 shows the increase in number of publications per year that are listed in Web of Science [1] on the topic of eigenvalues.

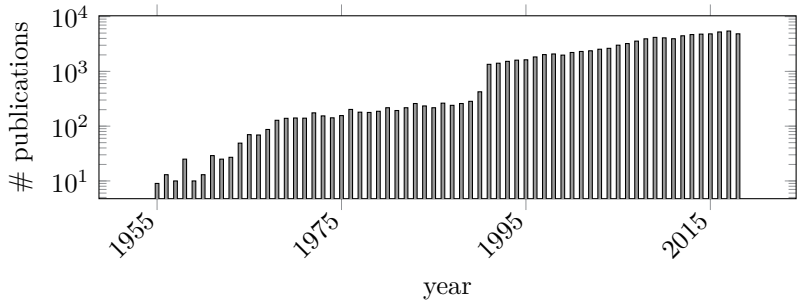


Figure 1.1: Number of publications per year on the topic of *eigenvalues* since 1955. Data downloaded from Web of Science [1] on February 9, 2019.

A few of the corresponding Web of Science categories include "(applied) mathematics", "electrical and electronic engineering", "mathematical physics","optics", "acoustics", and even "computer science artificial intelligence". This brief meta-analysis demonstrates that the eigenvalue problem has been a well-studied problem for many decades and will likely remain so for the foreseeable future.

The remainder of this chapter consists of two parts. Section 1.1 gives a concise historical overview of the development of numerical methods for the eigenvalue problem. Section 1.2 presents the plan for the thesis.

1.1 A concise historical overview

In 1829, Cauchy proved that a symmetric matrix can be orthogonally diagonalized and that all its eigenvalues are real [21]. Although somewhat arbitrary, Hawkins [53] argues that this is the beginning of the modern day study of matrix eigenvalue problems. Cauchy himself used the term *racine caractéristique* instead of eigenvalue. The derived name *characteristic polynomial* for $p(\lambda) = \det(A - \lambda I)$ is still used at present. The term eigenvalue dates back to the beginning of the 20th century and is attributed to German mathematician Hilbert who used the terminology *eigenfunktion* and *eigenwert* in a study of linear integral equations [56].

With the advent of digital computers in the 20th century, the study of numerical methods for the solution of the matrix eigenvalue problem increased exponentially, see Figure 1.1.

A first major breakthrough is due to Rutishauser. He invented the quotient-difference (qd) algorithm [98–100] to find zeros of polynomials or poles of rational functions. He noticed that the rhombus update rules for the qd transformation admit the following matrix interpretation,

$$\hat{L}\hat{R} = RL. \tag{1.1}$$

In the qd algorithm L is a bidiagonal, lower triangular matrix and R a bidiagonal upper triangular matrix. Rutishauser soon noticed that the iterative rule (1.1) of factoring a matrix as a product of an upper and lower triangular matrix, reversing their order, and multiplying them together can also be applied to general dense matrices. This resulted in his LR algorithm [101]. More information on the early developments that lead Rutishauser from the qd to the LR algorithm is available in [50].

The second major breakthrough is attributed to both Francis [39, 40] and Kublanovskaya [69]. Both authors independently proposed the implicitly shifted

QR algorithm. This algorithm displays better numerical stability properties than the LR algorithm as it uses unitary transformations. Furthermore, the implicit double-shift approach allows for real arithmetic in real-valued problems.

In its simplest form, the explicit QR algorithm applies the following iterative rule:

$$A = QR \quad \rightarrow \quad \hat{A} = RQ. \tag{1.2}$$

As Q is unitary, $\hat{A} = Q^*AQ$ which means that $\hat{A}$ is unitarily similar to A and thus has the same eigenvalues. Repeating the iterative rule (1.2) over and over, it is nearly certain that the matrix converges to upper triangular form with the eigenvalues of the original matrix readily available on the diagonal.

The practical QR algorithm, which is derived from the results of Francis and Kublanovskaya, is at present still the method of choice for computing all eigenvalues of a small to medium-sized dense matrix. It bears little resemblance to the simple iterative procedure (1.2). We go into more detail on what makes the QR algorithm both stable and efficient in Chapter 2.

For more historical details on the development of the QR method, we refer to the review papers by Watkins [133, 134, 139], Parlett [88] and Golub [45]. For a complete treatment of the most important developments in numerical methods for the matrix eigenvalue in the past decades, we refer to the monographs [66, 138].

1.2 Overview of the thesis

The thesis is organized as follows.

Chapter 1 provides a high-level problem setting, gives a short historical overview, and presents the plan for the thesis.

Chapter 2 mostly contains preliminary material. In this chapter we introduce relevant properties and decompositions for the matrix eigenvalue problem. We also provide an overview of Krylov subspace methods and of QR-type and QZ-type methods for standard and generalized eigenvalue problems. We pay special attention to the connection between these seemingly different classes of methods and discuss that their convergence is determined by polynomials.

Chapter 3 presents the *rational QZ method* for the numerical solution of the generalized eigenvalue problem. This chapter begins with a study of Hessenberg

pairs and their properties. Two operations to change the *poles* of a Hessenberg pair are introduced: pole introduction and pole swapping. We present a backward stable pole swapping algorithm. These two operations are used to formulate a direct reduction method to a Hessenberg pair with prescribed poles and to formulate the rational QZ method. Numerical experiments show that the reduction algorithm can induce *premature middle deflations* which can significantly reduce the computational cost of the overall algorithm. It is also observed that the rational QZ method can outperform the polynomial QZ method by effectively reducing the required number of iterations provided a good choice of poles is made. The implicit rational QZ method is motivated by a theoretical analysis which reveals a connection with rational Krylov. This connection is exploited to prove an implicit Q theorem showing that the rational QZ iterates are unique. It also allows us to prove that the rational QZ method implicitly performs nested subspace iteration accelerated by rational functions. Finally, we provide an exactness result, which shows that, in exact arithmetic, a rational QZ step deflates an eigenvalue in a single iteration.

The majority of this chapter is based on the article [19]:

Camps D., Meerbergen K., and Vandebril R., A rational QZ method. (2019) SIAM J. Matrix Anal. Appl. Vol. 40, No. 3, pp. 943–972.

The discussion on the backward stable algorithms to compute the swapping transformations in Section 3.3.2 is based on [16]:

Camps D., Mach T., Vandebril R., and Watkins D. S., On pole-swapping algorithms for the eigenvalue problem. (2019) Submitted.

This article by Watkins and coauthors also studies the effect of a single pole swap and shows how they can be combined to achieve convergence results for any pole swapping method.

Chapter 4 generalizes the rational QZ method of Chapter 3 to the multishift, multipole rational QZ method for block Hessenberg pencils. We also incorporate the aggressive early deflation strategy into the algorithm and use it both at the top-left and bottom-right sides of the pencil. Special attention is paid to the swapping transformations involving 2×2 blocks. We present the results of some numerical experiments obtained with our Fortran package `libRQZ` which implements the described algorithms. The numerical experiments illustrate that our algorithm is able to outperform LAPACK [2] in terms of accuracy, speed and empirical time complexity. The software is made publicly available on:

`http://numa.cs.kuleuven.be/software/rqz`

This chapter is based on the paper [18]:

Camps D., Meerbergen K., and Vandebril R., A multishift, multipole rational QZ method with aggressive early deflation. (2019) Submitted.

Section 4.4 is partially based on [17]:

Camps D., Mastronardi N., Vandebril R., and Van Dooren P., Swapping 2×2 blocks in the Schur and generalized Schur form. (2019) Accepted for publication in J. Comput. Appl. Math.

Chapter 5 presents a specification of the rational QZ method for Hessenberg, unitary Hessenberg pencils that requires approximately half the storage space and computational cost of the dense rational QZ method from Chapter 3. This efficiency gain is achieved by using a compact representation of the unitary matrix in terms of *core transformations*. We show how the pole swapping operations are implemented such that the compact representation is accurately preserved throughout the algorithm. The main purpose of this algorithm is to enable the use a pole swapping algorithm for the standard eigenvalue without the need of explicitly storing two dense $n \times n$ matrices. We refer to it as the *rational QR method* if the algorithm is used in this sense. Numerical experiments show a significant reduction in CPU time compared to one of the QR routines from LAPACK [2].

This chapter is based on a paper which is currently in preparation:

Camps D., Mach T., Vandebril R., and Watkins D. S., Pole swapping methods for Hessenberg, unitary Hessenberg pencils: Rational QR algorithms. In preparation.

Chapter 6 presents a class of two-sided pole swapping algorithms for (block) tridiagonal pencils. These methods make use of non-unitary transformations required to preserve the (block) tridiagonal structure throughout the algorithm. We discuss how the pole manipulation operations can be carried out using non-unitary transformations that are (nearly) optimally scaled. We provide uniqueness and convergence results for the tridiagonal case using the rational Krylov theory of Chapter 3. For the class of diagonalizable, symmetric (block) tridiagonal pencils we propose a *rational TT^T (or T3) algorithm* which uses

congruence transformations. For unsymmetric tridiagonal pencils we present a *rational LR* algorithm which allows us to independently manipulate the lower and upper pole tuples in order to induce convergence of eigenvalues at the subdiagonal or superdiagonal, respectively.

This chapter is based on a paper which is currently in preparation:

Camps D., Vandebril R., and Van Dooren P., Two-sided rational LR iterations for tridiagonal pencils. In preparation.

Chapter 7 focuses on the iterative rational Krylov method for large-scale eigenproblems. We provide three contributions. Firstly, we show how Ritz values obtained with the rational Krylov method can be computed with the rational QZ method. Secondly, we study the structure of the Galerkin projection on a rational Krylov basis. Thirdly, we test the pole swapping technique to implicitly filter and restart the rational Krylov method and make a comparison with existing methods which shows the superiority of this approach.

This chapter is based on the paper [20]:

Camps D., Meerbergen K., and Vandebril R., An implicit filter for rational Krylov using core transformations. (2019) Linear Algebra and its Applications. Volume 561, 15 January 2019, Pages 113-140.

Chapter 8 concludes the thesis by summarizing the main findings and results and provides an outlook for future research directions.

Appendix A discusses core transformations and the extended Hessenberg form. Extended Hessenberg matrices and pencils come into play in extended QR and QZ methods, which can be viewed as an intermediate step, in terms of generality, between polynomial and rational QR and QZ methods.

Appendix B provides the detailed error analysis for the backward stable pole swapping methods of Section 3.3.2 and numerical evidence that the new approach is more accurate than existing methods. This material is based on [16].

Publicly available software

Reference implementations for most of the algorithms discussed in this thesis are made available online on the webpage:

`http://numa.cs.kuleuven.be/software/rqz`

The most notable contribution is the Fortran implementation of the multishift, multipole rational QZ method with aggressive early deflation as part of the `libRQZ` package. We are aware that this is still research code which is in no way of the same quality as robust implementations like these provided by LAPACK [2]. It is our hope that it can prove to be valuable for further development of eigenvalue solvers for the dense, generalized eigenvalue problem.

We intend to also provide reference implementations of the algorithms that are discussed in Chapter 5 and Chapter 6 as soon as the articles are finished.

Chapter 2

Krylov and QR eigenvalue methods

This chapter has two main goals. The first objective is to give a thorough introduction to the properties and decompositions related to both the standard and generalized eigenvalue problems. The second goal is to introduce two types of numerical algorithms for the eigenvalue problem: Krylov subspace methods and QR/Z-type methods. We pay special attention to the connection between both.

The chapter is organized as follows. Section 2.1 gives an introduction to invariant subspaces and Schur decompositions for the standard eigenvalue problem and to deflating subspaces and the generalized Schur decomposition for the generalized eigenvalue problem. This material is inspired by the excellent introductions to (generalized) eigenvalue problems presented in the monographs [46, 66, 138]. Section 2.2 introduces Krylov subspace methods for the solution of large-scale eigenvalue problems. Section 2.3 discusses the implicit QR and QZ methods and highlights the connections with Krylov-type methods. Section 2.4 gives a conclusion by summarizing the key results.

2.1 Properties and decompositions

2.1.1 The standard eigenvalue problem

The eigenvalues of a matrix $A \in \mathbb{F}^{n\times n}$, with $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$, are the n roots of its *characteristic polynomial* $p(\lambda) = \det(A - \lambda I)$. We call the multiset,

$$\Lambda(A) = \{\lambda \in \mathbb{C} : \det(A - \lambda I) = 0\}, \tag{2.1}$$

the spectrum of A. In general, a root λ of $p(\lambda)$ can have a higher multiplicity and occur more than once in the multiset $\Lambda(A)$. The multiplicity of λ as a root of $p(\lambda)$ is called the *algebraic* multiplicity of the eigenvalue. An eigenvalue with algebraic multiplicity 1 is called *simple*.

If $\lambda \in \Lambda(A)$ then $A - \lambda I$ is a singular matrix. This implies that there exist nonzero vectors $\boldsymbol{x}$ that satisfy $A\boldsymbol{x} = \lambda\boldsymbol{x}$. This was our original problem formulation at the beginning of Chapter 1. The vector $\boldsymbol{x}$ is then called a (*right*) eigenvector. A *left eigenvector* $\boldsymbol{x}^*$ satisfies similarly $\boldsymbol{x}^*A = \lambda\boldsymbol{x}^*$. The subspace $\mathcal{S}_\lambda \subseteq \mathbb{F}^n$ generated as the linear span of all eigenvectors related to a single eigenvalue λ is called the *eigenspace* of λ. The dimension of $\mathcal{S}_\lambda$ is called the *geometric multiplicity* of λ. It is equal to the dimension of the nullspace of $A - \lambda I$. The algebraic multiplicity of an eigenvalue is larger than or equal to the geometric multiplicity. If the algebraic multiplicity of λ is strictly larger then the geometric multiplicity, the eigenvalue is called *defective*.

A subspace $\mathcal{X} \subseteq \mathbb{F}^n$ is called a *(right) invariant subspace* of A if $A\mathcal{X} \subseteq \mathcal{X}$. If $\mathcal{X}$ is of dimension k and we consider the matrix $X \in \mathbb{F}^{n\times k}$ whose columns form a basis for the k-dimensional invariant subspace $\mathcal{X}$, then there exists a matrix $A|_X \in \mathbb{F}^{k\times k}$ such that,

$$AX = XA|_X. \tag{2.2}$$

It follows from (2.2) that $\Lambda(A|_X) \subseteq \Lambda(A)$. The matrix $A|_X$ is unique and is called the *representation* of A with respect to X. An explicit expression for $A|_X$ is given by

$$A|_X = X^\dagger AX. \tag{2.3}$$

The matrix $X^\dagger$ is called the *Moore-Penrose pseudoinverse* of X and because X is of maximal rank k, $X^\dagger = (X^*X)^{-1}X^*$ [46] from which it follows that $X^\dagger X = I_k$ and (2.3) is immediate.

In case the invariant subspace under consideration is the entire n-dimensional vector space, i.e. $\mathcal{X} = \mathbb{F}^n$, we have that $X \in \mathbb{F}^{n\times n}$ is nonsingular and consequently $A|_X = X^{-1}AX$. The matrices A and $A|_X$ are called *similar* in this case and the transformation is called a *similarity* transformation. A

similarity transformation has the property that it preserves the eigenvalues, $\Lambda(A|_X) = \Lambda(A)$, and can be regarded as a change of basis.

If A, X, and $A|_X$ satisfy (2.2) then there exists a unitary matrix $Q \in \mathbb{F}^{n\times n}$ such that,

$$Q^*AQ = \begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix}, \tag{2.4}$$

with $T_{11} \in \mathbb{F}^{k\times k}$ and $\Lambda(T_{11}) = \Lambda(A|_X)$. The matrix $Q = [Q_1\, Q_2]$ can be computed from the QR factorization of X with Q_1 a unitary basis for $\mathcal{X}$ and Q_2 for its orthogonal complement. This idea can be used to go from the block triangular form of (2.4) to the triangular form of the *Schur decomposition* using inductive reasoning.

Theorem 2.1.1 (Schur Decomposition)**.** *Let $A \in \mathbb{F}^{n\times n}$. Then there exists a unitary matrix $Q \in \mathbb{F}^{n\times n}$ such that,*

$$Q^*AQ = T, \tag{2.5}$$

where $T \in \mathbb{F}^{n\times n}$ is an upper triangular matrix having the eigenvalues of A on its diagonal. The unitary similarity transformation Q can be chosen such that the eigenvalues appear in any order on the diagonal of T.

As $\mathbb{R}$ is not algebraically closed, it follows from (2.1) that a real-valued matrix can have eigenvalues with nonzero imaginary part. However, these complex eigenvalues do come in conjugate pairs $\alpha \pm \imath\beta$. The Schur decomposition (2.5) of a real matrix will thus be complex unless the matrix only has real eigenvalues. In order to avoid the computationally more expensive complex arithmetic and because the problem is a real-valued problem, the *real Schur decomposition* is typically used for real-valued matrices.

Theorem 2.1.2 (real Schur Decomposition)**.** *Let $A \in \mathbb{R}^{n\times n}$. Then there exists an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that,*

$$Q^TAQ = T = \begin{bmatrix} T_{11} & T_{12} & \dots & T_{1k} \\ 0 & T_{22} & \ddots & T_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & T_{kk} \end{bmatrix}, \tag{2.6}$$

where the diagonal blocks T_{ii}, i=1, …, k, are of dimension 1×1 for real eigenvalues and 2×2 for complex conjugate eigenvalues. The orthogonal similarity transformation Q can be chosen such that the eigenvalue blocks appear in any order on the diagonal of T.

Assume $\lambda{=}\alpha{+}\imath\beta$ is a complex eigenvalue of $A \in \mathbb{R}^{n\times n}$ with complex eigenvector $\boldsymbol{x} = \boldsymbol{x}_1 + \imath\boldsymbol{x}_2$, $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$. Then by construction,

$$A(\boldsymbol{x}_1 + \imath\boldsymbol{x}_2) = (\alpha + \imath\beta)(\boldsymbol{x}_1 + \imath\boldsymbol{x}_2) = (\alpha\boldsymbol{x}_1 - \beta\boldsymbol{x}_2) + \imath(\beta\boldsymbol{x}_1 + \alpha\boldsymbol{x}_2).$$

Rearranging the equation in matrix terms gives,

$$A \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 \end{bmatrix} \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}.$$

Notice that the above is a relation of the form (2.2) and that $\boldsymbol{x}_1$ must be linear independent of $\boldsymbol{x}_2$ as otherwise $\beta = 0$. This implies that $X = [\boldsymbol{x}_1\ \boldsymbol{x}_2]$ is a basis for the invariant subspace related with eigenvalues $\lambda, \bar{\lambda}$. Using a change of basis from X to an orthonormal basis of the invariant subspace and applying decompositions of the form (2.4), one can prove the existence of the real Schur form (2.6).

The purpose of the QR algorithm is to compute the Schur form (2.5) for complex-valued matrices or the real Schur form (2.6) for real-valued matrices. Before we turn our attention to computational methods, we first introduce the *generalized eigenvalue problem*.

2.1.2 The generalized eigenvalue problem

The generalized eigenvalues of a pair of matrices $A, B \in \mathbb{F}^{n\times n}$ are denoted as $\Lambda(A, B)$ and defined by,

$$\Lambda(A, B) = \{\lambda = \alpha/\beta \in \bar{\mathbb{C}} : \det(\beta A - \alpha B) = 0\}, \tag{2.7}$$

with $\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$. The matrix pair is denoted as (A, B) and is often also referred to as a *matrix pencil* $A - \lambda B$. We will use both terms interchangeably. Analogous to the standard eigenvalue problem, $\boldsymbol{x}$ is called a (*generalized*) right eigenvector corresponding to $\lambda = \alpha/\beta$ if,

$$\beta A\boldsymbol{x} = \alpha B\boldsymbol{x}. \tag{2.8}$$

Computing the pairs $(\lambda = \alpha/\beta, \boldsymbol{x})$ is the objective of the generalized eigenvalue problem. If (2.8) is, for a given $\boldsymbol{x}$, satisfied for $\beta = 0$, then the corresponding eigenvalue of (A, B) is located at ∞.

Throughout this thesis we assume that the pair (A, B) is *regular* which means that its characteristic polynomial $\det(A{-}\lambda B)$ differs from zero. Indeed, a matrix pair (A, B) can have a characteristic polynomial that vanishes everywhere. One such case occurs if A and B have a common nullspace. In that case there exists a

nonzero vector $\boldsymbol{x}$ for which $A\boldsymbol{x} = B\boldsymbol{x} = \mathbf{0}$ which implies that $(\beta A - \alpha B)\boldsymbol{x} = \mathbf{0}$ for all values α, β and the characteristic polynomial is equal to the zero polynomial. Pencils that are not regular are called *singular*. For regular pencils $\det(A - \lambda B)$ has exactly n roots in $\bar{\mathbb{C}}$ counting multiplicities.

In case B is a nonsingular matrix, (2.8) can be transformed to $B^{-1}A\boldsymbol{x} = \lambda\boldsymbol{x}$ and we end up with a standard eigenvalue problem for which the theory of Section 2.1.1 applies and the QR method can be used. The same is possible when A is nonsingular. However, from a numerical point of view this is a poor idea as B can be nonsingular but still be ill-conditioned with respect to inversion. If that is the case, then computing the eigenvalues of $B^{-1}A$ can be sensitive even when the generalized eigenvalues themselves are well-conditioned. This motivates the study of the generalized eigenvalue problem (2.8).

An important concept is the notion of a pair of *deflating* subspaces. This can be seen as the generalization of the invariant subspace (2.2). A pair of k-dimensional subspaces $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{F}^n$ is called a *pair of right deflating subspaces* of the regular $n \times n$ matrix pair (A, B) if for any matrices $X, Y \in \mathbb{F}^{n \times k}$ whose columnspace are respectively equal to $\mathcal{X}$ and $\mathcal{Y}$, we have that:

$$(A, B)Y = X(A, B)|_{(X,Y)}. \tag{2.9}$$

Here, the shorthand notation $(A, B)Y = (AY, BY)$ is used and $(A, B)|_{(X,Y)}$ is the unique $k \times k$ matrix pair:

$$(A, B)|_{(X,Y)} = X^{\dagger}(A, B)Y.$$

It follows that $\Lambda((A, B)|_{(X,Y)}) \subseteq \Lambda(A, B)$. A subspace $\mathcal{Y} \subseteq \mathbb{F}^n$ is part of a right deflating pair for the regular pair (A, B) if and only if [113, Definition 5.1],

$$\dim(A\mathcal{Y} + B\mathcal{Y}) = \dim(\mathcal{Y}). \tag{2.10}$$

In the case that $\mathcal{X}, \mathcal{Y}$ are equal to the entire n-dimensional vector space $\mathbb{F}^n$, we have that $X, Y \in \mathbb{F}^{n \times n}$ and $(A, B)|_{(X,Y)} = X^{-1}(A, B)Y$. In this case, the matrix pairs $(A, B)|_{(X,Y)}$ and (A, B) are called *equivalent*. The transformation is called an *equivalence* transformation and it preserves the eigenvalues: $\Lambda((A, B)|_{(X,Y)}) = \Lambda(A, B)$.

A pair of subspaces is called *left deflating* for (A, B) if it is right deflating for (A^*, B^*).

If (A, B), X, Y, and $(A, B)|_{(X,Y)}$ satisfy (2.9) then there exist unitary matrices $Q, Z \in \mathbb{F}^{n \times n}$ such that,

$$Q^*(A, B)Z = \left(\begin{bmatrix} S_{11} & S_{12} \\ & S_{22} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix} \right), \tag{2.11}$$

where (S_{11}, T_{11}) is a $k \times k$ matrix pair having $\Lambda(S_{11}, T_{11}) = \Lambda((A,B)|_{(X,Y)})$. The matrix $Q = [Q_1\ Q_2]$ can be computed from the QR factorization of X with Q_1 a unitary basis for $\mathcal{X}$ and Q_2 for $\mathcal{X}^\perp$. Similarly, the matrix $Z = [Z_1\ Z_2]$ can be computed from the QR factorization of Y with Z_1 a unitary basis for $\mathcal{Y}$ and Z_2 for $\mathcal{Y}^\perp$.

We can go from the block triangular form of (2.11) to the triangular form of the *generalized Schur decomposition* using inductive reasoning.

Theorem 2.1.3 (generalized Schur decomposition)**.** *Let (A, B) be an $n \times n$ regular matrix pair. Then there exist unitary matrices $Q, Z \in \mathbb{F}^{n \times n}$ such that,*

$$Q^*(A,B)Z = (S,T), \tag{2.12}$$

with (S, T) an $n \times n$ regular, upper triangular matrix pair having the eigenvalues of (A, B) as the ratios of its diagonal elements s_{ii}/t_{ii}, $i = 1, \ldots, n$. The unitary equivalence transformation can be chosen such that the eigenvalues appear in any order on the diagonals of (S, T).

For real-valued matrix pairs, the *real generalized Schur decomposition* is typically used to avoid complex arithmetic for complex-conjugate pairs of eigenvalues.

Theorem 2.1.4 (real generalized Schur decomposition)**.** *Let (A, B) be an $n \times n$ real-valued, regular matrix pair. Then there exist orthonormal matrices $Q, Z \in \mathbb{F}^{n \times n}$ such that,*

$$Q^T(A,B)Z = (S,T) = \left(\begin{bmatrix} S_{11} & S_{12} & \ldots & S_{1m} \\ 0 & S_{22} & \ddots & S_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & S_{mm} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} & \ldots & T_{1m} \\ 0 & T_{22} & \ddots & T_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & T_{mm} \end{bmatrix} \right), \tag{2.13}$$

where the diagonal subpencils (S_{ii}, T_{ii}), $i = 1, \ldots, m$ are of dimension 1×1 or 2×2 and correspond with respectively the real and complex conjugate eigenvalues of (A, B). The orthonormal equivalence transformation can be chosen such that the eigenvalues appear in any order along the diagonal.

A final useful definition is the notion of a *zero* of a (rectangular) $n_1 \times n_2$ matrix pencil $A - \lambda B$. This is also known as a *Smith zero* [41,121] or an *invariant zero* of a linear time invariant system in systems and control theory [37].

We limit our explanation to the case of pencils of *full normal rank*, more detailed characterizations of the zero structure of general pencils can be found in [41,121]. A pencil $A - \lambda B$ is said to be of full normal rank if the generic rank of $A - \lambda B$ is $\min(n_1, n_2)$. Equivalently this means that the *Kronecker indices* of $A - \lambda B$ are zero [121].

Definition 2.1.5. Let $A - \lambda B$ be an $n_1 \times n_2$ matrix pencil of full normal rank. A scalar $\zeta \in \bar{\mathbb{C}}$ is called a zero of $A - \lambda B$ if $\text{rank}(A - \zeta B) < \min(n_1, n_2)$.

For regular, square pencils the zeros and eigenvalues coincide.

2.2 Krylov subspace methods

We have characterized the standard and generalized eigenvalue problems in the previous section and discussed the existence of an eigenvalue revealing (generalized) Schur decomposition. Thus far we have ignored the problem of how to compute these decompositions and a satisfactory answer to this question will only be provided in Section 2.3. The central topic in this section are Krylov subspaces and the accompanying Krylov subspace methods for the eigenvalue problem such as the Arnoldi [3] and Lanczos [73] methods. From a numerical linear algebra perspective, the Arnoldi, Lanczos, and derived methods are most often used to compute a selected subset of eigenvalues of a large, often sparse, matrix. We will discuss why Krylov subspace methods are useful for large-scale eigenvalue problems. Our second motivation to study them is the central role they play in the theoretical analysis of the QR method in Section 2.3.

Suppose the matrix $A \in \mathbb{F}^{n\times n}$ is *nondefective* meaning that it does not have any defective eigenvalues. This means that it has n linear independent eigenvectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ that are a basis of $\mathbb{F}^n$ and that A can be diagonalized. Furthermore assume that A has a dominant eigenvalue such that the eigenvalues can be sorted as $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$. Let $\boldsymbol{v} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i \in \mathbb{F}^n$ then

$$A^k \boldsymbol{v} = \sum_{i=1}^{n} \lambda_i^k \alpha_i \boldsymbol{x}_i = \lambda_1^k \sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda_1}\right)^k \alpha_i \boldsymbol{x}_i,$$

for $k = 1, 2, \ldots$. It follows that the sequence of vectors,

$$\boldsymbol{v}, A\boldsymbol{v}, A^2\boldsymbol{v}, A^3\boldsymbol{v}, \ldots, \tag{2.14}$$

converges to the dominant eigenvector $\boldsymbol{x}_1$ of A with a linear convergence rate $|\lambda_2/\lambda_1| < 1$ under the mild assumption that $\alpha_1 \neq 0$. This idea is the basis for the simple *power method* to compute the dominant eigenvector and eigenvalue of A.

The vectors in the sequence (2.14) are called the *Krylov vectors* and the linear span of all vectors combined is known as a *Krylov subspace* which forms the basis for many iterative methods in numerical linear algebra. They are named after Russian naval engineer and applied mathematician Alexei Krylov who first introduced the idea in 1931 [68].

Definition 2.2.1 (Krylov subspace)**.** The mth order Krylov subspace generated by the matrix $A \in \mathbb{F}^{n\times n}$ and nonzero starting vector $\boldsymbol{v} \in \mathbb{F}^n$ is denoted by $\mathcal{K}_m(A, \boldsymbol{v})$ and defined as,

$$\mathcal{K}_m(A, \boldsymbol{v}) = \mathcal{R}\left(\boldsymbol{v}, A\boldsymbol{v}, \ldots, A^{m-1}\boldsymbol{v}\right). \tag{2.15}$$

Definition 2.2.1 introduced the $\mathcal{R}(\cdot)$-operator which represents the subspace generated by the linear span of its arguments. These arguments can be either vectors, as in (2.15), or a matrix in which case $\mathcal{R}(A)$ is considered as the linear span of columns of A, i.e. its column space.

The subspace $\mathcal{K}_m(A, \boldsymbol{v})$ is equal to the subspace of all vectors $p_{m-1}(A)\boldsymbol{v}$ with $p_{m-1} \in \mathcal{P}_{m-1}$, the vector space of all polynomials of degree not greater than $m-1$. For this reason, we also refer to (2.15) as a *polynomial* Krylov subspace.

There exists a positive integer g, $n \geq g \geq 1$, called the *grade* of $\boldsymbol{v}$ with respect to A, for which $\dim(\mathcal{K}_g(A, \boldsymbol{v})) = \dim(\mathcal{K}_{g+1}(A, \boldsymbol{v})) = g$. If $\boldsymbol{v}$ is an eigenvector of A, then $g = 1$. On the other hand, g is not greater than n since $n+1$ vectors in $\mathbb{F}^n$ are linear dependent. When the grade is reached

$$A^g\boldsymbol{v} = \sum_{i=1}^{g-1} \alpha_i A^i \boldsymbol{v},$$

for some coefficients α_i. This can be rewritten as

$$p(A)\boldsymbol{v} := A^g - \sum_{i=1}^{g-1} \alpha_i A^i \boldsymbol{v} = \boldsymbol{0},$$

where p is the *minimal polynomial of* $\boldsymbol{v}$ *with respect to* A[1]. The grade g is, by construction, equal to the degree of this minimal polynomial of $\boldsymbol{v}$ with respect to A [78].

Krylov subspaces form a sequence of strictly nested subspaces up until order g,

$$\mathcal{K}_1 \subset \mathcal{K}_2 \subset \ldots \subset \mathcal{K}_g = \mathcal{K}_{g+1} = \ldots = \mathcal{K}_n, \tag{2.16}$$

and they become an invariant subspace of A at order g.

Lemma 2.2.2 (Properties of Krylov subspaces)**.** *Let* $A \in \mathbb{F}^{n\times n}$, $\boldsymbol{v} \in \mathbb{F}^n \setminus \{\boldsymbol{0}\}$, *and* m *a strictly positive integer. Krylov subspaces satisfy the following elementary properties:*

[1]This is different from the characteristic polynomial of A, which we introduced at the beginning of Section 2.1.1 and is often also referred to as the *minimal polynomial* of A. The grade g is not greater than the degree of the characteristic polynomial of A.

I. Scale invariance: *For* $\alpha, \beta \neq 0$*,*

$$\mathcal{K}_m(A, \boldsymbol{v}) = \mathcal{K}_m(\alpha A, \beta \boldsymbol{v}), \tag{2.17}$$

II. Shift invariance: *For any scalar* ϱ*,*

$$\mathcal{K}_m(A, \boldsymbol{v}) = \mathcal{K}_m(A + \varrho I, \boldsymbol{v}), \tag{2.18}$$

III. Change of basis: *Given a nonsingular matrix* $X \in \mathbb{F}^{n\times n}$*,*

$$\mathcal{K}_m(A, \boldsymbol{v}) = X\mathcal{K}_m(X^{-1}AX, X^{-1}\boldsymbol{v}). \tag{2.19}$$

IV. Expansion:

$$A\mathcal{K}_m(A, \boldsymbol{v}) \subseteq \mathcal{K}_{m+1}(A, \boldsymbol{v}). \tag{2.20}$$

The proof of the first, third and fourth property is trivial using Definition 2.2.1. The second property is a corollary of the isomorphism between $\mathcal{K}_m(A, \boldsymbol{v})$ and $\{p_{m-1}(A)\boldsymbol{v} : p_{m-1} \in \mathcal{P}_{m-1}\}$.

An evident matrix whose column space is a Krylov subspace is the *Krylov matrix*:

Definition 2.2.3 (Krylov matrix)**.** The mth order Krylov matrix generated by $A \in \mathbb{F}^{n\times n}$ and $\boldsymbol{v} \in \mathbb{F}^n \setminus \{\mathbf{0}\}$ is denoted by $K_m(A, \boldsymbol{v})$ and defined as,

$$K_m(A, \boldsymbol{v}) = \left[\boldsymbol{v}, A\boldsymbol{v}, \ldots, A^{m-1}\boldsymbol{v}\right]. \tag{2.21}$$

Properties I and II from Lemma 2.2.2 do not translate to Krylov matrices, but property III is valid for Krylov matrices:

$$K_m(A, \boldsymbol{v}) = XK_m(X^{-1}AX, X^{-1}\boldsymbol{v}), \tag{2.22}$$

and also property IV can be reformulated for a Krylov matrix in the following sense [120]:

$$AK_m(A, \boldsymbol{v}) = K_{m+1}(A, \boldsymbol{v})\overline{I}_{(m+1)\times m}, \tag{2.23}$$

with,

$$\overline{I}_{(m+1)\times m} = \begin{bmatrix} \mathbf{0}^T \\ I_m \end{bmatrix} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{m+1} \end{bmatrix} \in \mathbb{F}^{(m+1)\times m}, \tag{2.24}$$

an $m\times m$ identity matrix prepended with an additional row of zeros.

2.2.1 Orthonormal Krylov bases and Hessenberg matrices

From a numerical point of view, computing the Krylov matrix as a basis for a Krylov subspace is a poor idea. The vectors $A^i\boldsymbol{v}$ tend to converge to the direction of the dominant eigenvector as we have seen previously. The matrix $K_m(A, \boldsymbol{v})$ can rapidly become ill-conditioned as a result. A frequently used strategy in numerical linear algebra is to use an orthonormal basis instead. For unsymmetric matrices this results in the Arnoldi method [3] and the Lanczos method [73] is the variant for symmetric problems. Our focus is on the unsymmetric case and we will give a brief introduction of the method of Arnoldi. First, we study the matrix structure that (orthogonal) bases of Krylov subspaces impose. To this end let us first define the *upper Hessenberg* matrix which plays a central role in the QR method(s) discussed in this thesis.

Definition 2.2.4 (upper Hessenberg matrix)**.** A matrix $H \in \mathbb{F}^{n\times n}$ is called an *upper Hessenberg* matrix if it has no nonzero entries below its first subdiagonal. An upper Hessenberg matrix is called *unreduced* or *proper* if $h_{i+1,i} \neq 0$ for $i = 1, \ldots, n-1$.

An example of a proper Hessenberg matrix is given by the leading $m\times m$ part of $\overline{I}_{(m+1)\times m}$ in (2.24). In the following theorem, we present some further results on the connection between Krylov matrices and upper Hessenberg matrices.

Theorem 2.2.5. *The following relations between Krylov matrices and proper Hessenberg matrices hold:*

I. Let $A \in \mathbb{F}^{n\times n}$ and $\boldsymbol{v} \in \mathbb{F}^n \setminus \{\boldsymbol{0}\}$ be a vector with grade $g \leq n$ with respect to A. For $m = 1, \ldots, g$, consider the thin QR decomposition of the mth order Krylov matrix,

$$K_m(A, \boldsymbol{v}) = Q_m R_m \tag{2.25}$$

Then $H_m = Q_m^* A Q_m \in \mathbb{F}^{m\times m}$ is a proper upper Hessenberg matrix.

II. On the other hand, let $H \in \mathbb{F}^{n\times n}$ be a proper Hessenberg matrix, then for $m = 1, \ldots, n$,

$$K_m(H, \boldsymbol{e}_1) = \tilde{R}_m \in \mathbb{F}^{n\times m}, \tag{2.26}$$

is an upper triangular matrix with nonzero diagonal entries, i.e. $\tilde{r}_{ii} \neq 0$ for $i = 1, \ldots, m$.

Proof. We first prove part I. As $m \leq g$ the rank of $K_m(A, \boldsymbol{v})$ is m and R_m is a nonsingular upper triangular having $r_{ii} \neq 0$ for $i = 1, \ldots, m$. Denote,

$$K_{m+1}(A, \boldsymbol{v}) = Q_{m+1} R_{m+1} = \begin{bmatrix} Q_m & \boldsymbol{q}_{m+1} \end{bmatrix} \begin{bmatrix} R_m & \boldsymbol{r}_{m+1} \\ & r_{m+1,m+1} \end{bmatrix},$$

with $r_{m+1,m+1} \neq 0$ if and only if $m < g$. Plugging these expressions in (2.23) we get:

$$AQ_m R_m = \begin{bmatrix} Q_m & \boldsymbol{q}_{m+1} \end{bmatrix} \begin{bmatrix} R_m & \boldsymbol{r}_{m+1} \\ & r_{m+1,m+1} \end{bmatrix} \overline{I}_{(m+1)\times m}.$$

From which it follows that:

$$Q_m^* A Q_m = \begin{bmatrix} R_m & \boldsymbol{r}_{m+1} \end{bmatrix} \overline{I}_{(m+1)\times m} R_m^{-1}.$$

It remains to verify that the matrix on the right-hand side is indeed proper upper Hessenberg. Using (2.24), we have that:

$$\begin{aligned} \begin{bmatrix} R_m & \boldsymbol{r}_{m+1} \end{bmatrix} \overline{I}_{(m+1)\times m} R_m^{-1} &= \begin{bmatrix} R_m & \boldsymbol{r}_{m+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_2 \dots \boldsymbol{e}_{m+1} \end{bmatrix} R_m^{-1} \\ &= \begin{bmatrix} R_m(1{:}m, 2{:}m) & \boldsymbol{r}_{m+1} \end{bmatrix} R_m^{-1}, \end{aligned}$$

which is a proper Hessenberg matrix because R_m is nonsingular upper triangular. Right multiplication with the upper triangular matrix R_m^{-1} preserves the upper Hessenberg structure.

Part II trivially holds for $m = 1$. Assume, as an induction hypothesis, that (2.26) holds up to index $i-1$, with $1 \leq i \leq n-1$. For index i we get:

$$K_i(H, \boldsymbol{e}_1)\boldsymbol{e}_i = H^{i-1}\boldsymbol{e}_1 = H(H^{i-2}\boldsymbol{e}_1) = HK_{i-1}(H, \boldsymbol{e}_1)\boldsymbol{e}_{i-1} = H\tilde{R}_m \boldsymbol{e}_{i-1}.$$

By the induction hypothesis, we have that $\tilde{R}_m \boldsymbol{e}_{i-1}$ is a vector with only nonzero entries in its first $i-1$ rows and with $\tilde{r}_{i-1,i-1} \neq 0$. Consequently, $H\tilde{R}_m \boldsymbol{e}_{i-1}$ is a linear combination of the first $i-1$ columns of H with a nontrivial component in column $i-1$ meaning that $\tilde{r}_{i,i} \neq 0$ by the properness of H. □

The following is a direct corollary of part II of Theorem 2.2.5 but rephrased in terms of the corresponding Krylov subspace.

Corollary 2.2.6. *Let $H \in \mathbb{F}^{n\times n}$ be a proper Hessenberg matrix, then for $m = 1, \dots, n$,*

$$\mathcal{K}_m(H, \boldsymbol{e}_1) = \mathcal{E}_m, \tag{2.27}$$

with $\mathcal{E}_m = \mathcal{R}(\boldsymbol{e}_1, \dots, \boldsymbol{e}_m)$.

2.2.2 Arnoldi's iterative method

Theorem 2.2.5 shows that orthonormal bases of Krylov subspaces impose a Hessenberg structure. The Arnoldi method [3], summarized in Algorithm 1, is an iterative procedure that gradually constructs an orthonormal basis of the

Krylov subspace $\mathcal{K}_{m+1}(A, \boldsymbol{v})$ along with the Hessenberg representation of A in the Krylov basis.

The Arnoldi method has been studied extensively for the use of large-scale eigenvalue problems. Notable results are due to Saad [102–105], Paige [86, 87], Cullum [23, 24], Walker [131], and Wilkinson [141].

Algorithm 1 Arnoldi's method [3]

Input: A, $\boldsymbol{v}$, m **Output:** V_{m+1}, $\underline{H}_m$

1: *Start:* $\boldsymbol{v}_1 \leftarrow \boldsymbol{v} / \|\boldsymbol{v}\|_2$
2: *Iterate:*
3: **for** $j = 1, \ldots, m$ **do**
4: $\quad \boldsymbol{v}_{j+1} \leftarrow A\,\boldsymbol{v}_j$ ▷ Matrix-vector product
5: $\quad$ **for** $i = 1, \ldots, j$ **do** ▷ modified Gram-Schmidt
6: $\quad\quad h_{i,j} \leftarrow \boldsymbol{v}_i^*\,\boldsymbol{v}_{j+1}$
7: $\quad\quad \boldsymbol{v}_{j+1} \leftarrow \boldsymbol{v}_{j+1} - h_{i,j}\boldsymbol{v}_i$
8: $\quad$ **end for**
9: $\quad h_{j+1,j} \leftarrow \|\boldsymbol{v}_{j+1}\|_2$ ▷ normalize
10: $\quad \boldsymbol{v}_{j+1} \leftarrow \boldsymbol{v}_{j+1} \,/\, h_{j+1,j}$
11: **end for**

The only time the matrix A is referenced during Algorithm 1 is in the matrix-vector product carried out in line 4 at the beginning of the main loop in the algorithm. Because of this, the Arnoldi method is well suited for sparse, large scale matrices and for structured matrices for which a matrix-vector product can be computed efficiently. The matrix does not need to be known explicitly to run the Arnoldi method. A function that returns the result $A\boldsymbol{v}$ for any vector $\boldsymbol{v}$ suffices.

The next computational part of the Arnoldi method is a pass of the modified Gram-Schmidt procedure which starts in line 5. The idea behind Gram-Schmidt is straightforward. Given two vectors $\boldsymbol{x}, \boldsymbol{y}$ of unit norm, the vector $\hat{\boldsymbol{y}} = \boldsymbol{y} - (\boldsymbol{x}^*\boldsymbol{y})\boldsymbol{x}$ is the orthogonalization of $\boldsymbol{y}$ with respect to $\boldsymbol{x}$. Indeed, $\boldsymbol{x}^*\hat{\boldsymbol{y}} = \boldsymbol{x}^*\boldsymbol{y} - (\boldsymbol{x}^*\boldsymbol{y})\boldsymbol{x}^*\boldsymbol{x} = 0$. Modified Gram-Schmidt, as stated in Algorithm 1, displays better numerically stability than the regular Gram-Schmidt procedure. Nonetheless, sometimes a second pass of modified Gram-Schmidt is required to ensure numerically orthogonal vectors in finite precision arithmetic [43]. The Gram-Schmidt orthogonalization has computational complexity $O(m^2 n)$. The growth is quadratic in the number of Krylov vectors. *Restarted* Krylov subspace methods [75, 111, 112] are often used in practice to limit the number of Krylov vectors thereby limiting the storage and computational requirements. We discuss restarted *rational* Krylov methods in detail in Chapter 7.

Taking a close look at Algorithm 1, it is not difficult to see that the following recurrence relationship is satisfied at iteration j,

$$A\boldsymbol{v}_j = \sum_{i=1}^{j+1} h_{i,j}\boldsymbol{v}_i. \tag{2.28}$$

Combining iterations $j = 1, \ldots, m$ of (2.28), we get the matrix relation,

$$AV_m = V_{m+1}\underline{H}_m, \tag{2.29}$$

which is the well-known Arnoldi recurrence relationship. The matrix $V_{m+1} \in \mathbb{F}^{n\times(m+1)}$ is the orthonormal basis of $\mathcal{K}_{m+1}(A, \boldsymbol{v})$ and $\underline{H}_m \in \mathbb{F}^{(m+1)\times m}$ is the upper Hessenberg representation of A in the orthonormal Krylov basis. The matrix $\underline{H}_m$ is constructed from the orthonormalization coefficients and is a proper upper Hessenberg matrix as long as the vector $\boldsymbol{v}_{j+1}$ in line 9 of Algorithm 1 is a nonzero vector. The pair $(V_{m+1}, \underline{H}_m)$ is referred to as an *Arnoldi pair* and is called *proper* if $\underline{H}_m$ is proper. The leading $m\times m$ part of $\underline{H}_m$ is denoted as H_m.

If $\|\boldsymbol{v}_{j+1}\|_2$ becomes numerically zero in line 9, the Arnoldi method experiences a *lucky breakdown*. As the name suggests, this is positive since the Arnoldi recurrence reduces to $AV_m = V_m H_m$ which is a relation of the form (2.2), implying that the Krylov subspace becomes an invariant subspace of A and the eigenvalues of H_m form a subset of $\Lambda(A)$.

A lucky breakdown occurs in exact arithmetic when m reaches the grade of A with respect to $\boldsymbol{v}$. In practice, Arnoldi's method is rarely continued until an invariant subspace is obtained. Instead eigenvalue approximations are extracted from Krylov subspaces of smaller dimension. This is the topic of the next part.

Galerkin and Petrov-Galerkin projections on Krylov subspaces

The Arnoldi method is a *projection method* in the sense that it projects the large-scale problem onto a lower-dimensional Krylov subspace and searches for approximate solutions in this lower-dimensional subspace. Two different projection conditions, known as the *Galerkin* and *Petrov-Galerkin* conditions, have been proposed in the literature. These conditions are formally defined in Definition 2.2.7, independent from the kind of problem and type of subspace.

Definition 2.2.7. Let $\mathcal{V}, \mathcal{W}$ be two m-dimensional subspaces of $\mathbb{F}^n$, $\boldsymbol{z} \in \mathcal{V}$ an approximate solution with residual vector $\boldsymbol{r}(\boldsymbol{z})$. Then $\boldsymbol{z}$ is called a Galerkin approximate solution if:

$$\boldsymbol{r}(\boldsymbol{z}) \perp \mathcal{V}, \tag{2.30}$$

and a Petrov-Galerkin approximate solution if:

$$\boldsymbol{r}(\boldsymbol{z}) \perp \mathcal{W}. \tag{2.31}$$

The Galerkin condition implies that the *solution* and *constraint* subspaces are both equal to the same subspace $\mathcal{V}$. This leads to an orthogonal projection method. In the Petrov-Galerkin condition, the solution subspace is $\mathcal{V}$ while the constraint subspace is some other subspace $\mathcal{W}$. This results in an oblique projection method [106].

Let us use Definition 2.2.7 to derive eigenvalue approximations based on the Arnoldi method (2.29). We will deduce two commonly used strategies to extract approximate eigenvalues from (2.29). Observe that when searching for approximate eigenvalues, the residual vector in Definition 2.2.7 reduces to:

$$\boldsymbol{r}(\boldsymbol{z}) = A\boldsymbol{z} - \vartheta\boldsymbol{z}, \tag{2.32}$$

with ϑ the approximate eigenvalue.

Lemma 2.2.8. *Given a proper Arnoldi pair* $(V_{m+1}, \underline{H}_m)$ *related to the Krylov subspace* $\mathcal{K}_{m+1}(A, \boldsymbol{v})$ *and satisfying an Arnoldi decomposition* (2.29)*. Imposing a Galerkin condition* (2.30) *with* $\mathcal{V} = \mathcal{K}_m(A, \boldsymbol{v})$ *leads to the* Ritz pairs $(\vartheta, \boldsymbol{z} = V_m\boldsymbol{y}_m)$ *characterized by the eigenvalue problem,*

$$H_m\boldsymbol{y}_m = \vartheta\boldsymbol{y}_m. \tag{2.33}$$

Imposing a Petrov-Galerkin condition (2.31) *with* $\mathcal{V} = \mathcal{K}_m(A, \boldsymbol{v})$*,* $\mathcal{W} = (A - \tau I)\mathcal{K}_m(A, \boldsymbol{v})$*,* $\tau \notin \Lambda(H_m)$*, leads to the* τ-harmonic Ritz pairs $(\vartheta, \boldsymbol{z} = V_m\boldsymbol{y}_m)$ *characterized by the eigenvalue problem,*

$$\tilde{H}_m^\tau\boldsymbol{y}_m = \vartheta\boldsymbol{y}_m, \tag{2.34}$$

with $\tilde{H}_m^\tau = H_m + |h_{m+1,m}|^2\boldsymbol{f}_m^\tau\boldsymbol{e}_m^T$*,* $\boldsymbol{f}_m^\tau = (H_m - \tau I_m)^{-*}\boldsymbol{e}_m$*.*

Proof. For the Galerkin approximation, we have $\boldsymbol{z} \in \mathcal{K}_m(A, \boldsymbol{v})$ thus $\boldsymbol{z} = V_m\boldsymbol{y}_m$ for some $\boldsymbol{y}_m \in \mathbb{C}^m$. Furthermore, we have the orthogonality constraint:

$$\begin{aligned}
&\boldsymbol{r}(z) \perp \mathcal{K}_m(A, \boldsymbol{v}) \\
\Leftrightarrow\ & AV_m\boldsymbol{y}_m - \vartheta V_m\boldsymbol{y}_m \perp V_m \\
\Leftrightarrow\ & V_m^*(V_{m+1}\underline{H}_m\boldsymbol{y}_m - \vartheta V_m\boldsymbol{y}_m) = \boldsymbol{0} \\
\Leftrightarrow\ & (H_m - \vartheta I_m)\boldsymbol{y}_m = \boldsymbol{0}.
\end{aligned} \tag{2.35}$$

The second equation explicitly rewrites the first in terms of the basis V_{m+1}, the third equation used (2.29) and rewrote the orthogonality constraint as an inproduct. The last equation follows immediately from the orthonormality of V_m. For the Petrov-Galerkin approximation, we still have $\boldsymbol{z} = V_m \boldsymbol{y}_m$, but the orthogonality constraint changes to:

$$\begin{aligned} & \boldsymbol{r}(z) \perp (A - \tau I)\mathcal{K}_m(A, \boldsymbol{v}) \\ \Leftrightarrow\ & AV_m \boldsymbol{y}_m - \vartheta V_m \boldsymbol{y}_m \perp (A - \tau I)V_m \\ \Leftrightarrow\ & V_{m+1}(\underline{H}_m - \vartheta \underline{I}_m)\boldsymbol{y}_m \perp V_{m+1}(\underline{H}_m - \tau \underline{I}_m) \\ \Leftrightarrow\ & (\underline{H}_m - \tau \underline{I}_m)^*(\underline{H}_m - \vartheta \underline{I}_m)\boldsymbol{y}_m = \boldsymbol{0}. \end{aligned} \tag{2.36}$$

Here, $\underline{I}_m$ is the $m \times m$ identity matrix appended with an additional row of zeros. The last equation corresponds to the small-scale generalized eigenvalue problem,

$$(\underline{H}_m - \tau \underline{I}_m)^* \underline{H}_m \boldsymbol{y}_m = \vartheta (\underline{H}_m - \tau \underline{I}_m)^* \underline{I}_m \boldsymbol{y}_m.$$

Left multiplication of both sides with[2] $(H_m - \tau I_m)^{-*}$ results in (2.34), considering that:

$$(H_m - \tau I_m)^{-*}(\underline{H}_m - \tau \underline{I}_m)^* = \begin{bmatrix} I_m & \bar{h}_{m+1,m}(H_m - \tau I_m)^{-*}\boldsymbol{e}_m \end{bmatrix}.$$

□

We refer to H_m as the *Galerkin projection* of A on $\mathcal{K}_m(A, \boldsymbol{v})$ and to $\tilde{H}_m^\tau$ as the *Petrov-Galerkin projection* of A on $(A - \tau I)\mathcal{K}_m(A, \boldsymbol{v})$. The corresponding eigenvalue approximations are referred to as respectively *Ritz* and *τ-harmonic Ritz* values. If the *target* τ is chosen at zero, the latter are called *harmonic Ritz* values. We remark that $\tilde{H}_m^\tau$ is a proper upper Hessenberg matrix as it only differs from H_m in its last column.

Ritz and τ-harmonic Ritz pairs often provide accurate approximations to some eigenvalues of A long before the theoretical grade is reached. The accuracy of the Ritz pairs can be assessed from their residual vector which, from (2.35), is,

$$\boldsymbol{r} = h_{m+1,m} \boldsymbol{v}_{m+1} \boldsymbol{e}_m^T \boldsymbol{y}_m, \tag{2.37}$$

and has norm $\|\boldsymbol{r}\|_2 = |h_{m+1,m}||\boldsymbol{e}_m^T \boldsymbol{y}_m|$. Ritz values tend to first converge to well-separated and extreme eigenvalues of A [70, 71].

Figure 2.1 illustrates the convergence of Ritz values with a numerical experiment on a 100×100 symmetric matrix with eigenvalues in $[11, 29] \cup \{39\}$.

[2]The condition $\tau \notin \Lambda(H_m)$ ensures that the inverse exists.

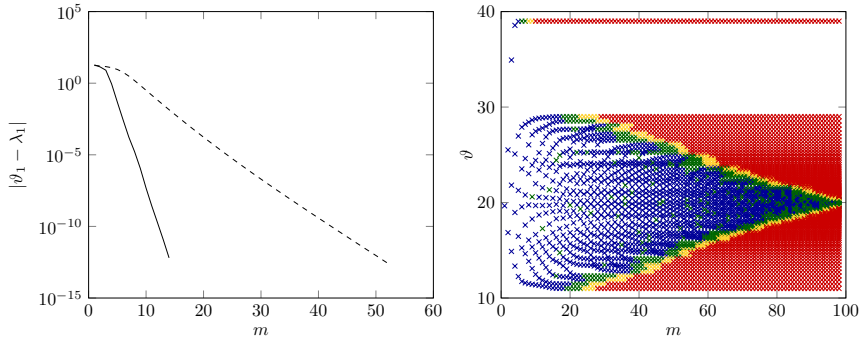


Figure 2.1: Convergence of the Arnoldi method compared with the convergence of the power method to the dominant eigenvalue *(left)* and convergence of all Ritz values in the Arnoldi method *(right)*.

The dashed line on the left side of Figure 2.1 shows the convergence of the power method to the dominant eigenvalue at 39 in function of the iteration step m. The convergence of the largest Ritz value ϑ_1 in function of the Krylov subspace dimension m to the dominant eigenvalue is shown with the full line. Both methods used the same starting vector. Using all information from the Krylov vectors significantly improves the convergence rate in comparison with the power method. The right hand side of Figure 2.1 shows a so-called *Ritz plot* that summarizes the convergence of all Ritz values in function of the subspace dimension m. The y-axis shows the value of the m Ritz values computed from the Krylov subspace of dimension m. The Ritz values themselves are indicated with $\times$-markers and their color indicates the accuracy of the Ritz value according to the color code in Table 2.1.

Table 2.1: Color code for Ritz plots

Accuracy $\lvert\lambda - \vartheta\rvert$	Color
$\left[\infty;\ 10^{-2.5}\right)$	blue
$\left[10^{-2.5};\ 10^{-5}\right)$	green
$\left[10^{-5};\ 10^{-7.5}\right)$	yellow
$\left[10^{-7.5};\ 0\right]$	red

The Ritz plot illustrates the typical convergence behavior: the extreme eigenvalue at 39 is found up to reasonable precision within 10 iterations and from $m \approx 20$ onwards the eigenvalues in the large cluster at $[11, 29]$ are found starting from the outside of the cluster and gradually converging to the interior eigenvalues.

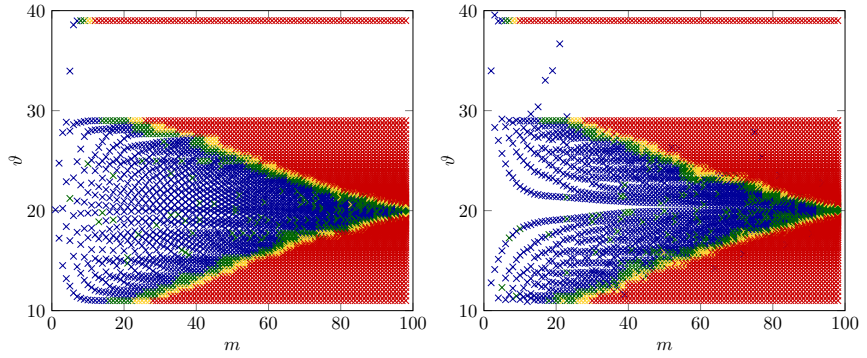


Figure 2.2: Convergence of harmonic Ritz values *(left)* and τ-harmonic Ritz values with $\tau = 20.25$ *(right)*.

The extreme eigenvalues are not always the eigenvalues of interest for the problem at hand and the (restarted) Arnoldi method might require a prohibitive number of iterations before converging to the desired eigenvalues. τ-Harmonic Ritz values have been shown to be able to provide more accurate approximations in the interior of the spectrum [85]. However, both Ritz and τ-harmonic Ritz values start from the same subspace $\mathcal{K}_{m+1}(A, \boldsymbol{v})$. Figure 2.2 shows a Ritz plot for harmonic and τ-harmonic approximations ($\tau = 20.25$, interior of spectrum) for the same matrix as Figure 2.1.

If, in the numerical example, we are interested in the eigenvalues in the neighborhood of 20, the Arnoldi method is impractical as it only converges to these eigenvalues for $m \approx n$. Computing the Ritz values requires in this case the solution of an eigenvalue problem with a dimension that is close to the original problem size.

The last decades, a lot of generalizations of the Arnoldi method were proposed that allow for faster convergence to a selected region of eigenvalues in the complex plane. We will discuss two of these generalizations, called the *extended* and *rational Krylov* methods in Chapter 7. In the next section we first study Francis' QR method that can be used to compute all eigenvalues of an upper Hessenberg matrix. This can, for example, be the Arnoldi Hessenberg matrix H_m.

2.3 The implicit QR method

The implicit QR algorithm [39, 40, 69] can be used to compute the Schur (2.5) or real Schur decomposition (2.6) of an upper Hessenberg matrix H. In order to compute the Schur decomposition of a general dense matrix $A \in \mathbb{F}^{n\times n}$ with the implicit QR algorithm, the matrix first needs to be *reduced* to an upper Hessenberg matrix via a similarity transformation. This reduction can, in theory, be computed with the Arnoldi method as it iteratively constructs an upper Hessenberg matrix that is, under the assumption that breakdown only occurs in step n, unitarily similar to A. This is never done in practice for reasons of numerical stability and because the computational cost is prohibitive. A better idea is to explicitly create zeros in A by a sequence of unitary similarity transformations. Section 2.3.1 reviews the *Householder* and *core* transformations that can be used for this task. Section 2.3.2 describes the implicit QR method in detail and Section 2.3.3 the implicit QZ method [83] to compute the generalized (real) Schur decomposition of a regular matrix pair (A, B).

2.3.1 Creating zeros in matrices

The first type of unitary transformation useful to introduce zeros in a matrix is the *Householder reflector*.

Definition 2.3.1 (Householder reflector)**.** Given a nonzero vector $\boldsymbol{v} \in \mathbb{F}^n$, the unitary matrix P given by:

$$P = I - \frac{2\boldsymbol{v}\boldsymbol{v}^*}{\boldsymbol{v}^*\boldsymbol{v}}, \tag{2.38}$$

is called a Householder reflector.

A Householder reflector is clearly Hermitian, $P = P^*$, and is also unitary, $P^*P = I$. For any nonzero vector $\boldsymbol{x} \in \mathbb{F}^n$ that is not a scalar multiple of $\boldsymbol{e}_1$ it is possible to construct an appropriate vector $\boldsymbol{v} \in \mathbb{F}^n$ such that the associated Householder reflector P has the property:

$$P\boldsymbol{x} = \sigma\|\boldsymbol{x}\|_2\boldsymbol{e}_1, \quad \text{with } |\sigma| = 1. \tag{2.39}$$

To achieve the desired result, the vector $\boldsymbol{v}$ needs be of the form,

$$\boldsymbol{v} = \boldsymbol{x} + \sigma\|\boldsymbol{x}\|_2\boldsymbol{e}_1, \tag{2.40}$$

with $|\sigma| = 1$. For reasons of numerical accuracy the choice $\sigma = \text{sign}(x_1)$ is made in most implementations.

Householder reflectors have the advantage that they can be easily computed based on (2.40) and that they can be applied to an arbitrary vector in $O(n)$ operations using (2.38) without the need to form P explicitly. Furthermore, they lead to backward stable algorithms given that they are unitary transformations. They can also be used to zero out part of a vector by embedding them in a larger matrix, e.g.,

$$Q = \begin{bmatrix} I_{n-k} & \\ & P_k \end{bmatrix}, \tag{2.41}$$

can be used to zero out the last $k-1$ entries of a vector of size n.

A second class of matrices that can be used to create zero elements in a matrix are called *core* transformations.

Definition 2.3.2 (Core transformation)**.** A core transformation, $C_i \in \mathbb{F}^{n\times n}$, acting on two consecutive rows i and $i+1$ is the embedding of a nonsingular 2×2 matrix at rows and columns i and $i+1$ of the identity matrix:

$$C_i = \begin{bmatrix} I_{i-1} & & & \\ & \times & \times & \\ & \times & \times & \\ & & & I_{n-i-1} \end{bmatrix}. \tag{2.42}$$

Throughout this thesis, we mostly consider unitary core transformations in which case the *active* 2×2 block in (2.42) can, for example, be chosen as a rotation matrix:

$$\begin{bmatrix} c & -\bar{s} \\ s & \bar{c} \end{bmatrix}, \quad \text{with,} \quad |c|^2 + |s|^2 = 1, \tag{2.43}$$

or as a small Householder reflector.

Given a vector $\boldsymbol{x} = [x_1\ x_2]^T \in \mathbb{F}^2$, it is always possible to compute a unitary core transformation C_1 such that $C_1\boldsymbol{x} = \|\boldsymbol{x}\|_2\boldsymbol{e}_1$. Left multiplication of an $n\times n$ matrix with a core transformation C_i only affects rows i and $i+1$ of the matrix, while right multiplication with the same core transformation only affects columns i and $i+1$. A commonly used graphical representation for core transformations is by means of a double-sided arrow pointing to the rows or columns on which it acts. Consider the following basic example:

$$\begin{bmatrix} \nearrow & \times & \times \\ \searrow & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times \\ & \times \end{bmatrix}.$$

In this example the core transformation introduces a zero in position $(2,1)$ of the 2×2 matrix which brings it to upper triangular form. This can be used to compute the QR decomposition.

The numerical properties of unitary core transformations are as favorable as these of Householder transformations. The major difference is that core transformations are typically used to create zeros in targeted positions, potentially without destroying existing structure. Householder transformations, on the other hand, are useful to create many zeros simultaneously and they are computationally more efficient for a generic, dense QR decomposition compared to core transformations [46, Section 5.2]. More details on core transformations are provided in Appendix A. Core transformations have been studied extensively in the context of QR-type methods [4, 123, 129, 130]. We will mainly make use of them in Chapter 5 as a way to efficiently represent a unitary Hessenberg matrix.

Unitary Hessenberg reduction

Let us now describe the process of constructing a unitary similarity transformation to reduce a generic dense matrix to Hessenberg form. Figure 2.3 shows the first step in the algorithm for a 5×5 matrix.

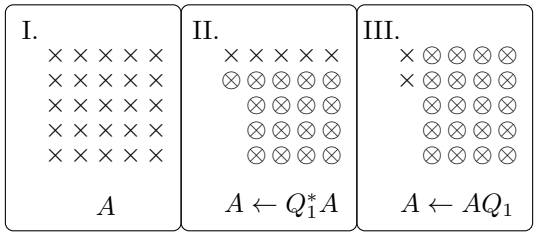


Figure 2.3: Unitary Hessenberg reduction method.

The algorithm starts with computing a Householder transformation Q_1^* that creates zeros in rows 3 to n of the first column of A. This operation is shown in pane II of Figure 2.3. The matrix elements that are changed under this operation are marked with $\otimes$. Since we require a similarity transformation to preserve the eigenvalues, the next step is a right multiplication with Q_1. Again the matrix elements that are changed by this transformation are highlighted with $\otimes$ in pane III of Figure 2.3. Observe that this does not destroy any zeros already created by Q_1^* such that we get a matrix that is similar to A but with its first column in upper Hessenberg form.

The reduction algorithm is continued in the same fashion for the remaining columns that are not yet Hessenberg form. In step $i = 1, \ldots, n-2$, the matrix elements in rows $i+2$ up to n of column i are set to zero by a Householder

reflector Q_i^* and the similarity is preserved by right multiplication with Q_i. This algorithm computes an overall similarity transformation:

$$H = \underbrace{Q_{n-2}^* \dots Q_1^*}_{Q^*} A \underbrace{Q_1 \dots Q_{n-2}}_{Q}. \tag{2.44}$$

This is the basic version of the Hessenberg reduction algorithm which has a computational cost of $O(10n^3/4)$ operations if only H is computed and $O(14n^3/4)$ operations if both H and Q are computed [46].

Variants of this algorithm that lead to more efficient implementations have been proposed in [28, 61, 91].

2.3.2 Implicit QR

Once our matrix is reduced to Hessenberg form, the iterative phase of the QR method can commence. Before discussing the implicit formulation of the algorithm, it is interesting to take a quick look at what the explicit (unshifted) QR step (1.2) looks like for a Hessenberg matrix. Figure 2.4 shows a graphical representation of a single iteration.

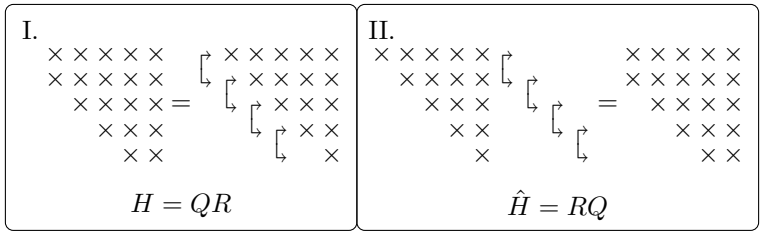


Figure 2.4: Explicit QR step on a Hessenberg matrix.

Pane I shows how the unitary matrix in the QR decomposition of an $n \times n$ Hessenberg matrix can be represented by a sequence of $n-1$ core transformations in a *descending order*. Such a representation representation always exists. Starting from a Hessenberg matrix H, we can compute core transformations C_1 up to C_{n-1} which create an upper triangular matrix

$$C_{n-1}^* \dots C_1^* H = R.$$

In this scheme C_i is computed such that it annihilates element $h_{i+1,i}$. If the Hessenberg matrix is proper, then all core transformations are nontrivial meaning that they have a nonzero off-diagonal entry.

In pane II the order of the upper triangular matrix and sequence of core transformations is reversed, just as in (1.2). Applying C_1 to the upper triangular is achieved by a multiplication with the first two columns of R, which creates a nonzero element in position $(2,1)$. Continuing this process, we again end up with an upper Hessenberg matrix.

The important conclusion is that *the Hessenberg structure is preserved under a QR step*. This means that the initial effort of creating zeros during the Hessenberg reduction is not in vain: where a single, explicit QR step (1.2) on a dense matrix requires $O(n^3)$ operations, this cost drops to $O(n^2)$ for Hessenberg matrices as their QR decomposition has the structure shown in Figure 2.4 and can be computed using $n-1$ core transformations.

Implicit use of shifts

To speed up convergence of the QR method, the Hessenberg matrix is typically shifted with some well-chosen shift ϱ. The explicit variant then becomes:

$$H - \varrho I = QR \quad \rightarrow \quad \hat{H} = Q^* H Q = RQ + \varrho I. \tag{2.45}$$

The computational cost of a single step in this iteration is still $O(n^2)$ since the Hessenberg structure is preserved under the degree-1 shift polynomial, i.e. $p_1(H) = H - \varrho I$ is an upper Hessenberg matrix that can be computed in $O(n)$ operations as only its diagonal elements differ from H.

Computing $p_m(H)$ for higher degree shift polynomials requires significantly more than $O(n)$ operations. Given m shifts, $\varrho_1, \ldots, \varrho_m$, the degree m shift polynomial,

$$p_m(H) = (H - \varrho_1 I) \ldots (H - \varrho_m I), \tag{2.46}$$

can be used in an explicit QR update as follows:

$$p_m(H) = QR \quad \rightarrow \quad \hat{H} = Q^* H Q. \tag{2.47}$$

The computation of $p_m(H)$ requires $m-1$ matrix-matrix products with Hessenberg matrices, resulting in a computational cost of $O(n^3)$ operations. This is too expensive for a practical algorithm. The approach of (2.46) and (2.47) has additional disadvantages: it is difficult to maintain real arithmetic for real-valued matrices and using sets of shifts that are closed under complex conjugation, it requires additional memory to store $p_m(H)$, and the upper Hessenberg form might not be accurately preserved in (2.47) in finite precision arithmetic.

Francis' implicit QR step [40] overcomes all of these disadvantages with an elegant solution. Assume we have a real-valued matrix H and a pair of complex-

conjugate shifts $\varrho, \bar{\varrho}$. The algorithm starts with computing the vector:

$$\boldsymbol{x} = p_2(H)\boldsymbol{e}_1 = (H - \bar{\varrho}I)(H - \varrho I)\boldsymbol{e}_1, \tag{2.48}$$

which is just the first column of (2.46) for $m = 2$. Observe that $\boldsymbol{x}$ only has nonzero elements in its first 3 rows thanks to the proper Hessenberg structure of H. It follows from part II of Theorem 2.2.5 that this is valid in general. With m shifts, the vector $\boldsymbol{x} = p_m(H)\boldsymbol{e}_1$, which is part of the column space of $K_{m+1}(H, \boldsymbol{e}_1)$, has nonzero entries in the first $m + 1$ rows only and can be computed based on the first m columns of H. The computational cost is negligible as long as $m \ll n$. Furthermore, in our example $\boldsymbol{x}$ is a real-valued vector since:

$$\overline{(H - \bar{\varrho}I)(H - \varrho I)} = (H - \varrho I)(H - \bar{\varrho}I) = (H - \bar{\varrho}I)(H - \varrho I). \tag{2.49}$$

The next step is to compute an orthonormal matrix Q_1 such that:

$$Q_1^T \boldsymbol{x} = \pm \|\boldsymbol{x}\|_2 \boldsymbol{e}_1. \tag{2.50}$$

This can be done with a small Householder reflector or with two core transformations. The details of how the transformation is computed, makes no mathematical difference as long as (2.50) is satisfied. In both cases, Q_1 is essentially a 3×3 matrix embedded in the first three rows and first three columns of an $n \times n$ identity matrix.

Now Q_1 is used to perform the initial similarity transformation $H_1 = Q_1^T H Q_1$. This process is shown in Figure 2.5 for a small scale example.

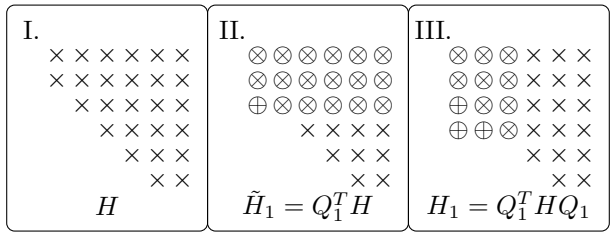


Figure 2.5: Introducing the perturbation in the Hessenberg matrix.

Left multiplication with Q_1^T introduces a nonzero element in position $(3, 1)$ of H. Right multiplication creates additional nonzero elements in positions $(4, 1)$ and $(4, 2)$. These 3 nonzero elements, indicated with + in pane I of Figure 2.6, constitute the *bulge* that has been introduced in H_1.

The remainder of the algorithm consists of *chasing* the bulge in order to restore the Hessenberg form. This is achieved by a selective Hessenberg reduction where

the transformations are carefully chosen in order not to introduce additional nonzero elements. The first step is shown in Figure 2.6 and consists of computing an orthonormal matrix Q_2 which restores the Hessenberg form in the first column of the matrix by left multiplication. This step is shown in pane II of Figure 2.6 where elements $(3,1)$ and $(4,1)$ are zeroed. A similarity transformation is required to preserve the eigenvalues, so in pane III of Figure 2.6 the matrix undergoes a right multiplication with Q_2. This creates two new nonzero elements in positions $(5,2)$ and $(5,3)$.

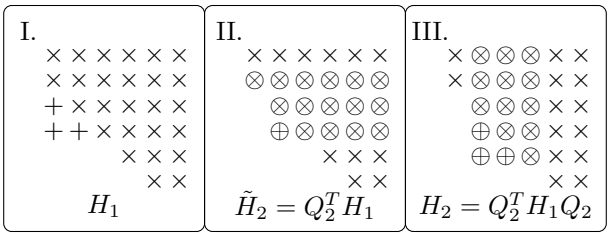


Figure 2.6: Chasing the bulge in the Hessenberg matrix.

We observe that the bulge has moved one position along the subdiagonal in the direction of the bottom-right corner of the matrix and that no additional nonzero entries have formed in H_2 compared to H_1. This procedure is continued until the bulge is eventually removed from the matrix in the bottom right corner. This constitutes a single QR *sweep* of multiplicity 2.

Uniqueness & convergence via a connection with Krylov subspaces

In this section, we discuss how an implicit QR sweep causes convergence of eigenvalues in the upper Hessenberg matrix H. Our discussion is greatly based on [138, 139].

The following well-known result on the essential uniqueness of the QR factorization of nonsingular matrices will be useful in the proof of the uniqueness of the QR method.

Lemma 2.3.3. *Let $A \in \mathbb{F}^{n\times n}$ be a nonsingular matrix and consider the QR factorizations $A = \hat{Q}\hat{R}$ and $A = \check{Q}\check{R}$. Then there exists a unitary diagonal matrix D such that $\hat{Q} = \check{Q}D$.*

Proof. We have,

$$\hat{Q}\hat{R} = \check{Q}\check{R} \;\Rightarrow\; \check{Q}^*\hat{Q} = \check{R}\hat{R}^{-1} \;\Rightarrow\; \check{Q}^*\hat{Q} = D.$$

Here the second equality follows from the first by using the unitarity of $\check{Q}$ and the nonsingularity of $\hat{R}$. The final equality follows from the property that $\check{R}\hat{R}^{-1}$ is an upper triangular matrix that must be unitary and hence is a diagonal matrix. □

From the discussion of the QR method in the previous section, it is clear that an implicit QR sweep with shift polynomial (2.46) constructs a similarity transformation, $\hat{H} = Q^*HQ$, with:

$$Q\boldsymbol{e}_1 = \gamma p_m(H)\boldsymbol{e}_1. \tag{2.51}$$

This follows from (2.48) and (2.50) and the observation that the subsequent chasing procedure does not alter $\boldsymbol{q}_1$.

The following theorem presents the well-known *implicit Q* theorem for proper Hessenberg matrices.

Theorem 2.3.4 (Implicit Q for proper Hessenberg matrices)**.** *Let $A \in \mathbb{F}^{n\times n}$ and let $\hat{Q}, \check{Q} \in \mathbb{F}^{n\times n}$ be unitary with $\hat{\boldsymbol{q}}_1 = \sigma\check{\boldsymbol{q}}_1$, $|\sigma| = 1$ such that,*

$$\hat{H} = \hat{Q}^*A\hat{Q}, \quad \textit{and,} \quad \check{H} = \check{Q}^*A\check{Q},$$

are both proper Hessenberg matrices. Then, $\hat{Q} = \check{Q}D$ and $\hat{H} = D^*\check{H}D$ for some unitary diagonal matrix D.

Proof. We have,

$$\begin{aligned}\hat{Q}K_n(\hat{H}, \boldsymbol{e}_1) &= \hat{Q}K_n(\hat{Q}^*A\hat{Q}, \boldsymbol{e}_1) = K_n(A, \hat{\boldsymbol{q}}_1)\\ &= \sigma K_n(A, \check{\boldsymbol{q}}_1) = \sigma\check{Q}K_n(\check{Q}^*A\check{Q}, \boldsymbol{e}_1)\\ &= \sigma\check{Q}K_n(\check{H}, \boldsymbol{e}_1).\end{aligned}$$

The second and fourth equalities applied (2.22). By part II of Theorem 2.2.5, the above equation shows the equality between two QR factorizations, $\hat{Q}K_n(\hat{H}, \boldsymbol{e}_1) = \sigma\check{Q}K_n(\check{H}, \boldsymbol{e}_1)$, with nonsingular upper triangular matrices. It follows from Lemma 2.3.3 that $\hat{Q} = \check{Q}D$ and consequently $\hat{H} = D^*\check{H}D$. □

Theorem 2.3.4 motivates the implicit bulge chasing approach of Francis' algorithm, the vector $\boldsymbol{q}_1$ is fixed once the shifts are chosen according to (2.51). The implicit Q theorem thus guarantees that the outcome of an implicit QR sweep, $\hat{H} = Q^*HQ$, is *essentially* unique once the shifts are determined. The essential uniqueness is up to multiplication with a unitary diagonal matrix D

which does not influence the convergence of the algorithm. This is valid under the assumption that the Hessenberg matrix remains proper.

If at some stage the matrix becomes improper in the sense that $h_{i+1,i} = 0$ for $1 \leq i < n$, this is in fact positive as it allows us to split the matrix in smaller independent problems:

$$\begin{matrix} i & n-i \\ \begin{bmatrix} H_{11} & H_{12} \\ & H_{22} \end{bmatrix} & \begin{matrix} i \\ n-i \end{matrix} \end{matrix}, \tag{2.52}$$

as the matrix becomes of block upper triangular form.

We have characterized the essential uniqueness of a QR sweep, let us now discuss the convergence of the method. The following lemma will be useful in the proof of the main result on convergence given in Theorem 2.3.6.

Lemma 2.3.5. *Given $A, Q, H \in \mathbb{F}^{n\times n}$ with Q unitary and such that $H = Q^*AQ$ is a proper Hessenberg matrix. Then for $m = 1, \ldots, n$,*

$$\mathcal{K}_m(A, \boldsymbol{q}_1) = Q\mathcal{E}_m. \tag{2.53}$$

Proof. Combining (2.19) and Corollary 2.2.6 gives:

$$\mathcal{K}_m(A, \boldsymbol{q}_1) = Q\mathcal{K}_m(Q^*AQ, Q^*\boldsymbol{q}_1) = Q\mathcal{K}_m(H, \boldsymbol{e}_1) = Q\mathcal{E}_m.$$

□

Theorem 2.3.6 (Polynomial acceleration for QR)**.** *Assume that $H, \hat{H} \in \mathbb{F}^{n\times n}$ are both proper Hessenberg matrices that are unitary similar, $\hat{H} = Q^*HQ$, with the similarity transformation obtained from a single implicit QR sweep with $\boldsymbol{q}_1$ determined by $p_m(H)$ as in* (2.51)*. Then, for $k = 1, \ldots, n$,*

$$Q\mathcal{E}_k = p_m(H)\mathcal{E}_k.$$

Proof. Using Lemma 2.3.5, the property that H commutes with itself and with the identity matrix, and Corollary 2.2.6, we get:

$$Q\mathcal{E}_k = \mathcal{K}_k(H, \boldsymbol{q}_1) = \mathcal{K}_k(H, p_m(H)\boldsymbol{e}_1) = p_m(H)\mathcal{K}_k(H, \boldsymbol{e}_1) = p_m(H)\mathcal{E}_k$$

□

This theorem allows us to interpret the QR sweep, $\hat{H} = Q^*HQ$, as an effective procedure for *polynomial accelerated nested subspace iteration* in combination with a change of basis [139].

Let us briefly describe what this means. Subspace iteration can be regarded as a higher dimensional extension of the power method (2.14). Given $A \in \mathbb{F}^{n\times n}$ and a k-dimensional subspace $\mathcal{X}_k \subset \mathbb{F}^n$, *polynomial accelerated subspace iteration* constructs the sequence of subspaces:

$$\mathcal{X}_k,\ p(A)\mathcal{X}_k,\ p(A)^2\mathcal{X}_k,\ p(A)^3\mathcal{X}_k,\ \ldots, \tag{2.54}$$

for some polynomial p. Assuming the eigenvalues of $p(A)$ are ordered as

$$|p(\lambda_1)| \geq \ldots \geq |p(\lambda_k)| > |p(\lambda_{k+1})| \geq \ldots \geq |p(\lambda_k)|,$$

the rate of convergence of $p(A)^i\mathcal{X}_k$ to the invariant subspace related to $\lambda_1, \ldots, \lambda_k$ is equal to $|p(\lambda_{k+1})|/|p(\lambda_k)|$. If the polynomial is large in magnitude at $\lambda_1, \ldots, \lambda_k$ and small otherwise, then convergence to the invariant subspace occurs rapidly. This convergence rate is potentially much faster than with a constant polynomial, i.e. unshifted subspace iteration.

Under the assumptions of Theorem 2.3.6, a QR sweep implicitly performs one step of *nested* polynomial accelerated subspace iteration on the special sequence of subspaces $\mathcal{E}_k$. This means that for $k = 1, \ldots, n$,

$$\mathcal{K}_k(H, \boldsymbol{e}_1) = \mathcal{E}_k \ \mapsto\ p_m(H)\mathcal{E}_k = Q\mathcal{E}_k.$$

The change of basis, $\hat{H} = Q^*HQ$, maps $p_m(H)\mathcal{E}_k$ back to $\mathcal{E}_k$.

Typically an adaptive shifting strategy is used in a practical bulge chasing algorithm. A common choice are the eigenvalues of a trailing submatrix of the Hessenberg matrix. Francis' implicit double-shift algorithm [39, 40] for real Hessenberg matrices uses the eigenvalues of the trailing 2-by-2 submatrix, $\left[\begin{smallmatrix} h_{n-1,n-1} & h_{n-1,n} \\ h_{n,n-1} & h_{n,n} \end{smallmatrix}\right]$, as shifts. The Wilkinson shift is equal to the Francis shift closest to $h_{n,n}$ and is often used in a single-shift algorithm [45].

Both strategies typically lead to quadratic convergence of eigenvalues and invariant subspaces for which the shift polynomial is chosen. Because of the change of basis, this invariance translates to a deflation in the Hessenberg matrix like in (2.52). The converged eigenvalues will split from the rest of the matrix in the trailing $k\times k$ block. Repeated application of this process gradually drives the matrix to (real) Schur form.

2.3.3 Implicit QZ

Where the QR method is the method of choice to compute the Schur (2.5) or real Schur (2.6) decomposition of a general unsymmetric matrix, the QZ method [83] is the method of choice to reduce a regular unsymmetric matrix pair to generalized Schur (2.12) or generalized real Schur (2.13) form.

The QZ method consists conceptually of 2 phases, just as the QR algorithm:

I. A direct reduction of the matrix pair (A, B) to an equivalent Hessenberg, triangular matrix pair (H, R).

II. An iterative phase during which deflating subspaces of the matrix pair (H, R) are determined and the matrix pair is essentially reduced to the triangular, triangular pair (S, T).

Various modifications and additions to the original algorithm have been proposed after its original introduction. Kaufman [62] added a deflation strategy and Ward [132] further refined various aspects of the method. Watkins & Elsner [140] generalized the QZ algorithm to a class of GZ iterations which make use of transformations that are not necessarily unitary.

In this section we review the two phases of the basic QZ method and describe the connection with the QR method.

Reduction to Hessenberg, triangular form

The Hessenberg, triangular reduction of a dense pair (A, B) starts with reducing B to upper triangular form via its QR decomposition: $(Q^*A, Q^*B = R)$. This first step is shown in pane I of Figure 2.7. Next, zeros are introduced in A without perturbing the upper triangular form of B too much. Notice that if we are too ambitious and set the entries in rows 3 to n in the first column of A to zero with a single Householder reflector then the update of B under the equivalence would destroy almost all zeros created in B. Instead, we introduce a single zero in position $(n, 1)$ of A by means of a core transformation acting on the last two rows. This is shown in pane II of Figure 2.7. This core transformation only introduces one additional nonzero element in position $(n, n-1)$ of B which is restored to zero with a core transformation acting on the last two columns, as shown in pane III of Figure 2.7.

This is continued by introducing zeros from the bottom to the top, row by row, in the first column of A until the first column is in upper Hessenberg form. Each zero that is introduced in A by a core transformation acting from the left subsequently introduces a nonzero element in B which is removed by a core transformation acting from the right. After the first column of (A, B) is in Hessenberg, triangular form, the process continues on the second column until the entire pencil is in Hessenberg, triangular form. In the end, we have computed a unitary equivalence transformation,

$$(H, R) = Q^*(A, B)Z. \tag{2.55}$$

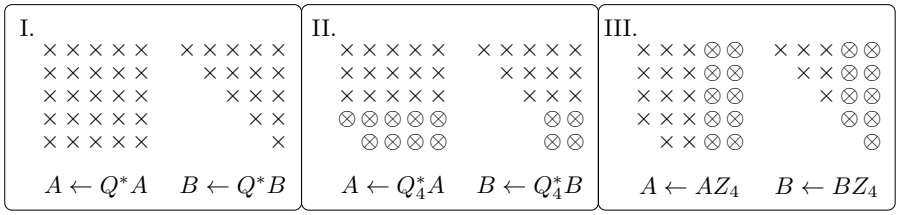


Figure 2.7: Start of unitary Hessenberg, triangular reduction method.

The computational cost of this algorithm is $O(8n^3)$ operations if only (H, R) is formed. Accumulating Q and Z requires an additional $O(7n^3)$ [46]. Variants of the basic reduction algorithm, which are more efficient on modern computer architectures, have been proposed in [25, 59]

Implicit QZ step

In the second phase, a bulge chasing algorithm is used to reduce the Hessenberg, triangular pencil (H, R) to (real) generalized Schur form. Similar to the QR method, this starts with computing and introducing a perturbation.

Assume (H, R) is a real-valued matrix pair and $\varrho, \bar{\varrho}$ are the complex-conjugate shifts that we want to use. We compute the vector:

$$\begin{aligned} \boldsymbol{x} &= p_2(HR^{-1})\boldsymbol{e}_1 \\ &= (HR^{-1} - \bar{\varrho}I)(HR^{-1} - \varrho I)\boldsymbol{e}_1 \\ &= (H - \bar{\varrho}R)R^{-1}(H - \varrho R)R^{-1}\boldsymbol{e}_1. \end{aligned} \tag{2.56}$$

Observe that HR^{-1} is an upper Hessenberg matrix and as such $\boldsymbol{x}$ only has nonzero elements in its first 3 rows. The computational cost to compute $\boldsymbol{x}$ is $O(1)$ because of the Hessenberg, triangular structure. Furthermore by (2.49), $\boldsymbol{x}$ is a real-valued vector in our example. The next step is to compute an orthonormal matrix Q_1 according to (2.50).

Figure 2.8 shows how Q_1 introduces a perturbation in $(\tilde{H}_1, \tilde{R}_1)$. Matrix entries that become nonzero under this transformations are indicated with a $\oplus$. The remainder of the implicit QZ step consists of restoring the Hessenberg, triangular form by chasing the perturbation along the subdiagonal.

The first bulge chasing step is visualized in Figure 2.9. In pane I, an orthonormal transformation matrix Z_1 is computed which restores the upper triangular

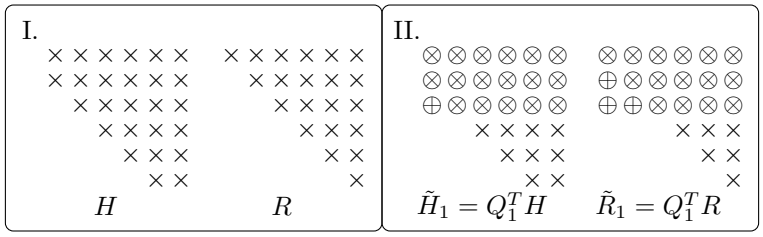


Figure 2.8: Introducing the perturbation in the Hessenberg, triangular pencil.

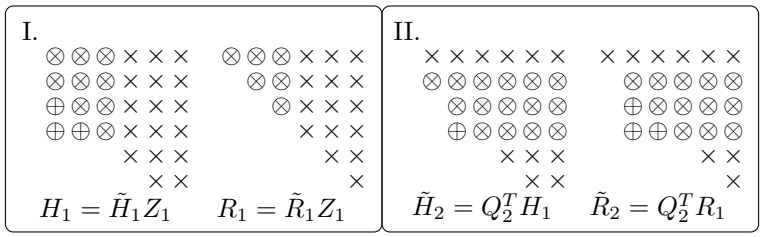


Figure 2.9: Chasing the bulge in the Hessenberg, triangular pencil.

structure in R_1. This can be achieved with two Householder reflectors or three core transformations. The update of H_1 creates additional nonzero entries in positions $(4,1)$ and $(4,2)$. Next, in pane II, a Householder reflector is computed which restores the upper Hessenberg form in the first column of $\tilde{H}_2$. Thereby creating new nonzero elements in $\tilde{R}_2$. Notice that the bulges in $(\tilde{H}_2, \tilde{R}_2)$ have shifted one row down and one column to the right compared to $(\tilde{H}_1, \tilde{R}_1)$.

This process of bulge chasing is repeated until the Hessenberg, triangular form is restored.

For our theoretical discussion of the implicit QZ method, we assume that the pencil (H, R) is made up of a proper Hessenberg matrix H and a nonsingular, upper triangular R. As we will show in the more general setting of Chapter 3, nonsingularity of R is not required but it simplifies things considerably at this stage. Under these assumptions, it is clear that an implicit QZ step, $(\hat{H}, \hat{R}) = Q^*(H, R)Z$, simultaneously performs two similarity transformations:

$$\hat{H}\hat{R}^{-1} = Q^* H R^{-1} Q, \quad \text{and,} \quad \hat{R}^{-1}\hat{H} = Z^* R^{-1} H Z, \tag{2.57}$$

on the proper Hessenberg matrices HR^{-1} and $R^{-1}H$. The implicit approach is motivated in the implicit Q theorem for Hessenberg, triangular pencils.

Theorem 2.3.7 (Implicit Q for Hessenberg, triangular pencils)**.** *Let (A, B) be an $n \times n$ regular matrix pencil and let $\hat{Q}, \check{Q}, \hat{Z}, \check{Z} \in \mathbb{F}^{n \times n}$ be unitary matrices satisfying $\hat{\boldsymbol{q}}_1 = \sigma \check{\boldsymbol{q}}_1$, $|\sigma| = 1$, such that,*

$$(\hat{H}, \hat{R}) = \hat{Q}^*(A, B)\hat{Z}, \quad \textit{and,} \quad (\check{H}, \check{R}) = \check{Q}^*(A, B)\check{Z},$$

are both Hessenberg, triangular pencils in proper Hessenberg, nonsingular triangular form. Then $\hat{Q} = \check{Q}D_1$, $\hat{Z} = \check{Z}D_2$, and $(\hat{H}, \hat{R}) = D_1^*(\check{H}, \check{R})D_2$ with D_1 and D_2 unitary diagonal matrices.

Proof. From (2.57) and Theorem 2.3.4 it directly follows that $\hat{Q} = \check{Q}D_1$. Furthermore, we have that $\hat{\boldsymbol{z}}_1 = \tilde{\sigma}\check{\boldsymbol{z}}_1$, $|\tilde{\sigma}| = 1$, since,

$$\hat{\boldsymbol{z}}_1 = B^{-1}\hat{Q}\hat{R}\boldsymbol{e}_1 = \hat{\gamma}B^{-1}\hat{\boldsymbol{q}}_1$$

$$\check{\boldsymbol{z}}_1 = B^{-1}\check{Q}\check{R}\boldsymbol{e}_1 = \check{\gamma}B^{-1}\check{\boldsymbol{q}}_1.$$

From (2.57) and Theorem 2.3.4 it directly follows that also $\hat{Z} = \check{Z}D_2$. □

So the outcome of an implicit QZ step is uniquely determined once $\boldsymbol{q}_1$ is fixed. The implicit QZ method implicitly performs a nested, polynomial accelerated subspace iteration just like the QR algorithm. Let us make this more precise in the following theorem.

Theorem 2.3.8 (Polynomial acceleration for QZ)**.** *Assume that (H, R) and $(\hat{H}, \hat{R})$ are both proper Hessenberg, nonsingular triangular pencils that are unitary equivalent, $(\hat{H}, \hat{R}) = Q^*(H, R)Z$, with the equivalence transformation obtained from a single implicit QZ sweep with $\boldsymbol{q}_1$ a scalar multiple of $p_m(HR^{-1})\boldsymbol{e}_1$. Then, for $k = 1, \ldots, n$,*

$$Q\mathcal{E}_k = p_m(HR^{-1})\mathcal{E}_k, \quad \textit{and,} \quad Z\mathcal{E}_k = p_m(R^{-1}H)\mathcal{E}_k.$$

Proof. The first part follows from directly from (2.57) and Theorem 2.3.6. For the second property, we have, just like in the proof of Theorem 2.3.7:

$$\boldsymbol{z}_1 = \gamma R^{-1}\boldsymbol{q}_1 = \gamma R^{-1}p_m(HR^{-1})\boldsymbol{e}_1 = \tilde{\gamma}p_m(R^{-1}H)\boldsymbol{e}_1.$$

Combining this with (2.57) and Theorem 2.3.6 concludes the proof. □

From this analysis and (2.57) we conclude that the implicit QZ method simultaneously performs two QR iterations on the Hessenberg matrices HR^{-1} and $R^{-1}H$. A good choice of shifts will lead to convergence of eigenvalues in the pencil driving it to (real) generalized Schur form.

2.3.4 BLAS and levels

Basic linear algebra subprograms or BLAS is a collection of matrix and vector operations that is available as part of LAPACK [2]. The collection consists of three *levels*, see also [46, Section 1.1.17].

Level-1 BLAS operations are vector-vector operations like a vector inner product or scaled vector addition. These operations involve an amount of data and require an amount of work on the data that both scale linear in the problem dimension. *Level-2* BLAS operations require a quadratic amount of data and perform a quadratic amount of work on the data. Examples are scaled matrix-vector multiplication and vector outer products. Finally, *level-3* BLAS operations require a quadratic amount of data but perform a cubic amount of work on the data An example is given by matrix-matrix multiplication, which is called `xGEMM` in BLAS [2].

As level-3 operations perform more work than level-2 and level-1 operations for the same amount of data, they can be much more efficient on current hardware where the cost of memory access often dominates over computations.

The frequent row and column updates in the basic single or double-shift QR and QZ algorithms that we discussed in this chapter make them rich in level-2 operations and thus ill-suited from a performance perspective. Blocking techniques for the QR method [14,15] and QZ method [58] have been successfully used to mitigate this issue and obtain algorithms that are rich in level-3 operations. We explore the use of blocking within our pole swapping framework in Chapter 4.

2.4 Conclusion

In this chapter we introduced the standard and generalized eigenvalue problems and the (generalized) Schur decomposition as an eigenvalue revealing decomposition. We studied the theoretical properties of polynomial Krylov subspace methods and discussed iterative Krylov methods such as Arnoldi's method. We showed how the polynomial QR and QZ methods proceed by first reducing the problem to Hessenberg or Hessenberg, triangular form, and afterwards iterating to (generalized) Schur form. We have seen that these methods implicitly perform nested subspace iteration with a change of basis accelerated by polynomials.

Chapter 3

A rational QZ method

This chapter is based on [19]:

Camps D., Meerbergen K., and Vandebril R. A rational QZ method. (2019) SIAM J. Matrix Anal. Appl. Vol. 40, No. 3, pp. 943–972.

The swapping algorithm outlined in Section 3.3.2 is based on [16]:

Camps D., Mach T., Vandebril R., and Watkins D. S. On pole-swapping algorithms for the eigenvalue problem. (2019) Submitted.

3.1 Introduction

Chapter 2 presented an overview of the classical QR/Z methods and discussed the connection with polynomial Krylov subspaces. In this chapter, we present a fully implicit method of QZ-type for the unsymmetric, generalized eigenvalue problem which is founded on *rational Krylov* theory. This *rational QZ* (RQZ) method is a *pole swapping* method which acts on pencils in Hessenberg, Hessenberg form. We will refer to this form as *Hessenberg pairs* or *Hessenberg pencils* for the sake of conciseness.

As we will demonstrate in detail, Hessenberg pairs and the associated rational Krylov subspaces are determined by *poles* that can be exploited to improve the convergence of the method. Both the original QZ algorithm [83] and the

extended QZ algorithm [130] turn out to be special instances of the RQZ method determined by a specific choice of poles.

Numerical experiments show that the RQZ method outperforms the classical QZ method by effectively reducing the number of iterations required to compute the generalized Schur form.

We will show that the RQZ method executes nested subspace iteration with *rational acceleration*. In our theoretical analysis, we rely directly on the pair (A, B) instead of rephrasing the relations in terms of a single matrix AB^{-1} or $B^{-1}A$ as is usually done. The proofs of *uniqueness* and convergence rely on rational Krylov theory, just like the theoretical results of the QR/Z methods in Chapter 2 relied on polynomial Krylov theory.

Related work on QR-type methods with rational acceleration can be found in [127, 128]. However, earlier work mainly centered around combining a shifted QR step with an RQ step with a single pole [127]. As we will see, our method sustains multiple poles in the Hessenberg pair and we rely on the simple and well-understood mechanism of pole swapping, while earlier work used more specialized *semiseparable plus diagonal* matrices [128].

The idea of pole swapping in Hessenberg pencils was first introduced by Berljafa and Güttel [11] in the context of rational Krylov methods. Their article studies uniqueness of rational Krylov decompositions and proposes an algorithm to move the poles in a rational Krylov decomposition. The authors also present how this method can be used to restart the rational Krylov method. We expand upon this idea in Chapter 7.

The work in this chapter is closely connected to the results of Berljafa and Güttel [11]. Our main contribution is the study of pole swapping methods for the direct solution of the eigenvalue problem supported by a detailed uniqueness and convergence analysis. Furthermore, we provide a novel backward stable swapping algorithm.

The material in this chapter is organized as follows. The notion of a Hessenberg pair is formally defined in Section 3.2 and its properties are studied subsequently. Two types of operations on Hessenberg pairs are discussed in Section 3.3: the introduction of a new pole and the *swapping* of poles. The algorithm we propose to compute the swapping transformations is shown to be backward stable in Appendix B. Section 3.4 proposes a method to reduce a dense matrix pair to a Hessenberg pencil with prescribed poles by means of unitary equivalence transformations. This is the RQZ analogue of the initial reduction phase in the QZ algorithm. We demonstrate that a good choice of poles can already lead to *premature deflations* during the reduction phase. The generalization of the iterative phase is presented in Section 3.5. It is illustrated how an RQZ

step with a single shift can be performed implicitly and numerical experiments illustrate the speed and accuracy. An implicit Q theorem for Hessenberg pairs is stated and used to prove that the RQZ iteration implicitly performs nested subspace iteration accelerated by a set of rational functions in Sections 3.6 and 3.7. Section 3.8 provides an *exactness* result which shows that the RQZ method deflates an eigenvalue in a single iteration provided a *perfect shift* is available. We conclude the chapter in Section 3.9.

3.2 Hessenberg pairs and their poles

In this section we repeat necessities from the literature and introduce some basic concepts linked to Hessenberg pairs. These pairs appear naturally in the context of the *rational Krylov* method introduced and studied by Ruhe [92–94, 96]. We elaborate further on this connection in Chapter 7.

3.2.1 Proper Hessenberg pairs

From Definition 2.2.4 we recall that a *proper* Hessenberg matrix has all its subdiagonal elements different from zero. Being proper ensures that there are no obvious *deflations* allowing us to split the Hessenberg matrix into block upper triangular form (2.52) with smaller subproblems. For a pair of Hessenberg matrices there is a subtlety, as there are two less obvious possibilities for deflation.

Definition 3.2.1 (Proper Hessenberg pair)**.** A pair of Hessenberg matrices $A, B \in \mathbb{F}^{n\times n}$ is said to be proper if the following two conditions are met:

I. There is no i in $1, \ldots, n-1$ so that $a_{i+1,i}$ and $b_{i+1,i}$ are simultaneously zero;

II. The first columns of A and B are linearly independent, as are the last rows of A and B.

For a proper Hessenberg pair we define its ordered pole tuple as:

$$\Xi = (\xi_1, \ldots, \xi_{n-1}),\ \xi_i \in \bar{\mathbb{C}},\ \text{where } \xi_i = a_{i+1,i}/b_{i+1,i},\ i = 1, \ldots, n-1.$$

The ratios of the subdiagonal elements of A over the subdiagonal elements of B are thus called the poles of the proper Hessenberg pair. Since we set division

by zero equal to ∞ in $\bar{\mathbb{C}}$, a pole is located at ∞ if the respective subdiagonal element of B is zero.

The first condition of being proper means that all poles are well-defined over $\bar{\mathbb{C}}$, so there is no $0/0$. Just like in the classical case $a_{i+1,i} = b_{i+1,i} = 0$ allows us to deflate the problem into two independent subproblems.

The second condition is less obvious, but it is simple to deflate an eigenvalue if it is not met. Construct a core transformation Q_1, acting on the first two rows such that Q_1^* maps the first column of A and B in the direction of $\boldsymbol{e}_1$, then the pair $Q_1^*(A, B)$ allows for a deflation. Similarly we can construct a rotation Z_{n-1} to transform $(A, B)Z_{n-1}$ to a deflatable format in case the last rows are linearly dependent. If condition II does not hold then the pair can be transformed into an equivalent pair for which condition I does not hold in the first or last subdiagonal position.

Condition II is equivalent with the condition that the pencils $(A - \lambda B)\boldsymbol{e}_1$ and $\boldsymbol{e}_n^T(A - \lambda B)$ do not have a zero according to Definition 2.1.5.

We remark that even if condition II of the definition of a proper Hessenberg pair were not met, we still define the first pole ξ_1 and last pole ξ_{n-1} as in Definition 3.2.1. Suppose, however, that there exists a scalar γ such that $\boldsymbol{a}_1 = \gamma \boldsymbol{b}_1$, with $\boldsymbol{a}_1$ and $\boldsymbol{b}_1$ the first columns of A and B respectively and that $a_{21} \neq 0$. This means that γ is both the first pole, $\xi_1 = a_{21}/b_{21} = \gamma$, and an eigenvalue, $A\boldsymbol{e}_1 = \gamma B\boldsymbol{e}_1$. Similarly, the last pole ξ_{n-1} is an eigenvalue if the last rows of A and B are linearly dependent.

Theorem 2.3.4 shows that properness of the Hessenberg matrix ensures *essential uniqueness* of the QR iterates, which is crucial in the design of an *implicit QR* algorithm [39, 40] for the standard eigenvalue problem. We prove an implicit Q theorem for Hessenberg pairs in Section 3.6 which shows that also proper Hessenberg pairs inherit a type of essential uniqueness allowing for the design of an implicit method.

The other pencils for which QZ algorithms were designed fit in Definition 3.2.1. Matrix pairs in Hessenberg, triangular form [83] are proper with poles $\Xi = (\infty, \infty, \ldots, \infty)$ and a matrix pair in *extended Hessenberg* form [130] is also a proper Hessenberg pair with poles being either 0 or ∞. Appendix A explains the extended Hessenberg structure in more detail.

The properties of proper Hessenberg pairs discussed in the next lemma are frequently used throughout this chapter.

Lemma 3.2.2. *Let $(A, B) \in \mathbb{C}^{n\times n}$ be a proper Hessenberg pair with poles $\Xi = (\xi_1, \ldots, \xi_{n-1})$. Then the following statements hold:*

I. For $\mu, \nu \in \mathbb{C}$, such that $\mu/\nu \notin \Xi$, we have that $(\nu A - \mu B)$ is a proper Hessenberg matrix.

II. For $\mu, \nu \in \mathbb{C}$, such that μ/ν is equal to a certain pole ξ_k, $1 \leq k \leq n-1$, we have that $N = (\nu A - \mu B)$ is block upper triangular,

$$N = \begin{bmatrix} N_{11} & N_{12} \\ & N_{22} \end{bmatrix},$$

where N_{11} and N_{22} are Hessenberg matrices respectively of sizes $k \times k$ and $(n-k) \times (n-k)$.

III. For $\mu, \nu, \alpha, \beta \in \mathbb{C}$, such that $\mu\beta \neq \alpha\nu$, we have that,

$$(M, N) = (\beta A - \alpha B, \nu A - \mu B),$$

is a proper Hessenberg pair with poles,

$$\hat{\xi}_k = \frac{\beta a_{k+1,k} - \alpha b_{k+1,k}}{\nu a_{k+1,k} - \mu b_{k+1,k}} \quad \text{for} \quad k = 1, \ldots, n-1.$$

IV. For $k = 1, \ldots, n-1$ we have that $\mathcal{R}(\boldsymbol{a}_1, \ldots \boldsymbol{a}_k) \neq \mathcal{R}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$.

Proof. Statements I and II are trivial. The pencil of statement III satisfies the definition of a proper Hessenberg pair: M and N are clearly upper Hessenberg matrices, their kth subdiagonal elements are,

$$\begin{bmatrix} m_{k+1,k} \\ n_{k+1,k} \end{bmatrix} = \begin{bmatrix} \beta & -\alpha \\ \nu & -\mu \end{bmatrix} \begin{bmatrix} a_{k+1,k} \\ b_{k+1,k} \end{bmatrix}.$$

The vector on the left is different from zero since the matrix is nonsingular and the vector on the right is nonzero. The first column of M is also linear independent from the first column of N because the same nonsingular matrix is used in the transformation. The same holds for the last row. The proof of statement IV is by induction and contradiction. The case $k = 1$ follows from the definition of a proper Hessenberg pair. Suppose the statement holds up to column k. We assume now, by contradiction, that it breaks down at column $k+1$, thus $\mathcal{R}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{k+1}) = \mathcal{R}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k+1})$. The equality implies the existence of a $(k+1) \times (k+1)$ matrix C such that,

$$\begin{bmatrix} \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k+1} \end{bmatrix} \begin{bmatrix} c_{11} & \cdots & c_{1,k+1} \\ \vdots & \ddots & \vdots \\ c_{k+1,1} & \cdots & c_{k+1,k+1} \end{bmatrix}. \tag{3.1}$$

It follows from the induction hypothesis that there is a j with $1 \leq j \leq k$ such that $\boldsymbol{a}_j \notin \mathcal{R}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$. Therefore $c_{k+1,j} \neq 0$. By the Hessenberg structure,

$$0 = a_{k+2,j} = \sum_{i=1}^{k+1} b_{k+2,i}\, c_{i,j} = b_{k+2,k+1} c_{k+1,j}.$$

This implies that $b_{k+2,k+1}$ must be zero and as a consequence (3.1) implies that also $a_{k+2,k+1} = 0$. These two values being simultaneously zero contradicts the properness. □

3.3 Manipulating the poles of Hessenberg pairs

In this section we study two operations for manipulating the poles of a Hessenberg pair, namely changing the first or the last pole, and swapping consecutive poles. These methods are also applied in [11] to change the poles of a rational Krylov recurrence.

3.3.1 Changing poles at the boundaries

Let $A, B \in \mathbb{C}^{n\times n}$ be a proper Hessenberg pair and assume the first pole ξ_1 different from the eigenvalues of (A, B). The pole ξ_1 can be changed to another pole $\hat{\xi}_1 \in \bar{\mathbb{C}}$ by multiplying (A, B) from the left with a unitary transformation Q_1^*, where $Q_1^*\boldsymbol{x} = \alpha\boldsymbol{e}_1$ and,

$$\boldsymbol{x} = \hat{\gamma}\,(\hat{\beta}_1 A - \hat{\alpha}_1 B)(\beta_1 A - \alpha_1 B)^{-1}\boldsymbol{e}_1 = \gamma\,(A - \hat{\xi}_1 B)(A - \xi_1 B)^{-1}\boldsymbol{e}_1, \tag{3.2}$$

with γ and $\hat{\gamma}$ convenient scaling factors; and $\hat{\alpha}_1, \hat{\beta}_1, \alpha_1, \beta_1 \in \mathbb{C}$ are chosen to satisfy the new pole $\hat{\xi}_1 = \hat{\alpha}_1/\hat{\beta}_1$ and the old pole $\xi_1 = \alpha_1/\beta_1$. The notation with α and β to denote $(\beta A - \alpha B)$ is factually the most correct one. For notational simplicity, however, we will often use the shorthand notation $(A - \xi B)$, where $\xi = \alpha/\beta$ instead. As $\hat{\xi}_1 \neq \xi_1$, otherwise nothing needs to be done, $\boldsymbol{x}$ must be a vector with only the two leading elements nonzero and thus Q_1 is always well defined and can, for example, be chosen as a rotation matrix.

If Q_1 is used to compute $(\hat{A}, \hat{B}) = Q_1^*(A, B)$ then $\hat{\xi}_1$ will become the first pole of $(\hat{A}, \hat{B})$ because the first subdiagonal element of $(\hat{A} - \hat{\xi}_1\hat{B})$ is zero:

$$\begin{aligned}(\hat{A} - \hat{\xi}_1\hat{B})\boldsymbol{e}_1 &= Q_1^*(A - \hat{\xi}_1 B)\boldsymbol{e}_1 \\ &= \tilde{\gamma}\, Q_1^*(A - \hat{\xi}_1 B)(A - \xi_1 B)^{-1}\boldsymbol{e}_1 = \tilde{\gamma}\gamma^{-1}\, Q_1^*\boldsymbol{x} = \alpha\,\tilde{\gamma}\gamma^{-1}\boldsymbol{e}_1.\end{aligned}$$

Theoretically, under the assumption that B is nonsingular, we could equally well define $\boldsymbol{x} = \gamma(AB^{-1} - \hat{\xi}_1 I)(AB^{-1} - \xi_1 I)^{-1}\boldsymbol{e}_1$. Practically, however, to avoid the nonsingularity assumption of B, and for reasons of numerical stability, we stick to (3.2).

Remark 3.3.1. As $(A - \xi_1 B)^{-1}\boldsymbol{e}_1$ is scalar multiple of $\boldsymbol{e}_1$ there is no need to compute this in practice. Moreover, even if ξ_1 is an eigenvalue, a scalar multiple of $\boldsymbol{e}_1$ is always a solution of $(A - \xi B)\boldsymbol{y} = \boldsymbol{e}_1$. The inverse factor is included to emphasize the rational function used to update the pole and moreover, we will see in Chapter 4 that it does play a role in the multishift setting. In practice we compute $\boldsymbol{x} = \gamma(A - \hat{\xi}_1 B)\boldsymbol{e}_1$ in $O(1)$ operations.

We can also compute an equivalence transformation to change the last pole by operating on the last two columns of the Hessenberg pair in a comparable way. Assume ξ_{n-1} different from the eigenvalues of (A, B). We can change the pole ξ_{n-1} to $\hat{\xi}_{n-1} \in \bar{\mathbb{C}}$ if we consider the row vector,

$$\boldsymbol{x}^T = \gamma \boldsymbol{e}_n^T (A - \xi_{n-1} B)^{-1}(A - \hat{\xi}_{n-1} B),$$

with γ a convenient scaling factor and a transformation Z_{n-1} that introduces a zero in the penultimate position of $\boldsymbol{x}^T$: $\boldsymbol{x}^T Z_{n-1} = \alpha \boldsymbol{e}_n^T$. If Z_{n-1} is computed in this way then the last pole in the Hessenberg pair $(\hat{A}, \hat{B}) = (A, B)Z_{n-1}$ is $\hat{\xi}_{n-1}$.

Again, the system $\boldsymbol{e}_n^T(A - \xi_{n-1}B)^{-1}$ is never solved in practice as the solution is a scalar multiple of $\boldsymbol{e}_n^T$, but is only included for theoretical purposes. In practice we compute $\boldsymbol{x}^T = \gamma \boldsymbol{e}_n^T(A - \hat{\xi}_{n-1}B)$.

3.3.2 Swapping poles

Any two consecutive poles ξ_i and ξ_{i+1} in a proper Hessenberg pair (A, B) can be *swapped* via a unitary equivalence on (A, B). We assume both poles to be different, otherwise nothing needs to be done. The procedure is summarized pictorially in Figure 3.1 where poles $\xi_3 = ③/ⓒ$ and $\xi_4 = ④/ⓓ$ are swapped. In this case the swapping operation is achieved by computing unitary matrices Q_4 and Z_3 that change the order of the eigenvalues in the 2×2 subpencil $A(4{:}5, 3{:}4) - \lambda B(4{:}5, 3{:}4)$. This subpencil is in generalized Schur form and has eigenvalues ξ_3 and ξ_4. The relevant part of the Hessenberg pair is indicated by the shaded region in Figure 3.1. The equivalence transformation changes all elements marked with $\otimes$ in pane II of Figure 3.1. Observe that the ratios ❹/ⓓ and ❸/ⓒ are preserved under swapping but the subdiagonal values themselves can change.

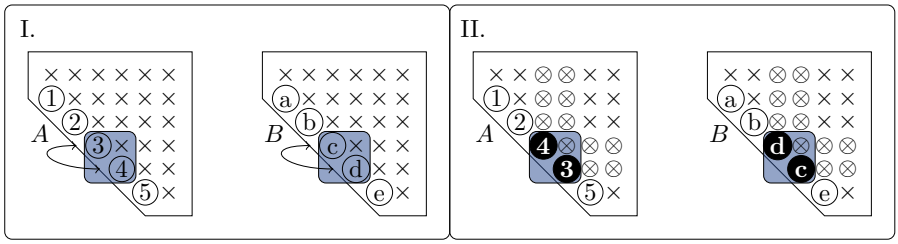


Figure 3.1: Swapping poles in a Hessenberg pair: (I) before swap, (II) after swap.

Swapping eigenvalues in an upper triangular pencil by means of a unitary equivalence transformation is a classical problem. It is typically used to reorder eigenvalues in the generalized Schur form for the sake of computing deflating subspaces. The problem has a unique solution which is determined by a coupled Sylvester equation. Algorithms based on solving the coupled Sylvester equation have been proposed by Kågström [57, 60]. Van Dooren [122] proposes a direct method which implicitly solves the Sylvester equation. The method of Van Dooren is used in the reordering routines `xTGEX2` of LAPACK [2].

In this section we introduce a swapping algorithm which is related to the method of Van Dooren [122], but has an improved backward error bound. The detailed error analysis is included in Appendix B. In the current section we describe the swapping algorithm and state the result of our error analysis in Lemma 3.3.2. This lemma shows that our swapping method is matrix-wise backward stable, while the error of earlier methods is bounded by $\epsilon_m \max(\|A\|_2, \|B\|_2)$. Here, ϵ_m is the machine precision and A and B refer to the small 2×2 upper triangular matrices taken out of the large Hessenberg pair. We will use this notation for the remainder of the current section. Numerical experiments are included in Appendix B which demonstrate the improved accuracy.

We are thus looking for an equivalence transformation $Q^*(A-\lambda B)Z = \hat{A}-\lambda\hat{B}$ of the following form:

$$Q^*\left(\begin{bmatrix}\alpha_1 & a\\ & \alpha_2\end{bmatrix} - \lambda\begin{bmatrix}\beta_1 & b\\ & \beta_2\end{bmatrix}\right) Z = \begin{bmatrix}\hat{\alpha}_1 & \hat{a}\\ & \hat{\alpha}_2\end{bmatrix} - \lambda\begin{bmatrix}\hat{\beta}_1 & \hat{b}\\ & \hat{\beta}_2\end{bmatrix}, \tag{3.3}$$

with $\alpha_1/\beta_1 = \hat{\alpha}_2/\hat{\beta}_2 = \xi_1$ and $\alpha_2/\beta_2 = \hat{\alpha}_1/\hat{\beta}_1 = \xi_2$.

Solution in exact arithmetic. To achieve the swapping of (3.3), we need to construct $Z = [\boldsymbol{z}_1\ \boldsymbol{z}_2]$, $Q = [\boldsymbol{q}_1\ \boldsymbol{q}_2]$ in such a way that:

- $\boldsymbol{q}_1$, $\boldsymbol{z}_1$ are a deflating pair (2.9) for $A-\lambda B$ corresponding to the eigenvalue ξ_2, i.e.
$$(A-\lambda B)\boldsymbol{z}_1 = \gamma_1 \boldsymbol{q}_1(\alpha_2 - \lambda\beta_2),$$
- similarly, $\boldsymbol{q}_2$, $\boldsymbol{z}_2$ are a deflating pair for ξ_1,
$$(A-\lambda B)\boldsymbol{z}_2 = \gamma_2 \boldsymbol{q}_2(\alpha_1 - \lambda\beta_1).$$

It then follows from the orthogonality of Q, Z that

$$\boldsymbol{q}_2^* A\boldsymbol{z}_1 = \boldsymbol{q}_2^* B\boldsymbol{z}_1 = 0,$$

and thus the swapping is achieved.

There are two options to compute this equivalence:

Method 1 On one hand, we can first construct a unitary matrix Z having its first column equal to the right eigenvector of $A-\lambda B$ related to ξ_2. To this end, the matrix $H_1 = \beta_2 A - \alpha_2 B$ is constructed. Observe that the second row of H_1 is zero by construction. Next, a rotation Z is computed such that HZ has a zero element in position $(1,1)$ such that $\boldsymbol{z}_1$ is indeed the right eigenvector of $A-\lambda B$ associated with $\xi_2 = \alpha_2/\beta_2$ as:

$$H_1 Z = (\beta_2 A - \alpha_2 B)Z = \begin{bmatrix} 0 & \times \\ 0 & 0 \end{bmatrix}. \tag{3.4}$$

This implies that $A\boldsymbol{z}_1$ and $B\boldsymbol{z}_1$ are parallel vectors and that a rotation Q can be computed to simultaneously introduce a zero in position $(2,1)$ of both AZ and BZ, thereby retrieving (3.3).

Method 2 On the other hand, we can first compute a unitary matrix Q having its second column equal to the Hermitian conjugate of the left eigenvector of $A-\lambda B$ related to ξ_1. Therefore we consider the matrix $H_2 = \beta_1 A - \alpha_1 B$ which has zeros in its first column by construction. Now, compute a rotation Q such that Q^*H has a zero in position $(2,2)$ such that $\boldsymbol{q}_2^*$ is indeed a left eigenvector of $A-\lambda B$ associated with ξ_1 as:

$$Q^* H_2 = Q^*(\beta_1 A - \alpha_1 B) = \begin{bmatrix} 0 & \times \\ 0 & 0 \end{bmatrix}. \tag{3.5}$$

This implies that $\boldsymbol{q}_2^* A$ and $\boldsymbol{q}_2^* B$ are parallel vectors and that a rotation Z can be computed to simultaneously introduce a zero in position $(2,1)$ of Q^*A and Q^*B, thereby retrieving (3.3). This second approach can be regarded as the dual of the first method.

Solution in floating point arithmetic. In Appendix B we prove the following error result for computing the swapping transformations in floating point arithmetic. We use a tilde to indicate computed quantities.

Lemma 3.3.2. *Let*

$$A - \lambda B = \begin{bmatrix} \alpha_1 & a \\ & \alpha_2 \end{bmatrix} - \lambda \begin{bmatrix} \beta_1 & b \\ & \beta_2 \end{bmatrix},$$

with $\alpha_1/\beta_1 = \xi_1$, *and* $\alpha_2/\beta_2 = \xi_2$. *Furthermore, assume* $|\xi_1| \geq |\xi_2|$. *If the swapping is computed by first deriving* $\tilde{Z}$, *as described in method 1 above, and afterwards computing* $\tilde{Q}$ *such that* $Q^*(BZ\boldsymbol{e}_1) = \gamma \boldsymbol{e}_1$, *then we have that the computed transformations satisfy:*

$$\tilde{Q}^*(A + E_A, B + E_B)\tilde{Z} = \left(\begin{bmatrix} \tilde{\alpha}_1 & \tilde{a} \\ & \tilde{\alpha}_2 \end{bmatrix}, \begin{bmatrix} \tilde{\beta}_1 & \tilde{b} \\ & \tilde{\beta}_2 \end{bmatrix} \right),$$

with $\|E_A\|_2 \leq c\epsilon_m \|A\|_2$, $\|E_B\|_2 \leq c\epsilon_m \|B\|_2$, c *a small constant.*

Lemma 3.3.2 guarantees that the method computes and exact swapping transformation of a nearby problem and that the off-diagonal elements in position $(2, 1)$ can be safely dismissed if $|\xi_1| \geq |\xi_2|$.

A corollary of Lemma 3.3.2 is that a backward stable swapping is achieved in case $|\xi_2| > |\xi_1|$ by first computing $\tilde{Z}$ via method 1 and afterwards obtaining $\tilde{Q}$ such that $Q^*(AZ\boldsymbol{e}_1) = \gamma \boldsymbol{e}_1$. This implicitly computes the transformations for $B - \lambda A$ whose eigenvalues are the inverse of $A - \lambda B$, such that Lemma 3.3.2 holds.

Another possibility is to use method 2. This leads to a backward stable swap when Z is computed from $(\boldsymbol{e}_2^* Q^* A)Z = \gamma \boldsymbol{e}_2^*$ if $|\xi_1| \geq |\xi_2|$ and from $(\boldsymbol{e}_2^* Q^* B)Z = \gamma \boldsymbol{e}_2^*$ otherwise. This can be verified by applying Lemma 3.3.2 to the *pertransposed* pencil. This is a transposition along the anti-diagonal.

All strategies that lead to a backward stable swap are summarized in Table 3.1. The first row of the table corresponds to method 1, the second with method 2. Option A is stable when $|\xi_1| \geq |\xi_2|$, while option B is stable when $|\xi_1| < |\xi_2|$.

3.4 Direct reduction to a proper Hessenberg pair

The rational QZ algorithm we propose in Section 3.5 operates on a proper Hessenberg pair. If we are given an arbitrary matrix pencil $A - \lambda B$ not yet in proper Hessenberg form, we first need to reduce it to this form. We use

Table 3.1: Numerical methods to compute a backward stable pole swap.

$\lvert\xi_1\rvert \geq \lvert\xi_2\rvert$	$\lvert\xi_1\rvert < \lvert\xi_2\rvert$
1.A) First Z, then Q from $Q^*(BZ\boldsymbol{e}_1) = \gamma\boldsymbol{e}_1$	1.B) First Z, then Q from $Q^*(AZ\boldsymbol{e}_1) = \gamma\boldsymbol{e}_1$
2.A) First Q, then Z from $(\boldsymbol{e}_2^*Q^*A)Z = \gamma\boldsymbol{e}_2^*$	2.B) First Q, then Z from $(\boldsymbol{e}_2^*Q^*B)Z = \gamma\boldsymbol{e}_2^*$

equivalences since we are interested in the eigenvalues and, for reasons of numerical stability, we will stick to unitary equivalences. At the end of the section we will illustrate with a numerical experiment that good pole selection can lead to deflations, already during the reduction process.

3.4.1 The reduction algorithm

The algorithm will transform an $n \times n$ matrix pair (A, B) to a unitary equivalent Hessenberg pair with a prescribed tuple of poles $\Xi = (\xi_1, \ldots, \xi_{n-1})$. The algorithm proceeds similarly to the direct reduction to Hessenberg, triangular form summarized in Figure 2.7, with the major difference that a pole is introduced at every step.

Just like in the classical reduction to Hessenberg, upper triangular form we commence with computing a QR factorization of $B = QR$ and updating the matrix pair to (Q^*A, Q^*B). The matrix Q^*B is now already in upper triangular form. This is shown in pane I of Figure 3.2 for our running example matrix pair of size 5×5. Moreover, we assume in the remainder of this section, that all zeros on the diagonal of B –infinite eigenvalues– are removed. An algorithm for this can be found in [138, Section 6.5].

We will now bring the first column of A to Hessenberg form. In pane II, a zero is introduced in position $(5, 1)$ of matrix A by operating on the last two rows. This destroys the upper triangular shape in the last two rows of B. The upper triangular shape can be restored by acting on columns 4 and 5 as shown in pane III without destroying the newly created zero in A.

The process of introducing zeros in the first column of A by acting on the rows and maintaining the upper triangular structure in B by acting on the columns can be repeated until the first column of A is brought to upper Hessenberg shape. This coincides with the classical reduction to a Hessenberg, triangular pair as discussed in Chapter 2.

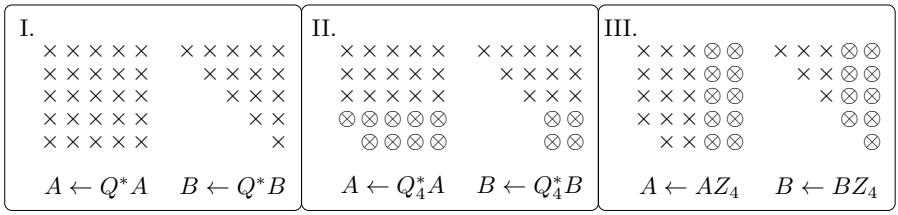


Figure 3.2: Reduction to a Hessenberg pencil. First part.

At this stage we have arrived at pane I of Figure 3.3. The first column of (A, B) is now already in the correct form but still has a pole at ∞. We can replace ∞ by another pole using the technique from Section 3.3 applied to the first column of (A, B) which is in Hessenberg form. This is always possible, except when there is an obvious deflation in the top left corner, meaning that the current pole is undefined as $0/0$. This does not pose any problems: deflate and continue. Under the assumption that there is no deflation, we start by first introducing the last pole $\xi_4 = \textcircled{4}/\textcircled{d}$ first. In the following steps of the reduction procedure this pole will move down to end up at the correct position at the bottom of the subdiagonal. The current state of the pair is visualized in pane II of Figure 3.3.

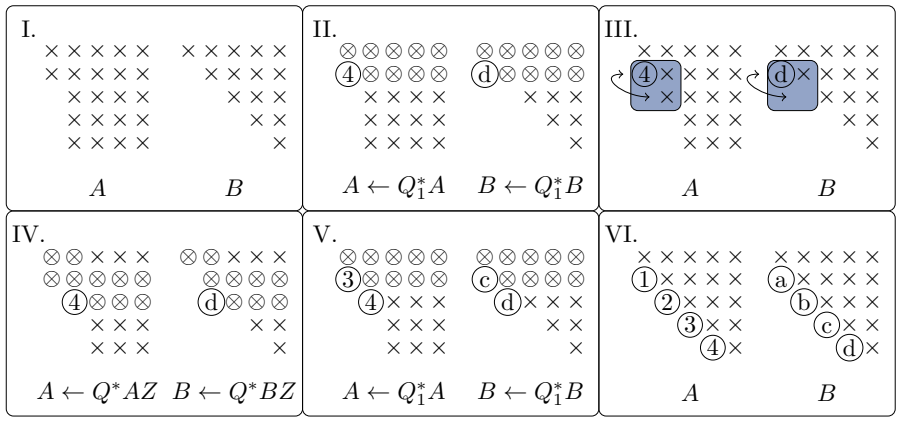


Figure 3.3: Reduction to a Hessenberg pencil. Second part.

The second column has been brought to Hessenberg, triangular form in pane III of Figure 3.3 via the classical procedure of introducing zeros in the second column of A and maintaining the upper triangular structure in B. Observe that this does not affect the existing pole ξ_4. At this moment, the first pole equals ξ_4, while the second pole is ∞. The poles in the shaded region of pane III are now swapped using the techniques from Section 3.3. This moves the pole at ∞

to the top of the matrix pair in pane IV. The swapping technique can be used, as the two leading columns of (A, B) are in Hessenberg form at this stage of the reduction algorithm. The swapping is always well defined, even if there is a succession of identical poles. The pole ξ_4 has moved one position down along the subdiagonal. The pair is now ready for the introduction of pole ξ_3 which is completed in pane V. This entire process of creating zeros, swapping poles, and introducing a new pole, can be repeated until the end result of pane VI is obtained, and the matrix is in the desired Hessenberg form.

After the reduction process, the matrix does not necessarily need to be in proper Hessenberg form. Possibly the pole ξ_{n-1} coincides with an eigenvalue, allowing for deflation in the lower right corner. In this case one deflates ξ_{n-1} and checks whether ξ_{n-2} leads to a deflation, and so forth, until the matrix has become proper. It can also happen that during the reduction any of the interior poles deflate. In this case the reduction can be continued on the separated parts of the pencil. This situation is studied in the numerical example of Section 3.4.2.

The introduction of the poles takes an additional $O(6n^3)$ flops on top of the $O(8n^3)$ operations required to reduce a pencil to Hessenberg, triangular form [46].

3.4.2 Numerical experiment

We study two matrix pairs from the magnetohydrodynamics (MHD) dataset available in the Matrix Market collection [13]. The matrices are of sizes 416 and 1280 respectively and known to be ill-conditioned. They originate from a Galerkin finite element discretization of the underlying MHD problem. Their spectrum consists of a tail along the negative real axis and a set of eigenvalues close to the imaginary axis. In this numerical experiment we determine deflating subspaces for the two regions of eigenvalues already during the reduction phase. The tests were run in Matlab R2017b.

Inspired by the link between contour integration methods [90, 107] and rational filtering techniques [116, 119], the idea is to introduce poles that make up a rational filter that slices the spectrum. To achieve this effect, the poles are chosen on a contour Γ in the complex plane that contains the eigenvalues along the negative real axis. In Section 3.7 we explain in full detail how introducing and swapping poles implicitly applies a rational filter to the pencil.

The poles are chosen on an elliptical contour $\Gamma = e(c, r_x, r_y)$, where c is the center of the ellipse, r_x is the radius in x-direction (along the real axis), r_y is the radius in the y-direction (along the imaginary axis). For the smaller problem, Γ is selected as $e(-1.3, 1.5, 3)$ and discretized in 120 nodes. For the

larger problem, $\Gamma = e(-25, 27, 6)$ and it is discretized in 400 nodes. These nodes are the poles introduced during the reduction to Hessenberg form. The aim is to get the pair improper, enforcing thereby a middle deflation separating the two regions. In case of a *middle* deflation we continue introducing poles on the separated parts.

The results are presented in Figures 3.4 to 3.6. Figure 3.4 shows an overview of the spectrum of both matrix pairs. The two regions of eigenvalues are indicated with different markers. The box in Figure 3.4 marks the area in which Figure 3.5 will zoom in; it shows where the regions meet in detail, together with the poles of the Hessenberg pair.

Figure 3.6 displays the magnitude of the subdiagonal elements $|a_{i+1,i}| + |b_{i+1,i}|$. All poles which are considered numerically zero and thus lead to a deflation are emphasized in a shaded rectangle. Typically some of the first and last poles are deflated, but more important is the presence of interior deflations. This happens at poles 103 to 106 after 160 poles have been introduced in the pair of size 416. For the larger pair, poles 317 to 321 are deflated after 621 poles have been introduced. The eigenvalues outside Γ are located in the top left part of the Hessenberg pair, those inside Γ appear after the interior deflation.

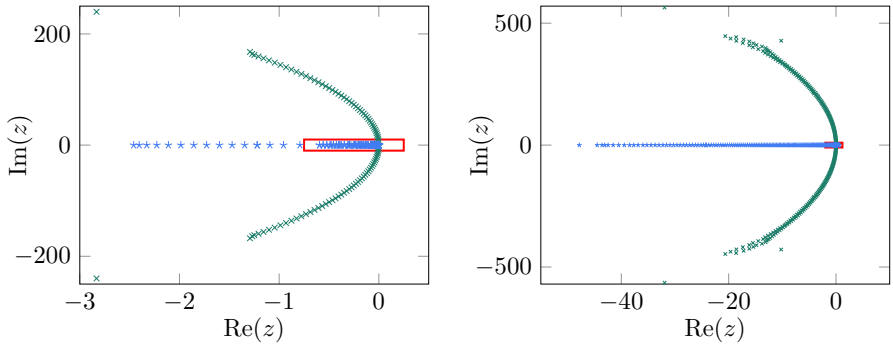


Figure 3.4: Eigenvalues in region 1 (×; bow-shape) and region 2 (*; close to the real axis). On the left we have the problem of size 416 and on the right 1280.

This numerical experiment shows that deflating subspaces containing regions of eigenvalues can be found already during the reduction to Hessenberg form. We like to stress that deflation is obtained without any of the poles converging towards an eigenvalue, but by choosing poles on a contour such that they construct an effective rational filter.

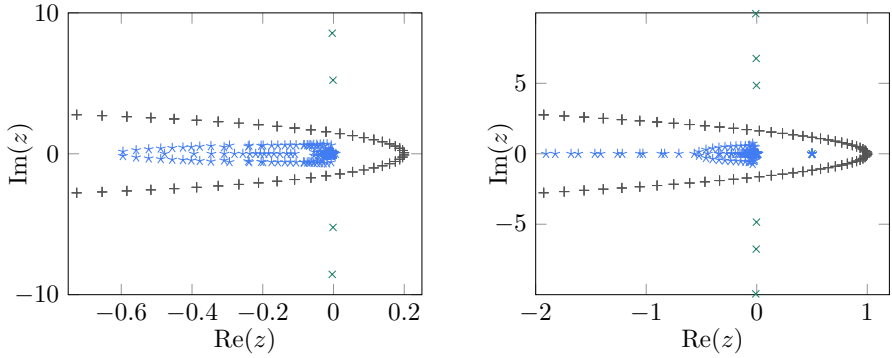


Figure 3.5: Close-up of the central part where the regions meet for the problem of size 416 and 1280. The legend is identical to the one of Figure 3.4 extended with the poles (+; on the ellipse around the real axis) .

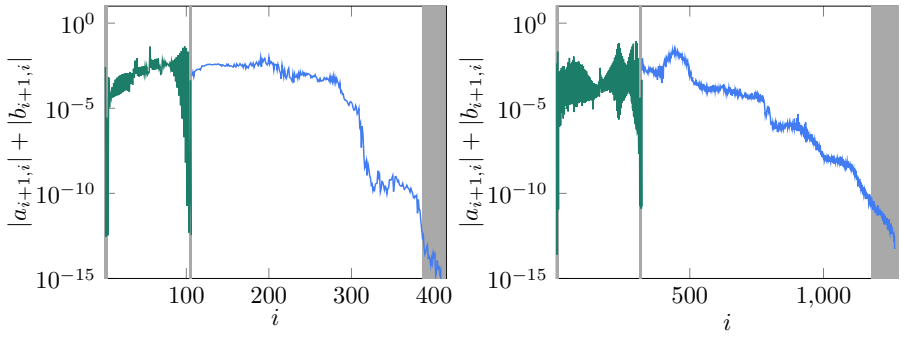


Figure 3.6: Magnitudes of the subdiagonal elements in the matrix pair after the Hessenberg reduction for the problem of size 416 (left) and 1280 (right).

3.5 Implicitly single shifted rational QZ step

In this section we present the implicit RQZ step for a Hessenberg pair. Numerical experiments are included at the end of this section and illustrate the performance and accuracy of the algorithm.

The algorithm operates on proper Hessenberg pairs. These pairs could be the result of the reduction procedure presented in Section 3.4 or they could be given directly, e.g., as coming from an iterative rational Krylov method, where one would like to compute the eigenvalues of the projected Hessenberg pair, see also Chapter 7.

3.5.1 The algorithm

Before describing the algorithm we like to comment on the nomenclature. We use both the terms *poles* ξ and *shifts* ϱ to refer to elements on the subdiagonal of a Hessenberg pair. In fact our shifts are poles as well, but we typically consider poles as subdiagonal elements that are sustained in the Hessenberg pair, while shifts are introduced and removed in a single implicit RQZ step. A shift is pushed in at the top, chased to the bottom, and removed at the end.

We introduce the RQZ procedure with an example. Given a 5×5 Hessenberg pair (A, B) with poles $\xi_1 = ①/ⓐ$, $\xi_2 = ②/ⓑ$, $\xi_3 = ③/ⓒ$, $\xi_4 = ④/ⓓ \in \bar{\mathbb{C}}$. The RQZ step consists of three stages, similar to all algorithms of implicit QR-type. These are an initialization, a chasing, and a finalization phase.

Initialization. Suppose we are given a shift $\varrho = \oplus/\oslash \in \bar{\mathbb{C}}$, for instance the Wilkinson shift [46]. Pane I in Figure 3.7 shows the Hessenberg pair in its initial state. The shift[1] is introduced in pane II by changing the first pole with a transformation Q_1 obtained by using the results from Section 3.3.

Chasing. Panes III-V show how the shift is relocated from the first position on the subdiagonal to position $n-1$ by repeatedly swapping it with the poles of the Hessenberg pair. The shift is chased to the bottom. The matrix elements that are changed in every step are marked with an $\otimes$.

During this procedure the shift will move from the top-left to the bottom-right and all poles will move up one position in the direction of the top-left corner. The assumption that the shift differs from the poles $\varrho \neq \xi_i$, for all i, ensures that none of the swapping operations equals an identity, otherwise the downward movement of the shift will undo the upward movement of the corresponding pole.

Finalization. Finally, in pane VI, one last operation can be performed where we have the possibility to remove the shift ϱ and introduce any new pole $\hat{\xi}_4 = ⑤/ⓔ \in \bar{\mathbb{C}}$, via the procedure described in Section 3.3.

It is clear that the algorithm described here generalizes the classical QZ algorithm [83] which we discussed in Chapter 2. In the QZ algorithm [83] one chases a bulge and in the final step the new pole was always put to ∞ thereby restoring

[1] A shift equal to a pole will not result in a breakdown, but leads to slow or no convergence at all (see Section 3.7). In practice shifts should be taken different from the poles.

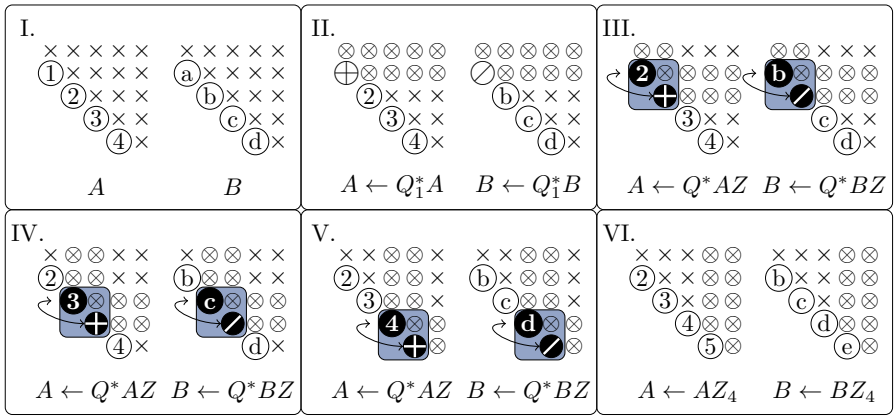


Figure 3.7: Single shifted implicit RQZ step on a 5×5 Hessenberg pair with shift ϱ.

the upper triangular form of B. The extended QZ algorithm [130] acts on extended Hessenberg pencils and allows for poles at 0 or ∞.

In the rational QZ algorithm we chase a shift instead of a bulge or a rotation. However, the shift is encoded in the bulge as well, as it is found as an eigenvalue of Watkins' bulge pencil [137, Section 5], [138]; the other eigenvalue in the bulge pencil is ∞. If we consider the same bulge pencil in the rational QZ case we see that the eigenvalue at ∞ is replaced by a pole of the pencil. Moreover, also the pole swapping technique is nothing else than the bulge exchange interpretation of Watkins [136].

3.5.2 Shifts, poles, and deflation

In order to implement the RQZ algorithm and in particular a single RQZ step, we need good strategies to select the shift, the new pole introduced at the very end, and a procedure to check if there are deflations.

For the shifts we typically take the Wilkinson shift [46, 141, 142]. This is the eigenvalue of the trailing 2×2 block that is closest to a_{nn}/b_{nn}. For the poles there are several options: one could as well consider a Wilkinson strategy determined by the 2×2 block in the upper-left corner or one could use other techniques such as poles on a contour to do filtering, see, e.g., Section 3.4.2. Optimal pole selection is a difficult and problem specific issue which is beyond the scope of this thesis. In our numerical experiments we test pole selection strategies based on localized eigenvalue estimates such as *Wilkinson poles*.

The deflation criterion for the interior poles $\xi_2, \ldots, \xi_{n-2}$ is obvious. If one of these is undefined in $\bar{\mathbb{C}}$, i.e. a 0/0, the problem can be split into smaller, independent problems as the pencil is block triangular form. This means in fact that for a certain i, two subdiagonal elements $a_{i+1,i}$ and $b_{i+1,i}$ are simultaneously zero. To numerically check this we use the classical relative criterion taking the sizes of the neighbouring elements into consideration [46],

$$|a_{i+1,i}| \leq c\epsilon_m(|a_{i,i}| + |a_{i+1,i+1}|) \quad \text{and} \quad |b_{i+1,i}| \leq c\epsilon_m(|b_{i,i}| + |b_{i+1,i+1}|),$$

with ϵ_m the machine precision and c a small constant.

The situation for the exterior poles ξ_1 and ξ_{n-1} is more peculiar. Whereas the interior poles are fixed, the exterior ones can be altered. Instead of changing ξ_1 or ξ_{n-1} to another pole, we would like to know whether it is possible to move them outside of $\bar{\mathbb{C}}$: we would like to deflate an eigenvalue. To this end we need to create simultaneous zeros on the subdiagonal of $A - \lambda B$ with a single operation such that the pair is no longer proper. We consider the situation at the bottom-right, the top-left corner proceeds similar. Suppose we would like to check if a deflation is possible for the last subdiagonal positions, which are $a_{n,n-1}$ and $b_{n,n-1}$.

This is only possible if the matrix $\left[\begin{smallmatrix} a_{n,n-1} & a_{n,n} \\ b_{n,n-1} & b_{n,n} \end{smallmatrix}\right]$ is of rank 1, or, equivalently, the pencil $\boldsymbol{e}_n^T(A - \lambda B)$ has a zero according to Definition 2.1.5. If this is the case, we can simultaneously annihilate the subdiagonal elements by creating a rotation Z_{n-1} which rotates the last rows of (A, B) in the direction of $\boldsymbol{e}_n^T$ an deflates the zero of $\boldsymbol{e}_n^T(A - \lambda B)$ as an eigenvalue of $A - \lambda B$. In our numerical experiments we assume the matrix to be of rank 1 if $\sigma_{\min}/\sigma_{\max} < \epsilon_m$.

3.5.3 Numerical experiment

We apply the RQZ method on two sets of problems: random matrix pairs and two problems from fluid dynamics. We are interested in the accuracy and speed.

Random matrix pairs. We test the single shift RQZ algorithm on 9 randomly generated, complex-valued matrix pairs with sizes ranging from 100 to 1000. The results are averaged over 10 runs. The pairs are first reduced to Hessenberg pairs with all poles at infinity, implying that no additional computational work has been done compared to the reduction phase of the QZ method. The shift is always taken as the Wilkinson shift. The poles are selected according to four different strategies: poles at infinity, poles at zero, random poles, and poles chosen as the Wilkinson shift from the upper-left 2×2 block. The last choice is called the Wilkinson pole.

The results are summarized in Figures 3.8 and 3.9. The left pane of Figure 3.8 shows the relative backward errors $\|\hat{A}-Q^*AZ\|_2/\|A\|_2$ and $\|\hat{B}-Q^*BZ\|_2/\|B\|_2$ for the reduction to a Hessenberg pair (lines without markers) and the backward error on the Schur form for the four different pole strategies. The backward error is small in all cases. The right pane shows the average number of iterations per eigenvalue. Clearly, the Wilkinson pole requires the least number of iterations per eigenvalue. It requires on average 1.5% less iterations than the classic choice of poles at infinity. Random poles and poles at zero perform the worst.

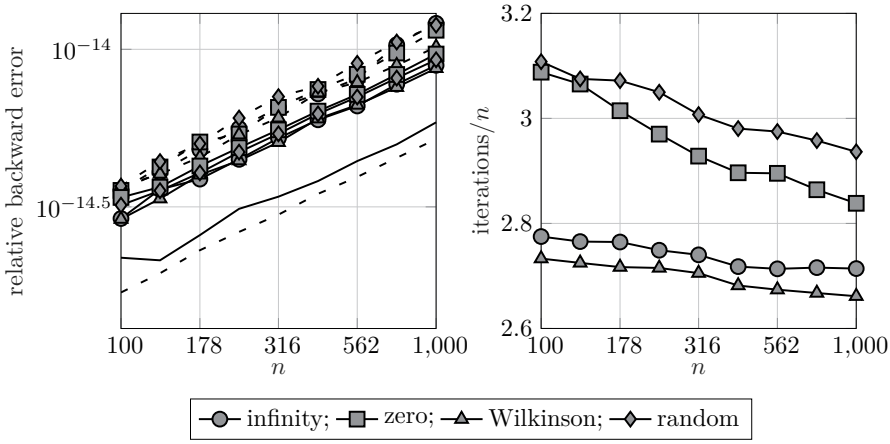


Figure 3.8: On the left the relative backward errors related to the reduction to a Hessenberg pair (no markers) and to the Schur form (with markers) are demonstrated. The error on A is represented with a dashed line and the error on B with a full line. On the right we see the average number of iterations per eigenvalue for the four different pole strategies.

Figure 3.9 shows the total number of pole swaps scaled with n^2. The scaling factor is used since the number of pole swaps per iteration is $O(n)$ and the expected number of iterations is also $O(n)$ resulting in a total of $O(n^2)$ swaps. This measure of performance depends heavily on the positions where deflations occur and as such gives a much better view on the algorithmic behavior. The order of the four strategies remains the same, but the gains with Wilkinson poles increase up to 4%. This signals the occurrence of deflations at other spots than only in the lower-right corner as is typically the case in the classical setting.

IFISS problems. In this experiment we apply the RQZ method on two problems from fluid dynamics generated with IFISS [34,35]. The first problem

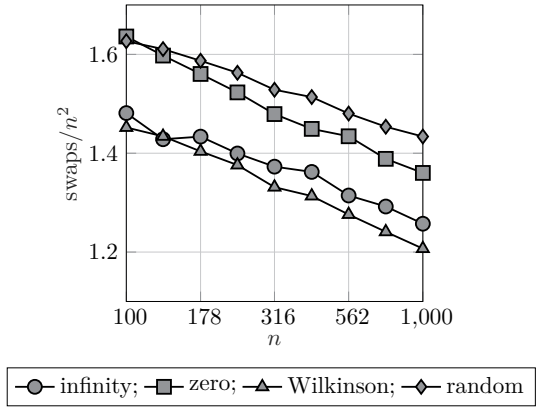


Figure 3.9: The total number of swaps scaled by n^2 for the computation of the Schur form for the four different pole strategies.

originates from a model for the flow in a unit-square cavity, the second problem comes from a model for the flow around an obstacle. Both models are discretized, resulting in two real, generalized eigenvalue problems. The *cavity flow* problem is of size 2467, the *obstacle flow* problem of size 2488. We applied the single shift RQZ method after initial reduction to Hessenberg form with poles at infinity. Wilkinson shifts are employed in all cases. We used poles at infinity and Wilkinson poles. The spectra of the matrices are shown in Figure 3.10.

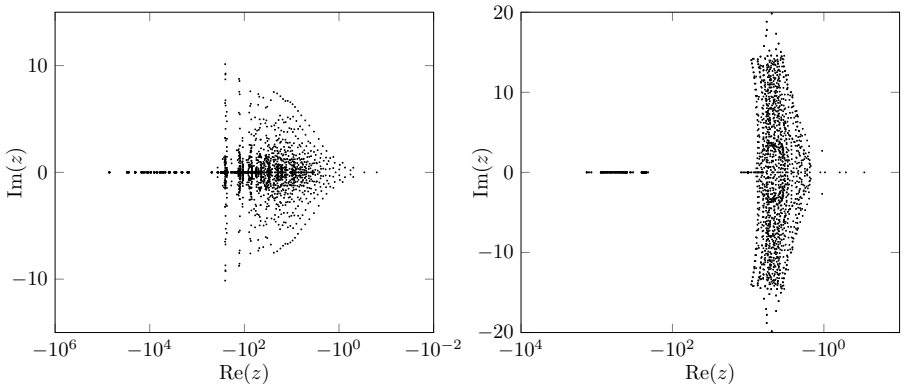


Figure 3.10: On the left the spectrum of the cavity flow problem and on the right the spectrum of the obstacle flow problem are shown.

The results of the experiment are summarized in Table 3.2. It lists the relative

backward error on the Schur form for both A and B for both problems and the two pole strategies. The backward error is very good in all cases. The table also lists the average iterations per eigenvalue and how this compares relatively to the result of poles at infinity. We observe that the average number of swaps and iterations when employing Wilkinson poles is always below the numbers generated by the classical approach.

Table 3.2: Results of the RQZ method on the IFISS problems. The first column lists the problem, the second column the pole strategy. Columns 3 and 4 present the backward error on A and B, columns 5 and 6 the average number of iterations and performance compared to QZ, and columns 7 and 8 the total number of swaps and the performance compared to QZ.

Problem	pole	error A	error B	it/n	%	swaps/n^2	%
Cavity flow	∞	$7.5 \cdot 10^{-15}$	$4.4 \cdot 10^{-15}$	2.49	100	0.446	100
	Wilk.	$7.8 \cdot 10^{-15}$	$4.1 \cdot 10^{-15}$	2.34	94.2	0.443	99.3
Obstacle flow	∞	$9.2 \cdot 10^{-15}$	$7.8 \cdot 10^{-15}$	2.54	100	0.617	100
	Wilk.	$8.8 \cdot 10^{-15}$	$7.8 \cdot 10^{-15}$	2.36	93.0	0.595	96.3

3.5.4 Tightly-packed shifts

The single shifted RQZ method is, just like the classical QZ method, sequential in nature and not very cache efficient. To enhance cache performance one can go for a *multishift* approach and chase m shifts simultaneously or one can chase m single shifts as close as possible after each other. In the next chapter, we will study multishift, multipole RQZ steps. The theory in this chapter is not suited for a multishift setting and we will confine ourselves for now to a description and a numerical experiment using *tightly-packed shifts*.

Assume we would like to chase m tightly-packed shifts, which are typically the eigenvalues of the bottom-right $m \times m$ block of (A, B). These shifts are introduced one after another in the Hessenberg pair. The first shift is introduced and swapped down one row. Next the second shift is introduced and both shifts need to be swapped down a single row, starting with the lower-right one first. As a result there is space to introduce the third shift, and the procedure continues. At this stage, the first m subdiagonal elements of the pair (A, B) encode the shifts.

In order to chase the block of m shifts, one needs to swap all shifts down one row, starting again with the one in lower-right corner first. In total this requires m equivalence transformations (3.3) which are accumulated to update the necessary parts of the matrices in a cache efficient manner. The first batch

of swaps in illustrated in Figure 3.11. It is also possible to accumulate all transformations that swap the m shifts with the subsequent k poles, this is tested in Chapter 4.

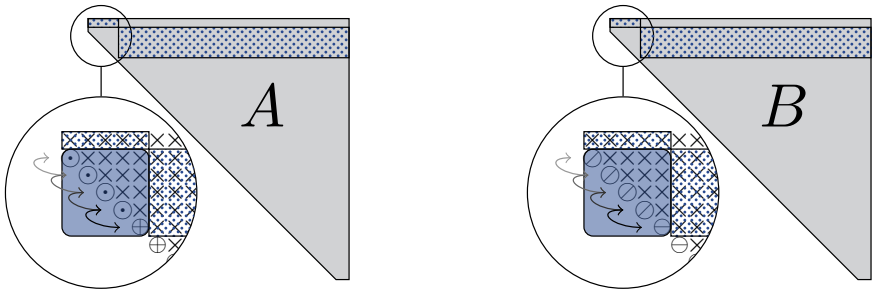


Figure 3.11: First swap in the RQZ method with $m = 4$ tightly-packed shifts. The magnified parts show the 4 shifts $\odot/\oslash$ right after the initialization phase and indicates how they can all be swapped with the next pole $\oplus/\ominus$ by four swaps. These transformation are computed based on the shaded regions in the magnified area. The dotted sections in the matrix pair are the rows and columns that can be updated efficiently after the swap.

The finalization phase commences when the shifts occupy the last subdiagonal positions in the Hessenberg pair. We can now introduce m new poles. The first new pole is introduced in the final subdiagonal element and swapped up m positions thereby swapping all remaining shifts down. The second new pole is now introduced and this course of action continues until the new poles occupy the last m subdiagonal elements.

We test the tightly-packed RQZ method on randomly generated matrix pairs of size 600 that are first reduced to Hessenberg pairs with poles at infinity. We run the RQZ method for shift batches of sizes $m = 2, 4, 8, 16, 32$. The results are averaged over 10 runs. The poles are selected following three criteria: always at infinity (classical QZ), m times the Wilkinson pole of the leading 2×2 block, or as the eigenvalues of the leading $m\times m$ block, the *Rayleigh* poles.

Figure 3.12 displays the performance in terms of the average number of iterations per eigenvalue (left) and total number of swaps scaled with n^2 (right) in function of the batch size m for the three types of poles. We observe that the number of iterations increases significantly for larger m. This effect is most pronounced with the Wilkinson and Rayleigh poles. Also in terms of the number of swaps the poles at infinity are the most efficient choice. We attribute this effect to the spectrum of the randomly generated problems. All, except typically one, of the eigenvalues are located in one cluster around zero. Likely, due to the increased

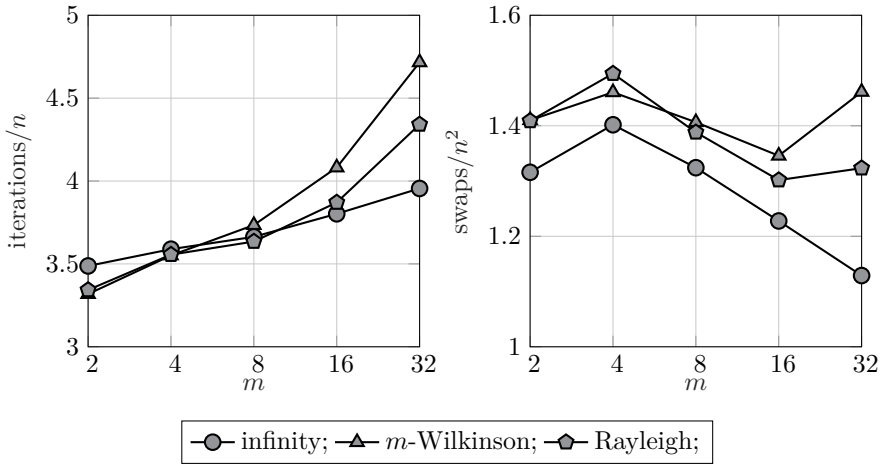


Figure 3.12: On the left the average number of iterations per eigenvalue is depicted in function of batch size m for three different pole strategies. On the right the average number of swaps scaled with n^2 in function of the batch size m. These results are for the random problem.

batch size, some of the Wilkinson and Rayleigh poles will somehow be too close to each other, thereby deteriorating the convergence.

Therefore, we have repeated this experiment with 10 randomly generated matrix pairs of size 600 having two equally sized clusters of eigenvalues centered around 0 and 10. The results are shown in Figure 3.13. Now the Wilkinson and Rayleigh poles outperform the poles at infinity in terms of total number of swaps for all batch sizes indicating that the poles do improve the convergence rate of the method.

We conclude that we can pack the shifts tightly without a significant degradation in convergence behavior. The advantages of allowing pole selection remain but become more problem specific. The packing of the shifts along the subdiagonal of the Hessenberg pair is optimal. This is trivial in our pole swapping context, but not in a bulge chasing algorithm [66].

The numerical results shown here are obtained with a Matlab implementation. A cache efficient implementation of tightly-packed pole swapping algorithms is studied in Chapter 4, where we show that level-3 BLAS performance (Section 2.3.4) can reduce CPU times by nearly an order of magnitude.

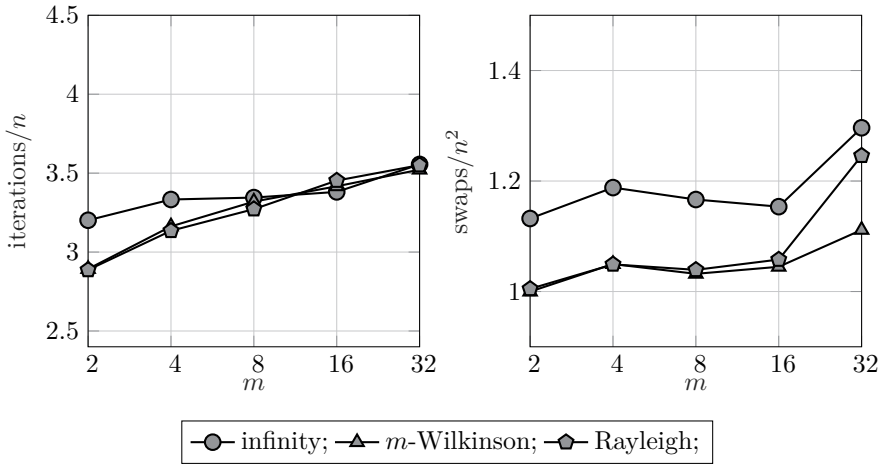


Figure 3.13: On the left the average number of iterations per eigenvalue is depicted in function of batch size m for three different pole strategies. On the right the average number of swaps scaled with n^2 in function of the batch size m. These are the results for the random problems with two clusters.

3.6 Implicit Q theorem

In this section we prove the following implicit Q theorem for proper Hessenberg pairs justifying the implicit approach since the result of a rational QZ step is uniquely determined.

Theorem 3.6.1 (Implicit Q theorem for proper Hessenberg pairs)**.** *Let (A, B) be a regular matrix pair and let $\hat{Q}$, $\check{Q}$, $\hat{Z}$, $\check{Z}$ be unitary matrices with $\hat{Q}\boldsymbol{e}_1 = \sigma\check{Q}\boldsymbol{e}_1$, $|\sigma| = 1$, such that,*

$$(\hat{A}, \hat{B}) = \hat{Q}^*(A, B)\hat{Z} \quad \textit{and} \quad (\check{A}, \check{B}) = \check{Q}^*(A, B)\check{Z},$$

are both proper Hessenberg pairs having both the same pole tuple $\Xi = (\xi_1, \ldots, \xi_{n-1})$, $\xi_i \in \bar{\mathbb{C}}$, with the poles different from the spectrum of the pair.

Then the pairs $(\hat{A}, \hat{B})$ and $(\check{A}, \check{B})$ are essentially identical*, meaning that,*

$$\hat{A} = D_1^*\check{A}D_2 \quad \textit{and} \quad \hat{B} = D_1^*\check{B}D_2, \tag{3.6}$$

with D_1 and D_2 unitary diagonal matrices.

The implicit Q theorem guarantees that the unitary equivalence transformations, which are applied in the direct reduction to a Hessenberg pair and in a rational

QZ step are essentially unique. Once the reduction or the rational QZ step is initiated, the outcome is determined.

The remainder of this section contains all ingredients to prove this theorem. Various related implicit Q theorems already exist. Mastronardi, Vandebril, and Van Barel [126] provide one for semiseparable plus diagonal matrices linked to rational Krylov spaces. Pranic, Mach, and Vandebril [80] formulate a variant for extended Hessenberg plus diagonal matrices linked to general rational Krylov subspaces as did Berljafa and Güttel for (rectangular) Hessenberg pairs connected with rational Krylov recurrences [11], see Chapter 7.

The proof we provide here significantly differs from the proof in [11], which relies on direct computations and utilize the invertibility of B to formulate the theory for the single matrix setting. We make use of the properties of the associated Krylov matrices, as done by Watkins for the classical case [138] and which we summarized in Section 2.3. This rational Krylov connection allows us to easily prove that the rational QZ algorithm performs nested subspace iteration accelerated by rational functions.

3.6.1 Rational Krylov matrices and subspaces

We define *rational Krylov matrices* generated by a matrix pair (A, B), a vector $\boldsymbol{v}$, and a driving rational function determined by shifts P and poles Ξ in this section. These rational Krylov matrices span Krylov subspaces, which, for consistency, we will name *rational Krylov subspaces*. The description holds for regular matrix pairs, so the matrices do not need to be of Hessenberg form.

For the aim of a concise notation we introduce two *elementary rational matrices* generated from the pair (A, B) with shift $\varrho = \mu/\nu \in \bar{\mathbb{C}}$ and pole $\xi = \alpha/\beta \in \bar{\mathbb{C}}$:

$$\begin{aligned} M(\varrho, \xi) &= (\nu A - \mu B)(\beta A - \alpha B)^{-1}, \\ N(\varrho, \xi) &= (\beta A - \alpha B)^{-1}(\nu A - \mu B). \end{aligned} \tag{3.7}$$

We assume, throughout the remainder of the text, the shift different from the pole $\varrho \neq \xi$ and since we take inverses, the pole may not be an eigenvalue $\xi \notin \Lambda$. Notice that $M(\varrho, \xi)$ and $N(\varrho, \xi)$ represent an entire class of matrices generated by parameters that result in the correct shift and pole. These are all scalar multiples of one another and as the theory remains scale invariant, every nontrivial representative is fine.

We remark that if the matrices A and B commute, $AB = BA$, then also

$$(\beta A - \alpha B)(\nu A - \mu B) = (\nu A - \mu B)(\beta A - \alpha B),$$

and consequently $M(\varrho,\xi) = N(\varrho,\xi)$. In case B is invertible the following relations hold,

$$M(\varrho,\xi) = (\nu AB^{-1} - \mu I)(\beta AB^{-1} - \alpha I)^{-1} = (\beta AB^{-1} - \alpha I)^{-1}(\nu AB^{-1} - \mu I),$$
$$N(\varrho,\xi) = (\beta B^{-1}A - \alpha I)^{-1}(\nu B^{-1}A - \mu I) = (\nu B^{-1}A - \mu I)(\beta B^{-1}A - \alpha I)^{-1}. \tag{3.8}$$

This could be helpful to link this analysis to existing theorems of Berljafa & Güttel [11], and Watkins [138].

The elementary rational matrices are used to define rational Krylov matrices.

Definition 3.6.2 (rational Krylov matrices)**.** Let $(A,B) \in \mathbb{C}^{n\times n}$ be a regular matrix pair, $\boldsymbol{v} \in \mathbb{C}^n$ a nonzero vector, $\Xi = (\xi_1,\ldots,\xi_{k-1})$, $\xi_i \in \bar{\mathbb{C}}$, the pole tuple with the poles different from the spectrum, and $P = (\varrho_1,\ldots,\varrho_{k-1})$, $\varrho_i \in \bar{\mathbb{C}}$, the tuple of shifts distinct from the poles, with $k \le n$. The corresponding rational Krylov matrices are defined as:

$$\begin{aligned} K_k^{\mathrm{rat}}(A,B,\boldsymbol{v},\Xi,P) &= \left[\boldsymbol{v}, M(\varrho_1,\xi_1)\boldsymbol{v}, M(\varrho_2,\xi_2)M(\varrho_1,\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} M(\varrho_i,\xi_i)\boldsymbol{v}\right], \\ L_k^{\mathrm{rat}}(A,B,\boldsymbol{v},\Xi,P) &= \left[\boldsymbol{v}, N(\varrho_1,\xi_1)\boldsymbol{v}, N(\varrho_2,\xi_2)N(\varrho_1,\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} N(\varrho_i,\xi_i)\boldsymbol{v}\right]. \end{aligned} \tag{3.9}$$

The following properties of the elementary rational matrices are frequently used in the remainder of the text.

Lemma 3.6.3. *The elementary rational matrices* (3.7) *satisfy:*

I. Commutativity: For shifts $\varrho,\hat{\varrho}$ different from the poles $\xi,\hat{\xi}$,

$$\begin{aligned} M(\varrho,\xi)\, M(\hat{\varrho},\hat{\xi}) &= M(\hat{\varrho},\hat{\xi})\, M(\varrho,\xi), \\ N(\varrho,\xi)\, N(\hat{\varrho},\hat{\xi}) &= N(\hat{\varrho},\hat{\xi})\, N(\varrho,\xi). \end{aligned} \tag{3.10}$$

II. Repositioning shifts: For shifts $\varrho,\hat{\varrho}$ different from the poles $\xi,\hat{\xi}$,

$$\begin{aligned} M(\varrho,\xi)\, M(\hat{\varrho},\hat{\xi}) &= M(\hat{\varrho},\xi)\, M(\varrho,\hat{\xi}), \\ N(\varrho,\xi)\, N(\hat{\varrho},\hat{\xi}) &= N(\hat{\varrho},\xi)\, N(\varrho,\hat{\xi}). \end{aligned} \tag{3.11}$$

III. Inverse: If the shift is not an eigenvalue, $\varrho \notin \Lambda$, and is different from the pole, $\varrho \neq \xi$, then,

$$\begin{aligned} M(\varrho,\xi)^{-1} &= M(\xi,\varrho), \\ N(\varrho,\xi)^{-1} &= N(\xi,\varrho). \end{aligned} \tag{3.12}$$

IV. Shift invariance: For any nonzero vector $\boldsymbol{v} \in \mathbb{C}^n$, and parameters $\varrho, \hat{\varrho} \neq \xi$,

$$\begin{aligned} \mathcal{R}(\boldsymbol{v},\ M(\varrho,\xi)\boldsymbol{v}) &= \mathcal{R}(\boldsymbol{v},\ M(\hat{\varrho},\xi)\boldsymbol{v}), \\ \mathcal{R}(\boldsymbol{v},\ N(\varrho,\xi)\boldsymbol{v}) &= \mathcal{R}(\boldsymbol{v},\ N(\hat{\varrho},\xi)\boldsymbol{v}). \end{aligned} \tag{3.13}$$

Proof. If B is invertible, properties I and II of the Lemma follow from (3.8) and the property that any matrix commutes with its shifted inverse. For singular B the same result follows from an elementary continuity argument. Property III is trivial. For property IV, we consider first the case that $\xi \neq \infty$. Assuming $\varrho \neq \infty$, it follows from (3.7) that,

$$\begin{aligned} M(\varrho,\xi) &= \frac{\nu}{\beta}(I + (\xi-\varrho)B(A-\xi B)^{-1}) \equiv I + (\xi-\varrho)\, M(\infty,\xi), \\ N(\varrho,\xi) &= \frac{\nu}{\beta}(I + (\xi-\varrho)(A-\xi B)^{-1}B) \equiv I + (\xi-\varrho)\, N(\infty,\xi). \end{aligned} \tag{3.14}$$

The second part of the equation is considered as an equivalence: both sides belong to the same class of rational matrices but differ by a finite, nonzero scalar factor. It is clear that both $\mathcal{R}(\boldsymbol{v}, M(\varrho,\xi)\boldsymbol{v}) = \mathcal{R}(\boldsymbol{v}, M(\infty,\xi)\boldsymbol{v})$ and $\mathcal{R}(\boldsymbol{v}, N(\varrho,\xi)\boldsymbol{v}) = \mathcal{R}(\boldsymbol{v}, N(\infty,\xi)\boldsymbol{v})$. This equality holds trivially in case $\varrho = \infty$. Consequently, the shift invariance property is satisfied for $\xi \neq \infty$. In case $\xi = \infty$, assuming $\varrho \neq 0$, (3.7) reads,

$$M(\varrho,\infty) \equiv \varrho I - M(0,\infty),\ N(\varrho,\infty) \equiv \varrho I - N(0,\infty), \tag{3.15}$$

and the shift invariance also follows for a pole at ∞ as the shift can always be moved to zero. This is trivial in case $\varrho = 0$. □

Theorem 3.6.4 (Nested shift invariance)**.** *For any nonzero vector $\boldsymbol{v} \in \mathbb{C}^n$, all shifts ϱ_i different from all poles ξ_j for i, j from 1 to $k-1$, and an alternative shift $\hat{\varrho}$ different from all poles, $k \leq n$,*

$$\begin{aligned} \mathcal{R}\left(\boldsymbol{v}, M(\varrho_1,\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} M(\varrho_i,\xi_i)\boldsymbol{v}\right) &= \mathcal{R}\left(\boldsymbol{v}, M(\hat{\varrho},\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} M(\hat{\varrho},\xi_i)\boldsymbol{v}\right), \\ \mathcal{R}\left(\boldsymbol{v}, N(\varrho_1,\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} N(\varrho_i,\xi_i)\boldsymbol{v}\right) &= \mathcal{R}\left(\boldsymbol{v}, N(\hat{\varrho},\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} N(\hat{\varrho},\xi_i)\boldsymbol{v}\right). \end{aligned} \tag{3.16}$$

Proof. We prove the first relation of (3.16) by induction, the proof for the second relation proceeds similarly. The case k=1 is trivial, the case k=2 is equal to the shift invariance property IV of Lemma 3.6.3. Assume now Theorem 3.6.4 holds up to index k and denote this subspace as $\mathcal{U}_k$. We remark that (3.16) also implies that,

$$\mathcal{R}\left(\boldsymbol{v}, M(\varrho_1,\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} M(\varrho_i,\xi_i)\boldsymbol{v}\right) = \mathcal{R}\left(\boldsymbol{v}, M(\hat{\varrho}_1,\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} M(\hat{\varrho}_i,\xi_i)\boldsymbol{v}\right), \tag{3.17}$$

for arbitrary shifts $\hat{\varrho}_i$ different from all poles. The subspace $\mathcal{U}_{k+1}$ is equal to:

$$\mathcal{U}_{k+1} = \mathcal{R}\left(\boldsymbol{v},\, M(\varrho_1,\xi_1)\boldsymbol{v}, \ldots, \prod_{i=1}^{k} M(\varrho_i,\xi_i)\boldsymbol{v}\right) = \mathcal{U}_k + \mathcal{R}\left(\prod_{i=1}^{k} M(\varrho_i,\xi_i)\boldsymbol{v}\right).$$

By the induction hypothesis, the result holds for $\mathcal{U}_k$. We now modify the additional term in the subspace $\mathcal{U}_{k+1}$ to prove the result:

$$\begin{aligned}
\mathcal{U}_{k+1} &= \mathcal{U}_k + M(\varrho_k,\xi_k)\,\mathcal{R}\left(\prod_{i=1}^{k-1} M(\varrho_i,\xi_i)\boldsymbol{v}\right) \\
&= \mathcal{U}_k + M(\hat{\varrho},\xi_k)\,\mathcal{R}\left(\prod_{i=1}^{k-1} M(\varrho_i,\xi_i)\boldsymbol{v}\right) \\
&= \mathcal{U}_k + M(\varrho_{k-1},\xi_k)\,\mathcal{R}\left(M(\hat{\varrho},\xi_{k-1})\prod_{i=1}^{k-2} M(\varrho_i,\xi_i)\boldsymbol{v}\right) \\
&= \mathcal{U}_k + M(\hat{\varrho},\xi_k)\,\mathcal{R}\left(M(\hat{\varrho},\xi_{k-1})\prod_{i=1}^{k-2} M(\varrho_i,\xi_i)\boldsymbol{v}\right) \\
&= \mathcal{U}_k + M(\varrho_{k-2},\xi_k)\,\mathcal{R}\left(M(\hat{\varrho},\xi_{k-2})M(\hat{\varrho},\xi_{k-1})\prod_{i=1}^{k-3} M(\varrho_i,\xi_i)\boldsymbol{v}\right) \\
&= \ldots \\
&= \mathcal{U}_k + \mathcal{R}\left(\prod_{i=1}^{k} M(\hat{\varrho},\xi_i)\boldsymbol{v}\right)
\end{aligned}$$

The second equality above applies the shift invariance property IV of Lemma 3.6.3 to change ϱ_k to $\hat{\varrho}$. This is permitted as $\prod_{i=1}^{k-1} M(\varrho_i,\xi_i)\boldsymbol{v}$ is a vector in $\mathcal{U}_k$. In the third equality the shifts $\hat{\varrho}$ and ϱ_{k-1} are interchanged

based on property II of Lemma 3.6.3. The fourth equality again applies the shift invariance property IV to change ϱ_{k-1} to $\hat{\varrho}$. This is again permitted: $M(\hat{\varrho}, \xi_{k-1}) \prod_{i=1}^{k-2} M(\varrho_i, \xi_i)\boldsymbol{v}$ is a vector in $\mathcal{U}_k$ based on the induction hypothesis and (3.17). This reasoning is continued in the fifth equality where $\hat{\varrho}$ and ϱ_{k-2} are interchanged, until eventually all shifts are changed to $\hat{\varrho}$ in the final equality. □

We can now define the *rational Krylov subspaces* as the column spaces of the rational Krylov matrices from Definition 3.6.2. It follows directly from the nested shift invariance property of Theorem 3.6.4 that these subspaces are independent of the choice of P.

Definition 3.6.5 (rational Krylov subspaces)**.** We define the rational Krylov subspaces $\mathcal{K}_k^{\text{rat}}$ and $\mathcal{L}_k^{\text{rat}}$, $k \leq n$, associated with the regular pair $(A, B) \in \mathbb{C}^{n\times n}$, a vector $\boldsymbol{v} \in \mathbb{C}^n$, and pole tuple $\Xi = (\xi_1, \ldots, \xi_{k-1})$, assuming the poles different from the eigenvalues as,

$$\begin{aligned} \mathcal{K}_k^{\text{rat}}(A, B, \boldsymbol{v}, \Xi) &= \mathcal{R}(K_k^{\text{rat}}(A, B, \boldsymbol{v}, \Xi, \mathrm{P})), \\ \mathcal{L}_k^{\text{rat}}(A, B, \boldsymbol{v}, \Xi) &= \mathcal{R}(L_k^{\text{rat}}(A, B, \boldsymbol{v}, \Xi, \mathrm{P})), \end{aligned} \tag{3.18}$$

where the shift tuple P is freely chosen, assuming all shifts different from all poles.

The two rational Krylov subspaces reduce to the same subspace if A and B commute. A special case is when B is equal to the identity matrix. In this case the definition is in agreement with earlier definitions. The rational Krylov subspaces satisfy the following elementary properties.

Lemma 3.6.6 (properties of rational Krylov subspaces)**.** *The rational Krylov subspaces $\mathcal{K}^{rat}$ and $\mathcal{L}^{rat}$ generated from $(A, B) \in \mathbb{C}^{n\times n}$, $\boldsymbol{v} \in \mathbb{C}^n$, and $\Xi = (\xi_1, \ldots, \xi_{n-1})$, assuming all poles different from the eigenvalues, satisfy the following properties.*

I. They form a sequence of nested subspaces,

$$\mathcal{K}_1^{rat} \subseteq \mathcal{K}_2^{rat} \subseteq \ldots \subseteq \mathcal{K}_n^{rat} \quad \text{and} \quad \mathcal{L}_1^{rat} \subseteq \mathcal{L}_2^{rat} \subseteq \ldots \subseteq \mathcal{L}_n^{rat}. \tag{3.19}$$

II. For $k = 1, \ldots, n-1$, with the shift $\hat{\varrho}$ different from all eigenvalues and poles, and an alternative shift $\check{\varrho} \neq \hat{\varrho}$ we get:

$$\begin{aligned} \mathcal{K}_k^{rat}(A, B, \boldsymbol{v}, \Xi) &= \prod_{i=1}^{k-1} M(\hat{\varrho}, \xi_i)\ \mathcal{K}_k(M(\check{\varrho}, \hat{\varrho}), \boldsymbol{v}) = \mathcal{K}_k\left(M(\check{\varrho}, \hat{\varrho}), \prod_{i=1}^{k-1} M(\hat{\varrho}, \xi_i)\ \boldsymbol{v}\right), \\ \mathcal{L}_k^{rat}(A, B, \boldsymbol{v}, \Xi) &= \prod_{i=1}^{k-1} N(\hat{\varrho}, \xi_i)\ \mathcal{K}_k(N(\check{\varrho}, \hat{\varrho}), \boldsymbol{v}) = \mathcal{K}_k\left(N(\check{\varrho}, \hat{\varrho}), \prod_{i=1}^{k-1} N(\hat{\varrho}, \xi_i)\ \boldsymbol{v}\right), \end{aligned} \tag{3.20}$$

which connects rational Krylov subspaces with regular Krylov subspaces.

III. For $k = 1, \ldots, n-1$*, and* $\varrho_k \notin \Xi$*,*

$$\begin{aligned} M(\varrho_k, \xi_k)\ \mathcal{K}_k^{rat}(A, B, \boldsymbol{v}, \Xi) &\subseteq \mathcal{K}_{k+1}^{rat}(A, B, \boldsymbol{v}, \Xi), \\ N(\varrho_k, \xi_k)\ \mathcal{L}_k^{rat}(A, B, \boldsymbol{v}, \Xi) &\subseteq \mathcal{L}_{k+1}^{rat}(A, B, \boldsymbol{v}, \Xi). \end{aligned} \tag{3.21}$$

IV. If for any $k < n$*,* $\mathcal{K}_k^{rat} = \mathcal{K}_{k+1}^{rat}$ *or* $\mathcal{L}_k^{rat} = \mathcal{L}_{k+1}^{rat}$ *the subspaces become respectively* M*- or* N*-invariant.*

Proof. The nestedness follows directly from the definition. To prove the second property we rely on Theorem 3.6.4,

$$\begin{aligned} \mathcal{K}_k^{\mathrm{rat}}(A, B, \boldsymbol{v}, \Xi) &= \mathcal{R}\left(\boldsymbol{v}, M(\hat{\varrho}, \xi_1)\,\boldsymbol{v}, \ldots, \prod_{i=1}^{k-1} M(\hat{\varrho}, \xi_i)\,\boldsymbol{v}\right) \\ &= \prod_{i=1}^{k-1} M(\hat{\varrho}, \xi_i)\ \mathcal{R}\left(\prod_{i=1}^{k-1} M(\xi_i, \hat{\varrho})\,\boldsymbol{v}, \prod_{i=2}^{k-1} M(\xi_i, \hat{\varrho})\,\boldsymbol{v}, \ldots,\ \boldsymbol{v}\right) \\ &= \prod_{i=1}^{k-1} M(\hat{\varrho}, \xi_i)\ \mathcal{R}\left(\prod_{i=1}^{k-1} M(\check{\varrho}, \hat{\varrho})\,\boldsymbol{v}, \prod_{i=2}^{k-1} M(\check{\varrho}, \hat{\varrho})\,\boldsymbol{v}, \ldots,\ \boldsymbol{v}\right) \\ &= \prod_{i=1}^{k-1} M(\hat{\varrho}, \xi_i)\ \mathcal{K}_k(M(\check{\varrho}, \hat{\varrho}), \boldsymbol{v}). \end{aligned}$$

The first equality is the definition with $\mathrm{P} = (\hat{\varrho}, \ldots, \hat{\varrho})$. The second equality extracts the last rational term. The third equality applies the nested shift invariance property of Theorem 3.6.4 to change all shifts ξ_i to $\check{\varrho}$. We end up with a Krylov subspace in the last equality. The result for $\mathcal{L}^{\mathrm{rat}}$ is proven in a similar way. The third property follows from the second property and the nestedness of Krylov subspaces, setting $\hat{\varrho} = \varrho_k$. The fourth property follows from (3.21) by imposing $\mathcal{K}_k^{\mathrm{rat}} = \mathcal{K}_{k+1}^{\mathrm{rat}}$ or $\mathcal{L}_k^{\mathrm{rat}} = \mathcal{L}_{k+1}^{\mathrm{rat}}$ □

We remark that property II states that rational Krylov subspaces are nothing else than Krylov subspaces whose starting vector is modified by a rational function determined by the poles Ξ.

3.6.2 Proper Hessenberg pairs and rational Krylov

In the previous section (A, B) could be any regular pair. Now we'll see that if (A, B) is a proper Hessenberg pair, the rational Krylov subspaces and matrices have a special structure.

Theorem 3.6.7. *Let $(A, B) \in \mathbb{C}^{n\times n}$ be a proper Hessenberg pair having poles $\Xi = (\xi_1, \ldots, \xi_{n-1})$ distinct from the eigenvalues. Then for k from 1 to n,*

$$\mathcal{K}_k^{rat}(A, B, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_{k-1})) = \mathcal{E}_k, \tag{3.22}$$

while for k from 1 to $n-1$,

$$\mathcal{L}_k^{rat}(A, B, \boldsymbol{e}_1, (\xi_2, \ldots, \xi_k)) = \mathcal{E}_k. \tag{3.23}$$

Proof. We prove the results by induction on the subspace dimension. The case $k = 1$ is trivial for both statements. To prove (3.22), assume the result holds up to dimension $k \leq n-1$,

$$\mathcal{K}_k^{\text{rat}}(A, B, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_{k-1})) = \mathcal{E}_k.$$

From the nestedness of rational Krylov subspaces, we have by induction,

$$\mathcal{E}_k \subseteq \mathcal{K}_{k+1}^{\text{rat}}(A, B, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_k)).$$

It remains to be shown that $\boldsymbol{e}_{k+1} \in \mathcal{K}_{k+1}^{\text{rat}}(A, B, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_k))$. From (3.21) and the induction hypothesis we deduce,

$$M(\varrho_k, \xi_k)\ \mathcal{E}_k \subseteq \mathcal{K}_{k+1}^{\text{rat}}(A, B, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_k)), \tag{3.24}$$

for $\varrho_k \notin \Xi$. Now consider the vector $\boldsymbol{k}_k = (\beta_k A - \alpha_k B)\boldsymbol{e}_k$, with $\alpha_k/\beta_k = \xi_k$. As $\beta_k A - \alpha_k B$ is an upper Hessenberg matrix with a zero in position $(k+1, k)$, $\boldsymbol{k}_k \in \mathcal{E}_k$. It follows that,

$$\boldsymbol{k}_{k+1} = M(\varrho_k, \xi_k)\ \boldsymbol{k}_k = (\nu_k A - \mu_k B)(\beta_k A - \alpha_k B)^{-1}\ \boldsymbol{k}_k = (\nu_k A - \mu_k B)\ \boldsymbol{e}_k,$$

is a vector in $\mathcal{E}_{k+1}$ with $k_{k+1} \neq 0$ and by (3.24), $\boldsymbol{k}_{k+1} \in \mathcal{K}_{k+1}^{\text{rat}}$. This proves the first result.

In order to prove (3.23), we can start in a similar way. Assume the result holds up to dimension $k < n-1$[2]. We get from the nestedness of rational Krylov subspaces and the induction hypothesis that,

$$\mathcal{E}_k \subseteq \mathcal{L}_{k+1}^{\text{rat}}(A, B, \boldsymbol{e}_1, (\xi_2, \ldots, \xi_{k+1})).$$

[2]For $\mathcal{K}_k^{\text{rat}}$, $k+1$ can be as large as n since (3.22) goes up to ξ_{k-1}. For $\mathcal{L}_k^{\text{rat}}$, $k+1$ is limited to $n-1$ as we don't want to run out of poles.

From (3.21) and the induction hypothesis we deduce,

$$N(\varrho_{k+1}, \xi_{k+1})\ \mathcal{E}_k \subseteq \mathcal{L}^{\text{rat}}_{k+1}(A, B, \boldsymbol{e}_1, (\xi_2, \ldots, \xi_{k+1})),$$

for $\varrho_{k+1} \notin \Xi$. To complete the proof, we need to show as before that there exists a pair of vectors $\boldsymbol{\ell}_k, \boldsymbol{\ell}_{k+1}$, with $\boldsymbol{\ell}_k \in \mathcal{E}_k$ and $\boldsymbol{\ell}_{k+1} \in \mathcal{E}_{k+1}$ whose $(k{+}1)$st element $\ell_{k+1} \neq 0$, that are related as,

$$\boldsymbol{\ell}_{k+1} = N(\varrho_{k+1}, \xi_{k+1})\ \boldsymbol{\ell}_k = (\beta_{k+1}A{-}\alpha_{k+1}B)^{-1}(\nu_{k+1}A{-}\mu_{k+1}B)\ \boldsymbol{\ell}_k, \quad (3.25)$$

An explicit construction is not possible in this case. Nonetheless, by (3.25) we have that $(\boldsymbol{\ell}_k, \boldsymbol{\ell}_{k+1})$ must satisfy

$$(\beta_{k+1}A{-}\alpha_{k+1}B)\ \boldsymbol{\ell}_{k+1} = (\nu_{k+1}A{-}\mu_{k+1}B)\ \boldsymbol{\ell}_k.$$

From properties I and II of Lemma 3.2.2, we have that the matrix $\beta_{k+1}A{-}\alpha_{k+1}B$ is an upper Hessenberg matrix that admits a block upper triangular partition with a leading block of size $(k{+}1){\times}(k{+}1)$, while the matrix $\nu_{k+1}A - \mu_{k+1}B$ is a proper upper Hessenberg matrix since the shift ϱ_{k+1} is different from all the poles. Observe that all vectors $\boldsymbol{\ell}_k \in \mathcal{E}_k$ would lead to a vector $\boldsymbol{\ell}_{k+1}$ with element $\ell_{k+1} = 0$ if and only if the first k columns of $(\beta_{k+1}A{-}\alpha_{k+1}B)$ would span the same subspace as the first k columns of $(\nu_{k+1}A{-}\mu_{k+1}B)$. It follows from property III and IV of Lemma 3.2.2 that this cannot be true. We conclude that a valid pair $(\boldsymbol{\ell}_k, \boldsymbol{\ell}_{k+1})$ must exist. □

A direct corollary of the theorem considers the structure of rational Krylov matrices generated from proper Hessenberg pairs.

Corollary 3.6.8. *Let $(A, B) \in \mathbb{C}^{n\times n}$ be a proper Hessenberg pair with poles $\Xi = (\xi_1, \ldots, \xi_{n-1})$ different from the eigenvalues of (A, B) and let $(\varrho_1, \ldots, \varrho_{n-1})$ be a shift tuple different from the poles. Then, for k from 1 to n,*

$$K^{rat}_k(A, B, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_{k-1}), (\varrho_1, \ldots, \varrho_{k-1})),$$

and, for k from 1 to $n{-}1$,

$$L^{rat}_k(A, B, \boldsymbol{e}_1, (\xi_2, \ldots, \xi_k), (\varrho_2, \ldots, \varrho_k)),$$

are upper triangular matrices with non-vanishing diagonal elements.

3.6.3 Proof of the implicit Q theorem

We are ready to prove Theorem 3.6.1.

Proof. Choose a tuple of $n-1$ shifts P different from the poles Ξ. Corollary 3.6.8 states that $K_n^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, \Xi, \mathrm{P})$ and $K_n^{\mathrm{rat}}(\check{A}, \check{B}, \boldsymbol{e}_1, \Xi, \mathrm{P})$ are $n \times n$ nonsingular upper triangular matrices. The elementary rational matrix $M(\varrho, \xi)$ is transformed via $\hat{Q}$ and $\check{Q}$ to:

$$\hat{M}(\varrho, \xi) = \hat{Q}^* M(\varrho, \xi) \hat{Q} \quad \text{and} \quad \check{M}(\varrho, \xi) = \check{Q}^* M(\varrho, \xi) \check{Q}.$$

It follows that,

$$\begin{aligned}
&\hat{Q}\, K_n^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, \Xi, \mathrm{P}) \\
&= \hat{Q} \left[\boldsymbol{e}_1 \; \hat{M}(\varrho_1, \xi_1)\, \boldsymbol{e}_1 \; \ldots \; \left(\prod_{i=1}^{n-1} \hat{M}(\varrho_i, \xi_i) \right) \boldsymbol{e}_1 \right] \\
&= \hat{Q} \left[\boldsymbol{e}_1 \; \hat{Q}^* M(\varrho_1, \xi_1) \hat{Q}\, \boldsymbol{e}_1 \; \ldots \; \hat{Q}^* \left(\prod_{i=1}^{n-1} M(\varrho_i, \xi_i) \right) \hat{Q}\, \boldsymbol{e}_1 \right] \\
&= \left[\hat{\boldsymbol{q}}_1 \; M(\varrho_1, \xi_1)\, \hat{\boldsymbol{q}}_1 \; \ldots \; \left(\prod_{i=1}^{n-1} M(\varrho_i, \xi_i) \right) \hat{\boldsymbol{q}}_1 \right] \\
&= \sigma \left[\check{\boldsymbol{q}}_1 \; M(\varrho_1, \xi_1)\, \check{\boldsymbol{q}}_1 \; \ldots \; \left(\prod_{i=1}^{n-1} M(\varrho_i, \xi_i) \right) \check{\boldsymbol{q}}_1 \right] \\
&= \sigma \check{Q}\, K_n^{\mathrm{rat}}(\check{A}, \check{B}, \boldsymbol{e}_1, \Xi, \mathrm{P}).
\end{aligned}$$

Since the upper triangular matrices K_n^{rat} are nonsingular, the uniqueness of the QR factorization, stated in Lemma 2.3.3, implies the existence of a unitary diagonal matrix D_1 such that $\hat{Q} = \check{Q} D_1$.

It remains to prove that a similar relation holds for the matrices $\hat{Z}$ and $\check{Z}$. Let us first prove that $\hat{Z}$ and $\check{Z}$ also share a first column up to unimodular scaling. From the relations $(\beta_1 \hat{A} - \alpha_1 \hat{B}) = \hat{Q}^* (\beta_1 A - \alpha_1 B) \hat{Z}$ and $(\beta_1 \check{A} - \alpha_1 \check{B}) = \check{Q}^* (\beta_1 A - \alpha_1 B) \check{Z}$, with $\xi_1 = \alpha_1 / \beta_1$, it follows that,

$$\begin{aligned}
\hat{\boldsymbol{z}}_1 = \hat{Z} \boldsymbol{e}_1 &= (\beta_1 A - \alpha_1 B)^{-1} \hat{Q} (\beta_1 \hat{A} - \alpha_1 \hat{B}) \boldsymbol{e}_1, \\
\check{\boldsymbol{z}}_1 = \check{Z} \boldsymbol{e}_1 &= (\beta_1 A - \alpha_1 B)^{-1} \check{Q} (\beta_1 \check{A} - \alpha_1 \check{B}) \boldsymbol{e}_1.
\end{aligned} \tag{3.26}$$

Since both $(\beta_1 \hat{A} - \alpha_1 \hat{B}) \boldsymbol{e}_1$ and $(\beta_1 \check{A} - \alpha_1 \check{B}) \boldsymbol{e}_1$ reduce to a scalar multiple of $\boldsymbol{e}_1$ and $\hat{Q} \boldsymbol{e}_1 = \sigma \check{Q} \boldsymbol{e}_1$ we get $\check{\boldsymbol{z}}_1 = \tilde{\sigma} \hat{\boldsymbol{z}}_1$. Using Corollary 3.6.8 and similar reasoning as before, it is shown that the following two QR factorizations are equal,

$$\hat{Z}\, L_{n-1}^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, \Xi_s, \mathrm{P}_s) = \tilde{\sigma} \check{Z} L_{n-1}^{\mathrm{rat}}(\check{A}, \check{B}, \boldsymbol{e}_1, \Xi_s, \mathrm{P}_s),$$

with $\Xi_s = (\xi_2, \ldots, \xi_{n-1})$ and $\mathrm{P}_s = (\varrho_2, \ldots, \varrho_{n-1})$. In this case the L_{n-1} matrices are of size $n \times n-1$. Uniqueness of the QR factorization implies essential uniqueness of the first $n-1$ columns of $\hat{Z}$ and $\check{Z}$. Nonetheless also the last column of $\hat{Z}$ and $\check{Z}$ are essentially the same as they are orthogonal to the first $n-1$ columns. We conclude that $\hat{Z} = \check{Z} D_2$, with D_2 a unitary diagonal matrix. □

When the Hessenberg pair is not proper, uniqueness can only be guaranteed up to the pole that causes the problem. This is similar to the Hessenberg case. In practice this is in fact good news as a breakdown signals a deflation.

3.7 Implicit rational subspace iteration

Francis' QR algorithm [39, 40] effects nested subspace iteration with a change of coordinate system accelerated by polynomial Krylov subspaces, see Theorem 2.3.6. Theorem 2.3.8 showed that the convergence behaviour of the QZ method is also determined by polynomials. This result is generalized in this section for the rational QZ method.

We first give a result which relates the invariant subspaces of the elementary rational matrices (3.7) to the deflating subspaces of (A, B). This might help the reader to gain more insight in the convergence result. The following lemma is useful in the proof of the next theorem. It makes use of the Hermitian conjugate of a subspace which is characterized as $\boldsymbol{v}^* \in \mathcal{S}^*$ if and only if $\boldsymbol{v} \in \mathcal{S}$.

Lemma 3.7.1. *Let (A, B) be a regular matrix pair, $\mu, \nu, \alpha, \beta \in \mathbb{C}$ such that $\mu\beta \neq \alpha\nu$, and $\mathcal{S} \subseteq \mathbb{C}^n$. Then,*

$$(\beta A - \alpha B)\mathcal{S} + (\nu A - \mu B)\mathcal{S} = A\mathcal{S} + B\mathcal{S}$$
$$\mathcal{S}^*(\beta A - \alpha B) + \mathcal{S}^*(\nu A - \mu B) = \mathcal{S}^* A + \mathcal{S}^* B$$

Proof. We prove the first result for a *right* subspace, the proof of the second result is similar. We clearly have that,

$$(\beta A - \alpha B)\mathcal{S} + (\nu A - \mu B)\mathcal{S} \subseteq A\mathcal{S} + B\mathcal{S},$$

because for $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathcal{S}$ we have:

$$(\beta A - \alpha B)\boldsymbol{s}_1 + (\nu A - \mu B)\boldsymbol{s}_2 = A(\beta \boldsymbol{s}_1 + \nu \boldsymbol{s}_2) + B(-\alpha \boldsymbol{s}_1 - \mu \boldsymbol{s}_2) \in A\mathcal{S} + B\mathcal{S}.$$

On the other hand, $\boldsymbol{x} \in A\mathcal{S} + B\mathcal{S}$ satisfies,

$$\boldsymbol{x} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix},$$

for some $\boldsymbol{y}, \boldsymbol{z} \in \mathcal{S}$. Consider the invertible matrix $T = \left[\begin{smallmatrix} \beta & \nu \\ -\alpha & -\mu \end{smallmatrix}\right]$, which yields:

$$\boldsymbol{x} = \begin{bmatrix} A & B \end{bmatrix} (T \otimes I)(T^{-1} \otimes I) \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} \beta A - \alpha B & \nu A - \mu B \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{y}} \\ \hat{\boldsymbol{z}} \end{bmatrix},$$

with $\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}} \in \mathcal{S}$. □

The following theorem extends [72, Theorem 1.6.5].

Theorem 3.7.2. *Let (A, B) be a regular matrix pair, $M(\varrho, \xi)$ and $N(\varrho, \xi)$ be two elementary rational matrices generated by (A, B) with $\varrho \neq \xi$ and $\xi \notin \Lambda$ according to* (3.7)*. Then,*

I. A subspace $\mathcal{Y}$ is right-invariant under $N(\varrho, \xi)$ if and only if $\mathcal{Y}$ is a right-deflating subspace for (A, B).

II. A subspace $\mathcal{X}^*$ is left-invariant under $M(\varrho, \xi)$ if and only if $\mathcal{X}^*$ is a left-deflating subspace for (A, B).

Proof. We prove only the first statement; the second one is similar. It follows from Lemma 3.7.1 that for $\mathcal{S} \subseteq \mathbb{C}^n$,

$$\dim(A\mathcal{S}+B\mathcal{S}) = \dim((\beta A-\alpha B)\mathcal{S}+(\nu A-\mu B)\mathcal{S}) = \dim(\mathcal{S}+N(\varrho,\xi)\mathcal{S}), \quad (3.27)$$

with $\varrho = \mu/\nu$, $\xi = \alpha/\beta \notin \Lambda$. Assume $\mathcal{S}$ is a right-deflating subspace for (A, B). According to (2.10) this implies $\dim(A\mathcal{S} + B\mathcal{S}) \leq \dim(\mathcal{S})$, such that also $\dim(\mathcal{S} + N(\varrho, \xi)\mathcal{S}) \leq \dim(\mathcal{S})$ by (3.27). The latter implies that $N(\varrho, \xi)\mathcal{S} \subseteq \mathcal{S}$, i.e. $\mathcal{S}$ is a right-invariant subspace of $N(\varrho, \xi)$. Conversely, if $\mathcal{S}$ is a right-invariant subspace of $N(\varrho, \xi)$ it follows that $\dim(\mathcal{S} + N(\varrho, \xi)\mathcal{S}) = \dim(\mathcal{S})$, from (3.27) and (2.10) we get that $\mathcal{S}$ is also a right-deflating subspace of (A, B). □

We are now ready to study the subspace iteration of the RQZ method. Starting with a proper Hessenberg pair (A, B) with $\Xi = (\xi_1, \ldots, \xi_{n-1})$, a single iteration of the rational QZ method with shift ϱ and new pole $\hat{\xi}_{n-1}$ results in a new proper Hessenberg pair,

$$(\hat{A}, \hat{B}) = Q^* \, (A, B) \, Z,$$

with $\hat{\Xi} = (\xi_2, \ldots, \xi_{n-1}, \hat{\xi}_{n-1})$. This equivalence transformation simultaneously performs two similarity transformations on the matrices,

$$\hat{M}(\varrho, \xi) = Q^* \, M(\varrho, \xi) \, Q \quad \text{and} \quad \hat{N}(\varrho, \xi) = Z^* \, N(\varrho, \xi) \, Z, \qquad (3.28)$$

for all ϱ and ξ.

The following theorem formalizes the convergence behavior of the RQZ method.

Theorem 3.7.3. *Consider a single RQZ step* $(\hat{A}, \hat{B}) = Q^* (A, B) Z$*, with shift* ϱ*, pole tuple* $\Xi = (\xi_1, \ldots, \xi_{n-1})$ *prior to the RQZ step, and* $\hat{\Xi} = (\xi_2, \ldots, \xi_{n-1}, \hat{\xi}_{n-1})$ *afterwards. Assume all poles different from the eigenvalues, and the shift* ϱ *different from all eigenvalues and poles. For* $k = 1, \ldots, n-1$*, this effects subspace iteration driven by* $M(\varrho, \xi_k)$ *and* $N(\varrho, \xi_{k+1})$*, we get:*

$$\mathcal{R}(Q(:, 1:k)) = M(\varrho, \xi_k)\, \mathcal{E}_k, \qquad \text{and} \qquad \mathcal{R}(Z(:, 1:k)) = N(\varrho, \xi_{k+1})\, \mathcal{E}_k, \tag{3.29}$$

with $\xi_n \coloneqq \hat{\xi}_{n-1}$*. The change of coordinate system maps both* $\mathcal{R}(Q(:, 1:k))$ *and* $\mathcal{R}(Z(:, 1:k))$ *back to* $\mathcal{E}_k$*.*

Proof. We make use of the result from Lemma 3.6.3, Lemma 3.6.6, Theorem 3.6.7, (3.28) and $\boldsymbol{q}_1 = \gamma M(\varrho, \xi_1)\, \boldsymbol{e}_1$ by (3.2). We get,

$$\begin{aligned}
\mathcal{R}(Q(:, 1:k)) = Q\, \mathcal{E}_k &= Q\, \mathcal{K}_k^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, \hat{\Xi}) \\
&= Q \prod_{i=2}^{k} \hat{M}(\varrho, \xi_i) \cdot \mathcal{K}_k(\hat{M}(\check{\varrho}, \varrho), \boldsymbol{e}_1) \\
&= \prod_{i=2}^{k} M(\varrho, \xi_i) \cdot \mathcal{K}_k(M(\check{\varrho}, \varrho), Q\boldsymbol{e}_1) \\
&= \prod_{i=2}^{k} M(\varrho, \xi_i) \cdot \mathcal{K}_k(M(\check{\varrho}, \varrho), M(\varrho, \xi_1)\boldsymbol{e}_1) \\
&= M(\varrho, \xi_k) \prod_{i=1}^{k-1} M(\varrho, \xi_i) \cdot \mathcal{K}_k(M(\check{\varrho}, \varrho), \boldsymbol{e}_1) \\
&= M(\varrho, \xi_k)\, \mathcal{E}_k.
\end{aligned}$$

The second equality uses Theorem 3.6.7. The third equality applies part II of Lemma 3.6.6. The fourth equality relies on (3.28) to change from $\hat{M}$ to M. The fifth equality uses the expression for $\boldsymbol{q}_1$, the sixth uses the commutativity property, and the last equality again applies Lemma 3.6.6 and Theorem 3.6.7.

The second result follows a similar reasoning. The only difference is the relation between $\boldsymbol{z}_1$ and $\boldsymbol{e}_1$. Starting from the same argument as in (3.26) we get, for some constants $\gamma, \check{\gamma}$ and $\tilde{\gamma}$,

$$\boldsymbol{z}_1 = \gamma(\beta_2 A - \alpha_2 B)^{-1} \boldsymbol{q}_1 = \check{\gamma}\, (\beta_2 A - \alpha_2 B)^{-1} M(\varrho, \xi_1)\, \boldsymbol{e}_1 = \tilde{\gamma}\, N(\varrho, \xi_2)\, \boldsymbol{e}_1.$$

□

A single shifted RQZ step will execute a QR step with shift ϱ on the entire space simultaneously with RQ steps having shifts ξ_i on selected subspaces. The shift ϱ is rapidly moving from top to bottom and thus affects all subspaces. The poles on the other hand are slowly moving upwards, one row during each step, and as such do not act on all subspaces in a single RQZ step. The shifts will rapidly initiate convergence at the bottom, the poles slowly push converged eigenvalues to the top. This is an explanation for why, in the classical QZ algorithm, the zero eigenvalues in B appear at the top: they are pushed there by the poles at infinity [137]. Moreover, it is also clear from the analysis that picking a shift equal to a pole will lead to cancellation in some of the factors thereby slowing down convergence.

Note that in the formulation of Theorem 3.7.3 the shift and poles are assumed to be different from the eigenvalues of the matrix pair. This is imposed to ensure that the required inverses exist. However, in practical implementations, these parameters will typically converge towards an eigenvalue. This is in fact a desirable situation as it will lead to deflations as we will show in Section 3.8.

In the QZ algorithm [83], all poles are at ∞ and the two driving functions reduce to $M(\varrho, \infty)$ and $N(\varrho, \infty)$ which is equivalent to $AB^{-1} - \varrho I$ and $B^{-1}A - \varrho I$, cfr. Theorem 2.3.8. This connection also explains why we assumed B invertible in Theorem 2.3.7 and Theorem 2.3.8. A proper Hessenberg pair in Hessenberg, triangular form has all poles at ∞ and as the poles are required to be different from the eigenvalues for our theory to hold, B should be nonsingular.

In the RQZ method, the poles can be chosen freely and as such they can be utilized to influence the convergence of the method as was illustrated in the numerical experiments of Sections 3.4.2 and 3.5.3. Note that, as the poles only shift one row up during every RQZ step, it takes $n-1$ iterations before a pole has moved from the bottom to the top and has influenced all vectors in the subspace iteration.

A more modular convergence analysis for pole swapping algorithms is included in [16]. In this paper, the subspace iteration that is executed by a single swap is studied based on Theorem 3.6.7. Furthermore, it is shown how these *mini-iterations* can be combined to prove Theorem 3.7.3 for the RQZ method and, more general, how similar results can be deduced for any pole swapping method, e.g. a *reverse* RQZ step where a shift is swapped from the bottom to the top of the pencil.

3.7.1 An example of a rational filter

To further clarify the result of Theorem 3.7.3 consider the simplified case where all the poles of the Hessenberg pair are equal to same value ξ different from the eigenvalues of (A, B). Assume that the RQZ algorithm is applied s times on this proper Hessenberg pair with the same shift ϱ. At the end of each RQZ step the last pole is again restored to ξ. Then the subspace iterations, as considered from the initial pair, are given by,

$$Q:\ \mathcal{E}_k \to M(\varrho,\xi)^s\mathcal{E}_k, \quad \text{and} \quad Z:\ \mathcal{E}_k \to N(\varrho,\xi)^s\mathcal{E}_k.$$

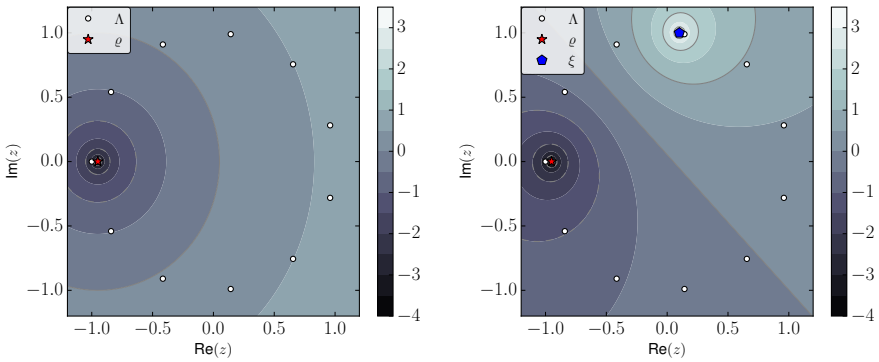
Denote $q(z) = (z-\varrho)/(z-\xi)$ and let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of the pair (A, B), so that $q(\lambda_i)^s$ is the rational filter that is implicitly applied during these s iterations to λ_i. Assume the eigenvalues are ordered such that,

$$|q(\lambda_1)^s| \leq |q(\lambda_2)^s| \leq \ldots \leq |q(\lambda_{n-1})^s| \leq |q(\lambda_n)^s|,$$

then the convergence factor of an eigenvalue at the end of the Hessenberg pencil is given by $|q(\lambda_1)^s|/|q(\lambda_2)^s|$, while the convergence factor at the top of the Hessenberg pencil is given by $|q(\lambda_{n-1})^s|/|q(\lambda_n)^s|$. As such, a good choice of both poles and shifts can accelerate convergence and lead to deflations.

As a concrete example consider a problem of size 11 with eigenvalues located on the unit circle in the complex plane. Figure 3.14 shows the absolute value of the rational filter after s=2 iterations for two different choices for the rational function q. Pane (a) shows the filter, $q_\infty(z)^2$, with shift $\varrho = -0.95$ and all the poles at ∞. This situation corresponds to the QZ method applied twice with the same shift to a Hessenberg, triangular pair. The shift ϱ is located close to the eigenvalue $\lambda_1 = -1$ such that $|q_\infty(\lambda_1)^2| = 2.5 \cdot 10^{-3}$ is the minimal value of the filter over all eigenvalues. The convergence factor of λ_1 at the end of the pencil is approximately $8.22 \cdot 10^{-3}$. At the top of the pencil there is no convergence in this case as $|q(\lambda_{n-1})^2|/|q(\lambda_n)^2| = 1$. Pane (b) shows the same experiment but this time the poles are located at $\xi = 0.1+1i$ which is in the vicinity of another eigenvalue. This situation corresponds to the RQZ method applied twice with the same shift to a Hessenberg pair with $\Xi = (\xi, \ldots, \xi)$. The rational filter, $q_\xi(z)^2$, leads to a convergence factor of λ_1 at the end of the pencil of approximately $1.21 \cdot 10^{-2}$. The convergence of λ_1 at the end of the pencil is slower with q_ξ^2 compared to q_∞^2. However, q_ξ^2 will also lead to convergence at the top of the pencil as the convergence factor is $|q(\lambda_{n-1})^2|/|q(\lambda_n)^2| \approx 7.46 \cdot 10^{-3}$. We observe that using q_ξ leads to convergence of another eigenvalue, where q_∞ does not.

It is clear that both the shifts and the poles can accelerate the convergence but they do influence each other.



(a) Filter with $\xi = \infty$

(b) Filter with $\xi = 0.1 + 1i$

Figure 3.14: Logarithm of the absolute value of the rational filter, $|q(\lambda_i)^s|$, after $s = 2$ iterations with $\varrho = -0.95$, and ξ either at ∞ or at $0.1 + 1i$. The eigenvalues λ_i are shown with circles, the shift ϱ is indicated with a star, the pole with a pentagon. Darker regions agree with convergence at the end of the pencil and lighter regions with convergence at the top of the pencil.

When the shifts are changed in every iteration and the poles of the Hessenberg pair are not the same then the filter q becomes dependent on the index k and will be a product of terms with different shifts,

$$q_k(\lambda) = \prod_{i=1}^{s} (\lambda - \varrho_i)/(\lambda - \xi_k^{(i)}), \tag{3.30}$$

with $\xi_k^{(i)}$ the pole at iteration i in position k (or k+1) for Q (or Z) as shown in Theorem 3.7.3.

Provided a good choice of shifts and poles is made during repeated application of the RQZ algorithm the pair (A, B) will converge to a pair of upper triangular matrices.

3.8 Perfect shifts in rational QZ

The previous section studied the convergence of the rational QZ method by exploiting the connection with rational Krylov to analyze the underlying subspace iteration. In this section, we take a different approach and show that if we have a shift ϱ at our disposal that is an exact eigenvalue of the proper

Hessenberg pencil, then a single rational QZ step deflates it. This result is related to recent work on perfect shifts in the QR method [82].

We remark that, unlike the results in Sections 3.6 and 3.7, Theorem 3.8.1 does not assume the shift and poles to be different from the eigenvalues.

Theorem 3.8.1. *Let $A - \lambda B$ be an $n \times n$ proper Hessenberg pencil with pole tuple $\Xi = (\xi_1, \ldots, \xi_{n-1})$, where the poles are not necessarily distinct from the eigenvalues. Furthermore, let $\varrho \in \bar{\mathbb{C}}$ be an eigenvalue of $A - \lambda B$ with $\varrho \notin \Xi$. Then we have that a rational QZ step with shift ϱ on $A - \lambda B$ leads to a deflation in the last row in exact arithmetic.*

Proof. Consider $H = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] := A - \varrho B$ which is a proper Hessenberg *matrix* since $h_{i+1,i} \neq 0$ for $i = 1, \ldots n-1$ as $\varrho \notin \Xi$. H is also singular as $\det(A - \varrho B) = 0$ by assumption. It follows from the properness of the Hessenberg structure in H that the subspace generated by its $n-1$ first columns, $\mathcal{R}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{n-1})$, is of maximal dimension $n-1$. In combination with the singularity, it is clear that the following property is satisfied:

$$\boldsymbol{h}_n \in \mathcal{R}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{n-1}). \tag{3.31}$$

The first step in the rational QZ method is to introduce the shift ϱ in $A - \lambda B$ by rotating the first two rows of the pencil with an appropriate rotation $Q_{1:2}^*$. This is always possible thanks to the properness of the pencil. The subscript in the equivalence transformations indicates on which rows or columns they act. We get,

$$\hat{A} - \lambda \hat{B} := Q_{1:2}^*(A - \lambda B),$$

such that $\hat{\Xi} = (\varrho, \xi_2, \ldots, \xi_{n-1})$. Merely rotating the first two rows does not affect the properness, so $\hat{A} - \lambda \hat{B}$ is still a proper Hessenberg pencil. The Hessenberg matrix $\hat{H} := \hat{A} - \varrho \hat{B}$ is however no longer proper as $\hat{h}_{2,1} = 0$. Nonetheless, (3.31) is preserved for $\hat{H}$:

$$\hat{\boldsymbol{h}}_n \in \mathcal{R}(\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_{n-1}), \tag{3.32}$$

as the column spaces clearly do not change under the transition from H to $\hat{H}$.

The second step in the rational QZ method is to swap ϱ to position $n, n-1$. This is done by an equivalence,

$$\check{A} - \lambda \check{B} = Q_{2:n}^*(\hat{A} - \lambda \hat{B}) Z_{1:n-1},$$

such that $\check{A} - \lambda \check{B}$ is a Hessenberg pencil with $\check{\Xi} = (\xi_2, \ldots, \xi_{n-1}, \varrho)$. We will now show that this pencil cannot be proper. Denote $\check{H} := \check{A} - \varrho \check{B}$ which is a

Hessenberg matrix with $\check{h}_{n,n-1} = 0$. Property (3.31) for H and (3.32) for $\check{H}$ still holds in a similar fashion for $\check{H}$:

$$\check{\boldsymbol{h}}_n \in \mathcal{R}(\check{\boldsymbol{h}}_1, \ldots, \check{\boldsymbol{h}}_{n-1}), \tag{3.33}$$

because $Q^*_{2:n}$ does not change the column spaces and $Z_{1:n-1}$ is an invertible transformation on the first $n-1$ columns such that,

$$\mathcal{R}(\check{\boldsymbol{h}}_1, \ldots, \check{\boldsymbol{h}}_{n-1}) = \mathcal{R}(\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_{n-1}) = \mathcal{R}(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{n-1}).$$

Combining (3.33) and $\check{h}_{n,n-1} = 0$ directly implies that also $\check{h}_{n,n} = 0$ by the Hessenberg structure of $\check{H}$. This in turn means that $\check{A} - \lambda\check{B}$ is deflatable in its last row. □

Remark 3.8.2. Theorem 3.8.1 does not hold in floating point arithmetic as the effect of shift blurring may cause $\check{h}_{n,n-1} \neq 0$. It cannot be guaranteed that a single iteration will suffice.

Remark 3.8.3. We point out that Theorem 3.8.1 holds in a similar fashion for a *perfect pole*. If we introduce an eigenvalue as a pole at the end of the subdiagonal, it will cause a deflation in the first columns of $A - \lambda B$ once it has been swapped to the first subdiagonal position. This under the assumptions that the pole is not yet present in Ξ and that all operations are carried out in exact arithmetic.

3.9 Conclusion

In this chapter we proposed a rational QZ algorithm for the numerical solution of the dense, unsymmetric, generalized eigenvalue problem. The new algorithm operates on matrix pairs in Hessenberg, Hessenberg form rather than the Hessenberg, triangular form used in the classical QZ method. Hessenberg pairs link to rational Krylov and the associated poles are encoded in the subdiagonal elements of both Hessenberg matrices. We presented a backward stable algorithm to compute the swapping transformations. A direct reduction method of a regular matrix pair to Hessenberg, Hessenberg form was proposed. Moreover, we have demonstrated that a good choice of poles can lead to premature middle deflations during the reduction phase. The iterative rational QZ algorithm differs from the classical QZ algorithm in the sense that also poles can be introduced in each QZ step. Numerical experiments confirm that a good choice of poles allows the RQZ method to outperform the QZ algorithm by reducing the number of iterations per eigenvalue. The implicit chasing technique is justified by an implicit Q theorem, which is proved in a novel manner operating

directly on the matrix pair and exploiting the connections with rational Krylov. Our theoretical analysis revealed that an RQZ iteration implicitly performs nested subspace iteration accelerated by rational functions. Finally, we proved an exactness result which shows that the RQZ method deflates a perfect shift in a single iteration.

Chapter 4

A multishift, multipole rational QZ method with aggressive early deflation

This chapter is based on [18]:

Camps D., Meerbergen K., and Vandebril R. A multishift, multipole rational QZ method with aggressive early deflation. (2019) Submitted.

Part of Section 4.4 is based on [17]:

Camps D., Mastronardi N., Vandebril R., and Van Dooren P. Swapping 2×2 blocks in the Schur and generalized Schur form. (2019) Accepted for publication in J. Comput. Appl. Math.

4.1 Introduction

The rational QZ method that we presented in the previous chapter is useful to solve the unsymmetric generalized eigenvalue problem defined by a pair of matrices $A, B \in \mathbb{F}^{n \times n}$, $\mathbb{F} \in \{\mathbb{C}, \mathbb{R}\}$. The method acts on pencils in Hessenberg,

Hessenberg form instead of the Hessenberg, triangular form used in the QZ method. It relies on *pole swapping* instead of *bulge chasing*.

Both the single shift RQZ method and the RQZ method with tightly-packed shifts, as formulated in the previous chapter, are applicable to real- and complex-valued pencils. However it requires complex arithmetic for real-valued pencils having complex conjugate eigenvalues. The RQZ method computes the generalized Schur form (2.12) of (A, B) and not the real generalized Schur form (2.13).

In the current chapter we introduce the *multishift, multipole RQZ method* which acts on pencils in *block Hessenberg* form. The main benefit of using shifts and poles of higher multiplicity is that complex conjugate pairs of shifts and poles can be represented in real arithmetic for real-valued pencils.

This is similar to the well-known implicit double-shift QR step introduced by Francis [40] and the double-shift QZ step [83] we discussed in Chapter 2. The focus of this chapter is thus on the case $\mathbb{F} = \mathbb{R}$. The multishift, multipole RQZ method no longer converges to the triangular, triangular pencil of the generalized Schur form (2.12). Instead, for $A, B \in \mathbb{R}^{n\times n}$, it will converge to the generalized real Schur form (2.13).

The remainder of this chapter consists of two parts. Sections 4.2 and 4.3 make up the theoretical part. In Section 4.2, we study matrix pencils in block Hessenberg form. We extend the definition of properness to block Hessenberg pencils, and define their *pole pencil* and *pole tuple*. We show how the pole tuple can be altered by changing pole blocks at the edge of the pencil and by swapping neighboring pole blocks. The multishift, multipole RQZ step follows directly from this discussion. Section 4.3 extends the implicit Q theorem for Hessenberg pencils (Theorem 3.6.1) to block Hessenberg pencils and briefly discusses the convergence behaviour of the multishift, multipole method.

In the second part of the chapter, we follow a more practical approach and discuss how a multishift, multipole RQZ method can be implemented in finite precision arithmetic. The QR method suffers from a degraded performance when moderate to large shift multiplicities are used. Watkins [135] studied this phenomenon and demonstrated that shifts become *blurred* during a QR iteration of higher shift multiplicity. This severely decreases the effectiveness of the shifts. For the QR method, this issue is mitigated in the *small bulge* multishift variant introduced by Braman, Byers & Mathias [14]. This approach is extended to the QZ method by Kågström & Kressner [58]. In Section 4.4, we demonstrate that the multishift, multipole RQZ method is also prone to numerical issues when shifts and poles of moderate to large multiplicities are used. To overcome the numerical difficulties, we propose a multishift, multipole RQZ method that uses

tightly-packed, small blocks. Specifically, blocks of dimension 2×2 for complex conjugate shifts and poles in real pencils and of dimension 1×1 for real shifts and poles in real pencils and in complex pencils. In Section 4.4, we pay special attention to the backward stability of the swapping operations that are required in the algorithm.

The last tool we adapt from recent improvements to the QR [15] and QZ [58] methods to the RQZ method is the use of advanced deflation strategies. Specifically we implement the *aggressive early deflation* technique during the RQZ iteration in order to obtain level-3 BLAS performance. This is discussed in Section 4.5.

The resulting methods are implemented as part of the Fortran package `libRQZ` which is made publicly available at `numa.cs.kuleuven.be/software/rqz`. Section 4.6 illustrates the performance of `libRQZ` with some numerical experiments. We conclude the chapter in Section 4.7.

4.2 Block Hessenberg pencils

In the first part of this section we define block Hessenberg matrices and pencils and study their characteristics. The second part of this section uses the rational Krylov theory from Section 3.6 to prove that rational Krylov spaces generated from block Hessenberg pencils have a block structure. The third and last part of this section describes two relevant operations on a block Hessenberg pencil.

4.2.1 Definitions and elementary results

We first define a block upper triangular matrix and the notation we will use for it.

Definition 4.2.1. A matrix $R \in \mathbb{F}^{n\times n}$ is called a block upper triangular matrix with block partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, $s_1 + \ldots + s_m = n$, if it admits the form,

$$\begin{bmatrix} R_{11} & R_{12} & \ldots & R_{1m} \\ & R_{22} & \ldots & R_{2m} \\ & & \ddots & \vdots \\ & & & R_{mm} \end{bmatrix}, \tag{4.1}$$

with block R_{jk} of size $s_j \times s_k$ for $1 \leq j \leq k \leq m$. The vector $\boldsymbol{s}$ defines the sizes of the blocks and is called the *partition vector*. For the sake of clarity, the block partition can be explicitly denoted as $R_{(s_1,\ldots,s_m)}$ or $R_{\boldsymbol{s}}$.

A special case of a block upper triangular matrix is a block diagonal matrix $D_{\boldsymbol{s}}$ in which all off-diagonal blocks are zero:

$$D_{\boldsymbol{s}} = \begin{bmatrix} D_{11} & & & \\ & D_{22} & & \\ & & \ddots & \\ & & & D_{mm} \end{bmatrix}. \tag{4.2}$$

We sometimes use the notation $D_{\boldsymbol{s}} = \operatorname{diag}(D_{11}, D_{22}, \ldots, D_{mm})$ for block diagonal matrices. Further note that if $R_{\boldsymbol{s}}$ is a nonsingular block upper triangular matrix, $\hat{R}_{\boldsymbol{s}} = R_{\boldsymbol{s}}^{-1}$ is also a block upper triangular matrix with an identical block partition $\boldsymbol{s}$.

Next we define a *block upper Hessenberg* matrix based on the definition of a block upper triangular matrix.

Definition 4.2.2. A matrix $H \in \mathbb{F}^{n\times n}$ is called a block upper Hessenberg matrix with block partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, $s_1 + \ldots + s_m = n-1$, if it admits the form,

$$H_{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{h}_{11}^T & h_{12} \\ H_{21} & \boldsymbol{h}_{22} \end{bmatrix}, \tag{4.3}$$

with H_{21} an $(n-1)\times(n-1)$ block upper triangular matrix with block partition $\boldsymbol{s}$, $\boldsymbol{h}_{11}$ and $\boldsymbol{h}_{22}$ vectors of length $n-1$ and h_{12} a scalar.

Definition 4.2.2 is now extended in an evident manner for matrix pencils. In addition to that, we also introduce the notion of the *pole pencil* and the *pole tuple* of a block Hessenberg pencil.

Definition 4.2.3. The $n\times n$ matrix pencil (A, B) is called a block upper Hessenberg pencil with block partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ if both A and B are block upper Hessenberg matrices with a coinciding block partition,

$$A = \begin{bmatrix} \boldsymbol{a}_{11}^T & a_{12} \\ A_{21} & \boldsymbol{a}_{22} \end{bmatrix}, \quad B = \begin{bmatrix} \boldsymbol{b}_{11}^T & b_{12} \\ B_{21} & \boldsymbol{b}_{22} \end{bmatrix}, \tag{4.4}$$

and if A_{21}, B_{21} $(n-1)\times(n-1)$ are both block upper triangular matrices having block partition $\boldsymbol{s} = (s_1, \ldots, s_m)$. The block upper triangular pencil (A_{21}, B_{21}) in (4.4) is called the *pole pencil* of (A, B). If the pole pencil is regular, the poles $\Xi(A, B)$ are defined as the eigenvalues of the pole pencil, $\Lambda(A_{21}, B_{21})$. Since (A_{21}, B_{21}) admits the partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, the pole tuple,

$$\Xi(A, B) = \Lambda(A_{21}, B_{21}) = (\Xi^1, \ldots, \Xi^m) = (\{\xi_1^1, \ldots, \xi_{s_1}^1\}, \ldots, \{\xi_1^m, \ldots, \xi_{s_m}^m\}), \tag{4.5}$$

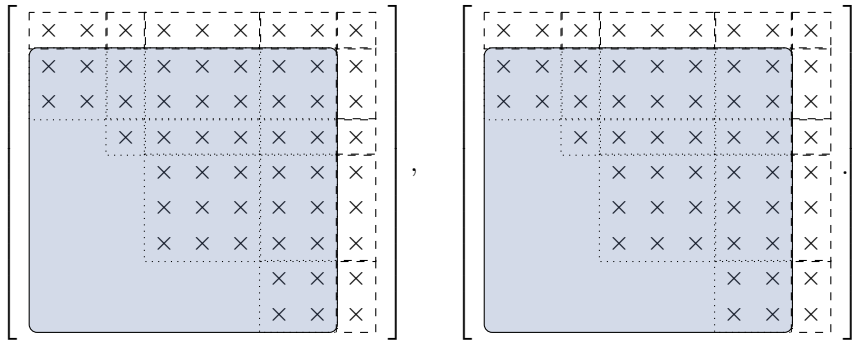
admits the same partition. This imposes no specific ordering of the poles within a block but the mutual blocks are ordered.

The previous definitions are illustrated in more detail in the next example.

Example 4.2.4. The $n\times n$ matrices A, B form a block Hessenberg pencil with partition vector $\boldsymbol{s} = (s_1, \ldots, s_m)$, if they can be partitioned as:

$$\begin{bmatrix} \boldsymbol{a}_{1,1}^T & \boldsymbol{a}_{1,2}^T & \ldots & \boldsymbol{a}_{1,m}^T & a_{1,m+1} \\ A_{2,1} & A_{2,2} & \ldots & A_{2,m} & \boldsymbol{a}_{2,m+1} \\ & A_{3,2} & \ldots & A_{3,m} & \boldsymbol{a}_{3,m+1} \\ & & \ddots & \vdots & \vdots \\ & & & A_{m+1,m} & \boldsymbol{a}_{m+1,m+1} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{b}_{1,1}^T & \boldsymbol{b}_{1,2}^T & \ldots & \boldsymbol{b}_{1,m}^T & b_{1,m+1} \\ B_{2,1} & B_{2,2} & \ldots & B_{2,m} & \boldsymbol{b}_{2,m+1} \\ & B_{3,2} & \ldots & B_{3,m} & \boldsymbol{b}_{3,m+1} \\ & & \ddots & \vdots & \vdots \\ & & & B_{m+1,m} & \boldsymbol{b}_{m+1,m+1} \end{bmatrix}, \tag{4.6}$$

with all subdiagonal blocks $A_{j+1,j}$, $B_{j+1,j}$ of size $s_j \times s_j$ (square) and $s_1 + \ldots + s_m = n-1$. As a specific example, the pencil (A, B) is a 9×9 block upper Hessenberg pencil with block partition $\boldsymbol{s} = (2, 1, 3, 2)$ if it has the form:



The shaded part of the matrices is the pole pencil which is clearly in block upper triangular form with partition $\boldsymbol{s} = (2, 1, 3, 2)$. The pole tuple is in this case given by,

$$\Xi(A, B) = (\Xi^1 = \{\xi_1^1, \xi_2^1\},\ \Xi^2 = \{\xi_1^2\},\ \Xi^3 = \{\xi_1^3, \xi_2^3, \xi_3^3\},\ \Xi^4 = \{\xi_1^4, \xi_2^4\}).$$

We remark that a given block Hessenberg pencil can admit more than one partition. If (A, B) is a block Hessenberg pencil with partition $\boldsymbol{s} = (s_1, \ldots, s_k, s_{k+1}, \ldots, s_m)$, it also admits the partition $\hat{\boldsymbol{s}} = (s_1, \ldots, s_k + s_{k+1}, \ldots, s_m)$. Consecutive blocks can be grouped together. Similarly, every $n\times n$ pencil (A, B) can be considered a block Hessenberg pencil with the trivial partition $(n-1)$. We say that $\boldsymbol{s}^{\max} = (s_1, \ldots, s_m)$ is the *maximal partition* of a block Hessenberg pencil if none of its blocks can be split into smaller blocks. For example, a Hessenberg pencil has maximal partition $\boldsymbol{s}^{\max} = (1, 1, \ldots, 1)$, but admits any other partition. The *cumulative partition* vector $\boldsymbol{s}^c$ of a block Hessenberg pencil with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, is defined as:

$$\boldsymbol{s}^c = (s_1, s_1 + s_2, \ldots, \sum_{i=1}^{m} s_i = n-1). \tag{4.7}$$

The last definition we generalize from the Hessenberg pencils of the RQZ method to the block Hessenberg pencils for the multishift, multipole RQZ method is the concept of *properness* or *irreducibility*. Properness of the pencil guarantees that there are no obvious options for deflations that split the problem into smaller, independent problems.

Definition 4.2.5. An $n \times n$ block upper Hessenberg pair (A, B) with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ is said to be proper (or irreducible) if:

I. Its pole pencil is regular,

II. The first block column of $A - \lambda B$ of size $(s_1+1) \times s_1$,

$$\begin{bmatrix} \boldsymbol{a}_{1,1}^T - \lambda \boldsymbol{b}_{1,1}^T \\ A_{2,1} - \lambda B_{2,1} \end{bmatrix},$$

does not have a zero according to Definition 2.1.5,

III. The last block row of (A, B) of size $s_m \times (s_m+1)$,

$$\begin{bmatrix} A_{m+1,m} - \lambda B_{m+1,m} & \boldsymbol{a}_{m+1,m+1} - \lambda \boldsymbol{b}_{m+1,m+1} \end{bmatrix},$$

does not have a zero according to Definition 2.1.5.

We remark that condition III is the same as condition II for the pertransposed pencil. Furthermore observe that if (A, B) is a Hessenberg pair then the conditions of Definition 4.2.5 reduce to the same conditions as Definition 3.2.1. Conditions II also ensures that property IV of Lemma 3.2.2 is satisfied within the first block column. We illustrate the notion of (im)properness of a block Hessenberg pencil on a small example to clarify Definition 4.2.5.

Example 4.2.6. Consider the 4×4 real-valued block Hessenberg pencil (A, B) with maximal partition $(2, 1)$ given by:

$$\begin{bmatrix} -0.3 & 0.075 & 0.5 & 0.25 \\ 0.395 & 0.52 & -0.35 & 2 \\ -0.14 & 0.86 & 1.35 & -0.8 \\ & & 1 & 0.85 \end{bmatrix}, \quad \begin{bmatrix} -0.15 & -0.6 & 0.15 & -1.5 \\ 0.16 & 0.94 & -5 & 1.35 \\ -0.12 & -0.08 & -2.4 & -1 \\ & & 0.2 & 1.8 \end{bmatrix}. \tag{4.8}$$

Condition I of Definition 4.2.5 is satisfied, the pole pencil is regular and the pole tuple of (A, B) is given by:

$$\Xi = (\{1.5 + i\sqrt{15/8}, 1.5 - i\sqrt{15/8}\}, 5). \tag{4.9}$$

The 2×2 block thus contains a pair of complex conjugate poles. Condition III of Definition 4.2.5 is also satisfied. For the last block row of (A, B), we clearly have

that $\mathcal{R}(\begin{bmatrix}1 & 0.85\end{bmatrix}) \neq \mathcal{R}(\begin{bmatrix}0.2 & 1.8\end{bmatrix})$. Notice that this implies that we cannot simultaneously create a zero in position $(4,3)$ of both A and B by rotating the last two columns. The block Hessenberg pencil (4.8) is however *improper* since Condition II of Definition 4.2.5 is violated. We have that $\mathcal{R}(\boldsymbol{a}_1) \neq \mathcal{R}(\boldsymbol{b}_1)$, but $\mathcal{R}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \mathcal{R}(\boldsymbol{b}_1, \boldsymbol{b}_2)$. If we compute an orthonormal basis Q_1 of $\mathcal{R}(\boldsymbol{a}_1, \boldsymbol{a}_2)$ and extend this upto an orthonormal matrix $Q = \begin{bmatrix}Q_1 & \boldsymbol{q}_2\end{bmatrix}$, then $(\hat{A}, \hat{B}) = Q^T(A, B)$ has zero elements in positions $(3,1)$ and $(3,2)$. This deflates the complex conjugate pair of poles in (4.9) as eigenvalues of the pencil.

The next lemma shows that any proper block Hessenberg pair can be transformed to a proper Hessenberg pair with the same poles.

Lemma 4.2.7. *Given an $n \times n$ proper block Hessenberg pair (A, B) with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ and accordingly partitioned poles $\Xi(A, B)$. Then there exist $n \times n$ unitary block diagonal matrices Q, Z,*

$$Q = \mathit{diag}(1, Q_1, \ldots, Q_m) \qquad \text{and} \qquad Z = \mathit{diag}(Z_1, \ldots, Z_m, 1), \tag{4.10}$$

with Q_j, Z_j unitary matrices of size $s_j \times s_j$, such that $(\hat{A}, \hat{B}) = Q^*(A, B)Z$ is a proper Hessenberg pair according to Definition 3.2.1 with poles $\Xi = (\pi_1(\Xi^1), \ldots, \pi_m(\Xi^m))$. Here, $\pi_j(\Xi^j)$ is a permutation of $\xi_1^j, \ldots, \xi_{s_j}^j$.

Proof. Since (A, B) is a proper block Hessenberg pencil, the pole pencil is regular and any Schur decomposition of it reduces the block Hessenberg pair to a Hessenberg pair with the same pole tuple as the block Hessenberg pencil. The order of the poles in the Hessenberg pair is, however, dependent on the Schur decomposition.

Moreover, since the pole pencil is a block upper triangular pencil with m blocks, m independent Schur decompositions can be combined as in (4.10). The pole tuple of the Hessenberg pencil is in this case clearly as described: the poles of the different blocks remain mutually ordered, but within a block any order, or permutation π_j, of the poles is permissible. It remains to verify that conditions II and III of Definition 4.2.5 are preserved under this transformation. Denote $\hat{Q} = \operatorname{diag}(Q_1, \ldots, Q_m)$ and $\hat{Z} = \operatorname{diag}(Z_1, \ldots, Z_m)$, with Q_j, Z_j as in (4.10). Then,

$$\hat{A} = Q^*AZ = \operatorname{diag}(1, \hat{Q}^*) \begin{bmatrix} \boldsymbol{a}_{11}^T & a_{12} \\ A_{21} & \boldsymbol{a}_{22} \end{bmatrix} \operatorname{diag}(\hat{Z}, 1) = \begin{bmatrix} \boldsymbol{a}_{11}^T \hat{Z} & a_{12} \\ \hat{Q}^* A_{21} \hat{Z} & \hat{Q}^* \boldsymbol{a}_{22} \end{bmatrix},$$

$$\hat{B} = Q^*BZ = \operatorname{diag}(1, \hat{Q}^*) \begin{bmatrix} \boldsymbol{b}_{11}^T & b_{12} \\ B_{21} & \boldsymbol{b}_{22} \end{bmatrix} \operatorname{diag}(\hat{Z}, 1) = \begin{bmatrix} \boldsymbol{b}_{11}^T \hat{Z} & b_{12} \\ \hat{Q}^* B_{21} \hat{Z} & \hat{Q}^* \boldsymbol{b}_{22} \end{bmatrix}.$$

The first block column of $(\hat{A}, \hat{B})$ is equal to,

$$\left(\begin{bmatrix}\hat{\boldsymbol{a}}_{1,1}^T \\ \hat{A}_{2,1}\end{bmatrix}, \begin{bmatrix}\hat{\boldsymbol{b}}_{1,1}^T \\ \hat{B}_{2,1}\end{bmatrix}\right) = \begin{bmatrix}1 & \\ & Q_1^*\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{a}_{1,1}^T \\ A_{2,1}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_{1,1}^T \\ B_{2,1}\end{bmatrix}\right) Z_1.$$

The left and right multiplication of the first block column of (A, B) with unitary matrices clearly preserves condition II of Definition 4.2.5. Also condition III is preserved under the equivalence transformation of (4.10). This directly implies that the resulting Hessenberg pair is also proper according to Definition 3.2.1. □

We remark that since a real-valued block Hessenberg pencil can have complex conjugate pairs of poles, its proper Hessenberg form of Lemma 4.2.7 will be complex-valued.

4.2.2 Rational Krylov and block Hessenberg pencils

In this section we study the structure of *rational Krylov subspaces* generated by proper block Hessenberg matrices. These results are useful for the analysis of the *pole introduction* operation introduced in Section 4.2.3 and to study *uniqueness* of a multishift, multipole RQZ step in Section 4.3.

We use the same notational conventions as in Chapter 3 and rely on the elementary rational matrices $M(\varrho, \xi)$ and $N(\varrho, \xi)$ of (3.7) to simplify notation. We will often use the properties of Lemma 3.6.3. One additional useful property is the *merging* of two rational matrices into one:

$$\begin{aligned} M(\varrho, \xi_1)M(\xi_1, \xi_2) &= M(\varrho, \xi_2), \\ N(\varrho, \xi_1)N(\xi_1, \xi_2) &= N(\varrho, \xi_2), \end{aligned} \tag{4.11}$$

which is possible when there is a pole equal to a shift.

The following theorem is a block generalization of Theorem 3.6.7 and shows that the rational Krylov subspaces $\mathcal{K}^{\text{rat}}$ and $\mathcal{L}^{\text{rat}}$ have a specific structure if they are generated from a proper block Hessenberg pair.

Theorem 4.2.8. *Given an $n \times n$ proper block Hessenberg pair (A, B) with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, cumulative partition $\boldsymbol{s^c}$, poles $\Xi = (\Xi^1, \ldots, \Xi^m)$ with $\Xi^i = \{\xi_1^i, \ldots \xi_{s_i}^i\}$ that are all different from the eigenvalues. Then for $j = 0, 1, \ldots, m$,*

$$\mathcal{K}_{s_j^c+1}^{rat}(A, B, \boldsymbol{e}_1, (\Xi^1, \ldots, \Xi^j)) = \mathcal{E}_{s_j^c+1}, \tag{4.12}$$

with $s_0^c \equiv 0$. While for $j = 1, \ldots, m$,

$$\mathcal{L}_{s_j^c}^{rat}(A, B, \boldsymbol{z}_1, (\breve{\Xi}^1, \Xi^2, \ldots, \Xi^j)) = \mathcal{E}_{s_j^c}, \tag{4.13}$$

with $\breve{\Xi}^1 = \{\xi_1^1, \ldots, \xi_{s_1-1}^1\}$*, and* $\boldsymbol{z}_1$ *the right eigenvector of the pole pencil corresponding with pole* $\xi_{s_1}^1$*. Here* $\xi_{s_1}^1$ *can be any of the poles in* Ξ^1*.*

Proof. We rely on the transformation $(\hat{A}, \hat{B}) = Q^*(A, B)Z$ from proper block Hessenberg pencil (A, B) to proper Hessenberg pencil $(\hat{A}, \hat{B})$ as defined in Lemma 4.2.7. Denote with $\hat{\Xi} = (\xi_1, \ldots, \xi_{n-1})$ the pole tuple of the proper Hessenberg pair $(\hat{A}, \hat{B})$ after renumbering. Note that, by construction, in (4.10), $\boldsymbol{q}_1 = \boldsymbol{e}_1$ and denote $\hat{M}(\varrho, \xi) = Q^* M(\varrho, \xi) Q$ as the elementary rational matrix (3.7) in terms of $(\hat{A}, \hat{B})$. Further we apply Theorem 3.6.7 to $(\hat{A}, \hat{B})$ such that for k from 1 to n,

$$\mathcal{E}_k = \mathcal{K}_k^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_{k-1})) = \prod_{i=1}^{k-1} \hat{M}(\hat{\varrho}, \xi_i) \cdot \mathcal{K}_k(\hat{M}(\check{\varrho}, \hat{\varrho}), \boldsymbol{e}_1)$$

$$= Q^* \prod_{i=1}^{k-1} M(\hat{\varrho}, \xi_i) \cdot \mathcal{K}_k(M(\check{\varrho}, \hat{\varrho}), \boldsymbol{q}_1) = Q^* \mathcal{K}_k^{\mathrm{rat}}(A, B, \boldsymbol{e}_1, (\xi_1, \ldots, \xi_{k-1})).$$

Multiplying both sides of this equation with Q and using that (4.10) implies that $Q\mathcal{E}_k = \mathcal{E}_k$ for $k \in \{1, s_1^c + 1, \ldots, s_m^c + 1 = n\}$, proves the first part of the theorem. The second part of the theorem can be proven in an analogous manner. Denote $\hat{N}(\varrho, \xi) = Z^* N(\varrho, \xi) Z$ as the second elementary rational matrix (3.7) in terms of $(\hat{A}, \hat{B})$ and apply again Theorem 3.6.7 to $(\hat{A}, \hat{B})$ such that for k from 1 to $n-1$,

$$\mathcal{E}_k = \mathcal{L}_k^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, (\xi_2, \ldots, \xi_k)) = \prod_{i=2}^{k} \hat{N}(\hat{\varrho}, \xi_i) \cdot \mathcal{K}_k(\hat{N}(\check{\varrho}, \hat{\varrho}), \boldsymbol{e}_1)$$

$$= Z^* \prod_{i=2}^{k} N(\hat{\varrho}, \xi_i) \cdot \mathcal{K}_k(N(\check{\varrho}, \hat{\varrho}), \boldsymbol{z}_1) = Z^* \mathcal{L}_k^{\mathrm{rat}}(A, B, \boldsymbol{z}_1, (\xi_2, \ldots, \xi_k)).$$

Now multiply both sides with Z and again use (4.10) to show that $Z\mathcal{E}_k = \mathcal{E}_k$ for $k \in \{s_1^c, \ldots, s_m^c, n\}$ and the second part of the theorem is proven. Recall from Lemma 4.2.7 that $\boldsymbol{z}_1$ is the right eigenvector of the pole pencil related to ξ_1 and that ξ_1 can be any of the poles of Ξ^1 since for any ξ_j^1 there exists a block Schur decomposition (4.10) that places ξ_j^1 as the first pole in the Hessenberg pencil $(\hat{A}, \hat{B})$. □

4.2.3 Manipulating poles of block Hessenberg pencils

Throughout this section, the pencil (A, B) is assumed to be an $n \times n$ proper block Hessenberg pencil with maximal partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ and pole tuple

$\Xi = (\Xi^1, \ldots, \Xi^m)$, where $\Xi^j = \{\xi_1^j, \ldots, \xi_{s_j}^j\}$. All poles are assumed different from the eigenvalues.

We review two different operations to change the pole tuple Ξ. The first operation changes the first or last ℓ poles of the pencil, the second operation swaps two adjacent pole blocks Ξ^i and Ξ^{i+1}.

Changing poles at the boundary The first $\ell = s_1 + \ldots + s_i = s_i^c$ poles in the first i pole blocks $\Xi^1, \ldots, \Xi^i$ can be changed to ℓ new poles $\mathrm{P} = \{\varrho_1, \ldots, \varrho_\ell\}$. We assume P distinct from the original poles. For this purpose consider the vector,

$$\boldsymbol{x} = \gamma \prod_{j=1}^{\ell} M(\varrho_j, \xi_j)\, \boldsymbol{e}_1, \tag{4.14}$$

with $\xi_1, \ldots, \xi_\ell$ the poles of $\Xi^1, \ldots, \Xi^i$. The following procedure can be used to compute $\boldsymbol{x}$,

$$\begin{aligned} &\boldsymbol{x} \leftarrow \boldsymbol{e}_1 \\ &\text{for } j = 1, \ldots, \ell \\ &\quad \boldsymbol{x} \leftarrow \gamma_j\, M(\varrho_j, \xi_j)\boldsymbol{x} \end{aligned} \tag{4.15}$$

The scalars γ_j can be chosen as some suitable scaling factors. Now, just like previously, compute a unitary matrix Q such that,

$$Q^* \boldsymbol{x} = \alpha \boldsymbol{e}_1. \tag{4.16}$$

We claim that the new poles P are introduced in the block Hessenberg pair by updating $(\hat{A}, \hat{B}) = Q^*(A, B)$. Specifically, $(\hat{A}, \hat{B})$ is a block Hessenberg pair with maximal partition $\hat{\boldsymbol{s}} = (\ell, s_{i+1}, \ldots, s_m)$ and poles $\hat{\Xi} = (\mathrm{P}, \Xi^{i+1}, \ldots, \Xi^m)$.

From (3.20) and Theorem 4.2.8 we have that,

$$\boldsymbol{x} \in \mathcal{K}_{\ell+1}^{\mathrm{rat}}(A, B, \boldsymbol{e}_1, \Xi) = \mathcal{E}_{\ell+1}. \tag{4.17}$$

This implies that Q in (4.16) is of the form $\mathrm{diag}(Q_{\ell+1}, I)$, with $Q_{\ell+1}$ an $(\ell{+}1){\times}(\ell{+}1)$ unitary matrix. It follows that the first block $\hat{\Xi}^1$ in $(\hat{A}, \hat{B})$ is

indeed of size ℓ. Furthermore, for $j = 0, 1, \ldots, m - i + 1$,

$$\begin{aligned}
\mathcal{K}^{\mathrm{rat}}_{\hat{s}^c_j+1}(\hat{A}, \hat{B}, \boldsymbol{e}_1, (\mathrm{P}, \Xi^{i+1}, \ldots, \Xi^m)) &= \prod_{k=1}^{\hat{s}^c_j} \hat{M}(\hat{\varrho}, \hat{\xi}_k) \cdot \mathcal{K}_{\hat{s}^c_j+1}(\hat{M}(\check{\varrho}, \hat{\varrho}), \boldsymbol{e}_1) \\
&= Q^* M(\hat{\varrho}, \varrho_1) \ldots M(\hat{\varrho}, \varrho_\ell) M(\hat{\varrho}, \xi_{\ell+1}) \ldots M(\hat{\varrho}, \xi_{\hat{s}^c_j}) \cdot \mathcal{K}_{\hat{s}^c_j+1}(M(\check{\varrho}, \hat{\varrho}), \boldsymbol{q}_1) \\
&= Q^* M(\hat{\varrho}, \varrho_1) \ldots M(\hat{\varrho}, \varrho_\ell) M(\hat{\varrho}, \xi_{\ell+1}) \ldots M(\hat{\varrho}, \xi_{\hat{s}^c_j}) \cdot \mathcal{K}_{\hat{s}^c_j+1}(M(\check{\varrho}, \hat{\varrho}), \prod_{k=1}^{\ell} M(\varrho_k, \xi_k) \boldsymbol{e}_1) \\
&= Q^* M(\hat{\varrho}, \xi_1) \ldots M(\hat{\varrho}, \xi_\ell) M(\hat{\varrho}, \xi_{\ell+1}) \ldots M(\hat{\varrho}, \xi_{\hat{s}^c_j}) \cdot \mathcal{K}_{\hat{s}^c_j+1}(M(\check{\varrho}, \hat{\varrho}), \boldsymbol{e}_1) \\
&= Q^* \mathcal{K}^{\mathrm{rat}}_{\hat{s}^c_j+1}(A, B, \boldsymbol{e}_1, (\Xi^1, \ldots, \Xi^i, \Xi^{i+1}, \ldots, \Xi^m)) \\
&= Q^* \mathcal{E}_{\hat{s}^c_j+1} = \mathcal{E}_{\hat{s}^c_j+1}.
\end{aligned}$$

In the first equality we used (3.20), we applied $\hat{M} = Q^* M Q$ in the second equality, and combined (4.14) with (4.16) to get $\boldsymbol{q}_1 = \prod_{k=1}^{\ell} M(\varrho_k, \xi_k) \boldsymbol{e}_1$ in the third equality. The fourth equality uses the commutativity of the M matrices and the property of (4.11). This results in the rational Krylov subspace of the original pencil with the original poles in the fifth equality and by Theorem 4.2.8 we know that this is equal to $\mathcal{E}_{\hat{s}^c_j+1}$. Finally, since Q has a block diagonal structure, it does not affect the $\mathcal{E}_{\hat{s}^c_j+1}$ for the given sizes. It is clear that $(\hat{A}, \hat{B})$ is a proper block Hessenberg pencil with partition $\hat{\boldsymbol{s}} = (\ell, s_{i+1}, \ldots, s_m)$ by construction. The last poles are unchanged by the block diagonal structure of Q and the first ℓ poles are changed to P which follows from the uniqueness of block Hessenberg pencils, see Theorem 2.3.4.

We remark that in order to compute the vector $\boldsymbol{x}$ in (4.15), ℓ shifted linear systems need to be solved as $M(\varrho_i, \xi_i) = (\nu_i A - \mu_i B)(\beta_i A - \alpha_i B)^{-1}$. These linear systems are essentially of size ℓ because $(\beta_\ell A - \alpha_\ell B)^{-1}$ is a block upper triangular matrix with a leading block of size $\ell \times \ell$. This limits the computational cost of computing $\boldsymbol{x}$ to $O(\ell^4)$, which is small as long as $\ell \ll n$. It also follows that the vector $\boldsymbol{x}$ can be computed even when poles in $\Xi^1, \ldots, \Xi^i$ are equal to eigenvalues of the pencil. Properness ensures that the leading $\ell \times \ell$ block is nonsingular.

The last ℓ poles in the last i blocks $\Xi^{m-i+1}, \ldots, \Xi^m$ of (A, B) can be changed to $\mathrm{P} = \{\varrho_1, \ldots, \varrho_\ell\}$ in a similar fashion. We compute first the row vector,

$$\boldsymbol{x}^T = \gamma \boldsymbol{e}_n^T \prod_{j=m-\ell+1}^{m} N(\varrho_j, \xi_j), \tag{4.18}$$

and then a unitary matrix $Z = \mathrm{diag}(I, Z_{\ell+1})$ such that $\boldsymbol{x}^T Z = \alpha \boldsymbol{e}_n^T$. The pencil $(\hat{A}, \hat{B}) = (A, B)Z$ then becomes a block Hessenberg pencil with pole tuple (Ξ^1, …, Ξ^{m-i}, P).

We remark that if (A, B) is a real-valued pencil and the poles and shifts considered in (4.14) and (4.18) are both closed under complex conjugation, then the vectors $\boldsymbol{x}$ and $\boldsymbol{x}^T$ and consequently the matrices Q and Z are also real-valued. This follows from the commutativity property in combination with the property that $M(\bar{\varrho}, \bar{\xi}) = \overline{M(\varrho, \xi)}$ for real-valued pencils. We have,

$$\overline{M(\varrho,\xi)M(\bar{\varrho},\bar{\xi})} = \overline{M(\bar{\varrho},\bar{\xi})M(\varrho,\xi)} = M(\varrho,\xi)M(\bar{\varrho},\bar{\xi}) \tag{4.19}$$

so $M(\varrho, \xi)M(\bar{\varrho}, \bar{\xi})$ is a real-valued matrix if A and B are real-valued.

Swapping adjacent pole blocks A second operation to change the pole tuple of a block Hessenberg pencil is swapping two consecutive blocks in the pole pencil. Swapping block i with block $i{+}1$ requires the computation of a unitary equivalence essentially of size $(s_i{+}s_{i+1}){\times}(s_i{+}s_{i+1})$ that updates the pencil $(\hat{A}, \hat{B}) = Q^*(A, B)Z$ in such a way that the new pole tuple and partition vector are given by,

$$\hat{\Xi} = (\Xi^1, \ldots, \Xi^{i-1}, \Xi^{i+1}, \Xi^i, \Xi^{i+2}, \ldots, \Xi^m),$$

$$\hat{\boldsymbol{s}} = (s_1, \ldots, s_{i-1}, s_{i+1}, s_i, s_{i+2}, \ldots, s_m).$$

This problem is equivalent to reordering eigenvalues in the generalized Schur form. Two different approaches to solve this problem have been proposed in the literature. The first approach, studied by Kågström [57, 60], requires the solution of a coupled Sylvester equation. This method is applicable for general blocksizes s_i, s_{i+1}. The second approach, studied by Van Dooren [122], is a direct method that relies on the computation of a right eigenvector of a pole in block $i{+}1$. This method has been studied for swapping a block of dimension $1{\times}1$ or $2{\times}2$ with a block of dimension $1{\times}1$, or vice versa. We will discuss the problem of computing a swapping transformation in more detail in Section 4.4.

4.2.4 Multishift, multipole RQZ step

Combining the operations from the previous subsection, we propose the following three step procedure as the generic multishift, multipole RQZ step.

I. Starting from a proper block Hessenberg pencil with pole tuple $\Xi = (\Xi^1, \ldots, \Xi^m)$ and partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, select or compute $\ell =$

$s_1 + \ldots + s_i = s_i^c$ shifts P. Introduce the shifts in the block Hessenberg pencil by computing the vector $\boldsymbol{x}$ via (4.15) and the orthonormal matrix Q via (4.16) and updating the pencil accordingly. The pencil now has pole tuple $\Xi = (\mathrm{P}, \Xi^{i+1}, \ldots, \Xi^m)$ and partition vector $\boldsymbol{s} = (\ell, s_{i+1}, \ldots, s_m)$.

II. Repeatedly use the swapping procedure to construct a unitary equivalence that moves the shifts P to the last pole block. This changes the pole tuple to $\Xi = (\Xi^{i+1}, \ldots, \Xi^m, \mathrm{P})$ and the partition vector to $\boldsymbol{s} = (s_{i+1}, \ldots, s_m, \ell)$.

III. Compute or select ℓ new poles Ξ^{m+1} and introduce them at the end of the pencil to change the pole tuple to $\Xi = (\Xi^{i+1}, \ldots, \Xi^m, \Xi^{m+1})$.

These three steps constitute a single multishift, multipole RQZ sweep. After every sweep, the properness of the pencil is checked and the problem is split into independent subproblems wherever possible.

The multishift QZ method is a special case of this algorithm where the pencil initially has pole tuple $(\infty, \ldots, \infty)$ and partition $(1, \ldots, 1)$ and where this form is always restored in step III of the algorithm. The single shift RQZ method is also a special case of this algorithm.

In Sections 4.4 and 4.5 we address a couple of numerical challenges that make the multishift, multipole RQZ step stable and efficient in finite precision arithmetic. First, Section 4.3 provides further theoretical foundation for the implicit approach.

4.3 Uniqueness and convergence

In this section we motivate the implicit approach used in the multishift, multipole RQZ step in the form of an implicit Q theorem for block Hessenberg pencils. We also discuss the subspace iteration that is implicitly performed during the multishift, multipole RQZ step.

The following lemma extends the essential uniqueness of the QR factorization from Lemma 2.3.3 to a form of essential uniqueness in the factorization of a matrix as a product of a unitary matrix and a block upper triangular matrix.

Lemma 4.3.1. *Given a nonsingular $n \times n$ matrix A and consider $A = \hat{Q}\hat{R}_{\boldsymbol{s}}$, $A = \check{Q}\check{R}_{\boldsymbol{s}}$ two block QR factorizations where $\hat{Q}, \check{Q}$ are unitary matrices and $\hat{R}_{\boldsymbol{s}}, \check{R}_{\boldsymbol{s}}$ are block upper triangular matrices with the same partition $\boldsymbol{s} = (s_1, \ldots, s_m)$. Then $\hat{Q} = \check{Q}D_{\boldsymbol{s}}$ with $D_{\boldsymbol{s}}$ a unitary block diagonal matrix with an identical partition $\boldsymbol{s}$ as $\hat{R}_{\boldsymbol{s}}$ and $\check{R}_{\boldsymbol{s}}$.*

Proof. From $\hat{Q}\hat{R}_{\boldsymbol{s}} = \check{Q}\check{R}_{\boldsymbol{s}}$ it follows that, $\check{Q}^*\hat{Q} = \check{R}_{\boldsymbol{s}}\hat{R}_{\boldsymbol{s}}^{-1} = \tilde{R}_{\boldsymbol{s}} = D_{\boldsymbol{s}}$, with $\tilde{R}_{\boldsymbol{s}}$ a unitary block upper triangular matrix with partition $\boldsymbol{s}$. The only unitary block upper triangular matrices are block diagonal matrices $D_{\boldsymbol{s}}$. □

Before presenting the implicit Q theorem, we first give this direct corollary of Theorem 4.2.8 that considers the structure of rational Krylov matrices instead of the subspaces. This is the block generalization of Corollary 3.6.8.

Corollary 4.3.2. *Given an $n\times n$ proper block Hessenberg pair (A, B) with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ and poles $\Xi = (\Xi^1, \ldots, \Xi^m)$ that are different from the eigenvalues. Then for a tuple of shifts* P *different from the poles, $K_n^{rat}(A, B, \boldsymbol{e}_1, \Xi, \mathrm{P})$ is a full rank $n\times n$ block upper triangular matrix with partition $(1, s_1, s_2, \ldots, s_m)$. While, $L_{n-1}^{rat}(A, B, \boldsymbol{z}_1, (\breve{\Xi}^1, \Xi^2, \ldots, \Xi^m), \mathrm{P})$ is a full rank $n\times n-1$ block upper triangular matrix with partition $(s_1, s_2, \ldots, s_m)$. Here $\boldsymbol{z}_1$ and $\breve{\Xi}^1$ are chosen as described in Theorem 4.2.8.*

We are now ready to state the block implicit Q theorem.

Theorem 4.3.3. *Let (A, B) be a regular matrix pair and let $\hat{Q}, \check{Q}, \hat{Z}, \check{Z}$ be unitary matrices with $\hat{Q}\boldsymbol{e}_1 = \sigma\check{Q}\boldsymbol{e}_1$, $|\sigma| = 1$, such that,*

$$(\hat{A}, \hat{B}) = \hat{Q}^*(A, B)\hat{Z} \quad \textit{and} \quad (\check{A}, \check{B}) = \check{Q}^*(A, B)\check{Z},$$

are both proper block Hessenberg pairs with the same partition $(s_1, \ldots, s_m)$ and poles $\Xi = (\Xi^1, \ldots, \Xi^m)$ different from the eigenvalues. Then the pairs $(\hat{A}, \hat{B})$ and $(\check{A}, \check{B})$ are identical up to multiplication with two unitary block diagonal matrices,

$$\hat{A} = D_1^*\check{A}D_2 \quad \textit{and} \quad \hat{B} = D_1^*\check{B}D_2,$$

with D_1 having partition $(1, s_1, \ldots, s_m)$ and D_2 having partition $(s_1, \ldots, s_m, 1)$.

Proof. Corollary 4.3.2 states that $K_n^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, \Xi, \mathrm{P})$ and $K_n^{\mathrm{rat}}(\check{A}, \check{B}, \boldsymbol{e}_1, \Xi, \mathrm{P})$ are both block upper triangular matrices of full rank with block partition $(1, s_1, \ldots, s_m)$. We thus have,

$$\begin{aligned}
&\hat{Q}\, K_n^{\mathrm{rat}}(\hat{A}, \hat{B}, \boldsymbol{e}_1, \Xi, \mathrm{P}) \\
&= \hat{Q} \left[\boldsymbol{e}_1,\ \hat{M}(\varrho_1, \xi_1)\, \boldsymbol{e}_1,\ \ldots,\ \left(\prod_{i=1}^{n-1} \hat{M}(\varrho_i, \xi_i) \right)\, \boldsymbol{e}_1 \right] \\
&= \hat{Q} \left[\boldsymbol{e}_1,\ \hat{Q}^* M(\varrho_1, \xi_1)\hat{Q}\, \boldsymbol{e}_1,\ \ldots,\ \hat{Q}^* \left(\prod_{i=1}^{n-1} M(\varrho_i, \xi_i) \right) \hat{Q}\, \boldsymbol{e}_1 \right]
\end{aligned}$$

$$
\begin{aligned}
&= \left[\hat{\boldsymbol{q}}_1,\ M(\varrho_1,\xi_1)\,\hat{\boldsymbol{q}}_1,\ \ldots,\ \left(\prod_{i=1}^{n-1} M(\varrho_i,\xi_i)\right)\,\hat{\boldsymbol{q}}_1\right] \\
&= \sigma\left[\check{\boldsymbol{q}}_1,\ M(\varrho_1,\xi_1)\,\check{\boldsymbol{q}}_1,\ \ldots,\ \left(\prod_{i=1}^{n-1} M(\varrho_i,\xi_i)\right)\,\check{\boldsymbol{q}}_1\right] \\
&= \sigma\check{Q}\,K_n^{\mathrm{rat}}(\check{A},\check{B},\boldsymbol{e}_1,\Xi,\mathrm{P}).
\end{aligned}
$$

From Lemma 4.3.1 we have that this equality between two block QR factorizations implies that $\hat{Q} = \check{Q}D_{(1,s_1,\ldots,s_m)}$. For the relation between $\hat{Z}$ and $\check{Z}$, consider,

$$
(\hat{\hat{A}},\hat{\hat{B}}) = \hat{\hat{Q}}^*\,(\hat{A},\hat{B})\,\hat{\hat{Z}}, \quad \text{and,} \quad (\check{\check{A}},\check{\check{B}}) = \check{\check{Q}}^*\,(\check{A},\check{B})\,\check{\check{Z}},
$$

both reductions of the block Hessenberg pencils to a proper Hessenberg pencil as defined in Lemma 4.2.7 and assume, without loss of generality, that $\xi_{s_1}^1$ is the first pole in both $(\hat{\hat{A}},\hat{\hat{B}})$ and $(\check{\check{A}},\check{\check{B}})$. Thus $\hat{\hat{\boldsymbol{z}}}_1$ is the right eigenvector of the pole pencil of $(\hat{\hat{A}},\hat{\hat{B}})$ associated with the eigenvalue $\xi_{s_1}^1$ and the same holds for $\check{\check{\boldsymbol{z}}}_1$ and $(\check{\check{A}},\check{\check{B}})$. This implies,

$$
\hat{\hat{Q}}^*\,(\hat{A}-\xi_{s_1}^1\hat{B})\hat{\hat{\boldsymbol{z}}}_1 = \hat{\gamma}\boldsymbol{e}_1, \quad \text{and,} \quad \check{\check{Q}}^*\,(\check{A}-\xi_{s_1}^1\check{B})\check{\check{\boldsymbol{z}}}_1 = \check{\gamma}\boldsymbol{e}_1,
$$

by the proper Hessenberg structure of $(\hat{\hat{A}},\hat{\hat{B}})$ and $(\check{\check{A}},\check{\check{B}})$. Since by eq. (4.10), $\hat{\hat{Q}}\boldsymbol{e}_1 = \check{\check{Q}}\boldsymbol{e}_1 = \boldsymbol{e}_1$, we also have,

$$
(\hat{A}-\xi_{s_1}^1\hat{B})\hat{\hat{\boldsymbol{z}}}_1 = \hat{\gamma}\boldsymbol{e}_1, \quad \text{and,} \quad (\check{A}-\xi_{s_1}^1\check{B})\check{\check{\boldsymbol{z}}}_1 = \check{\gamma}\boldsymbol{e}_1,
$$

Thus,

$$
\hat{Q}^*(A-\xi_{s_1}^1 B)\hat{Z}\hat{\hat{\boldsymbol{z}}}_1 = \hat{\gamma}\boldsymbol{e}_1, \quad \text{and,} \quad \check{Q}^*(A-\xi_{s_1}^1 B)\check{Z}\check{\check{\boldsymbol{z}}}_1 = \check{\gamma}\boldsymbol{e}_1.
$$

Using, $\hat{Q} = \check{Q}D_{(1,s_1,\ldots,s_m)}$, $D_{(1,s_1,\ldots,s_m)}\boldsymbol{e}_1 = \sigma\boldsymbol{e}_1$, we get that,

$$
\begin{aligned}
\hat{Z}\hat{\hat{\boldsymbol{z}}}_1 &= \hat{\gamma}(A-\xi_{s_1}^1 B)^{-1}\check{Q}D_{(1,s_1,\ldots,s_m)}\boldsymbol{e}_1 = \sigma\hat{\gamma}(A-\xi_{s_1}^1 B)^{-1}\check{Q}\boldsymbol{e}_1 \\
\check{Z}\check{\check{\boldsymbol{z}}}_1 &= \check{\gamma}(A-\xi_{s_1}^1 B)^{-1}\check{Q}\boldsymbol{e}_1,
\end{aligned}
$$

from which we conclude that $\hat{Z}\hat{\hat{\boldsymbol{z}}}_1 = \tilde{\sigma}\check{Z}\check{\check{\boldsymbol{z}}}_1$ for some $\tilde{\sigma}$ with $|\tilde{\sigma}| = 1$. Now use this result in combination with Corollary 4.3.2,

$$
\hat{Z}\,L_{n-1}^{\mathrm{rat}}(\hat{A},\hat{B},\hat{\hat{\boldsymbol{z}}}_1,\Xi,\mathrm{P})
$$

$$
\begin{aligned}
&= \hat{Z} \begin{bmatrix} \hat{\hat{\boldsymbol{z}}}_1,\ \hat{N}(\varrho_1,\xi_1)\,\hat{\hat{\boldsymbol{z}}}_1,\ \ldots,\ \left(\prod_{i=2}^{n-1}\hat{N}(\varrho_i,\xi_i)\right)\,\hat{\hat{\boldsymbol{z}}}_1 \end{bmatrix} \\
&= \begin{bmatrix} \hat{Z}\hat{\hat{\boldsymbol{z}}}_1,\ N(\varrho_1,\xi_1)\,\hat{Z}\hat{\hat{\boldsymbol{z}}}_1,\ \ldots,\ \left(\prod_{i=2}^{n-1}N(\varrho_i,\xi_i)\right)\,\hat{Z}\hat{\hat{\boldsymbol{z}}}_1 \end{bmatrix} \\
&= \tilde{\sigma} \begin{bmatrix} \check{Z}\check{\check{\boldsymbol{z}}}_1,\ N(\varrho_1,\xi_1)\,\check{Z}\check{\check{\boldsymbol{z}}}_1,\ \ldots,\ \left(\prod_{i=2}^{n-1}N(\varrho_i,\xi_i)\right)\,\check{Z}\check{\check{\boldsymbol{z}}}_1 \end{bmatrix} \\
&= \tilde{\sigma}\,\check{Z}\,L_{n-1}^{\mathrm{rat}}(\check{A},\check{B},\check{\check{\boldsymbol{z}}}_1,\Xi,\mathrm{P}).
\end{aligned}
$$

Based on Lemma 4.3.1 we can now guarantee that the first $n-1$ columns of $\hat{Z}$ are equal to the first $n-1$ columns of $\check{Z}$ multiplied with some $n-1\times n-1$ unitary block diagonal matrix $D_{(s_1,\ldots,s_m)}$. Observe that this also determines $\hat{\boldsymbol{z}}_n = \check{\sigma}\check{\boldsymbol{z}}_n$, $|\check{\sigma}| = 1$. This concludes the proof. □

In Theorem 3.7.3 it is shown that an RQZ step with shift ϱ on a Hessenberg pencil with pole tuple $\Xi = (\xi_1,\ldots,\xi_{n-1})$ and new pole ξ_n performs nested subspace iteration accelerated by

$$
q_k^Q(z) = \frac{z-\varrho}{z-\xi_k}, \quad \text{and} \quad q_k^Z(z) = \frac{z-\varrho}{z-\xi_{k+1}}, \tag{4.20}
$$

for the kth column vector of respectively Q and Z. Based on Lemma 4.2.7, this can be extended to block Hessenberg pencils under the condition that the partition $\boldsymbol{s}$ prior to the multishift, multipole RQZ step is the same as the partition $\hat{\boldsymbol{s}}$ afterwards. We omit this generalization as the condition $\boldsymbol{s} = \hat{\boldsymbol{s}}$ limits the practical application of the theoretical result. Combining Theorem 3.7.3 with Lemma 4.2.7, it is clear, however, that in the multishift, multipole RQZ method shifts that have been swapped along the subdiagonal of the block Hessenberg pencil will lead to deflations at the end of the pencil, while poles that have been moved to the front of the pencil lead to convergence of eigenvalues at the beginning. This holds under the assumption that a good choice of poles and shifts is made.

4.4 Numerical considerations

In this section, we discuss numerical experiments related to the pole introduction and swapping operations and draw conclusions for the practical implementation of the multishift, multipole RQZ method.

4.4.1 Introducing pole blocks

In finite precision arithmetic, the introduction of a large amount of poles in a block Hessenberg pencil via the computation of the vectors as described in (4.14) and (4.18) becomes increasingly inaccurate already for small to medium blocksizes. This comes as no surprise. Kressner [67] studied the use of larger bulges in the QR method and made a connection between the introduction of the multishift block in the Hessenberg matrix and the pole placement problem in systems and control theory. It has been shown in control theory that placing many poles in a high dimensional system is intrinsically ill-conditioned [54].

To illustrate the increasing inaccuracy of the pole introduction we have performed a numerical experiment for which the results are summarized in Figure 4.1. We introduced pole blocks containing $\ell = 2, 4, 6, \ldots, 30$ randomly generated pairs of complex-conjugate shifts ϱ_i in a real-valued Hessenberg matrix, a real-valued Hessenberg pencil, and a real-valued block Hessenberg pencil with leading blocksize ℓ. The procedure based on (4.14) was used for this. All problems are of size n=100. The Hessenberg matrix is obtained from the Hessenberg reduction of a randomly generated matrix with normally distributed entries with mean 0 and variance 1. In this case the shift vector $\boldsymbol{x}$ is computed in the classical way according to (2.48) [135] which is compatible with (4.14). Then an orthonormal matrix Q is computed having $\boldsymbol{q}_1 = \boldsymbol{x}$. The shifts are introduced as $Q^T(A, I)$, which is a block Hessenberg pencil. The actual shifts $\hat{\varrho}_i$ are then computed as the eigenvalues of the leading subdiagonal block of $Q^T(A, I)$. The blue line in Figure 4.1 shows the median absolute error $|\varrho_i - \hat{\varrho}_i|$ over all shifts and 100 repetitions of the experiment. The green line in Figure 4.1 shows the results of the same experiment but now starting from a Hessenberg pencil (A, B) where each individual matrix is generated as before. Now a procedure based (4.14) is used to compute $\boldsymbol{x}$. Finally, the orange line shows the results when (A, B) is initially a block Hessenberg pencil with leading blocksize $s_1 = \ell$ and all other blocks of size 1.

We remark that, in all three experiments, we obtain a block Hessenberg pencil with partition $(\ell, 1, \ldots, 1)$ after the pole block has been introduced. The only difference is the procedure to compute $\boldsymbol{x}$ and the form of the pencil prior the pole introduction.

We observe from Figure 4.1 that the accuracy of the shifts rapidly decreases for larger blocksizes in all three cases. We conclude from this experiment that the blocksize should be limited in a practical implementation in order to avoid losing all accuracy in the shifts already at the initialization stage. Indeed, there is not much hope for an effective multishift, multipole RQZ sweep if the shifts that are introduced in the block Hessenberg pencil have few to none significant

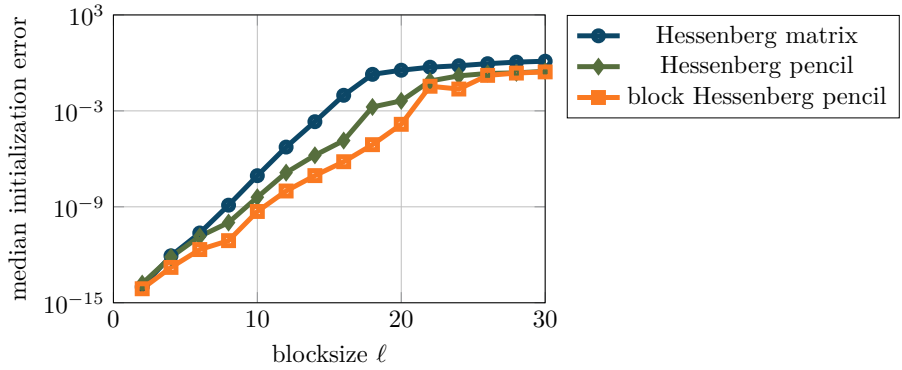


Figure 4.1: Initialization error in function of blocksize for multishift QR (Hessenberg matrix), RQZ (Hessenberg pencil), and multishift, multipole RQZ (block Hessenberg pencil). Median result over 100 randomly generated problems of size n=100.

digits in common with the intended shifts. Nonetheless, Watkins [135] showed that in a multishift QR iteration shifts that are off at start of the sweep can still come into focus later on.

4.4.2 Swapping pole blocks

Swapping two consecutive pole blocks of sizes n_1 and n_2 requires in general the solution of a coupled Sylvester equation. The problem formulation is as follows. We are interested in an equivalence transformation on a block triangular pencil, which in our case is a subpencil of a block Hessenberg pencil:

$$Q^T \left(\begin{bmatrix} A_{11} & A_{12} \\ & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ & B_{22} \end{bmatrix} \right) Z = \left(\begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ & \hat{A}_{22} \end{bmatrix}, \begin{bmatrix} \hat{B}_{11} & \hat{B}_{12} \\ & \hat{B}_{22} \end{bmatrix} \right), \quad (4.21)$$

with blocks (A_{11}, B_{11}), $(\hat{A}_{22}, \hat{B}_{22})$ of dimension n_1 and blocks (A_{22}, B_{22}), $(\hat{A}_{11}, \hat{B}_{11})$ of dimension n_2. Furthermore, we require:

$$\begin{cases} \Lambda(A_{11}, B_{11}) = \Lambda(\hat{A}_{22}, \hat{B}_{22}) = \Xi^1 \\ \Lambda(A_{22}, B_{22}) = \Lambda(\hat{A}_{11}, \hat{B}_{11}) = \Xi^2 \end{cases},$$

and we assume that Ξ^1 and Ξ^2 are disjoint sets. Under these assumptions, the following lemma, taken from [57], uniquely identifies the deflating subspaces and formulates necessary and sufficient conditions for (4.21).

Lemma 4.4.1 ([57]). *Let the pencil (A, B) be block upper triangular form with block sizes $n_1 \times n_1$ and $n_2 \times n_2$ partitioned as in (4.21), where the spectra of (A_{11}, B_{11}) and (A_{22}, B_{22}) are disjoint. Let $X, Y \in \mathbb{R}^{n_1 \times n_2}$ be the solution of:*

$$\begin{cases} A_{11}Y - XA_{22} = A_{12}, \\ B_{11}Y - XB_{22} = B_{12}. \end{cases} \tag{4.22}$$

Then a pair of right deflating subspaces (2.9) for (A_{22}, B_{22}) are spanned by the columns of:

$$\begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix}, \quad \begin{bmatrix} -X \\ I_{n_2} \end{bmatrix}. \tag{4.23}$$

Similarly, a pair of left deflating subspaces for (A_{11}, B_{11}) is given by the row spaces of:

$$\begin{bmatrix} I_{n_1} & X \end{bmatrix}, \quad \begin{bmatrix} I_{n_1} & Y \end{bmatrix}. \tag{4.24}$$

Moreover, the orthogonal equivalence transformations Q and Z swap the spectra of the diagonal blocks in $Q^T(A, B)Z$ if and only if:

$$\begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix} = Z \begin{bmatrix} R_Y \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} I_{n_1} & X \end{bmatrix} = \begin{bmatrix} 0 & R_X \end{bmatrix} Q^T, \tag{4.25}$$

where R_X and R_Y are square and invertible.

The Sylvester equations (4.22) can be solved by a linear system of dimension $2n_1n_2 \times 2n_1n_2$ with Kronecker product structure. The computational cost for the swapping transformations thus rapidly grows for increasing blocksize.

For this reason and because larger multiplicities lead to inaccurate shifts, cfr. Figure 4.1, we propose to represent real poles as subdiagonal blocks of dimension 1 and complex-conjugate pairs as subdiagonal blocks of dimension 2 having complex-conjugate eigenvalues. The 2×2 blocks can be easily maintained in the standard form:

$$\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \begin{bmatrix} b_{11} & b_{12} \\ & b_{22} \end{bmatrix} \right), \tag{4.26}$$

which has the advantage that B is always an upper Hessenberg matrix throughout the iteration. To this end, we need to be able to swap blocks with $n_1 \in \{1, 2\}$ and $n_2 \in \{1, 2\}$. We review the different cases.

Swapping 1×1 with 1×1 blocks

This case was discussed in Section 3.3. Lemma 3.3.2 is proven in Appendix B and shows that the swapping transformation can be performed such that the norm of the off-diagonal elements in $Q^T(A - \lambda B)Z$ is smaller than relative machine precision. The method we proposed is an extension of the work in [122].

Swapping 2×2 with 1×1 blocks

This case is also studied in [122]. We briefly review the method. We are interested in an equivalence transformation $Q^T(A,B)Z = (\hat{A},\hat{B})$ of the following form:

$$Q^T\left(\begin{bmatrix} A_{11} & \boldsymbol{a}_{12} \\ & a_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & \boldsymbol{b}_{12} \\ & b_{22} \end{bmatrix}\right) Z = \left(\begin{bmatrix} \hat{a}_{11} & \hat{\boldsymbol{a}}_{12}^T \\ & \hat{A}_{22} \end{bmatrix}, \begin{bmatrix} \hat{b}_{11} & \hat{\boldsymbol{b}}_{12}^T \\ & \hat{B}_{22} \end{bmatrix}\right), \tag{4.27}$$

with $\Lambda(A_{11},B_{11}) = \Lambda(\hat{A}_{22},\hat{B}_{22}) = \Xi^1 = \{\xi_1^1, \bar{\xi}_1^1\}$, $a_{33}/b_{33} = \hat{a}_{11}/\hat{b}_{11} = \xi_1^2$.

Similar to the discussion in Section 3.3, the swapping is achieved by constructing an orthonormal equivalence $[\boldsymbol{q}_1\ Q_2]$, $[\boldsymbol{z}_1\ Z_2]$ such that $\boldsymbol{q}_1$, $\boldsymbol{z}_1$ is a deflating pair (2.9) for the eigenvalue ξ_1^2, since in that case $Q_2^T A\boldsymbol{z}_1 = Q_2^T B\boldsymbol{z}_1 = \boldsymbol{0}$.

To construct an orthonormal Z with its first column equal to the right eigenvector of (A,B) corresponding to ξ_1^2, we consider the matrix $H = b_{33}A - a_{33}B$. Observe that the last row of H is equal to zero. Next we compute $\hat{H} = C_1 H$ with C_1 a core transformation that introduces a zero in position $(2,1)$ of H. The matrix Z can then be computed by means of two core transformations as follows:

$$\hat{H}\underbrace{Z_2 Z_1}_{Z} = \begin{matrix} \times\times\times \\ 0\times\times \\ 0\ 0\ 0 \end{matrix} \begin{matrix} \curvearrowright \\ \curvearrowright \\ \end{matrix} = \begin{matrix} 0\times\times \\ 0\ 0\times \\ 0\ 0\ 0 \end{matrix},$$

where Z_2 first eliminates element $(2,2)$ in $\hat{H}$ and afterwards Z_1 eliminates element $(1,1)$. Observe that this Z matrix also sets the first column of H to zero, which implies that $\boldsymbol{z}_1$ indeed corresponds to the desired eigenvector.

The matrix BZ_2Z_1 is an upper Hessenberg matrix which can be restored to upper triangular form by two core transformations Q_1^T, Q_2^T, i.e. $\hat{B} = Q_2^T Q_1^T B Z_1 Z_2$. Here Q_1^T first eliminates element $(2,1)$ and afterwards Q_2^T eliminates element $(3,2)$. This maintains the standard form (4.26). The equivalence transformation $Q^T = Q_2^T Q_1^T$, $Z = Z_1 Z_2$ swaps the blocks in (4.27).

Under the condition that $|b_{33}| \geq |a_{33}|$ the following bound on the backward error is provided in [122]. In case $|b_{33}| < |a_{33}|$, the roles of A and B can be reversed to obtain the same bound.

Lemma 4.4.2. *Let $Q^T(A,B)Z = (\hat{A},\hat{B})$ be related as in (4.27) with Q and Z computed as described above and let $|b_{33}| \geq |a_{33}|$. We have that the computed quantities satisfy:*

$$\tilde{Q}^T(A+E_A, B+E_B)\tilde{Z} = \left(\begin{bmatrix} \tilde{a}_{11} & \tilde{\boldsymbol{a}}_{12}^T \\ & \tilde{A}_{22} \end{bmatrix}, \begin{bmatrix} \tilde{b}_{11} & \tilde{\boldsymbol{b}}_{12}^T \\ & \tilde{B}_{22} \end{bmatrix}\right),$$

with $\|E_A\|_2 \leq \epsilon_m \Delta$, $\|E_B\|_2 \leq \epsilon_m \Delta$, $\Delta = \max(\|A\|_2, \|B\|_2)$.

Remark 4.4.3. Numerical evidence suggests that changing the criterion in Lemma 4.4.2 from $|b_{33}| \geq |a_{33}|$ to $|\xi_1^1| \geq |\xi_1^2|$ leads to a method for which $\|E_A\|_2 \leq c\epsilon_m \|A\|_2$ and $\|E_B\|_2 \leq c\epsilon_m \|B\|_2$. A detailed error analysis supporting this finding, like in Appendix B, is at the time of writing still under development.

Swapping 1×1 with 2×2 blocks

This case is dual to the 2×2 with 1×1 swap and can be solved with an analogous method.

We remark that in the previous three cases it was always possible to directly compute the deflating subspaces (4.23), (4.24) or both because at least one of the blocks is of dimension 1, which allowed us to easily computed the related eigenvectors. We (partially) solved (4.22) implicitly to do so.

Swapping 2×2 with 2×2 blocks

In case $n_1 = n_2 = 2$ it is no longer possible to easily compute the required subspaces by introducing zeros like before and we solve (4.22) for X and Y. The idea is then to construct a pair of orthonormal equivalence transformations Q and Z that achieve the swapping from the QR factorizations (4.25).

The following lemma summarizing the error analysis from [57] holds in this case.

Lemma 4.4.4 ([57])**.** *Let* $\tilde{X}$ *and* $\tilde{Y}$ *be the computed solutions of the generalized Sylvester equation* (4.22)*. Let*

$$E = -A_{12} - A_{11}\tilde{Y} + \tilde{X}A_{22}, \quad \textit{and,} \quad F := -B_{12} - B_{11}\tilde{Y} + \tilde{X}B_{22},$$

be their residuals and let $\tilde{Q}$ *and* $\tilde{Z}$ *be the computed factors of the QR factorizations*

$$\begin{bmatrix} -\tilde{Y} \\ I \end{bmatrix} = \tilde{Z} \begin{bmatrix} \tilde{R}_Y \\ 0 \end{bmatrix}, \quad \begin{bmatrix} I \\ \tilde{X}^T \end{bmatrix} = \tilde{Q} \begin{bmatrix} 0 \\ \tilde{R}_X^T \end{bmatrix}.$$

Then the computed equivalence transformation satisfies:

$$\left(\begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \Delta_A & \tilde{A}_{22} \end{bmatrix}, \begin{bmatrix} \tilde{B}_{11} & \tilde{B}_{12} \\ \Delta_B & \tilde{B}_{22} \end{bmatrix} \right) = \tilde{Q}^T \left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix} \right) \tilde{Z},$$

where,

$$\|\Delta_A\|_2 \leq \|E\|_F / \sqrt{(1+\sigma_2(X)^2)(1+\sigma_2(Y)^2)},$$

$$\|\Delta_B\|_2 \leq \|F\|_F / \sqrt{(1+\sigma_2(X)^2)(1+\sigma_2(Y)^2)}.$$

The error bound does *not* imply that the off-diagonal block (Δ_A, Δ_B) can be safely dismissed according to $\|\Delta_A\| \leq \epsilon_m \|A\|_2$, $\|\Delta_B\| \leq \epsilon_m \|B\|_2$. Nevertheless, the bound is often pessimistic and the observed errors often allow (Δ_A, Δ_B) to be safely dismissed.

Swap refinement

Of all the swapping procedures that we have discussed, only the 1×1 with 1×1 case has a backward error bound that guarantees that the off-diagonal blocks can always be safely dismissed, albeit we have a method that satisfies this in practice for 1×1 with 2×2 swaps.

The error bound for 2×2 with 2×2 swaps of Lemma 4.4.4 indicates that the off-diagonal block cannot always be dismissed.

Whenever the result of a swapping transformation has an off-diagonal block in A, B, or both that is too large to be directly dismissed. We propose to iteratively refine the transformation [17] based on a linear approximation of the involved Riccati equations. Refinement is required when the situation after the initial transformation is as follows:

$$\left(\begin{bmatrix} A_{11} & A_{12} \\ \Delta_A & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ \Delta_B & B_{22} \end{bmatrix} \right) \tag{4.28}$$

with the $(1,1)$ blocks of dimension $n_2 \times n_2$, the $(2,2)$ blocks of dimension $n_1 \times n_1$, and the $(2,1)$ blocks satisfying:

$$\|\Delta_A\|_2 > c\epsilon_m \|A\|_2 \quad \text{and/or} \quad \|\Delta_B\|_2 > c\epsilon_m \|B\|_2,$$

with c some small refinement threshold.

In this case we need to solve the system of quadratic matrix equations:

$$\begin{aligned} \Delta_A - A_{22}Y + XA_{11} - XA_{12}Y &= 0, \\ \Delta_B - B_{22}Y + XB_{11} - XB_{12}Y &= 0, \end{aligned}$$

for $X, Y \in \mathbb{R}^{n_1 \times n_2}$. These quadratic equations can be approximated by the system of linear matrix equations:

$$\begin{aligned} \Delta_A &= A_{22}Y - XA_{11}, \\ \Delta_B &= B_{22}Y - XB_{11}, \end{aligned}$$

since $\|X\|_2$ and $\|Y\|_2$ are typically very small as $\|\Delta_A\|_2$ and $\|\Delta_B\|_2$ are already small. The solution (X, Y) of this linear system can be computed using Kronecker products.

Solving the linearized Riccati equation corresponds to a single step of a Newton method, which is a widely used method to solve quadratic matrix equations, see e.g. [72].

The result is used to construct the orthonormal equivalence transformation:

$$Q_{up} = \begin{bmatrix} I_{n_2} & X^T \\ -X & I_{n_1} \end{bmatrix} \begin{bmatrix} R_X & 0 \\ 0 & R_{X^T} \end{bmatrix}, \quad Z_{up} = \begin{bmatrix} I_{n_2} & Y^T \\ -Y & I_{n_1} \end{bmatrix} \begin{bmatrix} R_Y & 0 \\ 0 & R_{Y^T} \end{bmatrix}$$

where R_X, R_{X^T} R_Y and R_{Y^T} are normalization factors to make Q_{up} and Z_{up} orthonormal. Which updates (4.28) to $(\breve{A}, \breve{B}) = Q_{up}^T(A, B)Z_{up}$.

The norm of the $(2, 1)$ blocks is checked again and if required the same procedure can be repeated to further reduce it.

Our numerical experiments indicated that iterative refinement is required in about 5% of all 2×2 with 2×2 swaps during a typical RQZ iteration. A single refinement iteration suffices in the majority of the cases. If the method does not converge after 5 iterations, the swap is rejected.

4.4.3 Deflation monitoring

In order to limit both the computational cost of the pole introduction and swapping, and the loss of accuracy, we propose a *tightly-packed small-block* multishift, multipole RQZ sweep. The shifts and poles are tightly-packed similar to Section 3.5.4.

This also simplifies the deflation criteria based on Definition 4.2.5. The ith pole along the subdiagonal is considered deflated if,

$$|a_{i+1,i}| < c\epsilon_m(|a_{i,i}| + |a_{i+1,i+1}|), \quad \text{and,} \quad |b_{i+1,i}| < c\epsilon_m(|b_{i,i}| + |b_{i+1,i+1}|), \tag{4.29}$$

in the case of a single pole. If the ith pole is a double pole in standard form (4.26), we consider it deflated if either,

$$\begin{aligned} |a_{i+1,i}| + |a_{i+2,i}| < c\epsilon_m(|a_{i,i}| + |a_{i+1,i+1}|), \quad \text{and,} \\ |b_{i+1,i}| < c\epsilon_m(|b_{i,i}| + |b_{i+1,i+1}|), \end{aligned} \tag{4.30}$$

or,

$$\begin{aligned} |a_{i+2,i}| + |a_{i+2,i+1}| < c\epsilon_m(|a_{i+1,i+1}| + |a_{i+2,i+2}|), \quad \text{and,} \\ |b_{i+2,i+1}| < c\epsilon_m(|b_{i+1,i+1}| + |b_{i+2,i+2}|). \end{aligned} \tag{4.31}$$

Deflations in the first block column and last block row of the pencil are also checked according to Definition 4.2.5. The first pole block of size $s_1 = 1$ or 2 can be deflated whenever there exists an $(s_1+1)\times(s_1+1)$ orthogonal matrix Q such that,

$$Q^T \left(\begin{bmatrix} \boldsymbol{a}_{1,1}^T \\ A_{2,1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}_{1,1}^T \\ B_{2,1} \end{bmatrix} \right) = \left(\begin{bmatrix} A_{1,1} \\ \boldsymbol{0}^T \end{bmatrix}, \begin{bmatrix} B_{1,1} \\ \boldsymbol{0}^T \end{bmatrix} \right) \tag{4.32}$$

Here, the last row is considered numerically zero according to a relative tolerance similar to (4.29), (4.30), and (4.31). Again, we make use of the standard form (4.26) to efficiently check if a suitable deflation transformation Q can be computed in case $s_1 = 2$. A similar approach is used to check for deflations in the last block row.

4.5 Aggressive early deflation

Aggressive early deflation (AED) significantly speeds up the convergence of the QR [15] and QZ [58] methods by identifying deflatable eigenvalues before classical deflation criteria are able to detect them. This avoids the reuse of converged shifts in subsequent iterations, thereby initiating convergence of other eigenvalues sooner.

In this section, we describe how aggressive early deflation is implemented for the RQZ method. The process consists of 3 stages and is summarized in Figure 4.2. Because the shifts lead to convergence in the bottom-right corner of the pencil and the poles cause convergence in the upper-left corner, AED can be performed at both sides of the pencil. We present the description of the AED process simultaneously for the upper-left and bottom-right sides of the pencil, but they can be treated separately in a practical implementation. The deflation window sizes are w_e for the bottom-right side and w_s for the upper-left side of the pencil. The window sizes are chosen such that they cover an integer number of blocks, avoiding thereby subdivision of 2×2 blocks. The deflation windows are shown in Pane I of Figure 4.2.

In the first phase, shown in pane II of Figure 4.2, the parts of the pencil within the deflation windows are reduced to real Schur form. This can be done with the RQZ method as all subpencils in the deflation windows are in block Hessenberg

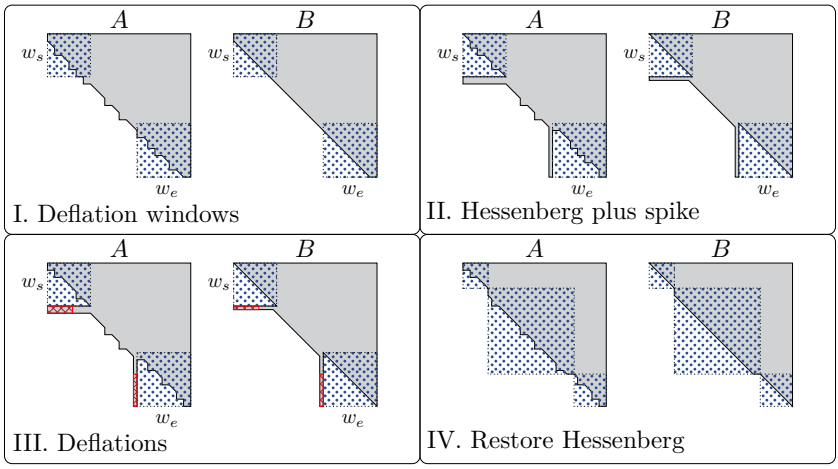


Figure 4.2: Visualization of the three stages of aggressive early deflation for block Hessenberg pencils; both at the front and back of the pencil. The matrix A is in block Hessenberg form with 2×2 blocks representing complex conjugate pairs of shifts, the matrix B is in Hessenberg form.

form. The pencil (A, B) is subdivided as,

$$A = \begin{array}{c|ccc|c} & 1 \text{ or } 2 & & & \\ \hline A_{11} & A_{12} & A_{13} & A_{14} & \\ A_{21} & A_{22} & A_{23} & A_{24} & 1 \text{ or } 2 \\ & A_{32} & A_{33} & A_{34} & \\ \hline & & A_{43} & A_{44} & \end{array}, \quad B = \begin{array}{c|ccc|c} & 1 & & & \\ \hline B_{11} & B_{12} & B_{13} & B_{14} & \\ B_{21} & B_{22} & B_{23} & B_{24} & 1 \\ & B_{32} & B_{33} & B_{34} & \\ \hline & & B_{43} & B_{44} & \end{array}, \tag{4.33}$$

and the subpencils (A_{11}, B_{11}) and (A_{44}, B_{44}) are the upper-left and bottom-right deflation windows. Their reduction to real Schur form is given by,

$$(S_{11}, T_{11}) = Q_s^T (A_{11}, B_{11}) Z_s, \quad \text{and,} \quad (S_{44}, T_{44}) = Q_e^T (A_{44}, B_{44}) Z_e, \tag{4.34}$$

which, when applied as an equivalence transformation to (A, B) gives the following result:

$$\check{A} = \begin{array}{c|ccc} S_{11} & Q_s^T A_{12} & Q_s^T A_{13} & Q_s^T A_{14} Z_e \\ A_{21} Z_s & A_{22} & A_{23} & A_{24} Z_e \\ & A_{32} & A_{33} & A_{34} Z_e \\ \hline & & Q_e^T A_{43} & S_{44} \end{array}, \check{B} = \begin{array}{c|ccc} T_{11} & Q_s^T B_{12} & Q_s^T B_{13} & Q_s^T B_{14} Z_e \\ B_{21} Z_s & B_{22} & B_{23} & B_{24} Z_e \\ & B_{32} & B_{33} & B_{34} Z_e \\ \hline & & Q_e^T B_{43} & T_{44} \end{array}. \tag{4.35}$$

The blocks $(A_{21}, B_{21}) Z_s$ and $Q_e^T (A_{43}, B_{43})$ are the spikes shown in pane II of Figure 4.2. Because B is an upper Hessenberg matrix by (4.26), $B_{21} =$

$b_{w_s+1,w_s}\boldsymbol{e}_{w_s}^T$ is of dimension $1\times w_s$ and $B_{43} = b_{n-w_e+1,n-w_e}\boldsymbol{e}_1$ is of dimension $w_e\times 1$. The spikes at the side of A can be of dimension $2\times w_s$ or $w_e\times 2$ if there is a 2×2 block just after the deflation window in the upper-left side of the pencil (the example of Figure 4.2 illustrates this situation), or right before the deflation window at the bottom-right side of the pencil. In this case, the 2 rows of $A_{21}Z_s$ are scalar multiples of each other. The same holds for the 2 columns of $Q_e^T A_{43}$. We denote with $\boldsymbol{p}_s^B = b_{w_s+1,w_s}\boldsymbol{e}_{w_s}^T Z_s$ the spike at the upper-left deflation window of B. Similarly, $\boldsymbol{p}_s^A = \zeta \boldsymbol{e}_{w_s}^T Z_s$, with ζ equal to the maximum of $|a_{w_s+1,w_s}|$ and $|a_{w_s+2,w_s}|$, denotes the spike at the upper-left side of A

The second phase in the AED process is illustrated in Pane III of Figure 4.2 and entails testing for deflatable eigenvalues inside the deflation windows. The deflation test starts at the left of the spikes $\boldsymbol{p}_s^A$ and $\boldsymbol{p}_s^B$. If there is a 1×1 real eigenvalue located at this position, we test if:

$$|\boldsymbol{p}_{s,1}^A| < c\epsilon_m(|a_{1,1}| + |a_{2,2}|), \quad \text{and,} \quad |\boldsymbol{p}_{s,1}^B| < c\epsilon_m(|b_{1,1}| + |b_{2,2}|). \tag{4.36}$$

If there is a 2×2 complex conjugate pair of eigenvalues at this position, we test if:

$$|\boldsymbol{p}_{s,1}^A| + |\boldsymbol{p}_{s,2}^A| < c\epsilon_m\|A(1{:}2,1{:}2)\|_F, \quad \text{and,} \quad |\boldsymbol{p}_{s,1}^B| + |\boldsymbol{p}_{s,2}^B| < c\epsilon_m\|B(1{:}2,1{:}2)\|_F. \tag{4.37}$$

If the first eigenvalue is deflatable according to (4.36) or (4.37), the corresponding spike elements in $\boldsymbol{p}_s^A$ and $\boldsymbol{p}_s^B$ are set to zero and the next eigenvalue is tested according to the same criterion. If the first eigenvalue is not deflatable, another eigenvalue that has not yet been tested, is swapped to the front of the spike. Then it is checked if this is deflatable according to (4.36) or (4.37). This procedure is continued until all deflatable eigenvalues inside the deflation window are identified. The swapping of eigenvalues within the deflation window does not change the form of (4.34) but the equivalences $\hat{Q}_s$ and $\hat{Z}_s$ are changed which also changes $\hat{\boldsymbol{p}}_s$ and $(\hat{S}_{11}, \hat{T}_{11})$. The same strategy is used for AED at the bottom-right side of the pencil. In pane III of Figure 4.2 all spike elements that have been zeroed are marked in red.

In the third and last phase, the nonzero spike elements are handled in such a way that the (block) Hessenberg form is restored. The restored form is shown in pane IV of Figure 4.2, where the larger block in the middle is in block Hessenberg form and the smaller blocks at the upper-left and bottom-right side of the pencil are in real Schur form. The block Hessenberg restoration is achieved by a sequence of rotations as follows. Assume that the spikes after the deflation procedure of second phase are $\hat{\boldsymbol{p}}_s = \hat{\zeta}\boldsymbol{e}_{w_s}^T \hat{Z}_s$ and that the first i entries in $\hat{\boldsymbol{p}}_s$ are zeroed during the deflation step. We then compute rotations $G_{i+1}, \ldots, G_{w_s-1}$ such that,

$$\hat{\boldsymbol{p}}_s G_{i+1}, \ldots, G_{w_s-1} = \hat{\zeta}\boldsymbol{e}_{w_s}^T \hat{Z}_s G_{i+1}, \ldots, G_{w_s-1} = \sigma\hat{\zeta}\boldsymbol{e}_{w_s}^T. \tag{4.38}$$

Updating $\tilde{Z}_s = \hat{Z}_s G_{i+1}, \ldots, G_{w_s-1}$ gives the final equivalence such that the block Hessenberg form is restored as all spikes are scalar multiples. The same idea is used for the deflation window at the bottom-right side of the pencil. We remark that for complex-valued problems the Hessenberg form can be restored in the third phase by a row or column permutation for respectively AED at the upper-left or bottom-right side of the pencil.

4.6 Numerics

The numerical tests have been performed on an Intel Xeon E5-2697 v3 CPU with 14 cores and 128GB of RAM. Our implementation of the multishift, multipole RQZ method with aggressive deflation is compiled with *gfortran* version 4.8.5 using compilation flag `-O3`. LAPACK 3.8.0 [2] and BLAS 3.8.0 are used. The library `libRQZ` supports both real-valued (`dRQZm`) and complex-valued (`zRQZm`) problems.

4.6.1 dRQZm and zRQZm

As discussed in Section 4.4, `dRQZm` uses 1×1 blocks for real poles and 2×2 blocks for pairs of complex-conjugate shifts, `zRQZm` always uses 1×1 blocks. Both algorithms proceed as follows:

I. Check for deflations at the upper-left side of the pencil using AED with window size w_s.

II. Check for interior deflations along the subdiagonal.

III. Compute m shifts as the eigenvalues of the trailing $m \times m$ block with the RQZ method and introduce these as consecutive poles in the first m subdiagonal positions of the block Hessenberg pencil. This is achieved by using the operations of Section 4.2.3. The involved transformations are accumulated and the pencil is updated by level-3 BLAS matrix-matrix multiplication, cfr. Section 2.3.4.

IV. Chase the batch of m shifts to the last m positions on the subdiagonal of the block Hessenberg pencil. The chasing is performed by repeatedly swapping the m shifts with the next k poles. Every time one sequence of swaps is computed, all transformations are accumulated and the pencil is updated by level-3 BLAS matrix-matrix multiplication, cfr. Section 2.3.4.

V. Check for deflations at the bottom-right side of the pencil using AED with window size w_e.

VI. Compute m poles as the eigenvalues of the leading $m \times m$ block with the RQZ method and introduce these as consecutive poles in the last m subdiagonal positions of the block Hessenberg pencil. This is achieved by using the operations of Section 4.2.3. The involved transformations are accumulated and the pencil is updated by level-3 BLAS matrix-matrix multiplication, cfr. Section 2.3.4.

This algorithm actively chases shifts from the upper-left corner to the bottom-right corner. This typically leads to fast convergence of eigenvalues near the bottom-right side of the pencil. The swapping also slowly moves the poles that are introduced at the bottom-right corner to the upper-left side of the pencil which, in turn, induces convergence of eigenvalues near the upper-left corner of the pencil.

The heuristics used for block sizes in `libRQZ` are summarized in Table 4.1. The first column lists the size of the pencil. The second column lists the batch size m of shifts that are handled in one iteration. The third column lists the swap size k: after the m shifts have been swapped with the next k poles, the transformations are accumulated and the entire pencil is updated with a BLAS `xGEMM` call. In our experience, choosing k equal to m gives the best performance. The fourth column lists the window size w_e for aggressive early deflation at the bottom-right side of the pencil. Finally, the fifth column lists the window size w_s for aggressive early deflation at the upper-left side of the pencil.

Table 4.1: `libRQZ` settings: n problem size, m step multiplicity, k swap range, w_e AED window size at the bottom-right side of the pencil, w_s AED window size at the upper-left side of the pencil.

n	m	k	w_e	w_s
$[1; 80[$	1—2	1—2	1—2	1—2
$[80; 150[$	4	4	6	4
$[150; 250[$	8	8	10	4
$[250; 501[$	16	16	18	6
$[501; 1001[$	32	32	34	10
$[1001; 3000[$	64	64	66	16
$[3000; 6000[$	128	128	130	32
$[6000; \infty[$	256	256	266	48

We compare `zRQZm` and `dRQZm` with respectively `ZHGEQZ` and `DHGEQZ` from LAPACK [2] in terms of speed and accuracy.

4.6.2 Random problems

For our first numerical experiment, we have generated random matrix pairs of increasing dimension. The entries of the matrices are drawn from the standard normal distribution. The experiment is performed both for real-valued and complex-valued matrix pairs; for the latter class of problems, both the real and imaginary part are randomly generated.

The matrix pairs are initially reduced to Hessenberg, triangular form by means of the LAPACK [2] routines `xGEQRFP` and `xGGHRD`. After this initial reduction, the matrix pairs are further reduced to (real) generalized Schur form, $(S, T) = Q^*(A, B)Z$, with `libRQZ` and LAPACK [2]. In all cases, the entire Schur decomposition is computed.

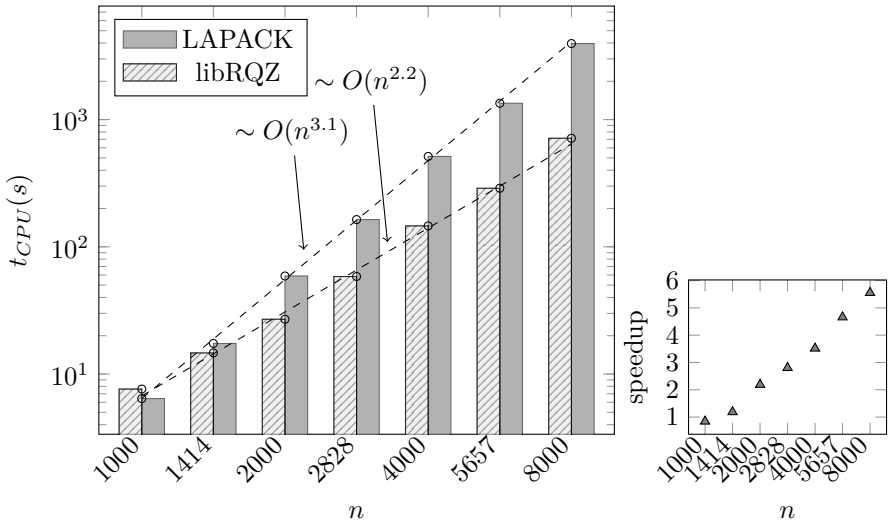


Figure 4.3: CPU time of `DHGEQZ` of LAPACK and `dRQZm` of libRQZ on randomly generated real-valued matrix pencils (*left*). Speedup of libRQZ over LAPACK (*right*).

The left part of Figure 4.3 shows the CPU time of `dRQZm` and `DHGEQZ` for problems of size 1000 up to 8000 on a loglog scale. The dashed lines indicate the slopes of the time complexity in function of problem size, which are estimated in a least-squares sense. The least-square fits are computed based on the (n_i, t_i) data indicated with the circular markers that show the exact height of the bars in the graph. For `DHGEQZ` we observe an empirical time complexity close to $O(n^3)$, while the empirical time complexity of `dRQZm` is significantly lower than $O(n^3)$ with a leading exponent close to 2.2. This improved time complexity can

be attributed to the effectiveness of aggressive early deflation in combination with the rational iteration leading to occasional deflations situated more in the interior part of the pencil.

The right part of Figure 4.3 shows the speedup achieved by `dRQZm` over `DHGEQZ`. The crossover point where `dRQZm` becomes faster than `DHGEQZ` is situated between $n = 1000$ and $n = 1414$. Our method, `dRQZm`, is slower than `DHGEQZ` for problems of smaller size because the computational overhead of computing swaps of 2×2 with 2×2 blocks and 2×2 with 1×1 blocks leads to larger lower-order terms in the time complexity.

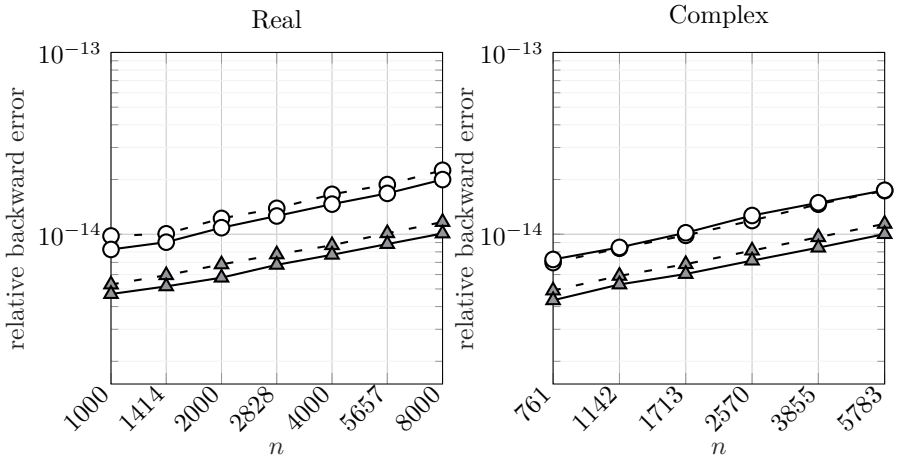


Figure 4.4: Relative backward error on Schur decomposition computed with LAPACK (*circles*) and libRQZ (*triangles*) on A (*full lines*) and B (*dashed lines*). Both for real-valued (*left*) and complex-valued (*right*) randomly generated matrix pairs.

The left part of Figure 4.4 shows the relative backward errors,

$$\|S - Q^T AZ\|_F / \|A\|_F, \quad \text{and,} \quad \|T - Q^T BZ\|_F / \|B\|_F,$$

on the generalized real Schur decompositions obtained with `dRQZm` and `DHGEQZ`. We observe that the relative backward errors of `dRQZm` are about half of these of `DHGEQZ`.

Figure 4.5 shows the results of an experiment similar to Figure 4.3 but for complex-valued pencils. Again, `ZHGEQZ` shows an empirical time complexity larger than $O(n^3)$, while `zRQZm` stays below $O(n^3)$. The crossover point where `ZHGEQZ` is faster than `zRQZm` is not shown in Figure 4.5, but is situated around $n = 200$. This is significantly lower than for `dRQZm` and is explained by the

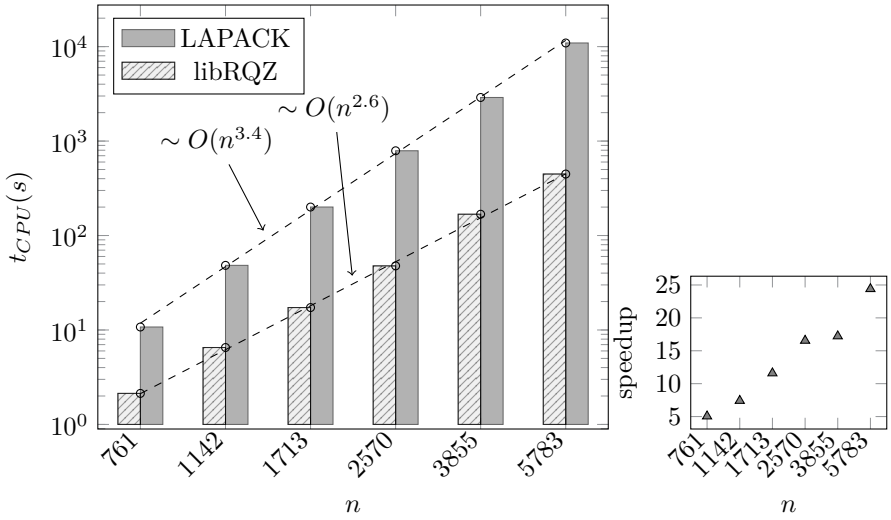


Figure 4.5: CPU time of `ZHGEQZ` of LAPACK and `zRQZm` of libRQZ on randomly generated complex-valued matrix pencils (*left*). Speedup of libRQZ over LAPACK (*right*).

fact that only 1×1 with 1×1 swaps are used in this case. These have a lower computational overhead than larger swaps. The right part of Figure 4.4 shows the relative backward errors on the generalized Schur decompositions for the complex-valued pencils. Again, the relative backward error of `zRQZm` is about half of `ZHGEQZ`.

4.6.3 Problems from applications

In this section we test `libRQZ` on seven pencils originating from applications. We study the *cavity* and *obstacle flow* pencils generated with IFISS [34, 35]. The same pencils were studied in the previous chapter. Besides these pencils, we have selected four pencils from Matrix market [13] originating from the BFWAVE, BCSSTRUC4, and MHD collection and the *rail* pencil from the Oberwolfach benchmark collection [65].

The results of the numerical tests are summarized in Table 4.2. The table lists the CPU time and maximum of the relative backward errors on A and B for the generalized real Schur form computed with LAPACK [2] and `libRQZ`. Again, `libRQZ` requires less CPU time and has the smaller backward error.

Table 4.2: CPU times and maximum relative backward error on the generalized real Schur form computed with LAPACK and `libRQZ` for pencils originating from applications.

		`DHGEQZ`		`dRQZm`	
Problem	n	$t_{\text{CPU}}(s)$	max error (A, B)	$t_{\text{CPU}}(s)$	max error (A, B)
BFW782	782	4.5	$2.8 \cdot 10^{-14}$	3.7	$5.0 \cdot 10^{-15}$
BCSST26	1922	33.0	$4.1 \cdot 10^{-14}$	12.8	$3.9 \cdot 10^{-15}$
Cavity Flow	2467	50.1	$1.2 \cdot 10^{-14}$	20.4	$4.7 \cdot 10^{-15}$
Obstacle Flow	2488	64.0	$9.9 \cdot 10^{-15}$	27.9	$5.9 \cdot 10^{-15}$
MHD3200	3200	60.8	$9.0 \cdot 10^{-15}$	39.6	$3.1 \cdot 10^{-15}$
MHD4800	4800	194.1	$1.4 \cdot 10^{-14}$	92.2	$3.4 \cdot 10^{-15}$
RAIL	5177	1287.5	$1.7 \cdot 10^{-13}$	87.5	$1.1 \cdot 10^{-14}$

4.7 Conclusion and future work

In this chapter we have generalized the rational QZ method from Hessenberg to block Hessenberg pencils. This allows for the use of complex conjugate shifts and poles in real arithmetic. Numerical considerations have shown that medium to large multiplicities are unfavorable due to inherent inaccuracies and an increasing computational complexity. In the spirit of recent developments for the QR [14, 15] and QZ [58] methods, this urged us to use small shift and pole multiplicities, but they can be tightly-packed together. This approach maintains accurate shifts and poles in combination with level 3 BLAS performance. We also implemented the aggressive early deflation strategy for block Hessenberg pencils. Numerical experiments indicated that this combination leads to an efficient algorithm for the generalized eigenvalue problem that can outperform LAPACK [2] in terms of speed, accuracy, and time complexity. However, the implementation of the QZ method in LAPACK 3.8.0 does not include aggressive early deflation and blocking. A further comparison with [58] would be interesting. In Chapter 5, we provide further numerical evidence that pole swapping algorithms can outperform bulge chasing algorithms.

In a future update of `libRQZ`, we plan to implement *bidirectional* RQZ sweeps that actively chase poles from the bottom-right to the upper-left corner of the pencil in parallel to chasing shifts from the upper-left to the bottom-right corner. Bidirectional chasing can, for a large part, be performed independently in both directions. It is hence an excellent opportunity for shared-memory parallelization. On the theoretical side, a further investigation of shift and pole selection strategies that stimulate interior deflations would be an interesting undertaking.

Chapter 5

Rational QZ for Hessenberg, unitary Hessenberg pencils

This chapter proposes a computationally efficient formulation of the rational QZ method for the case of proper Hessenberg pencils where one of both matrices is unitary. The content is based on:

Camps D., Mach T., Vandebril R., and Watkins D. S., Pole swapping methods for Hessenberg, unitary Hessenberg pencils: Rational QR algorithms. In preparation.

5.1 Introduction

The rational QZ method computes the generalized Schur decomposition (2.12) of a dense, unsymmetric Hessenberg pair (A, B) using a pole swapping algorithm. The pole swapping generalization has several advantages over the bulge chasing strategy. The most notable advantage is an improved convergence behaviour determined by subspace iteration accelerated by rational functions, cfr. Theorem 3.7.3.

It would be advantageous to have a similar method for the standard eigenvalue problem (2.1) defined by a dense, unsymmetric Hessenberg matrix A. In principle the rational QZ method of Chapter 3 can be directly used for a

proper Hessenberg pair (A, I) but this has two major disadvantages over using Francis' bulge chasing algorithm. Firstly, it requires double the storage space as it works on two $n \times n$ matrices instead of one. Secondly, the computational cost of an implicit RQZ step on the Hessenberg pair (A, I) is also double the cost of an implicit QR step on the Hessenberg matrix A as the former works with equivalence transformations on two matrices while the latter carries out similarity transformations on a single matrix. The reduction in the number of iterations and swaps as a result of using a rational instead of a polynomial iteration is insufficient to compensate for this.

In the current chapter we present an efficient formulation of the rational QZ algorithm for matrix pairs where one of both matrices is *unitary*. We denote Hessenberg, unitary Hessenberg pairs as (A, U). It is clear that the unitarity of U is preserved under unitary equivalence transformations. Furthermore it is self-evident that an upper Hessenberg matrix A can also be represented as a Hessenberg matrix pair (A, U).

The rational QZ method that we propose in this chapter uses an efficient storage scheme for the unitary matrix U as a sequence of *core transformations*. This approach effectively overcomes both drawbacks: both the storage requirements and the computational cost approximately halve. We pay special attention to stability of the numerical scheme.

We mainly consider general Hessenberg, unitary Hessenberg pencils (A, U) but if the algorithm is used for the solution of a standard eigenvalue problem, (A, I), we can always compute a generalized Schur decomposition (2.12) such that $Q^*(A, I)Z = (T, I)$. It follows from $Q^*Z = I$ that this is a Schur decomposition (2.5) of A. If the Schur vectors are required, it suffices to only accumulate Q. This has the same computational complexity as for Francis' polynomial QR method. We call the algorithm the *rational QR* method if it is used for the standard eigenvalue problem.

This chapter is organized as follows. Section 5.2 introduces the representation and discusses the details required to perform a rational QZ step on the format such as deflation monitoring, pole introduction, and pole swapping. Section 5.3 discusses that a rational QZ method based on the proposed format indeed requires approximately half the storage space and computational cost of the dense rational QZ method. Section 5.4 present numerical results obtained with `ZLAHPS`, our fortran implementation of the rational QR algorithm, and compare them to the `ZLAHQR` routine from LAPACK [2]. We demonstrate that the pole swapping algorithm can lead to a reduction of more than 25% in CPU time and leads to a more accurate Schur decomposition. Section 5.5 concludes this chapter.

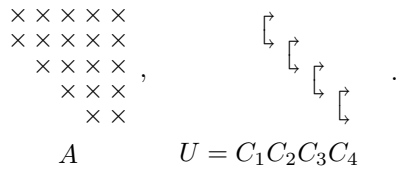
5.2 Hessenberg, unitary Hessenberg pencils

Starting from a proper $n \times n$ Hessenberg, unitary Hessenberg pair (A, U), we store the unitary Hessenberg matrix U as a *sequence* or *pattern* of $n-1$ unitary core transformations (2.42). Within the class of unitary core transformations, we make a further restriction to core transformations which have a rotation matrix (2.43) as *active part*. Furthermore, in `ZLAHPS` we represent the complex rotation by a complex cosine and a real sine without loss of generality. This leads to a reduction in computational cost as the multiplication of a real and complex variable requires less than half the number of operations compared to multiplying two complex variables.

It is always possible to factorize a unitary upper Hessenberg U as a sequence of core transformations. To this end, compute $n-1$ core transformations that create zeros in the subdiagonal elements of the upper Hessenberg matrix U [46]:

$$C_{n-1}^* \dots C_1^* U = R = D_\alpha = \mathrm{diag}(1, \dots, 1, \alpha).$$

The result is an upper triangular matrix R but since the left-hand side above is unitary, R is also unitary and hence must be diagonal. Furthermore, the cosine of the rotation C_i can always be chosen such that $d_{ii} = 1$. The result is a diagonal matrix $D_\alpha = \mathrm{diag}(1, \dots, 1, \alpha)$ and the original pencil is thus $(A, C_1 \dots C_{n-1} D_\alpha)$. The equivalent pencil $(AD_\alpha^*, C_1 \dots C_{n-1})$ only differs in its last column from (A, U), it is in the desired factorized form and has the exact same poles as (A, U). Without loss of generality, we can thus represent any proper Hessenberg, unitary Hessenberg pencil in the format illustrated for a problem of dimension 5:



This requires $O(n^2)$ storage space for A but only $O(n)$ for U if every core transformation is stored by its sine and cosine. For large n, the storage requirements for $(A, C_1 \dots C_{n-1})$ is thus approximately half of the dense representation.

For the standard eigenvalue problem all core transformations can be initialized as identity transformations.

The poles of a proper Hessenberg pencil $(A, U_1 \ldots, U_{n-1})$ are, by (2.43), equal to:

$$\xi_i = \frac{a_{i+1,i}}{s_i} \in \bar{\mathbb{C}} \quad \text{for} \quad i = 1, \ldots n-1. \tag{5.1}$$

Properness of the Hessenberg pencil translates in a straightforward manner to the factorized format and also deflations can also be easily monitored. To detect a deflatable eigenvalue at the top-left position of the pencil, it suffices to compute $C_1^* A(1{:}2, 1{:}2)$ and check if the $(2,1)$-element is negligible up to relative machine precision. A similar strategy is used for the bottom-right pole by considering $A(n{-}1{:}n, n{-}1{:}n)C_{n-1}^*$. Poles along the subdiagonal can be safely deflated if:

$$|a_{i+1,i}| < \epsilon_m(|a_{i,i}| + |a_{i+1,i+1}|), \quad \text{and,} \quad |s_i| < \epsilon_m.$$

5.2.1 Manipulating poles

In this section we review the two pole manipulation operations for proper Hessenberg pencils, introduced in Section 3.3, for a proper Hessenberg pair in the form $(A, U = C_1 \ldots C_{n-1})$ with pole tuple $\Xi = (\xi_1, \ldots, \xi_{n-1})$, ξ_i given by (5.1).

Changing the first and last pole. If we want to change ξ_1 to another value ϱ, we consider the vector:

$$\boldsymbol{x} = \gamma(A - \varrho U)(A - \xi_1 U)^{-1}\boldsymbol{e}_1 = \hat{\gamma}(A - \varrho U)\boldsymbol{e}_1 = \hat{\gamma}(A - \varrho C_1)\boldsymbol{e}_1, \tag{5.2}$$

with $\gamma, \hat{\gamma}$ some scalars. The second equality follows from the observation that $(A - \xi_1 U)\boldsymbol{e}_1 = \beta \boldsymbol{e}_1$. The third equality follows from $C_i \boldsymbol{e}_1 = \boldsymbol{e}_1$ if $i > 1$. Notice that only the elements in the first two rows of $\boldsymbol{x}$ contain nonzero elements. The next step is to compute a core transformation that introduces a zero in the second row of $\boldsymbol{x}$, $Q_1^* \boldsymbol{x} = \alpha \boldsymbol{e}_1$. The pencil $(\hat{A}, \hat{U}) = Q_1^*(A, U)$ is equivalent to (A, U) but with its first pole replaced by ϱ. We remark that $\hat{U} = Q_1^* U$ can be computed by the *fusion* $\hat{C}_1 = Q_1^* C_1$ of two core transformations in $O(1)$ operations. More details on the fusion operation are provided in Appendix A.1.

The last pole ξ_{n-1} of (A, U) can be changed to another value ϱ using a similar strategy. Consider the row vector:

$$\boldsymbol{y}^T = \gamma \boldsymbol{e}_n^T (A - \xi_{n-1} U)^{-1}(A - \varrho U) = \hat{\gamma} \boldsymbol{e}_n^T (A - \varrho U) = \hat{\gamma} \boldsymbol{e}_n^T (A - \varrho C_{n-1}).$$

The second equality follows from $\boldsymbol{e}_n^T (A - \xi_{n-1} U) = \beta \boldsymbol{e}_n^T$, and the third equality from $\boldsymbol{e}_n^T C_i = \boldsymbol{e}_n^T$ if $i < n-1$. We get that $\boldsymbol{y}^T$ is a row vector with only nonzero

elements in the last two columns and can thus compute a core transformation Z_{n-1} such that $\boldsymbol{y}^T Z_{n-1} = \alpha \boldsymbol{e}_n^T$. The last pole is then replaced by ϱ in the equivalent pencil $(A, U)Z_{n-1}$. The unitary matrix can again be updated by a fusion operation $\hat{C}_{n-1} = C_{n-1}Z_{n-1}$ in $O(1)$ operations.

Swapping poles. To swap two consecutive poles ξ_i and ξ_{i+1} in (A, U), we devise a numerically stable algorithm based on Lemma 3.3.2 and Table 3.1. We consider the 2×2 upper triangular subpencil positioned at rows $i+1$ and $i+2$ and columns i and $i+1$ of (A, U) which is if the form:

$$(\breve{A}, \breve{U}) = \left(\begin{bmatrix} \alpha_1 & a \\ & \alpha_2 \end{bmatrix}, \begin{bmatrix} s_1 & \bar{c}_1 c_2 \\ & s_2 \end{bmatrix} \right), \text{ with } \begin{cases} |c_1|^2 + s_1^2 = 1 \\ |c_2|^2 + s_2^2 = 1 \end{cases}. \tag{5.3}$$

The scalars c_1 and s_1 determine C_i, while c_2 and s_2 determine C_{i+1}.

The eigenvalues of the pencil $(\breve{A}, \breve{U})$ need to be reordered to swap the poles in (A, U). For this we make use of the methods discussed in Section 3.3.2. The only additional requirement is that the representation of U as a product of core transformations is accurately preserved under the swapping operation. We use the *turnover* operation for a *V-shaped* pattern of core transformations for this. A turnover operation changes to V-shaped pattern of cores, $E_i F_{i+1} G_i$, acting on consecutive rows i and $i+1$, to a *hat-shaped* pattern or vice versa:

$$E_i F_{i+1} G_i = \hat{E}_{i+1} \hat{F}_i \hat{G}_{i+1}.$$

The turnover operation requires $O(1)$ operations and is backward stable [4] in the sense that the in finite precision arithmetic it computes $E_i F_{i+1} G_i = \hat{E}_{i+1} \hat{F}_i \hat{G}_{i+1} + \Delta$ with $\|\Delta\|_2 \leq \epsilon_m$. More details on the turnover operation are provided in Appendix A.1.

Case I If $|\xi_i| \geq |\xi_{i+1}|$, then, according to Table 3.1, we can use either method 1.A or 2.A to compute a backward stable swap. Since we want to use the turnover operation to compute the second transformation, we must choose method 1.A in this case.

The procedure goes as follows. We first compute a rotation Z_i to introduce a zero in position $(1, 1)$ of H_1 according to (3.4),

$$\boldsymbol{e}_1^* H_1 Z_i = \boldsymbol{e}_1^* (s_2 \breve{A} - \alpha_2 \breve{U}) Z_i = [0 \times].$$

Next, this rotation is introduced in the pattern of cores of the$U$ matrix. It commutes with all core transformations C_j, $|i - j| > 1$ such that the relevant part of the pattern is $C_i C_{i+1} Z_i$. This can undergo a (backward stable) turnover

to $Q_{i+1}\hat{C}\hat{C}_iC_{i+1}$. The core Q_{i+1} can be removed up to machine precision by a fusion from the left with Q_{i+1}^*.

From the discussion in Section 3.3.2 we conclude that $|\boldsymbol{e}_2^*Q_{i+1}^*\breve{A}Z_i\boldsymbol{e}_1| \leq c\epsilon_m\|\breve{A}\|_2$. A similar bound holds for $\breve{U}$ based on the backward stability of the turnover.

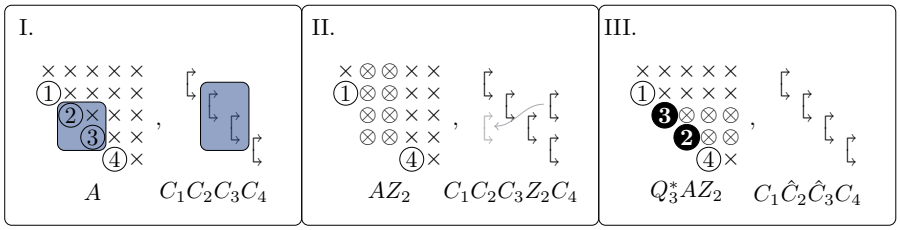


Figure 5.1: First backward stable strategy to perform a pole swap if $|\xi_i| \geq |\xi_{i+1}|$.

The first method is illustrated pictorially in Figure 5.1 for a 5×5 pencil where the second and third pole undergo a swapping transformation. The relevant part of A and the core transformations which are used to form H_1 and compute Z_2 are indicated in pane I. In pane II, Z_2 is applied to A and introduced in U. This shows the V-shaped pattern of $C_2C_3Z_2$, the turnover operation is indicated with the gray arrow and core transformation Q_3 that appears on the other side of the pattern. This core transformation is removed in pane III where also the rows of A are updated to maintain the equivalence. This completes the swap.

Case II If $|\xi_i| < |\xi_{i+1}|$, then we can use method 2.B from Table 3.1 in combination with the turnover operation. We compute Q_{i+1} according to (3.5) such that introduces a zero in position $(2,2)$ of H_2,

$$Q_{i+1}^*H_2\boldsymbol{e}_2 = Q_{i+1}^*(s_1\breve{A} - \alpha_1\breve{U})\boldsymbol{e}_2 = [\times\ 0]^T\,.$$

This rotation is introduced in the pattern of U, where the relevant part now is the hat-shaped pattern $Q_{i+1}^*C_iC_{i+1}$ and Z_i is retrieved from the turnover of this pattern. This process is summarized pictorially in Figure 5.2.

5.3 Computational cost

The rational QZ step for Hessenberg, unitary Hessenberg pencils $(A, C_1 \ldots C_{n-1})$ proceeds entirely similar to the dense rational QZ step of Chapter 3 but with the adapted pole manipulation techniques from Section 5.2.1.

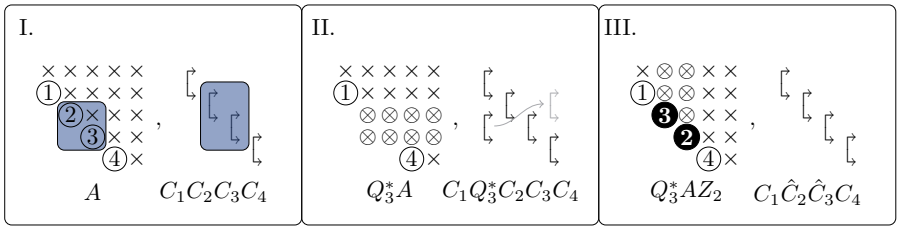


Figure 5.2: Second backward stable strategy to perform a pole swap if $|\xi_i| < |\xi_{i+1}|$.

As all pole manipulation techniques only use fusion and turnover operations, with an individual computational cost of $O(1)$, to update U, the computational cost of a single rational QZ step,

$$(\hat{A}, \hat{U} = \hat{C}_1 \dots \hat{C}_{n-1}) = Q^*(A, U = C_1 \dots C_{n-1})Z,$$

is $O(n)$ for $\hat{U}$ and $O(n^2)$ for $\hat{A}$. This is approximately half the cost of a dense rational QZ step for large n.

5.4 Numerical experiment

This section presents the results obtained with `ZLAHPS` in comparison with `ZLAHQR` from LAPACK [2]. All tests were performed on an Intel Xeon E5-2697 v3 CPU with 14 cores and 128GB of RAM using LAPACK 3.8.0 and BLAS 3.8.0 [2]. The code was compiled with *gfortran* version 4.8.5 using compilation flag `-O3`.

`ZLAHQR` is the lowest level routine in the collection of QR algorithms from LAPACK. It implements a single-shift bulge-chase method for complex Hessenberg matrices with a Wilkinson shifting strategy. This routine is called by the higher level methods, which implement cache blocking and aggressive early deflation (cfr. Chapter 4). It is used to compute the shifts, perform aggressive early deflation, or solve the problem if the dimension is smaller than the heuristic 150 [2].

`ZLAHPS` is an adaptation of `ZLAHQR` which uses a pole swapping algorithm for Hessenberg, unitary Hessenberg pencils as described in this chapter. It uses a Wilkinson shift strategy and a Wilkinson pole strategy. The turnover, fusion and rotation generator routines used in `ZLAHPS` are adapted from EISCOR [4]. All rotations for the U matrix are represented with 3 real variables: the real and imaginary part of the cosine, and a real sine.

We ran both routines an a set of randomly generated matrices of dimension 5 up to 1000 that are first reduced to Hessenberg form via similarity transformations. The CPU time and the relative backward error, $\|Q^*HQ - T\|_2/\|H\|_2$, are summarized in respectively Table 5.1 and Table 5.2.

Table 5.1: Comparison between `ZLAHQR` from LAPACK and `ZLAHPS` in terms of CPU time for random matrices reduced to Hessenberg form. The first column states the problem size, the second column the number of runs over which the results have been averaged, the third column shows the CPU time of `ZLAHQR`, the fourth column shows the same data for `ZLAHPS`, and the fifth column shows the runtime of `ZLAHPS` relative to `ZLAHQR`.

n	# runs	`ZLAHQR` $t_{CPU}(s)$	`ZLAHPS` $t_{CPU}(s)$	% t_{CPU}
5	1000	$1.4 \cdot 10^{-5}$	$1.4 \cdot 10^{-5}$	100%
10	1000	$6.0 \cdot 10^{-5}$	$5.6 \cdot 10^{-5}$	93%
20	500	$2.8 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$	86%
40	250	$1.6 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	81%
80	125	$1.0 \cdot 10^{-2}$	$7.4 \cdot 10^{-3}$	74%
150	80	$6.3 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	71%
300	80	$5.0 \cdot 10^{-1}$	$3.2 \cdot 10^{-1}$	64%
600	40	$3.6 \cdot 10^{0}$	$2.3 \cdot 10^{0}$	64%
1000	40	$1.6 \cdot 10^{1}$	$1.0 \cdot 10^{1}$	63%

The fifth column of Table 5.1 shows the CPU time of `ZLAHPS` relative to `ZLAHQR`. We observe reductions up to 29% for problems of dimension not greater than 150 – the typical use case – and, except for the smallest problem size of 5, `ZLAHPS` is always faster than `ZLAHQR`. The speedup increases to up to 37% for problems of dimension 1000.

Table 5.2 shows the same comparison in terms of the accuracy of the Schur decomposition. We observe that `ZLAHPS` is always more accurate than `ZLAHQR` which can at least partially be attributed to the reduction in number of operations by using the rational iteration instead of the polynomial one.

5.5 Conclusion

This chapter proposed a specification of the rational QZ method for Hessenberg, unitary Hessenberg matrix pairs which requires approximately half the storage space and computational cost in comparison with the dense rational QZ method of Chapter 3. The method uses a representation of the unitary matrix in terms

Table 5.2: Same comparison between `ZLAHQR` from LAPACK and `ZLAHPS` as in Table 5.1 but in terms of relative backward error on the Schur decomposition.

n	# runs	`ZLAHQR` BWE	`ZLAHPS` BWE
5	1000	$1.8 \cdot 10^{-15}$	$1.2 \cdot 10^{-15}$
10	1000	$1.8 \cdot 10^{-15}$	$1.4 \cdot 10^{-15}$
20	500	$2.8 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$
40	250	$3.7 \cdot 10^{-15}$	$2.6 \cdot 10^{-15}$
80	125	$5.7 \cdot 10^{-15}$	$3.7 \cdot 10^{-15}$
150	80	$7.8 \cdot 10^{-15}$	$4.9 \cdot 10^{-15}$
300	80	$1.1 \cdot 10^{-14}$	$6.9 \cdot 10^{-15}$
600	40	$1.5 \cdot 10^{-14}$	$9.5 \cdot 10^{-15}$
1000	40	$2.0 \cdot 10^{-14}$	$1.2 \cdot 10^{-14}$

of core transformations. We have seen how the pole manipulation techniques can be adapted to this format such that the representation is accurately preserved throughout the algorithm and the resulting method is backward stable.

The main objective of this format is to enable the use of a pole swapping algorithm for the standard, unsymmetric eigenvalue problem whilst the storage requirements and computational cost remains on par with the QR algorithm. This method is considered as a rational QR algorithm.

Numerical experiments compared our rational QR algorithm, `ZLAHPS`, with `ZLAHQR` from LAPACK and showed a significant reduction in compute time in combination with a more accurate result in favor of our method.

Further research is required to devise a way to include multishift, multipole rational QZ steps and aggressive early deflation techniques within this proposed format. Our experiments confirm that this has the potential to be competitive with state-of-the-art eigenvalue solvers for the dense, standard eigenvalue problem if blocking for cache efficiency is used.

Chapter 6

Two-sided pole swapping for tridiagonal pencils

This chapter studies the pole swapping method for tridiagonal matrix pencils. The content is based on:

Camps D., Vandebril R., and Van Dooren P., Two-sided rational iterations for tridiagonal pencils. In preparation.

6.1 Introduction

In this chapter we study pole swapping methods for regular, block tridiagonal matrix pencils that preserve the block tridiagonal structure of the matrix pencil throughout the iteration. Non-unitary equivalence transformations in lower and upper triangular form are well-suited for this task as they allow us to locally alter the poles of the pencil without perturbing the tridiagonal structure. We propose to use (nearly) optimally scaled lower and upper triangular transformations. Backward stability of the algorithm is unfortunately not assured unlike in the unitary case.

The proposed algorithm has two major potential advantages over the unitary algorithms of previous chapters. Firstly, thanks to the sparsity of the pencil the computational complexity of the algorithm is only $O(n^2)$ if the equivalence

transformations are not accumulated. Secondly, as we will see, tridiagonal pencils have poles on both the subdiagonal and the superdiagonal and these poles can be independently used for a pole swapping step. This gives additional degrees of freedom for novel shifting strategies.

We pay special attention to symmetric pencils that can be diagonalized by congruence transformations. A sufficient but not a necessary condition for diagonalizability of a symmetric pencil is definiteness [89]. Furthermore, any dense, symmetric pencil can be reduced in $O(n^3)$ operations to symmetric tridiagonal form [42, 108] which admits the use of our algorithms.

The symmetric generalized eigenvalue problem with definite B is often solved using a hybrid method which combines the Cholesky factorization $B = LL^T$ with the symmetric QR method for the standard eigenvalue problem defined by the matrix $L^{-1}AL^{-T}$. This approach was first formulated by Wilkinson [141, Section 71]. It is advised to use a Cholesky factorization with pivoting to improve stability of the algorithm [26]. This method has some disadvantages. Firstly, if B is ill-conditioned, then the computed eigenvalues may be inaccurate [46]. Secondly, the matrix $L^{-1}AL^{-T}$ becomes, in general, a full matrix. Nonetheless, $L^{-1}AL^{-T}$ has a *quasi-separable* structure in case $A - \lambda B$ is tridiagonal which can be exploited to devise an improved $O(n^2)$ eigenvalue solver [124].

Where the rational QZ and QR method of the previous chapters can be viewed as methods that are derived from the QR algorithm [39, 40], the non-unitary methods discussed in the current chapter perhaps compare more to the LR algorithm [101]. For this reason, we refer to the method as a *rational LR* algorithm for tridiagonal matrix pairs.

6.2 Tridiagonal pencils

Definition 6.2.1. An $n \times n$ matrix pair (A, B) is said to be a tridiagonal matrix pair with respectively *lower* and *upper* pole tuples,

$$\Xi(A, B) = (\xi_1, \ldots, \xi_{n-1}), \qquad \text{and,} \qquad \Psi(A, B) = (\psi_1, \ldots, \psi_{n-1}), \tag{6.1}$$

if both A and B are tridiagonal matrices and if:

$$\xi_i = \frac{a_{i+1,i}}{b_{i+1,i}} \in \bar{\mathbb{C}}, \quad \text{and} \quad \psi_i = \frac{a_{i,i+1}}{b_{i,i+1}} \in \bar{\mathbb{C}} \quad \text{for all} \quad i \in \{1, \ldots, n-1\}.$$

The definition of the lower pole tuple $\Xi(A, B)$ is compatible with Definition 3.2.1. Furthermore, it is obvious that $\Psi(A, B) = \Xi(A^T, B^T)$. We illustrate the previous definition with a small-scale example.

Example 6.2.2. The 3×3 real, unsymmetric, tridiagonal matrix pair,

$$A = \begin{bmatrix} 1 & 2 & \\ 3 & 2 & 9 \\ & 4 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 2 & \\ 6 & 1 & 3 \\ & 1 & 3 \end{bmatrix}, \tag{6.2}$$

has lower pole tuple $\Xi(A, B) = (0.5, 4)$, upper pole tuple $\Psi(A, B) = (1, 3)$, and eigenvalues $\Lambda(A, B) = \{0.5, 1.33 \pm \imath 0.94\}$

The previous example illustrates that real, tridiagonal matrix pairs can have complex-conjugate eigenvalues. To develop an eigenvalue algorithm using real arithmetic for real-valued matrix pairs, we allow for a relaxation of Definition 6.2.1 similar to the relaxation of block Hessenberg pencils in Definition 4.2.3. For the same numerical reasons as discussed in Section 4.4, we restrict ourselves to 2×2 blocks for complex-conjugate poles and 1×1 blocks for real poles.

Definition 6.2.3. A matrix pair $(A, B) \in \mathbb{R}^{n\times n}$ is called a *block tridiagonal* matrix pair if it can be simultaneously partitioned as:

$$\begin{array}{c|c|c} n-1 & 1 & \\ \hline \boldsymbol{a}_{11}^T - \lambda \boldsymbol{b}_{11}^T & a_{12} - \lambda b_{12} & 1 \\ \hline A_{21} - \lambda B_{21} & \boldsymbol{a}_{22} - \lambda \boldsymbol{b}_{22} & n-1 \end{array}, \quad \text{and} \quad \begin{array}{c|c|c} 1 & n-1 & \\ \hline \hat{\boldsymbol{a}}_{11} - \lambda \hat{\boldsymbol{b}}_{11} & \hat{A}_{12} - \lambda \hat{B}_{12} & n-1 \\ \hline \hat{a}_{21} - \lambda \hat{b}_{21} & \hat{\boldsymbol{a}}_{22}^T - \lambda \hat{\boldsymbol{b}}_{22}^T & 1 \end{array}, \tag{6.3}$$

and if the $n-1 \times n-1$ pencils $A_{21} - \lambda B_{21}$ and $\hat{A}_{12}^T - \lambda \hat{B}_{12}^T$ are in real, generalized Schur form (2.13). The lower pole tuple is defined as $\Xi(A, B) = \Lambda(A_{21}, B_{21})$ and the upper pole tuple as $\Psi(A, B) = \Lambda(\hat{A}_{21}, \hat{B}_{21})$.

Observe that Definition 6.2.3 implies that a block tridiagonal pencil is a *pentadiagonal* pencil. The last ingredient required for (block) tridiagonal pencils is a definition of properness.

Definition 6.2.4. A (block) tridiagonal matrix pair (A, B) is called *proper* or *irreducible* if both (A, B) and (A^T, B^T) are proper (block) Hessenberg pairs according to Definition 4.2.5.

The example tridiagonal pair in (6.2) is not proper as the first column of A is a scalar multiple of the first column of B. This implies that ξ_1 is an eigenvalue of (A, B) with right eigenvector $\boldsymbol{e}_1$.

In case of a symmetric, (block) tridiagonal matrix pair, $(A, B) = (A^T, B^T)$, we have that $\Xi(A, B) = \Psi(A, B)$.

The following lemma shows that if a block tridiagonal pair undergoes an equivalence transformations with upper triangular matrices the result is a block

upper Hessenberg pair. Lower triangular equivalences lead to block *lower* Hessenberg matrix pairs. A pair (A, B) is said to be a lower block Hessenberg pair if (A^T, B^T) is an upper block Hessenberg pair.

Lemma 6.2.5. *Let (A, B) be an $n \times n$ proper block tridiagonal matrix pair with lower pole tuple $\Xi(A, B)$ and upper pole tuple $\Psi(A, B)$. If R, $\check{R}$ are a pair of $n \times n$ nonsingular upper triangular matrices, then $R(A, B)\check{R}$ is a proper, upper block Hessenberg pair with pole tuple $\Xi(A, B)$. If L, $\check{L}$ are a pair of $n \times n$ nonsingular lower triangular matrices, then $L(A, B)\check{L}$ is a proper, lower block Hessenberg pair with pole tuple $\Psi(A, B)$.*

Proof. We can rewrite $R(A, B)\check{R}$ using the first partitioning of (6.3) and a block partitioning of the upper triangulars R and $\check{R}$ as follows:

$$\left(\begin{array}{c|c|c} 1 & n-1 & \\ \hline \begin{bmatrix} r_{11} \\ 0 \end{bmatrix} & \begin{bmatrix} \boldsymbol{r}_{12}^T \\ R_{22} \end{bmatrix} & \begin{matrix} 1 \\ n-1 \end{matrix} \end{array}\right) \left(\begin{array}{c|c|c} n-1 & 1 & \\ \hline \begin{bmatrix} \boldsymbol{a}_{11}^T - \lambda \boldsymbol{b}_{11}^T \\ A_{21} - \lambda B_{21} \end{bmatrix} & \begin{bmatrix} a_{12} - \lambda b_{12} \\ \boldsymbol{a}_{22} - \lambda \boldsymbol{b}_{22} \end{bmatrix} & \begin{matrix} 1 \\ n-1 \end{matrix} \end{array}\right) \left(\begin{array}{c|c|c} n-1 & 1 & \\ \hline \begin{bmatrix} \check{R}_{11} \\ 0 \end{bmatrix} & \begin{bmatrix} \check{\boldsymbol{r}}_{12} \\ r_{22} \end{bmatrix} & \begin{matrix} n-1 \\ 1 \end{matrix} \end{array}\right).$$

The $(n-1) \times (n-1)$ pole pencil in position $(2, 1)$ of $R(A, B)\check{R}$ above is thus given by $R_{22}(A_{21} - \lambda B_{21})\check{R}_{11}$. This is an equivalence transformation of a block upper triangular pencil in generalized Schur form (2.13) with nonsingular, upper triangular matrices. This preserves the block upper triangular structure and the ordering of the eigenvalues of the pole pencil. The second partitioning of (6.3) cannot be preserved by an upper triangular equivalence. This proves the first result. The proof of the second result proceeds similarly by considering a block partitioning of L, $\check{L}$ which preserves the second partitioning of (6.3) but not the first. □

6.3 Swapping poles on the subdiagonal

In this section we discuss how two consecutive poles on the subdiagonal of a block tridiagonal pencil $A - \lambda B$ can be swapped without altering the block tridiagonal structure in the pencil and without changing the upper poles $\Psi(A, B)$. We limit ourselves to the subdiagonal poles without loss of generality as reordering superdiagonal poles of $A - \lambda B$ is equivalent to reordering subdiagonal poles of $A^T - \lambda B^T$.

To this end we consider the pencil $A - \lambda B$ which is block upper triangular and has two sub-pencils $A_{ii} - \lambda B_{ii}$ of dimensions $n_i \times n_i, i = 1, 2$ with disjoint

spectra :

$$A - \lambda B := \begin{bmatrix} A_{11} - \lambda B_{11} & A_{12} - \lambda B_{12} \\ 0 & A_{22} - \lambda B_{22} \end{bmatrix}, \quad \Lambda(A_{11}, B_{11}) \cap \Lambda(A_{22}, B_{22}) = \emptyset. \tag{6.4}$$

As we have seen previously in Section 4.4, there always exist invertible equivalence transformations that permute the spectra of the two diagonal blocks:

$$S(A-\lambda B)T = \hat{A}-\lambda\hat{B} := \begin{bmatrix} \hat{A}_{11} - \lambda\hat{B}_{11} & \hat{A}_{12} - \lambda\hat{B}_{12} \\ 0 & \hat{A}_{22} - \lambda\hat{B}_{22} \end{bmatrix}, \ \det S \neq 0, \ \det T \neq 0, \tag{6.5}$$

where the spectra (and sizes) of $\lambda B_{11} - A_{11}$ and $\lambda\hat{B}_{22} - \hat{A}_{22}$ are equal, and those of $\lambda B_{22} - A_{22}$ and $\lambda\hat{B}_{11} - \hat{A}_{11}$ are equal.

The left and right deflating subspaces of $A-\lambda B$ are well defined by Lemma 4.4.1 [57]. The following lemma provides the required conditions which an *invertible* equivalence transformation (S, T), must satisfy to get (6.5).

Lemma 6.3.1. *Let the pencil $A - \lambda B$ be given as in* (6.4)*. The invertible equivalence transformations S and T swap the spectra of the diagonal blocks in $S(A - \lambda B)T$ if and only if the first block column of T and the last block row of S are spanning the right and left deflating subspaces described in Lemma 4.4.1, i.e. :*

$$\begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix} = T \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix} M_T, \quad \text{and} \quad M_S \begin{bmatrix} 0 & I_{n_1} \end{bmatrix} S = \begin{bmatrix} I_{n_1} & X \end{bmatrix}, \tag{6.6}$$

where M_T and M_S are square invertible matrices.

Proof. The right hand side of (6.5) implies that the column space of $\begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}$ is a right deflating subspace of the pencil $\hat{A} - \lambda\hat{B}$ with spectrum $\Lambda(\hat{A}_{11}, \hat{B}_{11}) = \Lambda(A_{22}, B_{22})$ and that the row space of $\begin{bmatrix} 0 & I_{n_1} \end{bmatrix}$ is a left deflating subspace of the pencil $\hat{A} - \lambda\hat{B}$ with spectrum $\Lambda(\hat{A}_{22}, \hat{B}_{22}) = \Lambda(A_{11}, B_{11})$. The equality with the left hand side then implies that the column space of $T \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix}$ and the row space of $\begin{bmatrix} 0 & I_{n_1} \end{bmatrix} S$ span, respectively, the corresponding right and left deflating subspaces of $A - \lambda B$. Lemma 4.4.1 says these are also spanned, respectively, by $\begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix}$ and $\begin{bmatrix} I_{n_1} & X \end{bmatrix}$. They must therefore be related by invertible basis transformations M_T and M_S. □

Lemma 4.4.1 also provides a direct manner to compute the swapping equivalences S and T if they are chosen as unitary in the form of (4.25). Lemma 4.4.4 provides an error bound for this case.

We cannot rely on unitary equivalences to preserve the block tridiagonal structure of the pencil. If, in the current setting, we use an equivalence transformation in the class of invertible lower triangular matrices, we do preserve the block tridiagonal structure according to Lemma 6.2.5.

In other words, we need to construct invertible lower triangular matrices L_X and L_Y such that

$$\begin{bmatrix} I_{n_1} & X \end{bmatrix} = \begin{bmatrix} 0 & M_X \end{bmatrix} L_X, \quad \text{and} \quad \begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix} = L_Y \begin{bmatrix} M_Y \\ 0 \end{bmatrix}, \tag{6.7}$$

where the matrices M_X and M_Y are invertible.

Lemma 6.3.2. *Let the pencil $A - \lambda B$ be given as in* (6.4)*, where the spectra of $A_{11} - \lambda B_{11}$ and $A_{22} - \lambda B_{22}$ are disjoint. Then there exist lower triangular equivalence transformations L_X and L_Y that swap the spectra of the diagonal blocks in the transformed pencil $L_X(A - \lambda B)L_Y$ if and only if the matrices*

$$\begin{bmatrix} I_{n_1} & X \end{bmatrix} \begin{bmatrix} 0 \\ I_{n_1} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix}, \tag{6.8}$$

are invertible.

Proof. We first prove the rank conditions are necessary. The leading $n_2 \times n_2$ block of $\begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix} = L_Y \begin{bmatrix} M_Y \\ 0 \end{bmatrix}$ is the matrix

$$\begin{bmatrix} I_{n_2} & 0 \end{bmatrix} \begin{bmatrix} -Y \\ I_{n_2} \end{bmatrix} = \left(\begin{bmatrix} I_{n_2} & 0 \end{bmatrix} L_Y \begin{bmatrix} I_{n_2} \\ 0 \end{bmatrix} \right) M_Y \tag{6.9}$$

which must be invertible if M_Y is invertible and L_Y is lower triangular and invertible. Similarly, the trailing $n_1 \times n_1$ block of $\begin{bmatrix} I_{n_1} & X \end{bmatrix} = \begin{bmatrix} 0 & M_X \end{bmatrix} L_X$ is the matrix

$$\begin{bmatrix} I_{n_1} & X \end{bmatrix} \begin{bmatrix} 0 \\ I_{n_1} \end{bmatrix} = M_X \left(\begin{bmatrix} 0 & I_{n_1} \end{bmatrix} L_X \begin{bmatrix} 0 \\ I_{n_1} \end{bmatrix} \right) \tag{6.10}$$

which must be invertible if M_X is invertible and L_X is lower triangular and invertible. We now prove the conditions are also sufficient. If the matrices (6.8) are invertible, then so are the right hand sides of (6.9) and (6.10). We can then choose M_Y such that the $(1,1)$-block of L_Y is lower triangular and invertible and M_X such that the $(2,2)$-block of L_X is lower triangular and invertible, e.g.,

by using QR factorizations of the matrices (6.8). Since the remaining blocks of L_Y and L_X are free to choose, we can complete these matrices to be lower triangular and invertible, e.g. by choosing the $(2,2)$ block of L_Y to be I_{n_1} and the $(1,1)$ block of L_X to be I_{n_2}, i.e. :

$$L_X := \begin{bmatrix} I_{n_2} & 0 \\ M_X^{-1} & M_X^{-1}X \end{bmatrix}, \quad L_Y := \begin{bmatrix} -YM_Y^{-1} & 0 \\ M_Y^{-1} & I_{n_1} \end{bmatrix}. \tag{6.11}$$

□

6.3.1 Diagonal scaling of transformations

The spectral transformations described above are clearly not unique and since they are not orthogonal, we may ask the question what are the best transformations in terms of numerical stability to perform the swapping. Since the transformations L_X and L_Y are constrained to be lower triangular there is still a degree of freedom to scale them as follows :

$$D_X L_X, \quad \text{and} \quad L_Y D_Y$$

where D_X and D_Y are real block diagonal matrices, with two blocks of dimensions $n_2 \times n_2$ and $n_1 \times n_1$. Clearly, these additional diagonal scalings do not affect the block structure and spectral properties of the transformed pencil $\hat{A} - \lambda\hat{B}$.

We recall two theorems proven in [63, Theorem 8] and [8, Theorem 3.7], that address the one-sided scaling problem.

Theorem 6.3.3 ([63]). *Let $\hat{L}_X \in \mathbb{R}^{k\times k}$ and $\hat{L}_Y \in \mathbb{R}^{k\times k}$ have, respectively, equal row and column norms, then they are nearly optimally scaled in the sense that*

$$\kappa(\hat{L}_X) \leq \sqrt{k} \min_{D_X} \kappa(D_X L_X), \quad \text{and} \quad \kappa(\hat{L}_Y) \leq \sqrt{k} \min_{D_Y} \kappa(L_Y D_Y).$$

Theorem 6.3.4 ([8]). *Let $\hat{L}_X \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ and $\hat{L}_Y \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ have, respectively, two orthogonal block rows L_{X_1} and L_{X_2} and two orthogonal block columns L_{Y_1} and L_{Y_2} of dimensions n_1 and n_2, then they are optimally scaled under block diagonal transformations $D_X = diag\{D_{X_2}, D_{X_1}\}$ and $D_Y = diag\{D_{Y_2}, D_{Y_1}\}$, where $D_{X_i}, D_{Y_i} \in \mathbb{R}^{n_i\times n_i}$ for $i = 1, 2$:*

$$\kappa(\hat{L}_X) = \min_{D_X} \kappa(D_X L_X), \quad \text{and} \quad \kappa(\hat{L}_Y) = \min_{D_Y} \kappa(L_Y D_Y).$$

Moreover,

$$\kappa(\hat{L}_X) = \frac{1+\cos(\phi_X)}{\sin(\phi_X)}, \quad \text{and} \quad \kappa(\hat{L}_Y) = \frac{1+\cos(\phi_Y)}{\sin(\phi_Y)},$$

where ϕ_X and ϕ_Y are the minimal angles between the spaces spanned by the images of L_{X_1} and L_{X_2} and of L_{Y_1} and L_{Y_2}, respectively.

We point out that in the proposed construction (6.11), the first block row of L_X and the last block column of L_Y are already orthogonal and even orthonormal. The additional scalings $D_X = \mathrm{diag}\{I_{n_2}, N_X\}$ and $D_Y = \mathrm{diag}\{N_Y, I_{n_1}\}$ will maintain the lower triangular structure in

$$L_X := \begin{bmatrix} I_{n_2} & 0 \\ N_X M_X^{-1} & N_X M_X^{-1} X \end{bmatrix}, \quad L_Y := \begin{bmatrix} -Y M_Y^{-1} N_Y & 0 \\ M_Y^{-1} N_Y & I_{n_1} \end{bmatrix},$$

and at the same time achieve the normalization constraint, provided we choose lower triangular solutions for N_X and N_Y in

$$M_X^{-1}(I_{n_1} + XX^T)M_X^{-T} = N_X^{-1}N_X^{-T}, \quad M_Y^{-T}(I_{n_2} + Y^TY)M_Y^{-1} = N_Y^{-T}N_Y^{-1},$$

or, equivalently,

$$M_X^T(I_{n_1} + XX^T)^{-1}M_X = N_X^T N_X, \quad M_Y(I_{n_2} + Y^TY)^{-1}M_Y^T = N_Y N_Y^T.$$

6.3.2 Iterative refinement

The *numerical* implementation of the transformations L_X and L_Y as described in the previous section, does *not* imply that the $(2,1)$-block in the transformed pencil

$$\hat{A} - \lambda\hat{B} = \begin{bmatrix} \hat{A}_{11} - \lambda\hat{B}_{11} & \hat{A}_{12} - \lambda\hat{B}_{12} \\ \Delta_A - \lambda\Delta_B & \hat{A}_{22} - \lambda\hat{B}_{22} \end{bmatrix}.$$

can be safely dismissed. Just like this is not guaranteed in the unitary case, cfr. Lemma 4.4.4.

Updating the equivalence transformation to further reduce (and then safely dismiss) the $n_2 \times n_1$ block $\Delta_A - \lambda\Delta_B$ in $\hat{A} - \lambda\hat{B}$ can be done via a similar method as in Section 4.4.2 but with lower triangular instead of unitary transformations. We are interested in an equivalence transformation of the form:

$$\begin{bmatrix} \check{A}_{11} - \lambda\hat{B}_{11} & \check{A}_{12} - \lambda\hat{B}_{12} \\ 0 & \check{A}_{22} - \lambda\hat{B}_{22} \end{bmatrix} := L_{X,up} \begin{bmatrix} \lambda\hat{A}_{11} - \hat{B}_{11} & \hat{A}_{12} - \lambda\hat{B}_{12} \\ \Delta_A - \lambda\Delta_B & \hat{A}_{22} - \lambda\hat{B}_{22} \end{bmatrix} L_{Y,up},$$

with,

$$L_{X,up} := \begin{bmatrix} I_{n_2} & 0 \\ 0 & N_X \end{bmatrix} \begin{bmatrix} I_{n_2} & 0 \\ X & I_{n_1} \end{bmatrix}, \quad L_{Y,up} := \begin{bmatrix} I_{n_2} & 0 \\ Y & I_{n_1} \end{bmatrix} \begin{bmatrix} N_Y & 0 \\ 0 & I_{n_1} \end{bmatrix},$$

where N_X and N_Y are lower triangular normalization factors to optimally scale the corresponding block row and block column of $L_{X,up}$ and $L_{Y,up}$ to be orthonormal, and where (X, Y) are computed from the system of quadratic matrix equations

$$\hat{\Delta}_A + \hat{A}_{22}Y + X\hat{A}_{11} + X\hat{A}_{12}Y = 0, \quad \hat{\Delta}_B + \hat{B}_{22}Y + X\hat{B}_{11} + X\hat{B}_{12}Y = 0.$$

These can be approximated by the system of linear equations

$$\hat{\Delta}_A + \hat{A}_{22}Y + X\hat{A}_{11} = 0, \quad \hat{\Delta}_B + \hat{B}_{22}Y + X\hat{B}_{11} = 0,$$

since $\|X\|_2$ and $\|Y\|_2$ are very small. The solution (X, Y) of this linear system can be computed using Kronecker products.

6.3.3 Special cases

Let us make this more explicit for the cases where n_1 and n_2 are either 1 or 2, since these are the cases we need in practice.

Case $n_1 = n_2 = 1$

The system of equations (4.22) for the pencil

$$A - \lambda B := \begin{bmatrix} a_{11} - \lambda b_{11} & a_{12} - \lambda b_{12} \\ 0 & a_{22} - \lambda b_{22} \end{bmatrix}$$

becomes

$$\begin{bmatrix} a_{11} & -a_{22} \\ b_{11} & -b_{22} \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} a_{12} \\ b_{12} \end{bmatrix}.$$

Since

$$d := \det \begin{bmatrix} a_{11} & -a_{22} \\ b_{11} & -b_{22} \end{bmatrix} \neq 0,$$

this system has the (unique) solution

$$x = \det \begin{bmatrix} a_{11} & a_{12} \\ b_{11} & b_{12} \end{bmatrix} / d, \quad \text{and} \quad y = \det \begin{bmatrix} a_{12} & -a_{22} \\ b_{12} & -b_{22} \end{bmatrix} / d.$$

The conditions of Lemma 6.3.2 become $x \neq 0$ and $y \neq 0$ and according to Theorem 6.3.4, optimally scaled matrices are given by

$$L_X = \begin{bmatrix} 1 & 0 \\ 0 & (1+x^2)^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & x \end{bmatrix} \quad \text{and} \quad L_Y = \begin{bmatrix} -y & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} (1+y^2)^{-\frac{1}{2}} & 0 \\ 0 & 1 \end{bmatrix}.$$

Remark 6.3.5. The solvability conditions that x and y must be different from 0 can also be interpreted in terms of the data :

$$x \neq 0 \;\Leftrightarrow\; \Lambda(a_{11}, b_{11}) \neq \Lambda(a_{12}, b_{12}), \quad y \neq 0 \;\Leftrightarrow\; \Lambda(a_{22}, b_{22}) \neq \Lambda(a_{12}, b_{12}).$$

Case $n_1 = 2, n_2 = 1$

The system of equations (4.22) for the pencil

$$A - \lambda B := \begin{bmatrix} A_{11} - \lambda B_{11} & \mathbf{a}_{12} - \lambda \mathbf{b}_{12} \\ 0 & a_{22} - \lambda b_{22} \end{bmatrix}$$

now involves two 2×1 vectors $\mathbf{x}$ and $\mathbf{y}$ that are a solution of

$$\begin{bmatrix} A_{11} & -a_{22} I_2 \\ B_{11} & -b_{22} I_2 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{12} \\ \mathbf{b}_{12} \end{bmatrix}.$$

The conditions of Lemma 6.3.2 become $y_1 \neq 0$ and $\det \begin{bmatrix} 0 & x_1 \\ 1 & x_2 \end{bmatrix} = -x_1 \neq 0$ and according to Theorem 6.3.4, nearly optimally scaled matrices are given by:

$$L_X = \left[\begin{array}{c|cc} 1 & \mathbf{0} \\ \hline \mathbf{0} & N_X \end{array}\right] \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline -x_2 & y_1 & 0 \\ 1 & 0 & x_1 \end{array}\right]$$

where the lower triangular scaling matrix N_X orthonormalizes the bottom block row of L_X, and

$$L_Y = \left[\begin{array}{cc|cc} -y_1 & 0 & 0 \\ -y_2 & 1 & 0 \\ \hline 1 & 0 & 1 \end{array}\right] \left[\begin{array}{cc|cc} (1 + y_1^2 + y_2^2)^{-\frac{1}{2}} & 0 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right].$$

Case $n_1 = 1, n_2 = 2$

This case is dual to the previous case and is skipped for the sake of brevity.

Case $n_1 = n_2 = 2$

The system of equations (4.22) for the pencil

$$A - \lambda B := \begin{bmatrix} A_{11} - \lambda B_{11} & A_{12} - \lambda B_{12} \\ 0 & A_{22} - \lambda B_{22} \end{bmatrix}$$

now involves two 2×2 matrices X and Y that can be solved using Kronecker products. The solvability conditions boil down to $\det X \neq 0$, and $\det Y \neq 0$. Nearly optimally scaled matrices can be obtained as follows. Let Q_X and Q_Y be orthogonal transformations such that $Q_X X$ and $Y Q_Y$ are lower triangular, then

$$L_X = \begin{bmatrix} I_2 & 0 \\ 0 & N_X \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ Q_X & Q_X X \end{bmatrix} \quad \text{and} \quad L_Y = \begin{bmatrix} -Y Q_Y & 0 \\ Q_Y & I_2 \end{bmatrix} \begin{bmatrix} N_Y & 0 \\ 0 & I_2 \end{bmatrix}$$

where N_X and N_Y both scale the corresponding column norms and row norms to 1.

6.4 Swapping poles in symmetric block tridiagonal pencils

In case $A - \lambda B$ is a symmetric, block tridiagonal pencil, it is natural to preserve the symmetry by considering congruence transformations $T(A - \lambda B)T^T$, where T is a nonsingular matrix, that simultaneously swap two consecutive poles on both the subdiagonal and superdiagonal.

To illustrate how this is done, we consider a 5×5 tridiagonal pencil

$$A - \lambda B = \left[\begin{array}{cc|ccc} a_{11} & a_{21} & & & \\ a_{21} & a_{22} & a_{32} & & \\ \hline & a_{32} & a_{33} & a_{43} & \\ & & a_{43} & a_{44} & a_{54} \\ \hline & & & a_{54} & a_{55} \end{array}\right] - \lambda \left[\begin{array}{cc|ccc} b_{11} & b_{21} & & & \\ b_{21} & b_{22} & b_{32} & & \\ \hline & b_{32} & b_{33} & b_{43} & \\ & & b_{43} & b_{44} & b_{54} \\ \hline & & & b_{54} & b_{55} \end{array}\right]$$

in which we want to swap the pole pencils $\lambda b_{32} - a_{32}$ and $\lambda b_{43} - a_{43}$. This is achieved by computing row and column transformation transformations

$$\begin{bmatrix} 1 & 0 \\ x_{21} & x_{22} \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} y_{11} & 0 \\ y_{21} & 1 \end{bmatrix},$$

applied to rows 3 and 4, and columns 2 and 3 respectively, and where the vectors $[x_{21}\ x_{22}]$ and $[y_{11}\ y_{21}]^T$ are computed as explained in Section 6.3.3 and normalized to row and column norm 1, respectively. This transformation swaps the subdiagonal poles. To preserve the symmetry in $A - \lambda B$ the transposed transformations are also applied to the pencil as they swap the same poles on the superdiagonal. Both transformations can be combined in the 3×3 row and column transformations

$$T = \begin{bmatrix} y_{11} & y_{21} & 0 \\ 0 & 1 & 0 \\ 0 & x_{21} & x_{22} \end{bmatrix}, \quad \text{and} \quad T^T = \begin{bmatrix} y_{11} & 0 & 0 \\ y_{21} & 1 & x_{21} \\ 0 & 0 & x_{22} \end{bmatrix},$$

which moreover still have normalized rows and columns, respectively. According to Theorem 6.3.3, these combined transformations are therefore nearly optimally scaled.

It is also possible to combine pole swaps involving one or two 2×2 blocks into a congruence transformation on a symmetric, block tridiagonal pencil. We

consider the problem of swapping two 2×2 blocks in $A - \lambda B$. The other cases proceed similar. Below, we show a 6×6 symmetric pencil, where the diagonal blocks of the 5×5 sub-pencil are of dimension $(2, 2, 1)$:

$$\left[\begin{array}{ccccc|c} a_{11} & a_{21} & a_{31} & & & \\ a_{21} & a_{22} & a_{32} & & & \\ a_{31} & a_{32} & a_{33} & a_{43} & a_{53} & \\ & & a_{43} & a_{44} & a_{54} & \\ & & a_{53} & a_{54} & a_{55} & a_{65} \\ & & & & a_{65} & a_{66} \end{array}\right] - \lambda \left[\begin{array}{ccccc|c} b_{11} & b_{21} & b_{31} & & & \\ b_{21} & b_{22} & b_{32} & & & \\ b_{31} & b_{32} & b_{33} & b_{43} & b_{53} & \\ & & b_{43} & b_{44} & b_{54} & \\ & & b_{53} & b_{54} & b_{55} & b_{65} \\ & & & & b_{65} & b_{66} \end{array}\right].$$

The pencil is partitioned according to the first partitioning of (6.3). If we want to swap the spectra of the two 2-dimensional blocks on the subdiagonal, we must apply the lower 4×4 triangular row and column transformations:

$$\begin{bmatrix} I_2 & \\ X_{21} & X_{22} \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} Y_{11} & \\ Y_{21} & I_2 \end{bmatrix},$$

to rows 2 to 5 and columns 1 to 4, respectively. The matrices X_{22} and Y_{11} are invertible and lower triangular and the last block row and first block column make up the deflating subspaces of Lemma 4.4.1. To preserve the symmetry they are combined in a single 5×5 congruence transformation,

$$T = \begin{bmatrix} Y_{11}^T & Y_{21}^T & \\ & 1 & \\ & X_{21} & X_{22} \end{bmatrix},$$

which must be applied to rows 1 to 5, while its transpose is applied to columns 1 to 5. This implies that X_{22} and Y_{11} must be brought to lower triangular form by invertible upper triangular transformations R_X and R_Y,

$$R_X \begin{bmatrix} I & X \end{bmatrix} = \begin{bmatrix} X_{21} & X_{22} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -Y \\ I \end{bmatrix} R_Y = \begin{bmatrix} Y_{11} & Y_{21} \end{bmatrix},$$

to preserve the deflating subspaces in T. The congruence transformation T can be nearly optimal scaled by a diagonal scaling to equal row norms.

6.5 Pole introduction

Introducing poles in a block tridiagonal pencil $A - \lambda B$ is achieved in two steps that are similar to the (block) Hessenberg case in Chapter 4.

To change the first pole(s) on the subdiagonal, we first compute the vector $\boldsymbol{x}$ according to (4.14). Afterwards, we compute an invertible, lower triangular

matrix L_X such that $L_X \boldsymbol{x} = \gamma \boldsymbol{e}_1$. We can use Gaussian transformations for this. L_X can be nearly optimally scaled to equal row norm by a diagonal scaling. It follows from the argument provided in Chapter 4 in combination with Theorem 6.6.1 that the first subdiagonal poles in $L_X(A - \lambda B)$ have been changed accordingly. The tridiagonal structure and all other poles are preserved according to Lemma 6.2.5.

The last pole(s) on the subdiagonal can be changed in $(A - \lambda B)L_Y$ by a nearly optimally scaled, invertible, lower triangular matrix L_Y such that $\boldsymbol{x}^T L_Y = \gamma \boldsymbol{e}_n^T$ with $\boldsymbol{x}$ computed from (4.18).

Changing the first and last pole(s) on the superdiagonal is equivalent to respectively changing the first and last pole(s) on the subdiagonal of $A^T - \lambda B^T$.

Simultaneous introduction of the same first or last pole(s) in a symmetric, block tridiagonal pencil is simply achieved by respectively computing L_X or L_Y as described above and applying them as a congruence transformations:

$$L_X(A - \lambda B)L_X^T \quad \text{and} \quad L_Y^T(A - \lambda B)L_Y.$$

6.6 Rational LR and TT$^{\mathsf{T}}$ (T3) algorithms

Let $A - \lambda B$ be a proper (block) tridiagonal pencil with lower pole tuple $\Xi(A, B)$ and upper pole tuple $\Psi(A, B)$. A *lower* rational LR (RLR) sweep computes a lower triangular equivalence transformation,

$$\hat{A} - \lambda \hat{B} := L_X(A - \lambda B)L_Y, \tag{6.12}$$

based on the following three steps:

I. Select or compute some shift(s) and introduce them as the first subdiagonal pole(s) by computing $L_{X,in}(A - \lambda B)$ based on Section 6.5,

II. Swap these pole(s) with the methods of Section 6.3 along the subdiagonal to the final subdiagonal position in $L_{X,sw}L_{X,in}(A - \lambda B)L_{Y,sw}$,

III. Select or compute some pole(s) and introduce them as the final subdiagonal pole(s) by computing $L_{X,sw}L_{X,in}(A - \lambda B)L_{Y,sw}L_{Y,in}$ based on Section 6.5.

The final equivalence (6.12) is given by the accumulated lower triangular transformations $L_X := L_{X,sw}L_{X,in}$ and $L_Y := L_{Y,sw}L_{Y,in}$.

Similarly, an *upper* rational LR sweep computes an upper triangular equivalence transformation,

$$\hat{A} - \lambda \hat{B} := R_X(A - \lambda B)R_Y, \tag{6.13}$$

based on the same three steps applied to $A^T - \lambda B^T$ and with $R_X = L_Y^T$, $R_Y = L_X^T$.

In a practical implementation of an RLR-type algorithm it is possible to handle lower RLR sweeps to large extent in parallel with upper RLR sweeps. If the chasing procedures along the subdiagonal and superdiagonal are performed in opposite directions, they only influence each other when they cross each other. The same is true however for a bidirectional chase along the subdiagonal. At the moment of writing, it is still unclear if this interesting property of the RLR algorithm can exploited. A parallel implementation is also future work.

If the (block) tridiagonal pencil $A - \lambda B$ is symmetric and diagonalizable, it is natural to use a rational TT^T (RTT$^{\mathrm{T}}$ or RT3) sweep which computes a congruence transformation,

$$\hat{A} - \lambda \hat{B} := T(A - \lambda B)T^T, \tag{6.14}$$

based on the same three steps that now chase a shift simultaneously along the subdiagonal and superdiagonal making use of the symmetric swaps from Section 6.4.

The computational cost of a single lower RLR (6.12), upper RLR (6.13) or RTT$^{\mathrm{T}}$ (6.14) step is $O(n)$ if only $\hat{A} - \lambda \hat{B}$ computed. This follows from the observation that all pole introduction and swapping methods require $O(1)$ operations since they preserve the tridiagonal structure. If the equivalence transformations are accumulated throughout the algorithm, the cost of a single sweep increases to $O(n^2)$ as these involve row and column updates of dimension n. If the overall algorithm converges in $O(n)$ iterations, the total computational cost of computing the eigenvalues of $A - \lambda B$ is $O(n^2)$ or $O(n^3)$ if the equivalences are accumulated.

6.6.1 Uniqueness and convergence

In this section we provide some uniqueness and convergence results for a single shift lower and upper RLR sweep on a tridiagonal pencil based on the theory of Chapter 3. We omit the block generalization, but any proper block tridiagonal pencil can be reduced to a tridiagonal pencil via a similar procedure as Lemma 4.2.7 but with either lower or upper triangular equivalences.

Theorem 6.6.1 (Implicit LR theorem)**.** *Let $A - \lambda B$ be a proper tridiagonal pencil with lower pole tuple $\Xi(A, B)$, upper pole tuple $\Psi(A, B)$ and with all the poles different from the spectrum.*

Let $\hat{L}_X$, $\check{L}_X$, $\hat{L}_Y$ and $\check{L}_Y$ be invertible, lower triangular matrices with $\hat{L}_X^{-1}\boldsymbol{e}_1 = \alpha\check{L}_X^{-1}\boldsymbol{e}_1$, $\alpha \in \mathbb{C}\setminus\{0\}$, such that

$$\hat{A}-\lambda\hat{B} := \hat{L}_X(A-\lambda B)\hat{L}_Y, \quad \text{and} \quad \check{A}-\lambda\check{B} := \check{L}_X(A-\lambda B)\check{L}_Y,$$

are both proper tridiagonal pencils having the same lower pole tuple $\breve{\Xi}(A,B)$ with poles different from the spectrum. Then $\hat{A}-\lambda\hat{B}$ and $\check{A}-\lambda\check{B}$ are essentially identical *in the sense that $\hat{L}_X = D_X\check{L}_X$ and $\hat{L}_Y = \check{L}_Y D_Y$ with D_X and D_Y invertible diagonal matrices.*

Similarly, let $\acute{R}_X$, $\tilde{R}_X$, $\acute{R}_Y$ and $\tilde{R}_Y$ be invertible, upper triangular matrices with $\boldsymbol{e}_1^T\acute{R}_X^{-1} = \alpha\boldsymbol{e}_1^T\tilde{R}_X^{-1}$, $\alpha\in\mathbb{C}\setminus\{0\}$, such that

$$\acute{A}-\lambda\acute{B} := \acute{R}_X(A-\lambda B)\acute{R}_Y, \quad \text{and} \quad \tilde{A}-\lambda\tilde{B} := \tilde{R}_X(A-\lambda B)\tilde{R}_Y,$$

are both proper tridiagonal pencils having the same upper pole tuple $\breve{\Psi}(A,B)$ with poles different from the spectrum. Then $\acute{A}-\lambda\acute{B}$ and $\tilde{A}-\lambda\tilde{B}$ are essentially identical *in the sense that $\acute{R}_X = D_X\tilde{R}_X$ and $\acute{R}_Y = \tilde{R}_Y D_Y$ with D_X and D_Y invertible diagonal matrices.*

Proof. The idea of the proof of the first statement is the same as in the proof of Theorem 3.6.1. Using Corollary 3.6.8 and the shorthand notation,

$$\hat{M}_i = \hat{L}_X M_i \hat{L}_X^{-1}, \quad \text{and} \quad \check{M}_i = \check{L}_X M_i \check{L}_X^{-1},$$

for the elementary rational matrices (3.7) with shift ϱ_i and pole ξ_i before and after the equivalence. We get the following equalities,

$$\begin{aligned}
\hat{L}_X^{-1}K_n^{\mathrm{rat}}(\hat{A},\hat{B},\boldsymbol{e}_1,\breve{\Xi},\mathrm{P}) &= \hat{L}_X^{-1}\begin{bmatrix}\boldsymbol{e}_1 & \hat{M}_1\boldsymbol{e}_1 & \dots & \prod_{i=1}^{n-1}\hat{M}_i\boldsymbol{e}_1\end{bmatrix}\\
&= \alpha\check{L}_X^{-1}\begin{bmatrix}\boldsymbol{e}_1 & \check{M}_1\boldsymbol{e}_1 & \dots & \prod_{i=1}^{n-1}\check{M}_i\boldsymbol{e}_1\end{bmatrix}\\
&= \check{L}_X^{-1}K_n^{\mathrm{rat}}(\check{A},\check{B},\boldsymbol{e}_1,\breve{\Xi},\mathrm{P}),
\end{aligned}$$

which is an equality between two LU decompositions which is unique up to an invertible diagonal matrix D_X, i.e. $\hat{L}_X^{-1} = \check{L}_X^{-1}D_X^{-1}$. We now show that $\hat{L}_Y\boldsymbol{e}_1 = \tilde{\alpha}\check{L}_Y$ for some nonzero α. This follows from:

$$\hat{A}-\breve{\xi}_1\hat{B} = \hat{L}_X(A-\breve{\xi}_1 B)\hat{L}_Y, \quad \text{and} \quad \check{A}-\breve{\xi}_1\check{B} = \check{L}_X(A-\breve{\xi}_1 B)\check{L}_Y,$$

with $\breve{\xi}_1$ the same first subdiagonal pole in $\hat{A}-\lambda\hat{B}$ and $\check{A}-\lambda\check{B}$. Which yields,

$$\hat{L}_Y\boldsymbol{e}_1 = (A-\breve{\xi}_1 B)^{-1}\hat{L}_X^{-1}(\hat{A}-\breve{\xi}_1\hat{B})\boldsymbol{e}_1$$

$$\check{L}_Y\boldsymbol{e}_1 = (A-\breve{\xi}_1 B)^{-1}\check{L}_X^{-1}(\check{A}-\breve{\xi}_1\check{B})\boldsymbol{e}_1.$$

It follows that $\hat{L}_Y \boldsymbol{e}_1 = \tilde{\alpha}\check{L}_Y$ because $(\hat{A} - \check{\xi}_1\hat{B})\boldsymbol{e}_1 = \hat{\alpha}\boldsymbol{e}_1$ and $(\check{A} - \check{\xi}_1\check{B})\boldsymbol{e}_1 = \check{\alpha}$ for some nonzero $\hat{\alpha}, \check{\alpha}$. The proof of uniqueness of L_Y now follows from Corollary 3.6.8 just like before The second part of the theorem is essentially the same as the first part but for $A^T - \lambda B^T$. □

Theorem 6.6.1 implies that the result of a lower and upper RLR sweep is uniquely determined if the shift in step I and the pole in step III of the algorithm are fixed because choosing the shift for a lower RLR sweep determines $L_X^{-1}\boldsymbol{e}_1 = \gamma\boldsymbol{x}$ up to scaling. The same is valid for an upper RLR sweep. We also have that congruence transformation applied in an RT3 step on a proper, symmetric tridiagonal pencil is unique up to an invertible diagonal transformation as it just combines the lower and upper RLR sweeps.

We can also extend the convergence result of Theorem 3.7.3 to the non-unitary setting in a straightforward manner.

Theorem 6.6.2. *Let $A - \lambda B$ be a proper tridiagonal pencil with lower pole tuple $\Xi(A, B)$, upper pole tuple $\Psi(A, B)$ and with all the poles different from the spectrum. A lower RLR sweep* (6.12) *with shift ϱ and new pole ξ_n performs nested subspace iteration with a change of basis, similar to Theorem 3.7.3, accelerated by:*

$$\mathcal{R}\left(L_X^{-1}(:, 1:k)\right) = M(\varrho, \xi_k)\mathcal{E}_k, \quad \text{and} \quad \mathcal{R}\left(L_Y(:, 1:k)\right) = N(\varrho, \xi_{k+1})\mathcal{E}_k.$$

Similarly an upper RLR sweep with shift ϱ and new pole ψ_n implicitly performs nested subspace iteration accelerated by:

$$\mathcal{R}\left(R_X^T(:, 1:k)\right) = M(\varrho, \psi_{k+1})^T\mathcal{E}_k, \quad \text{and} \quad \mathcal{R}\left(R_Y^{-T}(:, 1:k)\right) = N(\varrho, \psi_k)^T\mathcal{E}_k.$$

In case $A - \lambda B$ is symmetric, an RT3 sweep with shift ϱ and new pole $\xi_n = \psi_n$ implicitly performs nested subspace iteration accelerated by:

$$\begin{aligned}\mathcal{R}\left(T^{-1}(:, 1:k)\right) &= M(\varrho, \xi_k)\mathcal{E}_k = N(\varrho, \psi_k)^T\mathcal{E}_k,\\ \mathcal{R}\left(T^T(:, 1:k)\right) &= N(\varrho, \xi_{k+1})\mathcal{E}_k = M(\varrho, \psi_{k+1})^T\mathcal{E}_k.\end{aligned}$$

We omit the proof as it is the same as the proof of Theorem 3.7.3. The interpretation of this result is also the same as in Chapter 3: a good selection of shifts and poles for lower RLR sweeps leads to rapid deflations of eigenvalues by driving subdiagonal elements in $A - \lambda B$ to zero, while a good selection of shift and poles for the upper RLR sweeps has the same effect on the superdiagonal. In the RT3 case, convergence occurs simultaneously on the sub- and superdiagonals.

6.7 Numerical experiments

As a "proof of concept" we test our implementation of the RT3 algorithm on randomly generated, symmetric matrix pencils. The real-valued matrices A and B, defining the pencils, are generated by first drawing their entries from the standard normal distribution and afterwards taking the symmetric part $(A + A^T, B + B^T)$. The result is an indefinite dense, symmetric pencil which is very likely to be block diagonalizable[1] having blocks of dimension 1 for real eigenvalues and dimension 2 for complex-conjugate eigenvalues.

These pencils are first reduced to symmetric tridiagonal form having all poles at 1 via the reduction algorithm of Sidje [108]. Finally, the RT3 algorithm for symmetric, block tridiagonal pencils is used to reduce them to a congruent block diagonal pencil. The results of this experiment are summarized in Figures 6.1 and 6.2 for pencils of dimension 50 up to 200.

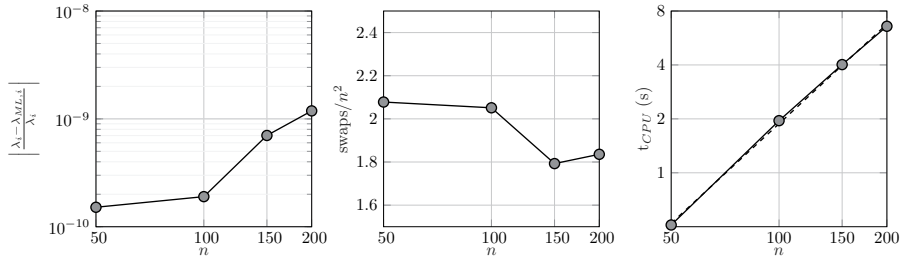


Figure 6.1: Results of a numerical experiment with the RT3 algorithm on indefinite symmetric pencils. Error on the eigenvalues (*left*), swaps/n^2 (*middle*), and CPU time with least-squares fit (*right*).

The left pane of Figure 6.1 shows the relative error on the eigenvalues obtained with RT3 in comparison to Matlab's `eig` function. We observe an accuracy of about 9 to 10 digits. The middle pane shows that we need about two swaps per eigenvalue squared, which is more than for the experiments in Chapters 3 and 5. It signals that convergence is slower. A potential explanation is that our deflation criteria are not yet on par with the criteria used in the RQZ method. We tested for a deflation by comparing subdiagonal blocks with the norm of (block) diagonal elements, but since the non-unitary transformations do not preserve the norm of the pencil, this could be suboptimal. Nonetheless, the CPU time shown in the right pane does closely match the expected $O(n^2)$ complexity in case the congruence transformations are not accumulated. The

[1] We did confirm numerically that the pencils are block diagonalizable, this is possibly an extension of the result that the set of non-diagonalizable matrices has measure zero. An analysis is outside the scope of this thesis.

dashed line shows the least-squares fit of the data with a leading exponent of $O(n^{1.9})$.

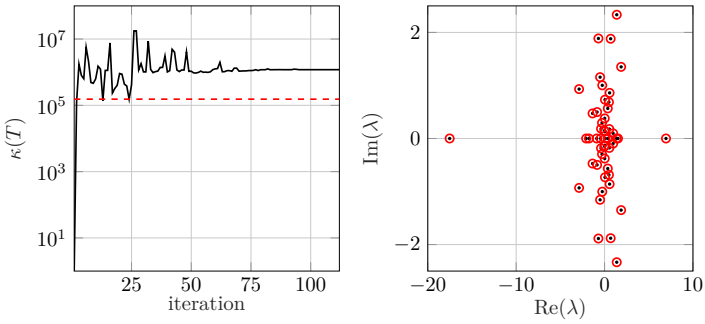


Figure 6.2: Part 2 of the results of a numerical experiment with the RT3 algorithm on indefinite symmetric pencils of dimension 50. Growth of the $\kappa(T)$ throughout the RT3 algorithm in black and $\kappa(V)$ in red (*left*), eigenvalues from `eig` in black and from RT3 in red (*right*).

The left pane of Figure 6.2 displays the growth in the condition number $\kappa(T)$ throughout the RT3 algorithm for a problem of dimension 50. The condition number of the eigenvectors computed with Matlab, $\kappa(V)$, is shown in red. In this example, the eigenvectors from RT3 have a condition number that is about one order of magnitude larger than Matlab and are thus suboptimal. The right part shows the spectrum of the pencil computed by RT3 and `eig`.

6.8 Conclusion

This chapter proposed and studied non-unitary pole swapping methods for computing the eigenvalues of (block) tridiagonal pencils. The algorithms require $O(n^2)$ operations but numerical stability is a difficult problem.

Nearly optimal scaled lower and upper triangular equivalence transformations were proposed for general (block) tridiagonal pencils and nearly optimal scaled congruence transformations for diagonalizable, symmetric tridiagonal pencils. The resulting RLR and RT3 algorithms perform a two-sided iteration accelerated by rational functions.

Numerical experiments confirm the validity of our approach but also highlight the remaining challenges. Future work entails a detailed analysis of the stability of the proposed schemes for certain subclasses of tridiagonal eigenproblems and the inclusion of (partial) balancing [77].

Chapter 7

Implicitly filtering the rational Krylov method

This chapter is based on the papers [19, 20]:

Camps D., Meerbergen K., and Vandebril R. A rational QZ method. (2019) SIAM J. Matrix Anal. Appl. Vol. 40, No. 3, pp. 943–972.
Camps D., Meerbergen K., and Vandebril R. An implicit filter for rational Krylov using core transformations. (2019) Linear Algebra and its Applications. Volume 561, 15 January 2019, Pages 113-140.

7.1 Introduction

The Arnoldi algorithm, discussed in Section 2.2.2, exhibits an orthogonalization cost which increases quadratically in the subspace dimension, while the storage requirements depend linearly on the number of basis vectors. This cost can become prohibitive for large-scale problems where the eigenvalues of interest are difficult to approximate in a limited number of iterations. This problem can be adequately solved by a *restart* of the Arnoldi method. Sorensen introduced the *implicitly restarted Arnoldi* method (IRA) [111]. His algorithm applies implicitly shifted QR steps [39, 40] to the Arnoldi Hessenberg matrix $\underline{H}_m$. Starting from an Arnoldi decomposition (2.29) related with a Krylov subspace

of dimension $m+1$, IRA implicitly applies p shifted QR steps with shifts $\{\varrho_i\}_{i=1}^p$ to end up with a reduced order Arnoldi decomposition related with the Krylov subspace $\mathcal{K}_{m-p+1}(A, p(A)\boldsymbol{v})$ with $p(A) = \prod_{i=1}^p (A - \varrho_i I)$. This process is depicted schematically as:

$$\mathcal{K}_{m+1}(A, \boldsymbol{v}) \xrightarrow[p \text{ shifted QR steps}]{p(z)=\prod_{i=1}^p (z-\varrho_i)} \mathcal{K}_{m-p+1}(A, p(A)\boldsymbol{v}).$$

IRA implicitly applies a *polynomial filter* determined by the shifts to the polynomial Krylov subspace. The implicitly restarted Arnoldi method was further analyzed by Morgan [84] and refined by Lehoucq & Sorensen [75]. It is to this date still used by ARPACK [76] to compute a few eigenpairs of a large-scale matrix. ARPACK is called by Matlab when `eigs` is invoked.

Stewart [112] introduced the Krylov-Schur algorithm where a proper subspace is extracted from the Krylov subspace via the Schur decomposition of the Arnoldi Hessenberg matrix.

In this chapter we apply the pole swapping technique of the RQZ method to implicitly apply a *rational* filter in the rational Krylov method. This approach can be viewed as a generalization of the use of the QR method in the implicitly restarted Arnoldi method. The pole swapping technique for the rational Krylov method was first proposed by Berljafa & Güttel in [11, Section 4.3]. Our additional contribution to this result is twofold. First, we make the connection with the RQZ method and the theory of Chapter 3, which allows us to easily characterize the implicit filter. Second, we compare the implicit filter with the filter method proposed by De Samblanx, Meerbergen & Bultheel [27]. Their method uses an explicit QZ step on the rational Krylov Hessenberg pencil. We will demonstrate in our numerical examples that the implicit pole swapping approach has several advantages over the explicit method.

The chapter is organized as follows. Section 7.2 introduces the rational Krylov method as an iterative method to construct an orthonormal basis for a rational Krylov subspace. We review how eigenvalue approximations can be extracted from the rational Krylov method and show in Lemma 7.2.5 that this is possible by computing the eigenvalues of a small-scale Hessenberg pair. This small-scale problem can be easily solved with the RQZ algorithm. We also study the matrix structure which is encoded in the Galerkin projection on a rational Krylov basis. Section 7.3 describes how the pole swapping technique is used to filter the rational Krylov method. Section 7.4 illustrates the *implicitly filtered rational Krylov* algorithm with four numerical examples that indicate that the method can outperform comparable methods. Some concluding remarks are given in Section 7.5.

7.2 Rational Krylov methods

The *rational Krylov* method was originally introduced by Ruhe in 1984 [92] and revisited by the same author in a series of papers [93–95] a decade after the initial article. The initial work of Ruhe culminates in [96] and by that time research into rational Krylov methods really started to gain traction [27, 74, 110].

Ruhe proposed the rational Krylov method for the solution of the eigenvalue problem and this will also be our main application of interest for rational Krylov in this thesis. Rational Krylov methods have been successful in many more applications after their introduction. Examples include – but are not limited to – the solution of nonlinear eigenvalue problems [52, 97, 117, 118], matrix equations [30, 109], model order reduction [7, 32, 47–49], and the computation of $f(A)\boldsymbol{v}$, i.e. the action of a matrix function on a vector [31, 44, 51].

7.2.1 Rational Krylov matrices and subspaces

In this section we briefly specify the rational Krylov theory from Section 3.6.1, which is formulated for matrix pairs, to the matrix case. The following elementary property is of great use to study the rational Krylov matrices and subspaces generated by a matrix. It is the well-known result that any matrix commutes with its (shifted) inverse.

Property 7.2.1 (Commutativity)**.** *Given a matrix A, shift $\varrho \in \bar{\mathbb{C}}$, and pole $\xi \in \bar{\mathbb{C}} \setminus \Lambda(A)$, we have that:*

$$(A-\varrho I)(A-\xi I)^{-1} = (A-\xi I)^{-1}(A-\varrho I).$$

It follows from this property that the elementary rational matrices (3.8) for a single matrix A, which corresponds with the pencil (A, I) satisfy,

$$M(\varrho, \xi) = N(\varrho, \xi).$$

Consequently, we also have,

$$K_k^{\text{rat}}(A, I, \boldsymbol{v}, \Xi, \mathrm{P}) = L_k^{\text{rat}}(A, I, \boldsymbol{v}, \Xi, \mathrm{P}),$$

$$\mathcal{K}_k^{\text{rat}}(A, I, \boldsymbol{v}, \Xi) = \mathcal{L}_k^{\text{rat}}(A, I, \boldsymbol{v}, \Xi),$$

according to Definitions 3.6.2 and 3.6.5. We can thus define a single rational Krylov matrix and subspace for the matrix case. The following definition specifies this and applies property II of Lemma 3.6.6 to the matrix case.

Definition 7.2.2 (rational Krylov matrix and subspace)**.** Let $A \in \mathbb{F}^{n\times n}$, $\boldsymbol{v} \in \mathbb{F}^n \setminus \{\mathbf{0}\}$, $\Xi = (\xi_1, \ldots, \xi_m)$, $\xi_i \in \bar{\mathbb{C}} \setminus \Lambda(A)$, the pole tuple, and $\mathrm{P} = (\varrho_1, \ldots, \varrho_m)$, $\varrho_i \in \bar{\mathbb{C}} \setminus \Xi$, the shift tuple. The corresponding *rational Krylov matrix*, $K^{\mathrm{rat}}_{m+1} \in \mathbb{F}^{n\times(m+1)}$, is defined as:

$$K^{\mathrm{rat}}_{m+1}(A, \boldsymbol{v}, \Xi, \mathrm{P}) = K^{\mathrm{rat}}_{m+1}(A, I, \boldsymbol{v}, \Xi, \mathrm{P}) \tag{7.1}$$

The *rational Krylov subspace*,

$$\mathcal{K}^{\mathrm{rat}}_{m+1}(A, \boldsymbol{v}, \Xi) = \mathcal{R}\left(K^{\mathrm{rat}}_{m+1}(A, \boldsymbol{v}, \Xi, \mathrm{P})\right), \tag{7.2}$$

is defined as the columnspace of the rational Krylov matrix. The rational Krylov subspace satisfies:

$$\mathcal{K}^{\mathrm{rat}}_{m+1}(A, \boldsymbol{v}, \Xi) = q_m(A)^{-1}\mathcal{K}_{m+1}(A, \boldsymbol{v}) = \mathcal{K}_{m+1}(A, q_m(A)^{-1}\boldsymbol{v}), \tag{7.3}$$

with $q_m(z) = \prod_{i=1}^{m}(z - \xi_i) \in \mathcal{P}_m$, a polynomial with roots Ξ. Every root in Ξ equal to ∞ reduces the degree of q_m by one.

Observe that (7.3) shows that a rational Krylov subspace is nothing else than a Krylov subspace with a special starting vector. This result follows from Lemma 3.6.6, but by using Property 7.2.1 it can be easily proven directly in the matrix case.

Proof of (7.3). Combining (7.2) with Property 7.2.1 gives:

$$\mathcal{K}^{\mathrm{rat}}_{m+1}(A, \boldsymbol{v}, \Xi) =$$

$$\prod_{i=1}^{m}(A-\xi_i I)^{-1} \cdot \mathcal{R}\left(\prod_{i=1}^{m}(A-\xi_i I)\boldsymbol{v}, (A-\varrho_1 I)\prod_{i=2}^{m}(A-\xi_i I)\boldsymbol{v}, \ldots, \prod_{i=1}^{m}(A-\varrho_i I)\boldsymbol{v}\right).$$

As the shifts P are chosen different from the poles Ξ, the vectors remaining inside $\mathcal{R}(\cdot)$ in the equation above are $m+1$ vectors of the form $p_m(A)\boldsymbol{v}$ with $p_m \in \mathcal{P}_m$ a polynomial of degree $\leq m$ having at least 1 root distinct from all other polynomials. It follows by a simple inductive argument that the polynomials are linearly independent under the conditions on Ξ and P. Consequently, the subspace $\mathcal{R}(\cdot)$ above is isomorphic to the subspace $\mathcal{R}\left(p_m(A)\boldsymbol{v} \,|\, p_m \in \mathcal{P}_m\right)$, i.e. the Krylov subspace $\mathcal{K}_{m+1}(A, \boldsymbol{v})$. □

7.2.2 Ruhe's iterative method

Algorithm 2 lists the rational Krylov algorithm, also known as the *rational Arnoldi algorithm*. It iteratively constructs an orthonormal basis of

$\mathcal{K}^{\text{rat}}_{m+1}(A, \boldsymbol{v}, \Xi)$. Comparing this algorithm to Algorithm 1, the only difference is in lines 4 and 5 where first the *continuation combination* is computed or selected and next a *rational expansion* step is taken. The remaining part of the algorithm just uses modified Gram-Schmidt to orthonormalize the basis vectors.

Algorithm 2 rational Krylov algorithm [92, 93, 96]

Input: A, $\boldsymbol{v}$, m

1: *Start:* $\boldsymbol{v}_1 \leftarrow \boldsymbol{v} / \|\boldsymbol{v}\|_2$
2: *Iterate:*
3: **for** $j = 1, 2, \ldots, m$ **do**
4: $\quad \xi_j = \alpha_j/\beta_j$, $\varrho_j = \gamma_j/\delta_j$, and $\boldsymbol{t}_j \in \mathbb{C}^j$ ▷ Cont. combination
5: $\quad \boldsymbol{v}_{j+1} \leftarrow (\delta_j A - \gamma_j I)(\beta_j A - \alpha_j I)^{-1} V_j \boldsymbol{t}_j$ ▷ Rational expansion
6: $\quad$ **for** $i = 1, \ldots, j$ **do** ▷ modified Gram-Schmidt
7: $\quad\quad h_{i,j} \leftarrow \boldsymbol{v}_i^* \boldsymbol{v}_{j+1}$
8: $\quad\quad \boldsymbol{v}_{j+1} \leftarrow \boldsymbol{v}_{j+1} - h_{i,j} \boldsymbol{v}_i$
9: $\quad$ **end for**
10: $\quad h_{j+1,j} \leftarrow \|\boldsymbol{v}_{j+1}\|_2$ ▷ normalize
11: $\quad \boldsymbol{v}_{j+1} \leftarrow \boldsymbol{v}_{j+1} / h_{j+1,j}$
12: **end for**

The continuation combination consists of the *pole* ξ_j, the *shift* ϱ_j, and the *continuation vector* $\boldsymbol{t}_j$ which is used to construct a vector $V_j \boldsymbol{t}_j$ to expand the subspace with in line 5. In the Arnoldi method, the continuation vector is always $\boldsymbol{t}_j = \boldsymbol{e}_j$ as the subspace is expanded with $A\boldsymbol{v}_j$, cfr. Algorithm 1. We will discuss the choice of continuation combination in some more detail in a moment.

Inspecting Algorithm 2, we observe the following recurrence relation at iteration j:

$$(\delta_j A - \gamma_j I)(\beta_j A - \alpha_j I)^{-1} V_j \boldsymbol{t}_j = \sum_{i=1}^{j+1} h_{i,j} \boldsymbol{v}_i, \tag{7.4}$$

which is the rational Krylov variant of (2.28). Using Property 7.2.1 to reorder the left-hand side and rewriting the right-hand side as a matrix-vector product gives,

$$(\beta_j A - \alpha_j I)^{-1} (\delta_j A - \gamma_j I) V_j \boldsymbol{t}_j = V_{j+1} \underline{\boldsymbol{h}}_j, \tag{7.5}$$

with $\underline{\boldsymbol{h}}_j \in \mathbb{F}^{j+1}$ the vector of orthonormalization coefficients. Left multiplication with $(\beta_j A - \alpha_j I)$ yields:

$$(\delta_j A - \gamma_j I) V_j \boldsymbol{t}_j = (\beta_j A - \alpha_j I) V_{j+1} \underline{\boldsymbol{h}}_j, \tag{7.6}$$

which can be rearranged to:

$$AV_{j+1}(\delta_j \underline{\boldsymbol{t}}_j - \beta_j \underline{\boldsymbol{h}}_j) = V_{j+1}(\gamma_j \underline{\boldsymbol{t}}_j - \alpha_j \underline{\boldsymbol{h}}_j), \tag{7.7}$$

with $\underline{\boldsymbol{t}}_j^T = \begin{bmatrix} \boldsymbol{t}_j^T & 0 \end{bmatrix}^T$ If we combine iterations $j = 1, \ldots, m$, we get the rational Krylov recurrence relation:

$$AV_{m+1}\underline{K}_m = V_{m+1}\underline{L}_m, \tag{7.8}$$

where $\underline{L}_m, \underline{K}_m \in \mathbb{F}^{(m+1)\times m}$ are a pair of Hessenberg matrices with the nonzero entries in column j equal to:

$$\underline{\boldsymbol{k}}_j = \delta_j \underline{\boldsymbol{t}}_j - \beta_j \underline{\boldsymbol{h}}_j, \quad \text{and,} \quad \underline{\boldsymbol{\ell}}_j = \gamma_j \underline{\boldsymbol{t}}_j - \alpha_j \underline{\boldsymbol{h}}_j. \tag{7.9}$$

It follows that the Hessenberg pairs satisfies $\ell_{j+1,j}/k_{j+1,j} = \alpha_j/\beta_j = \xi_j$, i.e. the poles used in the rational Krylov algorithm are *encoded* in the Hessenberg matrices $\underline{L}_m, \underline{K}_m$ as the ratio of their subdiagonal elements. We refer to $(\underline{L}_m, \underline{K}_m)$ as the *rational Krylov Hessenberg pair* and to $(V_{m+1}, \underline{L}_m, \underline{K}_m)$ as the *rational Krylov triplet*. The rational Krylov Hessenberg pair is of full rank m as long as there is no breakdown in line 10 [27]. We call both the rational Krylov Hessenberg pair *and* triplet *proper* if no breakdown occurs. It is easy to verify that properness of the rational Krylov Hessenberg pair is in agreement with two out of three conditions of Definition 3.2.1. Only the condition on the last rows of the pencil does not hold because of the rectangular form.

Let us now return to the problem of choosing the continuation combination which is used for the expansion step. The first parameter is the pole ξ_j. Poles are typically chosen in such a way that the method rapidly converges to the quantities of interest. This makes pole selection a highly nontrivial problem. Optimal strategies have been proposed for the solution of matrix functions [51] and the convergence of eigenvalue approximations in rational Krylov iterations has been studied in [9]. The second parameter, ϱ, and the third parameter, $\boldsymbol{t}$, do *not* affect the resulting subspace, i.e. $\mathcal{R}(V_{m+1})$ in (7.8) is independent of the choice of shift and continuation vector as long as *admissible* shifts and continuation vectors are used [12]. Admissible means that the vector computed in line 5 satisfies:

$$(A - \varrho_j I)(A - \xi_j I)^{-1} V_j \boldsymbol{t}_j \notin \mathcal{R}(V_j), \tag{7.10}$$

such that the subspace is expanded. An admissible shift and continuation vector always exist as long as the subspace is not A-invariant [12, 96]. This is also a corollary of Theorem 3.6.4 and Lemma 3.6.6. A common choice for the continuation vector $\boldsymbol{t}_j$ is [96]:

$$\boldsymbol{t}_j = \begin{cases} \boldsymbol{e}_j & \text{if } \xi_j = \xi_{j-1}, \text{ or } j = 1 \\ \boldsymbol{q}_j = Q_j \boldsymbol{e}_j & \text{otherwise} \end{cases}, \tag{7.11}$$

with Q_j computed from the QR factorization of $\beta_j \underline{L}_{j-1} - \alpha_j \underline{K}_{j-1}$. Choosing $\boldsymbol{t}_j$ according to (7.11) ensures an admissible continuation combination for any choice of shift [12].

Berljafa & Güttel have proven existence and *essential* uniqueness results for rational Krylov decompositions (7.8) in [11]. These are summarized in the following two results.

Theorem 7.2.3 (Theorem 2.5 in [11])**.** *Let $(V_{m+1}, \underline{L}_m, \underline{K}_m)$ be a proper rational Krylov triplet satisfying* (7.8) *for $A \in \mathbb{F}^{n\times n}$. Let $\boldsymbol{v} = V_{m+1}\boldsymbol{e}_1$, and $\Xi = (\xi_1, \ldots, \xi_m)$, $\xi_i \notin \Lambda(A)$, be the poles of the rational Krylov Hessenberg pair. Then we have, for $k = 1, \ldots, m+1$:*

$$\mathcal{R}(V_k) = \mathcal{K}_k^{rat}(A, \boldsymbol{v}, \Xi).$$

Theorem 7.2.4 (Theorem 3.2 in [11])**.** *If the starting vector $\boldsymbol{v}$ and the ordering of the poles Ξ is fixed, then a proper rational Krylov triplet, $(V_{m+1}, \underline{K}_m, \underline{L}_m)$, satisfying* (7.8) *is unique up to* equivalent *triplets of the form,*

$$(V_{m+1}D_{m+1}, D_{m+1}^{*}\underline{K}_m R_m, D_{m+1}^{*}\underline{L}_m R_m),$$

with $D_{m+1} \in \mathbb{F}^{(m+1)\times(m+1)}$ a unitary diagonal matrix, and $R_m \in \mathbb{F}^{m\times m}$ a nonsingular upper triangular matrix.

The combination of these results indicates that Algorithm 2 produces a rational Krylov recurrence which is linked uniquely with a rational Krylov subspace up to equivalent triplets.

Extended Krylov

A special case of rational Krylov is the *extended* Krylov method. This specification of rational Krylov was originally proposed by Druskin & Knizhnerman in 1998 [29] for the approximation of matrix functions and studied further by Knizhnerman & Simoncini [64].

The extended Krylov method is nothing else then the rational Krylov method where the choice of poles is limited to $\xi_j \in \{0, \infty\}$. This implies that the matrix A must be nonsingular to construct an extended Krylov subspace. In every step of Algorithm 2, the rational expansion is in the extended Krylov method performed either with $AV_j\boldsymbol{t}_j$ or $A^{-1}V_j\boldsymbol{t}_j$.

The resulting *extended Krylov Hessenberg* pair $(\underline{L}_m, \underline{K}_m)$ has the property that $\ell_{j+1,j} = 0$ when $\xi_j = 0$, and $k_{j+1,j} = 0$ when $\xi_j = \infty$, i.e. there is exactly one nonzero subdiagonal element for every column j. This is in agreement with the extended Hessenberg pencils introduced in Appendix A.

The extended Krylov method can, in some cases, be preferred over the rational Krylov method. It has fewer free parameters making it less difficult to select the poles. Furthermore, it requires only one matrix factorization to solve the linear systems in Algorithm 2, while the rational Krylov method requires a new factorization for every unique pole.

7.2.3 Ritz values in rational Krylov

To use the rational Krylov algorithm for eigenvalue problems, we need a way to extract approximate eigenvalues or Ritz values satisfying a Galerkin constraint. Just like for the Arnoldi method where the Ritz values can be computed as the eigenvalues of the leading $m \times m$ upper Hessenberg matrix H_m according to (2.35). To get a Galerkin projection condition (Definition 2.2.7), we require an orthogonality constraint against an m-dimensional subspace of $\mathcal{K}^{\text{rat}}_{m+1}(A, \boldsymbol{v}, \Xi)$. A common choice for Ritz values in rational Krylov is $\mathcal{V} = \mathcal{R}(V_{m+1}\underline{K}_m)$ [10]. The following lemma characterizes a generalized eigenvalue problem satisfying this constraint.

Lemma 7.2.5 (Ritz values in rational Krylov)**.** *Let $(V_{m+1}, \underline{L}_m, \underline{K}_m)$ be a rational Krylov triplet satisfying a rational Krylov recurrence* (7.8) *for $A \in \mathbb{F}^{n\times n}$. Then $(\vartheta, \boldsymbol{z} = V_{m+1}\underline{K}_m\boldsymbol{y}_m)$ is a Ritz pair of A with respect to $\mathcal{R}(V_{m+1}\underline{K}_m)$ if and only if $(\vartheta, \boldsymbol{y}_m)$ is an eigenpair of the $m\times m$ Hessenberg pencil:*

$$(\tilde{L}_m, \tilde{K}_m) = (L_m + \ell_{m+1,m}\bar{k}_{m+1,m}\boldsymbol{f}_m\boldsymbol{e}_m^T, K_m + |k_{m+1,m}|^2\boldsymbol{f}_m\boldsymbol{e}_m^T), \tag{7.12}$$

with $\boldsymbol{f}_m = K_m^{-*}\boldsymbol{e}_m$.

Proof. $(\vartheta, \boldsymbol{z} = V_{m+1}\underline{K}_m\boldsymbol{y}_m)$ is a Ritz pair of A with respect to $\mathcal{R}(V_{m+1}\underline{K}_m)$ if and only if:

$$\begin{aligned} V_{m+1}\underline{K}_m \perp A\boldsymbol{z} - \vartheta\boldsymbol{z} &= AV_{m+1}\underline{K}_m\boldsymbol{y}_m - \vartheta V_{m+1}\underline{K}_m\boldsymbol{y}_m \\ &= V_{m+1}(\underline{L}_m - \vartheta\underline{K}_m)\boldsymbol{y}_m, \end{aligned}$$

which is satisfied if:

$$\underline{K}_m^*(\underline{L}_m - \vartheta\underline{K}_m)\boldsymbol{y}_m = \boldsymbol{0}.$$

The Ritz values are thus the eigenvalues of the pencil $(\underline{K}_m^*\underline{L}_m, \underline{K}_m^*\underline{K}_m)$. This is not a Hessenberg, Hessenberg pencil but left multiplication with the nonsingular matrix K_m^{-*} gives the equivalent Hessenberg, Hessenberg pencil $(\tilde{L}_m, \tilde{K}_m)$ of (7.12). □

The Hessenberg pencil $(\tilde{L}_m, \tilde{K}_m)$ in (7.12) only differs from (L_m, K_m) in its last column and a single $m\times m$ linear system with the lower Hessenberg matrix K_m^*

needs to be solved to compute the update vector $\boldsymbol{f}_m$. The rational QZ method can be directly used to compute the eigenvalues of $(\tilde{L}_m, \tilde{K}_m)$. We remark that, in case $\xi_m = \infty$, $k_{m+1,m} = 0$ which means that $(\tilde{L}_m, \tilde{K}_m) = (L_m, K_m)$ by (7.12).

As shown in the proof of Lemma 7.2.5, the Ritz values are also equal to the eigenvalues of $(\underline{K}_m^* \underline{L}_m, \underline{K}_m^* \underline{K}_m)$. This pencil can be transformed to the equivalent matrix $(\underline{K}_m^* \underline{K}_m)^{-1} \underline{K}_m^* \underline{L}_m$ which is equal to:

$$H_m^{\text{rat}} = \underline{K}_m^{\dagger} \underline{L}_m, \tag{7.13}$$

with $\underline{K}_m^{\dagger}$ the Moore-Penrose pseudoinverse of the full rank matrix $\underline{K}_m$ [46]. H_m^{rat} is referred to as the Galerkin projection of A on $\mathcal{R}(V_{m+1}\underline{K}_m)$, similar to the polynomial Krylov case. We will show in the next section that H_m^{rat} has a *rational Hessenberg* structure. The Ritz values are clearly also determined by the eigenvalues of H_m^{rat}, however, in practice using (7.12) to determine the Ritz values has two main advantages. Firstly, it is computationally less expensive to compute (7.12) than (7.13). The former requires the solution of a single linear system of dimension m, the latter requires m least-square solves. Secondly, the Hessenberg pencil in (7.12) can be directly solved with RQZ, while the matrix (7.13) first needs to be reduced to Hessenberg form.

Ritz values in the rational Krylov method often converge much faster to the eigenvalues of interest compared to the polynomial Krylov method provided a good choice of poles is made. We illustrate the convergence behaviour in Figure 7.1 which shows a Ritz plot obtained with the rational Krylov method where the poles are cyclically chosen at $18, 20.05$, and 22, for the same matrix that was studied in Figure 2.1. Compared to the Arnoldi method, the convergence of eigenvalues in the interior part of the spectrum now proceeds much faster and the extremal eigenvalue at 39 only converges near the end of the iteration.

The convergence of rational Ritz values was studied in detail in [9] where it is shown that poles do *attract* convergence of Ritz values.

7.2.4 Structure in the Galerkin projection

In this section, we study the matrix structure of the Galerkin projection (7.13) of the original large-scale problem onto the rational Krylov subspace $\mathcal{R}(V_{m+1}\underline{K}_m)$. To this end, we consider the core factorized form of the rational Krylov Hessenberg pair in combination with the *transfer* operation for core transformations. The transfer operation is introduced in Appendix A.

It is well-known that the Galerkin projection on a rational Krylov subspace is of a particular rank-structured form in the Hermitian case [38] and the

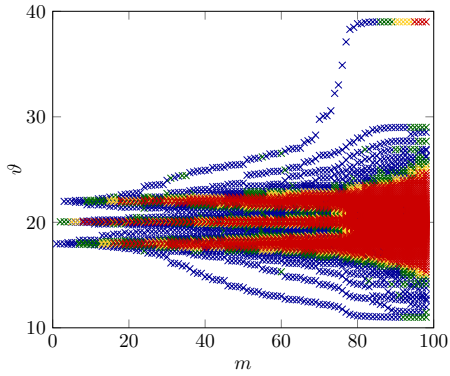


Figure 7.1: Convergence of rational Ritz values.

unsymmetric case [79, 115]. We will prove this same result but via a direct proof that exploits the matrix structure instead of using the theory of orthogonal rational functions [79, 115]. Up to our knowledge this is a novel approach for this result.

Let us start with recapitulating the appropriate matrix structures. These are also illustrated in Figure 7.2. We are already acquainted with the factorized representation of the Hessenberg matrix shown in pane I. Lemma 2.2.8 shows that the Galerkin projection on a polynomial Krylov subspace has this structure. The *extended Hessenberg* structure illustrated in pane II provides additional flexibility by admitting different orderings in the sequence of core transformations. Finally, pane III shows an example of an extended Hessenberg plus diagonal matrix. Lemma 7.2.6 shows that this matrix structure is linked with the Galerkin projection on a rational Krylov subspace. For this reason, we introduce the alternative name *rational Hessenberg* matrix for this matrix structure. Appendix A provides more information on extended Hessenberg matrices and the *transfer operation* for core transformations which we use for the proof of Lemma 7.2.6.

Lemma 7.2.6. *Let* $(V_{m+1}, \underline{K}_m, \underline{L}_m)$ *be a proper rational Krylov triplet corresponding to the rational Krylov subspace* $\mathcal{K}^{rat}_{m+1}(A, \boldsymbol{v}_1, \Xi = (\xi_1, \ldots, \xi_m))$. *Consider the vector* $\boldsymbol{d} \in \mathbb{C}^m$ *with* $d_i = \xi_i$ *for* $\xi_i \neq \infty$, *otherwise* d_i *can be any scalar. Then we have that* $H^{rat}_m = \underline{K}^{\dagger}_m \underline{L}_m$ *is a rational Hessenberg matrix* $QR + D$ *with:*

- $Q = C_{k_1} \ldots C_{k_{m-1}}$ *satisfying* $\begin{bmatrix} C_i C_{i+1} & \text{if} \quad \xi_{i+1} = \infty \\ C_{i+1} C_i & \text{if} \quad \xi_{i+1} \neq \infty \end{bmatrix}, i = 1, \ldots m-2,$

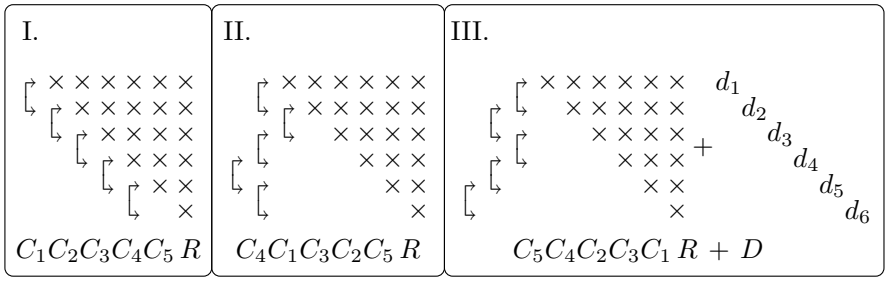


Figure 7.2: Examples of the representation of (I) Hessenberg, (II) extended Hessenberg and (III) rational Hessenberg matrices.

- $D = diag(\boldsymbol{d})$.

Proof. Consider the matrix $\underline{G}_m = \underline{L}_m - \underline{K}_m D$, which is an $(m+1)\times m$ upper Hessenberg matrix as both $\underline{L}_m$ and $\underline{K}_m D$ are upper Hessenberg of size $(m+1)\times m$. The subdiagonal elements of $\underline{G}_m$ are $h_{i+1,i} = \ell_{i+1,i} - d_{ii}k_{i+1,i}$ for $i \in \{1,\ldots,m\}$. As $d_{ii} = \xi_i = \ell_{i+1,i}/k_{i+1,i}$ if $k_{i+1,i} \neq 0$, we have that $h_{i+1,i} = 0$ if the pole ξ_i is not at infinity. For poles at infinity, $h_{i+1,i} = \ell_{i+1,i} \neq 0$ since there is no breakdown. The matrix $\underline{G}_m$ has thus a zero subdiagonal element whenever $\underline{K}_m$ has a nonzero subdiagonal element and vice versa. We have $\underline{K}_m^\dagger \underline{L}_m = \underline{K}_m^\dagger \underline{G}_m + \underline{K}_m^\dagger \underline{K}_m D$. As $\underline{K}_m$ is full rank, $\underline{K}_m^\dagger \underline{K}_m = I_m$ and it remains thus to examine the QR factorization of $\underline{K}_m^\dagger \underline{G}_m$ to prove the Lemma.

We get,

$$\underline{K}_m^\dagger \underline{G}_m = \underline{R}_K^\dagger Q_K^* Q_G \underline{R}_G = \begin{bmatrix} R_K^{-1} & 0 \end{bmatrix} Q_K^* Q_G \begin{bmatrix} R_G \\ 0 \end{bmatrix},$$

where R_K^{-1} is well-defined since $\underline{K}_m$ is of maximal rank. The unitary matrices can be represented as a product of core transformations as,

$$Q_K = \tilde{C}_{i_1}^* \cdots \tilde{C}_{i_k}^*, \text{ and } Q_G = \tilde{C}_{j_1} \cdots \tilde{C}_{j_\ell},$$

where $k+\ell = m$, $i_1 < i_2 < \ldots < i_k$, $j_1 < j_2 < \ldots < j_\ell$ and $\{i_1,\ldots,i_k\} \cup \{j_1,\ldots,j_l\}$ equal to $\{1,\ldots,m\}$. The Hermitian conjugates in Q_K are only introduced for convenience. The product of both unitary matrices, $\tilde{Q} = Q_K^* Q_G$ is equal to:

$$\tilde{Q} = Q_K^* Q_G = \tilde{C}_{i_k} \ldots \tilde{C}_{i_1} \tilde{C}_{j_1} \ldots \tilde{C}_{j_l} = \tilde{C}_{k_1} \cdots \tilde{C}_{k_{n-1}}.$$

We will prove next that the mutual ordering of $\tilde{C}_i$ and $\tilde{C}_{i+1}$ in the factorization of $\tilde{Q}$ is imposed by ξ_{i+1} as specified in the formulation of the Lemma.

- If $\xi_{i+1} = \infty$, then $i{+}1 \in \{j_1, \ldots, j_l\}$ as it was designed to create a zero in $\underline{G}_m$. There are two possibilities, either $\tilde{C}_i$ appears in Q_K^* or in Q_G. If it is in Q_K^* it is clearly to the left, if it is in Q_G, then $j_1 < \ldots < j_l$ ensures that is located to the left of C_{i+1}.
- If $\xi_{i+1} \neq \infty$, then $i{+}1 \in \{i_1, \ldots, i_k\}$ and an analogous reasoning shows that $\tilde{C}_i$ must be positioned right of $\tilde{C}_{i+1}$.

There are two possibilities for the mth core transformation. If $\xi_m \neq \infty$ then the mth core transformation in $\tilde{Q}$ is located on the left of core transformation $m-1$ and we can write $\tilde{Q} = C_m \tilde{Q}_{1\ldots m-1}$ with $\tilde{Q}_{1\ldots m-1}$ the unitary matrix formed by the first $m-1$ core transformations. This gives:

$$\underline{K}_m^\dagger \underline{G}_m = \begin{bmatrix} R_K^{-1} & 0 \end{bmatrix} C_m \tilde{Q}_{1\ldots m-1} \begin{bmatrix} R_G \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{R} & \otimes \end{bmatrix} \tilde{Q}_{1\ldots m-1} \begin{bmatrix} R_G \\ 0 \end{bmatrix} = Q_{1\ldots m-1} R.$$

In the second equality C_m is applied to columns m and $m+1$ of $\underline{R}_K^\dagger$, this preserves the upper triangular structure in the left $m \times m$ block. For the third equality $\tilde{Q}_{1\ldots m-1}$ is transferred from the right of $[\tilde{R} \otimes]$ to the left. Since $\tilde{Q}_{1\ldots m-1}$ only affects the first m columns, both the upper triangularity in the left $m \times m$ block and the mutual ordering of the core transformations are preserved.

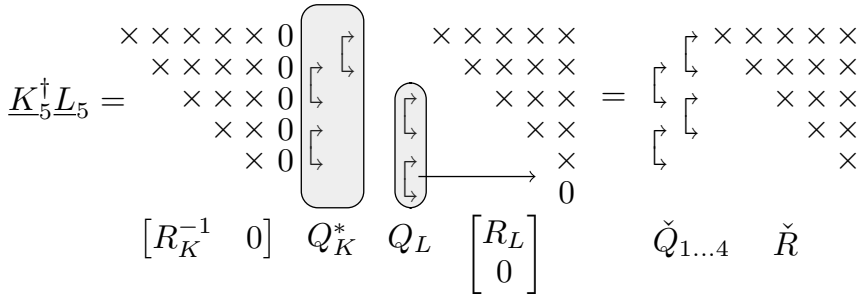
Similarly, if $\xi_m = \infty$, the mth core transformation is located right from core transformation $m-1$ and we can write $\tilde{Q} = \tilde{Q}_{1\ldots m-1} C_m$ to get:

$$\underline{K}_m^\dagger \underline{G}_m = \begin{bmatrix} R_K^{-1} & 0 \end{bmatrix} \tilde{Q}_{1\ldots m-1} C_m \begin{bmatrix} R_G \\ 0 \end{bmatrix} = \begin{bmatrix} R_K^{-1} & 0 \end{bmatrix} \tilde{Q}_{1\ldots m-1} \begin{bmatrix} \tilde{R} \\ \otimes \end{bmatrix} = Q_{1\ldots m-1} R.$$

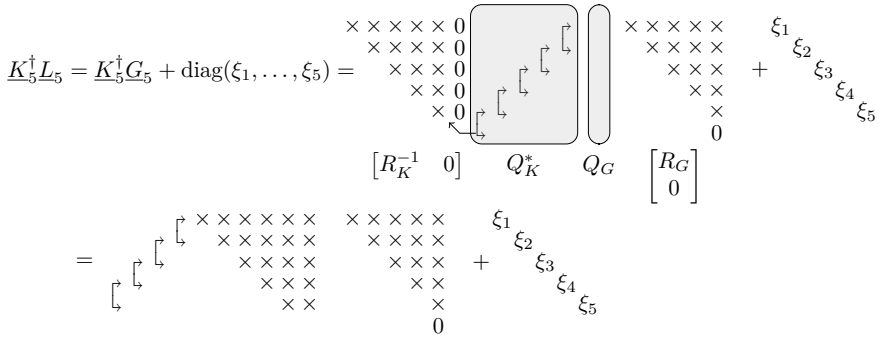
□

Let us further illustrate this structure with two simple examples.

Example 7.2.7. In the case of the projection on an *extended Krylov subspace*, the vector $\boldsymbol{d}$ can be chosen as the zero vector according to Lemma 7.2.6 such that we can choose $\underline{G}_m = \underline{L}_m$ in the proof of Lemma 7.2.6. We end up with a matrix in extended Hessenberg format because of this. For example, the Galerkin projection $\underline{K}_m^\dagger \underline{L}_m$ on an extended Krylov subspace with pole tuple $\Xi_{\text{ext}} = (0, 0, \infty, 0, \infty)$ is of the form:



The Galerkin projection $\underline{K}_m^\dagger \underline{L}_m$ on the rational Krylov subspace with $\Xi = (\xi_1, \xi_2, \xi_3, \xi_4, \xi_5)$, assuming all $\xi_i \neq \infty$, is of the form:



Since there are no infinite poles in this example the matrix $\underline{G}_5$, as used in the proof of Lemma 7.2.6, is an upper triangular matrix. As a consequence, all 5 core transformations are located in an ascending pattern that forms Q_K^*. In the second equality, the core transformation acting on columns 5 and 6 is merged into the upper triangular matrix. This creates nonzero elements in the last column, but the remaining core transformation can be transferred to the left because the first columns are still in upper triangular form. The result is shown on the second row. It is clear that the product of the 5×6 and 6×5 upper triangular matrices result in a 5×5 matrix in upper triangular form.

7.3 Filtering the rational Krylov method

To implicitly apply a rational filter in the rational Krylov method we can use the pole swapping concept of the RQZ method. The filter mechanism is summarized in Figure 7.3 for a small example. The algorithm starts from a proper rational Krylov triplet $(V_{m+1}, \underline{L}_m, \underline{K}_m)$ with poles $\Xi = (\xi_1, \ldots, \xi_m)$. The initial state of the rational Krylov Hessenberg pencil is shown in pane I on the left. In pane II, the first pole ξ_1 is changed to a shift ϱ by computing a unitary transformation Q such that,

$$\boldsymbol{q}_1 = \check{\gamma}(\underline{L}_m - \varrho\underline{K}_m)(\underline{L}_m - \xi_1 \underline{K}_m)^\dagger \boldsymbol{e}_1 = \hat{\gamma}(\underline{L}_m - \varrho\underline{K}_m)\boldsymbol{e}_1. \tag{7.14}$$

The principle is the same as (3.2), the only difference is that the inverse is replaced with the Moore-Penrose pseudoinverse $(\underline{L}_m - \xi_1 \underline{K}_m)^\dagger$.

It is well-known [46] that $\boldsymbol{x}_{LS} = (\underline{L}_m - \xi_1 \underline{K}_m)^\dagger \boldsymbol{b}$ is the least squares solution of minimal norm $\|\boldsymbol{x}\|_2$. As $\|\gamma \boldsymbol{e}_1 - (\underline{L}_m - \xi_1 \underline{K}_m)\boldsymbol{e}_1\|_2 = 0$ when $\gamma = \ell_{11} - \xi_1 k_{11}$, we conclude that,

$$(\underline{L}_m - \xi_1 \underline{K}_m)^\dagger \boldsymbol{e}_1 = \gamma \boldsymbol{e}_1. \tag{7.15}$$

Pane II of Figure 7.3 further shows how the shift is swapped to the last position on the subdiagonal of $(\underline{L}_m, \underline{K}_m)$. The end result is displayed in pane III.

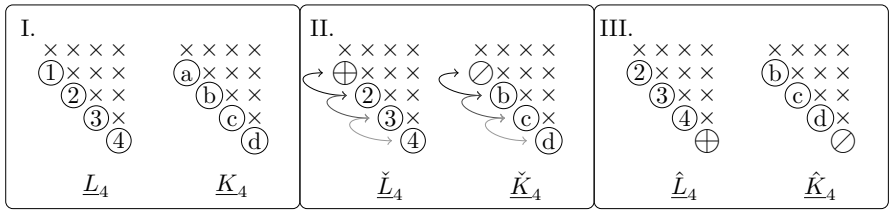


Figure 7.3: RQZ-like procedure to change the first pole in a rational Krylov Hessenberg pair to a new shift (Pane II) and move it to the last subdiagonal position in the rational Krylov Hessenberg pair (Pane II-III).

The process shown in Figure 7.3 effectively updates,

$$(\hat{\underline{L}}_m, \hat{\underline{K}}_m) = Q^*(\underline{L}_m, \underline{K}_m)Z,$$

in such a way that the pole tuple is changed to $\hat{\Xi} = (\xi_2, \ldots, \xi_m, \varrho)$. To maintain the rational Krylov recurrence (7.8) the orthonormal basis is also updated as $\hat{V}_{m+1} = V_{m+1}Q$.

This does *not* change the span of V_{m+1}, i.e. $\mathcal{R}(\hat{V}_{m+1}) = \mathcal{R}(V_{m+1})$, but the vectors are *rearranged*. The new starting vector is given by:

$$\hat{\boldsymbol{v}} = \hat{V}_{m+1}\, \boldsymbol{e}_1 = V_{m+1}\, \boldsymbol{q}_1 = \hat{\gamma} V_{m+1}(\underline{L}_m - \varrho\underline{K}_m)\boldsymbol{e}_1. \tag{7.16}$$

The rational Krylov recurrence (7.8) implies,

$$(A-\varrho I)\,V_{m+1}\,(\underline{L}_m-\xi_1\underline{K}_m) = (A-\xi_1 I)\,V_{m+1}\,(\underline{L}_m-\varrho\underline{K}_m). \tag{7.17}$$

Rearranging terms in (7.17) and combining this with (7.15) and (7.16), we see that the new starting vector is given by:

$$\hat{\boldsymbol{v}} = \gamma(A-\xi_1 I)^{-1}(A-\varrho I)\,\boldsymbol{v}. \tag{7.18}$$

From the uniqueness of a rational Krylov triplet (7.8) stated in Theorem 7.2.4 in combination with Theorem 7.2.3 it follows that $\mathcal{R}(\hat{V}_{m+1}) = \mathcal{K}^{\mathrm{rat}}_{m+1}(A, \hat{\boldsymbol{v}}, \hat{\Xi})$.

The filter operation is finalized by removing the last pole ϱ from the subspace which reduces the order of the rational Krylov recurrence by one. This means that the trailing column and row of $(\hat{\underline{L}}_m, \hat{\underline{K}}_m)$ are removed, as well as the last vector of $\hat{V}_{m+1}$. The result is a proper rational Krylov triplet $(\hat{V}_m, \hat{\underline{L}}_{m-1}, \hat{\underline{K}}_{m-1})$ of reduced order with poles $\Xi = (\xi_2, \ldots, \xi_m)$ and start vector (7.18).

Repeating this process p times, we pictorially get the transformation:

$$\mathcal{K}^{\mathrm{rat}}_{m+1}(A, \boldsymbol{v}, (\xi_1, \ldots, \xi_m)) \xrightarrow[p \text{ shifted RQZ steps}]{q(z)=\prod_{i=1}^{p} \frac{z-\varrho_i}{z-\xi_i}} \mathcal{K}^{\mathrm{rat}}_{m-p+1}(A, q(A)\boldsymbol{v}, (\xi_{p+1}, \ldots, \xi_m)),$$

which is the rational Krylov equivalent of implicitly restarted Arnoldi discussed in Section 7.1.

The implicit filter, and more broadly the rational QZ method, can also be formulated and implemented in terms of elementary operations on core transformations. This formulation is studied in [20].

7.4 Numerical experiments

A general approach for a restarted rational Krylov method is listed in Algorithm 3. This algorithm leaves open two major questions.

The first question is an approach to select the poles during the expansion phase. The rational Krylov method allows for plenty of freedom in this respect. If one has no a priori knowledge about the problem at hand, the extended Krylov method can be a good alternative as it contains fewer parameters. This is especially true during the first iterations, afterwards a motivated choice of poles might be made based on information already available. If eigenvalues in a certain region of interest are searched after, the poles can be chosen in such a way that they form a rational filter which emphasizes the eigenvalues inside

Algorithm 3 Restarted rational Krylov algorithm

Input: $A \in \mathbb{C}^{N\times N}$, $\mathbf{0} \neq \boldsymbol{v} \in \mathbb{C}^N$, maximal subspace dimension m, restart length p, number of desired Ritz pairs l $(p + l \leq m)$
Output: $\{(\vartheta_k, \boldsymbol{x}_k)\}_{k=1}^{l}$

1: Start:
   a: Select poles Ξ_m
   b: $[V_{m+1}, \underline{K}_m, \underline{L}_m] \leftarrow \mathtt{RK}(A, \boldsymbol{v}, \Xi_m)$ ▷ Algorithm 2.
   c: Check convergence of l *most* desired Ritz pairs $\{(\vartheta_k, \boldsymbol{x}_k)\}_{k=1}^{l}$
2: **while** not converged **do**
3: Select p shifts $(\varrho_k)_{k=1}^{p}$
4: **for** $j = 1 \ldots p$ **do**
5: $[V_{m-j+1}, \underline{K}_{m-j}, \underline{L}_{m-j}] \leftarrow \mathtt{RKQZ}(V_{m-j+2}, \underline{K}_{m-j+1}, \underline{L}_{m-j+1}, \varrho_j)$ ▷ Pole swapping filter.
6: **end for**
7: Select $m - p$ new poles Ξ_{m-p}
8: Expand: $[V_{m+1}, \underline{K}_m, \underline{L}_m] \leftarrow \mathtt{RK}(A, V, \underline{K}, \underline{L}, \Xi_{m-p})$ ▷ Algorithm 2.
9: Check convergence of l *most* desired Ritz pairs $\{(\vartheta_k, \boldsymbol{x}_k)\}_{k=1}^{l}$
10: **end while**

the region of interest, see [119] for a detailed description and a connection with contour integration techniques. In Example 7.4.3 we will use this approach to compute eigenvalues inside a contour. We will not go into further detail on the problem of pole selection.

A second issue is how to pick the shifts for the filter polynomial. Different practices have been proposed in the literature. They all attempt to create a filter polynomial $p_f \in \mathcal{P}_p$ that has the property that $|p_f(z)|$ is large on Ω_{wanted} and small on Ω_{unwanted}, where Ω_{wanted} and Ω_{unwanted} are disjoint compact sets in $\mathbb{C}$. A first method is the use of *exact* shifts [111]. These are the p Ritz values that are most distant from Ω_{wanted}. Another option is to use shifts as the zeros of Chebyshev polynomials on an ellipse [103, 111]. The use of *Leja* shifts, proposed in [5, 6], is a third possibility.

Example 7.4.1. In the first experiment, we use Algorithm 3 to determine the rightmost eigenvalues of a small test problem using extended and rational Krylov subspaces and exact shifts for the filter. The exact shifts are the leftmost Ritz values. We consider a matrix $A \in \mathbb{R}^{102\times 102}$ which is nonzero in the first 100 diagonal entries and in the last 2×2 block only. The diagonal entries are equal to $-100, -99, \ldots, -1$ and the 2×2 block leads to the complex conjugate pair of eigenvalues $\pm 25i$. This construction mimics the physical situation in the double-diffusive convection example [22, 114]. The spectrum of A is shown in Figure 7.4.

Figure 7.4: Spectrum of the problem of size 102 in Example 7.4.1

The rightmost eigenvalues of this matrix are $\pm 25i$. Assume we can only store a maximum of m=8 basis vectors in memory. For the restart phase we choose the parameter p=6. The starting vector is $[1 \cdots 1]^T$ and the iteration is repeated until the complex conjugate pair of Ritz values has converged to an error smaller than 10^{-8}.

Figure 7.5 shows the convergence of the desired Ritz values for 3 different options of poles Ξ. The error $|\lambda_{1,2} - \vartheta|/|\lambda_{1,2}|$ is shown in function of the dimension of the subspace. The left pane shows the result for $\Xi_1 = (0, 0, \ldots)$, meaning that only operations with A^{-1} are used and we have an extended Krylov subspace. We observe that the convergence for the complex conjugate pair is slow and 5 restarts are required to meet the convergence criterion. The middle pane shows the convergence for $\Xi_2 = (\infty, \infty, \ldots)$, a polynomial Krylov subspace. The convergence is much faster in this case and only 3 restarts are required since the error is reduced by approximately two orders of magnitude after every restart.

Considering the spectrum of A, this is what one would expect. The complex conjugate pair of eigenvalues is situated at $\pm 0.04i$ for A^{-1}. They lie in the cluster of eigenvalues near zero and are not well separated. This has a large impact on the convergence of the method. In the spectrum of the original matrix A the complex conjugate pair of eigenvalues is well separated. Hence the more rapid convergence with Ξ_2.

The right pane displays the result for a fully rational pole selection strategy. The initial subspace is constructed using the rational Krylov pole tuple $(-70.5, -60.5, \ldots, -10.5)$ with poles along the negative real axis. As this does not lead to significant convergence, the pole tuple is changed to $(22i, -22i, 16i, -16i, 10i, -10i)$ after the first restart. These poles along the imaginary axes speed up the convergence and only two restarts are required with this strategy.

Example 7.4.2. We study the benchmark problem from [33]. This problem also stems from fluid dynamics and is a model for the flow in a unit-square cavity with the lid moving from left to right. The Q_2 — Q_1 finite element discretization with IFISS [35] resulted in a generalized eigenvalue problem

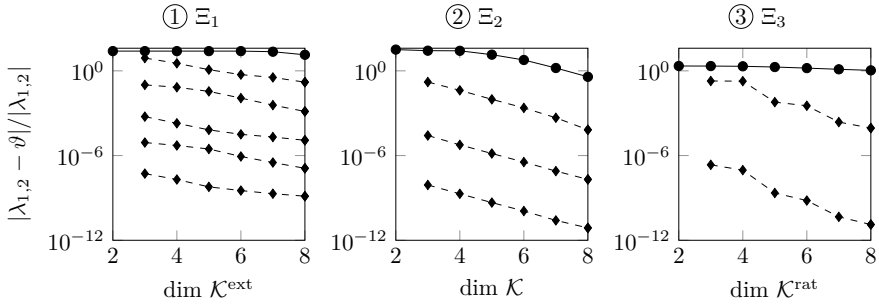


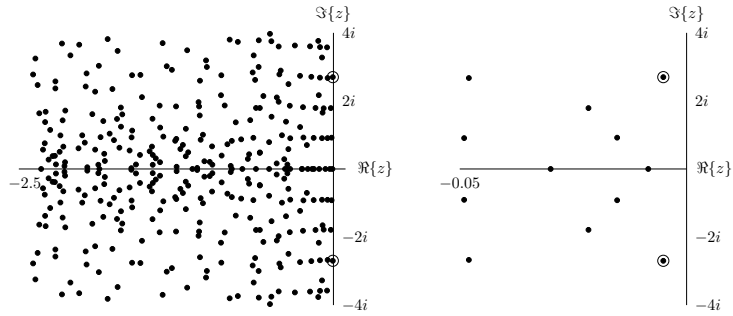
Figure 7.5: Convergence behavior for the restarted rational Krylov iteration for three different choices of Ξ. The convergence of the initial subspace is shown with a solid line, convergence after restarting is indicated with a dashed line. *Left:* extended, *Middle:* polynomial, *Right:* rational Krylov iteration.

$(A, B) \in \mathbb{R}^{9540\times 9540}$. The Reynolds number Re is 7800 for the pencil we consider. The critical Reynolds number of this problem Re_c is slightly less than 7929 [33, 36]. The pencil we consider is thus stable.

Both matrices A and B of the matrix pencil (A, B) are nonsingular such that we can apply an extended Krylov method for the generalized eigenvalue problem. This leads to operations with $A\,B^{-1}$ for poles at ∞ and with $B\,A^{-1}$ for poles at 0. The LU factorization of A takes 142s in Matlab and 27s for B on an Intel Xeon CPU E5-2697. It is feasible to factorize both A and B once, but repeating this every few iterations is costly. Hence we prefer the extended Krylov method over the rational Krylov method.

Figure 7.6 shows the spectrum of (A, B). The left pane shows 343 eigenvalues in a region of the complex plane near the imaginary axis. The rightmost eigenvalues of (A, B) appear in the complex conjugate pair $\lambda_{1,2} = -0.005135 \pm 2.698447i$. They are encircled in Figure 7.6. The right pane provides a closeup of the region near $\lambda_{1,2}$.

Table 7.1 lists the results for 3 different experiments with 3 different choices of Ξ_{ext}. These are cyclic pole tuples and the first column of Table 7.1 lists the first cycle in Ξ_{ext}. The ratio of poles at 0 decreases from the first to the third row as is indicated in the second column which shows the ratio of operations with poles at 0 with the total number of operations. The third column gives the requested tolerance for convergence, the fourth the number of restarts and the last column the residual of the rightmost Ritz values.



(a) 343 eigenvalues of (A, B) (b) 12 eigenvalues of (A, B)

Figure 7.6: The spectrum of the driven cavity problem. The encircled eigenvalues are the rightmost eigenvalues.

The residual is in this case evaluated as,

$$\frac{\|Ax - \lambda Bx\|_\infty}{\|A\|_\infty + |\lambda| + \|B\|_\infty}.$$

In all three experiments, we create a subspace of dimension m=100, which is then reduced with p=50 exact (leftmost) shifts during the restart. Since this is a rather 'difficult' problem, the dimension of the subspace is kept comparatively large. In order to retrieve the rightmost eigenvalues, the convergence criterion is applied to the 12 rightmost Ritz values. If all 12 have a residual less than the tolerance, the algorithm is halted. The tolerance is adjusted for each Ξ_{ext} in such a way that a good result is achieved within a reasonable number of restarts.

The results indicate that retrieving the rightmost eigenvalues of this problem up to good accuracy is feasible with a small number of restarts. The first two choices of Ξ_{ext} give a significantly faster convergence than the third. When the tolerance in the third experiment is lowered to 10^{-8}, the method fails to converge in a reasonable number of restarts. We conclude that for this problem it is beneficial to include operations with pole at 0.

The ARPACK [76] implementation of implicitly restarted Arnoldi, which is available in Matlab as the command `eigs`, did not retrieve the rightmost eigenvalues. This experiment demonstrates that the extended Krylov method can sometimes be a suitable choice for finding a few eigenvalues of a matrix if two conditions are satisfied. First, the convergence of the polynomial Krylov method is too slow to find the eigenvalues of interest within a reasonable number of restarts and with subspaces of small enough dimensions. Second, the computational cost of computing an LU factorization of the matrix is too

large to repeat every few iterations, which excludes the rational Krylov method as a viable option, but it is small enough to do once. This second condition leaves both the extended Krylov method and shift-and-invert Arnoldi as suitable options since they both require only one matrix factorization.

Ξ_{ext}	$\frac{\#BA^{-1}\text{op.}}{\text{all op.}}$	tolerance	restarts	residual norm
$\infty \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \cdots$	4/5	$3 \cdot 10^{-10}$	12	$9.3 \cdot 10^{-12}$
$\infty \;\; 0 \;\; 0 \;\; 0 \;\; \cdots$	3/4	$8 \cdot 10^{-10}$	9	$2.7 \cdot 10^{-11}$
$\infty \;\; 0 \;\; 0 \;\; \cdots$	2/3	$1.5 \cdot 10^{-8}$	6	$1.4 \cdot 10^{-8}$

Table 7.1: Summary of the results of Algorithm 3 on the cavity flow model with m=100, p=50 and $v = [1 \cdots 1]^T$ with three different options of Ξ_{ext}. The convergence is checked for the 12 rightmost Ritz values. The first column specifies the first cycle of Ξ_{ext}, the second column lists the fraction of poles at 0 in Ξ_{ext}, the third column gives the requested tolerance, the fourth column the number of restarts and the last column the residual norm upon convergence.

Example 7.4.3. In this example, we make a direct comparison between the results obtained with Algorithm 3 and the explicit QZ step of [27] which is listed in Algorithm 4. In line 2 of Algorithm 4 an orthogonal matrix $Z \in \mathbb{C}^{m \times m-1}$ is computed for which the vector $(\gamma \underline{L}_m^* - \delta \underline{K}_m^*)\boldsymbol{q}$ is in the nullspace of Z^*. This condition is not restrictive and does not define Z uniquely. Two choices for Z are used in our experiment: Z_1 as computed by Algorithm 6.1 of [27] and Z_2 computed from the full QR factorization $\begin{bmatrix} \boldsymbol{z} & Z_2 \end{bmatrix} \begin{bmatrix} \alpha \\ 0 \end{bmatrix} = (\gamma \underline{L}_m^* - \delta \underline{K}_m^*)\boldsymbol{q}$.

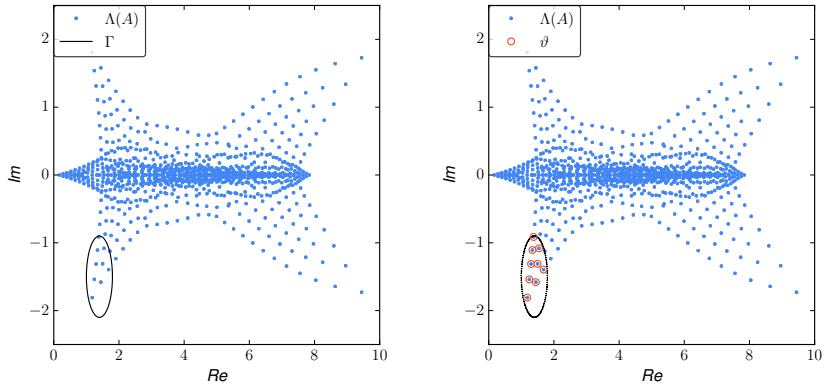
Algorithm 4 Single shift, explicit QZ step for rational Krylov [27]

Input: V_{m+1}, $\underline{K}_m$, $\underline{L}_m$, $\varrho = \gamma/\delta$
Output: $\check{V}_m$, $\underline{\check{K}}_{m-1}$, $\underline{\check{L}}_{m-1}$

1: Compute full QR factorization $\begin{bmatrix} \check{Q} & \boldsymbol{q} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} := \delta \underline{L}_m - \gamma \underline{K}_m$
2: Compute Z satisfying $\boldsymbol{q}^*(\bar{\gamma} \underline{L}_m - \bar{\delta} \underline{K}_m)Z = 0$
3: $\underline{\check{K}}_{m-1} := \check{Q}^* \underline{K}_m Z$
4: $\underline{\check{L}}_{m-1} := \check{Q}^* \underline{L}_m Z$
5: $\check{V}_m = V_{m+1} \check{Q}$

The matrix we consider is `PDE900` from the MatrixMarket collection. This is a real matrix of size 900×900. We are interested in determining the 9 eigenvalues of this matrix inside the elliptical contour Γ, shown in Figure 7.7(a). For this purpose, the contour is discretized with $N = 110$ points and both the poles Ξ and filter shifts ϱ are located at these discretization nodes. For more details on this choice of rational filter and connections with contour integration methods,

see [119]. This approach has a significant advantage over contour integration techniques as it only requires the solution to a single linear system in every iteration.



(a) Spectrum and continuous contour. (b) Spectrum, discretized contour and Ritz values with implicit QZ

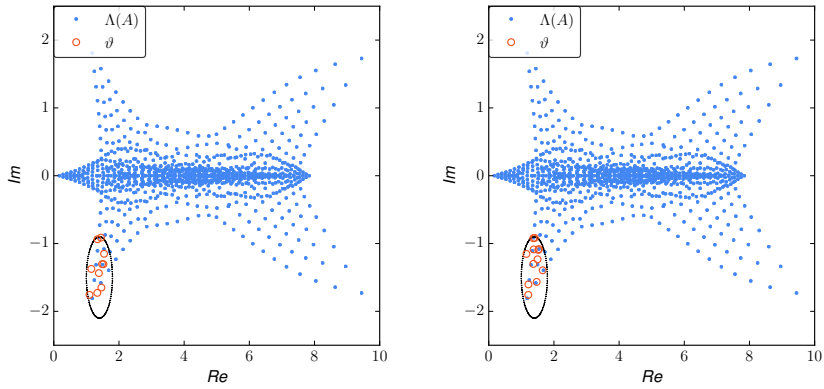
Figure 7.7: Problem setting and results with the implicit QZ method.

Given that all poles are on a contour in $\mathbb{C}$, we are dealing with a proper rational Krylov iteration. The tolerance is set to 10^{-7}. After 3 outer iterations of adding N poles and applying N filter shifts with the implicit QZ step, the problem can be deflated. The iteration found an invariant subspace containing the Ritz values of interest which are shown in Figure 7.7(b).

With the explicit QZ step of Algorithm 4, deflation does not occur or goes unnoticed and after the maximum of 6 outer iterations the Ritz values obtained with the choice of Z_1 are shown in Figure 7.8(a) and in Figure 7.8(b) for Z_2. Clearly, the method did not converge and the explicit QZ step distorts the information in the rational Krylov subspace.

This example demonstrates that the implicit QZ step is superior to the explicit step. Not only is it computationally more efficient, it behaves more stable and allows for accurate deflation monitoring.

Example 7.4.4. As our final numerical experiment, we revisit the two fluid flow problems from Section 3.5.3. Instead of computing all eigenvalues we are now only interested in determining if the problems are stable by computing the rightmost eigenvalues. The settings of Algorithm 3 and results obtained with the algorithm are summarized in Table 7.2. In both cases, we selected poles along the imaginary axis, $\Xi = (-20i, -18i, \ldots, 18i, 20i)$, as we expect the



(a) Spectrum, discretized contour and Ritz values with explicit QZ with option Z_1.

(b) Spectrum, discretized contour and Ritz values with explicit QZ with option Z_2.

Figure 7.8: Results with the explicit QZ method.

rightmost eigenvalue to be situated close to it. Exact shifts were used in the filter step.

Table 7.2: Summary of the settings and results of the restarted rational Krylov iteration. The columns list the maximal subspace dimension m, the restart length p, the number of wanted Ritz values ℓ, the tolerance `tol`, and the required number of restarts to reach convergence.

Problem	m	p	ℓ	`tol`	# restarts
Cavity flow	40	20	8	10^{-7}	8
Obstacle flow	60	25	7	10^{-7}	11

Figure 7.9 shows the rightmost part of the spectrum and the converged Ritz values. As can be seen, the method successfully converged to the correct eigenvalues within a reasonable number of restarts.

7.5 Conclusion

In this chapter we have applied the pole swapping technique of the rational QZ method to implicitly filter a rational Krylov iteration. Lemma 7.2.5 derived a small-scale Hessenberg pair which satisfies a Galerkin constraint with respect to a subspace computed with the rational Krylov method. This shows how

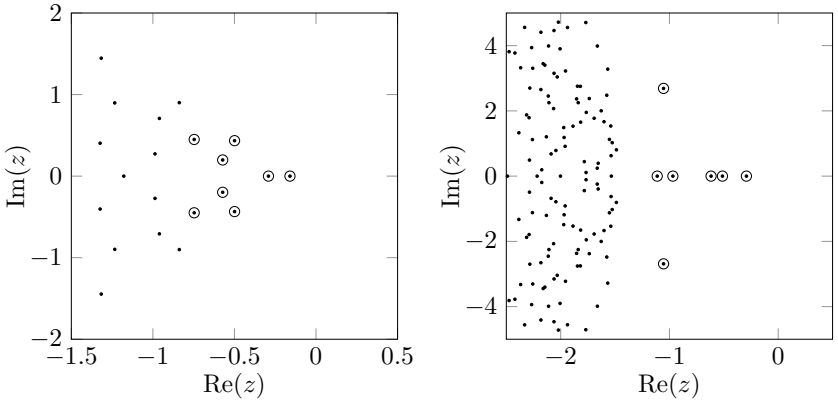


Figure 7.9: Rightmost part of the spectrum of the cavity flow (left) and obstacle flow (right) problems. The eigenvalues (·) and Ritz values (○) are shown.

eigenvalue approximations can be obtained using the rational Krylov method in combination with the rational QZ method. We studied the Galerkin projection on a rational Krylov subspace in Lemma 7.2.6 which revealed that it is of rational Hessenberg form.

Numerical experiments tested the filter algorithm on four examples and demonstrated the validity of the implicit approach both for the extended and rational Krylov methods. We showed how extended Krylov can, in particular cases, be an interesting method for the computation of the rightmost eigenvalues. We compared our method with ARPACK and with the implicit restart method proposed in [27], and showed that the new method can outperform these in some scenarios.

Chapter 8

Conclusions and outlook

In this thesis we presented a class of eigenvalue methods that are founded on a pole swapping strategy. We have shown that this change in perspective from classical bulge chasing algorithms allows for more general shifting strategies that lead us to more efficient eigenvalue algorithms.

We developed a sound theoretical understanding of pole swapping algorithms: both their uniqueness and their convergence has been analyzed using rational Krylov theory. This revealed that pole swapping methods implicitly perform nested subspace iteration accelerated by rational functions, thereby significantly generalizing the well-known polynomial-driven nested subspace iteration of bulge chasing algorithms.

We adapted many recent developments in dense eigenvalue solvers to our novel framework. These include tightly-packed, multishift and multipole batches and an aggressive early deflation strategy. We obtained a high-performing algorithm for the generalized eigenvalue problem which is rich in level-3 BLAS operations and shown to be more accurate and faster than LAPACK [2].

A compact representation of Hessenberg, unitary Hessenberg pencils enabled us to apply the pole swapping techniques for the solution of the standard eigenvalue problem in an efficient manner. Numerical experiments showed promising results with this approach to solve the standard eigenvalue problem.

We studied pole swapping algorithms for both symmetric and unsymmetric tridiagonal pencils using non-unitary transformations. Optimality conditions for the swapping transformations are provided, and uniqueness and convergence results are extended to the tridiagonal case. Numerical experiments show

promising results, but numerical stability remains challenging.

We used the connection between pole swapping algorithms and the rational Krylov method in order to efficiently filter the iterative rational Krylov method.

8.1 Contributions

In this section we wish to highlight the main contributions of the different chapters over existing results.

The contents of Chapter 3 have been submitted for publication [19]. The main contributions are:

- Section 3.2 rigorously defined the notion of properness for Hessenberg pairs. Lemma 3.2.2 gave four results for proper Hessenberg pairs that are useful for their theoretical understanding.
- Section 3.3.2 reviewed the pole introduction and swapping operations on Hessenberg pairs and provided a novel numerical algorithm to compute the swapping transformations which is backward stable. An error analysis is included in Appendix B.
- Section 3.4 proposed and tested a novel reduction algorithm to Hessenberg form with prescribed pole tuple. The numerical test showed that a good pole selection can induce premature middle deflations during the reduction process.
- Section 3.5 proposed and tested the novel rational QZ algorithm for Hessenberg pairs. Numerical tests showed that a good choice of poles allows the pole swapping method to significantly outperform bulge chasing methods.
- Section 3.6 reviewed and extended rational Krylov theory to prove an implicit Q theorem for proper Hessenberg pencils. In Theorem 3.6.4 we formally proved the shift invariance property for subspaces generated from elementary rational matrices. Theorem 3.6.7 is a new result that reveals the structure in rational Krylov subspaces generated from proper Hessenberg pairs.
- Section 3.7 contains the main theoretical result of this thesis in the form of Theorem 3.7.3 which essentially shows that a pole swapping method implicitly performs nested subspace iteration accelerated by rational functions.

- Section 3.8 provides an exactness result which shows that the rational QZ method with perfect shift or pole leads to a deflation.

The majority of the content from Chapter 4 has been submitted for publication in [18]. Section 4.4 contains results submitted for publication in [17]. The main contributions are:

- Section 4.2 formally defined and studied proper block Hessenberg pairs.
- Theorem 4.2.8 showed the *blocked structure* of rational Krylov subspaces generated by proper block Hessenberg pairs used to prove the block implicit Q theorem in Theorem 4.3.3.
- Section 4.2.3 studied the pole placement problem for blocks of poles.
- Section 4.4 reviewed the swapping problem and proposed a numerical scheme for the iterative refinement of pole swaps.
- Section 4.5 studied how an aggressive early deflation strategy can be applied within the rational QZ method.
- The implementation of the algorithm in the Fortran package `libRQZ`.

The content of Chapter 5 is based on an article that is currently in preparation. The main contributions are:

- An efficient storage scheme for Hessenberg, unitary Hessenberg pencils which allows for a backward stable pole swapping algorithm.
- A comparison between our pole swapping method `ZLAHPS` and the bulge chasing method `ZLAHQR` from LAPACK that reveals a reduction in CPU time up to 37%.

The content of Chapter 6 is based on an article that is currently in preparation. The main contributions are:

- A study of (nearly) optimally scaled, non-unitary pole swapping methods for block tridiagonal pencils that preserve the tridiagonal form which results in an $O(n^2)$ eigenvalue algorithm for tridiagonal pencils.
- Section 6.6.1 extends the uniqueness and convergence results of Chapter 3 to the non-unitary case.

The content of Chapter 7 is based on [20]. The main contributions are:

- Lemma 7.2.5 presents a novel manner to compute the Ritz values in rational Krylov as the eigenvalues of a Hessenberg pair.
- Lemma 7.2.6 analyzes the structure of the Galerkin projection on a rational Krylov subspace.
- Sections 7.3 and 7.4 formulated and tested an implicit filter for rational Krylov and showed that it can outperform existing methods for computing a subset of eigenvalues.

8.2 Outlook

Scientific research is a never ending process. This thesis answered some questions regarding pole swapping methods for eigenvalue problems but also opens up potential new research directions. We list some ideas below.

- Throughout the thesis we mainly relied on pole selection strategies that use eigenvalue approximations based on the top-left part of the matrix. The numerical experiment in Section 3.4.2 gave an example where a different pole selection strategy allowed us to construct a rational filter which induces a middle deflation that splits the problem in two large regions of eigenvalues. We believe that there is significant potential in novel shifting and pole strategies which construct good rational filters for the use in a pole swapping algorithm and that this requires further research.
- We devised a way to incorporate the aggressive early deflation strategy in the multishift, multipole rational QZ step. The computational cost of aggressive early deflation is, although limited compared with a rational QZ sweep, still significantly larger than classical deflation criteria. It would be interesting to be able to predict a priori if aggressive early deflation will find a significant number of eigenvalues, similar to [15].
- The idea of aggressive early deflation has also been investigated to perform *aggressive middle deflation* in QR-type algorithms [81]. If this strategy can be combined with rational filters that tend to induce middle deflations, it can have the potential to drastically speed-up eigenvalue computations.
- With respect to software development, it would be interesting to develop a parallel implementation of the multishift, multipole rational QZ method with active bidirectional chasing.

- An extension of the rational QR method based on the Hessenberg, unitary Hessenberg pole swapping technique to multishift, multipole sweeps that preserve the compact format.
- A way to perform aggressive early deflation on the compact Hessenberg, unitary Hessenberg form.
- Investigating if the RLR or RT3 pole swapping algorithms for tridiagonal pencils can be shown to be numerically stable for certain classes of problems or certain shifting strategies.
- A study of balancing techniques with the goal of improving numerical stability of the RLR and RT3 methods.
- An investigation of the adaptation of pole swapping techniques to structured eigenproblems.

Appendix A

Core transformations and the extended Hessenberg form

Section A.1 of this appendix provides an overview of three operations that can be performed with core transformations as defined in Definition 2.3.2. These operations are extensively used in the literature in the context of generalizations of the QR method [123, 129, 130] and for the representation of *rank-structured* matrices. An in-depth overview of core transformations and their use for the eigenvalue problem can be found in [4]. Section A.2 introduces the concept of *extended Hessenberg* matrix and pencils.

A.1 Three operations on core transformations

Three useful operations with core transformations are the *transfer*, *fusion*, and *turnover* operations. The transfer of a core transformation from left to right, or vice versa, through a nonsingular, upper triangular matrix is shown in Figure A.1.

Elements of the upper triangular matrix that are altered during the transfer from left to right are indicated with $\otimes$ in Figure A.1. The core transformation on the left is different from the one on the right but its index is not changed and the upper triangular shape is preserved. The computational complexity of a transfer operation is clearly $O(n)$, with n the dimension of the upper triangular matrix. If multiple core transformations are present in a given pattern or shape, for example the *descending* pattern of a Hessenberg matrix (Figure 2.4), then

Figure A.1: Transfer of a core transformation from the left of an upper triangular matrix to the right or vice versa.

the complete pattern of transformations can be transferred through the upper triangular matrix. This operation preserves the mutual ordering of the core transformations. An example is shown in Figure A.2.

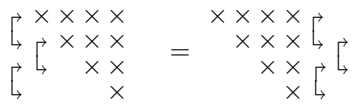
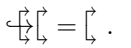


Figure A.2: The ordering of core transformations is preserved under the transfer operation.

Two core transformations that act consecutively on the same rows or columns can be multiplied and the result is again a core transformation. This is called a fusion of core transformations and is depicted as:



The turnover of a *V-shaped* pattern of 3 core transformations is shown in Figure A.3. This operation flips a factorization of 3 core transformations that act on rows $(i, i{+}1)$, $(i{+}1, i{+}2)$, $(i, i{+}1)$ into a factorization acting on rows $(i{+}1, i{+}2)$, $(i, i{+}1)$, $(i{+}1, i{+}2)$ or vice versa. A turnover is always possible in the unitary case. This can be proven by considering two variants for factoring a unitary $3{\times}3$ matrix [125]. The computational complexity of both the fusion and turnover operation is $O(1)$.

Finally, we remark that two core transformation C_i and C_j commute if $|i{-}j| > 1$. As a consequence the mutual ordering in a pattern of core transformations is not necessarily unique. We say that consecutive core transformations, C_i and C_{i+1}, are in *descending* order if they are ordered as $C_i\, C_{i+1}$. If they are ordered as $C_{i+1}\, C_i$, we say that they are in ascending order. The Hessenberg form has all core transformations in descending order and can be considered as a *strictly* descending pattern.

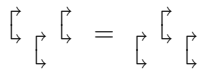


Figure A.3: Turnover of a V-shaped pattern of core transformations.

A.2 Extended Hessenberg matrices and pencils

The extended Hessenberg form is a relaxation of the Hessenberg form in the sense that the sequence of core transformations is no longer required to be strictly descending. Definition A.2.1 formalizes this and furthermore defines the notion of an *extended Hessenberg pencil*.

Definition A.2.1. A matrix $A \in \mathbb{F}^{n\times n}$ is called an *extended Hessenberg* matrix if it has a QR decomposition of the form:

$$A = C_{\pi(1)} \dots C_{\pi(n-1)} R,$$

with π a permutation of $(1, \dots, n-1)$. A pair of matrices $A, B \in \mathbb{F}^{n\times n}$ is called an *extended Hessenberg* pair if the matrices have a QR decomposition:

$$A = Q_A R_A,\ B = Q_B R_B, \quad \text{with:} \quad Q_A Q_B = C_{\pi(1)} \dots C_{\pi(n-1)},$$

where π is again a permutation of $(1, \dots, n-1)$. An extended Hessenberg matrix and an extended Hessenberg pair are called *proper* if none of the core transformations C_i are diagonal.

In order to clarify Definition A.2.1, Figure A.4 provides examples of a Hessenberg matrix, an extended Hessenberg matrix, and an extended Hessenberg pencil. As the ordering of core transformations is not unique if they do not act on consecutive rows, there are multiple alternative orderings for the extended forms in panes II and III.

We observe that the extended Hessenberg pencil, $(C_2C_1C_5R_A, C_4C_3R_B)$, from pane III can be easily transformed to an equivalent extended Hessenberg pencil with *strictly descending* patterns of core transformations in A and B. Strictly descending means that the core transformations in both Q_A and Q_B satisfy the ordering C_iC_j if $j > i$. The transformation to the strictly descending form can be achieve *without* changing the upper triangular matrices R_A or R_B. For our example, this can be done by a left multiplication with $C_2^*C_4^*$ which gives the equivalent pencil $(C_1C_4^*C_5R_A, C_2^*C_3R_B)$.

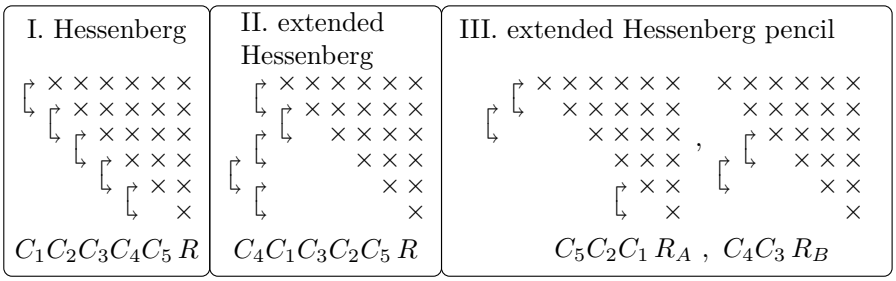


Figure A.4: Examples of a Hessenberg matrix, extended Hessenberg matrix and extended Hessenberg pencil.

Similarly, an extended Hessenberg matrix can also be transformed to an equivalent extended Hessenberg pencil with strictly descending patterns of core transformations. For example, the extended Hessenberg matrix in pane II of Figure A.4 can be interpreted as the pencil $(C_4C_1C_3C_2C_5R, I)$ which can be left multiplied with $C_3^*C_4^*$ to get the equivalent extended Hessenberg pencil $(C_1C_2C_5R, C_3^*C_4^*)$. This has a strictly descending pattern of core transformations on both matrices. Observe that the second matrix in the equivalent matrix pair remains unitary in this case.

The previous two observations are true in general and are formalized in the next lemma.

Lemma A.2.2. *Any proper extended Hessenberg pencil, (Q_AR_A, Q_BR_B), can be transformed to an equivalent* strictly descending *extended Hessenberg pencil, $(\tilde{Q}_AR_A, \tilde{Q}_BR_B)$, where both $\tilde{Q}_A$ and $\tilde{Q}_B$ are unitary matrices that admit a factorization in a strictly descending sequence of core transformations by left multiplication with a unitary matrix. This does not alter the upper triangular matrices R_A and R_B.*

Proof. Trivial observation. □

A direct corollary of Lemma A.2.2 is that any proper extended Hessenberg matrix and pencil can be transformed to an equivalent proper Hessenberg pencil according to Definition 3.2.1 with poles $\xi_i \in \{0, \infty\}$. Properness of the resulting Hessenberg pair follows from the property that the core transformations are non-diagonal.

Starting from an extended Hessenberg matrix results in a Hessenberg pair with a unitary matrix such that the method of Chapter 5 can be used. For

extended Hessenberg pencils, we can use the method of Chapter 3 after the transformation.

As shown in Lemma 7.2.6, the extended Hessenberg form is linked with the Galerkin projection on an extended Krylov subspace.

Appendix B

Backward stable pole swapping

In this appendix we present the error analysis for Lemma 3.3.2 and the results of numerical experiments supporting the analysis.

B.1 Error analysis

The swapping operation is a unitary equivalence, and such transformations generally are stable [55], but there is one thing we have to check. The rotation Q is designed so that $Q^*(BZ)$ has a zero in the $(2,1)$ position. This automatically creates a zero in the $(2,1)$ position of $Q^*(AZ)$ because the first columns of AZ and BZ are scalar multiples. This is true in exact arithmetic. We just need to check that in floating-point arithmetic the entry that is created in the $(2,1)$ position of Q^*AZ is small enough that backward stability is not compromised by setting it to zero. For this it suffices that its magnitude be no bigger than a modest multiple of $\epsilon_m \| A \|$. Here and throughout the analysis, the norm symbol will denote the 2-norm.

The swapping operation begins with the computation of Z in (3.4) which is uniquely defined by the vector

$$\boldsymbol{x} = \begin{bmatrix} \alpha_2 b - \beta_2 a \\ \beta_2 \alpha_1 - \alpha_2 \beta_1 \end{bmatrix}, \tag{B.1}$$

which is a right eigenvector of $A - \lambda B$ in (3.3) associated with ξ_2. In floating-point arithmetic we get

$$\mathrm{fl}(\boldsymbol{x}) = \begin{bmatrix} \alpha_2 b(1+\epsilon_1) - \beta_2 a(1+\epsilon_2) \\ \beta_2\alpha_1(1+\epsilon_3) - \alpha_2\beta_1(1+\epsilon_4) \end{bmatrix}, \tag{B.2}$$

where each ϵ_i is the result of two roundoff errors and therefore satisfies $|\,\epsilon_i\,| \le 2\epsilon_m + O(\epsilon_m^2)$. We will use the abbreviation $|\,\epsilon_i\,| \lesssim \epsilon_m$ to mean that $|\,\epsilon_i\,|$ is no bigger than a modest constant times ϵ_m.

The next step is to actually compute Z, which has $\boldsymbol{x}/\|\,\boldsymbol{x}\,\|$ as its first column. In practice we do this using $\mathrm{fl}(\boldsymbol{x})$ and make additional roundoff errors in the computation. We get $\tilde{Z} = \mathrm{fl}(Z)$ satisfying

$$\tilde{Z}\boldsymbol{e}_1 = \tilde{\boldsymbol{x}} = \tilde{\gamma}^{-1} \begin{bmatrix} \mathrm{fl}(x_1)(1+\epsilon_5) \\ \mathrm{fl}(x_2)(1+\epsilon_6) \end{bmatrix}. \tag{B.3}$$

Here $\tilde{\gamma} = \|\,\mathrm{fl}(\boldsymbol{x})\,\|$. A tiny relative error is made during this norm computation, and another tiny error is made when $\mathrm{fl}(x_1)$ is divided by $\tilde{\gamma}$. These are the causes of the error ϵ_5, and we have $|\,\epsilon_5\,| \lesssim \epsilon_m$. Similarly $|\,\epsilon_6\,| \lesssim \epsilon_m$.

The vector $\tilde{\boldsymbol{x}}$ defined by (B.3) is our computed (and normalized) version of a right eigenvector associated with eigenvalue ξ_2. For later use we wish to show that $\tilde{\boldsymbol{x}}$ is exactly an eigenvector of a slightly perturbed pencil. Thus we seek perturbed quantities $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\beta}_1$, and $\tilde{\beta}_2$ such that

$$\left(\tilde{\beta}_2 \begin{bmatrix} \tilde{\alpha}_1 & a \\ & \tilde{\alpha}_2 \end{bmatrix} - \tilde{\alpha}_2 \begin{bmatrix} \tilde{\beta}_1 & b \\ & \tilde{\beta}_2 \end{bmatrix}\right) \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{B.4}$$

Notice that we are not going to map back any of the error onto a or b. This equation is equivalent to

$$(\tilde{\beta}_2\tilde{\alpha}_1 - \tilde{\alpha}_2\tilde{\beta}_1)\tilde{x}_1 + (\tilde{\beta}_2 a - \tilde{\alpha}_2 b)\tilde{x}_2 = 0.$$

Filling in the values of $\tilde{x}_1$ and $\tilde{x}_2$ from (B.3) and (B.2), we can check that this equation holds if we make the assignments

$$\tilde{\alpha}_1 = \alpha_1 \frac{(1+\epsilon_3)(1+\epsilon_6)}{(1+\epsilon_2)(1+\epsilon_5)}, \qquad \tilde{\alpha}_2 = \alpha_2(1+\epsilon_1)(1+\epsilon_5),$$

$$\tilde{\beta}_1 = \beta_1 \frac{(1+\epsilon_4)(1+\epsilon_6)}{(1+\epsilon_1)(1+\epsilon_5)}, \qquad \tilde{\beta}_2 = \beta_2(1+\epsilon_2)(1+\epsilon_5).$$

Clearly $|\,\tilde{\alpha}_i - \alpha_i\,| \lesssim \epsilon_m |\,\alpha_i\,|$ and $|\,\tilde{\beta}_i - \beta_i\,| \lesssim \epsilon_m |\,\beta_i\,|$ for $i = 1, 2$. Equation (B.4) can be written more compactly as

$$\tilde{\beta}_2 \tilde{A}\tilde{x} = \tilde{\alpha}_2 \tilde{B}\tilde{x}. \tag{B.5}$$

Thus $\tilde{\boldsymbol{x}}$ is an eigenvector of the perturbed pencil $\tilde{A} - \lambda\tilde{B}$ associated with eigenvalue $\tilde{\xi}_2 = \tilde{\alpha}_2/\tilde{\beta}_2$. We also write

$$\tilde{A} = A + \delta A \quad \text{and} \quad \tilde{B} = B + \delta B_1, \tag{B.6}$$

with δA and δB_1 diagonal matrices satisfying $\|\,\delta A\,\| \lesssim \epsilon_m \|\,A\,\|$ and $\|\,\delta B_1\,\| \lesssim \epsilon_m \|\,B\,\|$.

Finally we compute Q. In exact arithmetic Q is constructed so that $Q^*(BZ\boldsymbol{e}_1) = \gamma\,\boldsymbol{e}_1$, for some γ, so the first column of Q must be proportional to $BZ\boldsymbol{e}_1$. In practice, instead of $BZ\boldsymbol{e}_1$ we use

$$\check{\boldsymbol{y}} = \mathrm{fl}(B\tilde{Z}\boldsymbol{e}_1) = \mathrm{fl}(B\tilde{\boldsymbol{x}}) = \tilde{\gamma}^{-1} \begin{bmatrix} \beta_1\tilde{x}_1(1+\epsilon_1') + b\tilde{x}_2(1+\epsilon_2') \\ \beta_2\tilde{x}_2(1+\epsilon_3') \end{bmatrix},$$

where $|\,\epsilon_i'\,| \lesssim \epsilon_m$ for $i = 1,\ 2,\ 3$. The computed version of Q is $\tilde{Q} = \mathrm{fl}(Q)$ satisfying

$$\tilde{Q}e_1 = \check{\zeta}^{-1} \begin{bmatrix} \check{y}_1(1+\epsilon_4') \\ \check{y}_2(1+\epsilon_5') \end{bmatrix},$$

where $\check{\zeta} = \|\,\check{\boldsymbol{y}}\,\|$, and ϵ_4' and ϵ_5' are due to the tiny roundoff errors in the calculation.

For our analysis we need to establish that there is a slightly perturbed matrix

$$\hat{B} = B + \delta B_2 = \begin{bmatrix} \hat{\beta}_1 & b \\ & \hat{\beta}_2 \end{bmatrix}$$

such that $\tilde{Q}^*\hat{B}\tilde{Z}$ has an exact zero in the $(2,1)$ position. This means that $\tilde{\boldsymbol{y}} = \tilde{Q}\boldsymbol{e}_1$ is exactly proportional to $\hat{B}\tilde{Z}\boldsymbol{e}_1 = \hat{B}\tilde{\boldsymbol{x}}$. It is easy to check that the choice

$$\hat{\beta}_1 = \beta_1\frac{(1+\epsilon_1')}{(1+\epsilon_2')}, \qquad \hat{\beta}_2 = \beta_2\frac{(1+\epsilon_3')(1+\epsilon_5')}{(1+\epsilon_2')(1+\epsilon_4')}$$

does the trick. Clearly $|\,\hat{\beta}_1 - \beta_1\,| \lesssim \epsilon_m\,|\,\beta_1\,|$ and $|\,\hat{\beta}_2 - \beta_2\,| \lesssim \epsilon_m\,|\,\beta_2\,|$, and δB_2 is a diagonal matrix satisfying $\|\,\delta B_2\,\| \lesssim \epsilon_m\,\|\,B\,\|$.

Our final computed results are $\mathrm{fl}(\tilde{Q}^*A\tilde{Z})$ and $\mathrm{fl}(\tilde{Q}^*B\tilde{Z})$. We have to show that the $(2,1)$ entries of these matrices are small enough that we can set them to zero without compromising backward stability. The "B" part is routine. Focusing on the $(2,1)$ entry, we have

$$\boldsymbol{e}_2^T\,\mathrm{fl}(\tilde{Q}^*B\tilde{Z})\boldsymbol{e}_1 = \boldsymbol{e}_2^T\tilde{Q}^*B\tilde{Z}\boldsymbol{e}_1 + \boldsymbol{e}_2^TE_1\boldsymbol{e}_1,$$

where E_1 is the matrix of roundoff errors incurred in multiplying the three matrices together and satisfies $\|\,E_1\,\| \lesssim \epsilon_m\,\|\,\tilde{Q}\,\|\,\|\,B\,\|\,\|\,\tilde{Z}\,\|$, i.e. $\|\,E_1\,\| \lesssim \epsilon_m\,\|\,B\,\|$. The remaining term is

$$\boldsymbol{e}_2^T\tilde{Q}^*B\tilde{Z}\boldsymbol{e}_1 = \boldsymbol{e}_2^T\tilde{Q}^*\hat{B}\tilde{Z}\boldsymbol{e}_1 - \boldsymbol{e}_2^T\tilde{Q}^*\delta B_2\tilde{Z}\boldsymbol{e}_1.$$

The first term on the right-hand side is exactly zero by construction. The second is bounded above by $\|\,\delta B_2\,\| \lesssim \epsilon_m \,\|\,B\,\|$.

The "A" part is more delicate. We have

$$\boldsymbol{e}_2^T \mathrm{fl}(\tilde{Q}^* A \tilde{Z})\boldsymbol{e}_1 = \boldsymbol{e}_2^T \tilde{Q}^* A \tilde{Z}\boldsymbol{e}_1 + \boldsymbol{e}_2^T E_2 \boldsymbol{e}_1,$$

where E_2 is the matrix of roundoff errors incurred in multiplying the three matrices together and satisfies $\|\,E_2\,\| \lesssim \boldsymbol{e}_1 \,\|\,A\,\|$. The remaining term is

$$\boldsymbol{e}_2^T \tilde{Q}^* A \tilde{Z}\boldsymbol{e}_1 = \boldsymbol{e}_2^T \tilde{Q}^* \tilde{A} \tilde{Z}\boldsymbol{e}_1 - \boldsymbol{e}_2^T \tilde{Q}^* \delta A \tilde{Z}\boldsymbol{e}_1.$$

The second term on the right-hand side is bounded above by $\|\,\delta A\,\| \lesssim \epsilon_m \,\|\,A\,\|$, so now we can just focus on the other term. Here we make use of (B.5), which can be written as $\tilde{A}\tilde{Z}\boldsymbol{e}_1 = (\tilde{\alpha}_2/\tilde{\beta}_2)\tilde{B}\tilde{Z}\boldsymbol{e}_1$.

$$\boldsymbol{e}_2^T \tilde{Q}^* \tilde{A} \tilde{Z}\boldsymbol{e}_1 = \frac{\tilde{\alpha}_2}{\tilde{\beta}_2}\boldsymbol{e}_2^T \tilde{Q}^* \tilde{B} \tilde{Z}\boldsymbol{e}_1 = \frac{\tilde{\alpha}_2}{\tilde{\beta}_2}\boldsymbol{e}_2^T \tilde{Q}^* \hat{B} \tilde{Z}\boldsymbol{e}_1 + \frac{\tilde{\alpha}_2}{\tilde{\beta}_2}\boldsymbol{e}_2^T \tilde{Q}^* (\delta B_1 - \delta B_2) \tilde{Z}\boldsymbol{e}_1.$$

The term containing $\hat{B}$ is zero by construction, so now we just need to concentrate on the other term. Let $\delta B = \delta B_1 - \delta B_2$. From the definitions of δB_1 and δB_2 we see that

$$\delta B = \begin{bmatrix} \epsilon_1'' \,\beta_1 & 0 \\ 0 & \epsilon_2'' \,\beta_2 \end{bmatrix},$$

where $|\,\epsilon_i''\,| \lesssim \epsilon_m$ for $i = 1,\, 2$. Moreover $\dfrac{\tilde{\alpha}_2}{\tilde{\beta}_2} = \dfrac{\alpha_2}{\beta_2}(1 + \epsilon_3'')$ for some tiny ϵ_3''. We also use our assumption $|\,\xi_1\,| \geq |\,\xi_2\,|$ to deduce that $|\,\beta_1\alpha_2/\beta_2\,| \leq |\,\alpha_1\,|$. Thus

$$\begin{aligned} |\,(\tilde{\alpha}_2/\tilde{\beta}_2)\delta B\,| &= (1 + \epsilon_3'') \begin{bmatrix} |\,\epsilon_1'' \,\beta_1\alpha_2/\beta_2\,| & \\ & |\,\epsilon_2'' \,\alpha_2\,| \end{bmatrix} \\ &\leq (1 + \epsilon_3'') \begin{bmatrix} |\,\epsilon_1'' \,\alpha_1\,| & \\ & |\,\epsilon_2'' \,\alpha_2\,| \end{bmatrix}, \end{aligned}$$

so

$$\|\,(\tilde{\alpha}_2/\tilde{\beta}_2)\delta B\,\| \lesssim \epsilon_m \,\|\,A\,\|.$$

We conclude that our one remaining term, which is $(\tilde{\alpha}_2/\tilde{\beta}_2)\boldsymbol{e}_2^T \tilde{Q}^* (\delta B) \tilde{Z}\boldsymbol{e}_1$, satisfies

$$|\,(\tilde{\alpha}_2/\tilde{\beta}_2)\boldsymbol{e}_2^T \tilde{Q}^* (\delta B) \tilde{Z}\boldsymbol{e}_1\,| \lesssim \epsilon_m \,\|\,A\,\|.$$

We have demonstrated that

$$|\,\boldsymbol{e}_2^T \mathrm{fl}(\tilde{Q}^* A \tilde{Z})\boldsymbol{e}_1\,| \lesssim \epsilon_m \,\|\,A\,\| \quad \text{and} \quad |\,\boldsymbol{e}_2^T \mathrm{fl}(\tilde{Q}^* B \tilde{Z})\boldsymbol{e}_1\,| \lesssim \epsilon_m \,\|\,B\,\|,$$

so we can set these numbers to zero without compromising backward stability. The $\lesssim$ symbols hide constants, but these constants are not too large due to the small total number of operations required by the swap.

B.2 Numerical experiments

We generated sixty-four million random 2×2 upper triangular pencils where the six nonzero entries are approximately logarithmically distributed and vary in magnitude from 10^{-12} up to 10^{12}. For all pencils we computed the swapping transformations using three different algorithms: our method, the method of Van Dooren [122], and a method that solves the generalized Sylvester equation explicitly to determine Q and Z [60]. The computations were done in double precision arithmetic, for which $\epsilon_m \approx 10^{-16}$. Table B.1 shows that our method always produces residuals $|\,a_{21}\,|/\|\,A\,\|$ and $|\,b_{21}\,|/\|\,B\,\|$ that are under 10^{-15}, and more than 99.7% of them are under 10^{-16}. In contrast, the Van Dooren and Sylvester methods sometimes produce much larger residuals, approaching 10^0 in a few cases. If we change the criterion and consider the residuals $|\,a_{21}\,|/\Delta$ and $|\,b_{21}\,|/\Delta$, where $\Delta = \max\{\|\,A\,\|, \|\,B\,\|\}$, then all methods perform well, as Table B.2 shows. By this criterion all residuals are under 10^{-15}. Our method and Van Dooren's method perform about equally well, and the Sylvester method is almost as good. We conclude that if $\|\,A\,\|$ and $\|\,B\,\|$ are roughly the same, it doesn't matter which method is used. However, in problems for which there can be large differences in magnitude between $\|\,A\,\|$ and $\|\,B\,\|$, our method is better.

Table B.1: Distribution of errors $|\,\hat{a}_{21}\,|/\|A\|$ and $|\,\hat{b}_{21}\,|/\|B\|$ for our method, Van Dooren's method, and the Sylvester method.

$\lvert\,\hat{x}_{21}\,\rvert/\lVert X\rVert$		$\left[0, 10^{-16}\right]$	$\left(10^{-16}, 10^{-15}\right]$	$\left(10^{-15}, 10^{-10}\right]$	$\left(10^{-10}, 10^{-5}\right]$	$\left(10^{-5}, 10^{0}\right]$
Our method	A	99.71%	0.29%	0%	0%	0%
	B	99.85%	0.15%	0%	0%	0%
Van Dooren	A	98.19%	0.55%	0.93%	0.27%	0.06%
	B	98.19%	0.55%	0.93%	0.27%	0.06%
Sylvester	A	93.34%	5.88%	0.57%	0.17%	0.04%
	B	93.34%	5.88%	0.57%	0.17%	0.04%

Table B.2: Distribution of errors $|\hat{a}_{21}|/\Delta$ and $|\hat{b}_{21}|/\Delta$ for our method, Van Dooren's method, and the Sylvester method.

$\lvert\hat{x}_{21}\rvert/\Delta$		$\left[0, 10^{-16}\right]$	$\left(10^{-16}, 10^{-15}\right]$	$\left(10^{-15}, 10^{-10}\right]$	$\left(10^{-10}, 10^{-5}\right]$	$\left(10^{-5}, 10^{0}\right]$
Our method	A	99.87%	0.13%	0%	0%	0%
	B	99.93%	0.07%	0%	0%	0%
Van Dooren	A	99.94%	0.06%	0%	0%	0%
	B	99.94%	0.06%	0%	0%	0%
Sylvester	A	97.26%	2.74%	0%	0%	0%
	B	97.26%	2.74%	0%	0%	0%

Bibliography

[1] Web of Science webpage. `https://apps.webofknowledge.com`. Accessed: February 9, 2019.

[2] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. *LAPACK Users' Guide*, third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[3] Arnoldi, W. E. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quart. Appl. Math. 9* (1951), 17–29.

[4] Aurentz, J. L., Mach, T., Robol, L., Vandebril, R., and Watkins, D. S. *Core-Chasing Algorithms for the Eigenvalue Problem*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, 2018.

[5] Baglama, J., Calvetti, D., and Reichel, L. IRBL: an implicitly restarted block-Lanczos method for large-scale Hermitian eigenproblems. *SIAM J. Sci. Comput. 24*, 5 (2003), 1650–1677.

[6] Baglama, J., and Reichel, L. An implicitly restarted block Lanczos bidiagonalization method using Leja shifts. *BIT 53*, 2 (2013), 285–310.

[7] Barkouki, H., Bentbib, A., and Jbilou, K. A matrix rational Lanczos method for model reduction in large-scale first- and second-order dynamical systems. *Numer. Linear Algebra Appl. 24*, 1 (2016), e2077.

[8] Bart, H., Gohberg, I., Kaashoek, M. A., and Van Dooren, P. Factorizations of transfer functions. *SIAM J. Control 18*, 6 (1980), 675–696.

[9] Beckermann, B., Güttel, S., and Vandebril, R. On the convergence of rational Ritz values. *SIAM J. Matrix Anal. Appl. 31*, 4 (2010), 1740–1774.

[10] Berljafa, M. *Rational Krylov Decompositions: Theory and Applications.* PhD thesis, The University of Manchester, 2017.

[11] Berljafa, M., and Güttel, S. Generalized rational Krylov decompositions with an application to rational approximation. *SIAM J. Matrix Anal. Appl. 36*, 2 (2015), 894–916.

[12] Berljafa, M., and Güttel, S. Parallelization of the rational Arnoldi algorithm. *SIAM J. Sci. Comput. 39*, 5 (2017), S197–S221.

[13] Boisvert, R. F., Pozo, R., Remington, K., Barrett, R. F., and Dongarra, J. J. Matrix market: A web resource for test matrix collections. In *Proceedings of the IFIP TC2/WG2.5 Working Conference on Quality of Numerical Software: Assessment and Enhancement* (London, UK, 1997), Chapman & Hall, Ltd., pp. 125–137.

[14] Braman, K., Byers, R., and Mathias, R. The multishift QR algorithm. Part I: maintaining well-focused shifts and level 3 performance. *SIAM J. Matrix Anal. Appl. 23*, 4 (2002), 929–947.

[15] Braman, K., Byers, R., and Mathias, R. The multishift QR algorithm. Part II: aggressive early deflation. *SIAM J. Matrix Anal. Appl. 23*, 4 (2002), 948–973.

[16] Camps, D., Mach, T., Vandebril, R., and Watkins, D. S. On pole-swapping algorithms for the eigenvalue problem. Submitted.

[17] Camps, D., Mastronardi, N., Vandebril, R., and Van Dooren, P. Swapping 2×2 blocks in the Schur and generalized Schur form. Accepted for publication in J. Comput. Appl. Math.

[18] Camps, D., Meerbergen, K., and Vandebril, R. A multishift, multipole rational QZ method with aggressive early deflation. Submitted.

[19] Camps, D., Meerbergen, K., and Vandebril, R. A rational QZ method. *SIAM J. Matrix Anal. Appl. 40*, 3 (2019), 943–972.

[20] Camps, D., Meerbergen, K., and Vandebril, R. An implicit filter for rational Krylov using core transformations. *Linear Algebra Appl. 561*, January (2019), 113–140.

[21] Cauchy, A. L. Sur l'équation à l'aide de laquelle on determine les inégalités séculaires des mouvements des planètes. *Exer. de math. 2*, 8 (1829), 95–174.

[22] Cliffe, K. A., Garratt, T. J., and Spence, A. Eigenvalues of the discretized Navier-Stokes equation with application to the detection of Hopf bifurcations. *Adv. Comput. Math. 1*, 3 (1993), 337–356.

[23] Cullum, J. K., and Willoughby, R. A. Computing eigenvalues of very large symmetric matrices – An implementation of a Lanczos algorithm with no reorthogonalization. *Journal of Computational Physics 44*, 2 (1981), 329 – 358.

[24] Cullum, J. K., and Willoughby, R. A. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 1 Theory*. Birhäuser, Boston, 1985.

[25] Dackland, K., and Kågström, B. Blocked algorithms and software for reduction of a regular matrix pair to generalized Schur form. *ACM Trans. Math. Softw. 25*, 4 (1999), 425–454.

[26] Davies, P. I., Higham, N. J., and Tisseur, F. Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem. *SIAM J. Matrix Anal. Appl. 23*, 2 (2001), 472–493.

[27] De Samblanx, G., Meerbergen, K., and Bultheel, A. The implicit application of a rational filter in the RKS method. *BIT Numer. Math. 37*, 4 (1997), 925–947.

[28] Dongarra, J. J., Sorensen, D. C., and Hammarling, S. J. Block reduction of matrices to condensed forms for eigenvalue computations. In *Parallel Algorithms for Numerical Linear Algebra*, Henk A. van der Vorst and Paul van Dooren, Ed., vol. 1 of *Advances in Parallel Computing*. North-Holland, 1990, pp. 215 – 227.

[29] Druskin, V., and Knizhnerman, L. Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM J. Matrix Anal. Appl. 19*, 3 (1998), 755–771.

[30] Druskin, V., Knizhnerman, L., and Simoncini, V. Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation. *SIAM J. Numer. Anal. 49*, 5 (2011), 1875–1898.

[31] Druskin, V., Knizhnerman, L., and Zaslavsky, M. Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts. *SIAM J. Sci. Comput. 31*, 5 (2009), 3760–3780.

[32] Druskin, V., Simoncini, V., and Zaslavsky, V. Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems. *SIAM J. Matrix Anal. Appl. 35*, 2 (2014), 476–498.

[33] Elman, H., Meerbergen, K., Spence, A., and Wu, M. Lyapunov inverse iteration for identifying Hopf bifurcations in models of incompressible flow. *SIAM J. Sci. Comput. 34*, 3 (2012), 1584–1606.

[34] Elman, H., Ramage, A., and Silvester, D. Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow. *ACM Trans. Math. Softw. 33* (2007), 2–14.

[35] Elman, H., Ramage, A., and Silvester, D. IFISS: A computational laboratory for investigating incompressible flow problems. *SIAM Rev. 56* (2014), 261–273.

[36] Elman, H. C., and Wu, M. Lyapunov inverse iteration for computing a rightmost eigenvalues of large generalized eigenvalue problems. *SIAM J. Matrix Anal. Appl. 34*, 4 (2013), 1685–1707.

[37] Emami-Naeini, A., and Van Dooren, P. Computation of zeros of linear multivariable systems. *Automatica 18*, 4 (1982), 415–430.

[38] Fasino, D. Rational Krylov matrices and QR steps on Hermitian diagonal-plus-semiseparable matrices. *Numer. Linear Algebr. with Appl. 12*, 8 (2005), 743–754.

[39] Francis, J. G. F. The QR transformation, a unitary analogue to the LR transformation—Part 1. *Comput. J. 4*, 3 (1961), 265–271.

[40] Francis, J. G. F. The QR transformation—Part 2. *Comput. J. 4*, 4 (1962), 332–345.

[41] Gantmacher, F. R. *The Theory of Matrices, I, II.* Chelsea Publishing Company., 1959.

[42] Garvey, S. D., Tisseur, F., Friswell, M. I., Penny, J. E. T., and Prells, U. Simultaneous tridiagonalization of two symmetric matrices. *Int. J. Numer. Meth. Eng. 57*, 12 (2003), 1643–1660.

[43] Giraud, L., Langou, J., and Rozloznik, M. The loss of orthogonality in the Gram-Schmidt orthogonalization process. *Comput. Math. Appl. 50*, 7 (2005), 1069 – 1075.

[44] Gockler, T. *Rational Krylov subspace methods for ϕ-functions in exponential integrators.* PhD thesis, 2014.

[45] Golub, G., and Uhlig, F. The QR algorithm: 50 years later its genesis by John Francis and Vera Kublanovskaya and subsequent developments. *IMA J. Numer. Anal. 29*, 3 (2009), 467–485.

[46] Golub, G. H., and Van Loan, C. F. *Matrix Computations*, 4th ed. 2012.

[47] Grimme, E., Gallivan, K., and Van Dooren, P. A rational Lanczos algorithm for model reduction. *Numer. Algorithms 12* (1998), 33–63.

[48] Grimme, E. J., Sorensen, D. C., and Van Dooren, P. Model reduction of state space systems via an implicitly restarted Lanczos method. *Numer. Algorithms 12* (1996), 1–31.

[49] Gugercin, S., Antoulas, A. C., and Beattie, C. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Anal. Appl. 30*, 2 (2008), 609–638.

[50] Gutknecht, M. H., and Parlett, B. N. From qd to LR, or, how were the qd and LR algorithms discovered? *IMA J. Numer. Anal. 31*, 3 (2011), 741–754.

[51] Güttel, S. Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM-Mitteilungen 36*, 1 (2013), 8–31.

[52] Güttel, S., Van Beeumen, R., Meerbergen, K., and Michiels, W. NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems. *SIAM J. Sci. Comput. 36*, 6 (2014), A2842–A2864.

[53] Hawkins, T. Cauchy and the spectral theory of matrices. *Historia Mathematica 2*, 1 (1975), 1 – 29.

[54] He, C., Laub, A. J., and Mehrmann, V. Placing plenty of poles is pretty preposterous. In *Preprint SPC 95-17, Forschergruppe 'Scientific Parallel Computing', Fak. f. Mathematik, TU Chemnitz-Zwickau* (1995).

[55] Higham, N. J. *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.

[56] Hilbert, D. Grundzüge einer allgemeinen theorie der linearen integralgleichungen. In *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* (1904), Vandenhoeck & Ruprecht, pp. 49–91.

[57] Kågström, B. A direct method for reordering eigenvalues in the generalized real Schur form of a regular matrix pair (A, B). In *Linear Algebra for Large Scale and Real-Time Applications*, M. S. Moonen, G. H. Golub, and B. L. R. De Moor, Eds. Springer Netherlands, Dordrecht, 1993, pp. 195–218.

[58] Kågström, B., and Kressner, D. Multishift variants of the QZ algorithm with aggressive early deflation. *SIAM J. Matrix Anal. Appl. 29*, 1 (2007), 199–227.

[59] Kågström, B., Kressner, D., Quintana-Ortí, E., and Quintana-Ortí, G. Blocked algorithms for the reduction to Hessenberg-triangular form revisited. *BIT Numerical Mathematics 48*, 3 (2008), 563–584.

[60] Kågström, B., and Poromaa, P. Computing eigenspaces with specified eigenvalues of a regular matrix pair (A, B) and condition estimation: theory, algorithms and software. *Numer. Algorithms 12* (1996), 369–407.

[61] Karlsson, L. *Scheduling of parallel matrix computations and data layout conversion for HPC and Multi-Core Architectures.* PhD thesis, Umeå UniversityUmeå University, Department of Computing Science, High Performance Compting Center North (HPC2N), 2011.

[62] Kaufman, L. Some thoughts on the QZ algorithm for solving the generalized eigenvalue problem. *ACM Trans. Math. Softw. 3*, l (1977), 65–75.

[63] Kautsky, J., Nichols, N. K., and Van Dooren, P. Robust pole assignment in linear state feedback. *Int. J. Control 41*, 5 (1985), 1129–1155.

[64] Knizhnerman, L., and Simoncini, V. A new investigation of the extended Krylov subspace method for matrix function evaluations. *Numer. Linear Algebr. with Appl. 17*, June 2009 (2009), 615–638.

[65] Korvink, J. G., and Rudnyi, E. B. Oberwolfach benchmark collection. In *Dimension Reduction of Large-Scale Systems* (Berlin, Heidelberg, 2005), P. Benner, D. C. Sorensen, and V. Mehrmann, Eds., Springer Berlin Heidelberg, pp. 311–315.

[66] Kressner, D. *Numerical Methods for General and Structured Eigenvalue Problems*, vol. 46. Springer-Verlag Berlin Heidelberg, 2005.

[67] Kressner, D. On the use of larger bulges in the QR algorithm. *Electron. Trans. Numer. Anal. 20* (2005), 50–63.

[68] Krylov, A. N. On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined. *Izv. AN SSSR (News Acad. Sci. USSR) 7*, 4 (1931), 491–539. (In Russian.).

[69] Kublanovskaya, V. N. On some algorithms for the solution of the complete eigenvalue problem. *USSR Comp. Math. Phys. 3* (1961), 637–657. (In Russian.).

[70] Kuijlaars, A. Which eigenvalues are found by the Lanczos method? *SIAM J. Matrix Anal. Appl. 22*, 1 (2000), 306–321.

[71] Kuijlaars, A. Convergence analysis of Krylov subspace iterations with methods from potential theory. *SIAM Rev. 48*, 1 (2006), 3–40.

[72] Lancaster, P., and Rodman, L. *Algebraic Riccati Equations*. Oxford science publications. Clarendon Press, 1995.

[73] Lanczos, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand. (1934). 45*, 4 (1950), 255.

[74] Lehoucq, R. B., and Meerbergen, K. Using generalized Cayley transformations within an inexact rational Krylov sequence method. *SIAM J. Matrix Anal. Appl. 20*, 1 (1998), 131–148.

[75] Lehoucq, R. B., and Sorensen, D. C. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl. 17*, 4 (1996), 789–821.

[76] Lehoucq, R. B., Sorensen, D. C., and Yang, C. *ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. 1997.

[77] Lemonnier, D., and Van Dooren, P. Balancing Regular Matrix Pencils. *SIAM J. Matrix Anal. Appl. 28*, 1 (2006), 253–263.

[78] Liesen, J., and Strakos, Z. *Krylov Subspace Methods: Principles and Analysis*. Oxford University Press, 2013.

[79] Mach, T., Barel, M. V., and Vandebril, R. Inverse eigenvalue problems for extended Hessenberg and extended tridiagonal matrices. *Journal of Computational and Applied Mathematics 272* (2014), 377 – 398.

[80] Mach, T., and Vandebril, R. On deflations in extended QR algorithms. *SIAM J. Matrix Anal. Appl. 35*, 2 (2014), 559–579.

[81] Mach, T., and Vandebril, R. On Deflations in Extended QR Algorithms. *SIAM J. Matrix Anal. Appl. 35*, 2 (2014), 559–579.

[82] Mastronardi, N., and Van Dooren, P. The QR Steps with Perfect Shifts. *SIAM J. Matrix Anal. Appl. 39*, 4 (2018), 1591–1615.

[83] Moler, C. B., and Stewart, G. W. An algorithm for generalized matrix eigenvalue problems. *SIAM J. Numer. Anal. 10*, 2 (1973), 1–52.

[84] Morgan, R. B. On restarting the Arnoldi method for large nonsymmetric eigenvalue problems. *Math. Comput. 65*, 215 (1996), 1213–1231.

[85] Morgan, R. B., and Zeng, M. Harmonic projection methods for large non-symmetric eigenvalue problems. *Numer. Linear Algebr. with Appl. 5*, 1 (1998), 33–55.

[86] Paige, C. C. *The computation of eigenvalues and eigenvectors of very large sparse matrices.* PhD thesis, University of London, UK, 1971.

[87] Paige, C. C. Computational Variants of the Lanczos Method for the Eigenproblem. *IMA Journal of Applied Mathematics 10*, 3 (1972), 373–381.

[88] Parlett, B. The QR algorithm. *Comput. Sci. Eng. 2*, 1 (2000), 38–42.

[89] Parlett, B. N. Symmetric matrix pencils. *J. Comput. Appl. Mat. 38*, 1–3 (1991), 373–385.

[90] Polizzi, E. Density-matrix-based algorithm for solving eigenvalue problems. *Phys. Rev. B 79* (2009), 115112.

[91] Quintana-Ortí, G., and van de Geijn, R. Improving the performance of reduction to Hessenberg form. *ACM Trans. Math. Softw. 32*, 2 (2006), 180–194.

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

[93] Ruhe, A. Rational Krylov algorithms for nonsymmetric eigenvalue problems. In *Recent Advances in Iterative Methods* (New York, NY, 1994), G. Golub, M. Luskin, and A. Greenbaum, Eds., Springer New York, pp. 149–164.

[94] Ruhe, A. Rational Krylov algorithms for nonsymmetric eigenvalue problems. II. matrix pairs. *Linear Algebra Appl. 197–198* (1994), 283 – 295.

[95] Ruhe, A. The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices. *BIT Numer. Math. 34*, 1 (1994), 165–176.

[96] Ruhe, A. Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM J. Sci. Comput. 19*, 5 (1998), 1535–1551.

[97] Ruhe, A. The rational Krylov algorithm for nonlinear matrix eigenvalue problems. *J. Math. Sci. 114*, 6 (2003), 1854–1856.

[98] Rutishauser, H. Anwendungen des quotienten–differenzen–algorithmus. *Z. Angew. Math. Phys. 5* (1954), 496–508.

[99] Rutishauser, H. Der quotienten–differenzen–algorithmus. *Z. Angew. Math. Phys. 5* (1954), 233–251.

[100] Rutishauser, H. Ein infinitesimales analogon zum quotienten–differenzen–algorithmus. *Arch. Math. 5* (1954), 132–137.

[101] Rutishauser, H. Solution of eigenvalue problems with the LR-transformation. *Nat. Bur. Standards Appl. Math 49* (1958), 47–81.

[102] Saad, Y. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra Appl. 34* (1980), 269–295.

[103] Saad, Y. Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems. *Math. Comput. 42*, 166 (1984), 567–588.

[104] Saad, Y. Numerical solution of large nonsymmetric eigenvalue problems. *Comput. Phys. Commun. 53*, 1-3 (1989), 71–90.

[105] Saad, Y. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, 1992.

[106] Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[107] Sakurai, T., and Sugiura, H. A projection method for generalized eigenvalue problems using numerical integration. *J. Comput. Appl. Math. 159*, 1 (2003), 119 – 128.

[108] Sidje, R. B. On the simultaneous tridiagonalization of two symmetric matrices. *Numer. Math. 118*, 3 (2011), 549–566.

[109] Simoncini, V. Analysis of the rational Krylov subspace projection method for large-scale algebraic Riccati equations. *SIAM J. Matrix Anal. Appl. 37*, 4 (2016), 1655–1674.

[110] Skoogh, D. A parallel rational Krylov algorithm for eigenvalue computations. In *Applied Parallel Computing Large Scale Scientific and Industrial Problems* (Berlin, Heidelberg, 1998), B. Kågström, J. Dongarra, E. Elmroth, and J. Waśniewski, Eds., Springer Berlin Heidelberg, pp. 521–526.

[111] Sorensen, D. C. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl. 13*, 1 (1992), 357–385.

[112] Stewart, G. A Krylov-Schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl. 23*, 3 (2001), 601–614.

[113] Stewart, G. W. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review 15*, 4 (1973), 727–764.

[114] Turner, J. S. *Buoyancy Effects in Fluids*. Cambridge University Press, 1973.

[115] Van Barel, M., Fasino, D., Gemignani, L., and Mastronardi, N. Orthogonal Rational Functions and Structured Matrices. *SIAM J. Matrix Anal. Appl. 26*, 3 (2005), 810–829.

[116] Van Barel, M., and Kravanja, P. Nonlinear eigenvalue problems and contour integrals. *J. Comput. Appl. Math. 292* (2016), 526–540.

[117] Van Beeumen, R., Meerbergen, K., and Michiels, W. A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems. *SIAM J. Sci. Comput. 35*, 1 (2013), A327–A350.

[118] Van Beeumen, R., Meerbergen, K., and Michiels, W. Compact rational Krylov methods for nonlinear eigenvalue problems. *SIAM J. Matrix Anal. Appl. 36*, 2 (2015), 820–838.

[119] Van Beeumen, R., Meerbergen, K., and Michiels, W. Connections between contour integration and rational Krylov methods for eigenvalue problems, 2016. Department of Computer Science, KU Leuven, Technical report TW 673.

[120] van der Vorst, H. A. *Iterative Krylov Methods for Large Linear Systems*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003.

[121] Van Dooren, P. The computation of Kronecker's canonical form of a singular pencil. *Linear Algebra Appl. 27*, C (1979), 103–140.

[122] Van Dooren, P. A generalized eigenvalue approach for solving Riccati equations. *SIAM Journal on Scientific and Statistical Computing 2*, 2 (1981), 121–135.

[123] Vandebril, R. Chasing bulges or rotations? A metamorphosis of the QR-Algorithm. *SIAM J. Matrix Anal. Appl. 32*, 1 (2011), 217–247.

[124] Vandebril, R., Golub, G., and Van Barel, M. A quasi-separable approach to solve the symmetric definite tridiagonal generalized eigenvalue problem. *SIAM J. Matrix Anal. Appl. 31*, 1 (2009).

[125] Vandebril, R., Van Barel, M., and Mastronardi, N. A parallel QR-factorization/solver of quasiseparable matrices. *Electron. Trans. Numer. Anal. 30* (2008), 144–167.

[126] Vandebril, R., Van Barel, M., and Mastronardi, N. *Matrix Computations and Semiseparable Matrices, Volume II: Eigenvalue and Singular Value Methods.* Johns Hopkins University Press, Baltimore, Maryland, USA, 2008.

[127] Vandebril, R., Van Barel, M., and Mastronardi, N. Rational QR-iteration without inversion. *Numer. Math. 110*, 4 (2008), 561–575.

[128] Vandebril, R., Van Barel, M., and Mastronardi, N. A new iteration for computing the eigenvalues of semiseparable (plus diagonal) matrices. *Electronic Transactions on Numerical Analysis 33* (2009), 126–150.

[129] Vandebril, R., and Watkins, D. S. A generalization of the multishift QR algorithm. *SIAM J. Matrix Anal. Appl. 33*, 3 (2012), 759–779.

[130] Vandebril, R., and Watkins, D. S. An extension of the QZ algorithm beyond the Hessenberg-upper triangular pencil. *Electron. Trans. Numer. Anal. 40* (2013), 17–35.

[131] Walker, H. F. Implementation of the GMRES method using Householder transformations. *SIAM J. Sci. Statist. Comput 9*, 1 (1988), 152–163.

[132] Ward, R. C. The combination shift QZ algorithm. *SIAM J. Numer. Anal. 12*, 6 (1975), 835–853.

[133] Watkins, D. S. Understanding the QR algorithm. *SIAM Rev. 24*, 4 (1982), 427–440.

[134] Watkins, D. S. Some perspectives on the eigenvalue problem. *SIAM Rev. 35*, 3 (1993), 430–471.

[135] Watkins, D. S. The transmission of shifts and shift blurring in the QR algorithm. *Linear Algebra Appl. 241-243*, 1996 (1996), 877–896.

[136] Watkins, D. S. Bulge exchanges in algorithms of *QR*-type. *SIAM Journal on Matrix Analysis and Applications 19*, 4 (1998), 1074–1096.

[137] Watkins, D. S. Performance of the QZ Algorithm in the Presence of Infinite Eigenvalues. *SIAM J. Matrix Anal. Appl. 22*, 2 (2000), 364–375.

[138] Watkins, D. S. *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods.* Society for Industrial and Applied Mathematics, 2007.

[139] Watkins, D. S. Francis's algorithm. *Am. Math. Mon. 118*, 5 (2011), 387–403.

[140] Watkins, D. S., and Elsner, L. Theory of decomposition and bulge-chasing algorithms for the generalized eigenvalue problem. *SIAM J. Matrix Anal. Appl. 15*, 3 (1994), 943–967.

[141] Wilkinson, J. H. *The Algebraic Eigenvalue Problem.* Oxford University Press, 1965.

[142] Wilkinson, J. H. Global convergene of tridiagonal QR algorithm with origin shifts. *Linear Algebra Appl. 1*, 3 (1968), 409–420.

Curriculum vitae

Personalia

Name: Daan Camps.
Date of birth: July 6, 1988.
Place of birth: Neerpelt, Belgium.

Education

2015–2019

Ph.D. in Engineering Science: Computer Science.
KU Leuven, Belgium.

2011–2013

M.Sc. in Engineering Science: Mathematical Engineering.
KU Leuven, Belgium.

2009–2011

M.Sc. in Astronomy.
KU Leuven, Belgium.

2006–2010

B.Sc. in Physics.
UHasselt, Belgium.

Teaching

2015–2016

Teaching assistant for *Numerieke Wiskunde [H01D8B]*.

2015–2019

Teaching assistant for *Numerieke Modellering en Benadering [H01P3A]*.

2016–2017

Teaching assistant for *Numerieke Wiskunde [G0N90B]*.

List of publications

This is a list of publications and scientific achievements by the author during the period 2015-2019.

Articles in international reviewed journals

- **Camps D.**, Meerbergen K., and Vandebril R., A rational QZ method. (2019) SIAM Journal on Matrix Analysis and Applications. Volume 40, Number 3, Pages 943–972.
- **Camps D.**, Mastronardi N., Vandebril R., and Van Dooren P., Swapping 2×2 blocks in the Schur and generalized Schur form. (2019) Journal on Computational and Applied Mathematics. Available online.
- **Camps D.**, Meerbergen K., and Vandebril R., An implicit filter for rational Krylov using core transformations. (2019) Linear Algebra and its Applications. Volume 561, 15 January 2019, Pages 113-140.

Submitted articles

- **Camps D.**, Mach T., Vandebril R., and Watkins D. S., On pole-swapping methods for the eigenvalue problem. (2019) Submitted.
- **Camps D.**, Meerbergen K., and Vandebril R., A multishift, multipole rational QZ method with aggressive early deflation. (2019) Submitted.

Articles in preparation

- **Camps D.**, Mach T., Vandebril R., and Watkins D. S., Pole swapping methods for Hessenberg, unitary Hessenberg pencils: Rational QR algorithms. In preparation.
- **Camps D.**, Vandebril R., and Van Dooren P., Two-sided rational iterations for tridiagonal pencils. In preparation.

Presentations at international conferences

- **Camps D.** (2019). Pole swapping methods for the eigenvalue problem – Rational QR algorithms. Presented at ICIAM, Valencia, Spain, 15 Jul 2019-19 Jul 2019.
- **Camps D.**, Güttel S., Mach T., Vandebril R. (2019). Approximate inverse-free rational Krylov methods and the link with FOM and GMRES. Presented at ETNA25, Santa Margherita di Pula, Italy, 27 May 2019-29 May 2019.
- **Camps D.**, Vandebril R., Meerbergen K. (2018). A rational QZ method. Presented at the NASCA, Kalamata, Greece, 02 Jul 2018-06 Jul 2018.
- **Camps D.**, Vandebril R., Meerbergen K. (2018). RQZ: A rational QZ method for the generalized eigenvalue problem. Presented at the SIAM Conference on Applied Linear Algebra, Hong Kong, 04 May 2018-08 May 2018.
- **Camps D.**, Meerbergen K., Vandebril R. (2017). On the implicit restart of the rational Krylov method: chasing algorithms for polynomial, extended and rational Krylov. Presented at the ILAS, Iowa State University - Ames, IA, 24 Jul 2017-28 Jul 2017
- **Camps D.**, Meerbergen K., Vandebril R. (2016). Towards a computational efficient, implicitly restarted rational Krylov method. Presented at the ILAS, Leuven, 11 Jul 2016-15 Jul 2016.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
NUMA
Celestijnenlaan 200A box 2402
B-3001 Leuven
daan.camps@cs.kuleuven.be
https://campsd.github.io/