Robust numerical methods for nonlinear eigenvalue problems

Šain Glibić, Ivana

University of Zagreb

FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICS

Ivana Šain Glibić

Robust numerical methods for nonlinear eigenvalue problems

DOCTORAL THESIS

Zagreb, 2018

Sveučilište u Zagrebu

PRIRODOSLOVNO - MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Ivana Šain Glibić

Robusne numeričke metode za nelinearne probleme svojstvenih vrijednosti

DOKTORSKI RAD

Zagreb, 2018.

University of Zagreb

FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICS

Ivana Šain Glibić

Robust numerical methods for nonlinear eigenvalue problems

DOCTORAL THESIS

Supervisor:
prof.dr.sc. Zlatko Drmač

Zagreb, 2018

Sveučilište u Zagrebu

PRIRODOSLOVNO - MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Ivana Šain Glibić

Robusne numeričke metode za nelinearne probleme svojstvenih vrijednosti

DOKTORSKI RAD

Mentor:
prof.dr.sc. Zlatko Drmač

Zagreb, 2018.

To my husband Marin, my rock and life partner.

Acknowledgements

My greatest gratitude goes to my supervisor professor Zlatko Drmač, for giving me the opportunity to continue my work from the graduate studies and for the guidance trough these four years. His motivational discussions where always inspiring. He always pushed me to work harder and better, to be independent in my work, but at the same had great impact on the shaping of this thesis.

I would also like to thank professor Saša Singer for his support and comforting words in the hardest times, and to professor Luka Grubišić for always being positive and encouraging, and for giving me the assignment project which helped with my research.

Special thanks goes to my colleagues at the Department who spent a lot of time hanging out with me at the office 033. To my roommate Petra, for going trough the good and bad times with me. You made these years much easier.

Out of the Department, there are a lot of people who supported me and helped me, both trough my previous studies and doctoral studies: my parents, brother and sister. Thank you for all the patience and faith during these years. Especially my mother, for making me go to Zagreb and pursue Mathematics. Without your unlimited faith in my abilities, I would never be where I am today.

My friends Dajana, Jelena, Valentina thank you for the constant optimism and encouragement, and especially to Ivana who spent the most time listening to all my problems and concerns during these four years. To my cousin Tijana, for having the ears and words for everything else.

I gratefully acknowledge the financial support of Croatian Science Foundation under the project 9345 during my PhD research.

Summary

In this thesis we study numerical methods for solving nonlinear eigenvalue problems of polynomial type, i.e. $P(\lambda)x \equiv (\sum_{\ell=0}^{k} \lambda^{\ell} A_{\ell})x = \mathbf{0}$, where $A_{\ell} \in \mathbb{C}^{n\times n}$, $\lambda \in \mathbb{C}$, $\mathbf{0} \neq x \in \mathbb{C}^{n}$. In particular, we are interested in the quadratic ($k = 2$) and the quartic ($k = 4$) eigenvalue problems. The methods are based on the corresponding linearization – the nonlinear problem is replaced with an equivalent linear problem of the type $(A - \lambda B)y = \mathbf{0}$, of dimension kn.

We propose several modifications and improvements of the existing methods for both the complete and partial solution; this results in new numerical algorithms that are a substantial improvement over the existing ones. In particular, as an improvement of the state of the art `quadeig` method of Hammarling, Munro and Tisseur, we develop a scheme to deflate all zero and infinite eigenvalues before calling the QZ algorithm for the linear problem. This provides numerically more robust procedure, which we illustrate by numerical examples. Further, we supplement the parameter scaling (designed to equilibrate the norms of the coefficient matrices) with a two–sided diagonal scaling to nearly equilibrate (in modulus) the nonzero matrix entries. In addition, we analyze the fine details of the rank revealing factorization used in the deflation process. We advocate to use complete pivoting in the QR factorization, and we also propose a LU based approach, which is shown to be competitive, or even better than the one based on the QR factorization. The new method is extended to the quartic problem.

For the partial quadratic eigenvalue problem (computing only a part of the spectrum), the iterative Arnoldi–like methods are studied, especially the implicitly restarted two level orthogonal Arnoldi algorithm (TOAR). We propose several improvements of the method. In particular, new shift selection strategy is proposed for the implicit restart for the class of overdamped quadratic eigenvalue problems. Also, we show the benefit of choosing the starting vector for TOAR, based on spectral information of a nearby proportionally damped pencil. Finally, we provide some new ideas for the development of a Krylov–Schur like methods that is capable of using arbitrary polynomial filters in the implicit restarting.

Keywords: polynomial eigenvalue problem, quadratic eigenvalue problem, quartic eigenvalue problem, projection method, Arnoldi like method, linearization, QZ, quadeig, deflation, rank determination, normwise backward error, componentwise backward error, TOAR, SOAR

Prošireni sažetak

Nelinearni problemi svojstvenih vrijednosti se javljaju u mnogim primjenama kako u prirodnim znanostima, tako i u inženjerstvu. Jedna od najpoznatijih klasa nelinearnih svojstvenih problema su polinomni svojstveni problemi. Tako se, na primjer, kvadratični svojstveni problem $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$ pojavljuje u dinamičkoj analizi mehaničkih i električnih struktura, u vibro–akustici, mehanici fluida, obradi signala. S druge strane, polinomni se problem četvrtog reda $(\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + K)x = \mathbf{0}$ pojavljuje u analizi stabilnosti Poiseuilleovog toka u cijevi.

Za razliku od linearnih problema svojstvenih vrijednosti, numeričke metode za nelinearne probleme još uvijek nisu dovoljno razrađene, niti numerički pouzdane, iako je algebarska teorija za polinomne probleme svojstvenih vrijednosti dobro razvijena.

Naglasak ove disertacije je na numeričkom rješavanju kvadratičnog svojstvenog problema. Cilj je razviti nove, robusnije numeričke metode koje se mogu koristiti u praksi kao pouzdan numerički softver.

U disertaciji se pročavaju dvije vrste metoda: *direktne* i *iterativne*. Direktne metode se razvijaju za računanje svih svojstvenih vrijednosti i odgovarajućih svojstvenih vektora zadanog problema. Kada nas zanima samo dio spektra, recimo one svojstvene vrijednosti koje su najveće po modulu ili one koje se nalaze u lijevoj kompleksnoj poluravnini, tada korisitimo iterativne metode. Ovdje je najčešće slučaj da je dimenzija originalnog problema mnogo veća od broja svojstvenih vrijednosti koje želimo izračunati. Ideja iterativnih metoda je konstruirati potprostor mnogo manje dimenzije od originalnog problema koji sadrži informaciju o traženom dijelu spektra, a aproksimacija traženog dijela spektra se onda izračuna koristeći projekciju problema na nađeni potprostor.

Osnova većine metoda za rješavanje polinomnih svojstvenih problema je linearizacija, to jest polinomni problem se zamijeni ekvivalentnim linearnim problemom koji se onda rješava koristeći već razvijene metode za linearne probleme. Međutim, naivno direktno korištenje linearnih metoda ne garantira zadovoljavajuće rezultate za originalni problem. Čak i ako izračunati svojstveni par ima malu grešku unazad za odgovarajuću linearizaciju, greška unazad za rekonstruirani svojstveni par originalnog problema može biti puno veća.

Prije razvijanja metoda, u Poglavlju 2 je predstavljena analiza grešaka unazad za polinomni svojstveni problem, bazirana na radu F. Tisseur [66]. Ideja analize grešaka unazad je da se izračunate aproksimacije interpretiraju kao egzaktna rješenja problema koji je blizu originalnom

problemu, i čiji matrični koeficijenti su definirani kao $A_\ell + \Delta A_\ell$ pri čemu je ΔA_ℓ malo. Međutim, u mnogim primjenama matrice A_ℓ imaju određenu strukturu, npr. hermitske su, ili anti hermitske. Prema tome, bilo bi prirodno zahtjevati da greška unazad ΔA_ℓ čuva ovu strukturu. U slučaju kad je ta stuktura hermitska i anti hermitska, postojeći rezultati za realne svojstvene vrijednosti su prošireni na općenite svojstvene vrijednosti.

U poglavlju 3 se proučavaju direktne metode za rješavanje kvadratičnog svojstvenog problema. Standardni pristup je korištenje QZ algoritma na odgovarajućoj linearizaciji. Međutim, ako originalni problem ima svojstvene vrijednosti koje su nula ili beskonačno, ovakav pristup je sklon numeričkim poteškoćama. 2011. Hammarling, Munro i Tisseur [37] su razvili `quadeig` algoritam koji prije korištenja QZ metode za linearni problem skalira originalni problem kako bi norme matričnih koeficijenata bile ujednačene te pokuša detektirati postojanje svojstvenih vrijednosti nula i beskonačno koje ona procesom deflacije ukloni iz linearizacije.

Deflacija se temelji na određivanju ranga matrica M i K. Kod `quadeiga` se koristi QR faktorizacija pivotiranjem stupaca. Koristeći ortogonalne transformacije $n - \text{rank}(M)$ beskonačnih i $n - \text{rank}(K)$ svojstvenih vrijednosti nula je uklonjeno iz odgovarajuće linearizacije. Glavni doprinos ovog poglavlja je novi algoritam za nalaženje svih svojstvenih vrijednosti kvadratično problema kojeg zovemo `KVADeig`. Kao motivacija za potrebu poboljšanja `quadeiga` je predstavljen primjer kod kojeg `quadeig` nije uspio detektirati sve beskonačne svojstvene vrijednosti. Štoviše, nakon što je uklonjen određen broj ovih svojstvenih vrijednosti, preostale izračunate svojstvene vrijednosti koje su konačne čak nemaju ni veliku apsolutnu vrijednost koja bi nas možda mogla nagnati na zaključak da bi one trebale biti proglašene beskonačnim. Problem nastane kada postoji više od jednog Jordanovog bloka za svojstvene vrijednosti nula i bekonačno. Naime, deflacija u `quadeigu` ukloni samo jedan Jordanov blok.

Kako bismo riješili ovaj problem razvili smo test koji služi za provjeru postoji li više od jednog Jordanovog bloka za svojstvene vrijednosti nula i bekonačno. On je baziran na Van Doorenovom algoritmu za određivanje Kroneckerove strukture generaliziranog svojstvenog problema. Dodatno se analizira utjecaj metoda koje se koriste kao faktorizacije za određivanje ranga te utjecaj kriterija po kojem se rang određuje. Pored skaliranja koje je predloženo u `quadeigu` uvodimo i dvostrano dijagonalno balansiranje čiji je cilj ujednačavanje elemenata u matricama koje definiraju problem. Na kraju razvijamo metodu baziranu na LU faktorizaciji potpunim pivotiranjem za određivanje ranga. Numerički eksperimenti u Sekciji 3.7 ilustriraju prednosti predložene metode.

U poglavlju 4 je razvijen novi algoritam `KVARTeig` za rješavanje polinomnog svojstvenog problema stupnja četiri. Umjesto direktne linearizacije koristimo kvadratifikaciju koja je uvedena u [17], tj. definiramo ekvivalentan kvadratični problem. Novi algoritam je baziran na `KVADeigu`, s tim da je skaliranje definirano na matricama originalnog problema i proces deflacije je prilagođen tako da što više iskoristi strukturu originalnog problema. Kao i za kvadratični problem, i ovdje je razvijen test za provjeru postojanja više od jednog Jordanovog bloka za svojstvene vrijednosti nula i beskonačno. Numerički primjeri u Sekciji 4.5 prikazuju prednost

nove metode nad `quadeigom` i `polyeigom` koji je implementiran u MATLABu.

U Poglavlju 5 se proučavaju iterativne metode Arnoldijevog tipa za kvadratični svojstveni problem. Bai i Su [3] su prvi primijetili da je u slučaju iterativnih metoda Arnoldijevog tipa bolje primijeniti Rayleigh–Ritzovu projekciju direktno na originalni kvadratični problem. U tu svrhu su definirani Krilovljev potprostor drugog reda i odgovarajući algoritam SOAR (Second Order Arnoldi) za računanje odgovarajuće baze. Ovaj algoritam je dodatno modificiran te je razvijen takozvani TOAR (Two level orthogonal Arnoldi) algoritam [49].

U ovom poglavlju predlažemo nekoliko modifikacija implicitno restartanog TOAR algoritma koje su temeljene na činjenici da algoritam koristimo za rješavanje kvadratičnog problema svojstvenih vrijednosti. Pod implicitnim restartanjem se misli na korištenje polinomih filtera kako bi se definirao novi početni vektor koji uvelike utječe na konvergenciju metode. Za posebnu klasu pregušenih problema svojstvenih vrijednosti predlažemo novi način definiranja polinomih filtera. Također, za općenite probleme, predlažemo novi izbor početnog vektora koji se temelji na aproksimaciji kvadratičnog svojstvenog problema problemom čije je gušenje linearno. Numerički primjeri pokazuju da predložene modifikacije rezultiraju manjim brojem restartanja potrebnih za nalažanje svojstvenih parova sa zadovoljavajućom greškom unatrag.

U drugom dijelu Poglavlja 5 dajemo pregled implicitno restartanog Krylov–Schurovog algoritma kojeg je uveo Stewart [64]. Ideja ovog algoritma je da se definira faktorizacija koja ne zahtijeva posebnu strukturu kao Arnoldijeva, i na koju će se lakše primijeniti implicitno restartanje. Međutim, prilikom ovakvog restartanja moguće je koristiti samo egzaktne pomake za definiranje polinomnog filtera. Drmač i Bujanović su razvili metodu koja omogućava korištenje proizvoljnih pomaka kod implicitno restartanog Krylov–Schurovog algoritma. U ovom poglavlju generalizairamo predloženi proces u svrhu korištenja Krylov–Schurovog algoritma za rješavanje kvadratičnog svojstvenog problema.

Contents

Introduction **1**

1 The Polynomial Eigenvalue Problem **7**

1.1 Problem setting 7

1.2 Canonical forms of matrix polynomials 9

1.2.1 Jordan normal form of matrix 9

1.2.2 The Smith form 11

1.2.3 Jordan chains 13

1.2.4 Invariant Pairs 18

1.3 Applications of polynomial eigenvalue problem 20

1.3.1 Disk brake squeal 20

1.3.2 Regularized Total Least Squares 22

1.3.3 Orr-Sommerfeld equation 25

1.4 Linearizations of Matrix Polynomials 27

1.5 Localization of eigenvalues of nonlinear eigenvalue problem 31

1.5.1 Pseudospectrum 32

1.6 Diagonalizable quadratic matrix polynomials 33

1.7 Minimax theory 36

1.7.1 The primary functional 37

1.7.2 The secondary functional 38

2 Backward error **41**

2.1 Optimal backward error for a given eigenpair 41

2.2 On Hermitian and skew–Hermitian backward error 43

2.2.1 The left eigenpair 44

2.3 Backward error for a homogeneous form of $P(\lambda)$ 46

2.3.1 Backward error bounds for the homogeneous form 47

2.3.2 Parameter scaling 50

2.4 Componentwise backward error 53

3 Complete solution of the QEP **59**
3.1 Rank revealing decompositions . . . 61
3.1.1 Singular Value Decomposition (SVD) . . . 61
3.1.2 QR factorization with column and complete pivoting . . . 62
3.1.3 The complete orthogonal factorization (URV) . . . 66
3.1.4 Rank revealing LU and Cholesky factorizations . . . 68
3.2 Kronecker's canonical form for general pencils . . . 72
3.3 The algorithm `quadeig` . . . 76
3.3.1 Parameter scaling . . . 76
3.3.2 Deflation process in `quadeig` . . . 78
3.3.3 Eigenvectors in `quadeig` . . . 86
3.4 Balancing by two-sided diagonal scalings . . . 90
3.4.1 The algorithm . . . 90
3.5 Improved deflation process. New algorihm – `KVADeig` . . . 92
3.5.1 A case study example . . . 93
3.5.2 Deflation process revisited . . . 96
3.5.3 Computing the Kronecker's Canonical form using rank revealing QR factorization . . . 100
3.5.4 Putting it all together: Deflation process in `KVADeig` . . . 103
3.5.5 Numerical examples . . . 107
3.6 LU based deflation . . . 108
3.6.1 The case of nonsingular M . . . 109
3.6.2 Rank deficient cases . . . 110
3.6.3 Eigenvectors . . . 110
3.6.4 Computing the Kronecker's Canonical form using rank revealing LU factorization . . . 114
3.7 Numerical examples. Comparison of rank revealing decompositions . . . 117
3.7.1 Example 1. `cd_player`. . . . 117
3.7.2 Example 2. Scaled `dirac`. . . . 119
3.7.3 Constrained least squares problem . . . 122

4 Complete solution of the quartic eigenvalue problem **125**
4.1 Quadratification . . . 125
4.1.1 Companion form of grade 2 . . . 126
4.2 Scaling . . . 127
4.2.1 Tropical scaling. . . . 128
4.2.2 Fan, Lin, Van Dooren generalization scaling. . . . 128
4.3 Deflation process . . . 128
4.3.1 Backward error analysis for the deflation process . . . 134

4.3.2 Eigenvector recovery . . . 136
4.4 Deflation process in KVARTeig algorithm . . . 138
4.5 Numerical experiments . . . 140

5 Iterative methods **145**
5.1 Arnoldi algorithm . . . 146
5.1.1 Implicitly restarted Arnoldi (IRA) . . . 149
5.2 Second Order Arnoldi (SOAR) . . . 151
5.3 Two level orthogonal Arnoldi factorization . . . 155
5.3.1 Implicitly restarting the TOAR procedure . . . 157
5.4 TOAR revisited . . . 159
5.4.1 Deflation and breakdown . . . 159
5.4.2 TOAR as a linear eigenvalue problem solver . . . 160
5.4.3 TOAR as a quadratic solver . . . 160
5.4.4 Polynomial filter for overdamped problems . . . 163
5.4.5 Numerical examples for overdamped problems . . . 165
5.5 Locking in IRA . . . 167
5.5.1 Locking in TOAR . . . 169
5.6 Rayleigh damping . . . 170
5.6.1 Numerical examples . . . 171
5.7 Krylov–Schur algorithm for the linear eigenproblem . . . 174
5.7.1 Using the arbitrary shifts in Krylov–Schur algorithm . . . 176
5.8 Implicitly restarted Krylov–Schur algorithm for the QEP . . . 177
5.8.1 Using arbitrary shifts in the Krylov–Schur algorithm for the quadratic eigenvalue problem . . . 179

Conclusion **183**

Bibliography **185**

List of Figures **191**

List of Tables **193**

Curriculum Vitae **195**

Introduction

Nonlinear eigenvalue problems arise in wide spectrum of applications in natural sciences and engineering. In particular, the polynomial eigenvalue problem is to find all complex scalars λ and nontrivial vectors x such that

$$P(\lambda)x \equiv (\sum_{\ell=0}^{k} \lambda^{\ell} A_{\ell})x = \mathbf{0},$$

where $A_0, \ldots, A_k$ are real or complex $n \times n$ matrices. So, for instance, the quadratic eigenvalue problem $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$, which is in the focus of this thesis, is at the core of dynamic analysis of mechanical and electrical structures, vibro-acoustics, computational fluid mechanics, signal processing; just to name a few. For an excellent review, we refer [67]. Another important class of the polynomial eigenvalue problems that we consider is the quartic eigenvalue problem $(\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + K)x = \mathbf{0}$. It appears, for example, in the analysis of the stability of the plane Poiseuille flow in a channel.

Unlike the linear case (i.e. the linear eigenvalue problem $(A - \lambda B)x = \mathbf{0}$), numerical methods for the nonlinear problems are not at the satisfactory level with respect to numerical reliability and robustness. Interestingly, the algebraic theory of the general polynomial eigenvalue problem is well developed and the spectral canonical structure of $P(\lambda)$ is well understood; yet, the numerical methods, despite the importance of the problem in many engineering applications, are not satisfactory. One of the main reasons is that the nonlinearity brings in many analytical and numerical difficulties which in some situations can be classified as pathological. For instance, some eigenvalues can be infinite.

The main focus of the thesis is numerical solution of the quadratic eigenvalue problem; our goal is to contribute with development of new, better robust numerical methods that can be implemented as reliable mathematical/numerical software and used in applications.

We consider the two main classes of problems and the corresponding solution methods. The so called *direct methods* are designed to compute all eigenvalues and the corresponding eigenvectors, and are usually deployed for small to moderate dimensions n. On the other hand, in some applications, only certain eigenvalues of particular interest are needed e.g. in an engineering design. For instance, eigenvalues in the left half plane close to the imaginary axis are important for studying the stability of the underlying dynamical system; or, the eigenvalues in some given $\Omega \subset \mathbb{C}$ might be requested. In such applications, the coefficient matrices originate

from a discretization process (e.g. by finite elements) and are usually of large dimension (e.g. $n > 10^4, 10^5$ or higher) and sparse (only small number of entries are nonzero) and structured. The idea of the so called *iterative methods* is to, iteratively, construct a subspace (of dimension much smaller than the original dimension n) such that the requested spectral information can be extracted from the problem projected onto that subspace.

In the kernel of most of these methods is the linearization, i.e., the polynomial eigenvalue problem is replaced with an equivalent linear eigenvalue problem, which is then solved using the well developed techniques for linear problems. For example, one linearization for the quadratic eigenvalue problem $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$ is

$$Ay - \lambda By \equiv \begin{pmatrix} C & -\mathbb{I} \\ K & \mathbf{0} \end{pmatrix} y - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I} \end{pmatrix} y = \mathbf{0}, \text{ where } y = \begin{pmatrix} \lambda x \\ x \end{pmatrix}.$$

Unfortunately, this elegant algebraic manipulation cannot be so simply turned into a robust numerical method. The finite arithmetic rounding errors and the truncation of the necessarily infinite iteration process when solving the linear problem create the backward errors ΔA, ΔB such that $\|\Delta A\|/\|A\|$ and $\|\Delta B\|/\|B\|$ are small, but this backward stability does not extend to the original problem, i.e., we cannot in general claim that the approximate solution corresponds to slightly backward perturbed original matrices M, C and K. Hence, for both the direct and the iterative methods, careful modifications are necessary.

The thesis is structured as follows:

Chapter 1 contains preliminaries. It provides an algebraic setting of the polynomial eigenvalue problem, including the theory of canonical forms of matrix polynomials, which will be used in the developments of numerical methods. In addition, we provide brief illustrations of two selected applications of the quadratic eigenvalue problem, and one of the quartic eigenvalue problem. We also present the theory of the linearization of matrix polynomials, which is essential for the development of numerical methods.

In Chapter 2 we present elements of backward error analysis of the polynomial eigenvalue problem. It is based on the work of F. Tisseur [66]. Backward error analysis is fundamental in assessing the quality of the computed approximations and it provides means for a posteriori estimation of the accuracy of the computed eigenvalues and eigenvectors. It is the backward error analysis that guides in removing the discrepancy between the backward stability of the auxiliary linear and the original quadratic problem. In particular, it shows that the norms of the coefficient matrices A_ℓ should be balanced, which is then achieved by a parameter scaling. The idea of backward error analysis is to interpret the computed (approximate) result as the exact result of a nearby problem, defined with the coefficient matrices $A_\ell + \Delta A_\ell$, with small ΔA_ℓ. However, in many applications the matrices A_ℓ have an additional structure, e.g., they are Hermitian or skew–Hermitian. Hence, for proper use of backward error, it is desirable to establish the existence of the optimal (smallest is some well defined sense) backward errors ΔA_ℓ that preserve the structure. In Section 2.2 we extend the existing results for only real eigenvalues

to the general case of any finite eigenvalues, when the required structure is the hermiticity or the skew–hermiticity. In addition, we provide new insights in the component-wise measured backward error.

In Chapter 3 we study the complete solution of quadratic eigenvalue problem $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$ by direct methods. The standard approach is to linearize it and then use the QZ algorithm for the corresponding generalized (linear) eigenvalue problem. This procedure is known to be prone to numerical difficulties in presence of zero and infinite eigenvalues. In 2011., Hammarling, Munro and Tisseur [37] proposed the `quadeig` algorithm that substantially alleviated these problems by careful preprocessing. Before calling the QZ algorithm, `quadeig` deploys parameter scaling to equilibrate the norms of the coefficient matrices, and then attempts to detect and deflate the zero and the infinite eigenvalues from the linearized problem.

The deflation process relies on rank determination of the coefficient matrices M, and K, and `quadeig` uses the (rank revealing) QR factorization with column pivoting. Using the orthogonal equivalence transformation on the linearization, $n - \mathrm{rank}(M)$ infinite and $n - \mathrm{rank}(K)$ zero eigenvalues are removed from the linearized pencil. The remaining eigenvalues are computed using the QZ algorithm. In Chapter 3 we analyze the numerical properties of the `quadeig` in more details. We present the backward error analysis of the deflation process in the case of only one singular matrix, M or K. The main contribution of this Chapter is the new algorithm for the complete solution of the quadratic eigenvalue problem, which we designated as `KVADeig`. To illustrate the need for improvements, we use numerical case study examples where `quadeig` fails to find all infinite eigenvalues; moreover, the eigenvalues that are computed instead of infinities are finite and they may not be of large absolute values to even indicate that they may correspond to infinities. This often poses difficulties in applications, because those eigenvalues cannot be interpreted in a physically meaningful way. A closer analysis reveals that the problem is when the infinite eigenvalues are carried in several Jordan blocks (in the canonical structure), and `quadeig` is capable of deflating only one of them.

To solve this problem, we have developed a test for the existence of Jordan blocks for zero and infinite eigenvalues, and we have developed a new algorithm for the deflation of all zero and infinite eigenvalues. It is based on Van Dooren's algorithm for the Kronecker canonical form of the generalized eigenvalue problem. Further, we analyze the influence of the rank revealing factorization, and rank determination (truncation) criteria used to determine the numerical ranks of M and K. Here we show some weaknesses in the rank determination in the `quadeig` algorithm. Furthermore, we advocate to equip the column pivoted rank revealing QR factorizations with row sorting in the ℓ_∞ norm (the Powell–Reid and Björck pivoting). Also, in addition to parameter scaling as in `quadeig`, we introduce a two–sided diagonal scaling that (nearly) equilibrates the matrix entries; this proves to be a very powerful technique both for theoretical estimate and the practical computation. And finally, we develop a rank-revealing LU analogue of the QR approach. It may seem surprising at first, but the LU approach, when properly implemented, can outperform the QR based preprocessing and can even be recommended as a method of choice.

Numerical experiments in Section 3.7 demonstrate the power of the newly proposed method.

In Chapter 4, we develop a new algorithm, designated as `KVARTeig`, for the complete solution of the quartic eigenvalue problem $(\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + K)x = \mathbf{0}$. Instead of the direct linearization, we first use the so called quadratification introduced as an algebraic tool in [17], i.e., we define an equivalent quadratic eigenvalue problem. The new algorithm is based on `KVADeig`, wherein the scaling is done on the original matrices, and the deflation process is modified so that the structure of the original problem is exploited as long as possible in the process. As in Chapter 3, we provide a test for the existence of Jordan blocks for zero and infinite eigenvalues in terms of the original matrix coefficients. Our numerical examples in Section 4.5 show that the new algorithm outperforms `quadeig` and the `polyeig` function in MATLAB.

In Chapter 5, we investigate computation of only a selected part of the spectrum of the quadratic eigenvalue problem, using Arnoldi–like methods. Bai and Su [3] were the first who realized that in the case of iterative Arnoldi-type methods, it would be advantageous to apply the Rayleigh-Ritz projection directly to the initial quadratic problem, instead of to the linearization. To that end, they introduced second order Krylov subspaces, and the corresponding second order Arnoldi procedure for generating orthonormal bases. The resulting method, called Second Order Arnoldi (SOAR), is further modified yielding TOAR (Lu, Su and Bai [49]).

Here we propose several modifications of the Implicitly restarted TOAR algorithm [49], which uses the fact that the linear problem is a linearization of the quadratic eigenvalue problem. Implicit restarting refers to an application of a polynomial filter (implicitly through QR iterations), designed to purge the initial vector from the directions of the unwanted eigenvalues. This is a nontrivial issue as two eigenvalues (e.g., one wanted and one unwanted) may share the same eigenvector. Selecting good shifts to define a good filter is also more complex. We devise a new selecting strategy of shifts for one particular class – the overdamped quadratic eigenvalue problems. Here we deploy polynomials in tropical algebra.

It is known that the quality of the approximation for eigenpair produced by the Arnoldi algorithm depends on the starting vector. In this chapter we propose a new procedure for picking the starting vector based on the approximation of the original quadratic problem with the proportionally damped one, which can be reduced to the linear eigenvalue problem. Numerical examples in Subsection 5.6.1 illustrate that this new choice of the starting vector, together with other modifications of implicitly restarted TOAR, results with a smaller number of the restarts.

In the second part of Chapter 5 we introduce the Krylov–Schur algorithm developed by Stewart in [64]. Here, restrictions on the structure of the factorization from the Arnoldi decomposition are removed resulting in a more elegant restarting procedure. However, during the implicit restart only exact shifts can be used. This was improved by Bujanović and Drmač in [11]. They proposed the 4R procedure for applying arbitrary shifts in the implicit restart of the Krylov–Schur algorithm.

The standard Krylov–Schur algorithm can be used for the quadratic eigenvalue problem

so that the TOAR procedure is used to compute the starting decomposition. This method is implemented in [14]. Again, only exact shifts can be used in the implicit restart. In order to use the shifts proposed for the overdamped problems, and to use any other shifts in the restart we extend the 4R procedure for the Krylov–Schur algorithm used as a quadratic eigenvalue problem solver. The numerical example at the end of the Chapter demonstrates the importance of the possibility to choose the arbitrary shifts.

The parts of this thesis were presented at the following scientific meetings: at 6*th Croatian Mathematical Congress*, Zagreb, Croatia (the talk "Second Order Krylov Schur Algorithm with Arbitrary Filter"), at *European School on Mathematical Modelling, Numerical Analysis and Scientific Computing*, Kacov, Czech Republic (the talk "On Improved Implicit Restarting of Arnoldi Methods for Quadratic Eigenvalue Problem", results from Chapter 5), at *International Workshop on Optimal Control of Dynamical Systems and Applications*, Osijek, Croatia (the talk "On Implicit Restarting Of Second Order Arnoldi Procedure For Quadratic Eigenvalue Problem", results from Chapter 5), at *6th IMA Conference on Numerical Linear Algebra and Optimization*, Birmingham, United Kingdom (the talk "On Deflation Process and Solving the Quadratic Eigenvalue Problems", results form Chapter 3), and at *Ninth Conference on Applied Mathematics and Scientific Computing*, Šibenik, Croatia (the talk "An Algorithm for the Solution of Quartic Eigenvalue Problems", results from Chapter 4).

Chapter 1

The Polynomial Eigenvalue Problem

This chapter provides definitions and a selection of theory and results for polynomial eigenvalue problem needed in the development of the results in the remaining chapters.

1.1 Problem setting

In this section we define the polynomial eigenvalue problem and introduce the two canonical forms for matrix polynomials, namely the Smith form, and the Jordan form. These forms will be used for developing algorithms in Chapters 3 and 4. In addition, we present the notion of invariant pairs, which is an analogue of invariant subspaces in the linear case.

Polynomial eigenvalue problem. Let $P(\lambda)$ be a matrix polynomial of degree k

$$P(\lambda) = \sum_{\ell=0}^{k} A_\ell \lambda^\ell, \tag{1.1}$$

where $A_\ell \in \mathbb{C}^{n\times n}$, $\ell = 0,\ldots,k$, and $A_k \neq \mathbf{0}$. $P(\lambda)$ is often called λ-*matrix*. The matrix polynomial (1.1) is said to be *regular* if $\det P(\lambda)$ is not identically zero for all values of λ, and *nonregular* otherwise.

A scalar $\lambda \in \mathbb{C}$ is called an *eigenvalue* of the matrix polynomial if there exists a vector $x \in \mathbb{C}^n \setminus \{\mathbf{0}\}$ so that

$$P(\lambda)x = \mathbf{0}. \tag{1.2}$$

In this case, x is called a *right eigenvector* (or just an eigenvector). A vector $y \in \mathbb{C}^n \setminus \{\mathbf{0}\}$ is called a *left eigenvector* if

$$y^* P(\lambda) = \mathbf{0}. \tag{1.3}$$

We refer to (x,λ) as an *eigenpair*, and (x,y,λ) as an *eigentriple*.

Equivalently, λ is said to be an eigenvalue of the matrix polynomial P if it is a zero of $\det P(\lambda)$. Since $\det P(\lambda) = \det A_k \lambda^{kn} + \textit{lower order powers of } \lambda$, we conclude that, if the coef-

ficient matrix A_k is regular, the number of eigenvalues for matrix polynomial of order k is kn. Therefore, the set of eigenvectors cannot be linearly independent, and it is possible for different eigenvalues to share the same eigenvector.

Example 1.1. Consider the quadratic eigenvalue problem

$$Q(\lambda)x \equiv \left\{\lambda^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \lambda \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix} + \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}\right\} x = \mathbf{0}.$$

The eigenvalues are the zeros of the polynomial

$$\det Q(\lambda) = (\lambda^2 + 5\lambda + 3)^2 - 1 = 0,$$

that is $-1, -4, \frac{-5+\sqrt{17}}{2}, \frac{-5-\sqrt{17}}{2}$. Eigenvalues -1 and -4 share the eigenvector $\begin{pmatrix} 1 & -1 \end{pmatrix}^T$, and $\frac{-5+\sqrt{17}}{2}$ and $\frac{-5-\sqrt{17}}{2}$ have the same eigenvector $\begin{pmatrix} 1 & 1 \end{pmatrix}^T$.

In addition, if the leading coefficient matrix A_k is singular, the degree r of the polynomial $\det P(\lambda)$ is smaller than kn and there are r finite and $kn - r$ infinite eigenvalues. Infinite eigenvalues are defined as the zero eigenvalues of the so called *reversal* problem

$$\operatorname{rev} P(\lambda) = \lambda^k P(1/\lambda) = \sum_{\ell=0}^{k} \lambda^\ell A_{k-\ell}. \tag{1.4}$$

Example 1.2. Consider the quadratic eigenvalue problem

$$Q(\lambda)x = \left\{\lambda^2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \lambda \begin{pmatrix} -3 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 2 & 0 \\ 0 & -3 \end{pmatrix}\right\}.$$

The degree of the polynomial $\det Q(\lambda)$ is 3

$$\det Q(\lambda) = \lambda^3 - 6\lambda^2 + 11\lambda - 6,$$

meaning that there is one infinite eigenvalue, and the remaining finite eigenvalues are $1, 2$ and 3. The reversed problem is

$$\operatorname{rev} Q(\lambda)x \equiv \left\{\mu^2 \begin{pmatrix} 2 & 0 \\ 0 & -3 \end{pmatrix} + \mu \begin{pmatrix} -3 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right\} x = \mathbf{0},$$

where $\mu = 1/\lambda$. The eigenvalues are the zeros of the polynomial

$$\det(\operatorname{rev} Q(\lambda)) = -6\lambda^4 + 11\lambda^3 - 6\lambda^2 + \lambda,$$

that is $0, 1, 1/2, 1/3$.

The *algebraic multiplicity* of an eigenvalue λ is the order of the corresponding zero in $\det P(\lambda)$. The *geometric multiplicity* of λ is the dimension of the nullspace $\ker P(\lambda)$. An eigenvalue λ is *simple* if its algebraic and geometric multiplicity are equal to 1. An eigenvalue λ is *semisimple* if its algebraic and geometric multiplicities coincide.

We will sometimes use the so called *homogeneous form* of the polynomial eigenvalue problem

$$P(\alpha,\beta) = \sum_{\ell=0}^{k} \alpha^{\ell}\beta^{k-\ell}A_{\ell}. \tag{1.5}$$

Here, λ is identified with any pair $(\alpha,\beta) \neq (0,0)$ for which $\lambda = \alpha/\beta$. The homogeneous form is useful because all eigenvalues, including infinity, are treated the same way. It is used in papers [42], [43] which consider backward errors and conditioning of linearizations of matrix polynomials. Analogously, we define homogeneous generalized (linear) eigenvalue problem

$$L(\alpha,\beta) = \beta A - \alpha B. \tag{1.6}$$

1.2 Canonical forms of matrix polynomials

The goal of this section is to describe Jordan structure of matrix polynomials. This is a generalization of the Jordan normal form for single matrix, and it is more complicated.

1.2.1 Jordan normal form of matrix

The Jordan normal form of a single matrix provides canonical structure that reveals complete spectral information; in the simplest case of diagonalizable matrix, the Jordan form is simply a diagonal matrix with the eigenvalues along the diagonal. If the matrix is not diagonalizable, the structure is more complex. We briefly review the key details.

For every integer ℓ and each eigenvalue λ_i of a matrix $A \in \mathbb{C}^{n\times n}$, it holds that $\mathrm{Ker}(A-\lambda_i\mathbb{I})^{\ell+1} \supset \mathrm{Ker}(A-\lambda_i\mathbb{I})^{\ell}$, and since we are dealing with finite dimensional space, there exists the smallest ℓ_i such that

$$\mathrm{Ker}(A-\lambda_i\mathbb{I})^{\ell_i+1} = \mathrm{Ker}(A-\lambda_i\mathbb{I})^{\ell_i},$$

and $\mathrm{Ker}(A-\lambda_i\mathbb{I})^{\ell} = \mathrm{Ker}(A-\lambda_i\mathbb{I})^{\ell_i}$ for all $\ell \geq \ell_i$. The integer ℓ_i is called the *index* of λ_i.
Denote with $M_i = \mathrm{Ker}(A-\lambda_i\mathbb{I})^{\ell_i}$ which is invariant subspace for A, and let $m_i = \dim(M_i)$.
In each invariant subspace M_i there are $\gamma_i \leq m_i$ independent eigenvectors which can be completed to form a basis by adding the elements of $\mathrm{Ker}(A-\lambda_i\mathbb{I})^2$, $\mathrm{Ker}(A-\lambda_i\mathbb{I})^3$, and so on. The process goes as follows:

- for each eigenvector $u \in \mathrm{Ker}(A-\lambda_i\mathbb{I})$, define z_1 so that $(A-\lambda_i\mathbb{I})z_1 = u$
- until it is possible, compute z_{i+1} as $(A-\lambda_i\mathbb{I})z_{i+1} = z_i$.

The vectors $z_i \in \mathrm{Ker}(A-\lambda_i\mathbb{I})^{i+1}$ are called *principal vectors*. There are at most ℓ_i principal vectors for each of the γ_i eigenvectors associated with the eigenvalue λ_i.

Finally, we can represent the matrix A with the respect to the basis made up of the p bases of invariant subspaces M_i

$$X^{-1}AX = J = \mathrm{diag}(J_1, J_2, \ldots, J_p), \tag{1.7}$$

where each J_i corresponds to the subspace M_i associated with the eigenvalue λ_i. J_i is of order m_i with following structure

$$J_i = \mathrm{diag}(J_{i1}, J_{i2}, \ldots, J_{i\gamma_i}),\ J_{ik} = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{pmatrix}. \tag{1.8}$$

Each J_{ik} corresponds to a different eigenvector of the eigenvalue λ_i, and its size is equal to the number of the principal vectors for the corresponding eigenvector. Previous reasoning is summed up in the following theorem.

Theorem 1.1 ([60])**.** *Any matrix A can be reduced to a block diagonal matrix consisting of p diagonal blocks, each associated with a distinct eigenvalue. Each diagonal block J_i has itself a block diagonal structure consisting of γ_i subblocks, where γ_i is the geometric multiplicity of the eigenvalue λ_i. Each of the subblocks, referred to as a Jordan block, is an upper bidiagonal matrix of size not exceeding ℓ_i, with the constant λ_i on the diagonal and the constant one on the super diagonal.*

Notice that, since A and J are similar, their characteristic polynomials are the same, and thus the algebraic multiplicity of the eigenvalue λ_i is the same, i.e., the algebraic multiplicity of λ_i is equal to m_i.

From all this we see that the Jordan form is very useful because it completely determines the structure of the eigenvalues of matrix A. However, the computation of it is numerically unstable. This is why the Schur form is used in numerical computation, because unitary matrix Q is used instead of regular X which can be ill conditioned. However, the form is no longer compact. The following theorem gives existence of the Schur form.

Theorem 1.2. *For any given matrix $A \in \mathbb{C}^{n\times n}$ there exists a unitary matrix Q such that $Q^*AQ = R$ is upper triangular.*

The Jordan structure for matrix polynomials provides the complete information about the structure of the eigenvalues. The main term we will define is Jordan pair. The first step is the definition of canonical Jordan chains, which are something like a basis in finite dimensional linear space [32]. The path of defining the Jordan pair is presented in Figure 1.1, therefore we start by defining the Smith form of P.

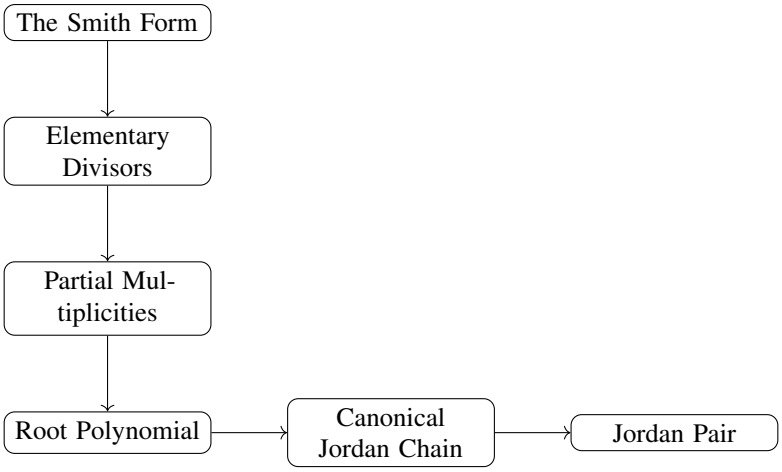


Figure 1.1: Diagram for defining the Jordan pair

1.2.2 The Smith form

The main result describing the Smith form is given in a more general form, meaning that it holds for matrix polynomials $\mathscr{P}(\lambda) = \sum_{\ell=0}^{k} A_\ell \lambda^\ell$, where $A_\ell \in \mathbb{C}^{m\times n}$ are rectangular matrices:

Theorem 1.3 ([32])**.** *Every $m \times n$ matrix polynomial $\mathscr{P}(\lambda)$ admits the representation*

$$\mathscr{P}(\lambda) = E(\lambda)D(\lambda)F(\lambda), \tag{1.9}$$

where

$$D(\lambda) = \begin{pmatrix} d_1(\lambda) & & & & 0 \\ & \ddots & & & \\ & & d_r(\lambda) & & \vdots \\ & & & 0 & \\ & & & & \ddots \\ 0 & & \cdots & & 0 \end{pmatrix} \tag{1.10}$$

is a diagonal polynomial matrix with monic scalar polynomials $d_i(\lambda)$ such that $d_i(\lambda)$ is divisible by $d_{i-1}(\lambda)$; $E(\lambda)$ and $F(\lambda)$ are matrix polynomials of sizes $m \times m$ and $n \times n$ respectively, with constant nonzero determinants.

Representation (1.9) is called *the Smith form of the matrix polynomial $\mathscr{P}(\lambda)$*. Sometimes, the matrix $D(\lambda)$ itself, given by (1.10), is also called the Smith form. The matrix polynomials $E(\lambda)$ and $F(\lambda)$ are not unique. However, $D(\lambda)$ is unique, and its diagonal polynomials can be expressed in terms of $\mathscr{P}(\lambda)$ as stated in the following theorem:

Theorem 1.4 ([32])**.** *Let $\mathscr{P}(\lambda)$ be an $m \times n$ matrix polynomial. Let $p_k(\lambda)$ be the greatest common divisor (with leading coefficient 1) of the minors of $\mathscr{P}(\lambda)$ of order k, if not all of them are zeros, and let $p_k(\lambda) \equiv 0$ if all minors of order k of $\mathscr{P}(\lambda)$ are zeros. Let $p_0(\lambda) = 1$ and*

$D(\lambda) = \operatorname{diag}(d_1(\lambda), \ldots, d_r(\lambda), 0, \ldots, 0)$ *be the Smith form of* $\mathscr{P}(\lambda)$. *Then* r *is the maximal integer such that* $p_r(\lambda) \not\equiv 0$, *and*

$$d_i(\lambda) = \frac{p_i(\lambda)}{p_{i-1}(\lambda)}, \;\; i = 1, 2, \ldots, r. \tag{1.11}$$

Invariant polynomials and Elementary Divisors. The diagonal elements $d_1(\lambda)$, $\ldots, d_r(\lambda)$ in (1.10) are called *invariant polynomials* of $\mathscr{P}(\lambda)$. Since $\operatorname{rank}\mathscr{P}(\lambda) = \operatorname{rank} D(\lambda)$ for every $\lambda \in \mathbb{C}$, and $\operatorname{rank} D(\lambda) = r$ if λ is not a zero of one of the invariant polynomials, and $\operatorname{rank} D(\lambda) < r$ otherwise, we conclude that

$$r = \max_{\lambda \in \mathbb{C}} \operatorname{rank} \mathscr{P}(\lambda).$$

If we represent each invariant polynomial as the product of factors

$$d_i(\lambda) = (\lambda - \lambda_{i1})^{\alpha_{i1}} \ldots (\lambda - \lambda_{i,k_i})^{\alpha_{i,k_i}}, \;\; i = 1, 2, \ldots, r,$$

where $\lambda_{i1}, \ldots, \lambda_{i,k_i}$ are different complex numbers and $\alpha_{i1}, \ldots, \alpha_{i,k_i}$ are positive integers, then the factors $(\lambda - \lambda_{ij})^{\alpha_{ij}}$, $j = 1, \ldots, k_i$, $i = 1, \ldots, r$ are called the *elementary divisors* of $\mathscr{P}(\lambda)$. An elementary divisor is said to be *linear* if $\alpha_{ij} = 1$, and *nonlinear* otherwise.
These characteristics will be important for developing the theory of Jordan structure. For better understanding of these concepts, let us present a simple example:

Example 1.3 ([32])**.** *Let*

$$P(\lambda) = \begin{pmatrix} \lambda(\lambda - 1) & 1 \\ 0 & \lambda(\lambda - 1) \end{pmatrix}.$$

The proof of theorem 1.3 describes the computation of the Smith form. However, we will not discuss the process here, but only state the final solution

$$D(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & \lambda^2(\lambda - 1)^2 \end{pmatrix}. \tag{1.12}$$

From (1.12) we read the elementary divisors: λ^2 *and* $(\lambda - 1)^2$.

Local Smith Form and Partial Multiplicities. We now return to consideration of matrix polynomial with square matrix coefficients (1.1). If $\det P(\lambda) \not\equiv 0$, that is, if P is regular, the next theorem describes the local Smith form:

Theorem 1.5 ([32])**.** *Let* $P(\lambda)$ *be and* $n \times n$ *matrix polynomial with* $\det P(\lambda) \not\equiv 0$. *Then for every* $\lambda_0 \in \mathbb{C}$, $P(\lambda)$ *admits the representation*

$$P(\lambda) = E_{\lambda_0}(\lambda) \begin{pmatrix} (\lambda - \lambda_0)^{\kappa_1} & & 0 \\ & \ddots & \\ 0 & & (\lambda - \lambda_0)^{\kappa_n} \end{pmatrix} F_{\lambda_0}(\lambda), \tag{1.13}$$

where $E_{\lambda_0}(\lambda)$ and $F_{\lambda_0}(\lambda)$ are matrix polynomials invertible at λ_0, and $\kappa_1 \leq \ldots \leq \kappa_n$ are nonnegative integers, which coincide (after removing zeros) with degrees of the elementary divisors of $P(\lambda)$ corresponding to λ_0 (i.e., of the form $(\lambda-\lambda_0)^n$). $\kappa_i = 0$, $i = 1,\ldots,n$ if λ_0 is not a root of an invariant polynomial of $P(\lambda)$.

The integers $\kappa_1 \leq \ldots \leq \kappa_n$ are called *partial multiplicities* of $P(\lambda)$, and they are uniquely determined by $P(\lambda)$ and λ_0. The representation (1.13) is called the *local Smith Form* of $P(\lambda)$ at λ_0.

Consider $P(\lambda)$ from Example 1.3. The partial multiplicities of eigenvalues 0 and 1 are $\kappa_1 = 0, \kappa_2 = 2$, and the partial multiplicities of $\lambda_0 \notin \{0,1\}$ are zeros.

Equivalence of Matrix Polynomials. Two matrix polynomials $P(\lambda)$ and $R(\lambda)$ of the same size are called *equivalent* (we write $P(\lambda) \sim R(\lambda)$) if

$$P(\lambda) = E(\lambda)R(\lambda)F(\lambda), \tag{1.14}$$

for some matrix polynomials $E(\lambda)$ and $F(\lambda)$ with constant nonzero determinants. This relation is indeed an equivalence relation. The important property of equivalent matrix polynomials is given in the following theorem

Theorem 1.6 ([32]). *$P(\lambda) \sim R(\lambda)$ if and only if the invariant polynomials of $P(\lambda)$ and $R(\lambda)$ are the same.*

1.2.3 Jordan chains

We will define a Jordan chain for matrix polynomial which is a generalization of a Jordan chain for a square matrix A.

As a motivation for the definition, consider the matrix polynomial $P(\lambda) = \sum_{\ell=0}^{k} A_\ell \lambda^\ell$, and the associated homogeneous differential equation

$$\sum_{\ell=0}^{k} A_\ell \frac{d^\ell}{dt^\ell} u(t) = \mathbf{0}, \tag{1.15}$$

where $u(t)$ is an n-dimensional vector valued function. Suppose that we seek the solution of (1.15) in the form

$$u(t) = p(t)e^{\lambda_0 t} = \left(\frac{t^m}{m!}x_0 + \frac{t^{m-1}}{(m-1)!}x_1 + \ldots + x_m\right) e^{\lambda_0 t}, \tag{1.16}$$

where $p(t)$ is an n-dimensional vector valued polynomial in t, λ_0 is a complex number, and $x_j \in \mathbb{C}^n$, $x_0 \neq 0$. Now, the following proposition holds

Proposition 1.1 ([32]). *The vector function $u(t)$ given by (1.16) is a solution of equation (1.15) if and only if the following equation holds:*

$$\sum_{p=0}^{i} \frac{1}{p!} P^{(p)}(\lambda_0) x_{i-p} = 0, \quad i = 0, 1, \ldots, m. \tag{1.17}$$

$P^{(p)}(\lambda)$ in (1.17) denotes the pth derivative of P with respect to λ.

The sequence of n-dimensional vectors $x_0, x_1, \ldots, x_m$ ($x_m \neq 0$) such that (1.17) holds is called a *Jordan chain of length $m+1$* for $P(\lambda)$, corresponding to the complex number λ_0. $P^{(p)}(\lambda)$ in (1.17) denotes the pth derivative of P with respect to λ. Its leading vector $x_0 \neq 0$ is an *eigenvector*, and the subsequent vectors $x_1, \ldots, x_m$ are called *generalized eigenvectors*.

It is important to notice that the vectors in a Jordan chain for the polynomial P, of order higher than one, need not be linearly independent. Indeed, the zero vector can be a generalized eigenvector as well. Example 1.4 illustrates this phenomena.

It is useful to note that the solutions of the linear system

$$\begin{pmatrix} P(\lambda_0) & 0 & \cdots & 0 \\ P'(\lambda_0) & P(\lambda_0) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{\ell!} P^{\ell}(\lambda_0) & \frac{1}{(\ell-1)!} P^{(\ell-1)}(\lambda_0) & \cdots & P(\lambda_0) \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_\ell \end{pmatrix} = \mathbf{0},$$

form the set of all Jordan chains $x_0, x_1, \ldots, x_\ell$ of $P(\lambda)$ with length not exceeding $\ell+1$ corresponding to λ_0.

The next proposition gives another way of writing a Jordan chain.

Proposition 1.2 ([32]). *The vectors $x_0, \ldots, x_{\ell-1}$ form a Jordan chain of the matrix polynomial $P(\lambda)$ of order k corresponding to λ_0 if and only if $x_0 \neq \mathbf{0}$ and*

$$A_0 X_0 + A_1 X_0 J_0 + \ldots + A_k X_0 J_0^k = \mathbf{0}, \tag{1.18}$$

where $X_0 = \begin{pmatrix} x_0 & \ldots & x_{\ell-1} \end{pmatrix}$ is an $n \times \ell$ matrix, and J_0 is the Jordan block of size $k \times k$ with λ_0 on the main diagonal.

Root Polynomials and Canonical set of Jordan Chains. An n-dimensional vector polynomial $\varphi(\lambda)$, such that $\varphi(\lambda_0) \neq 0$ and $P(\lambda_0)\varphi(\lambda_0) = 0$, is called a *root polynomial* of $P(\lambda)$ corresponding to λ_0. The multiplicity of the zero λ_0 of $P(\lambda)\varphi(\lambda)$ is called the *order* of the root polynomial $\varphi(\lambda)$.

Root polynomials are a tool for constructing the canonical set of Jordan chains:

1. Let $\varphi_1(\lambda) = \sum_{j=0}^{\kappa_1 - 1} (\lambda - \lambda_0)^j \varphi_{1j}$ be a root polynomial with the largest order κ_1.

2. Let $\varphi_2(\lambda) = \sum_{j=0}^{\kappa_2-1}(\lambda-\lambda_0)^j\varphi_{2j}$ be a root polynomial with the largest order among all the root polynomials whose eigenvector is not a scalar multiple of φ_{10}.

3. If $\varphi_1(\lambda),\dots,\varphi_{s-1}(\lambda)$ are already chosen, $\varphi_i = \sum_{j=0}^{\kappa_i-1}(\lambda-\lambda_0)^j\varphi_{ij}$, $i=1,\dots,s-1$, let $\varphi_s(\lambda) = \sum_{j=0}^{\kappa_s-1}(\lambda-\lambda_0)^j\varphi_{sj}$ be a root polynomial with the largest order κ_s among all the root polynomials whose eigenvectors are not in the span of the eigenvectors $\varphi_{10},\dots,\varphi_{s-1,0}$.

4. We continue this process until the set $\ker P(\lambda_0)$ of all eigenvectors of $P(\lambda)$ corresponding to λ_0 is exhausted. This means that we will construct $r = \dim\ker P(\lambda_0)$ root polynomials by this procedure.

Now, the Jordan chains

$$\varphi_{10},\dots,\varphi_{1,\kappa_1-1},\ \ \varphi_{20},\dots,\varphi_{1,\kappa_2-1},\ \ \dots\ \ \varphi_{r0},\dots,\varphi_{1,\kappa_r-1} \tag{1.19}$$

are called the *canonical set* of Jordan chains for $P(\lambda)$ corresponding to λ_0.

Example 1.4 ([32])**.** Let

$$P(\lambda) = \begin{pmatrix} \lambda^2(\lambda-1)(\lambda^2+1) & \lambda^3(\lambda-1) \\ \lambda^2(\lambda-1)^2 & \lambda^3(\lambda-1)^2 \end{pmatrix}.$$

The determinant is $\det P(\lambda) = \lambda^7(\lambda-1)$, meaning that the eigenvalues are 0 and 1. We will compute the Jordan chain for the eigenvalue 0. Let us write the derivatives

$$P'(\lambda) = \begin{pmatrix} 5x^4-4x^3+3x^2-2x & x^2(4x-3) \\ 2x(2x^2-3x+1) & x^2(5x^2-8x+3) \end{pmatrix}, \qquad P^{(IV)}(\lambda) = \begin{pmatrix} 24(5x-1) & 24 \\ 24 & 24(5x-2) \end{pmatrix},$$
$$P''(\lambda) = \begin{pmatrix} 20x^3-12x^2+6x-2 & 6x(2x-1) \\ 2(6x^2-6x+1) & 20x(10x^2-12x+3) \end{pmatrix}, \ P^{(V)}(\lambda) = \begin{pmatrix} 120 & 0 \\ 0 & 120 \end{pmatrix},$$
$$P'''(\lambda) = \begin{pmatrix} 60x^2-24x+6 & 24x-6 \\ 24x-12 & 60x^2-48x+6 \end{pmatrix}.$$

Since $P'(0) = 0$, we have that $P'(0)x_0 + P(0)x_1 = 0$ for all $x_0, x_1 \in \mathbb{C}^2$ with $x_0 \neq 0$, thus any combination x_0, x_1 forms a Jordan chain. Denote the elements of the vector x_i as x_{i1}, x_{i2}. Now,

$$\frac{1}{2!}P''(0)x_0 + P'(0)x_1 + P(0)x_2 = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x_{01} \\ x_{02} \end{pmatrix} = \mathbf{0},$$

implies that $x_{01} = 0$. The next equation

$$\frac{1}{3!}P'''(0)x_0 + \frac{1}{2!}P''(0)x_1 + P'(0)x_2 + P(0)x_3 = \begin{pmatrix} -x_{02}-x_{11} \\ x_{02}+x_{11} \end{pmatrix} = \mathbf{0}$$

implies that $x_{11} = -x_{02}$. Similarly,

$$\frac{1}{4!}P^{(IV)}(0)x_0 + \frac{1}{3!}P'''(0)x_1 + \frac{1}{2!}P''(0)x_2 + P'(0)x_3 + P(0)x_4 = \begin{pmatrix} -x_{12} - x_{21} \\ x_{12} + x_{21} \end{pmatrix} = \mathbf{0}$$

implies that $x_{21} = -x_{12}$. From the last equation

$$\frac{1}{5!}P^{(V)}(0)x_0 + \frac{1}{4!}P^{(IV)}(0)x_1 + \frac{1}{3!}P'''(0)x_2 + \frac{1}{2!}P''(0)x_3 + P'(0)x_4 + P(0)x_5 = \mathbf{0},$$

it is obvious that x_4, x_5 can be any two vectors. To conclude, our Jordan chain is of the form

$$\begin{pmatrix} 0 \\ x_{02} \end{pmatrix}, \begin{pmatrix} -x_{02} \\ x_{12} \end{pmatrix}, \begin{pmatrix} -x_{12} \\ x_{22} \end{pmatrix}, \begin{pmatrix} x_{31} \\ x_{32} \end{pmatrix}, \begin{pmatrix} x_{41} \\ x_{42} \end{pmatrix},$$

where $x_{02}, x_{12}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}$ are arbitrary complex numbers.
Now, to determine the canonical set of Jordan chains, we recall that if $x_{01} = 0$ the order of the root polynomial is 5, and if $x_{01} \neq 0$ the order is 2. This means that we can choose φ_{1j}, $j = 0, \ldots, 4$ to be

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

For φ_{2j}, $j = 0, 1$ we can choose

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Some of the useful properties of the canonical set of Jordan chains (proved in [32]) are:

- not unique,
- the numbers $\kappa_1, \ldots, \kappa_r$ are uniquely defined
- $\kappa_1, \ldots, \kappa_r$ are the nonzero partial multiplicities of $P(\lambda)$ at λ_0.

Jordan pair. Let (1.19) be the canonical Jordan chain of $P(\lambda)$ corresponding to the eigenvalue λ_0, and write it in the matrix form

$$X(\lambda_0) = \begin{pmatrix} \varphi_{10} & \ldots & \varphi_{1,\kappa_1-1} & \varphi_{20} & \ldots & \varphi_{2,\kappa_2-1} & \ldots & \varphi_{r0} & \ldots & \varphi_{r,\kappa_r-1} \end{pmatrix} \in \mathbb{R}^{n \times \kappa},$$
$$J(\lambda_0) = \operatorname{diag}(J_1, J_2, \ldots, J_r) \in \mathbb{R}^{\kappa \times \kappa},$$

where J_i is the Jordan block of size κ_i with the eigenvalue λ_0, and $\kappa = \sum_{j=1}^{r} \kappa_j$. The pair of matrices $(X(\lambda_0), J(\lambda_0))$ is called *Jordan pair* of $P(\lambda)$ corresponding to λ_0. The characterisation of Jordan pair is given by the next theorem

Theorem 1.7 ([32]). *Let $(\widehat{X},\widehat{J})$ be a pair of matrices, where $\widehat{X}$ is an $n\times p$ matrix and $\widehat{J}$ is a $p\times p$ Jordan matrix with unique eigenvalue λ_0. Then the following conditions are necessary and sufficient in order that $(\widehat{X},\widehat{J})$ be a Jordan pair of $P(\lambda)=\sum_{\ell=0}^{k}\lambda^{\ell}A_{\ell}$ corresponding to λ_0:*

(i) $\det P(\lambda)$ *has a zero* λ_0 *of multiplicity* p,

(ii) $$\operatorname{rank}\begin{pmatrix}\widehat{X}\\ \widehat{X}\widehat{J}\\ \vdots\\ \widehat{X}\widehat{J}^{k-1}\end{pmatrix}=p,$$

(iii) $A_k\widehat{X}\widehat{J}^{k}+A_{k-1}\widehat{X}\widehat{J}^{k-1}+\ldots+A_0\widehat{X}=\mathbf{0}$.

Let p be the number of different eigenvalues of $P(\lambda)$, and take the corresponding Jordan pair $(X(\lambda_j),J(\lambda_j))$ for every eigenvalue λ_j of $P(\lambda)$. The *finite Jordan pair* (X_F,J_F) of $P(\lambda)$ is

$$\begin{aligned}X_F&=\begin{pmatrix}X(\lambda_1) & X(\lambda_2) & \ldots & X(\lambda_p)\end{pmatrix},\\ J_F&=\operatorname{diag}(J(\lambda_1),J(\lambda_2),\ldots,J(\lambda_p)).\end{aligned}\tag{1.20}$$

Some useful facts about finite Jordan pair are:

- $X_F\in\mathbb{R}^{n\times\nu},J_F\in\mathbb{R}^{\nu\times\nu}$, where $\nu=\deg\det P(\lambda)$
- (X_F,J_F) is not determined uniquely
- (X_F,J_F) does not determine $P(\lambda)$ uniquely.

Because of the last fact, we need to define Jordan pair for infinite eigenvalue. This Jordan pair is defined as the Jordan pair for the reversed matrix polynomial $\operatorname{rev}P(\lambda)=\lambda^k P(\lambda^{-1})$ at eigenvalue zero. Denote

$$\begin{aligned}X_\infty&=\begin{pmatrix}\psi_{10} & \ldots & \psi_{1,s_1-1} & \psi_{20} & \ldots & \psi_{2,s_2-1} & \ldots & \psi_{q0} & \ldots & \psi_{q,s_q-1},\end{pmatrix}\\ J_\infty&=\operatorname{diag}(J_{\infty 1},J_{\infty 2},\ldots,J_{\infty q}),\end{aligned}\tag{1.21}$$

where $J_{\infty j}$ is the Jordan block of size s_j with eigenvalue zero. The pair (X_∞,J_∞) is called *infinite Jordan pair* of $P(\lambda)$. The characterisation is given in the following theorem.

Theorem 1.8 ([32]). *Let $(\widehat{X},\widehat{J})$ be a pair of matrices, where $\widehat{X}$ is $n\times p$ and $\widehat{J}$ is a $p\times p$ Jordan matrix with unique eigenvalue $\lambda_0=0$. Then the following conditions are necessary and sufficient in order that $(\widehat{X},\widehat{J})$ be an infinite Jordan pair of $P(\lambda)=\sum_{i=0}^{k}\lambda^i A_i$:*

(i) $\det(\lambda^k P(\lambda^{-1}))$ *has a zero at* $\lambda_0=0$ *of multiplicity* p,

(ii) $\operatorname{rank}\begin{pmatrix} \widehat{X} \\ \widehat{X}\widehat{J} \\ \vdots \\ \widehat{X}\widehat{J}^{k-1} \end{pmatrix} = p,$

(iii) $A_0\widehat{X}\widehat{J}^k + A_1\widehat{X}\widehat{J}^{k-1} + \ldots + A_k\widehat{X} = \mathbf{0}.$

1.2.4 Invariant Pairs

For given matrix polynomial $P(\lambda)$, a pair $(X,S) \in \mathbb{C}^{n\times\ell} \times \mathbb{C}^{n\times\ell}$ is called *invariant* if

$$\mathbb{P}(X,S) := A_0X + A_1XS + A_2XS^2 + \ldots + A_kXS^k = \mathbf{0}. \tag{1.22}$$

The definition of invariant pair is independent of the choice of the basis for X. When working with matrix polynomials we cannot define invariant subspace in the same way it was defined for the single matrix because the set of all eigenvectors of matrix polynomials is not linearly independent. Hence, given a full rank matrix X that is know to be a part of invariant pair for some matrix polynomial P may not uniquely determine the matrix S such that equation (1.22) holds. This is why we work with pairs instead of single matrices.

Simple Invariant Pair. In order to allow rank deficiencies in he matrix X of an invariant pair (X,S), Betcke and Kressner, in [6], introduced the notion of minimality. Namely, a pair $(X,S) \in \mathbb{C}^{n\times\ell} \times \mathbb{C}^{n\times\ell}$ is said to be *minimal* if there exists $m \in \mathbb{N}$ such that

$$V_m(X,S) := \begin{pmatrix} XS^{m-1} \\ \vdots \\ XS \\ X \end{pmatrix} \tag{1.23}$$

has full column rank. The smallest such m is called *minimality index* of (X,S).
They showed that it is always possible to extract the minimal pair from an invariant pair, thus it is enough to work with minimal pairs.
As generalization of simple eigenvalue, [6] defined *simple invariant pairs* (X,S) as invariant pairs which are minimal and the algebraic multiplicities of the eigenvalues of S are identical to the algebraic multiplicities of the corresponding eigenvalues of P.

Perturbation theory. Here we present the first order perturbation theory developed in [6]. The objective is to study the change of invariant pair (X,S) under the small perturbations of the coefficient matrices of the polynomial

$$(P+\Delta P)(\lambda) = (A_0+E_0) + \lambda(A_1+E_1) + \ldots + \lambda^k(A_k+E_k), \tag{1.24}$$

for general matrices $E_0, \ldots, E_k$. For given matrix polynomial P, define nonlinear matrix operator

$$\begin{aligned} \mathbb{P} : \mathbb{C}^{n\times \ell} \times \mathbb{C}^{\ell\times\ell} &\to \mathbb{C}^{n\times\ell}, \\ (X,S) &\mapsto A_0X + A_1XS + \ldots + A_kXS^k. \end{aligned} \tag{1.25}$$

By (1.22), a simple invariant pair satisfies $\mathbb{P}(X,S) = \mathbf{0}$. To this, we add "normalization condition", $W^*V_m(X,S) = \mathbb{I}$, where m is not smaller than the minimality index of (X,S) and the columns of W form an orthonormal basis of $\mathrm{span}(V_m(X,S))$. Now, we can formulate the problem as finding the pair $(\widetilde{X},\widetilde{S})$ such that

$$(\mathbb{P}+\Delta\mathbb{P})(\widetilde{X},\widetilde{S}) = \mathbf{0}, \;\; W^*V_m(\widetilde{X},\widetilde{S}) - \mathbb{I} = \mathbf{0}, \tag{1.26}$$

where $\mathbb{P}+\Delta\mathbb{P}$ is defined as in (1.25), but with perturbed coefficients.
The first order sensitivity of (X,S) under the perturbation is given in the following theorem.

Theorem 1.9 ([6]). *Let (X,S) be a simple invariant pair for a regular matrix polynomial P. For sufficiently small $\|\Delta P\| := \|(E_0,\, E_1,\, \ldots,\, E_k)\|_F$ the perturbed polynomial $P+\Delta P$ has a simple invariant pair $(\widetilde{X},\widetilde{S})$ satisfying*

$$(\widetilde{X},\widetilde{S}) = (X,S) - (I - \mathrm{Proj}) \circ \mathbb{L}^{-1}(\Delta\mathbb{P}(X,S),0) + \mathscr{O}(\|\Delta P\|^2), \tag{1.27}$$

where Proj *is the orthogonal projector onto the tangent space $T_{(X,S)}\mathscr{M} = \{(XM, SM-MS) : M \in \mathbb{C}^{\ell\times\ell}\}$ and*

$$\begin{aligned} \mathbb{L} : \mathbb{C}^{n\times\ell} \times \mathbb{C}^{\ell\times\ell} &\to \mathbb{C}^{n\times\ell} \times \mathbb{C}^{\ell\times\ell} \\ (\Delta X, \Delta S) &\mapsto (\mathbb{L}_P(\Delta X,\Delta S), \mathbb{L}_V(\Delta X,\Delta S)), \end{aligned} \tag{1.28}$$

$$\mathbb{L}_P : (\Delta X,\Delta S) \mapsto \mathbb{P}(\Delta X, S) + \sum_{j=1}^{k} A_j X \mathbb{D}S^j(\Delta S), \tag{1.29}$$

$$\mathbb{L}_V : (\Delta X,\Delta S) \mapsto W_0^H \Delta X + \sum_{j=1}^{m-1} W_j^H(\Delta X S^j + X\mathbb{D}S^j\Delta S)), \tag{1.30}$$

$$\mathbb{D}S^j : \Delta S \mapsto \sum_{i=0}^{j-1} S^i \Delta S S^{j-i-1}. \tag{1.31}$$

Here, $\mathscr{M} = \{(XT, T^{-1}ST) :\; T \in \mathbb{C}^{k\times k} \text{ invertible}\} \subset \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ is a manifold of invariant pairs generated by (X,S). Since we are evaluating the sensitivity of (X,S) under perturbations, the components of the error term $(\widetilde{X},\widetilde{S}) - (X,S)$ that are contained in $\mathscr{M}$ are neglected, and this is achieved by projecting out the components of $\mathbb{L}^{-1}(\Delta\mathbb{P}(X,S),\mathbf{0})$ contained in $T_{(X,S)}\mathscr{M}$.

1.3 Applications of polynomial eigenvalue problem

The polynomial eigenvalue problem arises in a variety of applications in natural sciences and engineering. The most common is the quadratic eigenvalue problem which appears in vibration analysis of mechanical systems, acoustics, fluid mechanics, and more. Moreover, quartic eigenvalue problem occurs in calibration of the central catadioptric vision system and spatial stability analysis of the Orr Sommerfeld equation. In this section we present two applications of the quadratic eigenvalue problem, which is the main focus of the thesis, and one application of the quartic eigenvalue problem.

1.3.1 Disk brake squeal

The quadratic eigenvalue problem arises in modelling and analysis of disk brakes [34]. In particular, one is interested only in eigenvalues with positive real part to determine the possibility of brake squeal.

The brake noise generation mechanisms are described in [1]. The ideal brake consists of a pair of pads that squeezes a rotating disk with a constant friction coefficient, and there are normal and tangential forces acting on the interface of pads and rotor. During the stationary contact the forces are uniformly distributed. However, during the relative motion the forces develop non-uniform distribution. The analysis of possible sources of instabilities is based on lab experiments, on numerical simulations based on finite element models, or on idealized minimal models mimicking the physics of a real brake [34]. We will consider here the finite element model and macroscopic equation of motion arising from it, as in [34]:

$$M_\Omega \ddot{u} + D_\Omega \dot{u} + K_\Omega u = f. \tag{1.32}$$

The terms in (1.32) are:

- $M_\Omega \in \mathbb{R}^{n\times n}$ represents the *mass matrix*, collecting acceleration terms; it is symmetric positive semidefinite;
- $D_\Omega \in \mathbb{R}^{n\times n}$ collects *damping and gyroscopic effects*, collecting velocity terms, typically nonsymmetric;
- $K_\Omega \in \mathbb{R}^{n\times n}$ collects *stiffness and circulatory effects*, collecting displacement terms, typically nonsymmetric;
- Ω is parameter vector;
- f is external force, $f \equiv \mathbf{0}$ for self-excited vibrations;
- $u : \mathbb{R} \to \mathbb{R}^n$ contains the coordinates in the FE basis of the displacements;

- $\dot{u}$ contains components of the velocity;

- $\ddot{u}$ contains components of the acceleration.

The coefficient matrices can depend on one or more parameters represented by Ω, typically including operating conditions (temperature, pad pressure, etc.), material properties (friction coefficient, brake geometry and mass distribution, effects of wear and damping, etc.) and rotation speed of the brake disk.

Above we mentioned that the brake squeal is a product of flutter-type instabilities. This type of instabilities is indicated by the coalescing of eigenvalues on the real axis, or by eigenvalues with positive real part of the quadratic eigenvalue problem

$$(\lambda^2 M_\Omega + \lambda D_\Omega + K_\Omega)x = \mathbf{0}. \tag{1.33}$$

The quadratic eigenvalue problem (1.33) is obtained by considering the homogeneous system of equations (1.32), i.e., $f = \mathbf{0}$. The general solution to the homogeneous problem can be written as

$$u(t) = \sum_{k=1}^{2n} \alpha_k x_k e^{\lambda_k t},$$

where (λ_k, x_k) are eigenpairs of (1.33).

The eigenvalues with positive real part of the problem (1.33) are usually called *unstable eigenvalues*, and the goal in this application is to determine those eigenvalues. It is important to have an efficient algorithm for computing these eigenvalues mostly because our problem is usually large scale and it has to be executed for many values of the parameter Ω.

Derivation of the model. Description of complicated dynamical systems, such as disk brakes, is usually developed using the Langevin equation. In this approach, one observes collective, macroscopic variables which are changing only slowly relative to other microscopic variables of the system. Those variables are degrees of freedom. Now, the Langevin equation describes the time evolution of a subset of the degrees of freedom. However, this kind of simulation is not computationally feasible. This is why the linearized finite element (FE) model is usually used in practice. It formulates the equations of motion assuming a very simplified description of the forcing term arising from a macroscopic friction law, and the results obtained from this model are useful [34].

In this model, one is interested in stability analysis of disk brakes which is done by computing the eigenvalues and eigenmodes. In particular, if our model has eigenvalues with positive real part then a self-excited vibration induced by friction may arise and in real model this can be represented by audible squeal.

The "zeroth" step of the analysis is the initial state of the brake. At this point the brake is stationary and unloaded. All possible contact zones are defined although they are not in contact

yet. The rotation of the disk is neglected. Finally, the equation of motion is

$$M\ddot{u} + D_M\dot{u} + K_E u = \mathbf{0}. \tag{1.34}$$

The matrix M represents mass, and is symmetric positive definite, the matrix D_M represents damping and it is symmetric positive semi-definite, the matrix K_E represents elastic stiffness and it is symmetric positive definite.

The first step is linear static analysis. One investigates the disk with the external load from the brake pad. The goal of the linear static analysis is to provide a location of contact and the normal and friction forces in the contact area. The disk is considered stationary, but to map the friction force at the contact correctly, velocity field information is assigned to each FE node. Further refinement of the model is obtained by considering the state of contact frozen and the contact points constrained in normal direction with multi-point constraints (MPCs). Equations of motion are

$$M\ddot{u} + (D_M + \frac{1}{\Omega}D_R)\dot{u} + (K_E + K_R)u = f. \tag{1.35}$$

Here K_R is nonsymmetric matrix describing circulatory effects, Ω is a parameter representing the rotational speed of the disk, and D_R is symmetric matrix describing the friction induced damping.

The second step is linear static analysis with centrifugal loads. One modifies the previous model by introducing the rotation of the disk brake. Instead of moving the nodes, they are applied with the load resulting from centrifugal forces. This analysis provides internal stress conditions. Equations of motions are

$$M\ddot{u} + (D_M + \frac{1}{\Omega}D_R + \Omega D_G)\dot{u} + (K_E + K_R + \Omega^2 K_g)u = f. \tag{1.36}$$

D_G is skew symmetric matrix (gyroscopic term) and K_g is symmetric matrix modelling the geometric stiffness.

Notice that in this model, we have only one parameter Ω representing the disk speed. In general, the coefficient matrices can depend on more than one parameter.

1.3.2 Regularized Total Least Squares

In [62], a new approach for solving regularized total least squares has been developed which includes solving the quadratic eigenvalue problem several times. Precisely, the rightmost eigenvalue and the corresponding eigenvector of certain quadratic eigenvalue problem is computed.

Total least squares (TLS) is a technique for solving overdetermined linear system of equations

$$Ax \approx b,\ A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ x \in \mathbb{R}^n\ (m > n). \tag{1.37}$$

Here, both the coefficient matrix A and the vector b are subject to errors. Problem (1.37) is

actually an optimization problem

$$\min_{x,\overline{A},\overline{b}} \| \begin{pmatrix} A & b \end{pmatrix} - \begin{pmatrix} \overline{A} & \overline{b} \end{pmatrix} \|_F^2 \text{ subject to } \overline{A}x = \overline{b}. \tag{1.38}$$

When using the ordinary least squares (LS) method for solving (1.38) we assume that the coefficient matrix A is error free and that b contains all the errors. However, in practice, all data are contaminated by noise and thus total least squares (TLS) approach should be used. Methods developed for TLS are based on the SVD decomposition and they also deal with problems when only some columns of A are contaminated by noise, and the remaining ones are noise free. When the matrix A is ill conditioned, both of these methods, LS and TLS might give a solution that is physically meaningless and certain regularization is needed in order to decrease the effect of the ill conditioning and data noise. This is why the Regularized Total Least Squares (RTLS) problem formulation is introduced. It imposes a quadratic constraint on the solution vector x in (1.38). This new constrained problem cannot be solved using SVD, and in [62] the new approach based on solution of a quadratic eigenvalue problem is developed. Here we present this method. It is referred to as a quadratically constrained formulation.
RTLS is formulated as follows

$$\min_{x,\overline{A},\overline{b}} \| \begin{pmatrix} A & b \end{pmatrix} - \begin{pmatrix} \overline{A} & \overline{b} \end{pmatrix} \|_F^2, \text{ subject to } \overline{A}x = \overline{b}, \ \|Lx\|_2^2 \leq \delta^2, \tag{1.39}$$

where $L \in \mathbb{R}^{p\times n}, p \leq n$ and $\delta > 0$. It is known that the objective function in (1.39) can be replaced by orthogonal distance $\frac{\|Ax-b\|_2^2}{1+\|x\|_2^2}$, so the problem reads as

$$\min_x \frac{\|Ax-b\|_2^2}{1+\|x\|_2^2} \text{ subject to } \overline{A}x = \overline{b}, \ \|Lx\|_2^2 = \delta^2, \tag{1.40}$$

for δ small enough (i.e., $\delta < \|Lx_{\text{TLS}}\|_2$). Since the norm $\|Lx_{\text{TLS}}\|_2$ can be large for ill conditioned problem (1.37), the assumption that δ is small enough can be considered guaranteed in practice, and thus the inequality in (1.39) can be replaced by equality. In practice, L is usually chosen to be approximation of the first or second-order derivative operators in order to impose a certain degree of smoothness in the solution.

So, where does the quadratic eigenvalue problem come from? Write the Lagrangean for the RTLS problem (1.40)

$$\mathscr{L}(x,\lambda) = \frac{\|Ax-b\|_2^2}{1+\|x\|_2^2} + \lambda(\|Lx\|_2^2 - \delta^2). \tag{1.41}$$

The first order optimality conditions are

$$B(x)x + \lambda L^T Lx = d(x), \ \|Lx\|_2^2 = \delta^2, \tag{1.42}$$

where

$$B(x) = \frac{A^T A}{1+\|x\|_2^2} - \frac{\|Ax-b\|_2^2}{(1+\|x\|_2^2)^2} I_n,\ d(x) = \frac{A^T b}{1+\|x\|_2^2}. \tag{1.43}$$

This system (1.43) is solved iteratively, where in every iteration we find x_{k+1} and λ_{k+1} which solve the system

$$B(x_k)x + \lambda L^T L x = d_k := d(x_k),\ \|Lx\|^2 = \delta^2, \tag{1.44}$$

corresponding to the eigenvalue with the largest real part λ using an equivalent quadratic eigenvalue problem.

In order to derive the QEP formulation let us dismiss the index k from (1.44), and consider that B is symmetric matrix. We distinguish two cases, when L is square and invertible, and when L is nonsquare.

L square and invertible. Impose a change of variable $z = Lx$ to get

$$\underbrace{L^{-T} B L^{-1}}_{=:W,\ \text{symmetric}} z + \lambda z = \underbrace{L^{-T} d}_{=:h},\ z^T z = \delta^2. \tag{1.45}$$

Solving this system is equivalent to finding the rightmost eigenvalue and the corresponding eigenvector for certain quadratic eigenvalue problem. Assuming that λ is large enough so that $W + \lambda I$ is positive definite, denote $u = (W+\lambda I)^{-2} h$. Now, $h^T u = z^T z = \delta^2$ and $h = \delta^{-2} h h^T u$, so we can write the condition (1.45) as $(W+\lambda I)^2 u = h$ which can be written as QEP

$$(\lambda^2 I + 2\lambda W + W^2 - \delta^{-2} h h^T) u = \mathbf{0}. \tag{1.46}$$

We are interested in the rightmost eigenvalue λ and the corresponding eigenvector u scaled so that $h^T u = \delta^2$. Now, the solution of the original problem is recovered by first computing $z = (W+\lambda I)u$ and then $x = L^{-1} z$.

Nonsquare L. In this case $L^T L$ is singular, because its rank is equal to the minimum of number of columns and number of rows. We write eigenvalue decomposition $L^T L = U S U^T$. Equivalent form of (1.44) is

$$U^T B U \underbrace{U^T x}_{=:y} + \lambda S y = U^T d,\ y^T S y = \delta^2. \tag{1.47}$$

Let $r = \operatorname{rank}(S)$ and $S_1 = S(1:r, 1:r)$. Partitioning elements of (1.47) with respect to r we get

$$\begin{cases} T_1 y_1 + T_2 y_2 + \lambda S_1 y_1 & = d_1, \\ T_2^T y_1 + T_4 y_2 & = d_2 \end{cases},\ y_1^T S_1 y_1 = \delta^2. \tag{1.48}$$

For the sake of simplicity, we will assume that T_4 is invertible and thus we can express

$$y_2 = T_4^{-1}(d_2 - T_2^T y_1). \tag{1.49}$$

If we place (1.49) in first equation in (1.48) we get

$$(T_1 - T_2T_4^{-1}T_2^T + \lambda S_1)y_1 = (d_1 - T_2T_4^{-1}d_2),$$

which is the system of form $(W + \lambda I)^2 u = h$ for $W = S_1^{-1/2}(T_1 - T_2T_4^{-1}T_2^T)S_1^{-1/2}$ and $h = S_1^{1/2}(d_1 - T_2T_4^{-1}d_2)$, as before.
The solution of (1.48) is given by

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} S_1^{-1/2}z \\ T_4^{-1}(d_2 - T_2^T S_1^{-1/2}z) \end{pmatrix}. \tag{1.50}$$

The final solution is $x = Uy$.

1.3.3 Orr-Sommerfeld equation

The quartic eigenvalue problem appears in the analysis of the stability of plane Poiseuille flow in a channel. In the case of Poiseuille flow, the undisturbed stream velocity is $U(y) = 1 - y^2$ in the x direction. The side walls are at $y = -1$ and $y = 1$. The Reynolds number is $R = 1/\nu$,

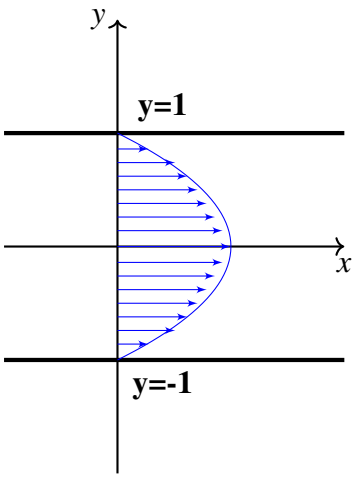


Figure 1.2: Poiseuille flow

where ν is the kinematic viscosity. The stability of the flow depends on the Reynolds number. The goal is to find the critical Reynolds number for which the flow becomes unstable.
In this example, the y component of the perturbation velocity is considered to be, as in [10], proportional to the real part of

$$\Phi(x,y,t) = \phi(y)e^{i(\lambda x - \omega t)}, \tag{1.51}$$

where λ is the wavenumber and ω is the angular frequency. By the linearization of the Navier-Stokes equations for the velocity perturbation (1.51), the Orr-Sommerfeld equation is obtained

$$\left[\left(\frac{d^2}{dy^2}-\lambda^2\right)^2 - iR\left\{(\lambda U-\omega)\left(\frac{d^2}{dy^2}-\lambda^2\right)-\lambda U''\right\}\right]\phi = 0, \tag{1.52}$$

with the boundary conditions

$$\phi(y) = 0,\ \ \phi'(y) = 0 \text{ at } y = \pm 1. \tag{1.53}$$

Discretization of the equation (1.52) leads to a quartic eigenvalue problem. The eigenvalue of interest are those closest to the real axis, and the system is stable if the imaginary part of eigenvalue is positive. We will consider the discretization using the Chebyshev polynomials, as in [55] and [5], that is ϕ is expanded in $[-1,1]$ as

$$\phi(y) = \sum_{n=0}^{\infty} a_n T_n(y), \tag{1.54}$$

where $T_n(\cos(\theta)) = \cos(n\theta)$ and

$$a_n = \frac{2}{\pi c_n}\int_{-1}^{1}\phi(y)T_n(y)\sqrt{1-y^2}dy,\ \ c_0 = 2, c_n = 1 \text{ for } n > 0. \tag{1.55}$$

The approximate solution is of form

$$\phi(y) = \sum_{n=0}^{N} a_n T_n(y). \tag{1.56}$$

Let D_N represent the Chebyshev differentiation matrix. The entries of D_N are given in [68]

$$\begin{aligned}
(D_N)_{11} &= \frac{2N^2+1}{6},\ \ (D_N)_{N+1,N+1} = -\frac{2N^2+1}{6},\\
(D_N)_{jj} &= \frac{-x_{j-1}}{2(1-x_{j-1}^2)},\ \ j = 2,\ldots,n,\\
(D_N)_{ij} &= \frac{c_i}{c_j}\frac{(-1)^{i+j}}{(x_{i-1}-x_{j-1})},\ \ i \neq j,\ i,j = 2,\ldots,N,
\end{aligned}$$

where $c_i = \begin{cases} 2 & i = 1, N+1 \\ 1 & \text{otherwise} \end{cases}$, and $x_j = \cos(j\pi/N), j = 0,\ldots,N$. The higher order derivatives are obtained as powers of D_N. By plugging in the derivative matrices D_N^j instead of $\frac{d^j}{dy^j}$ in (1.52) we derive the quartic pencil $\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E$ with

$$\begin{aligned}
A &= \mathbb{I}, & B &= iR\,\mathrm{diag}(1-x_i^2),\\
C &= -(i\omega R\mathbb{I} + 2D_N^2), & D &= -iR\,\mathrm{diag}(1-x_i^2)D_N^2 - 2iR\mathbb{I},\\
E &= D_N^4 + iR\omega D_N^2.
\end{aligned}$$

1.4 Linearizations of Matrix Polynomials

The most common approach when dealing with polynomial eigenvalue problem is to define an equivalent linear problem, i.e. to linearize it, and then work with the larger linear matrix pencil. The eigenvalues of the equivalent problems are the same, and there is an explicit connection between the corresponding eigenvectors. In this section we present most used linearizations and define the vector spaces of linearizations. Their most desirable property would be that the Jordan structure of the eigenvalues is preserved, and we will emphasize the linearizations with this property.

Definition 1.1. *Let $P(\lambda)$ be an $n \times n$ matrix polynomial of degree $k \geq 1$. A pencil $L(\lambda) = \lambda X + Y$ with $X, Y \in \mathbb{R}^{kn \times kn}$ is called a* linearization *of $P(\lambda)$ if there exist matrix polynomials $E(\lambda)$ and $F(\lambda)$, with constant nonzero determinant, so that*

$$E(\lambda)L(\lambda)F(\lambda) = \begin{pmatrix} P(\lambda) & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_{(k-1)n} \end{pmatrix}. \tag{1.57}$$

The most important and the most used linerizations in practice are the *first companion form* $C_1(\lambda) = \lambda X_1 + Y_1$ and the *second companion form* $C_2(\lambda) = \lambda X_2 + Y_2$ where

$$X_1 = \begin{pmatrix} A_k & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{pmatrix}, \quad Y_1 = \begin{pmatrix} A_{k-1} & A_{k-2} & \cdots & A_0 \\ -I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -I & 0 \end{pmatrix}, \tag{1.58}$$

$$X_2 = \begin{pmatrix} A_k & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{pmatrix}, \quad Y_2 = \begin{pmatrix} A_{k-1} & -I & \cdots & 0 \\ A_{k-2} & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -I \\ A_0 & 0 & \cdots & 0 \end{pmatrix}. \tag{1.59}$$

When all the eigenvalues of $P(\lambda)$ are finite, the Jordan structure can be recovered from any linearization. However, when the infinite eigenvalues are present this is not the case. So we define that the linearization $L(\lambda)$ for $P(\lambda)$ is a *strong linearization* if, in addition, $\operatorname{rev} L(\lambda)$ is a linearization for $\operatorname{rev} P(\lambda)$.

Vector spaces of linearizations. In [50], Mackey et al. defined vector spaces of matrix pencils which generalize the first and the second companion form. They proved that all pencils, which are linearizations, from these spaces are also strong linearizations. Here, we present those spaces and some of their important properties.

The definitions are

$$\mathbb{L}_1(P) := \{L(\lambda) = \lambda X + Y : X, Y \in \mathbb{R}^{nk\times nk}, L(\lambda)\cdot(\Lambda \otimes I_n) \in \mathscr{V}_P\}, \tag{1.60}$$

$$\mathbb{L}_2(P) := \{L(\lambda) = \lambda X + Y : X, Y \in \mathbb{R}^{nk\times nk}, (\Lambda^T \otimes I_n)\cdot L(\lambda) \in \mathscr{W}_P\}, \tag{1.61}$$

where $\Lambda(r) = \begin{pmatrix} r^{k-1} & r^{k-2} & \cdots & r & 1 \end{pmatrix}^T$ and

$$\mathscr{V}_P = \{v \otimes P(\lambda) : v \in \mathbb{R}^k\}, \tag{1.62}$$

$$\mathscr{W}_p = \{w^T \otimes P(\lambda) : w \in \mathbb{R}^k\}. \tag{1.63}$$

Here $\otimes$ represents the Kronecker product, i.e., for matrices $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$ the matrix $A \otimes B \in \mathbb{C}^{mp\times nq}$ is the block matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \dots & a_{mn}B \end{pmatrix}.$$

It is proven [50] that (1.60) and (1.61) are vector spaces, and that they have the same dimension $k(k-1)n^2 + k$. In order to introduce the characterization of these definitions, from which it is easier to construct the linearization, the *column shifted sum* for block matrices X and Y of the form

$$X = \begin{pmatrix} X_{11} & \dots & X_{1k} \\ \vdots & \ddots & \vdots \\ X_{k1} & \dots & X_{kk} \end{pmatrix}, \; Y = \begin{pmatrix} Y_{11} & \dots & Y_{1k} \\ \vdots & \ddots & \vdots \\ Y_{k1} & \dots & Y_{kk} \end{pmatrix}, \; X_{ij}, Y_{ij} \in \mathbb{C}^{n\times n}$$

is introduced as

$$X \boxplus\!\!\rightarrow Y = \begin{pmatrix} X_{11} & \dots & X_{1k} & \mathbf{0}_n \\ \vdots & \ddots & \vdots & \vdots \\ X_{k1} & \dots & X_{kk} & \mathbf{0}_n \end{pmatrix} + \begin{pmatrix} \mathbf{0}_n & Y_{11} & \dots & Y_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_n & Y_{k1} & \dots & Y_{kk} \end{pmatrix}, \tag{1.64}$$

and the *row shifted sum* as

$$X \boxplus\!\!\downarrow Y = \begin{pmatrix} X_{11} & \dots & X_{1k} \\ \vdots & \ddots & \vdots \\ X_{k1} & \dots & X_{kk} \\ \mathbf{0}_n & \dots & \mathbf{0}_n \end{pmatrix} + \begin{pmatrix} Y_{11} & \dots & Y_{1k} \\ \vdots & \ddots & \vdots \\ Y_{k1} & \dots & Y_{kk} \\ \mathbf{0}_n & \dots & \mathbf{0}_n \end{pmatrix}. \tag{1.65}$$

Now, it can be proven that

$$\mathbb{L}_1(P) = \left\{\lambda X + Y : X \boxplus\!\!\rightarrow Y = v \otimes \begin{pmatrix} A_k & A_{k-1} & \dots & A_0 \end{pmatrix}, v \in \mathbb{C}^k\right\}, \tag{1.66}$$

$$\mathbb{L}_2(P) = \left\{ \lambda X + Y : X \boxplus Y = w^T \otimes \begin{pmatrix} A_k \\ \vdots \\ A_0 \end{pmatrix}, w \in \mathbb{C}^k \right\}. \tag{1.67}$$

In addition, the theorem which gives an algorithm for determining if a pencil is a linearization is proven as well.
Finally, we state the theorems about recovery of both right and left eigenvectors.

Theorem 1.10 ([50]). *Let $P(\lambda)$ be an $n \times n$ matrix polynomial of degree k, and $L(\lambda)$ any pencil in $\mathbb{L}_1$ with nonzero right eigenvector z. Then $x \in \mathbb{C}^n$ is an eigenvector for $P(\lambda)$ with finite eigenvalue $\lambda \in \mathbb{C}$ if and only if $\Lambda \otimes x$ is an eigenvector for $L(\lambda)$ with eigenvalue λ. If, in addition, P is regular and $L \in \mathbb{L}_1(P)$ is a linearization for P, then every eigenvector of L with finite eigenvalue λ is of the form $\Lambda \otimes x$ for some eigenvector x of P.*

Theorem 1.11 ([42]). *Let $L \in \mathbb{L}_1(P)$ be a linearization of P, with vector v in (1.66). If u is a left eigenvector of L with eigenvalue λ then*

$$y = (v^* \otimes I)u \tag{1.68}$$

is a left eigenvector of P with eigenvalue λ. Moreover, any left eigenvector of P corresponding to λ can be recovered from one of L from the formula (1.68).

Theorem 1.12 ([42]). *Let $L \in \mathbb{L}_2(P)$ be a linearization of P, with vector w in (1.67). If z is a right eigenvector of L with eigenvalue λ then*

$$x = (w^T \otimes I)z \tag{1.69}$$

is a right eigenvector of P with eigenvalue λ. Moreover, any right eigenvector of P corresponding to λ can be recovered from one of L from the formula (1.69).

Theorem 1.13 ([50]). *Let $P(\lambda)$ be an $n \times n$ matrix polynomial of degree k, and $L(\lambda)$ any pencil in $\mathbb{L}_2$ with nonzero left vector u. Then $y \in \mathbb{C}^n$ is a left eigenvector for $P(\lambda)$ with finite eigenvalue $\lambda \in \mathbb{C}$ if and only if $\overline{\Lambda} \otimes y$ is an eigenvector for $L(\lambda)$ with eigenvalue λ. If, in addition, P is regular and $L \in \mathbb{L}_2(P)$ is a linearization for P, then every left eigenvector of L with finite eigenvalue λ is of form $\overline{\Lambda} \otimes y$ for some left eigenvector y of P.*

Theorem 1.14 ([50]). *Let $P(\lambda)$ be an $n \times n$ matrix polynomial of degree k, and $L(\lambda)$ any pencil in $\mathbb{L}_1$ (resp., $\mathbb{L}_2$) with nonzero right (left) vector w. Then $x \in \mathbb{C}^n$ is a right (left) eigenvector for $P(\lambda)$ with infinite eigenvalue if and only if $e_1 \otimes x$ is a right (left) eigenvector for $L(\lambda)$ with infinite eigenvalue. If, in addition, P is regular and $L \in \mathbb{L}_1(P)$ (resp., $\mathbb{L}_2(P)$ is a linearization for P, then every right (left) eigenvector of L with infinite eigenvalue is of form $e_1 \otimes x$ for some right (left) eigenvector x of P with infinite eigenvalue.*

From these theorems we see that the right eigenvector recovery is straightforward for the pencils in $\mathbb{L}_1$, and the left eigenvector is easy to recover for the pencils in $\mathbb{L}_2$. This is an attractive feature, which is why [50] defined the vector space $\mathbb{DL}(P) := \mathbb{L}_1(P) \cap \mathbb{L}_2(P)$, which has both properties. They derived characterisation for this space. Another significant property is that for symmetric P every pencil in $\mathbb{DL}(P)$ is also symmetric.
At the end, we present examples for two linearizations which will be used in the remaining sections.

Example 1.5 (First companion form)**.** Consider the first companion form linearization (1.58) $C_1(\lambda) \in \mathbb{L}_1(P)$. The corresponding vector v from the characterization (1.66) is $v = e_1$. Let x be the right eigenvector for $P(\lambda)$, and z the corresponding right eigenvector for $C_1(\lambda)$. Then

$$z = \Lambda \otimes x = \begin{pmatrix} \lambda^{k-1}x \\ \lambda^{k-2}x \\ \vdots \\ x \end{pmatrix}. \tag{1.70}$$

Now, let y be the left eigenvector for $P(\lambda)$, and u corresponding left eigenvector for $C_1(\lambda)$, where λ is finite and nonzero. Then

$$u = \begin{pmatrix} I \\ (\lambda A_k + A_{k-1})^* \\ \cdots \\ (\lambda^{k-1}A_k + \lambda^{k-2}A_{k-2} + \ldots + A_1)^* \end{pmatrix} y. \tag{1.71}$$

Example 1.6 (Second companion form)**.** Consider the second companion form linearization (1.59) $C_2(\lambda) \in \mathbb{L}_2(P)$. The corresponding vector w from the characterization (1.67) is $w = e_1$. Let x be the right eigenvector for $P(\lambda)$, and z the corresponding right eigenvector for $C_1(\lambda)$, λ finite nonzero. Then

$$z = \begin{pmatrix} I \\ (\lambda A_k + A_{k-1}) \\ \cdots \\ (\lambda^{k-1}A_k + \lambda^{k-2}A_{k-2} + \ldots + A_1) \end{pmatrix} x. \tag{1.72}$$

Now, let y be the left eigenvector for $P(\lambda)$, and u corresponding left eigenvector for $C_2(\lambda)$, where λ is finite and nonzero. Then

$$u = \Lambda \otimes y = \begin{pmatrix} \lambda^{k-1}y \\ \lambda^{k-2}y \\ \vdots \\ y \end{pmatrix}. \tag{1.73}$$

Linearization and invariant pairs. The connection between the invariant pairs of matrix polynomial and its linearization is given in the following lemma

Lemma 1.1 ([6]). *A minimal invariant pair (X,S) for a regular matrix polynomial is simple if and only if $(V_k(X,S),S)$ is a simple invariant pair for the corresponding companion linearization.*

The recovery of invariant pairs from linearization analogous to theorem 1.10 is given in the following theorem.

Theorem 1.15 ([6]). *Let $L(\lambda) = \lambda B + A \in \mathbb{L}_1(P)$ be a linearization of a regular matrix polynomial P. Then for every simple invariant pair $(Y,S) \in \mathbb{C}^{kn\times\ell} \times \mathbb{C}^{\ell\times\ell}$ of L there exists $X \in \mathbb{C}^{n\times\ell}$ such that $Y = V_k(X,S)$ and (X,S) is a simple invariant pair of P.*

1.5 Localization of eigenvalues of nonlinear eigenvalue problem

In this section we present the localization theorems, pseudospectral inclusion theorems and Bauer-Fike theorem for general nonlinear eigenvalue problems developed by Bindel and Hood in [7].

They study the nonlinear eigenvalue problem

$$T(\lambda)v = 0,\ v \neq \mathbf{0}, \tag{1.74}$$

where $T : \Omega \to \mathbb{C}^{n\times n}$ is analytic on the simply connected domain $\Omega \subset \mathbb{C}$, and regular, meaning that $\det(T(z)) \not\equiv 0$. The emphasis is only on finite eigenvalues.

We define the *number of eigenvalues inside* Γ, for $\Gamma \subset \mathbb{C}$, a simple closed contour, and $T(z)$ nonsingular for all $z \in \Gamma$, by the winding number

$$W_\Gamma(\det T(z)) = \frac{1}{2\pi i}\int_\Gamma \left[\frac{d}{dz}\log\det(T(z))\right] dz = \frac{1}{2\pi i}\int_\Gamma \operatorname{tr}(T(z)^{-1}T'(z))dz. \tag{1.75}$$

Now, the main lemma for the proofs of the localization theorems is the following:

Lemma 1.2 ([7]). *Suppose $T : \Omega \to \mathbb{C}^{n\times n}$ and $E : \Omega \to \mathbb{C}^{n\times n}$ are analytic and that $\Gamma \subset \Omega$ is a simple closed contour. If $T(z) + sE(z)$ is nonsingular for all $s \in [0,1]$ and all $z \in \Gamma$, then T and $T + E$ have the same number of eigenvalues inside Γ, counting the multiplicities.*

The nonlinear generalization of Gershgorin theorem states

Theorem 1.16 ([7], Nonlinear Gershgorin theorem). *Suppose $T(z) = D(z) + E(z)$, where $D,E : \Omega \to \mathbb{C}^{n\times n}$ are analytic and D is diagonal. Then for any $0 \leq \alpha \leq 1$,*

$$\Lambda(T) \subset \bigcup_{j=1}^{n} G_j^\alpha, \tag{1.76}$$

where G_j^α is the jth generalized Gershgorin region

$$G_j^\alpha = \{z \in \Omega : |d_{jj}(z)| \leq r_j(z)^\alpha c_j(z)^{1-\alpha}\}, \tag{1.77}$$

and r_j and c_j are jth absolute row and column sums of E, i.e.,

$$r_j(z) = \sum_{k=1}^{n} |e_{jk}(z)|,\ c_j(z) = \sum_{i=1}^{n} |e_{ij}(z)|. \tag{1.78}$$

Moreover, suppose that $\mathscr{U}$ is a bounded connected component of the union $\cup_j G_j^\alpha$ such that $\overline{\mathscr{U}} \subset \Omega$. Then $\mathscr{U}$ contains the same number of eigenvalues of T and D, and if $\mathscr{U}$ includes m connected components of the Gershgorin regions, it must contain at least m eigenvalues.

1.5.1 Pseudospectrum

The spectrum of a matrix A is a set of all $z \in \mathbb{C}^n$ such that resolvent operator $R(z) = (zI - A)^{-1}$ is not defined. The ε-*pseudospectrum* can be equivalently defined as [69]:

$$\Lambda_\varepsilon = \{z \in \mathbb{C} : \|(zI - A)^{-1}\|_2 > \varepsilon^{-1}\} \tag{1.79}$$

$$= \bigcup_{\|E\|_2 < \varepsilon} \Lambda(A + E) \tag{1.80}$$

$$= \{z \in \mathbb{C} : \|(z - A)v\|_2 < \varepsilon, v \in \mathbb{C}^n, \|v\|_2 = 1\}. \tag{1.81}$$

The motivation for the first definition in (1.79) is that asking if z is eigenvalue of the matrix A is the same as asking if the matrix $zI - A$ is singular. However, determination of the singularity of a matrix is not numerically robust, because arbitrary small perturbation can change the matrix from singular to regular. The better approach is to check if the norm $\|(zI - A)^{-1}\|_2$ is large, and thus the first definition of pseudospectrum.

The second definition in (1.80) is motivated by the eigenvalue perturbation theory. Namely, by this definition, ε-pseudospectrum is the set of all eigenvalues of all perturbed matrices $A + E$ with $\|E\|_2 < \varepsilon$.

The usual definition of ε-pseudospectrum for nonlinear eigenvalue problem is generalization of (1.80). For the space $\mathscr{F}$ consisting of some set of analytic matrix-valued functions of interest, the ε-pseudospectrum for $T \in \mathscr{F}$ is

$$\Lambda_\varepsilon(T) = \bigcup_{E \in \mathscr{F}, \|E\|_{\text{glob}} < \varepsilon} \Lambda(T + E), \tag{1.82}$$

where $\|E\|_{\text{glob}}$ is a global measure of the size of the perturbing function E. In [7], $\mathscr{F}$ is the space

of all analytic matrix-valued functions $C^{\omega}(\Omega, \mathbb{C}^{n\times n})$ with the global measure

$$\|E\|_{\text{glob}} \equiv \sup_{z\in\Omega} \|E(z)\|_2. \tag{1.83}$$

In this setting, three equivalent definitions of pseudospectrum, similar to (1.79)-(1.81) are provided in [7]

Theorem 1.17 ([7]). *Let $\mathscr{E} = \{E : \Omega \to \mathbb{C}^{n\times n}$, E analytic, $\sup_{z\in\Omega} \|E(z)\|_2 < \varepsilon\}$ and $\mathscr{E}_0 = \{E_0 \in \mathbb{C}^{n\times n} : \|E_0\|_2 < \varepsilon\}$. Then the following definitions are equivalent:*

$$\Lambda_\varepsilon(T) = \{z \in \Omega : \|T(z)^{-1}\|_2 > \varepsilon^{-1}\} \tag{1.84}$$

$$= \bigcup_{E\in\mathscr{E}} \Lambda(T+E) \tag{1.85}$$

$$= \bigcup_{E_0\in\mathscr{E}_0} \Lambda(T+E_0). \tag{1.86}$$

Another generalization of ε-pseudospectrum theory for linear problem is stated in the following proposition.

Proposition 1.3 ([7]). *Suppose $T : \Omega \to \mathbb{C}^{n\times n}$ is analytic and $\mathscr{U}$ is a bounded connected component of $\Lambda_\varepsilon(T)$ with $\overline{\mathscr{U}} \in \Omega$. Then $\mathscr{U}$ contains an eigenvalue of T.*

Connection with backward error is given in proposition

Proposition 1.4 ([7]). *Suppose $T(\widehat{\lambda})x = r$ and $\|r\|_2/\|x\|_2 < \varepsilon$. Then $\widehat{\lambda} \in \Lambda_\varepsilon(T)$.*

The comparison between eigenvalue problems via pseudospectra is given in the next theorem

Theorem 1.18 ([7]). *Suppose $T : \Omega \to \mathbb{C}^{n\times n}$ and $E : \Omega \to \mathbb{C}^{n\times n}$ are analytic, and let*

$$\Omega_\varepsilon \equiv \{z \in \Omega : \|E(z)\|_2 < \varepsilon\}.$$

Then

$$(\Lambda(T+E) \cap \Omega_\varepsilon) \subset (\Lambda_\varepsilon(T) \cap \Omega_\varepsilon).$$

Furthermore, if $\mathscr{U}$ is a bounded connected component of $\Lambda_\varepsilon(T)$ such that $\overline{\mathscr{U}} \subset \Omega_\varepsilon$, then $\mathscr{U}$ contains exactly the same number of eigenvalues of T and $T+E$.

1.6 Diagonalizable quadratic matrix polynomials

In this section we investigate under what assumptions we can diagonalize quadratic matrix polynomial. The diagonalization is done by congruence or direct equivalence transformation. We will also present the approach from Lancaster and Zaballa in [46] where the diagonalization is obtained by applying congruence or strict equivalence transformation to a linearization, while

preserving the structure of the original problem.
The pencil $(\lambda^2 M + \lambda C + K)$ is said to be *diagonal* or *decoupled* if M, C, K are diagonal matrices. Two pencils are *isospectral* if they have the same Jordan form, that is if they have the same eigenvalues and the same partial multiplicities. Finally, a pencil is diagonalizable if it admits an isospectral diagonal system.

Diagonalization without linearization. Here we list the quadratic pencils that allow diagonalization by congruence (Hermitian pencils) and by strict equivalence (no symmetry) without linearization. Before stating the theorem we must introduce the notion of sign characteristic. Hermitian pencils $A + \lambda B$ are congruent to pencil

$$\bigoplus_{j=1}^{r} \eta_j(\lambda - \alpha_j) \oplus \bigoplus_{j=r+1}^{s} \eta_j \begin{pmatrix} 1 & \lambda - \alpha_j \\ \lambda - \alpha_j & 0 \end{pmatrix} \oplus \bigoplus_{j=s+1}^{t} \begin{pmatrix} 0 & \lambda - (\mu_j + i\omega_j) \\ \lambda - (\mu_j - i\omega_j) & 0 \end{pmatrix},$$

where $\alpha_1, \ldots, \alpha_r$ are the real eigenvalues with partial multiplicities equal to one, $\alpha_{r+1}, \ldots, \alpha_s$ are the real eigenvalues with partial multiplicities equal to two, and $\mu_{s+1} \pm i\omega_{s+1}, \ldots, \mu_t \pm i\omega_t$ are complex conjugate pairs with partial multiplicities one. The numbers $\eta_1, \ldots, \eta_s$ take values ± 1 end represent the sign characteristic of the pencil.

Theorem 1.19 (Hermitian pencils,[46]). *Let $M, C, K \in \mathbb{C}^{n\times n}$ with $\det(M) \neq 0$, $M^* = M, C^* = C$ and $K^* = K$. Assume that $\lambda M + K$ is semisimple with all eigenvalues real and of definite type, and define*

$$\Lambda = \operatorname{diag}(\lambda_1 \mathbb{I}_1, \lambda_2 \mathbb{I}_2, \ldots, \lambda_s \mathbb{I}_s),\ S = \operatorname{diag}(\pm\mathbb{I}_1, \pm\mathbb{I}_2, \ldots, \pm\mathbb{I}_s),$$

where the size of the identity matrix I_j is a partial multiplicity of the eigenvalue λ_j for each j, and the sign of each term in S is determined by the corresponding $+1$ or -1 in the sign characteristic. Then there exists a nonsingular $U \in \mathbb{C}^{n\times n}$ such that U^*MU, U^*CU and U^*KU are diagonal if and only if $CM^{-1}K = KM^{-1}C$. If, in addition, M, C, K are real and symmetric, then there is a corresponding $U \in \mathbb{R}^{n\times n}$.

Theorem 1.20 (No symmetry, [46]). *Let $M, C, K \in \mathbb{R}^{n\times n}$ with $\det(M) \neq 0$ and assume that $\lambda M + K$ has n distinct eigenvalues. Then there exist nonsingular $U, V \in \mathbb{C}^{n\times n}$ such that $UMV = \mathbb{I}$ and UCV, UKV are diagonal if and only if $CM^{-1}K = KM^{-1}C$.*

Diagonalization by linearization. In the above theorems we saw that the certain commutativity conditions must be satisfied in order for pencil to be diagonalizable. Here, we are interested in the procedure for computing that diagonal pencil, and this is developed using the linearization

$$A = \begin{pmatrix} C & M \\ M & \mathbf{0} \end{pmatrix},\ B = \begin{pmatrix} -K & \mathbf{0} \\ \mathbf{0} & M \end{pmatrix}, \tag{1.87}$$

which is structure preserving.
Before the diagonalization of the original pencil we will first study the Jordan form of the desired diagonal pencil.

Definition 1.2 ([46]). *Let $\mathbb{J}_{n,\mathbb{C}}$ and $\mathbb{J}_{n,\mathbb{R}}$ be the classes of $2n\times 2n$ canonical Jordan matrices for $n\times n$ diagonal pencils, and $n\times n$ real diagonal pencils, respectively (so that $\mathbb{J}_{n,\mathbb{R}}\subset\mathbb{J}_{n,\mathbb{C}}\subset\mathbb{C}^{2n\times 2n}$).*

Denote by $\oplus x_j$ the direct diagonal sum of scalars or matrices $x_1,\ldots,x_k$.
Let $\lambda_1,\ldots,\lambda_t\in\mathbb{C}, 1\neq t\neq 2n$ be distinct eigenvalues, and let λ_i have partial multiplicities $\kappa_{i1}\geq\ldots\geq\kappa_{i,\mu_{g,i}}>0$ for each i. Then the eigenvalue λ_i has geometric multiplicity $\mu_{g,i}\leq n$ and the algebraic multiplicity $\mu_{a,i}=\sum_{j=1}^{\mu_{g,i}}\kappa_{ij}\leq 2n$. It holds that

$$\sum_{i=1}^{t}\sum_{j=1}^{\mu_{g,i}}\kappa_{ij}=2n. \tag{1.88}$$

Write diagonal pencil $Q(\lambda)=\bigoplus_{i=1}^{n}[m_i\lambda^2+c_i\lambda+k_i]$, where $\prod_{i=1}^{n}m_i\neq 0$. Then each diagonal entry has a linearization

$$\lambda\mathbb{I}_2-\begin{bmatrix}0 & 1\\ -k_i/m_i & -c_i/m_i\end{bmatrix},\ i=1,2,\ldots,t,$$

and $Q(\lambda)$ has the tridiagonal linearization $\lambda I-A$ where

$$A=\bigoplus_{i=1}^{n}\begin{bmatrix}0 & 1\\ -k_i/m_i & -c_i/m_i\end{bmatrix}.$$

The elementary divisors of $\lambda I-A$ are the disjoint unions of those of (1.87) and we have

$$1\leq\kappa_{ij}\leq 2,\ \text{for } 1\leq i\leq t, 1\leq j\leq\mu_{g,i}. \tag{1.89}$$

For each distinct eigenvalue λ_i, $i=1,2,\ldots,t$ we define the integers $s_i\geq 0$ by

$$\kappa_{ij}=\begin{cases}2, & j=1,2,\ldots,s_i\\ 1, & j=s_i+1,\ldots,\mu_{g,i},\end{cases} \tag{1.90}$$

$$\mu_{g,i}-s_i\leq n-p,\ i=1,2,\ldots,t. \tag{1.91}$$

Theorem 1.21 (Jordan form for diagonal pencil, [46]). *A Jordan matrix with partial multiplicities $\{\kappa_{ij}\}_{i=1,j=1}^{i=t,j=\mu_{g,i}}$ is in $\mathbb{J}_{n,\mathbb{C}}$ if and only if conditions (1.88), (1.89) and (1.91) hold where, for $i=1,2,\ldots,t$ the integers $s_i\geq 0$ appearing in (1.91) are defined by (1.90).*

Theorem 1.22 (Jordan form for real diagonal pencil, [46]). *A Jordan matrix J with partial multiplicities $\{\kappa_{ij}\}_{i=1,j=1}^{i=t,j=\mu_{g,i}}$ is in $\mathbb{J}_{n,\mathbb{R}}$ is and only if there is and n_0, $0 \le n_0 \le n$, such that $J = \operatorname{diag}(J_{n_0}, J_{n-n_0})$ for Jordan matrices J_{n_0}, J_{n-n_0} with $\sigma(J_{n_0}) \subset \mathbb{R}$ and $\sigma(J_{n-n_0}) \cap \mathbb{R} = \emptyset$ and*

(a) conditions (1.88), (1.89) and (1.91) (with n replaced by n_0) hold for J_{n_0} and

(b) $\sigma(J_{n-n_0})$ consists of conjugate pairs of nonreal semisimple eigenvalues λ_j, λ_j.

Now we consider the generalization of an isospectral diagonal system to our system $Q(\lambda)$ by the application of congruence or strict equivalence on the linearization $\lambda A - B$ in (1.87). First we define the transformation which will be used. They are all structure preserving transformations.

Definition 1.3. *(a) A system is* **DE**$\mathbb{C}$ *(diagonalizable by strict equivalence over $\mathbb{C}$) if there exist nonsingular $U, V \in \mathbb{C}^{2n\times 2n}$ such that*

$$U(\lambda A - B)V = \lambda\widehat{A} - \widehat{B},$$

where $\lambda\widehat{A} - \widehat{B}$ is the linearization of a (generally complex) diagonal system $\widehat{Q}(\lambda) = \lambda^2\widehat{M} + \lambda\widehat{C} + \widehat{K}$.

(b) A real system is **DE**$\mathbb{R}$ *if there exist nonsingular $U, V \in \mathbb{C}^{2n\times 2n}$ such that*

$$U(\lambda A - B)V = \lambda\widehat{A} - \widehat{B},$$

where $\lambda\widehat{A} - \widehat{B}$ is the linearization of a real diagonal system $\widehat{Q}(\lambda) = \lambda^2\widehat{M} + \lambda\widehat{C} + \widehat{K}$.

(c) A system is **DC**$\mathbb{R}$ *(diagonalizable by congruence) if there exist nonsingular $U \in \mathbb{C}^{2n\times 2n}$ such that*

$$U(\lambda A - B)U^* = \lambda\widehat{A} - \widehat{B},$$

where $\lambda\widehat{A} - \widehat{B}$ is the linearization of a real diagonal system $\widehat{Q}(\lambda) = \lambda^2\widehat{M} + \lambda\widehat{C} + \widehat{K}$.

Finally, we state the main theorem for this section

Theorem 1.23 ([46]). *(a) A system $Q(\lambda)$ with Jordan form J is* **DE**$\mathbb{C}$ *if and only if $J \in \mathbb{J}_{n,\mathbb{C}}$.*

(b) A real system $Q(\lambda)$ with Jordan form J is **DE**$\mathbb{R}$ *if and only if $J \in \mathbb{J}_{n,\mathbb{R}}$.*

(a) An Hermitian system $Q(\lambda)$ with Jordan form J is **DC**$\mathbb{R}$ *if and only if $J \in \mathbb{J}_{n,\mathbb{R}}$.*

1.7 Minimax theory

In [28] Duffin considered heavily damped dynamical systems. The aim of his work was to develop variational principles for overdamped systems analogous to variational principles for Hermitian matrices, i.e.

Theorem 1.24. *Let A be an $n \times n$ Hermitian matrix with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_k \leq \ldots \leq \lambda_n$. Then*

$$\lambda_k = \min_{\substack{U \\ \dim U = k}} \max_{\substack{x \in U \\ x \neq \mathbf{0}}} \frac{(Ax,x)}{(x,x)}. \tag{1.92}$$

The main tool for the theory is Rayleigh coefficient, which is replaced by the Rayleigh functional in he case of overdamped dynamical systems.
For given matrices M, C and K of order n and the associated quadratic forms

$$m(v) = (Mv,v),\ c(v) = (Cv,v),\ k(v) = (Kv,v), \tag{1.93}$$

we assume that

(i) M, C and K are symmetric,

(ii) $m(v) \geq 0$, $c(v) \geq 0$, and $k(v) \geq 0$, which is later replaced by weaker hypothesis $c(v) \geq 0$,

(iii) $c^2(v) - 4m(v)k(v) > 0$, *overdamping condition*.

If r is the rank of matrix M, then there are precisely $n + r$ roots of the equation

$$\det(\lambda^2 M + \lambda C + K) = 0, \tag{1.94}$$

which represent the eigenvalues of the quadratic eigenvalue problem

$$(\lambda^2 M + \lambda C + K)x = \mathbf{0}. \tag{1.95}$$

Duffin divided these eigenvalues into two groups, the primary and the secondary eigenvalues. Namely, $h_1 \leq h_2 \leq \ldots \leq h_r$, the r smallest roots of (1.94), are called *secondary eigenvalues*, and $k_1 \leq k_2 \leq \ldots \leq k_n$, the n largest roots, are called *primary eigenvalues*. The corresponding eigenvectors are called the *secondary eigenvectors* and *primary eigenvectors*, respectively.

1.7.1 The primary functional

The primary functional is defined as

$$p(v) = \frac{-2k(v)}{c(v) + d(v)}, \tag{1.96}$$

where

$$d(v) = \sqrt{c^2(v) - 4m(v)k(v)} > 0. \tag{1.97}$$

In order to state the main theorem, we introduce the number $P(Y)$ associated with each subspace Y of dimension one or greater by

$$P(Y) = \sup_{y \in Y} p(y). \tag{1.98}$$

The ith primary minimax value k_i is then defined as

$$k_i = \inf_{\dim Y = i} P(Y). \tag{1.99}$$

The first important theorem states that the eigenvectors of the primary eigenvalues are linearly independent, more precisely:

Theorem 1.25 ([28]). *There is an independent set of n primary eigenvectors $u_1, u_2, \ldots, u_n$. The corresponding eigenvalues are the minimax values $k_1, k_2, \ldots, k_n$. Any other primary eigenvector u is a linear combination of vectors of the set having the same eigenvalue as u.*

The minimax theorem reads as follows.

Theorem 1.26 ([28]). *If Y is a subspace of dimension ≥ 1, let*

$$P(Y) = \max_{y \in Y} p(y).$$

Then, for $i = 1, 2, \ldots, n$, the primary minimax value k_i is given by

$$k_i = \min P(Y),$$

for all subspaces of dimension i.

1.7.2 The secondary functional

The secondary functional $s(v)$ is defined for a vector v if and only if $m(v) \neq 0$ as

$$2s(v)m(v) + c(v) = -d(v). \tag{1.100}$$

A primary and a secondary eigenvectors can coincide, but the primary and secondary eigenvalues cannot. More precisely

Theorem 1.27 ([28]). *Let r be the rank of M. Then there is an independent set of r vectors $w_1, w_2, \ldots, w_r$. Each vector of the set is a secondary eigenvector. Any other secondary eigenvector is a linear combination of vectors of the set with the same eigenvalue.*

Theorem 1.28 ([28]). *The range of the primary functional and the range of the secondary functional have no common value.*

Consider the reversed quadratic eigenvalue problem which satisfies the conditions (i), (ii) and (iii). The primary functional for the reversed problem is

$$p^0(v) = \frac{-2m(v)}{c(v)+d(v)}.$$

Thus, if $m \neq 0$, then $s = 1/p^0$. So one can prove an analogous theorem to Theorem 1.26 for the secondary functional.

Chapter 2

Backward error

Backward error analysis provides an elegant way to justify the computed output: if the initial data is slightly perturbed (this perturbation is called *backward error*), then the computed (inexact) output can be reproduced by exact computation with this new data. This, of course, does not guarantee that the computed result is close to the exact one - the error depends on the sensitivity of the function we are trying to compute. If the size of the backward error is of the comparable size as the estimated uncertainty in the initial data, then we may say that the computed results is as good as warranted by te data.

In Section 2.2, we show that optimal Hermitian backward error (of the same minimal norm as in the unconstrained case) is possible for any eigenpair; this is an extension od the existing theory in which such optimal Hermitian backward error was established only for the case of real eigenvalue. The result is extended to allow both Hermitian and skew Hermitan perturbations in the coefficient matrices. Further, we derive a new more intrinsic proof of the explicit formula for the component-wise backward error.

2.1 Optimal backward error for a given eigenpair

In the case of matrix polynomial $P(\lambda)$ and its approximate eigenpair (x,λ), with λ finite, the minimal size of the normwise backward error, measured e.g. in the spectral norm $\|\cdot\|_2$, is defined by

$$\eta_P(x,\lambda) = \min\{\varepsilon \,:\, (P(\lambda)+\Delta P(\lambda))x = \mathbf{0},\ \|\Delta A_i\|_2 \le \varepsilon\|A_i\|_2,\ i=0,\ldots,k\}, \tag{2.1}$$

where $\Delta P(\lambda) = \sum_{i=0}^{k} \lambda^i \Delta A_i$ is the backward error in $P(\lambda)$, and

$$P(\lambda)+\Delta P(\lambda) = \sum_{i=0}^{k} \lambda^i (A_i + \Delta A_i).$$

In other words, we seek small perturbations ΔA_i of the coefficients A_i, that will render the computed pair (x,λ) an exact eigenpair of $P(\lambda)+\Delta P(\lambda)$.

Using the backward error to justify the computed result is usually illustrated by a commutative diagram as in Figure 2.1.

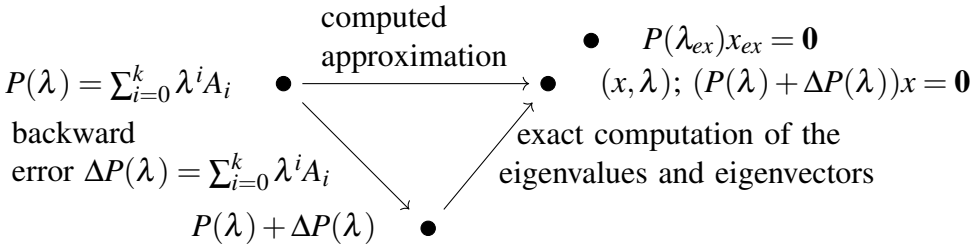


Figure 2.1: Commutative diagram for a backward perturbation in the computation of a right eigenpair (x,λ) of the matrix polynomial $P(\lambda)=\sum_{i=0}^{k}\lambda^i A_i$.

The optimal backward error (2.1) corresponds to the residual, and in practical computation it can be obtained using the explicit formula, derived in [66]:

$$\eta_P(x,\lambda)=\frac{\|P(\lambda)x\|_2}{\left(\sum_{\ell=0}^{k}|\lambda^\ell|\|A_\ell\|_2\right)\|x\|_2}. \tag{2.2}$$

If only the eigenvalue λ is of interest, we can always assume that the optimal eigenvector is available so that $\eta_P(\lambda,x)$ is minimal. Clearly, the $\|P(\lambda)x\|_2/\|x\|_2$ factor in (2.2) is minimized if x is the right singular vector that corresponds to the smallest singular value of $P(\lambda)$. Hence,

$$\eta_P(\lambda)\equiv\min_{x\neq\mathbf{0}}\frac{\|P(\lambda)x\|_2}{\left(\sum_{\ell=0}^{k}|\lambda^\ell|\|A_\ell\|_2\right)\|x\|_2}=\frac{\sigma_{\min}(P(\lambda))}{\left(\sum_{\ell=0}^{k}|\lambda^\ell|\|A_\ell\|_2\right)}=\frac{1}{\left(\sum_{\ell=0}^{k}|\lambda^\ell|\|A_\ell\|_2\right)\|P(\lambda)^{-1}\|_2},$$

where $\sigma_{\min}(\cdot)$ denotes the minimal singular value of a matrix. This trick of involving the singular vector of the smallest singular value is also at the core of the eigenvector refinement technique of Jia and Sun [45].

Remark 2.1. It is instructive to consider the special case $\lambda=0$. Obviously, if we set

$$\Delta A_\ell=\mathbf{0},\ \ell=0,\ldots,k;\ \ \Delta A_0=-A_0x\frac{x^*}{\|x\|_2^2}\ \left(\text{note that here }\frac{\|\Delta A_0\|_2}{\|A_0\|}=\frac{\|A_0x\|_2}{\|A_0\|\|x\|_2}\right), \tag{2.3}$$

then $(P(0)+\Delta P(0))x=(A_0+\Delta A_0)x=\mathbf{0}$. Recall that this ΔA_0 corresponds to the optimal backward error for $A_0x\approx\mathbf{0}$.

Remark 2.2. If the computed approximate eigenvalue is $\lambda=\infty$, then we can try to interpret it as a zero eigenvalue of a backward perturbed reversed problem. Using $P(\lambda)=\lambda^k\operatorname{rev}P(1/\lambda)$, $\mu=1/\lambda$, the expression (2.2) can be interpreted as

$$\eta_P(x,\lambda)=\frac{\|\lambda^k\sum_{\ell=0}^{k}(\lambda^{-\ell}A_{k-\ell})x\|_2}{|\lambda^k|(\sum_{\ell=0}^{k}|\lambda|^{-\ell}\|A_{k-\ell}\|_2)\|x\|_2}=\frac{\|\sum_{\ell=0}^{k}\mu^\ell A_{k-\ell}x\|_2}{\left(\sum_{\ell=0}^{k}|\mu^\ell|\|A_{k-\ell}\|_2\right)\|x\|_2}\equiv\eta_{\operatorname{rev}P}(x,\mu).$$

Hence, for infinite λ, the backward error can be defined analogously to (2.3) as $\Delta A_\ell = \mathbf{0}$, $\ell = 0,\ldots,k-1$, and $\Delta A_k = -A_k x x^*/\|x\|_2^2$. Clearly, $\|\Delta A_k\|_2/\|A_k\|_2 = \|A_k x\|_2/(\|A_k\|_2\|x\|_2)$, and $(A_k+\Delta A_k)x \equiv (\mathrm{rev}\,P(0)+\Delta\mathrm{rev}\,P(0))x = \mathbf{0}$.

2.2 On Hermitian and skew–Hermitian backward error

A backward error analysis is reassuring – it allows us to claim the computed result can be used with confidence because it corresponds almost to the given input data. However, in this interpretation of having solved a nearby problem, for many applications not only the size but also the structure of the backward perturbation matters. Suppose that the coefficient matrices A_ℓ are Hermitian (or real symmetric), where the symmetry is a result of the underlying physics of a concrete engineering application. In such cases non-hermitian/non-symmetric backward perturbed data $A_\ell + \Delta A_\ell$ make the interpretation of backward stability in terms of the original problem difficult.

Hence, it is of interest to determine the optimal backward error under the constraint that the backward errors in the coefficients A_ℓ are Hermitian:

$$\eta_P^{(H)}(x,\lambda) = \min\{\varepsilon \,:\, (P(\lambda)+\Delta P(\lambda))x = \mathbf{0},\ \Delta A_i^* = \Delta A_i,\ \|\Delta A_i\|_2 \le \varepsilon\|A_i\|_2,\ i=0,\ldots,k\}. \tag{2.4}$$

Note that in the definition (2.4) we do not require the coefficients A_i of $P(\lambda)$ to be Hermitian, although such Hermitian case is usually tacitly assumed if we are interested in $\eta_P^{(H)}(x,\lambda)$. The existence of optimal Hermitian backward error for Hermitian pencil, that matches the size of $\eta_P(x,\lambda)$, is established by Tisseur [66], but only for real eigenvalues.

Theorem 2.1 ([66])**.** *If all coefficient matrices of $P(\lambda)$ are Hermitian, and if λ is real, then $\eta_P(x,\lambda) = \eta_P^{(H)}(x,\lambda)$.*

In the next theorem, we extend the result of Tisseur to the entire finite spectrum, i.e. we now show that a Hermitian backward error is possible for any finite eigenvalue.

Theorem 2.2. *Let (x,λ) be an approximate eigenpair of $P(\lambda)$. Then $\eta_P(x,\lambda) = \eta_P^{(H)}(x,\lambda)$.*

Proof. Let $P(\lambda)x = r \neq \mathbf{0}$, and let $\lambda = \rho e^{i\varphi}$ be the polar form of λ. (For $\lambda = 0$, set $\rho = \varphi = 0$ and $\lambda^0 = 1$.) For $j = 0,\ldots,k$, we can construct Householder reflectors $H_j = H_j^* = H_j^{-1}$ such that

$$H_j x = -\frac{r}{\|r\|_2}\|x\|_2 e^{-ij\varphi}.$$

If we set $S_j = (\|r\|_2/\|x\|_2)H_j$, then $S_j^* = S_j$, $S_j x = -re^{-ij\varphi}$, and $\|S_j\|_2 = \|r\|_2/\|x\|_2$. Define backward errors

$$\Delta A_j = \frac{1}{\sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2}\|A_j\|_2 S_j \quad (\text{note that } \Delta A_j^* = \Delta A_j) \tag{2.5}$$

and check that

$$\lambda^j \Delta A_j x = \frac{-\lambda^j e^{-ij\varphi}\|A_j\|_2}{\sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2} r = \frac{-|\lambda|^j \|A_j\|_2}{\sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2} r,$$

and

$$\Delta P(\lambda)x = \sum_{j=0}^{k} \lambda^j \Delta A_j x = -r, \;\; (P(\lambda)+\Delta P(\lambda))x = \mathbf{0}.$$

Finally, note that the norm of ΔA_j matches the unconstrained optimal value, i.e.

$$\|\Delta A_j\|_2 = \frac{\|r\|_2}{\|x\|_2 \sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2} \|A_j\|_2 = \eta_P(x,\lambda)\|A_j\|_2.$$

□

The trick used in Theorem 2.2 can be slightly modified to analogously construct a skew–Hermitian perturbation.

Theorem 2.3. *Let* $\sigma = (\sigma_0, \ldots, \sigma_k) \in \{-1,1\}^{k+1}$ *and*

$$\eta_P^{(H,\sigma)}(x,\lambda) = \min\{\varepsilon \,:\, (P(\lambda)+\Delta P(\lambda))x = \mathbf{0},\ \Delta A_i^* = \sigma_i \Delta A_i,\ \|\Delta A_i\|_2 \le \varepsilon \|A_i\|_2,\ i = 0,\ldots,k\}, \tag{2.6}$$

i.e. the backward errors are required to be Hermitian or skew–Hermitian, as indicated in the prescribed signature $\sigma = (\pm 1, \ldots, \pm 1)$. *Then* $\eta_P^{(H,\sigma)}(x,\lambda) = \eta_P(x,\lambda)$.

Proof. Follow the proof of Theorem 2.2. For each $\sigma_j = 1$ define H_j as in (2.9) with $S_j = (\|r\|_2/\|x\|_2)H_j$. If $\sigma_j = -1$, define the Householder reflector H_j so that

$$H_j x = i\frac{r}{\|r\|_2}\|y\|_2 e^{-ij\varphi},$$

and set $S_j = i(\|r\|_2/\|x\|_2)H_j$. Then $S_j^* = -S_j$, $S_j x = -re^{-ij\varphi}$. If we define ΔA_j as in (2.5), then $\Delta A_j^* = \sigma_j \Delta A_j$ and the rest of the proof follows as in Theorem 2.2. □

2.2.1 The left eigenpair

If we have an approximate left eigenpair (y^*, λ) with finite λ, its backward error is defined analogously as

$$\eta_P(y^*,\lambda) = \min\{\varepsilon \,:\, y^*(P(\lambda)+\Delta P(\lambda)) = \mathbf{0},\ \|\Delta A_i\|_2 \le \varepsilon \|A_i\|_2,\ i = 0,\ldots,k\}, \tag{2.7}$$

and the corresponding explicit formula in terms of the residual reads

$$\eta_P(y^*,\lambda) = \frac{\|y^* P(\lambda)\|_2}{\left(\sum_{\ell=0}^k |\lambda^\ell| \|A_\ell\|_2\right)\|y\|_2}. \tag{2.8}$$

Of interest is, as discussed above, to determine minimal Hermitian backward error $\eta_P^{(H)}(y^*,\lambda)$, where

$$\eta_P^{(H)}(y^*,\lambda) = \min\{\varepsilon \,:\, y^*(P(\lambda)+\Delta P(\lambda)) = \mathbf{0},\ \Delta A_i^* = \Delta A_i,\ \|\Delta A_i\|_2 \le \varepsilon\|A_i\|_2,\ i=0,\ldots,k\}. \tag{2.9}$$

Clearly, we can use, *mutatis mutandis*, Theorems 2.2 and 2.3 to prove existence of the optimal Hermitian/skew-Hermitian backward error. For the sake of completeness, we provide the details.

Theorem 2.4. *Let (y^*,λ) be an approximate left eigenpair of $P(\lambda)$. Then $\eta_P^{(H)}(y^*,\lambda) = \eta_P(y^*,\lambda)$.*

Proof. Let $y^*P(\lambda) = r^* \neq \mathbf{0}$, and let $\lambda = \rho e^{i\varphi}$ be the polar form of λ. (For $\lambda = 0$, set $\rho = \varphi = 0$ and $\lambda^0 = 1$.) For $j = 0,\ldots,k$, we can construct Householder reflectors $H_j = H_j^* = H_j^{-1}$ such that

$$H_j y = -\frac{r}{\|r\|_2}\|y\|_2 e^{ij\varphi},$$

so that

$$y^*H_j = -\frac{r^*}{\|r\|_2}\|y\|_2 e^{-ij\varphi}.$$

If we set $S_j = (\|r\|_2/\|y\|_2)H_j$, then $S_j^* = S_j$, $y^*S_j = -r^*e^{-ij\varphi}$, and $\|S_j\|_2 = \|r\|_2/\|y\|_2$. Define backward errors

$$\Delta A_j = \frac{1}{\sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2}\|A_j\|_2 S_j \quad \text{(note that } \Delta A_j^* = \Delta A_j) \tag{2.10}$$

and check that

$$\lambda^j y^*\Delta A_j = \frac{-\lambda^j e^{ij\varphi}\|A_j\|_2}{\sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2} r^* = \frac{-|\lambda|^j\|A_j\|_2}{\sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2} r^*,$$

and

$$y^*\Delta P(\lambda) = \sum_{j=0}^k \lambda^j y^*\Delta A_j = -r^*,\ y^*(P(\lambda)+\Delta P(\lambda)) = \mathbf{0}.$$

Finally, note that the norm of ΔA_j matches the unconstrained optimal value, i.e.

$$\|\Delta A_j\|_2 = \frac{\|r\|_2}{\|y\|_2 \sum_{\ell=0}^k |\lambda|^\ell \|A_\ell\|_2}\|A_j\|_2 = \eta_P(y^*,\lambda)\|A_j\|_2.$$

□

Corollary 2.1. *Let $\sigma = (\sigma_0,\ldots,\sigma_k) \in \{-1,1\}^{k+1}$ and*

$$\eta_P^{(H,\sigma)}(y^*,\lambda) = \min\{\varepsilon \,:\, y^*(P(\lambda)+\Delta P(\lambda)) = \mathbf{0},\ \Delta A_i^* = \sigma_i \Delta A_i, \|\Delta A_i\|_2 \le \varepsilon\|A_i\|_2,\ i=0,\ldots,k\}, \tag{2.11}$$

i.e. the backward errors are required to be Hermitian or skew–Hermitian, as indicated in $\sigma = (\pm 1,\ldots,\pm 1)$. Then $\eta_P^{(H,\sigma)}(y^*,\lambda) = \eta_P(y^*,\lambda)$.

Proof. Follow the proof of Theorem 2.4. For each $\sigma_j = 1$ define H_j as in (2.2.1) with $S_j = (\|r\|_2/\|y\|_2)H_j$. If $\sigma_j = -1$, define the Householder reflector H_j so that

$$H_j y = -i\frac{r}{\|r\|_2}\|y\|_2 e^{ij\varphi},$$

so that

$$y^* H_j = i\frac{r^*}{\|r\|_2}\|y\|_2 e^{-ij\varphi},$$

and set $S_j = i(\|r\|_2/\|y\|_2)H_j$. Then $S_j^* = -S_j$, $y^*S_j = -r^*e^{-ij\varphi}$. If we define ΔA_j as in (2.10), then $\Delta A_j^* = \sigma_j \Delta A_j$ and the rest of the proof follows as in Theorem 2.4. □

Backward error for an approximate triple. The backward error for a triple (x, y^*, λ), computed by a numerical algorithm, is defined as

$$\eta(x,y^*,\lambda) = \min\{\varepsilon : (P(\lambda)+\Delta P(\lambda))x = \mathbf{0}, y^*(P(\lambda)+\Delta P(\lambda)) = \mathbf{0},\ \|A_\ell\|_2 \le \varepsilon\|A_\ell\|_2,\ \ell = 0,\ldots,k\}. \tag{2.12}$$

The explicit formula for (2.12) is given in the following theorem

Theorem 2.5 ([66])**.** *The normwise backward error for eigentriple is given by*

$$\eta(x,y^*,\lambda) = \frac{1}{\alpha}\max\left\{\frac{\|r\|_2}{\|x\|_2}, \frac{\|s\|_2}{\|y\|_2}\right\}, \tag{2.13}$$

where $r = P(\lambda)x$, $s^* = y^*P(\lambda)$ and $\alpha = \sum_{\ell=0}^{k}|\lambda|^\ell\|A_\ell\|_2$.

Notice that (2.13) actually says that $\eta(x,y^*,\lambda) = \max(\eta(x,\lambda),\eta(y^*,\lambda))$.

2.3 Backward error for a homogeneous form of $P(\lambda)$

As we emphasized in Section 1.4, the first step in most numerical methods for solving polynomial eigenvalue problems is linearization – the nonlinearity is traded for linear eigenvalue problem of higher dimension. Then, the next step is just direct deployment of the methods for the linear problem, and straightforward reconstruction of approximate eigenvalues and eigenvectors of the original nonlinear problem. In practice, it has been noticed that, although the backward error for a computed eigenpair for linear problem is small, the backward error of the corresponding approximation for the original polynomial problem can be much larger. It turns out that the relations between the norms of the coefficient matrices A_ℓ of $P(\lambda)$ affect the quality of the computed solution. This should be intuitively clear – if the norms $\|A_i\|_2$ vary widely over several orders of magnitude, and if some of those matrices are blocks in the coefficient matrix B of the linearization, then small $\|\delta B\|_2/\|B\|_2$ does not ensure small $\|\delta A_i\|_2/\|A_i\|_2$.

In some cases it is more convenient to define the backward errors for the homogeneous form of the matrix polynomial, where the homogeneous form is defined as

$$P(\alpha,\beta) = \sum_{\ell=0}^{k} \alpha^{\ell}\beta^{k-\ell}A_{\ell} \ (\equiv \beta^{k}\sum_{\ell=0}^{k}(\alpha/\beta)^{\ell}A_{\ell}).$$

The backward errors are then in forms of $\Delta P(\alpha,\beta)$ defined as

$$\eta_P(x,\alpha,\beta) = \min\{\varepsilon : (P(\alpha,\beta)+\Delta P(\alpha,\beta))x = \mathbf{0},\ \|\Delta A_i\|_2 \le \varepsilon\|A_i\|_2,\ i=0,\ldots,k\}, \quad (2.14)$$

$$\eta_P(y^*,\alpha,\beta) = \min\{\varepsilon : y^*(P(\alpha,\beta)+\Delta P(\alpha,\beta)) = \mathbf{0},\ \|\Delta A_i\|_2 \le \varepsilon\|A_i\|_2,\ i=0,\ldots,k\}. \quad (2.15)$$

An advantage of this representation of the backward error is that it uniformly applies to both finite and infinite eigenvalues. Using $P(\alpha,\beta) = \beta^k P(\alpha/\beta)$ for $\beta \neq 0$, and (2.2) and (2.8) we get explicit formulas for homogeneous form

$$\eta_P(x,\alpha,\beta) = \frac{\|P(\alpha,\beta)x\|_2}{(\sum_{i=0}^{k}|\alpha^i||\beta^{k-i}|\|A_i\|_2)\|x\|_2}, \quad (2.16)$$

$$\eta_P(y^*,\alpha,\beta) = \frac{\|y^*P(\alpha,\beta)\|_2}{(\sum_{i=0}^{k}|\alpha^i||\beta^{k-i}|\|A_i\|_2)\|y\|_2}. \quad (2.17)$$

Equivalent formulas for backward errors for the eigenpairs of a generalized (linear) eigenvalue problem in homogeneous form, $L(\alpha,\beta) = \beta X + \alpha Y$, are obtained by replacing $k=1$, $A_0 := X$ and $A_1 := Y$ in (2.16) and (2.17)

$$\eta_L(z,\alpha,\beta) = \frac{\|L(\alpha,\beta)z\|_2}{(|\alpha|\|X\|_2+|\beta|\|Y\|_2)\|z\|_2}, \quad (2.18)$$

$$\eta_L(u^*,\alpha,\beta) = \frac{\|u^*L(\alpha,\beta)\|_2}{(|\alpha|\|X\|_2+|\beta|\|Y\|_2)\|u\|_2}. \quad (2.19)$$

2.3.1 Backward error bounds for the homogeneous form

In [42], Higham, Li and Tisseur derived the bound for the backward error of an approximate eigenpair of $P(\lambda)$ in the terms of the backward error for the corresponding approximate eigenpair of L, from which is clear how the norms of the coefficient matrices affect the unevenness of the backward errors.

Let $L(\alpha,\beta)$ be a linearization of $P(\alpha,\beta)$, and let z be an approximate eigenvector for L and x an approximate eigenvector for P, both corresponding to the same eigenvalue (α,β). In order to compare $\eta_P(x,\alpha,\beta)$ and $\eta_L(z,\alpha,\beta)$, some well-defined relation between x and z is needed. The key assumption for deriving the backward error bounds is that there exists an $n \times kn$ matrix polynomial $G(\alpha,\beta)$ such that

$$G(\alpha,\beta)L(\alpha,\beta) = g^T \otimes P(\alpha,\beta), \quad (2.20)$$

for some nonzero $g \in \mathbb{C}^k$. Let $g^T = \begin{pmatrix} g_1 & \dots & g_k \end{pmatrix}$. Then we can write

$$g^T \otimes P(\alpha,\beta) = \begin{pmatrix} g_1 P(\alpha,\beta) & \dots & g_k P(\alpha,\beta) \end{pmatrix} = P(\alpha,\beta)(g^T \otimes \mathbb{I}_n).$$

Now, if z is an eigenvector of L then

$$G(\alpha,\beta)L(\alpha,\beta)z = P(\alpha,\beta)(g^T \otimes \mathbb{I}_n)z$$

implies that

$$x = (g^T \otimes \mathbb{I}_n)z \tag{2.21}$$

is an eigenvector of P. Now, if (2.20) is satisfied, and z is an approximate eigenvector of L, then x defined by (2.21), satisfies (see [42])

$$\begin{aligned} \eta_P(x,\alpha,\beta) &\leq \frac{\|G(\alpha,\beta)\|_2 \|L(\alpha,\beta)z\|_2}{(\sum_{j=0}^{k} |\alpha|^j |\beta|^{k-j} \|A_j\|_2)\|x\|_2} \\ &\leq \frac{|\alpha|\|X\|_2 + |\beta|\|Y\|_2}{\sum_{j=0}^{k} |\alpha|^j |\beta|^{k-j} \|A_j\|_2} \cdot \frac{\|G(\alpha,\beta)\|_2 \|z\|_2}{\|x\|_2} \cdot \eta_L(z,\alpha,\beta). \end{aligned} \tag{2.22}$$

Similarly, for a left eigenvector y^*, the assumption analogous to (2.20) requires existence of an $kn \times n$ matrix polynomial $H(\alpha,\beta)$ such that

$$L(\alpha,\beta)H(\alpha,\beta) = h \otimes P(\alpha,\beta), \tag{2.23}$$

for some nonzero $h \in \mathbb{C}^k$. The connection between the left eigenvectors u for L and y for P is then

$$y = (h^* \otimes I)u, \tag{2.24}$$

and the corresponding backward error is bounded by

$$\eta_P(y^*,\alpha,\beta) \leq \frac{|\alpha|\|X\|_2 + |\beta|\|Y\|_2}{\sum_{j=0}^{k} |\alpha|^j |\beta|^{k-j} \|A_j\|_2} \cdot \frac{\|H(\alpha,\beta)\|_2 \|u\|_2}{\|y\|_2} \cdot \eta_L(u^*,\alpha,\beta). \tag{2.25}$$

In the particular case of the first companion form ($L = C_1$), the ratio of the two backward errors can be bounded as shown in the following two theorems.

Theorem 2.6 ([42])**.** *Let z be an approximate right eigenvector of C_1, corresponding to the approximate eigenvalue (α,β). Then for $z_k = z((k-1)n+1:kn), k = 1,\dots,k$, we have*

$$\frac{1}{k^{1/2}} \leq \frac{\eta_P(z_k,\alpha,\beta)}{\eta_{C_1}(z,\alpha,\beta)} \leq k^{5/2} \frac{\max(1,\max_i \|A_i\|_2)^2}{\min(\|A_0\|_2,\|A_k\|_2)} \frac{\|z\|_2}{\|z_k\|_2}. \tag{2.26}$$

Theorem 2.7 ([42])**.** *Let u be an approximate left eigenvector of C_1 corresponding to the ap-*

proximate eigenvalue (α,β). *Then for* $u_1 = u(1:n)$, *we have*

$$\frac{1}{k^{1/2}} \leq \frac{\eta_P(u_1^*,\alpha,\beta)}{\eta_{C_1}(u,\alpha,\beta)} \leq k^{3/2}\frac{\max(1,\max_i \|A_i\|_2)}{\min(\|A_0\|_2,\|A_k\|_2)}\frac{\|u\|_2}{\|u_1\|_2}. \tag{2.27}$$

Since $C_2(P) = C_1(P^T)^T$, we can conclude that these bounds apply to the second companion form as well, but so that (2.26) applies to a left eigenpair, and (2.27) holds for the corresponding right eigenpair.

From both of these theorems we see that the backward errors of the initial nonlinear problem and its linearization differ only by a modest factor of the degree k, provided that the norms of the coefficient matrices A_i are close to one. To illustrate how unbalanced $\|A_i\|_2$'s influence the ratio between the two kinds of backward errors we present the following example.

Example 2.1. We consider the `power_plant` example from the NLEVP benchmark library [5]. It is a QEP $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$ of order 8, representing a reduced order model of dynamic behaviour of a nuclear power plant. The norms of the coefficient matrices are:

$$\begin{aligned} M &= \texttt{235000000}, \\ C &= \texttt{4.350043895953605e+010}, \\ K &= \texttt{1.692005328941397e+013}. \end{aligned}$$

The backward errors for the eigenvalue problem for the linearization

$$A - \lambda B = \begin{pmatrix} C & -\mathbb{I} \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I} \end{pmatrix}, \tag{2.28}$$

and for the original problem are shown in Figure 2.2.

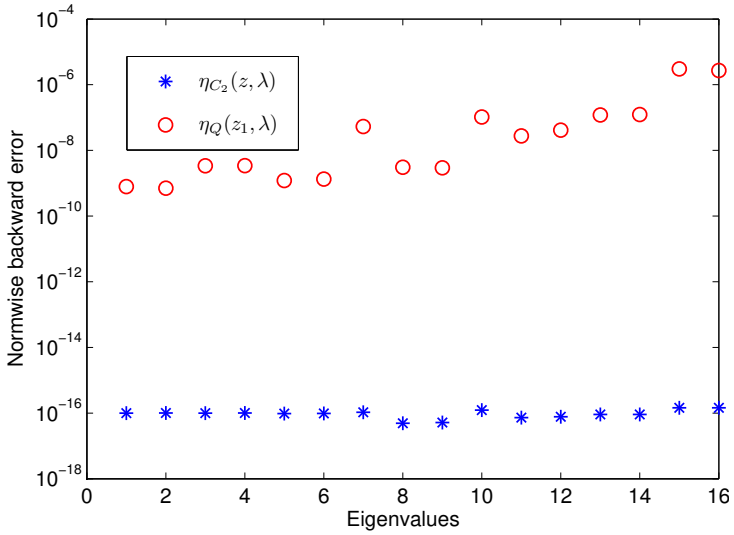


Figure 2.2: Backward errors for the eigenvalue problem of the linearization (2.28) of the test example `power_plant`, and for the original problem $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$.

It is clear from Figure 2.2 that there is a substantial gap between the backward error for the original quadratic problem and the error of the corresponding second companion form linearization. It is instructive to compare the gap between the two errors and the ratios of the norms of the coefficient matrices in the quadratic problem.

2.3.2 Parameter scaling

In order to solve the problem of non equilibrated norms of the coefficient matrices of matrix polynomial $P(\lambda)$, the parameter scaling is proposed by several authors, see e.g. [30],[31],[37]. The idea is to use two new parameters γ and δ to change the variables and define a new polynomial matrix $\widetilde{P}(\mu) = \sum_{\ell=0}^{k} \mu^{\ell} \widetilde{A}_{\ell}$ as

$$\lambda = \gamma\mu, \;\; \widetilde{P}(\mu) := P(\lambda)\delta = \mu^k \underbrace{(\gamma^k \delta A_k)}_{=:\widetilde{A}_k} + \mu^{k-1} \underbrace{(\gamma^{k-1} \delta A_{k-1})}_{=:\widetilde{A}_{k-1}} + \ldots + \underbrace{(\delta A_0)}_{=:\widetilde{A}_0}. \tag{2.29}$$

The free parameters γ and δ are then determined so that the ratio

$$\frac{\max(1, \max_i \|A_i\|_2)^2}{\min(\|A_0\|_2, \|A_k\|_2)}, \tag{2.30}$$

from the bounds (2.26) and (2.27) is as small as possible. Betcke proved in [4] that the optimal γ for minimizing

$$\rho(\gamma) := \frac{\max_i \gamma^i \|A_i\|_2}{\min(\|A_0\|_2, \gamma^k \|A_k\|_2)} \tag{2.31}$$

is

$$\gamma = \left(\frac{\|A_0\|_2}{\|A_k\|_2} \right)^{1/k}. \tag{2.32}$$

δ is then defined so that the norms of scaled matrices are close to 1. Fan, Lin and Van Dooren derived the parameters for quadratic eigenvalue problem in [30]. This type of scaling is used in `quadeig` algorithm for computing all eigenvalues and eigenvectors of quadratic eigenvalue problem [37]. The parameters will be presented in Subsection 3.3.1.

Finally, Gaubert and Sharify [31] proposed scaling using the tropical roots. Tropical algebra is relatively new and rarely present in the research in numerical linear algebra. For that reason, we briefly review the elementary notions from tropical algebra, that will be needed in the rest of the thesis.

Tropical scaling. The tropical algebra, or max–plus algebra is a semiring $(\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$ with operations defined as follows

$$x \oplus y = \max(x, y),$$

$$x \otimes y = x + y.$$

The zero element $\mathbb{0}$ of the tropical semiring, for which $\mathbb{0} \otimes a = a$ holds, is $\mathbb{0} = -\infty$. The unit element of the tropical semiring $\mathbb{1}$, for which $\mathbb{1} \otimes a = a \otimes \mathbb{1} = a$ holds, is $\mathbb{1} = 0$.
The max–times semiring is another variant of the tropical semiring. It is a set of nonnegative real numbers $\mathbb{R}^+$ equipped with the max operation as addition and the usual multiplication as multiplication. The tropical polynomial in max–times algebra is $t_\times p(x) = \max_{0 \le k \le n} a_k x^k$. The max–times and max–plus semirings are isomorphic by the map $x \mapsto \log x$.
The tropical polynomial tp of degree n, written as

$$\mathsf{tp} = \bigoplus_{\mathsf{k}=0}^{\mathsf{n}} \mathsf{a_k} \otimes \mathsf{x}^{\otimes \mathsf{k}}, \ \mathsf{a_k} \in \mathbb{R} \cup \{-\infty\}, \tag{2.33}$$

corresponds to $p(x) = \max_{0 \le k \le n}(a_k + kx)$ in the classical algebra. The finite tropical roots of the polynomial (2.33) are defined as the points at which the maximum $\max_{0 \le k \le n}(a_k + kx)$ is attained at least twice. There are n tropical roots, counting the multiplicities for the tropical polynomial of degree n. The analogue of the fundamental theorem of algebra for the tropical polynomials is that $p(x)$ can be uniquely written as $p(x) = a_n + \sum_{k=1}^{n} \max(x, c_k)$, where $c_1, \ldots, c_n \in \mathbb{R} \cup \{-\infty\}$ are the tropical roots. They are computed using the Newton polygons.

For tropical polynomial (2.33) we define the corresponding Newton polygon as the upper boundary of the convex hull of the set of points (k, a_k), $k = 1, \ldots, n$. It consists of a number of linear segments. Now, the roots are the opposites of the slopes of these segments, and the multiplicities are the width of the segments, that is the difference of the abscissae of its endpoints. Let $k_0 = 0 < \ldots < k_q = n$ be the abscissae of the vertices of the Newton polygon. Then (2.33) has q distinct roots

$$\alpha_j = -\frac{a_{k_j} - a_{k_{j-1}}}{k_j - k_{j-1}}, \ \ j = 1, \ldots, q, \tag{2.34}$$

with multiplicities $m_j = k_j - k_{j-1}, \ \ j = 1, \ldots, q$, respectively.
On the other hand, the tropical roots of tropical polynomial $t_\times p(x)$ in max–times semiring are the exponentials of the tropical roots of the max–plus polynomial $\mathsf{tp}(\mathsf{x}) = \max_{0 \le \mathsf{k} \le \mathsf{n}}(\log \mathsf{a_k} + \mathsf{kx})$

$$\gamma_j = \left(\frac{a_{k_{j-1}}}{a_{k_j}} \right)^{1/(k_j - k_{j-1})}, \tag{2.35}$$

and the multiplicites $m_j = k_j - k_{j-1}$ are the same.
The tropical roots can be computed in linear time, as shown in [31].

Example 2.2 ([54])**.** Consider the tropical polynomial

$$t_\times p(x) = \max(\underbrace{\texttt{7.5e-5}}_{=a_0}, \underbrace{\texttt{8.9e+2}}_{=a_1} x, \underbrace{\texttt{8.6e+2}}_{=a_2} x^2, \underbrace{\texttt{8.8e+8}}_{=a_3} x^3, \underbrace{\texttt{7.7e+7}}_{=a_4} x^4). \tag{2.36}$$

The Newton polygon corresponding to a max–plus tropical polynomial $\mathsf{tp}(\mathsf{x}) = \bigoplus_{\ell=0}^{4} \mathsf{a}_\ell \otimes \mathsf{x}^{\otimes \ell}$ is presented in Figure 2.3.

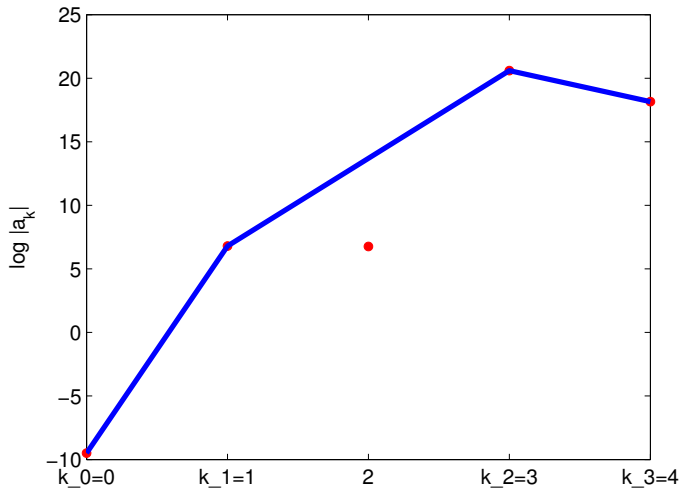


Figure 2.3: Newton polygon corresponding to $\mathsf{tp}(\mathsf{x})$

Now, the tropical roots, and their multiplicities, of (2.36) are

$$\gamma_1 = \left(\frac{a_0}{a_1}\right)^{1/(k_1-k_0)} = 8.426966292134831\text{e-}008,\ m_1 = 1,$$
$$\gamma_2 = \left(\frac{a_1}{a_3}\right)^{1/(k_3-k_1)} = 1.005665767719890\text{e-}003,\ m_2 = 2,$$
$$\gamma_3 = \left(\frac{a_3}{a_4}\right)^{1/(k_4-k_3)} = 1.142857142857143\text{e+}001,\ m_3 = 1.$$

For verification, let us compute $\mathsf{tp}(\gamma_i)$, $i = 1,2,3$:

$$\mathsf{tp}(\gamma_1) = \max(7.5000\text{e-}005, 7.5000\text{e-}005, 6.1072\text{e-}012, 5.2662\text{e-}013, 3.8831\text{e-}021),$$
$$\mathsf{tp}(\gamma_2) = \max(7.5000\text{e-}005, 8.9504\text{e-}001, 8.6977\text{e-}004, 8.9504\text{e-}001, 7.8760\text{e-}005),$$
$$\mathsf{tp}(\gamma_3) = \max(7.5000\text{e-}005, 1.0171\text{e+}004, 1.1233\text{e+}005, 1.3136\text{e+}012, 1.3136\text{e+}012).$$

We can see that the maximum is attained twice for every γ_i, $i = 1,2,3$, as it is required by the definition of the tropical roots.

For our purpose of scaling a matrix polynomial $\sum_{\ell=0}^{k} \lambda^\ell A_\ell$, define the tropical polynomial

$$\mathsf{tp}(\mathsf{x}) = \bigoplus_{\ell=0}^{\mathsf{k}} \|\mathsf{A}_\ell\|_2 \otimes \mathsf{x}^{\otimes \ell}, \tag{2.37}$$

where A_ℓ are the coefficients of the matrix polynomial.
The tropical roots of (2.37) are used for scaling of the polynomial eigenvalue problem in order to

improve backward error for the eigenpairs computed using the linearization. Let $\widetilde{P}(\mu) = P(\lambda)\delta$ be the scaled polynomial, where $\lambda = \gamma\mu$. Let $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_k$ be the tropical roots of $\mathtt{tp}(\cdot)$, counted with multiplicities. The scaling parameters are defined as

$$\gamma_i = \alpha_i,\ \delta_i = (\mathtt{tp}(\alpha_\mathsf{i}))^{-1}, \mathsf{i} = 1, \ldots, \mathsf{k}. \tag{2.38}$$

Notice that there are as many distinct scaling parameters as the number of distinct tropical roots of the polynomial $\mathtt{tp}(\cdot)$. The small backward error is expected only for those eigenvalues that are close to some root α_i. This is why [31] proposes the following procedure:

- Define the tropical polynomial $\mathtt{tp}$
- Find the k tropical roots $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_k$ counting the multiplicities
- For each distinct tropical root α_i define the corresponding tropical scaling (2.38). Compute the eigenvalues of the scaled problem (2.29) by using the QZ algorithm for the corresponding linearization. Sort the computed eigenvalues by the magnitude $\lambda_1, \ldots, \lambda_{kn}$, and divide them into k groups of n elements. The ith group would be $\lambda_{(i-1)n+1}, \ldots, \lambda_{in}$. For each α_i choose ith group of the eigenvalues as the approximation.

2.4 Componentwise backward error

The componentwise backward error for a matrix polynomial $P(\lambda)$ and its approximate eigenpair (x, λ), with λ finite is defined by

$$\omega_P(x, \lambda) = \min\{\varepsilon \ :\ (P(\lambda) + \Delta P(\lambda))x = \mathbf{0},\ |\Delta A_i| \leq \varepsilon |A_i|,\ i = 0, \ldots, k\}, \tag{2.39}$$

where $\Delta P(\lambda) = \sum_{\ell=0}^{k} \lambda^\ell \Delta A_\ell$ is, as before, the backward error in $P(\lambda)$. An explicit formula for component-wise backward error for the generalized eigenvalue problem $Ax = \lambda x$ and the corresponding approximate eigenpair (x, λ) is derived in [39] as

$$\omega_L(x, \lambda) = \max_i \frac{|r_i|}{((|A| + |\lambda||B|)|x|)_i}, \tag{2.40}$$

where $r = Ax - \lambda Bx$, and $\xi/0$ is interpreted as zero if $\xi = 0$, and infinity otherwise. In the following theorem we derive explicit formula for (2.39) for the quadratic eigenvalue problem $Q(\lambda) = \lambda^2 M + \lambda C + K$ using the component-wise backward error for the corresponding first companion form linearization. We provide a different proof, using the linearization of the quadratic problem and the corresponding explicit formula for component-wise backward error of generalized eigenvalue problem (2.40).

Theorem 2.8. *The componentwise backward error for the quadratic matrix polynomial $Q(\lambda)$, corresponding to an approximate eigenpair (x,λ) is given by*

$$\omega_Q(x,\lambda) = \max_i \frac{|r_i|}{((|\lambda|^2|M| + |\lambda||C| + |K|)|x|)_i}, \tag{2.41}$$

where $r = (\lambda^2 M + \lambda C + K)x$, and $\xi/0$ is interpreted as zero if $\xi = 0$, and infinity otherwise.

Proof. Let (x,λ) be an approximate eigenpair for $Q(\lambda)$. Then $\left(\begin{pmatrix}\lambda x\\ x\end{pmatrix},\lambda\right)$ is an approximate eigenpair for the corresponding first companion form linearization

$$\begin{aligned}(A-\lambda B)\begin{pmatrix}\lambda x\\ x\end{pmatrix} &= \left\{\begin{pmatrix}C & K\\ -\mathbb{I} & \mathbf{0}\end{pmatrix} - \lambda\begin{pmatrix}-M & \mathbf{0}\\ \mathbf{0} & -\mathbb{I}\end{pmatrix}\right\}\begin{pmatrix}\lambda x\\ x\end{pmatrix}\\ &= \begin{pmatrix}(\lambda^2 M + \lambda C + K)x\\ \mathbf{0}\end{pmatrix} = \begin{pmatrix}r\\ \mathbf{0}\end{pmatrix} = r_L.\end{aligned} \tag{2.42}$$

(2.41) applied on equation (2.42) implies that there exists ΔA and ΔB so that $(A+\Delta A - \lambda(B+\Delta B))\begin{pmatrix}\lambda x\\ x\end{pmatrix} = \mathbf{0}$, and $|\Delta A| \le \varepsilon|A|$, $|\Delta B| \le \varepsilon|B|$, with $\varepsilon = \omega_L\left(\begin{pmatrix}\lambda x\\ x\end{pmatrix},\lambda\right)$. Since this bound is component-wise, we conclude that there exist ΔM, ΔC, ΔK, E_1, E_2 so that

$$\left\{\begin{pmatrix}C+\Delta C & K+\Delta K\\ -(\mathbb{I}+E_1) & 0\end{pmatrix} - \lambda\begin{pmatrix}-(M+\Delta M) & \mathbf{0}\\ \mathbf{0} & -(\mathbb{I}+E_2)\end{pmatrix}\right\}\begin{pmatrix}\lambda x\\ x\end{pmatrix} = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \tag{2.43}$$

and $|\Delta M| \le \varepsilon|M|$, $|\Delta C| \le \varepsilon|C|$, $|\Delta K| \le \varepsilon|K|$, $|E_1| \le \varepsilon|I|$, $|E_2| \le \varepsilon|I|$ (notice that E_1 and E_2 are diagonal matrices.) By equating the corresponding block rows on the left and right side of the equation (2.43) we get

$$(\lambda^2(M+\Delta M) + \lambda(C+\Delta C) + (K+\Delta K))x = \mathbf{0}, \tag{2.44}$$

$$-\lambda E_1 x + \lambda E_2 x = \mathbf{0}. \tag{2.45}$$

Since E_1 and E_2 are diagonal, (2.45) reads $(E_1)_{ii}x_i = (E_2)_{ii}x_i$. Now, if $x_i \neq 0$ $(E_1)_{ii} = (E_2)_{ii}$. Otherwise, any $(E_1)_{ii},(E_2)_{ii}$ such that $|(E_1)_{ii}|,|(E_2)_{ii}| \le \varepsilon$ satisfies the equation, so we take $(E_1)_{ii} = (E_2)_{ii}$. From this reasoning we conclude that $E_1 = E_2$. Finally, by multiplying the equation (2.43) with $\begin{pmatrix}\mathbb{I} & \mathbf{0}\\ \mathbf{0} & (\mathbb{I}+E_1)^{-1}\end{pmatrix}$ from the left we derive

$$\left\{\begin{pmatrix}C+\Delta C & K+\Delta K\\ -\mathbb{I} & \mathbf{0}\end{pmatrix} - \lambda\begin{pmatrix}-(M+\Delta M) & \mathbf{0}\\ \mathbf{0} & -\mathbb{I}\end{pmatrix}\right\}\begin{pmatrix}\lambda x\\ x\end{pmatrix} = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}. \tag{2.46}$$

Moreover, notice that $|r_L| = \begin{pmatrix}|r|\\ \mathbf{0}\end{pmatrix}$ and that

$$(|A| + |\lambda||B|)\begin{pmatrix}|\lambda||x|\\ |x|\end{pmatrix} = \begin{pmatrix}|\lambda|^2|M||x| + |\lambda||C||x| + |K||x|\\ 2|\lambda||x|\end{pmatrix}.$$

Finally, we express ω_L in terms of the original data as

$$\omega_L\left(\begin{pmatrix}\lambda x\\ x\end{pmatrix},\lambda\right) = \max_i \frac{|(r_L)_i|}{\left((|A|+|\lambda||B|)\begin{pmatrix}|\lambda||x|\\ |x|\end{pmatrix}\right)_i} = \max_i \frac{\begin{pmatrix}|r|\\ \mathbf{0}\end{pmatrix}_i}{\begin{pmatrix}|\lambda|^2|M||x|+|\lambda||C||x|+|K||x|\\ 2|\lambda||x|\end{pmatrix}_i}$$
$$= \max_i \frac{|r_i|}{((|\lambda|^2|M|+|\lambda||C|+|K|)|x|)_i}.$$

Hence (2.41) holds. □

Another, more intrinsic, proof is be to directly define ΔM, ΔC, ΔK analogously to proof for the linear matrix pencil in [39]. Since the construction of the backward error directly in terms of the original problem is more insightful, we provide the details.

A more intrinsic proof of Theorem 2.8. Let $\widetilde{\omega}$ be the minimal ε such that $|\Delta M| \le \varepsilon|M|$, $|\Delta C| \le \varepsilon|C|$, $|\Delta K| \le \varepsilon|K|$, and $(\lambda^2(M+\Delta M)+\lambda(C+\Delta C)+(K+\Delta K))x = \mathbf{0}$. If $r = (\lambda^2 M+\lambda C+K)x$, then

$$|r| = |-(\lambda^2\Delta M+\lambda\Delta C+\Delta K)x| \le \varepsilon(|\lambda|^2|M|+|\lambda||C|+|K|)|x|,$$

that is, $\widetilde{\omega} \le \omega_Q(x,\lambda)$. On the other hand, this bound is attainable by the following perturbations

$$\Delta M = -\operatorname{sign}(\lambda^2)D_1|M|D_2,\ \ \Delta C = -\operatorname{sign}(\lambda)D_1|C|D_2,\ \ \Delta K = -D_1|K|D_2,$$

where

$$D_1 = \operatorname{diag}\left(\frac{r_i}{((|\lambda|^2|M|+|\lambda||C|+|K|)|x|)_i}\right),\ \ D_2 = \operatorname{diag}(\operatorname{sign}(x_i)).$$

To see this, check that

$$\begin{aligned}\Delta Q(\lambda)x =& -\lambda^2\operatorname{sign}(\lambda^2)\operatorname{diag}\left(\frac{r_i}{|\alpha|_i}\right)|M|\operatorname{diag}(\operatorname{sign}(x_i))x\\ &-\lambda\operatorname{sign}(\lambda)\operatorname{diag}\left(\frac{r_i}{|\alpha|_i}\right)|C|\operatorname{diag}(\operatorname{sign}(x_i))x-\operatorname{diag}\left(\frac{r_i}{|\alpha|_i}\right)|K|\operatorname{diag}(\operatorname{sign}(x_i))x\\ =& -r,\end{aligned}$$

and

$$|\Delta M| = \operatorname{diag}\frac{|r_i|}{|\alpha|_i}|M| = \omega_Q(x,\lambda)|M|,$$
$$|\Delta C| = \operatorname{diag}\frac{|r_i|}{|\alpha|_i}|C| = \omega_Q(x,\lambda)|C|,$$
$$|\Delta K| = \operatorname{diag}\frac{|r_i|}{|\alpha|_i}|K| = \omega_Q(x,\lambda)|K|,$$

where $|\alpha| = (\sum_{\ell=0}^{k} |\lambda|^{\ell} |A_\ell|)|x|$. □

The following example demonstrates the difference between the normwise and the componentwise backward error. There can be a gap between these errors suggesting that the computed eigenpair is not as good as we could conclude by just looking at the normwise backward error.

Example 2.3. Consider the `speaker_box` example from the NLEVP library. We computed all 214 eigenvalues and corresponding right eigenvectors using the algorithm `quadeig` which will be explained in Chapter 3. The normwise and the componentwise backward errors for all right eigenpairs are presented in the following figure

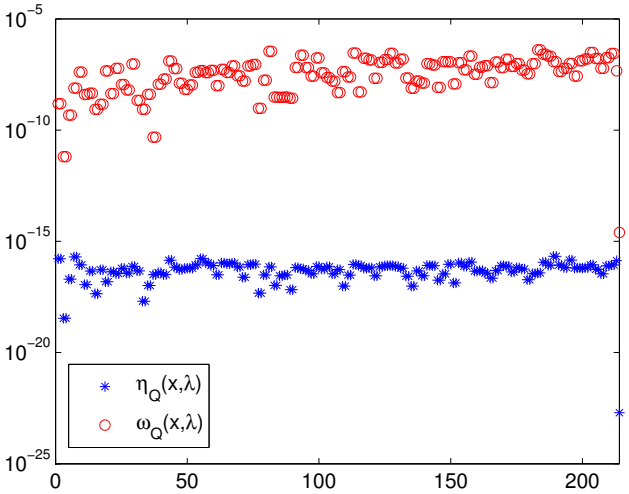


Figure 2.4: `speaker_box`, normwise and componentwise backward errors for all right eigenpairs

In order to prove the analogous theorem for a left eigenpair, we have to use the second companion form linearization.

Theorem 2.9. *The componentwise backward error for quadratic matrix polynomial $Q(\lambda)$ for approximate left eigenpair (y^*, λ) is given by*

$$\omega_Q(y^*, \lambda) = \max_i \frac{|r_i^*|}{(|y^*|(|\lambda|^2|M| + |\lambda||C| + |K|))_i}, \tag{2.47}$$

where $r^* = y^*(\lambda^2 M + \lambda C + K)$, and $\xi/0$ is interpreted as zero if $\xi = 0$, and infinity otherwise.

Proof. Let (y^*, λ) be an approximate left eigenpair for $Q(\lambda)$. Then $(\begin{pmatrix} \lambda y^* & y^* \end{pmatrix}, \lambda)$ is an approximate left eigenpair for the corresponding second companion form linearization

$$\begin{aligned} \begin{pmatrix} \lambda y^* & y^* \end{pmatrix} (A - \lambda B) &= \begin{pmatrix} \lambda y^* & y^* \end{pmatrix} \left\{ \begin{pmatrix} C & -\mathbb{I} \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I} \end{pmatrix} \right\} \\ &= \begin{pmatrix} y^*(\lambda^2 M + \lambda C + K) & \mathbf{0} \end{pmatrix} = \begin{pmatrix} r^* & \mathbf{0} \end{pmatrix} = r_L^*. \end{aligned} \tag{2.48}$$

(2.47) applied on (2.48) implies that there exists ΔA and ΔB so that $\begin{pmatrix}\lambda y^* & y^*\end{pmatrix}(A+\Delta A-\lambda(B+\Delta B))=0$, and $|\Delta A|\leq\varepsilon|A|, |\Delta B|\leq\varepsilon|B|$, with $\varepsilon=\omega_L\left(\begin{pmatrix}\lambda y^* & y^*\end{pmatrix},\lambda\right)$. Since this bound is componentwise, we conclude that there exist ΔM, ΔC, ΔK, E_1, E_2 so that

$$\begin{pmatrix}\lambda y^* & y^*\end{pmatrix}\left\{\begin{pmatrix}C+\Delta C & -(\mathbb{I}+E_1)\\ K+\Delta K & \mathbf{0}\end{pmatrix}-\lambda\begin{pmatrix}-(M+\Delta M) & \mathbf{0}\\ \mathbf{0} & -(\mathbb{I}+E_2)\end{pmatrix}\right\}=\begin{pmatrix}\mathbf{0} & \mathbf{0}\end{pmatrix}, \quad (2.49)$$

and $|\Delta M|\leq\varepsilon|M|$, $|\Delta C|\leq\varepsilon|C|$, $|\Delta K|\leq\varepsilon|K|$, $|E_1|\leq\varepsilon|I|$, $|E_2|\leq\varepsilon|I|$ (notice that E_1 and E_2 are diagonal matrices.) By equating the corresponding block rows on the left and the right side of the equation (2.49) we get

$$y^*(\lambda^2(M+\Delta M)+\lambda(C+\Delta C)+(K+\Delta K))=\mathbf{0}, \quad (2.50)$$

$$-\lambda y^*E_1+\lambda y^*E_2=\mathbf{0}. \quad (2.51)$$

Since E_1 and E_2 are diagonal, (2.51) reads $(E_1)_{ii}y_i=(E_2)_{ii}y_i$. Now, if $y_i\neq 0$ $(E_1)_{ii}=(E_2)_{ii}$. Otherwise, any $(E_1)_{ii},(E_2)_{ii}$ such that $|(E_1)_{ii}|,|(E_2)_{ii}|\leq\varepsilon$ satisfies the equation, so we take $(E_1)_{ii}=(E_2)_{ii}$. Form this reasoning we conclude that $E_1=E_2$. Finally, by multiplying the equation (2.49) with $\begin{pmatrix}\mathbb{I} & \mathbf{0}\\ \mathbf{0} & (I+E_1)^{-1}\end{pmatrix}$ from the right we derive

$$\begin{pmatrix}\lambda y^* & y^*\end{pmatrix}\left\{\begin{pmatrix}C+\Delta C & -\mathbb{I}\\ K+\Delta K & \mathbf{0}\end{pmatrix}-\lambda\begin{pmatrix}-(M+\Delta M) & \mathbf{0}\\ \mathbf{0} & -\mathbb{I}\end{pmatrix}\right\}=\begin{pmatrix}\mathbf{0} & \mathbf{0}\end{pmatrix}. \quad (2.52)$$

Moreover, notice that $|r_L^*|=\begin{pmatrix}|r^*| & \mathbf{0}\end{pmatrix}$ and

$$\begin{pmatrix}|\lambda||y^*| & |y^*|\end{pmatrix}(|A|+|\lambda||B|)=\begin{pmatrix}|\lambda|^2|y^*||M|+|\lambda||y^*||C|+|y^*||K| & 2|\lambda||y^*|\end{pmatrix}.$$

Finally, putting all together, we obtain

$$\begin{aligned}\omega_L\left(\begin{pmatrix}\lambda y^* & y^*\end{pmatrix},\lambda\right)&=\max_i\frac{|(r_L^*)_i|}{\left(\begin{pmatrix}|\lambda||y^*| & |y^*|\end{pmatrix}(|A|+|\lambda||B|)\right)_i}\\ &=\max_i\frac{\begin{pmatrix}|r^*| & \mathbf{0}\end{pmatrix}_i}{\begin{pmatrix}|\lambda|^2|y^*||M|+|\lambda||y^*||C|+|y^*||K| & 2|\lambda||y^*|\end{pmatrix}_i}\\ &=\max_i\frac{|r_i|}{(|y^*|(|\lambda|^2|M|+|\lambda||C|+|K|))_i}.\end{aligned}$$

Hence (2.47) holds. □

Theorem 2.8 and 2.9 can be generalized for arbitrary polynomial eigenvalue problem of order k. The only difference in the proof of the theorem is that the linerization will be the pencil

of order kn, and there will additional perturbations $E_{i,j}$, $i = 1,\ldots,k-1$, $j = 1,2$ on identity matrices

$$(A+\Delta A) = \begin{pmatrix} A_{k-1}+\Delta A_{k-1} & A_{k-2}+\Delta A_{k-2} & \ldots & A_0+\Delta A_0 \\ -(\mathbb{I}+E_{1,1}) & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ldots & -(\mathbb{I}+E_{k-1,1}) & \mathbf{0} \end{pmatrix}, \tag{2.53}$$

$$(B+\Delta B) = \operatorname{diag}(A_k+\Delta A_k, -(\mathbb{I}+E_{1,2}),\ldots,-(\mathbb{I}+E_{k-1,2})). \tag{2.54}$$

By the same reasoning as in the proof of Theorem 2.8, we can conclude that $E_{i,1} = E_{i,2}, i = 1,\ldots,k-1$. The rest of the proof is analogous.
Similarly, for the left eigenpair we will have

$$(A+\Delta A) = \begin{pmatrix} A_{k-1}+\Delta A_{k-1} & -(\mathbb{I}+E_{1,1}) & \ldots & \mathbf{0} \\ A_{k-2}+\Delta A_{k-2} & \mathbf{0} & \ldots & \vdots \\ \vdots & \ddots & \ddots & -(\mathbb{I}+E_{k-1,1}) \\ A_0+\Delta A_0 & \mathbf{0} & \ldots & \mathbf{0} \end{pmatrix}, \tag{2.55}$$

$$(B+\Delta B) = \operatorname{diag}(A_k+\Delta A_k, -(\mathbb{I}+E_{1,2}),\ldots,-(\mathbb{I}+E_{k-1,2})). \tag{2.56}$$

Here, we state the theorem for the sake of completeness

Theorem 2.10. *For the matrix polynomial $P(\lambda)$ of order k, the component-wise backward error for an approximate eigenpair (x,λ) is given by*

$$\omega_P(x,\lambda) = \max_i \frac{|r_i|}{\left((\sum_{\ell=0}^k |\lambda^\ell||A_\ell|)|x|\right)_i}, \tag{2.57}$$

where $r = (\sum_{\ell=0}^k \lambda^\ell A_\ell)x$, and $\xi/0$ is interpreted as zero if $\xi = 0$, and infinity otherwise.

Theorem 2.11. *For the matrix polynomial $P(\lambda)$, the component-wise backward error for an approximate left eigenpair (y^*,λ) is given by*

$$\omega_P(y^*,\lambda) = \max_i \frac{|r_i^*|}{\left(|y^*|(\sum_{\ell=0}^k |\lambda^\ell||A_\ell|)\right)_i}, \tag{2.58}$$

where $r^* = y^*(\sum_{\ell=0}^k \lambda^\ell A_\ell)$, and $\xi/0$ is interpreted as zero if $\xi = 0$, and infinity otherwise.

Chapter 3

Complete solution of the quadratic eigenvalue problem

In this chapter we study numerical methods for computing all eigenvalues with the corresponding eigenvectors of the $n \times n$ quadratic eigenvalue problem

$$Q(\lambda)x = (\lambda^2 M + \lambda C + K)x = \mathbf{0}. \tag{3.1}$$

This problem is at the kernel even of the methods for computing only selected eigenpairs of a large scale problem; in such cases $Q(\lambda)$ is restricted/projected on a small dimensional subspace (constructed by some algorithm) and full solution of the projected problem is required to advance an iterative method and/or to construct an approximate solution for the original problem.

The core of the state of the art methods is computation of the eigenvalues and eigenvectors of a particularly chosen linearization. The linearized problem is usually solved with the QZ method. In some cases, this may lead to difficulties, in particular if M is exactly or nearly rank deficient, which leads to (numerically) infinite eigenvalues. Even if QZ is not too much troubled by the presence of the infinite eigenvalues [72], it would be advantageous to deflate them early in the computational scheme. Similarly, if K is rank deficient, then its null space provides eigenvectors for the eigenvalue $\lambda = 0$ and removing it in a preprocessing phase facilitates more efficient computation of the remaining eigenvalues. In both cases a nontrivial decision about the numerical rank has to be made.

These issues have been addressed by Hammarling, Munro and Tisseur [37] who used the structure of the linearization pencil (3.2) to deflate certain number of zero and infinite eigenvalues using the rank revealing decompositions of the coefficient matrices M and K of the original quadratic eigenvalue problem (3.1). The resulting algorithm, designated as `quadeig` is shown to be more robust as e.g. the `polyeig()` function used in Matlab.

In this chapter we propose a new algorithm, following the philosophy of `quadeig`, but with more attention to fine numerical details that ensure numerically more robust and reliable computational procedure. Our supporting numerical analysis and numerical evidence indicate

that the new proposed algorithm can be recommended as method of choice for solving (3.1).

The Chapter is organized as follows. In Section 3.1 we present several rank revealing decompositions, and the corresponding error analysis. In Section 3.2, we introduce the Kronecker's canonical form and the Van Dooren's algorithm for computing the complete structure of zero eigenvalue, i.e. the number and the sizes of the associated Jordan blocks. The material of these two introductory sections is essential for the development of the new algorithm.

Section 3.3 provides details about the `quadeig` algorithm from Hammarling, Munro and Tisseur [37]. It is based on the second companion form

$$C_2(\lambda) = \begin{pmatrix} C & -\mathbb{I} \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I} \end{pmatrix}. \tag{3.2}$$

The main steps of the algorithm is the parameter scaling and deflation process that removes certain number of zero and infinite eigenvalues. We supplement the description of the main steps of `quadeig` with the analysis of backward stability; this, in turn, will reveal important issues that will guide the modification introduced in the rest of the chapter.

In Section 3.4 we tackle another problem of scaling. While the parameter scaling, successfully used in `quadeig`, removes the balance in norms of the coefficient matrices, it cannot remove different scaling of the matrix entries. Such imbalance between the entries of a particular matrix may be source of artificial ill-conditioning that causes to numerical algorithms that are sensitive to scaling. We propose to modify and deploy the balancing process [9], for problems in which the range of the elements of the coefficient matrices is high in absolute value. We provide brief review of the method and numerical examples to demonstrate the benefits of balancing.

Our main result is presented in Section 3.5. We first point out an interesting fact that the deflation process in `quadeig` algorithm is actually just the first step of the Van Dooren's algorithm for determining the canonical structure of the zero eigenvalue. We then present an interesting case study example where `quadeig` fails to determine all zero eigenvalues. The key problem is that there may be more than one Jordan block of the eigenvalue zero, and the deflation process in `quadeig` detects only one. After deflation, the QZ algorithm is unable to detect the remaining zeros.

We develop a test for the existence of Jordan blocks in terms of the original coefficient matrices. In addition, we develop a full deflation algorithm, which uses the structure of the linerization in the first two steps of the deflation. Finally, we present examples which demonstrates the power of the proposed method.

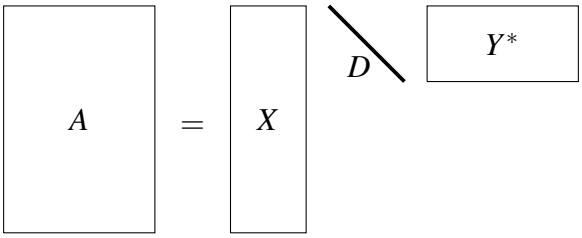
In Section 3.6, we develop the LU based `quadeig`, that is we derive the transformation matrices for deflation process when complete LU factorization is used for rank determination (instead of the QR factorization). Furthermore, we present an algorithm for computing the structure of the zero eigenvalue using the rank revealing LU factorization; this is a non–orthogonal (but numerically well founded analogon of the Van Dooren's algorithm).

In §3.7 we present examples that demonstrates the difference between the rank revealing

factorizations used in the deflation process. Also, we illustrate the importance of the choice of truncation strategy for rank determination in the first step of the preprocessing. It is clear from these examples that the norm-wise backward error can be misleading, and we propose to use the component-wise backward error instead.

3.1 Rank revealing decompositions

Since detecting zero or infinite eigenvalues is based on numerical rank decision, we briefly discuss rank revealing decompositions (RRD, see [19]). For a general $m \times n$ matrix A, we say that $A = XDY^*$ is a rank revealing decomposition if both X and Y are of full column rank and well conditioned, and D is diagonal nonsingular (for example, the SVD and the pivoted LDU decmposition).



In finite precision computation, such a decomposition is computed only approximately and we have $A + \delta A = \widetilde{X}\widetilde{D}\widetilde{Y}^*$, where δA denotes initial uncertainty and/or the backward error that corresponds to the numerically computed $\widetilde{X}$, $\widetilde{D}$ and $\widetilde{Y}$. Hence, any decision on the rank actually applies to $A + \delta A$.

Since the full rank matrices are open dense set in $\mathbb{C}^{m\times n}$ ($\mathbb{R}^{m\times n}$), it is unlikely that, in general, the rank will be determined correctly using a finite precision computation. Furthermore, in many applications the matrix has been already contaminated by errors (previous computational steps, measurement errors on the input etc.) and a firm statement about its rank is illusory.

The structure and the size of δA depends on a particular algorithm for computing a RRD. In some special cases, it is possible to compute such a rank revealing decomposition in a forward stable way so that the rank is determined exactly. For instance, Demmel [18] showed that the pivoted LU decomposition $P_1CP_2 = LDU$ of any Cauchy matrix $C = C(x,y)$ ($C_{ij} = 1/(x_i + y_j)$) can be computed so that each entry of L, D, U is computed to high relative accuracy, that all zeros are computed exactly and that L and U are well conditioned.

We refer to [19], [18], [33] for a more in depth discussion and definition of a numerical rank.

3.1.1 Singular Value Decomposition (SVD)

The ultimate rank revealing decomposition is the singular value decomposition (SVD), in particular because it provides not only the information on the rank, but also exact distances to matrices of lower ranks.

Theorem 3.1. *(Eckart-Young [29], Mirsky [52]) Let the SVD of $A \in \mathbb{C}^{m\times n}$ be*

$$A = U\Sigma V^*,\ \Sigma = \mathrm{diag}(\sigma_i)_{i=1}^{\min(m,n)},\ \sigma_1 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0.$$

For $k \in \{1,\ldots,\mathrm{rank}(A)\}$, define $U_k = U(:,1:k)$, $\Sigma_k = \Sigma(1:k,1:k)$, $V_k = V(:,1:k)$, and $A_k = U_k\Sigma_k V_k^*$. The optimal rank k approximations in $\|\cdot\|_2$ and the Frobenius norm $\|\cdot\|_F$ are

$$\min_{\mathrm{rank}(N)\leq k} \|A-N\|_2 = \|A-A_k\|_2 = \sigma_{k+1},\quad \min_{\mathrm{rank}(N)\leq k} \|A-N\|_F = \|A-A_k\|_F = \sqrt{\sum_{i=k+1}^{\min(n,m)} \sigma_i^2}.$$

The above theorem allows us to say something about the ranks of the matrices in the vicinity of A, and to estimate what change is needed to lower the rank. In a framework of numerical computation with noisy data, this kind of information is more proper than simply claiming the rank to be r.

State of the art packages for matrix computation such as LAPACK [2] provide several subroutines for computing the SVD:

- `xGESVD`, which implements the zero shift QR method [20] on the bidiagonal matrix.
- `xGESDD`, which implements the divide and conquer scheme on the bidiagonal matrix [35].
- `xGESVJ, xGEJSV` are the implementations of the Jacobi SVD, [25], [26].

In some cases we resort to less expensive tools, that usually perform well – the pivoted QR factorization and LU decomposition.

3.1.2 QR factorization with column and complete pivoting

QR factorization with column pivoting is a tool of trade in many applications, in particular when the numerical rank of a matrix plays an important role. Particularly successful is the Businger–Golub [12] pivot strategy which, for $A \in \mathbb{C}^{m\times n}$, computes a permutation matrix P, a unitary Q and an $\min(m,n)$ upper triangular (trapezoidal if $m < n$) matrix R such that

$$AP = Q\begin{pmatrix} R \\ \mathbf{0} \end{pmatrix},\ \text{where}\ |R_{ii}| \geq \sqrt{\sum_{k=i}^{j} |R_{kj}|^2},\ \text{for all}\ 1 \leq i \leq j \leq n. \tag{3.3}$$

Here, for the sake of brevity, we consider only the case $m \geq n$. If $m < n$, then R is $m \times n$ upper trapezoidal and the zero block in (3.3) is void. If $r_A = \mathrm{rank}(A)$, then $\prod_{i=1}^{r_A} R_{ii} \neq 0$ and $R(r_A+1:n, r_A+1:n) = \mathbf{0}$. In general, if $k \in \{1,\ldots,n\}$, and if we introduce the block partition

$$R = \begin{pmatrix} R_{[11]} & R_{[12]} \\ \mathbf{0} & R_{[22]} \end{pmatrix},\ R_{[11]} \in \mathbb{C}^{k\times k}, \tag{3.4}$$

then matrix $\begin{pmatrix} R_{[11]} & R_{[12]} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ can be interpreted as upper triangular matrix in QR decomposition of singular matrix $A+\Delta A$, i.e.

$$(A+\Delta A)P \equiv (A - Q\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R_{[22]} \end{pmatrix}P^T)P = Q\begin{pmatrix} R_{[11]} & R_{[12]} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \Delta A \equiv -Q\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R_{[22]} \end{pmatrix}P^T. \quad (3.5)$$

Hence, if k is such that $R_{[11]}$ is of full rank, then $A+\Delta A$ is of rank k and

$$\|\Delta A\|_F = \|R_{[22]}\|_F \leq \sqrt{n-k}|R_{k+1,k+1}|.$$

Hence, if $\gamma > 0$ is a given threshold, and if we can find an index k ($1 \leq k < n$) such that

$$\sqrt{n-k}|R_{k+1,k+1}|/\|A\|_F \leq \gamma, \quad (3.6)$$

then A is γ-close to the rank k matrix $A+\Delta A$, whose pivoted QR factorization (3.5) is obtained from (3.3) by setting in the partition (3.4) the block $R_{[22]}$ to zero. Of course, we would take the smallest possible k that satisfies (3.6).

The essence of rank revealing capability of the factorization is in the fact that such a r_A will very likely be visible on the diagonal of R if A is close to a rank r_A matrix. This is due to the fact that the $|R_{ii}|$'s mimic the distribution of the singular values of A, and to the fact that the SVD gives the exact distances to the lower rank approximations to A (see Theorem 3.1).

Example 3.1. To illustrate this discussion, we generate 200×200 matrix A as $A = XY^T + E$, where X and Y are 200×100 pseudo-random matrices generated in Matlab using the function `randn()`, and E is a pseudo-random matrix with entries bounded by 10^{-7}. In Figure 3.1, we display the singular values of A (as computed by the function `svd()`) and the absolute values of the diagonal entries of R, which is computed using the Businger-Golub pivoting.

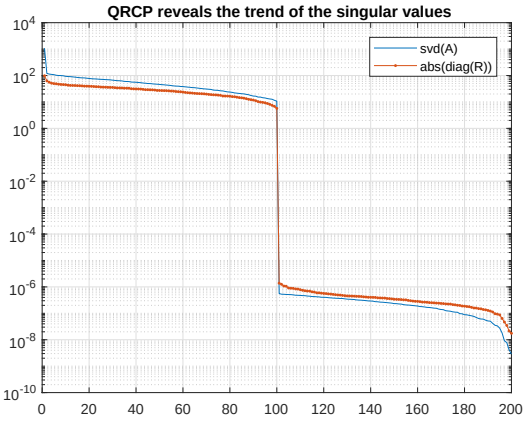


Figure 3.1: Comparison of the absolute values of the diagonal entries of R from (3.3) and the singular values of A. Note that the QR factorization correctly detects that A is $O(10^{-7})$ close to a matrix of rank 100.

Remark 3.1. The rank–k approximation $A+\Delta A$ defined in (3.5) in general does not share the optimality property of the matrix A_k from Theorem 3.1, but it has one distinctive feature: it always matches A exactly at the selected k columns, while A_k in general does not match any part of A.

An efficient and numerically reliable implementation of (3.3) is available e.g. in LAPACK [2] in the function `xGEQP3`, which is also under the hood of the Matlab's function `qr`; for the numerical and software details we refer to [23].

Computation of the QR factorization in finite precision arithmetic is backward stable [41]: for the computed factors $\widetilde{P}$, $\widetilde{Q}$, $\widetilde{R}$, there exists a backward error δA and a unitary matrix $\widehat{Q}$ such that

$$(A+\delta A)\widetilde{P} = \widehat{Q}\begin{pmatrix}\widetilde{R}\\ \mathbf{0}\end{pmatrix},\ \ \|\delta A\|_F \le \varepsilon_1\|A\|_F,\ \ \|\widetilde{Q}-\widehat{Q}\|_F \le \varepsilon_2. \tag{3.7}$$

In fact, the backward stability can be stated in a stronger form – the backward error in each column is small relative to its norm,

$$\|\delta A(:,i)\|_2 \le \varepsilon_3\|A(:,i)\|_2,\ \ i=1,\ldots,n. \tag{3.8}$$

This is an important feature if some columns of A are, by its nature, much smaller than the largest ones (different weighting factors, different physical units); (3.8) assures that the computed factorization contains the information carried by small columns of A. While (3.7, 3.8) hold independent of pivoting, pivoting is important for the accuracy of the computed factorization, and for the rank revealing. The error bounds ε_j are a moderate functions of the matrix dimensions times the machine roundoff unit $\mathbf{u}$.

If, for a suitable partition of $\widetilde{R}$, analogous to (3.4), we can determine k such that $\widetilde{R}_{[22]}$ can be chopped off, we have

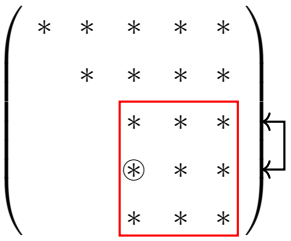
$$(A+\delta A+\Delta A)\widetilde{P} = \widehat{Q}\begin{pmatrix}\widetilde{R}_{[11]} & \widetilde{R}_{[12]}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix},\ \ \|\delta A\|_F \le \varepsilon_1\|A\|_F,\ \ \|\widetilde{Q}-\widehat{Q}\|_F \le \varepsilon_2, \tag{3.9}$$

where

$$\Delta A \equiv -\widehat{Q}\begin{pmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \widetilde{R}_{[22]}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\widetilde{P}^T. \tag{3.10}$$

Note that $(\Delta A\widetilde{P})(:,1:k)=\mathbf{0}$, so that the overall backward error of the computation of the factorization and truncation of $\widetilde{R}_{[22]}$ in the most important columns (as determined by pivoting) remains as in (3.8). Notice that in (3.9) we have additional δA from computation of QR decomposition in comparison with (3.5).

Complete pivoting. In some applications (e.g. weighted least squares) the rows of the data matrix may vary over several orders of magnitude, and it is desirable to have backward error that can be bounded row-wise analogously to (3.8). A pioneering work is done by Powell and Reid [58], who introduced QR factorization with complete pivoting. More precisely, in a j-th step, before deploying the Householder reflector to annihilate below-diagonal entries in the j-th column, row swapping is used to bring the absolutely largest entry to the diagonal position:



As any pivoting, this precludes efficient blocking and using BLAS 3 level primitives.

Björck [8] noted that the dynamic complete pivoting can be replaced with an initial sorting of the rows of A to obtain them in monotonically decreasing order with respect to the ℓ_∞ norm. If P_r is the corresponding row permutation matrix, and if we set $A := P_r A$, then

$$\|A(1,:)\|_\infty \geq \|A(2,:)\|_\infty \geq \ldots \geq \|A(m,:)\|_\infty, \tag{3.11}$$

and we proceed with the column pivoted factorization (3.3). An error analysis of this scheme and Householder reflector based QR factorization is given by Cox and Higham [15].

$$\max_{i=1:m} \frac{\|\delta A(i,:)\|_\infty}{\|A(i,:)\|_\infty} \leq \varepsilon_4 \max_{i=1:m} \frac{\alpha_i}{\|A(i,:)\|_\infty}, \quad \text{where} \quad \alpha_i = \max_{j,k} |\widetilde{A}_{ij}^{(k)}|, \tag{3.12}$$

and $\widetilde{A}^{(k)}$ is the kth computed (in finite precision arithmetic) intermediate matrix in the Householder QR factorization. As a result of initial row ordering and the column pivoting, [15] shows that

$$\alpha_i \leq \begin{cases} \sqrt{m-i+1}(1+\sqrt{2})^{i-1}\|A(i,:)\|_\infty, & i \leq n \\ (1+\sqrt{2})^{n-1}\|A(i,:)\|_\infty, & i > n \end{cases}. \tag{3.13}$$

It is worth mentioning that the factor $(1+\sqrt{2})^{n-1}$ is almost never experienced in practice.

An advantage of replacing the dynamic complete pivoting of Powell and Reid with the initial pre-sorting (3.11) followed by column pivoted QRF (3.3) is more efficient software implementation.

Remark 3.2. If we write the completely pivoted factorization as

$$P_r A P_c = Q \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix}, \quad \text{then} \quad A P_c = (P_r^T Q) \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix}$$

is the column pivoted QR factorization (since P_r is orthogonal) and the row pivoting brings

nothing new to the rank revealing property that is encoded in the triangular factor (this is because of the essential uniqueness of the factorization). However it makes difference in the backward stability because of (3.12) and (3.13).

Strong rank revealing pivoting. In some rare cases the column pivoting can miss small singular value of A, i.e. the structure of R may not reveal that A is close to rank deficiency. The most well known example is the Kahan matrix

$$\mathfrak{K}(n;c,s) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & s & 0 & 0 & 0 & 0 \\ 0 & 0 & s^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & s^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & s^{n-1} \end{pmatrix} \begin{pmatrix} 1 & -c & -c & -c & -c & -c \\ 0 & 1 & -c & -c & -c & -c \\ 0 & 0 & 1 & -c & -c & -c \\ 0 & 0 & 0 & 1 & -c & -c \\ 0 & 0 & 0 & 0 & \ddots & -c \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad c^2 + s^2 = 1,$$

which, for c approaching one, has one small singular value and in the factorization (3.3), $|R_{nn}|$ overestimates $\sigma_{\min}(\mathfrak{K}(n;c,s))$ by the factor 2^{n-1}; see e.g. [40, §6.2], [74].

In the strong rank revealing decomposition, the task is to find the permutation P so that the gap (i.e. sharp drop) in the singular values of A is revealed by the gap between the singular values of the diagonal blocks $R_{[11]}$ and $R_{[22]}$ in the partition (3.4); the partition parameter r is also determined in the process. The key idea is, for given r, to iteratively reshuffle the columns (thus updating the pivoting) with the goal to increase the singular values of $R_{[11]}$ as much as possible, and, at the same time, to decrease the singular values of $R_{[22]}$. The error factor between the singular values of A and the diagonal blocks of R is expected to be a moderate function of the dimensions n and r.

In the strong rank revealing pivoting in [36, Algorithms 4 and 5], an additional parameter $\eta > 1$ balances the trade-off between the sharpness of the estimate and the computational cost. The algorithm guarantees the following enclosures of the singular values

$$\begin{aligned} \frac{\sigma_j(A)}{\sqrt{1+\eta^2 r(n-r)}} &\leq \sigma_j(R_{[11]}) \leq \sigma_j(A), \qquad 1 \leq j \leq r \\ \sigma_{r+j}(A) &\leq \sigma_j(R_{[22]}) \leq \sqrt{1+\eta^2 r(n-r)}\,\sigma_{r+j}(A), \qquad 1 \leq j \leq n-r, \end{aligned}$$

at the cost of $\mathcal{O}\left((m+n\log_\eta n)n^2\right)$ arithmetic operations [36, Section 4.4].

3.1.3 The complete orthogonal factorization (URV)

Suppose that in the QR factorization (3.3), the matrix A is of rank $k < n$, so that in the block partition (3.4) $R_{[22]} = \mathbf{0}$. In many instances, it is convenient to compress the trapezoidal matrix

$(R_{[11]}\ R_{[12]})$ to triangular form by an additional LQ factorization.

This LQ is equivalent to computing the QR factorization

$$\begin{pmatrix} R_{[11]}^* \\ R_{[12]}^* \end{pmatrix} = Z_R^* \begin{pmatrix} T_{[11]}^* \\ \mathbf{0} \end{pmatrix},$$

where $T_{11} \in \mathbb{C}^{k\times k}$ is lower triangular and nonsingular. By a composition of these two steps we get the so called complete orthogonal decomposition of A

$$A = Q \begin{pmatrix} T_{[11]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} Z, \text{ where } Z = Z_R P^T. \tag{3.14}$$

The above described process for computing the complete orthogonal decomposition can be summarized in Algorithm 3.1.1.

Algorithm 3.1.1 Complete orthogonal decomposition of A

INPUT: $A \in \mathbb{C}^{m\times n}$
OUTPUT: Q, $T_{[11]}$, Z, so that $A = Q \begin{pmatrix} T_{[11]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} Z^*$

1: *Optional:* Compute permutation P_2 such that $\|e_i^T P_2 A\|_\infty \geq \|e_{i+1}^T P_2 A\|_\infty$, $i = 1,\ldots,n-1$.
2: Compute the QR factorization with column pivoting $(P_2^T A)P = Q \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix}$.
3: Compute the QR factorization with complete pivoting of the truncated matrix

$$R^*\Pi_1 = \begin{pmatrix} T_{[11]}^* \\ T_{[12]}^* \end{pmatrix} \Pi_1 = \Pi_2^T Z_R \begin{pmatrix} T_{[11]}^* \\ \mathbf{0} \end{pmatrix}.$$

4: $Z = P\Pi_2^T Z_R$.
5: **if** $P_2 \neq \mathbb{I}$ **then**
6: $\quad Q = P_2^T Q$
7: **end if**
8: $Q = Q \begin{pmatrix} \Pi_1 & \mathbf{0} \\ \mathbf{0} & \mathbb{I} \end{pmatrix}$

Backward error analysis. In the QR factorization, the matrix A is multiplied from the left by a sequence of unitary transformations. Hence, there is no mixing of the columns; we can analyse the process by following the changes of each column separately; that is why the column-wise backward error bound (3.8) is natural and straightforward to derive. The transformations from the right in the pivoted QR factorization are the error free column interchanges.

On the other hand, (3.14) involves nontrivial two–sided transformations of A, and more careful implementation and error analysis are needed to obtain backward stability similar to

the one described in §3.1.2. The following theorem provides a backward error bound for the Algorithm 3.1.1:

Theorem 3.2. *Let $\widetilde{Q}$, $\widetilde{T}_{[11]}$ and $\widetilde{Z}$ be the computed factors of complete orthogonal decomposition of A. Then they correspond to the exact complete orthogonal decomposition of matrix*

$$A+\delta A+\Delta A+\widehat{Q}\begin{pmatrix}\delta\widetilde{R}_{[11]} & \delta\widetilde{R}_{[22]}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\widetilde{P}^T=\widehat{Q}\begin{pmatrix}\Pi_1 & \mathbf{0}\\ \mathbf{0} & \mathbb{I}\end{pmatrix}\begin{pmatrix}\widetilde{T}_{[11]} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\widehat{Z}^*\Pi_2\widetilde{P}^T,$$

where $\widehat{Q}\approx\widetilde{Q}$ and $\widehat{Z}\approx\widetilde{Z}$ are orthogonal (unitary) and

$$\|\delta A\|_F\le\varepsilon_1\|A\|_F,\ \ \Delta A=-\widehat{Q}\begin{pmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \widetilde{R}_{[22]}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\widetilde{P}^T,\ \ \|\delta\widetilde{R}(:,i)\|\le\varepsilon_3\|\widetilde{R}(:,i)\|.$$

Proof. For the first step we have the relation (3.9). Set $\widetilde{R}=\begin{pmatrix}\widetilde{R}_{[11]} & \widetilde{R}_{[12]}\end{pmatrix}$ and compute the QR factorization of $\widetilde{R}^*$. We use the complete pivoting, and the the computed factors $\widetilde{Z}$, $\widetilde{T}^*_{[11]}$ satisfy

$$(\widetilde{R}+\delta\widetilde{R})^*\Pi_1=\Pi_2^T\widehat{Z}\begin{pmatrix}\widetilde{T}^*_{[11]}\\ \mathbf{0}\end{pmatrix},\tag{3.15}$$

where $\widehat{Z}$ is unitary, $\|\widetilde{Z}-\widehat{Z}\|_F\le\varepsilon_2$ and, by (3.12, 3.13), $\|\delta\widetilde{R}(:,i)\|\le\varepsilon_3\|\widetilde{R}(:,i)\|$. Including (3.15) in (3.9), we obtain

$$(A+\delta A+\Delta A+\widehat{Q}\begin{pmatrix}\delta\widetilde{R}_{[11]} & \delta\widetilde{R}_{[22]}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\widetilde{P}^T)=\widehat{Q}\begin{pmatrix}\Pi_1 & \mathbf{0}\\ \mathbf{0} & \mathbb{I}\end{pmatrix}\begin{pmatrix}\widetilde{T}_{[11]} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\widehat{Z}^*\Pi_2\widetilde{P}^T,$$

where δA is from (3.7, 3.8), ΔA is as in (3.10)

Hence, the k pivotal columns of A (as selected by $\widetilde{P}$) have, individually, small backward errors of the type (3.8). Note that the complete pivoting in (3.15) is essential for column-wise small backward error in $\widetilde{R}$ and thus is A. □

3.1.4 Rank revealing LU and Cholesky factorizations

Using Gaussian eliminations, every matrix $A\in\mathbb{R}^{n\times n}$ with all its leading principal minors different from zero can be factored as a product of a lower triangular matrix L and an upper triangular matrix U, that is $A=LU$. In every step k of Gaussian elimination the goal is to zero

out the elements below the diagonal in the k-th column by the following elementary operations

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)} \tag{3.16}$$

$$= a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}, \;\; i = k+1, \ldots, n, \;\; j = k+1, \ldots, n, \tag{3.17}$$

where $a_{ij}^{(k)}$ are the elements of the matrix $A^{(k)} = \begin{pmatrix} A_{[11]}^{(k)} & A_{[12]}^{(k)} \\ \mathbf{0} & A_{[22]}^{(k)} \end{pmatrix}$ in the kth step. It is clear from equations (3.16)-(3.17) that the problem occurs when $a_{kk}^{(k)} = 0$. Also m_{ik} can be large (if the pivot $a_{kk}^{(k)}$ is small) and this may result in loss of significant digits in finite precision arithmetics. This is why the following pivoting strategies are introduced:

- **partial pivoting.** in k-th step, the k-th and the r-th rows are interchanged where r is such that

  $$|a_{rk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|.$$

  The resulting LU is $PA = LU$, where P is the corresponding permutation matrix.

- **complete pivoting.** in k-th step, the k-th and the r-th row, and the k-th and the s-th column are interchanged, where r and s are such that

  $$|a_{rs}^{(k)}| = \max_{k \le i,j \le n} |a_{ij}^{(k)}|.$$

  The resulting LU is $PAQ = LU$, where P, Q are the corresponding permutation matrices.

Moreover, if partial pivoting is turned on, every square matrix A admits LU factorization $PA = LU$.

Let $A \in \mathbb{R}^{m \times n}$ and $m \ge n$; clearly the elimination process applies in the rectangular case as well. It is shown in [41] that, if the Gaussian eliminations run to completion, the computed factors $\widetilde{L} \in \mathbb{R}^{m \times n}$ and $\widetilde{U} \in \mathbb{R}^{n \times n}$ satisfy

$$\widetilde{L}\widetilde{U} = A + \Delta A, \;\; |\Delta A| \le \gamma_n |\widetilde{L}||\widetilde{U}|, \;\; \gamma_n = \frac{n\mathbf{u}}{1 - n\mathbf{u}}.$$

It is usually said that partial pivoting is good and reliable enough, so that the complete pivoting is not needed. However, there is a whole collection of problems for which Gaussian eliminations with partial pivoting are unstable, see e.g. [73].

The important difference between LU and QR factorization is discussed and exploited in [22]. The difference is that the LU factorization is invariant under row and column scaling. More precisely, assume that matrix A is permuted so that $A \equiv QAP = LU$ is the LU factorization with complete pivoting. Moreover, assume that A can be written as $A = D_1 Z D_2$, where the elements of the diagonal matrix D_1 are sorted in the increasing order by the element magnitude, and Z

admits an accurate LU factorization $Z = L_Z U_Z$ with moderate $\|L_Z\|_2$. Then the computed matrix $\widetilde{L}$ for A has a columnwise small relative error

$$\frac{\|(L-\widetilde{L})e_i\|_2}{\|Le_i\|_2} \leq \max_{j>i} \left| \frac{(D_1)_{jj}}{(D_1)_{ii}} \right| \|(L_Z - \widetilde{L}_Z)e_i\|_2.$$

In exact arithmetics, LU factorization with complete pivoting is rank revealing factorization, that is if $\mathrm{rank}(A) = r < n$ we have

$$PAQ = LU = \begin{pmatrix} L_{[11]} & \mathbf{0} \\ L_{[21]} & \mathbb{I}_{n-r} \end{pmatrix} \begin{pmatrix} U_{[11]} & U_{[12]} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

However, there are examples where LU factorization with complete pivoting fails to detect the nearly singular matrix, that is there are no small pivots in the factorization although matrix contains a small singular value. This matrix is of the following form

$$W = \begin{pmatrix} 1 & -1 & -1 & \dots & -1 \\ & 1 & -1 & \dots & -1 \\ & & \ddots & \vdots & \vdots \\ & & & & 1 \end{pmatrix}.$$

Pan proved in [56] that there exists a rank revealing LU factorization, and obtained the bounds similar to those for the strong rank revealing QR. Before we state the result, we recall the notion of matrix volume, and the local μ-maximum volume.

Definition 3.1. *Let $A \in \mathbb{R}^{m\times n}$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$, $p = \min(m,n)$, be the singular values of A. The volume of A is defined as*

$$\mathrm{vol}(A) = \sigma_1 \sigma_2 \ldots \sigma_p.$$

Definition 3.2. *Let $A \in \mathbb{R}^{m\times n}$ and B be a submatrix of A formed by any k columns (rows) of A. $\mathrm{vol}(B) \neq 0$ is said to be a local μ-maximum volume in A, $\mu \geq 1$ if*

$$\mu\,\mathrm{vol}(B) \geq \mathrm{vol}(B'), \tag{3.18}$$

for any B' that is obtained by replacing one column (row) of B by a column (row) of A which is not in B.

The $\mu \geq 1$ in (3.18) is user supplied parameter; its role is critical in a volume maximizing iterative scheme to avoid infinite loop that may be caused by rounding errors. Pan proposes to choose $\mu = 1 + \mathbf{u}$, where $\mathbf{u}$ is the machine precision. Pan [56] proved that, for a matrix $A \in \mathbb{R}^{m\times n}$

and any integer $1 \leq k \leq n$ there exists permutation matrices Γ and Π such that

$$\Gamma^T A \Pi = \begin{pmatrix} B_{[11]} & B_{[12]} \\ B_{[21]} & B_{[22]} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_k & \mathbf{0} \\ Z & \mathbb{I}_{n-k} \end{pmatrix} \begin{pmatrix} B_{[11]} & B_{[12]} \\ \mathbf{0} & U_{[22]} \end{pmatrix},$$

where $Z = B_{[21]} B_{[11]}^{-1}$, $U_{[22]} = B_{[22]} - Z B_{[12]}$ and

$$\sigma_k(A) \geq \sigma_{\min}(B_{[11]}) \geq \frac{1}{k(n-k)\mu^2 + 1} \sigma_k(A),$$

$$\sigma_{k+1}(A) \leq \|U_{[22]}\|_2 \leq \left(k(n-k)\mu^2 + 1\right) \sigma_{k+1}(A).$$

The permutation Π is determined so that the volume of the first k columns of $A\Pi$ is a local μ-maximum in A, and Γ is determined so that the volume of the first k rows, $\mathrm{vol}(B_{[11]})$, is a local μ-maximum in the first k columns of $A\Pi$.

Cholesky factorization. Let A be real symmetric positive definite, and let $A = LU$ be the corresponding LU factorization. Note that both L and U are nonsingular. Since $A = A^T$ we have

$$\begin{aligned} U^T L^T = LU \implies & \underbrace{L^{-1} U^T}_{\text{lower triangular}} = \underbrace{U L^{-T}}_{\text{upper triangular}} \\ \implies & L^{-1} U^T = U L^{-T} =: D, \text{ where } D \text{ is diagonal matrix} \\ \implies & U = D L^T. \end{aligned}$$

Hence, we can write $A = LDL^T$. Since A is positive definite, i.e. $x^T A x = x^T L D L^T x > 0$, $x \neq \mathbf{0}$ we can conclude that D is positive definite, and we can write $D = \sqrt{D}\sqrt{D}$. By denoting $R = \sqrt{D} L^T$, we obtain Cholesky factorization $A = R^T R$, where R is upper triangular matrix. If we in addition require that the diagonal of R is positive, the factorization is unique.

There is a similar result of backward stability for Cholesky factorization to that for LU factorization proven in [41]. Namely, if Cholesky factorization runs to completion then the computed factor R satisfies

$$\widetilde{R}^T \widetilde{R} = A + \Delta A, \;\; |\Delta A| \leq \gamma_{n+1} |\widetilde{R}^T| |\widetilde{R}|, \;\; \gamma_{n+1} = \frac{(n+1)\mathbf{u}}{1 - (n+1)\mathbf{u}}.$$

For symmetric positive definite matrix there is a unique Cholesky factorization. On the other hand, if A is only positive semidefinite, generally we do not have uniqueness. For example

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} 0 & \cos\theta \\ 0 & \sin\theta \end{pmatrix}.$$

However, we know that there exists a permutation Π such that $\Pi^T A \Pi$ has a unique Cholesky decomposition

$$\Pi^T A \Pi = R^T R, \;\; R = \begin{pmatrix} R_{[11]} & R_{[12]} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where $R_{[11]}$ is $r \times r$ upper triangular with positive definite diagonal elements. The pivoting strategy ensures that at each step k

$$a_{jj}^{(k)} = \max_{k \le i \le n} a_{ii}^{(k)}, \tag{3.19}$$

and it is equivalent to complete pivoting in Gaussian elimination. In exact arithmetics, the Cholesky factorization with pivoting (3.19) is a rank revealing decomposition.
For $1 \le k \le r$ partition A

$$A = \begin{pmatrix} A_{[11]} & A_{[12]} \\ A_{[12]}^T & A_{[22]} \end{pmatrix},$$

so that $A_{[11]} \in \mathbb{R}^{k \times k}$. Denote by $S_k(A) = A_{[22]} - A_{[12]}^T A_{[11]}^{-1} A_{[12]}$ the Schur complement of $A_{[11]}$ in A, and note that $S_r(A) = \mathbf{0}$. It is proven in [41] how the $S_k(A)$ changes when A is perturbed. Assume for symmetric E that $\|A_{[11]}^{-1} E_{[11]}\|_2 < 1$ holds. Then

$$S_k(A+E) = S_k(A) + E_{[22]} - (E_{[12]}^T W + W^T E_{[12]}) + W^T E_{[11]} W + O(\|E\|_2^2),$$

where $W = A_{[11]}^{-1} A_{[12]}$. This means that the sensitivity of $S_k(A)$ to the perturbation in A essentially depends on the matrix W. If the pivoting strategy (3.19) is used, the following inequality holds

$$\|A_{[11]}^{-1} A_{[12]}\|_2 \le \sqrt{\frac{1}{3}(n-r)(4^r - 1)}. \tag{3.20}$$

If no pivoting is used, the norm in (3.20) can be arbitrary large. Since in the practice, when the pivoting strategy (3.19) is used, $\|A_{[11]}^{-1} A_{[12]}\|_2$ rarely exceeds 10 [41] we can conclude that the Cholesky algorithm with this pivoting is stable algorithm for the semi–definite matrices.

3.2 Kronecker's canonical form for general pencils

Canonical (spectral) structure of a matrix pencil $A - \lambda B$ is, through linearization, an extremely powerful tool for the analysis of quadratic pencils $Q(\lambda) = \lambda^2 M + \lambda C + K$. In particular, since the second companion form

$$A - \lambda B = \begin{pmatrix} C & -\mathbb{I} \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I} \end{pmatrix}$$

is strong linearization, the partial multiplicities, and thus the structure of all eigenvalues (including infinity), are preserved. In a numerical algorithm for the QEP, it is desirable to remove the zero and infinite eigenvalues as early as possible and, thus, canonical structure can be used to

guide such a preprocessing step.

In this section, we briefly review the numerical algorithm by Van Dooren [21], developed for the computation of the structure of eigenvalue λ for a general $m \times n$ pencil $A - \lambda B$, i.e. the number and the orders of the Jordan blocks for λ. The final goal is the Kronecker's Canonical Form, that is a factorization of the form

$$P(A - \lambda B)Q = \operatorname{diag}(L_{\varepsilon_1}, \ldots, L_{\varepsilon_p}, L^P_{\eta_1}, \ldots, L^P_{\eta_q}, I - \lambda N, J - \lambda I), \tag{3.21}$$

where P, Q are constant invertible matrices and

$$L_\mu = \begin{pmatrix} \lambda & -1 & & \\ & \ddots & \ddots & \\ & & \lambda & -1 \end{pmatrix} \in \mathbb{C}^{\mu \times (\mu+1)}, \quad L^P_\mu = \begin{pmatrix} -1 & & \\ \lambda & \ddots & \\ & \ddots & -1 \\ & & \lambda \end{pmatrix} \in \mathbb{C}^{(\mu+1) \times \mu}.$$

N is nilpotent Jordan matrix, and J is in Jordan canonical form. Here, however, we focus our attention only on computing the structure of the eigenvalue 0. Notice that for the infinite eigenvalue one can reverse the pencil. For an arbitrary finite eigenvalue, a suitably shifted pencil is used; see [21] for more details.

For the sake of completeness and later references, we briefly describe the main steps of the staircase reduction for the zero eigenvalue. The pencil $A - \lambda B$ is assumed regular (thus square, $n \times n$), and $\lambda = 0$ is assumed to be among its eigenvalues.

1. Compute the singular value decomposition of A:

$$A = U_A \Sigma_A V_A^*, \tag{3.22}$$

and let $s_1 = n - \operatorname{rank}(A)$. (Since zero is assumed to be an eigenvalue of $A - \lambda B$, A must be column rank deficient.) Note that $AV_A = \left(\begin{array}{c|c} A_2 & \mathbf{0}_{n \times s_1} \end{array} \right)$, where A_2 is of full column rank $n - s_1$. Partition $BV_A = \left(\begin{array}{c|c} B_2 & B_1 \end{array} \right)$ in the compatible manner. If we multiply the pencil by V_A from the right we get

$$(A - \lambda B)V_A = \left(\begin{array}{c|c} A_2 - \lambda B_2 & -\lambda B_1 \end{array} \right). \tag{3.23}$$

2. Compute the singular value decomposition of B_1

$$B_1 = U_B \Sigma_B V_B^*. \tag{3.24}$$

The rank of B_1 is s_1 (full column rank) since the initial matrix pencil is assumed regular, and $U_B^* B_1 = \left(\begin{array}{c} B_{1,1} \\ \hline \mathbf{0}_{n - s_1 \times s_1} \end{array} \right)$, $\det B_{1,1} \neq 0$. Multiply the pencil (3.23) by U_B^* from the left to

get

$$U_B^*(A-\lambda B)V_A = \left(\begin{array}{c|c} A_{2,1}-\lambda B_{2,1} & -\lambda B_{1,1} \\ \hline A_{2,2}-\lambda B_{2,2} & \mathbf{0}_{n-s_1\times s_1} \end{array}\right). \tag{3.25}$$

3. Let P_B be the permutation matrix that swaps the row blocks in the above partition. Thus, we have unitary matrices $P_1 = P_B U_B^*$, $Q_1 = V_A$ so that

$$P_1(A-\lambda B)Q_1 = \left(\begin{array}{c|c} A_{2,2}-\lambda B_{2,2} & \mathbf{0}_{n-s_1\times s_1} \\ \hline A_{2,1}-\lambda B_{2,1} & -\lambda B_{1,1} \end{array}\right), \tag{3.26}$$

where

$$\begin{pmatrix} A_{2,2} \\ A_{2,1} \end{pmatrix} = P_1 A_2 \in \mathbb{C}^{n\times(n-s_1)}$$

is of full column rank.

This concludes the first step of the algorithm. Note that

$$|\det P_1 \det(A-\lambda B)\det Q_1| = |\det(A-\lambda B)| = |\lambda|^{s_1}|\underbrace{|\det B_{1,1}|}_{\neq 0}|\det(A_{2,2}-\lambda B_{2,2})|,$$

which clearly exposes s_1 copies of zero in the spectrum, and reduces the problem to the pencil $A_{2,2}-\lambda B_{2,2}$ of lower dimension $n_2 = n - s_1$. Clearly, if A_{22} is nonsingular, zero has been exhausted from the spectrum of $A-\lambda B$. Otherwise, in the next step, we repeat the described procedure on the $n_2 \times n_2$ pencil $A_{2,2}-\lambda B_{2,2}$ to obtain unitary matrices $\widehat{P}_2$, $\widehat{Q}_2$ so that

$$P_2P_1(A-\lambda B)Q_1Q_2 = \left(\begin{array}{c|c|c} A_{3,3}-\lambda B_{3,3} & \mathbf{0}_{n_3\times s_2} & \mathbf{0}_{n_3\times s_1} \\ \hline A_{3,2}-\lambda B_{3,2} & -\lambda B_{2,2} & \mathbf{0}_{s_2\times s_1} \\ \hline A_{3,1}-\lambda B_{3,1} & A_{2,1}-\lambda B_{2,1} & -\lambda B_{1,1} \end{array}\right),$$

where $P_2 = \operatorname{diag}(\widehat{P}_2, I_{s_1}), Q_2 = \operatorname{diag}(\widehat{Q}_2, I_{s_1})$, and $s_2 = n_2 - \operatorname{rank}(A_{22})$, $n_3 = n_2 - s_2$. As in the first step, $B_{2,2}$ is $s_2 \times s_2$ nonsingular, and $\begin{pmatrix} A_{3,3} \\ A_{3,2} \end{pmatrix}$ is of full column rank. Furthermore, since $\begin{pmatrix} \mathbf{0}_{n_3+s_2,s_2} \\ A_{2,1} \end{pmatrix}$ is a column block in the full column rank matrix, $A_{2,1}$ must have full column rank as well.

This procedure is repeated until in an ℓth step we obtain

$$P(A-\lambda B)Q = \left(\begin{array}{c|c|c|c|c} A_{\ell+1,\ell+1}-\lambda B_{\ell+1,\ell+1} & \mathbf{0}_{n_{\ell+1}\times s_\ell} & \cdots & \mathbf{0}_{n_{\ell+1}\times s_2} & \mathbf{0}_{n_{\ell+1}\times s_1} \\ \hline A_{\ell+1,\ell}-\lambda B_{\ell+1,\ell} & -\lambda B_{\ell,\ell} & \cdots & \mathbf{0}_{s_\ell\times s_2} & \mathbf{0}_{s_\ell\times s_1} \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline A_{\ell+1,2}-\lambda B_{\ell+1,2} & A_{\ell,2}-\lambda B_{\ell,2} & \ddots & -\lambda B_{2,2} & \mathbf{0}_{s_2,s_1} \\ \hline A_{\ell+1,1}-\lambda B_{\ell+1,1} & A_{\ell,1}-\lambda B_{\ell,1} & \ddots & A_{2,1}-\lambda B_{2,1} & -\lambda B_{1,1} \end{array}\right), \tag{3.27}$$

with nonsingular $A_{\ell+1,\ell+1}$. Then, by design, $B_{i,i}$ has full rank s_i for $i = 1,\ldots,\ell$, and $A_{i,i-1}$ has full column rank s_i for $i = 2,\ldots,\ell$.

This procedure is formally described in Algorithm 3.2.1.

Algorithm 3.2.1 Deflation of eigenvalue 0 using SVD [21]

1: $j = 1$; $A_{1,1} = A$; $B_{1,1} = B$; $n_1 = n$;
2: Compute the SVD: $A_{1,1} = U_A \Sigma_A V_A^*$
3: $s_1 = n_1 - \text{rank}(A_{1,1})$
4: **while** $s_j > 0$ **do**
5: Partition matrices: $\left(\begin{array}{c|c} A_{j+1} & \mathbf{0} \end{array} \right) = A_{j,j} V_A$, $\left(\begin{array}{c|c} B_{j+1} & B_j \end{array} \right) = B_{j,j} V_A$
6: Update and partition blocks in row j
7: **for** $i = 1 : j-1$ **do**
8: $\left(\begin{array}{c|c} A_{i,j+1} & A_{i,j} \end{array} \right) = A_{i,j} V_A$; $\left(\begin{array}{c|c} B_{i,j+1} & B_{i,j} \end{array} \right) = B_{i,j} V_A$;
9: **end for**
10: Compute the SVD of $s_j \times n_j$ matrix B_j: $B_j = U_B \Sigma_B V_B^*$
11: Compress B_j to full column rank, permute and partition:
12: $\begin{pmatrix} A_{j+1,j+1} \\ A_{j,j+1} \end{pmatrix} = P_B U_B^* A_{j+1}$; $\begin{pmatrix} B_{j+1,j+1} \\ B_{j,j+1} \end{pmatrix} = P_B U_B^* B_{j+1}$;
13: $\begin{pmatrix} \mathbf{0} \\ B_{j,j} \end{pmatrix} = P_B U_B^* B_j$
14: $n_{j+1} = n_j - s_j$, $j = j+1$
15: Compute the SVD: $A_{j,j} = U_A \Sigma_A V_A^*$
16: $s_j = n_j - \text{rank}(A_{j,j})$
17: **end while**

It has been proven that this algorithm completely determines the structure of the zero eigenvalue of the matrix pencil $A - \lambda B$.

Proposition 3.1 ([21])**.** *The indicies s_i given by Algorithm 3.2.1 completely determine the structure at 0 of the pencil $A - \lambda B$, i.e. $A - \lambda B$ has $s_j - s_{j+1}$ elementary divisors λ^j, $j = 1,\ldots,\ell$.*

Finally, we can conclude that this algorithm also determines the structure of zero eigenvalue for the quadratic eigenvalue problem via a (strong) linearization.

Theorem 3.3. *Algorithm 3.2.1 applied to pencil (3.2) completely determines the structure of the eigenvalue zero for the quadratic eigenvalue problem $Q(\lambda) = (\lambda^2 M + \lambda C + K)x = \mathbf{0}$.*

Proof. Every regular quadratic matrix polynomial $Q(\lambda)$ can be represented in the Smith form, that is

$$Q(\lambda) = E(\lambda) D(\lambda) F(\lambda), \tag{3.28}$$

where $D(\lambda) = \text{diag}(d_1(\lambda),\ldots,d_n(\lambda))$ is a diagonal polynomial matrix with monic scalar polynomials $d_i(\lambda)$ such that $d_i(\lambda)$ is divisible by $d_{i-1}(\lambda)$, and $E(\lambda), F(\lambda)$ are $n \times n$ matrix polynomials with constant nonzero determinants.
The elements $d_1(\lambda),\ldots,d_r(\lambda)$ in the Smith form are called invariant polynomials and they are

uniquely determined by $Q(\lambda)$. Recall that, for an eigenvalue λ_0 of $Q(\lambda)$, we can represent the invariant polynomials as

$$d_i(\lambda) = (\lambda - \lambda_0)^{\alpha_i} p_i(\lambda), \;\; \alpha_i \geq 0, \;\; p_i(\lambda_0) \neq 0, \tag{3.29}$$

where the numbers $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ represent partial multiplicities of eigenvalue λ_0. The elements $(\lambda - \lambda_0)^{\alpha_i}$ are the elementary divisors for the eigenvalue λ_0, and α_i represents the dimension of the ith Jordan block for the eigenvalue λ_0.

Now, since the second companion form is a strong linearization, this means that the partial multiplicities of all eigenvalues of $Q(\lambda)$, including infinity, are preserved. Proposition 3.1 states that h indices $\{s_i\}$ computed by the Algorithm 3.2.1 completely determine the structure of the eigenvalue zero for the given pencil. In our case the pencil is the second companion form the linearization, and thus they completely determine the structure of the eigenvalue zero for $Q(\lambda)$. □

3.3 The algorithm `quadeig`

As we discussed in the introduction of this chapter, zero and infinite eigenvalues are difficult to detect in finite precision arithmetic, and their presence may impair convergence of an algorithm for solving the linearized problem. Since those eigenvalues are related to the null spaces of M and K, and since non-singularity is a generic matrix property (holds on the open dense set), the distinction finite–infinite, or zero–nonzero, is numerically delicate issue. On the other hand, if we could remove at least some of them in a numerically safe way, that would save the QZ algorithm the trouble of dealing with zeros and infinities in the spectrum. Besides, removing those eigenvalues early in a computational scheme facilitates efficient iterations with reduced problem's dimension.

This motivated [37] to develop a new deflation scheme that removes $n - r_M$ infinite and $n - r_K$ zero eigenvalues, where $r_M = \operatorname{rank} M$ and $r_K = \operatorname{rank} K$. The remaining generalized linear eigenvalue problem is of the dimension $r_M + r_K$; it may still have some infinite and zero eigenvalues, and their detection then depends on the performance of the QZ algorithm. The computation is done in the framework of the linearization (3.2).

In this section, we analyze `quadeig` in detail. For the sake of the completeness, we first give a detailed algebraic description of the reduction in `quadeig`. In addition, we provide a backward error analysis of the deflation process.

3.3.1 Parameter scaling

The main feature of `quadeig` is the introduction of parameter scaling in order to equilibrate the backward errors for the original problem and the corresponding second companion form

linearization C_2 as described in Subsection 2.3.2, i.e. we consider the scaled quadratic eigenvalue problem

$$\lambda = \gamma\mu, \widetilde{Q}(\mu) = Q(\lambda)\delta = \mu^2(\gamma^2\delta M) + \mu(\gamma\delta C) + (\delta K),$$

where γ and δ are defined so that the norms of the coefficient matrices M, C and K are approximately equal and close to one.

In [37], two types of scaling are used:

Fan, Lin and Van Dooren scaling. γ and δ are defined as the solution of the minimization problem

$$\min_{\gamma,\delta} \max\{\|K\|_2 - 1, \|C\|_2 - 1, \|M\|_2 - 1\}, \tag{3.30}$$

that is,

$$\gamma = \sqrt{\frac{\|K\|_2}{\|M\|_2}}, \ \ \delta = \frac{2}{\|K\|_2 + \|C\|_2\gamma}. \tag{3.31}$$

Tropical scaling. γ and δ are defined as tropical roots of max-times scalar quadratic polynomial

$$q_{\text{trop}}(x) = \max(\|M\|_2 x^2, \|C\|_2 x, \|K\|_2), \ \ x \in [0, \infty\rangle. \tag{3.32}$$

Define $\tau_Q = \frac{\|C\|_2}{\sqrt{\|M\|_2\|K\|_2}}$. If $\tau_Q \leq 1$, (3.32) has the double root

$$\gamma^+ = \gamma^- = \sqrt{\frac{\|K\|_2}{\|M\|_2}},$$

and if $\tau_Q > 1$ there are two distinct roots

$$\gamma^+ = \frac{\|C\|_2}{\|M\|_2} > \gamma^- = \frac{\|K\|_2}{\|C\|_2}.$$

Hence, when $\tau_Q > 1$, scaling with the parameters

$$\gamma = \gamma^+, \ \ \delta = (q_{\text{trop}}(\gamma^+))^{-1}$$

is used to compute the eigenvalues outside of the unit circle, and scaling using the parameters

$$\gamma = \gamma^-, \ \ \delta = (q_{\text{trop}}(\gamma^-))^{-1}$$

is used to compute those eigenvalues inside the unit circle. With this choice, the denominator in the bound

$$\frac{1}{\sqrt{2}} \leq \frac{\eta_Q(z_1, \alpha, \beta)}{\eta_{C_2}(z, \alpha, \beta)} \leq 2^{3/2} \frac{\max(1, \max(\|M\|_2, \|C\|_2, \|K\|_2))}{|\alpha|^2\|M\|_2 + |\alpha||\beta|\|C\|_2 + |\beta|\|K\|_2} \frac{\|z\|_2}{\|z_1\|_2} \tag{3.33}$$

is $O(1)$.

Example 3.2. Recall the Example 2.1. If we use the Fan, Lin and Van Dooren scaling on this problem, the maximum backward error for QEP is `1.793925004288704e-016`. We added the backward errors for the eigenpairs obtained from the scaled problem $\widetilde{Q}(\lambda)$ to Figure 2.2 for better illustration of the importance of parameter scaling.

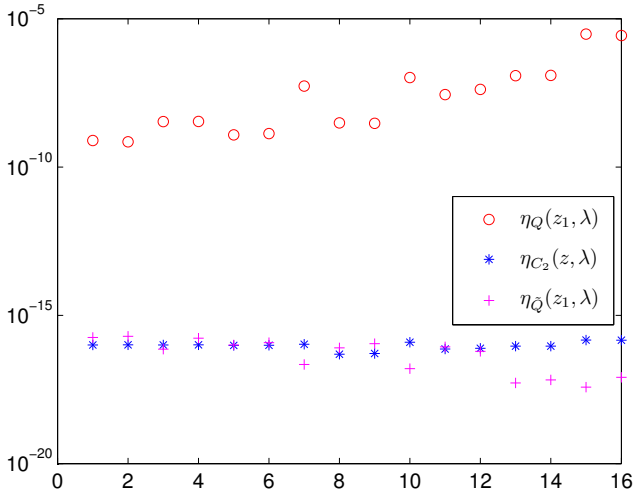


Figure 3.2: Backward errors for the linearization C_2, the original problem quadratic problem and the scaled pencil $\widetilde{Q}(\lambda)$, for the test problem `power_plant`.

3.3.2 Deflation process in `quadeig`

Before introducing the deflation procedure, we analyze the backward error induced by truncation, which will be used in the analysis of the backward error for the deflation process.

Backward error in rank revealing QR factorizations of M and K

The procedure starts with the pivoted (rank revealing) factorizations

$$(P_{r,M}M)\Pi_M = Q_M R_M,\ \ R_M = \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \widehat{R}_M \\ \mathbf{0}_{n-r_M,n} \end{pmatrix}, \tag{3.34}$$

$$(P_{r,K}K)\Pi_K = Q_K R_K,\ \ R_K = \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \widehat{R}_K \\ \mathbf{0}_{n-r_K,n} \end{pmatrix}. \tag{3.35}$$

The initial (optional) row sorting before the column pivoted QR factorization is indicated by the matrices $P_{r,M}$, $P_{r,K}$. Since the sorting is in the ℓ_∞ norm, it is exact even in finite precision. In the absence of row sorting both $P_{r,M}$, $P_{r,K}$ are implicitly set to the identity $\mathbb{I}_n$.

In finite precision (see §3.1.2) the computed matrices $\widetilde{Q}_M$, $\widetilde{R}_M$, $\widetilde{\Pi}_M$ satisfy, independent of

the choice of the permutation matrix $P_{r,M}$,

$$P_{r,M}(M+\delta M)\widetilde{\Pi}_M = \widehat{Q}_M\widetilde{R}_M, \tag{3.36}$$

where $\widehat{Q}_M = \widetilde{Q}_M + \delta\widetilde{Q}_M$ is exactly unitary with

$$\|\delta\widetilde{Q}_M\|_F \equiv \|\widehat{Q}_M - \widetilde{Q}_M\|_F \leq \varepsilon_2, \text{ and } \|\delta M(:,i)\|_2 \leq \varepsilon_3\|M(:,i)\|_2,\ \ i=1,\ldots,n,$$

where $\varepsilon_2, \varepsilon_3$ are as in (3.7), (3.8).

If $P_{r,M}$ is the row sorting permutation, then, in addition,

$$\|\delta M(i,:)\|_2 \leq \varepsilon_{qr}^{\rightrightarrows}\|M(i,:)\|_2,\ \ i=1,\ldots,n,$$

where $\varepsilon_{qr}^{\rightrightarrows}$ is defined using (3.12) and (3.13) in §3.1.2. Since $P_{r,M}$ is unitary, we can absorb it into $\widetilde{Q}_M$ and $\widehat{Q}_M$ and redefine $\widetilde{Q}_M := P_{r,M}^T\widetilde{Q}_M$, $\widehat{Q}_M := P_{r,M}^T\widehat{Q}_M$ and write, instead of (3.36),

$$(M+\delta M)\widetilde{\Pi}_M = \widehat{Q}_M\widetilde{R}_M. \tag{3.37}$$

Analogous statements (3.36–3.37) hold for the factorization (3.35).

Backward error induced by the truncation

However, if we truncate the triangular factor in an attempt to infer the numerical rank, we must push the truncated part into the backward error, as in (3.5). This changes the backward error structure, and the new error bounds depend on the truncation strategy and the threshold. Assume in (3.37) that we can partition $\widetilde{R}_M$ as

$$\widetilde{R}_M = \begin{pmatrix} (\widetilde{R}_M)_{[11]} & (\widetilde{R}_M)_{[12]} \\ \mathbf{0}_{n-k,k} & (\widetilde{R}_M)_{[22]} \end{pmatrix}, \text{ where the } (n-k)\times(n-k) \text{ block } (\widetilde{R}_M)_{[22]} \text{ "is small".}$$

Then we can write a backward perturbed rank revealing factorization

$$(M+\delta M+\underbrace{\widehat{Q}_M\begin{pmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & -(\widetilde{R}_M)_{[22]}\end{pmatrix}\widetilde{\Pi}_M^T}_{\Delta M})\widetilde{\Pi}_M = \widehat{Q}_M\begin{pmatrix}(\widetilde{R}_M)_{[11]} & (\widetilde{R}_M)_{[12]}\\ \mathbf{0}_{n-k,k} & \mathbf{0}_{n-k,n-k}\end{pmatrix}. \tag{3.38}$$

Obviously, ΔM is zero at the k "most linearly independent" columns of M selected by the pivoting, $(\Delta M)\widetilde{\Pi}_M(:,1:k) = \mathbf{0}_{n,k}$. At the remaining $n-k$ columns we have

$$\|(\Delta M)\widetilde{\Pi}_M(:,k+j)\|_2 = \|(\widetilde{R}_M)_{[22]}(:,j)\|_2 \leq |((\widetilde{R}_M)_{[22]})_{k+1,k+1}|.$$

Consider the following choices of k, for a given threshold parameter τ:

1. k is the first index for which $|((\widetilde{R}_M)_{[22]})_{k+1,k+1}| \leq \tau |((\widetilde{R}_M)_{[22]})_{k,k}|$. In that case

$$\max_{j=1:n-k} \|(\Delta M)\widetilde{\Pi}_M(:,k+j)\|_2 \leq \tau |((\widetilde{R}_M)_{[22]})_{k,k}| \leq \tau \min_{i=1:k} \|(M+\delta M)\widetilde{\Pi}_M(:,i)\|_2, \quad (3.39)$$

2. k is the first index for which $|((\widetilde{R}_M)_{[22]})_{k+1,k+1}| \leq \tau \cdot computed(\|M\|_F)$. In that case

$$\max_{j=1:n-k} \|(\Delta M)\widetilde{\Pi}_M(:,k+j)\|_2 \leq \tau \cdot computed(\|M\|_F),$$

3. k is the first index for which

$$|((\widetilde{R}_M)_{[22]})_{k+1,k+1}| \leq \tau \cdot computed(\max\{\|M\|_F, \|C\|_F, \|K\|_F\}).$$

   In that case

$$\max_{j=1:n-k} \|(\Delta M)\widetilde{\Pi}_M(:,k+j)\|_2 \leq \tau \cdot computed(\max\{\|M\|_F, \|C\|_F, \|K\|_F\}).$$

   This strategy (3.) is used in `quadeig` with $\tau = n\mathbf{u}$. Here it is necessary to assume that the coefficient matrices have been scaled so that their norms are nearly equal. Otherwise, such a truncation strategy may discard a block in $\widetilde{R}_M$ because it is small as compared e.g. to $\|C\|_F$ or $\|K\|_F$.

   Remark 3.3. It is important to emphasize that in `quadeig`, scaling the matrices is optional, and if the (also optional) deflation procedure is enabled, the truncation strategy opens a possibility for catastrophic error (severe underestimate of the numerical ranks) if the matrices are not scaled and if their norms differ by orders of magnitude. A user may not be aware of this situation, which can cause large errors.

We now go to the details of the deflation procedure, whose decision tree depends on the numerical ranks of the key matrices M and K.

The case of nonsingular M or nonsingular K

This case can be considered simple; it allows avoiding infinite eigenvalues by simply reversing the pencil.

Both matrices nonsingular. In the simplest case $\mathrm{rank}(M) = \mathrm{rank}(K) = n$, the linearized pencil is transformed by the following equivalence transformation:

$$\begin{pmatrix} Q_M^* & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \left\{ \begin{pmatrix} C & -\mathbb{I}_n \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \Pi_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix}$$

$$
\begin{aligned}
&= \left(\begin{array}{c|c} Q_M^* C \Pi_M & -Q_M^* \\ \hline K\Pi_M & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{c|c} -Q_M^* M \Pi_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array}\right) \\
&= \left(\begin{array}{c|c} Q_M^* C \Pi_M & -Q_M^* \\ \hline K\Pi_M & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{c|c} -R_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array}\right) \equiv A - \lambda B. \qquad (3.40)
\end{aligned}
$$

Proposition 3.2. *Let $\widetilde{A} - \lambda \widetilde{B}$ the computed linearization (3.40). Then it corresponds to an exact linearization of a quadratic pencil $\lambda^2(M+\delta M)+\lambda(C+\delta C)+K$, where, for all $i = 1,\ldots,n$,*

$$\|\delta C(:,i)\|_2 \le \varepsilon_C \|C(:,i)\|_2, \;\; \|\delta M(:,i)\|_2 \le \varepsilon_{qr} \|M(:,i)\|_2.$$

Further, if the row sorting is used in the QR factorization of M then, in addition,

$$\|\delta M(i,:)\|_2 \le \varepsilon_{qr}^{\rightrightarrows} \|M(i,:)\|_2$$

Proof: The proof can be read off as the special case of the proof of the Proposition 3.3 below. ■

Only one of M and K nonsingular. On the other hand, if e.g. $\mathrm{rank}(K) < \mathrm{rank}(M) = n$, then the transformation reads

$$
\begin{aligned}
&\begin{pmatrix} Q_M^* & \mathbf{0} \\ \mathbf{0} & Q_K^* \end{pmatrix} \left\{ \begin{pmatrix} C & -\mathbb{I}_n \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \Pi_M & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \\
&= \left(\begin{array}{c|c} Q_M^* C \Pi_M & -Q_M^* Q_K \\ \hline Q_K^* K \Pi_M & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{c|c} -Q_M^* M \Pi_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array}\right) \\
&= \left(\begin{array}{c|c} Q_M^* C \Pi_M & -Q_M^* Q_K \\ \hline \widehat{R}_K \Pi_K^T \Pi_M & \mathbf{0} \\ \hline \mathbf{0}_{n-r_K,n} & \end{array}\right) - \lambda \left(\begin{array}{c|c} -R_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array}\right) \\
&\equiv \left(\begin{array}{c|cc} X_{11} & X_{12} & X_{13} \\ \hline X_{21} & \mathbf{0}_{r_K,r_K} & \mathbf{0}_{r_K,n-r_K} \\ \hline \mathbf{0}_{n-r_K,n} & \mathbf{0}_{n-r_K,r_K} & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{c|cc} -R_M & \mathbf{0} & \\ \hline \mathbf{0} & -\mathbb{I}_{r_K} & \mathbf{0} \\ & \mathbf{0} & -\mathbb{I}_{n-r_K} \end{array}\right). \qquad (3.41)
\end{aligned}
$$

The reduced $(n+r_K)\times(n+r_K)$ pencil is

$$A - \lambda B = \left(\begin{array}{c|c} X_{11} & X_{12} \\ \hline X_{21} & \mathbf{0}_{r_K,r_K} \end{array}\right) - \lambda \left(\begin{array}{c|c} -R_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_{r_K} \end{array}\right). \qquad (3.42)$$

Consider now the backward stability of the reduction. We assume that the rank truncation is of type 1. (see §3.3.2) with the corresponding backward error as in (3.39), with $\tau = n\mathbf{u}$.

Proposition 3.3. *Let*

$$\widetilde{A}-\lambda\widetilde{B}=\left(\begin{array}{c|c}\widetilde{X}_{11} & \widetilde{X}_{12}\\ \hline \widetilde{X}_{21} & \mathbf{0}_{\widetilde{r}_K,\widetilde{r}_K}\end{array}\right)-\lambda\left(\begin{array}{c|c}-\widetilde{R}_M & \mathbf{0}\\ \hline \mathbf{0} & -\mathbb{I}_{\widetilde{r}_K}\end{array}\right) \tag{3.43}$$

be the computed linearization (3.42). Then it corresponds to exact reduced linearization of a quadratic pencil $\lambda^2(M+\delta M)+\lambda(C+\delta C)+(K+\delta K+\Delta K+\Delta' K)$, where, for all $i=1,\ldots,n$,

$$\|\delta M(:,i)\|_2\leq\varepsilon_{qr}\|M(:,i)\|_2,\ \ \|\delta C(:,i)\|_2\leq\varepsilon_C\|C(:,i)\|_2,\ \ \|\delta K(:,i)\|_2\leq\varepsilon_{qr}\|K(:,i)\|_2;\ \ (3.44)$$

$$\|\Delta' K(:,i)\|_2\leq\eta_K\|K(:,i)\|_2, \tag{3.45}$$

and the truncation error is

$$\max_{j=1:n-k}\|(\Delta K)\widetilde{\Pi}_K(:,k+j)\|_2\leq\tau\min_{i=1:k}\|(K+\delta K)\widetilde{\Pi}_K(:,i)\|_2;\ \ (\Delta K)\widetilde{\Pi}_K(:,1:k)=\mathbf{0}_{n,k}$$

Further, if the row sorting is used in the QR factorization of M then, in addition,

$$\|\delta M(i,:)\|_2\leq\varepsilon_{qr}^{\rightrightarrows}\|M(i,:)\|_2. \tag{3.46}$$

Proof:

(i) Using $\widetilde{R}_M$ in the computed pencil (3.43) can be justified by introducing δM as in (3.36–3.37). This will be the only backward error in M and it can be always estimated as in (3.44), and in the case of complete pivoting as in (3.46).

(ii) It holds that $\widetilde{X}_{11}=computed(\widetilde{Q}_M^*C\widetilde{\Pi}_M)=\widehat{Q}_M^*(C+\delta C)\widetilde{\Pi}_M$. To estimate δC, we start with the fact that

$$computed(\widetilde{Q}_M^*C)=\widetilde{Q}_M^*C+\mathfrak{G},\ \ |\mathfrak{G}|\leq\varepsilon_*|\widetilde{Q}_M^*||C|,\ \ 0\leq\varepsilon_*\leq 2n\mathbf{u}.$$

Since $\widetilde{Q}_M=(\mathbb{I}+\mathfrak{E})\widehat{Q}_M$, $\|\mathfrak{E}\|_2\leq\varepsilon_{qr}$, we have

$$computed(\widetilde{Q}_M^*C)=\widehat{Q}_M^*(\mathbb{I}+\mathfrak{E}^*)C+\mathfrak{G}=\widehat{Q}_M^*(C+\mathfrak{E}^*C+\widehat{Q}_M\mathfrak{G})\equiv\widehat{Q}_M^*(C+\delta C).$$

Since $|\mathfrak{G}|\leq\varepsilon_*|\widetilde{Q}_M^*||C|$, it follows that

$$\|\mathfrak{G}\|_2\leq\|\mathfrak{G}\|_F\leq\varepsilon_*\|\widetilde{Q}_K^*\|_F\|C\|_F\leq\varepsilon_* n\|(\mathbb{I}+\mathfrak{E})\widehat{Q}_K\|_2\|C\|_2\leq\varepsilon_* n(1+\|\mathfrak{E}\|_2)\|C\|_2.$$

Using this, we get column-wise estimates $\|\delta C(:,i)\|_2\leq(\|\mathfrak{E}^*\|_2+\varepsilon_* n(1+\|\mathfrak{E}^*\|_2))\|C(:,i)\|_2$, and (3.44) follows with $\varepsilon_C=(\varepsilon_{qr}+\varepsilon_* n(1+\varepsilon_{qr}))$. Note that the column permutation by $\widetilde{\Pi}_M$ is error free.

(iii) In the same way, using $\widetilde{Q}_K = (\mathbb{I}+\mathfrak{F})\widehat{Q}_K$

$$\begin{aligned}\begin{pmatrix}\widetilde{X}_{12} & \widetilde{X}_{13}\end{pmatrix} &= computed(\widetilde{Q}_M^*\widetilde{Q}_K) = \widehat{Q}_M^*(\mathbb{I}+\mathfrak{E}^*)\widetilde{Q}_K + \mathfrak{H} \ \text{(here } |\mathfrak{H}| \le \varepsilon_*|\widetilde{Q}_M^*||\widetilde{Q}_K|\text{)} \\ &= \widehat{Q}_M^*(\widehat{Q}_K + \mathfrak{F}\widehat{Q}_K + \mathfrak{E}^*\widetilde{Q}_K + \widehat{Q}_M\mathfrak{H}) \equiv \widehat{Q}_M^*(\widehat{Q}_K + \delta\widehat{Q}_K),\end{aligned}$$

with $\|\delta\widehat{Q}_K\|_2 \le \varepsilon_{qr} + \varepsilon_{qr}(1+\varepsilon_{qr}) + n\varepsilon_*(1+\varepsilon_{qr})^2$.

(iv) Note that in this moment the backward error in K contains both the floating point error δK and the truncation error ΔK analogous to (3.38), i.e. $(K+\delta K+\Delta K)\widetilde{\Pi}_K = \widehat{Q}_K\widetilde{R}_K$. Now, the $\delta\widehat{Q}_K$ that helped us justify the error in $\widetilde{X}_{12}, \widetilde{X}_{13}$ must be pushed back into the initial data. If we add it to $\widehat{Q}_K$, then we can write

$$(\widehat{Q}_K + \delta\widehat{Q}_K)\widetilde{R}_K = (K+\delta K+\Delta K+\Delta' K)\widetilde{\Pi}_K, \ \text{ where } \ \Delta' K = \delta\widehat{Q}_K\widetilde{R}_K\widetilde{\Pi}_K^T. \tag{3.47}$$

This is not the QR factorization as $\widehat{Q}_K + \delta\widehat{Q}_K$ need not be unitary. However, it will be of full rank and (3.47) is a rank revealing decomposition. If we set $\Delta_\Sigma K = \delta K + \Delta K + \Delta' K$, then we can represent the computed linearization as

$$\begin{aligned}&\begin{pmatrix}\widehat{Q}_M^* & \mathbf{0} \\ \mathbf{0} & (\widehat{Q}_K+\delta\widehat{Q}_K)^{-1}\end{pmatrix}\left\{\begin{pmatrix}C+\delta C & -\mathbb{I}_n \\ K+\Delta_\Sigma K & \mathbf{0}\end{pmatrix} - \lambda\begin{pmatrix}-M-\delta M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n\end{pmatrix}\right\}\begin{pmatrix}\widetilde{\Pi}_M & \mathbf{0} \\ \mathbf{0} & \widehat{Q}_K+\delta\widehat{Q}_K\end{pmatrix} \\ &= \left(\begin{array}{c|c||c|c} \widetilde{X}_{11} & & \widetilde{X}_{12} & \widetilde{X}_{13} \\ \hline\hline \widetilde{X}_{21} & & \mathbf{0}_{\widetilde{r}_K,\widetilde{r}_K} & \mathbf{0}_{\widetilde{r}_K,n-\widetilde{r}_K} \\ \hline \mathbf{0}_{n-\widetilde{r}_K,n} & & \mathbf{0}_{n-\widetilde{r}_K,\widetilde{r}_K} & \mathbf{0} \end{array}\right) - \lambda\left(\begin{array}{c|c||c|c} -\widetilde{R}_M & & \multicolumn{2}{c}{\mathbf{0}} \\ \hline\hline & & -\mathbb{I}_{\widetilde{r}_K} & \mathbf{0} \\ \hline \mathbf{0} & & \mathbf{0} & -\mathbb{I}_{n-\widetilde{r}_K} \end{array}\right).\end{aligned}$$

■

If $\operatorname{rank}(M) < n$ and $\operatorname{rank}(K) = n$, we proceed with the linearization of the reversed pencil.

Rank deficient case: both M and K rank deficient

We now consider the case when $r_K \le r_M < n$ or $r_M < r_K < n$. In this case, `quadeig` deploys the following transformation of the linear pencil (optionally, depending on r_K/r_M we may reverse the pencil):

$$\begin{aligned}&\begin{pmatrix}Q_M^* & \mathbf{0} \\ \mathbf{0} & Q_K^*\end{pmatrix}\left\{\begin{pmatrix}C & -\mathbb{I}_n \\ K & \mathbf{0}\end{pmatrix} - \lambda\begin{pmatrix}-M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n\end{pmatrix}\right\}\begin{pmatrix}\mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_K\end{pmatrix} \\ &= \left(\begin{array}{c|c||c} Q_M^*C & & -Q_M^*Q_K \\ \hline\hline \widehat{R}_K\Pi_K^T & & \mathbf{0}_{r_K,n} \\ \hline \mathbf{0}_{n-r_k,n} & & \mathbf{0}_{n-r_K,n} \end{array}\right) - \lambda\left(\begin{array}{c|c||c} -\widehat{R}_M\Pi_M^T & & \mathbf{0}_{r_M,n} \\ \mathbf{0}_{n-r_M,n} & & \mathbf{0}_{n-r_M,n} \\ \hline\hline \mathbf{0}_{n,n} & & -\mathbb{I}_n \end{array}\right)\end{aligned}$$

$$
\begin{aligned}
&\equiv \left(\begin{array}{cc|cc} (Q_M^*C)_{11} & (Q_M^*C)_{12} & \multicolumn{2}{c}{\multirow{2}{*}{$-Q_M^*Q_K$}} \\ (Q_M^*C)_{21} & (Q_M^*C)_{22} & \multicolumn{2}{c}{} \\ \hline (\widehat{R}_K\Pi_K^T)_{11} & (\widehat{R}_K\Pi_K^T)_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{cc|cc} -(\widehat{R}_M\Pi_M^T)_{11} & -(\widehat{R}_M\Pi_M^T)_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & -\mathbb{I} & \mathbf{0} \\ \multicolumn{2}{c|}{} & \mathbf{0} & -\mathbb{I} \end{array}\right) \\
&\equiv \left(\begin{array}{cc|cc} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ \hline X_{31} & X_{32} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{cc|cc} Y_{11} & Y_{12} & \mathbf{0}_{r_M,r_K} & \mathbf{0}_{r_M,n-r_K} \\ \mathbf{0}_{n-r_M,r_M} & \mathbf{0}_{n-r_M,n-r_M} & \mathbf{0}_{n-r_M,r_K} & \mathbf{0}_{n-r_M,n-r_K} \\ \hline \mathbf{0}_{r_K,r_M} & \mathbf{0}_{r_K,n-r_M} & -\mathbb{I}_{r_K} & \mathbf{0}_{r_K,n-r_K} \\ \mathbf{0}_{n-r_K,r_M} & \mathbf{0}_{n-r_K,n-r_M} & \mathbf{0}_{n-r_K,r_K} & -\mathbb{I}_{n-r_K} \end{array}\right) \\
&\equiv X - \lambda Y. \qquad (3.48)
\end{aligned}
$$

Note the difference in the transformation from the right: instead of Π_M, we now have $\mathbb{I}_n$, so that $Q_M^*M = R_M\Pi_M^T$ is not upper triangular. Preserving the triangular form in this moment does not seem important because it is likely that it will be destroyed in subsequent steps.

In the next step, `quadeig` computes the complete orthogonal decomposition (i.e. URV decomposition, using unitary matrices Q_X and Z_X)

$$
\begin{array}{c} \\ n-r_M \end{array}\!\!\begin{array}{c} \begin{array}{ccc} r_M & n-r_M & r_K \end{array} \\ \begin{pmatrix} X_{21} & X_{22} & X_{23} \end{pmatrix} \end{array} = Q_X \begin{pmatrix} R_X & \mathbf{0}_{n-r_M,r_M+r_K} \end{pmatrix} Z_X, \;\; R_X \in \mathbb{C}^{(n-r_M)\times(n-r_M)}. \qquad (3.49)
$$

It will be convenient to write this decomposition as

$$
Q_X^* \begin{pmatrix} X_{21} & X_{22} & X_{23} \end{pmatrix} Z_X^* \begin{pmatrix} \mathbf{0} & \mathbb{I}_{n-r_M} \\ \mathbb{I}_{r_M+r_K} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{n-r_M,r_M+r_K} & R_X \end{pmatrix}.
$$

Then (3.48) can be further transformed as follows:

$$
\begin{aligned}
&\left(\begin{array}{cccc} \mathbb{I}_{r_M} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{I}_{r_K} & \mathbf{0} \\ \mathbf{0} & Q_X^* & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I}_{n-r_K} \end{array}\right) \left(\begin{array}{cc|cc} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ \hline X_{31} & X_{32} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right) \left(\begin{array}{c|c} Z_X^* \begin{pmatrix} \mathbf{0} & \mathbb{I}_{n-r_M} \\ \mathbb{I}_{r_M+r_K} & \mathbf{0} \end{pmatrix} & \mathbf{0} \\ \hline \mathbf{0} & \mathbb{I}_{n-r_K} \end{array}\right) \\
&= \left(\begin{array}{cc|cc} \widetilde{X}_{11} & \widehat{X}_{12} & \widehat{X}_{13} & X_{14} \\ \widetilde{X}_{21} & \widehat{X}_{22} & \widehat{X}_{23} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & R_X & \widetilde{X}_{24} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right), \quad \text{where} \quad \begin{array}{l} \begin{pmatrix} \widetilde{X}_{11} & \widetilde{X}_{12} & \widetilde{X}_{13} \\ \widetilde{X}_{21} & \widetilde{X}_{22} & \widetilde{X}_{23} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{31} & X_{32} & \mathbf{0} \end{pmatrix} Z_X^* \begin{pmatrix} \mathbf{0} & \mathbb{I}_{n-r_M} \\ \mathbb{I}_{r_M+r_K} & \mathbf{0} \end{pmatrix}, \\ \widetilde{X}_{24} = Q_X^* X_{24}, \end{array}.
\end{aligned}
$$

The $(1,1)$ diagonal block in the new partition $(=\|=)$ is $(r_M+r_K)\times(r_M+r_K)$, and

- $\mathbf{n-r_M=r_K}$: $\widehat{X}_{ij} = \widetilde{X}_{ij}$, $i = 1,2$, $j = 2,3$;
- $\mathbf{n-r_M>r_K}$: $\widehat{X}_{12} = \widetilde{X}_{12}(:,1:r_K), \widehat{X}_{13} = (\widetilde{X}_{12}(:,r_K+1:n-r_M),\, \widetilde{X}_{13})$,
  $\widehat{X}_{22} = \widetilde{X}_{22}(:,1:r_K), \widehat{X}_{23} = (\widetilde{X}_{22}(:,r_K+1:n-r_M),\, \widetilde{X}_{23})$

- $\mathbf{n-r_M<r_K}$: $\widehat{X}_{12}=(\widetilde{X}_{12},\,\widetilde{X}_{13}(:,1:r_K+r_M-n)),\widehat{X}_{13}=\widetilde{X}_{13}(:,r_K+r_M-n+1:r_K)$,
  $\widehat{X}_{22}=(\widetilde{X}_{22},\,\widetilde{X}_{23}(:,1:r_K+r_M-n)),\widehat{X}_{23}=\widetilde{X}_{23}(:,r_K+r_M-n+1:r_K)$

On the right hand side, the transformation reads, analogously,

$$\left(\begin{array}{c|ccc} \mathbb{I}_{r_M} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbb{I}_{r_K} & \mathbf{0} \\ \mathbf{0} & Q_X^* & \mathbf{0} & \mathbf{0} \\ \hline 0 & \mathbf{0} & \mathbf{0} & \mathbb{I}_{n-r_K} \end{array}\right) Y \left(\begin{array}{cc|c} & \mathbb{I}_{n-r_M} & \\ Z_X^* \begin{pmatrix} \mathbf{0} \\ \mathbb{I}_{r_M+r_K} \end{pmatrix} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & & \mathbb{I}_{n-r_K} \end{array}\right) \tag{3.50}$$

$$= \left(\begin{array}{cc||cc} \widetilde{Y}_{11} & \widehat{Y}_{12} & \widehat{Y}_{13} & \mathbf{0}_{r_M,n-r_K} \\ \widetilde{Y}_{21} & \widehat{Y}_{22} & \widehat{Y}_{23} & \mathbf{0}_{r_K,n-r_K} \\ \hline\hline \mathbf{0}_{n-r_M,r_M} & \mathbf{0}_{n-r_M,n-r_M} & \mathbf{0}_{n-r_M,r_K} & \mathbf{0}_{n-r_M,n-r_K} \\ \mathbf{0}_{n-r_K,r_M} & \mathbf{0}_{n-r_K,n-r_M} & \mathbf{0} & -\mathbb{I}_{n-r_K} \end{array}\right) \tag{3.51}$$

$$\text{where } \begin{pmatrix} \widetilde{Y}_{11} & \widetilde{Y}_{12} & \widetilde{Y}_{13} \\ \widetilde{Y}_{21} & \widetilde{Y}_{22} & \widetilde{Y}_{23} \end{pmatrix} = \begin{pmatrix} Y_{11} & Y_{12} & \mathbf{0}_{r_M,r_K} \\ \mathbf{0}_{r_k,r_M} & \mathbf{0}_{r_K,n-r_M} & -\mathbb{I}_{r_K} \end{pmatrix} Z_X^* \begin{pmatrix} \mathbf{0} & \mathbb{I}_{n-r_M} \\ \mathbb{I}_{r_M+r_K} & \mathbf{0} \end{pmatrix}, \tag{3.52}$$

and

- $\mathbf{n-r_M=r_K}$: $\widehat{Y}_{ij}=\widetilde{Y}_{ij}$, $i=1,2$, $j=2,3$;
- $\mathbf{n-r_M>r_K}$: $\widehat{Y}_{12}=\widetilde{Y}_{12}(:,1:r_K),\widehat{Y}_{13}=(\widetilde{Y}_{12}(:,r_K+1:n-r_M),\,\widetilde{Y}_{13})$,
  $\widehat{Y}_{22}=\widetilde{Y}_{22}(:,1:r_K),\widehat{Y}_{23}=(\widetilde{Y}_{22}(:,r_K+1:n-r_M),\,\widetilde{Y}_{23})$
- $\mathbf{n-r_M<r_K}$: $\widehat{Y}_{12}=(\widetilde{Y}_{12},\,\widetilde{Y}_{13}(:,1:r_K+r_M-n)),\widehat{Y}_{13}=\widetilde{Y}_{13}(:,r_K+r_M-n+1:r_K)$,
  $\widehat{Y}_{22}=(\widetilde{Y}_{22},\,\widetilde{Y}_{23}(:,1:r_K+r_M-n)),\widehat{Y}_{23}=\widetilde{Y}_{23}(:,r_K+r_M-n+1:r_K)$.

Hence, the equivalent pencil is

$$\left(\begin{array}{cc||cc} \widetilde{X}_{11} & \widehat{X}_{12} & \widehat{X}_{13} & X_{14} \\ \widetilde{X}_{21} & \widehat{X}_{22} & \widehat{X}_{23} & \mathbf{0} \\ \hline\hline \mathbf{0} & \mathbf{0} & R_X & \widetilde{X}_{24} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{cc||cc} \widetilde{Y}_{11} & \widehat{Y}_{12} & \widehat{Y}_{13} & \mathbf{0} \\ \widetilde{Y}_{21} & \widehat{Y}_{22} & \widehat{Y}_{23} & \mathbf{0} \\ \hline\hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbb{I} \end{array}\right),$$

and it immediately reveals that the original quadratic pencil is singular if $\det(R_X)=0$. Otherwise, we first identify the $n-r_K$ zero eigenvalues and the $n-r_M$ infinite ones, and the remaining ones are computed from the linear generalized eigenvalue problem of the $(r_M+r_K)\times(r_M+r_K)$ pencil

$$A-\lambda B \equiv \left(\begin{array}{cc} \widetilde{X}_{11} & \widehat{X}_{12} \\ \widetilde{X}_{21} & \widehat{X}_{22} \end{array}\right) - \lambda \left(\begin{array}{cc} \widetilde{Y}_{11} & \widehat{Y}_{12} \\ \widetilde{Y}_{21} & \widehat{Y}_{22} \end{array}\right). \tag{3.53}$$

Remark 3.4. In the important moment of computing the rank revealing decomposition (3.49), quadeig uses the same truncation strategy and with the same threshold used to infer the numerical ranks of M and K. In our opinion, this is fundamentally wrong strategy that may lead

to catastrophically wrong results. Our new algorithm will determine the numerical rank more carefully.

3.3.3 Eigenvectors in `quadeig`

Once the deflated linearized problem is solved, we need to transform the variables to the original problem, i.e. to assemble the requested eigenvectors of the original quadratic pencil. Before giving the formulas, let us first briefly review the process of computing eigenvalues and corresponding eigenvectors for quadratic eigenvalue problems that is solved via the linearization by the second companion form. The eigenvalues are the same, and the right eigenvectors z and left eigenvectors w of the linearization are of the forms, respectively,

$$z = \begin{pmatrix} z_1 \\ \hline\hline z_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} \lambda x \\ \hline\hline -Kx \end{pmatrix}, & \lambda \neq 0 \\ \begin{pmatrix} \lambda x \\ \hline\hline Cx \end{pmatrix}, & \lambda = 0 \end{cases}, \tag{3.54}$$

$$w = \begin{pmatrix} w_1 \\ \hline\hline w_2 \end{pmatrix} = \begin{pmatrix} \lambda y \\ \hline\hline y \end{pmatrix}, \tag{3.55}$$

where x, y are the right and left eigenvector of quadratic eigenvalue problem. From the first relation we see that, when the matrix K is nonsingular, we have two choices for a right eigenvector, namely z_1 and $K^{-1}z_2$. If K is singular (or highly ill-conditioned), we choose z_1. For a left eigenvector we have two choices in both cases. We can either choose w_1 or w_2. In `quadeig` the eigenvector with smallest backward error is chosen in the case of both the right and the left eigenvector.

However, the deflation process in `quadeig` introduces an orthogonal transformation which is used to transform linearization $C_2(\lambda)$ to generalized eigenvalue problem $QC_2(\lambda)V$. The eigenvalues of the transformed problem are the same, but the right eigenvector $\widetilde{z}$ and the left eigenvector $\widetilde{w}$ are transformed in the following way

$$\begin{pmatrix} \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \end{pmatrix} = \widetilde{z} = Vz, \tag{3.56}$$

$$\begin{pmatrix} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix} = \widetilde{w} = Q^* w, \tag{3.57}$$

where z, w are as in (3.54) and (3.55). So, the process of extraction of the eigenvectors goes from the bottom to the top. We first obtain the eigenvectors for the linearization $C_2(\lambda)$, and then

choose the eigenvector for the quadratic problem.

Now we provide explicit reconstruction formulas for the eigenvectors.

The right eigenvectors

The case: $\operatorname{rank}(M) = \operatorname{rank}(K) = n$**.** The matrix K is nonsingular, and we have two choices for the right eigenvector. Let $\widetilde{z}$ be the right eigenvector for the transformed GEP. The corresponding right eigenvector for $C_2(\lambda)$ is

$$z = \begin{pmatrix} z_1 \\ \hline\hline z_2 \end{pmatrix} = \begin{pmatrix} \Pi_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \begin{pmatrix} \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \end{pmatrix} = \begin{pmatrix} \Pi_M \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \end{pmatrix}.$$

Hence, the two candidates for the eigenvector x are $\Pi_M \widetilde{z}_1$ and $K^{-1}\widetilde{z}_2$. Now, the candidate with the smallest normwise backward error is chosen as the output.

The second case: $\operatorname{rank}(K) < \operatorname{rank}(M) = n$**.** The matrix K is singular, and $n - r_K$ zero eigenvalues have been deflated. The eigenvectors corresponding to those eigenvalues span the nullspace of the matrix K. The basis for the nullspace is computed via the orthogonal complement of the range of K^*, using the QR decomposition of the upper triangular matrix $\widehat{R}_K^*$:

$$P_K \widehat{R}_K^* = Q_{\widehat{R}_K^*} R_{\widehat{R}_K^*}.$$

The wanted vectors are the last $n - r_K$ columns of the orthogonal matrix $Q_{\widehat{R}_K^*}$.
The remaining eigenvalues and the corresponding eigenvectors $\widetilde{z} \in \mathbb{C}^{n+r_K}$ are computed from the $(n + r_K) \times (n + r_K)$ GEP (3.42). Partition $\widetilde{z}^T = \begin{pmatrix} \widetilde{z}_1^T & \widetilde{z}_2^T \end{pmatrix}$, where $\widetilde{z}_1 \in \mathbb{C}^n$ and $\widetilde{z}_2 \in \mathbb{C}^{r_K}$. The corresponding eigenvector for $C_2(\lambda)$ is

$$\begin{pmatrix} z_1 \\ \hline\hline z_2 \end{pmatrix} = \begin{pmatrix} \Pi_M & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \begin{pmatrix} \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \\ \mathbf{0}_{n-r_K} \end{pmatrix} = \begin{pmatrix} \Pi_M \widetilde{z}_1 \\ \hline\hline Q_K \begin{pmatrix} \widetilde{z}_2 \\ \mathbf{0}_{n-r_K} \end{pmatrix} \end{pmatrix}.$$

The only choice for the right eigenvector x is $\Pi_M \widetilde{z}_1$.

The third case: $\operatorname{rank}(K) \leq \operatorname{rank}(M) < n$**.** Both matrices M and K are singular, and $n - r_M$ infinite and $n - r_K$ zero eigenvalues have been deflated. The eigenvectors for the zero eigenvalue are obtained as in the previous case, whilst the eigenvectors for the infinite eigenvalue form the basis for the nullspace of the matrix M. The basis for the nullspace is obtained as the orthogonal complement of the range of M^* represented by the last $n - r_M$ columns of the orthogonal matrix $Q_{\widehat{R}_M^*}$

$$P_M \widehat{R}_M^* = Q_{\widehat{R}_M^*} R_{\widehat{R}_M^*}.$$

The remaining eigenvalues with the corresponding eigenvectors $\widetilde{z} \in \mathbb{C}^{r_K+r_M}$ are obtained from the $(r_K+r_M)\times(r_K+r_M)$ GEP (3.53). The corresponding eigenvector for $C_2(\lambda)$ is

$$\begin{pmatrix} z_1 \\ \hline\hline z_2 \end{pmatrix} = \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \begin{pmatrix} Z_X^* & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_{n-r_K} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbb{I}_{n-r_M} & \mathbf{0} \\ \mathbb{I}_{r_K+r_M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{I}_{n-r_K} \end{pmatrix} \begin{pmatrix} \widetilde{z} \\ \hline\hline \mathbf{0}_{n-r_M} \\ \mathbf{0}_{n-r_K} \end{pmatrix}$$
$$= \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \begin{pmatrix} Z_X^* \begin{pmatrix} \mathbf{0}_{n-r_M} \\ \widetilde{z} \end{pmatrix} \\ \mathbf{0}_{n-r_K} \end{pmatrix}.$$

The only candidate for the right eigenvector x is $Z_X^* \begin{pmatrix} \mathbf{0}_{n-r_M} \\ \widetilde{z} \end{pmatrix}(1:n) = Z_X^* \begin{pmatrix} \mathbf{0}_{n-r_M} \\ \widetilde{z}(1:r_M) \end{pmatrix}$.

The left eigenvectors

We now describe how to assemble the left eigenvectors of the quadratic pencil.

The first case: $\operatorname{rank}(M) = \operatorname{rank}(K) = n$**.** Let $\widetilde{w}$ be the left eigenvector for the transformed GEP $QC_2(\lambda)V$. The corresponding left eigenvector for the linearization $C_2(\lambda)$ is w

$$\begin{pmatrix} w_1 \\ \hline\hline w_2 \end{pmatrix} = \begin{pmatrix} Q_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \begin{pmatrix} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix} = \begin{pmatrix} Q_M\widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix}.$$

The two candidates for the left eigenvector y of the quadratic eigenvalue problem are $Q_M\widetilde{w}_1$ and $\widetilde{w}_2$. The next step is to compute corresponding normwise backward errors and choose the candidate with the smallest one as the output.

Te second case: $\operatorname{rank}(K) < \operatorname{rank}(M) = n$**.** The left eigenvectors for the zero eigenvalue are the last $n-r_K$ columns of the matrix Q_K. Let $\begin{pmatrix} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix} \in \mathbb{C}^{n+r_K}$ be the eigenvector for the deflated $(n+r_K)\times(n+r_K)$ pencil (3.42), where $\widetilde{w}_1 \in \mathbb{C}^n$ and $\widetilde{w}_2 \in \mathbb{C}^{r_K}$. The corresponding eigenvector for the $2n\times 2n$ pencil, before truncation satisfies

$$\left(\begin{array}{c|c|cc} \widetilde{w}_1^* & \widetilde{w}_2^* & \widetilde{w}_3^* \end{array} \right) \left(\left(\begin{array}{c|c|cc} X_{11} & X_{12} & X_{13} \\ \hline\hline X_{21} & \mathbf{0}_{r_K,r_K} & \mathbf{0}_{r_K,n-r_K} \\ \hline \mathbf{0}_{n-r_K,n} & \mathbf{0}_{n-r_K,r_K} & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c|cc} -R_M & \mathbf{0} \\ \hline\hline \mathbf{0} & -\mathbb{I}_{r_K} & \mathbf{0} \\ \hline & \mathbf{0} & -\mathbb{I}_{n-r_K} \end{array} \right) \right) =$$
$$\begin{pmatrix} \widetilde{w}_1^*X_{11} + \widetilde{w}_2^*X_{21} + \lambda\widetilde{w}_1^*R_M \\ \hline\hline \widetilde{w}_1^*X_{12} + \lambda\widetilde{w}_2^* \\ \widetilde{w}_1^*X_{13} + \lambda\widetilde{w}_3^* \end{pmatrix} = \mathbf{0},$$

therefore, $\widetilde{w}_3 = X_{13}^* \widetilde{w}_1/\lambda$. The vector z for $C_2(\lambda)$ is

$$\begin{pmatrix} w_1 \\ \hline\hline w_2 \end{pmatrix} = \begin{pmatrix} Q_M & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \begin{pmatrix} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \\ \widetilde{w}_3 \end{pmatrix} = \begin{pmatrix} Q_M \widetilde{w}_1 \\ \hline\hline Q_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_3 \end{pmatrix} \end{pmatrix}.$$

The candidates for the left eigenvector y for the quadratic eigenvalue problem are $Q_M \widetilde{w}_1$ and $Q_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_3 \end{pmatrix}$. Again, the eigenvector with the smaller normwise backward error is chosen as the approximation.

The third case: $\operatorname{rank}(K) \leq \operatorname{rank}(M) < n$**.** The left eigenvectors for the zero eigenvalue are the last $n - r_K$ columns of Q_K, and for the infinite eigenvalue are the last $n - r_M$ columns of Q_M. Let $\begin{pmatrix} \widetilde{w}_1 \\ \widetilde{w}_2 \end{pmatrix} \in \mathbb{C}^{r_K + r_M}$ be a left eigenvector for the truncated $(r_K + r_M) \times (r_K + r_M)$ pencil (3.53). The corresponding eigenvector for the pencil $QC_2(\lambda)V$ satisfies

$$\left(\begin{array}{cc||cc} \widetilde{w}_1^* & \widetilde{w}_2^* & \widetilde{w}_3^* & \widetilde{w}_4^* \end{array} \right) \left(\left(\begin{array}{c|c||c|c} \widetilde{X}_{11} & \widehat{X}_{12} & \widehat{X}_{13} & X_{14} \\ \hline \widetilde{X}_{21} & \widehat{X}_{22} & \widehat{X}_{23} & \mathbf{0} \\ \hline\hline \mathbf{0} & \mathbf{0} & R_X & \widetilde{X}_{24} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c||c|c} \widetilde{Y}_{11} & \widehat{Y}_{12} & \widehat{Y}_{13} & \mathbf{0} \\ \hline \widetilde{Y}_{21} & \widehat{Y}_{22} & \widehat{Y}_{23} & \mathbf{0} \\ \hline\hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbb{I} \end{array} \right) \right) =$$

$$= \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \hline\hline \widetilde{w}_1^* \widehat{X}_{13} + \widetilde{w}_2^* \widehat{X}_{23} + \widetilde{w}_3^* R_X - \lambda \widetilde{w}_1^* \widehat{Y}_{13} - \lambda \widetilde{w}_2^* \widehat{Y}_{23} \\ \widetilde{w}_1^* \widehat{X}_{14} + \widetilde{w}_3^* \widetilde{X}_{24} + \lambda \widetilde{w}_4^* \end{pmatrix} = \mathbf{0}.$$

The components $\widetilde{w}_3^*, \widetilde{w}_4^*$ are thus computed as

$$\begin{aligned} \widetilde{w}_3^* &= \left(\lambda \widetilde{w}_1^* \widehat{Y}_{13} + \lambda \widetilde{w}_2^* \widehat{Y}_{23} - \widetilde{w}_1^* \widehat{X}_{13} - \widetilde{w}_2^* \widehat{X}_{23} \right) R_X^{-1}, \\ \widetilde{w}_4^* &= \left(-\widetilde{w}_1^* \widehat{X}_{14} - \widetilde{w}_3^* \widetilde{X}_{24} \right) / \lambda. \end{aligned}$$

The left eigenvector for $C_2(\lambda)$ is

$$w = \begin{pmatrix} Q_M & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \begin{pmatrix} \mathbb{I}_{r_M} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_X & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_{r_K} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I}_{n-r_K} \end{pmatrix} \begin{pmatrix} \widetilde{w}_1 \\ \widetilde{w}_2 \\ \hline\hline \widetilde{w}_3 \\ \widetilde{w}_4 \end{pmatrix} = \begin{pmatrix} Q_M \begin{pmatrix} \widetilde{w}_1 \\ Q_X \widetilde{w}_3 \end{pmatrix} \\ \hline\hline Q_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_4 \end{pmatrix} \end{pmatrix},$$

and the candidates for the left eigenvector y are $Q_M \begin{pmatrix} \widetilde{w}_1 \\ Q_X \widetilde{w}_3 \end{pmatrix}$ and $Q_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_4 \end{pmatrix}$. The eigenvector with the smaller normwise backward error is chosen as the approximation.

3.4 Balancing by two-sided diagonal scalings

As mentioned in the introduction, along with having the coefficient matrices with unbalanced norms, their elements can be highly unbalanced too, for example as a result of particular choice of physical units and/or different physical nature of the involved coupled variables. This results in badly conditioned coefficient matrices, and backward error may simply wipe out small but physically relevant parameters.

In order to improve the deflation process, we propose balancing of the coefficients as in [9], where the coefficient matrices A, E, B of a descriptor linear time invariant dynamical system $E\dot{x} = Ax + Bu$ are balanced for more numerically robust reduction. It is a generalization of Ward's balancing algorithm [70] for two matrices. Bosner's algorithm produces two diagonal matrices D_l and D_r such that the range of magnitude orders of all elements in the scaled matrices D_lAD_r, D_lED_r and D_lB is small. We extend that algorithm so that the third matrix is also scaled from the right; in means that we go over to a new equivalent QEP:

$$\widehat{Q}(\lambda) = \lambda^2(D_lMD_r) + \lambda(D_lCD_r) + (D_lKD_r). \tag{3.58}$$

For a computed (e.g. right) eigenpair (λ, x) of (3.58), the corresponding eigenpair for the original problem is $(\lambda, D_r x)$.

3.4.1 The algorithm

Define the range of elements in a matrix as the ratio of the element of the largest and the one with smallest (nonzero) magnitude. The matrices D_l and D_r are computed so that the ranges of the elements in D_lMD_r, D_lCD_r and D_lKD_r are moderate. The main idea is that the exponents in the exponential notation of all nonzero elements in D_lMD_r, D_lCD_r and D_lKD_r should be close to zero. The diagonal matrices are defined as $D_l = \mathrm{diag}(10^{l_1}, \ldots, 10^{l_n})$ and $D_r = \mathrm{diag}(10^{r_1}, \ldots, 10^{r_n})$.

The problem of balancing is then equivalent to minimization problem

$$\min_{l,r\in\mathbb{R}^n} \varphi(l,r) =$$

$$\min_{l,r\in\mathbb{R}^n} \sum_{i=1}^{n} \left[\sum_{\substack{j=1 \\ m_{ij}\neq 0}}^{n} (l_i + r_j + \log|m_{ij}|)^2 + \sum_{\substack{j=1 \\ c_{ij}\neq 0}}^{n} (l_i + r_j + \log|c_{ij}|)^2 + \sum_{\substack{j=1 \\ k_{ij}\neq 0}}^{n} (l_i + r_j + \log|k_{ij}|)^2 \right], \tag{3.59}$$

where $l = \left(l_1, \ldots, l_n\right)$ and $r = \left(r_1, \ldots, r_n\right)$. This is a linear least square problem with the

following system $Lx = p$ of normal equations

$$L = \begin{pmatrix} F_1 & G \\ G^T & F_2 \end{pmatrix}, \; p = \begin{pmatrix} -c \\ -d \end{pmatrix}, \; x = \begin{pmatrix} l \\ r \end{pmatrix},$$

where $F_1 = \text{diag}(n_{r_1}, \ldots, n_{r_n}) \in \mathbb{R}^{n \times n}$ with

$$n_{r_i} = \sum_{\substack{j=1 \\ m_{ij} \neq 0}}^{n} 1 + \sum_{\substack{j=1 \\ c_{ij} \neq 0}}^{n} 1 + \sum_{\substack{j=1 \\ k_{ij} \neq 0}}^{n} 1$$

being the number of nonzero elements in the i-th rows of M, C and K, $F_2 = \text{diag}(n_{c_1}, \ldots, n_{c_n}) \in \mathbb{R}^{n \times n}$ with

$$n_{c_j} = \sum_{\substack{i=1 \\ m_{ij} \neq 0}}^{n} 1 + \sum_{\substack{i=1 \\ c_{ij} \neq 0}}^{n} 1 + \sum_{\substack{i=1 \\ k_{ij} \neq 0}}^{n} 1$$

being the total number of nonzero elements in the $j-th$ columns of M, C and K, $G \in \mathbb{R}^{n \times n}$ is the sum of incidence matrices of M, C and K:

$$g_{ij} = \left\{ \begin{array}{ll} 1, & \text{if } m_{ij} \neq 0 \\ 0, & \text{if } m_{ij} = 0 \end{array} \right\} + \left\{ \begin{array}{ll} 1, & \text{if } c_{ij} \neq 0 \\ 0, & \text{if } c_{ij} = 0 \end{array} \right\} + \left\{ \begin{array}{ll} 1, & \text{if } k_{ij} \neq 0 \\ 0, & \text{if } k_{ij} = 0 \end{array} \right\},$$

the vector $c \in \mathbb{R}^n$ has elements

$$c_i = \sum_{\substack{j=1 \\ m_{ij} \neq 0}}^{n} \log |m_{ij}| + \sum_{\substack{j=1 \\ c_{ij} \neq 0}}^{n} \log |c_{ij}| + \sum_{\substack{j=1 \\ k_{ij} \neq 0}}^{n} \log |k_{ij}|,$$

and the vector $d \in \mathbb{R}^n$ has elements

$$d_j = \sum_{\substack{i=1 \\ m_{ij} \neq 0}}^{n} \log |m_{ij}| + \sum_{\substack{i=1 \\ c_{ij} \neq 0}}^{n} \log |c_{ij}| + \sum_{\substack{i=1 \\ k_{ij} \neq 0}}^{n} \log |k_{ij}|.$$

The system is solved as in [9], using the preconditioned conjugate gradient method. In order to demonstrate the importance of balancing in computation of eigenvalues and eigenvectors, we will use the componentwise backward error (see Section 2.4) for the eigenpair (x, λ):

$$\omega_Q(x, \lambda) = \max_i \frac{|((\lambda^2 M + \lambda C + K)x)_i|}{((|\lambda|^2 |M| + |\lambda||C| + |K|)|x|)_i}. \tag{3.60}$$

Example 3.3. In Table 3.1, we show the maximum component-wise backward errors for non-zero finite eigenvalues for selected examples from the NLEVP library, computed with and without balancing. In Table 3.2, we show the ranges in M, C and K for these examples with and without balancing. It is clear from these results that there is significant improvement in $\max \omega_Q$ after the balancing took place. There is large improvement in the range of elements is in the

Table 3.1: Comparison of component-wise backward errors

	No balancing		Balancing	
Problem	$\min \omega_Q$	$\max \omega_Q$	$\min \omega_Q$	$\max \omega_Q$
damped_beam	3.4787e-015	3.2404e-009	7.6779e-016	8.0865e-013
power_plant	7.7532e-014	1.5799e-010	2.1702e-015	1.0789e-013
speaker_box	2.2373e-008	6.9832e-006	1.3051e-010	3.2287e-008

Table 3.2: Comparison of range of elements in M, C, K

	No balancing		
Problem	M	C	K
damped_beam	1.0400e+006	1.2000e+005	1
power_plant	4.3519e+007	1.6131e+009	4.3473e+009
speaker_box	1.3017e+010	3.5943e+010	3.7253e+017
	Balancing		
Problem	M	C	K
damped_beam	240	100	1
power_plant	74.7664	849.2321	761.9298
speaker_box	1.3017e+008	3.5943e+008	2.2146e+017

first and the second example, which is followed by the smaller maximal component-wise error. However, in the third example balancing did not made significant improvement in the range of matrices, especially for matrix K. Nevertheless, the component-wise backward error is improved by two orders of magnitude.

We strongly believe that this balancing at the matrix elements level is an important preprocessing technique that will prove its value in the design of iterative methods as well. It is a subject of our ongoing and the future work.

3.5 Improved deflation process. New algorihm – `KVADeig`

After preprocessing by parameter scaling and diagonal balancing, both optional, the task is to detect and remove (deflate) the zero and the infinite eigenvalues. We have already discussed the importance of such deflation. From the design of `quadeig`, it is clear that it cannot guarantee removal of all zeros/infinites from the spectrum; in fact it can only deflate one Jordan block of these eigenvalues.

In this section, we go through the details of this initial deflation, and we propose to supplement it with additional steps. To motivate the need for the improvement of the deflation process, we use an example from the NLEVP collection [5].

3.5.1 A case study example

This is a 10×10 quadratic eigenvalue problem for the pencil $\mathscr{J}(\lambda) = \lambda^2 M + \lambda C + K$, whose real eigenvalues and the corresponding eigenvectors give the intersection points of a sphere, a cylinder and a plane.

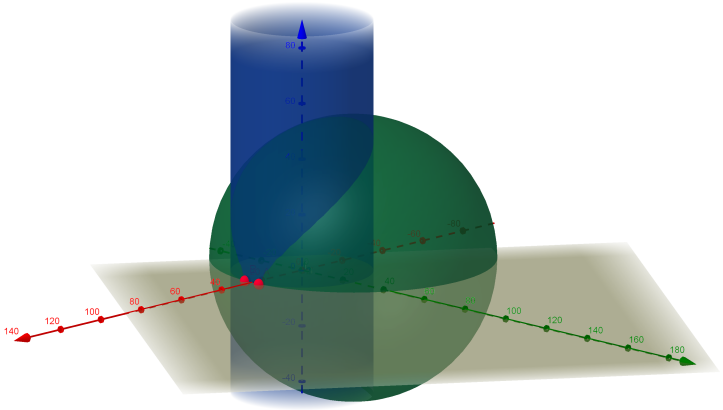


Figure 3.3: Intersection points of a sphere, a cylinder and a plane (`intersection` in NLEVP)

Although of small dimension and very simple structure, this example is an excellent illustration of difficulties in solving nonlinear eigenvalue problems.

It has been shown in [53], [51] that this problem has only four finite eigenvalues: two real ones and a complex conjugate pair. We take this example as a case study and compute the spectrum by several mathematically equivalent methods; all computation is done in Matlab 8.5.0.197613 (R2015a). If one plainly applies the QZ to a linearization of $\mathscr{J}(\lambda)$, such as the first or the second companion form with the Fan-Lin-Van Dooren scaling, the spectrum appears as

$$\mathscr{C}_1(\lambda) : \begin{cases} \lambda_1 = \texttt{2.476851749893558e+01} \\ \lambda_2 = \texttt{2.476851768196165e+01} \\ \lambda_3 = \texttt{-5.581844429198920e+08 - 1.628033679447590e+09i} \\ \lambda_4 = \texttt{-5.581844429198920e+08 + 1.628033679447590e+09i} \\ \lambda_5 = \texttt{2.570601782117493e+18} \\ \lambda_6 = \ldots = \lambda_{14} = \texttt{Inf},\ \lambda_{15} = \ldots = \lambda_{20} = \texttt{-Inf}, \end{cases} \tag{3.61}$$

$$\mathscr{C}_2(\lambda) : \begin{cases} \lambda_1 = \texttt{2.476851749893561e+01} \\ \lambda_2 = \texttt{2.476851768196167e+01} \\ \lambda_3 = \texttt{-2.653302084597818e+09} \\ \lambda_4 = \ldots = \lambda_{17} = \texttt{Inf},\ \lambda_{18} = \ldots = \lambda_{20} = \texttt{-Inf}. \end{cases} \tag{3.62}$$

If we use the same method, but with the reversed pencil $\mu^2 K + \mu C + M$, $(\lambda = 1/\mu)$ then from

the first companion form QZ has computed 12 finite eigenvalues (8 real and 2 complex conjugate pairs), and from the second 10 (6 real and 2 complex conjugate pairs).

If we run the Matlab's solver `polyeig()`, we obtain

$$\texttt{polyeig}(\mathscr{Y}(\lambda)) \; : \; \begin{cases} \lambda_1 = \texttt{2.476851768196161e+01} \\ \lambda_2 = \texttt{2.476851749893561e+01} \\ \lambda_3 = \texttt{1.426603361688555e+08} \\ \lambda_4 = \texttt{-1.353812777123886e+08} \\ \lambda_5 = \ldots = \lambda_{18} = \texttt{Inf},\ \lambda_{19} = \lambda_{20} = \texttt{-Inf}, \end{cases} \tag{3.63}$$

and if we scale the coefficient matrices then

$$\texttt{polyeig}(\mathscr{Y}_{scaled}(\lambda)) \; : \; \begin{cases} \lambda_1 = \texttt{2.476851768196165e+01} \\ \lambda_2 = \texttt{2.476851749893559e+01} \\ \lambda_3 = \texttt{-3.020295324523709e+08 + 1.229442619245432e+09i} \\ \lambda_4 = \texttt{-3.020295324523709e+08 - 1.229442619245432e+09i} \\ \lambda_5 = \ldots = \lambda_{18} = \texttt{Inf},\ \lambda_{19} = \lambda_{20} = \texttt{-Inf}. \end{cases} \tag{3.64}$$

Almost perfect match in λ_1 and λ_2 is reassuring, but there is an obvious disagreement in the total number and the nature (real or complex) of finite eigenvalues. With an earlier version of Matlab, the results that correspond to (3.61), (3.62), (3.63), and (3.64) coincide in the numbers of finite eigenvalues; λ_1 and λ_2 are close up to machine precision, but the remaining computed finite eigenvalues are substantially different.

The rank of the matrix M is exactly 3, and it will be correctly determined numerically due to a particularly simple sparsity structure of M. The matrix K is also sparse with $\kappa_2(K) \approx$ `4.09+03`, so there is no numerical rank issue. In this situation, a preprocessing procedure such as in `quadeig` will reverse the pencil and deflate 7 zero eigenvalues (infinite eigenvalues of the original problem) at the very beginning. The remaining eigenvalues are then computed (e.g. using `quadeig`) as[1]

$$\begin{array}{l|l} \lambda_1 = \texttt{2.4769e+001} & \lambda_8 = \texttt{-1.4660e+007 - 6.9064e+006i} \\ \lambda_2 = \texttt{2.4769e+001} & \lambda_9 = \texttt{-1.4660e+007 + 6.9064e+006i} \\ \lambda_3 = \texttt{1.1194e+006} & \lambda_{10} = \texttt{-4.5822e+015} \\ \lambda_4 = \texttt{-5.5674e+005 -1.0143e+006i} & \lambda_{11} = \texttt{-3.9134e+015} \\ \lambda_5 = \texttt{-5.5674e+005 + 1.0143e+006i} & \lambda_{12} = \texttt{-2.3047e+019} \\ \lambda_6 = \texttt{1.4679e+007 - 1.9395e+007i} & \lambda_{13} = \texttt{3.0862e+020} \\ \lambda_7 = \texttt{1.4679e+007 + 1.9395e+007i} & \end{array} \tag{3.65}$$

After the deflation of the 7 zero eigenvalues, in the thus obtained linearization $A - \lambda B$, the rank of the matrix A is 7, and it can be determined exactly because of sparsity (A has 6 zero columns,

[1] Here, to save the space, we display the computed values only to five digits.

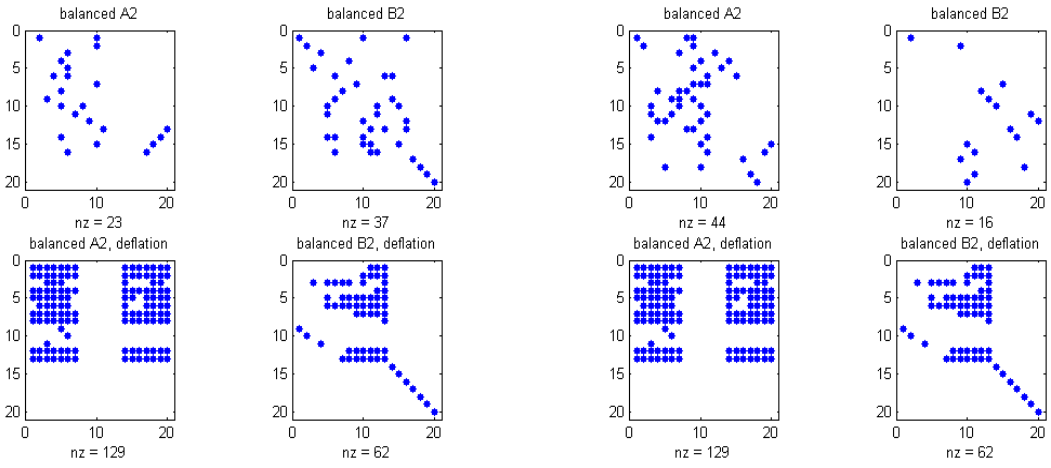
and the remaining 7 ones build a well conditioned 13×7 submatrix of A). The matrix B is well conditioned 5.550831520847275e+003. This means that at least 6 more zero eigenvalues are present in the reversed problem (infinities in the original problem); those are not detected by the QZ algorithm running on $A - \lambda B$.

Remark 3.5. It should be noted that the successful removal of many infinite eigenvalues in (3.61), (3.62), (3.63), and (3.64) is due to the sparsity that is successfully exploited by the preprocessing to the QZ algorithm. Recall, before the reduction to the triangular - Hessenberg form the matrices are scaled and permuted, as described in [70] in order to get equivalent pencil $\widehat{A} - \lambda \widehat{B}$ of form

$$\widehat{A} = \begin{pmatrix} A_{[11]} & A_{[12]}D_2G_2 & A_{[13]} \\ \mathbf{0} & G_1D_1A_{[22]}D_2G_2 & G_1D_1A_{[23]} \\ \mathbf{0} & \mathbf{0} & A_{[33]} \end{pmatrix}, \quad \widehat{B} = \begin{pmatrix} B_{[11]} & B_{[12]}D_2G_2 & B_{[13]} \\ \mathbf{0} & G_1D_1B_{[22]}D_2G_2 & G_1D_1B_{[23]} \\ \mathbf{0} & q0 & B_{[33]} \end{pmatrix},$$

where $A_{[11]}, A_{[33]}, B_{[11]}, B_{[33]}$ are upper triangular, and

$$P_1AP_2 = \begin{pmatrix} A_{[11]} & A_{[12]} & A_{[13]} \\ \mathbf{0} & A_{[22]} & A_{[23]} \\ \mathbf{0} & \mathbf{0} & A_{[33]} \end{pmatrix}, \quad P_1BP_2 = \begin{pmatrix} B_{[11]} & B_{[12]} & B_{[13]} \\ \mathbf{0} & B_{[22]} & B_{[23]} \\ \mathbf{0} & \mathbf{0} & B_{[33]} \end{pmatrix}.$$



Original problem. First row: balanced second companion form linearization pencil. Second row: Balanced truncated pencil after the deflation process.

Reversed problem. First row: balanced second companion form linearization pencil. Second row: Balanced truncated pencil after the deflation process.

Figure 3.4: Sparsity structure of the linearization pencil before and after deflation

The matrices D_1 and D_2 are computed so that the elements of $D_1A_{[22]}D_2$ and $D_1B_{[22]}D_2$ have magnitudes as close to one. G_2 is permutation matrix determined so that the ratio of the

column norms of $D_1 A_{[22]} D_2 G_2$ to the corresponding columns norms of $D_1 B_{[22]} D_2 G_2$ appear in decreasing order. G_1 is determined so that the ratios of the row norms of $G_1 D_1 A_{[22]} D_2 G_2$ to those of $G_1 D_1 B_{[22]} D_2 G_2$ appear in decreasing order. On the other hand, the transformation (3.41) (designed to expose the zero eigenvalues of the reversed pencil, that correspond to the null space of M) has introduced fill-in. This is illustrated in Figure 3.4.

3.5.2 Deflation process revisited

Recall the deflation process in the `quadeig` algorithm in the case of one singular matrix. There, the QR factorization of the matrix M is used to reduce the matrix B to upper triangular form. However, if we define the transformation matrices to maintain the identity in the upper right block of the matrix A in the linearization pencil $A-\lambda B$, we get:

$$\begin{aligned}\mathbf{P}_1(A-\lambda B)\mathbf{Q}_1 &= \left(\begin{array}{c|c} Q_K^* & \mathbf{0} \\ \hline \mathbf{0} & Q_K^* \end{array}\right)\left(\begin{pmatrix} C & -\mathbb{I}_n \\ K & \mathbf{0}\end{pmatrix} - \lambda\begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n\end{pmatrix}\right)\left(\begin{array}{c|c} \mathbb{I}_n & \mathbf{0} \\ \hline \mathbf{0} & Q_K \end{array}\right) \\ &= \left(\begin{array}{c|c} Q_K^* C & -\mathbb{I}_n \\ \hline \widehat{R}_K P_K^T & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0}\end{array}\right) - \lambda\left(\begin{array}{c|c} -Q_K^* M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array}\right).\end{aligned}$$

Note that $\operatorname{rank}(A) = n + \operatorname{rank}(K)$, so A and K have null spaces of equal dimensions. In essence, multiplication from the left with $Q_K^* \oplus Q_K^*$ (or with $Q_M^* \oplus Q_K^*$, or $\mathbb{I}_n \oplus Q_K^*$) is a rank revealing transformation of A. We now truncate the $s_1 = n - r_K$ copies of the eigenvalue $\lambda = 0$ and proceed with the truncated $(n+r_K)\times(n+r_K)$ pencil

$$A_{22} - \lambda B_{22} = \left(\begin{array}{c|c} Q_{K,1}^* C & -\mathbb{I}_{r_K} \\ \hline Q_{K,2}^* C & \mathbf{0} \\ \hline \widehat{R}_K P_K^T & \mathbf{0}\end{array}\right) - \lambda\left(\begin{array}{c|c} Q_K^* M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_{r_K} \end{array}\right). \tag{3.66}$$

Note that using the definition (3.66) of $A_{22} - \lambda B_{22}$ in (3.66) yields

$$\mathbf{P}_1(A-\lambda B)\mathbf{Q}_1 = \begin{pmatrix} A_{22}-\lambda B_{22} & \blacksquare \\ \mathbf{0} & A_{11}-\lambda B_{11}\end{pmatrix},\ A_{11} = \mathbf{0}_{n-r_K},\ B_{11} = -\mathbb{I}_{n-r_K}. \tag{3.67}$$

With $A_{11} := A$ and $B_{11} := B$, this procedure can be understood as the first step of the Van Dooren's algorithm (actually its transposed version, see §3.2) for the determination of the elementary divisors of the eigenvalue zero.

Using this modified transformation defined by $\mathbf{P}_1$ and $\mathbf{Q}_1$, for a rank revealing factorization of A_{22} it suffices to compute the rank revealing QR factorization of its $n \times n$ submatrix $A_{22}(r_K+1:$

$n+r_K, 1:n$),

$$\left(\begin{array}{c} Q_{K,2}^* C \\ \hline \widehat{R}_K P_K^T \end{array}\right) P_{A_{22}} = Q_{A_{22}} R_{A_{22}}. \tag{3.68}$$

This can be used to transform the pencil $A_{22} - \lambda B_{22}$ to

$$\widehat{\mathbf{P}}_2(A_{22} - \lambda B_{22}) = \left(\begin{array}{c|c} Q_{K,1}^* C & -\mathbb{I}_{r_K} \\ \hline R_{A_{22}} P_{A_{22}}^T & \mathbf{0} \end{array}\right) - \lambda \widehat{\mathbf{P}}_2 \left(\begin{array}{c|c} Q_K^* M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_{r_K} \end{array}\right), \ \widehat{\mathbf{P}}_2 = \left(\begin{array}{c|c} \mathbb{I}_{r_K} & \mathbf{0} \\ \hline \mathbf{0} & Q_{A_{22}}^* \end{array}\right). \tag{3.69}$$

If the factorization (3.68) shows no rank deficiency, there are no zeros in the spectrum of $A_{22} - \lambda B_{22}$. Otherwise, $A_{22}(r_K + 1 : n + r_K, 1 : n)$ is rank deficient; assume its rank to be r_{22}, $r_{22} < n$, and $s_2 = n + r_K - r_{22}$. Then

$$R_{A_{22}} = \begin{pmatrix} \widehat{R}_{A_{22}} \\ \mathbf{0}_{n-r_{22},n} \end{pmatrix}, \ \widehat{R}_{A_{22}} \in \mathbb{C}^{r_{22} \times n},$$

$$\widehat{\mathbf{P}}_2 A_{22} = \left(\begin{array}{c|c} Q_{K,1}^* C & -\mathbb{I}_{r_K} \\ \hline \widehat{R}_{A_{22}} P_{A_{22}}^T & \mathbf{0} \\ \hline \mathbf{0}_{n-r_{22},n} & \mathbf{0}_{n-r_{22},r_K} \end{array}\right), \ \widehat{\mathbf{P}}_2 B_{22} = \left(\begin{array}{c|c} Q_{K,1}^* M & \mathbf{0}_{r_K} \\ \hline \blacksquare & \blacktriangle \\ \hline \square & \triangle \end{array}\right). \tag{3.70}$$

The next step is to transform matrix $\widehat{\mathbf{P}}_2 B_{22}$ so that the block $\square$ is zero. This is done by computing the complete orthogonal decomposition (for the analysis see §3.1.3)

$$\widehat{\mathbf{P}}_2 B_{22} = U_B R_B V_B^*. \tag{3.71}$$

The column rank of $\widehat{\mathbf{P}}_2 B_{22}$ is s_2 (otherwise, the matrix pencil is singular), and $\widehat{\mathbf{P}}_2 B_{22} V_B = \begin{pmatrix} B_{22} & 0 \end{pmatrix}$ (here we abuse notation for B_{22}, for the sake of simplicity, as in Algorithm 3.5.1). Let P_B represent the permutation of the first s_2 and the last $n - s_2$ column blocks. The wanted structure is now obtained by multiplying the pencil (3.70) from the right with $V_B P_V$:

$$\widehat{\mathbf{P}}_2 A_{22} V_B P_B - \lambda \widehat{\mathbf{P}}_2 B_{22} V_B P_B = \begin{pmatrix} A_{33} - \lambda B_{33} & \spadesuit \\ \mathbf{0} & -\lambda B_{22} \end{pmatrix}. \tag{3.72}$$

The ext proposition shows that the existence of a second Jordan block for the zero eigenvalue depends on the relationship between the matrices K and C.

Proposition 3.4. *Assume that the matrix K from the quadratic pencil $\lambda^2 M + \lambda C + K$ has rank* $\operatorname{rank}(K) = r_K < n$. *There exists more than one Jordan block for the eigenvalue zero if*

$$(\ker(C) \cup \mathscr{X}) \cap \ker(K) \neq \{\mathbf{0}\}, \ \ \mathscr{X} = \{y \in \mathbb{C}^n : Cy \in \operatorname{Im}(K)\}.$$

Analogously, if the matrix M has rank $\operatorname{rank}(M) = r_M < n$, *there is more than one Jordan block*

for the infinite eigenvalue if

$$(\ker(C) \cup \mathscr{Y}) \cap \ker(M) \neq \{\mathbf{0}\}, \quad \mathscr{Y} = \{y \in \mathbb{C}^n : Cy \in \operatorname{Im}(M)\}$$

Proof. From Theorem 3.3 we know that the partial multiplicities, and thus the dimensions of the Jordan blocks for a quadratic eigenvalue problem can be obtained using Algorithm 3.2.1 for a corresponding strong linearization. If we use the second companion form, the very first step of the deflation yields the pencil (3.66). Now, if $\widetilde{A}_{22}$ is singular, we will have another Jordan block for the eigenvalue zero. The rank of the matrix $\widetilde{A}_{22}$ can be determined by the rank of the matrix $\begin{pmatrix} Q_{K,2}^* C \\ \hline \widehat{R}_K P_K^T \end{pmatrix}$. This matrix is rank deficient if its kernel is nontrivial, that is if $\ker \begin{pmatrix} Q_{K,2}^* C \\ \hline \widehat{R}_K P_K^T \end{pmatrix} = \ker(Q_{K,2} C) \cap \ker(\widehat{R}_K P_K^T) \neq \{\mathbf{0}\}$. The matrix $Q_{K,2}$ represents the basis for $\ker(K^*)$, and thus

$$\ker \begin{pmatrix} Q_{K,2}^* C \\ \widehat{R}_K P_K^T \end{pmatrix} = (\ker(C) \cup \mathscr{X}) \cap \ker(K),$$

where $\mathscr{X} = \{y \in \mathbb{C}^n : Cy = z,\ z \in \operatorname{Im}(K)\}$. □

From these two steps we see that, for this choice of linearization, the upper triangular form for (3.27) would be more fitting. This is why we propose the modification of Algorithm 3.2.1 using the rank revealing QR factorization, see §3.5.3 below.

Backward error

The following proposition states the backward stability for the first step od the deflation process (3.66) as in Subsection 3.3.2.

Proposition 3.5. *Let*

$$\widetilde{A} - \lambda \widetilde{B} = \left(\begin{array}{c|c} \widetilde{X}_{11} & -\mathbb{I}_{\widetilde{r}_K} \\ \hline \widetilde{R}_K \widetilde{\Pi}_K^T & \mathbf{0}_{\widetilde{r}_K, \widetilde{r}_K} \end{array} \right) - \lambda \left(\begin{array}{c|c} -\widetilde{Y}_{11} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_{\widetilde{r}_K} \end{array} \right)$$

be the computed linearization (3.66). Then it corresponds to an exact reduced linearization of a quadratic pencil $\lambda^2(M+\delta M)+\lambda(C+\delta C)+(K+\delta K+\Delta K)$, where, for all $i = 1, \ldots, n$,

$$\|\delta M(:,i)\|_2 \leq \varepsilon_M \|M(:,i)\|_2, \ \ \|\delta C(:,i)\|_2 \leq \varepsilon_C \|C(:,i)\|_2, \ \ \|\delta K(:,i)\|_2 \leq \varepsilon_{qr} \|K(:,i)\|_2; \quad (3.73)$$

and the truncation error is

$$\max_{j=1:n-k} \|(\Delta K)\widetilde{\Pi}_K(:,k+j)\|_2 \leq \tau \min_{i=1:k} \|(K+\delta K)\widetilde{\Pi}_K(:,i)\|_2; \ \ (\Delta K)\widetilde{\Pi}_K(:,1:k) = \mathbf{0}_{n,k}. \quad (3.74)$$

Proof. (i) Let $\widetilde{P}_K, \widetilde{Q}_K, \widetilde{R}_K$ be the computed factors of QR decomposition of K, i.e. $(K+\delta K)\widetilde{P}_K = \widehat{Q}_K \begin{pmatrix} \widetilde{R}_K \\ \mathbf{0} \end{pmatrix}$, $\|\widetilde{Q}_K - \widehat{Q}_K\|_F \leq \varepsilon_2$. It holds that $\widetilde{X}_{11} = computed(\widetilde{Q}_K^* C) = \widehat{Q}_K^*(C+\delta C)$. To estimate δC, we start with the fact that

$$computed(\widetilde{Q}_K^* C) = \widetilde{Q}_K^* C + \mathfrak{G}_C, \;\; |\mathfrak{G}_C| \leq \varepsilon_* |\widetilde{Q}_K^*||C|, \;\; 0 \leq \varepsilon_* \leq 2n\mathbf{u}.$$

Since $\widetilde{Q}_K = (\mathbb{I} + \mathfrak{E}_C)\widehat{Q}_K$, $\|\mathfrak{E}_C\|_2 \leq \varepsilon_{qr}$, we have

$$computed(\widetilde{Q}_K^* C) = \widehat{Q}_K^*(\mathbb{I}+\mathfrak{E}_C^*)C + \mathfrak{G}_C = \widehat{Q}_K^*(C + \underbrace{\mathfrak{E}_C^* C + \widehat{Q}_K \mathfrak{G}_C}_{:=\delta C}) \equiv \widehat{Q}_K^*(C+\delta C),$$

with column-wise estimates $\|\delta C(:,i)\|_2 \leq (\|\mathfrak{E}_C^*\|_2 + \varepsilon_* n(1+\|\mathfrak{E}_C^*\|_2))\|C(:,i)\|_2$, and (3.73) follows with $\varepsilon_C = (\varepsilon_{qr} + \varepsilon_* n(1+\varepsilon_{qr}))$ (derived as in Proposition 3.3).

(ii) By the same reasoning we get $\widetilde{Y}_{11} = \widehat{Q}_K(M+\delta M)$, where $\|\delta M(:,i)\|_2 \leq \varepsilon_M \|M(:,i)\|_2$, and $\varepsilon_M = (\varepsilon_{qr} + \varepsilon_* n(1+\varepsilon_{qr}))$.

(iii) Note that in this moment the backward error in K contains both the floating point error δK and the truncation error ΔK analogous to (3.38), i.e. $(K+\delta K+\Delta K)\widetilde{\Pi}_K = \widehat{Q}_K \widetilde{R}_K$. If we set $\Delta_\Sigma K = \delta K + \Delta K$, then we can represent the computed linearization as

$$\begin{pmatrix} \widehat{Q}_K^* & \mathbf{0} \\ \mathbf{0} & \widehat{Q}_K^* \end{pmatrix} \left\{ \begin{pmatrix} C+\delta C & -\mathbb{I}_n \\ K+\Delta_\Sigma K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M-\delta M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & \widehat{Q}_K \end{pmatrix}$$

$$= \left(\begin{array}{c|c|c} \widetilde{X}_{11} & -\mathbb{I}_{\widetilde{r}_K} & \mathbf{0}_{n-\widetilde{r}_K} \\ \hline \widetilde{R}_K \widetilde{\Pi}_K^T & \mathbf{0}_{\widetilde{r}_K,\widetilde{r}_K} & \mathbf{0}_{\widetilde{r}_K,n-\widetilde{r}_K} \\ \hline \mathbf{0}_{n-\widetilde{r}_K,n} & \mathbf{0}_{n-\widetilde{r}_K,\widetilde{r}_K} & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c|c} -\widetilde{Y}_{11} & \mathbf{0} & \\ \hline & -\mathbb{I}_{\widetilde{r}_K} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & -\mathbb{I}_{n-\widetilde{r}_K} \end{array} \right).$$

□

It is hard to say something about the backward stability of the second step of the deflation process in terms of the original coefficient matrices M, C and K since the transformation (3.70) destroys the block structure. However, we can say something about the rank revealing QR factorization for the block matrix

$$\left(\frac{\widehat{Q}_{K,2}^*(C+\delta C)}{\widetilde{R}_K \widetilde{\Pi}_K^T} \right) \Pi_A = Q_A R_A,$$

which is used to determine whether there are more Jordan blocks for the quadratic eigenvalue problem.

For the computed factors $\widetilde{\Pi}_A, \widetilde{Q}_A, \widetilde{R}_A$ it holds that

$$\left[\left(\frac{\widehat{Q}_{K,2}^*(C+\delta C)}{\widetilde{R}_K \widetilde{\Pi}_K^T} \right) + \begin{pmatrix} \mathfrak{C} \\ \mathfrak{K} \end{pmatrix} \right] \widetilde{\Pi}_A = \widehat{Q}_A \widetilde{R}_A,$$

where

$$\left\| \begin{pmatrix} \mathfrak{C} \\ \mathfrak{K} \end{pmatrix} (:,i) \right\|_2 \leq \varepsilon_{qr} \left\| \begin{pmatrix} \widehat{Q}_{K,2}^*(C+\delta C) \\ \widetilde{R}_K \widetilde{\Pi}_K^T \end{pmatrix} (:,i) \right\|_2 .$$

However, the norm of block vector $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is $\|x\|_2 = \sqrt{\|x_1\|_2^2 + \|x_2\|_2^2}$, which means that above inequality holds for both $\|\mathfrak{C}(:,i)\|_2$ and $\|\mathfrak{K}(:,i)\|_2$. On the other hand, we can estimate $\|x\|_2 \leq \sqrt{2}\max(\|x_1\|_2, \|x_2\|_2)$. Using these bounds we get

$$\|\mathfrak{C}(:,i)\|_2 \leq \varepsilon_{qr}\sqrt{2}\max\left(\|\widehat{Q}_{K,2}^*(C+\delta C)(:,i)\|_2, \|\widetilde{R}_K\widetilde{\Pi}_K^T(:,i)\|_2 \right),$$
$$\|\mathfrak{K}(:,i)\|_2 \leq \varepsilon_{qr}\sqrt{2}\max\left(\|\widehat{Q}_{K,2}^*(C+\delta C)(:,i)\|_2, \|\widetilde{R}_K\widetilde{\Pi}_K^T(:,i)\|_2 \right).$$

Moreover, if $C = Q_C R_C$ is the exact QR factorization of the matrix C, we have

$$\begin{aligned} \|\widehat{Q}_{K,2}^*(C+\delta C)(:,i)\|_2 &\leq \|\widehat{Q}_{K,2}^* C(:,i)\|_2 + \|\widehat{Q}_{K,2}^*\delta C(:,i)\|_2 \leq (1+\varepsilon_C)\|\widehat{Q}_{K,2}^* C(:,i)\|_2, \\ &= (1+\varepsilon_C)\|\widehat{Q}_{K,2}^* Q_C R_C(:,i)\| \leq (1+\varepsilon_C)\|\widehat{Q}_{K,2}^* Q_C\|_2 \|C(:,i)\|_2 \\ &= (1+\varepsilon_C)\cos\sphericalangle(\mathrm{Ker}(K), \mathrm{Im}(C))\|C(:,i)\|_2, \end{aligned}$$

and $\|\widetilde{R}_K\widetilde{\Pi}_K^T(:,i)\|_2 \leq (1+\varepsilon_{qr})\|K(:,i)\|_2$. Altogether we have

$$\|\mathfrak{C}(:,i)\|_2 \leq \varepsilon_{qr}\sqrt{2}\frac{\max\left((1+\varepsilon_C)\cos\sphericalangle(\mathrm{Ker}(K),\mathrm{Im}(C))\|C(:,i)\|_2, (1+\varepsilon_{qr})\|K(:,i)\|_2\right)}{\|C(:,i)\|_2}\|C(:,i)\|_2,$$
$$\|\mathfrak{K}(:,i)\|_2 \leq \varepsilon_{qr}\sqrt{2}\frac{\max\left((1+\varepsilon_C)\cos\sphericalangle(\mathrm{Ker}(K),\mathrm{Im}(C))\|C(:,i)\|_2, (1+\varepsilon_{qr})\|K(:,i)\|_2\right)}{\|K(:,i)\|_2}\|K(:,i)\|_2,$$

i.e.

$$\frac{\|\mathfrak{C}(:,i)\|_2}{\|C(:,i)\|_2} \leq \varepsilon_{qr}\sqrt{2}\max\left((1+\varepsilon_C)\cos\sphericalangle(\mathrm{Ker}(K),\mathrm{Im}(C)), (1+\varepsilon_{qr})\frac{\|K(:,i)\|_2}{\|C(:,i)\|_2} \right) \tag{3.75}$$
$$\frac{\|\mathfrak{K}(:,i)\|_2}{\|K(:,i)\|_2} \leq \varepsilon_{qr}\sqrt{2}\max\left((1+\varepsilon_C)\cos\sphericalangle(\mathrm{Ker}(K),\mathrm{Im}(C))\frac{\|C(:,i)\|_2}{\|K(:,i)\|_2}, (1+\varepsilon_{qr}) \right). \tag{3.76}$$

Notice that the bounds (3.75,3.76) can blow up if there is a large difference in the norms of columns $K(:,i), C(:,i)$ of the coefficient matrices K and C. This once more shows the importance of scaling and balancing.

3.5.3 Computing the Kronecker's Canonical form using rank revealing QR factorization

From the previous section, we know that, in order to exploit the structure of the second companion form linearization as much as possible, it is more convenient to deflate the (zero) eigenvalue by the transformations which lead to upper triangular forms (3.67),(3.72). This is

done by using the rank revealing QR factorization of the current matrix $A_{i,i}$ from the linearization pencil, instead of using the SVD. In this subsection, we derive such an algorithm. We will describe the first step in detail, and then formulate the complete algorithm.
Let $A,B \in \mathbb{C}^{n\times n}$. Denote by n_i the size of the current working matrix in step i, and s_i the defect of the working matrix in step i. Consider the following procedure.

1. Compute the rank revealing factorization of $A_{1,1} = A$

$$A_{1,1}P_A = Q_A R_A, \tag{3.77}$$

and denote $s_1 = n_1 - \operatorname{rank}(A) = n - \operatorname{rank}(A)$. Now, $Q_A^* A_{1,1} = \left(\begin{array}{c} A_2 \\ \hline \mathbf{0}_{s_1\times n} \end{array}\right)$, where A_2 is of full row rank $n-s_1$. Partition $Q_A^* B = \left(\begin{array}{c} B_2 \\ \hline B_1 \end{array}\right)$ in compatible manner. Multiply the pencil $(A-\lambda B)$ by Q_A^* on the left to get

$$Q_A^*(A-\lambda B) = \left(\begin{array}{c} A_2 - \lambda B_2 \\ \hline \lambda B_1 \end{array}\right). \tag{3.78}$$

2. Compute the complete orthogonal decomposition of B_1

$$B_1 = U_B R_B V_B^*. \tag{3.79}$$

The column rank of B_1 is s_1, if the matrix pencil is regular, and $B_1 V_B = \left(\begin{array}{c|c} B_{1,1} & \mathbf{0}_{s_1,n-s_1} \end{array}\right)$, where $B_{1,1}$ is upper triangular. Multiply the pencil (3.78) by V_B on the right to get

$$Q_A^*(A-\lambda B)V_B = \left(\begin{array}{c|c} A_{1,2} - \lambda B_{1,2} & A_{2,2} - \lambda B_{2,2} \\ \hline \lambda B_{1,1} & \mathbf{0} \end{array}\right).$$

3. Let P_B be the permutation matrix for permuting the first s_1 and the last $n-s_1$ columns. Define $P_1 = Q_A^*$ and $Q_1 = V_B P_B$. The first Jordan block for the eigenvalue 0 is deflated by the following orthogonal transformation:

$$P_1(A-\lambda B)Q_1 = \left(\begin{array}{c|c} A_{2,2} - \lambda B_{2,2} & A_{1,2} - \lambda B_{1,2} \\ \hline \mathbf{0} & \lambda B_{1,1} \end{array}\right), \tag{3.80}$$

with

$$\begin{pmatrix} A_{2,2} & A_{1,2} \end{pmatrix} = A_2 Q_1 \in \mathbb{C}^{(n-s_1)\times n}.$$

Since $|\det P_1 \det(A-\lambda B)\det Q_1| = |\det(A-\lambda B)| = |\lambda|^{s_1}|\det B_{1,1}\det(A_{2,2}-\lambda B_{2,2})|$ holds, it is clear that finding the additional zero eigenvalues reduces to the problem $A_{2,2} - \lambda B_{2,2}$. If $A_{2,2}$ is regular, there are no more zero eigenvalues, and the process stops. If $A_{2,2}$ is

singular, the process continues, that is, we find unitary matrices $\widehat{P}_2$ and $\widehat{Q}_2$ so that

$$P_2P_1(A-\lambda B)Q_1Q_2 = \left(\begin{array}{c|c|c} A_{33}-\lambda B_{3,3} & A_{2,3}-\lambda B_{2,3} & A_{1,3}-\lambda B_{1,3} \\ \hline \mathbf{0} & -\lambda B_{2,2} & A_{1,2}-\lambda B_{1,2} \\ \hline \mathbf{0} & \mathbf{0} & -\lambda B_{11} \end{array}\right),$$

where $P_2 = \mathrm{diag}(\widehat{P}_2, \mathbb{I}_{s_1}), Q_2 = \mathrm{diag}(\widehat{Q}_2, \mathbb{I}_{s_1})$.

The complete algorithm is described below

Algorithm 3.5.1 Deflation of eigenvalue 0

1: $j = 1$; $A_{1,1} = A$; $B_{1,1} = B$; $n_1 = n$;
2: Compute rank revealing QR: $A_{1,1}P_A = Q_AR_A$
3: $s_1 = n_1 - \mathrm{rank}(A_{1,1})$
4: **while** $s_j > 0$ **do**
5: Partition matrices: $\begin{pmatrix} A_{j+1} \\ \mathbf{0} \end{pmatrix} = Q_A^*A_{j,j}$, $\begin{pmatrix} B_{j+1} \\ B_j \end{pmatrix} = Q_A^*B_{j,j}$
6: Update and partition blocks in row j
7: **for** $i = 1 : j-1$ **do**
8: $\begin{pmatrix} A_{i,j+1} \\ A_{i,j} \end{pmatrix} = Q_A^*A_{i,j}$; $\begin{pmatrix} B_{i,j+1} \\ B_{i,j} \end{pmatrix} = Q_A^*B_{i,j}$;
9: **end for**
10: Compute complete orthogonal decomposition of $s_j \times n_j$ matrix B_j: $B_j = A_BR_BV_B^*$
11: Compress B_j to full column rank, permute and partition:
12: $\begin{pmatrix} A_{j+1,j+1} & A_{j,j+1} \end{pmatrix} = A_{j+1}V_BP_B$; $\begin{pmatrix} B_{j+1,j+1} & B_{j,j+1} \end{pmatrix} = B_{j+1}V_BP_B$;
13: $\begin{pmatrix} \mathbf{0} & B_{j,j} \end{pmatrix} = B_jV_BP_B$
14: $n_{j+1} = n_j - s_j$, $j = j+1$
15: Compute rank revealing QR $A_{j,j}P_A = Q_AR_A$
16: $s_j = n_j - \mathrm{rank}(A_{j,j})$
17: **end while**

This algorithm results in

$$P(A-\lambda B)Q = \left(\begin{array}{c|c|c|c|c} A_{\ell+1,\ell+1}-\lambda B_{\ell+1,\ell+1} & A_{\ell,\ell+1}-\lambda B_{\ell,\ell+1} & \dots & A_{2,\ell+1}-\lambda B_{2,\ell+1} & A_{1,\ell+1}-\lambda B_{1,\ell+1} \\ \hline \mathbf{0} & -\lambda B_{\ell,\ell} & \dots & A_{2,\ell}-\lambda B_{2,\ell} & A_{1,\ell}-\lambda B_{1,\ell} \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline \mathbf{0} & \mathbf{0} & \dots & -\lambda B_{2,2} & A_{1,2}-\lambda B_{1,2} \\ \hline \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & -\lambda B_{1,1} \end{array}\right). \tag{3.81}$$

The deflation of the infinite eigenvalue can be done by the same algorithm, but with the reversed pencil $B - \lambda A$.

Remark 3.6. Algorithm 3.5.1 can be used to determine the structure of an arbitrary eigenvalue α. The only difference is that the starting matrix $A_{1,1} = A - \alpha B$ is shifted. The matrix $B_{1,1} = B$ stays the same. Consider the shifted second companion form linearization

$$A_{1,1} = \begin{pmatrix} C & -\mathbb{I} \\ K & \mathbf{0} \end{pmatrix} - \alpha \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I} \end{pmatrix} = \begin{pmatrix} C+\alpha M & -\mathbb{I} \\ K & \alpha\mathbb{I} \end{pmatrix}.$$

The first step of the algorithm is to determine the rank of $2n \times 2n$ matrix $A_{1,1}$. However, if we multiply $A_{1,1}$ with $\begin{pmatrix} \alpha\mathbb{I} & \mathbb{I} \\ \mathbf{0} & \mathbb{I} \end{pmatrix}$ from the right we get

$$\begin{pmatrix} \alpha\mathbb{I} & \mathbb{I} \\ \mathbf{0} & \mathbb{I} \end{pmatrix} A_{1,1} = \begin{pmatrix} \alpha^2 M + \alpha C + K & \mathbf{0} \\ K & \alpha\mathbb{I} \end{pmatrix},$$

meaning that the $\operatorname{rank}(A_{1,1}) = n + \operatorname{rank}(\alpha^2 M + \alpha C + K)$, and thus it is enough to compute the rank of the $n \times n$ matrix $\alpha^2 M + \alpha C + K$.

3.5.4 Putting it all together: Deflation process in KVADeig

We now describe the global structure of the new procedure. We assume that the initial scaling and balancing are done as requested by an expert user.

The first step of the deflation process is the computation of the rank revealing decomposition of the matrices M and K.
After the determination of the numerical ranks, we have three main cases:

1. both matrices are regular,
2. one of the matrices is singular,
3. both matrices are singular.

1. Both matrices M and K are regular We proceed as in quadeig algorithm, that is we use the rank revealing decomposition of the matrix M to reduce matrix B to an upper triangular form (3.40).

2. One of the matrices is singular We can assume, without loss of generality, that K is singular, because in the case of singular M we just consider the reversed problem.
Before we continue with deflation process of the $n - r_K$ zero eigenvalues, we check whether there are Jordan blocks for this eigenvalue, that is, we compute the numerical rank of the $n \times n$ block matrix (3.68). As we mentioned before, the nullity of this matrix is equal to the nullity of the matrix $\widetilde{A}_{22}$, and the next step depends on it.

2.1. Regular matrix A_{22} In this case we proceed as in `quadeig` algorithm. That is, the $n - r_K$ zero eigenvalues are deflated, and the matrix B is reduced to the upper triangular form (3.42).

2.2. Singular matrix A_{22} In the notation of Algorithm 3.5.1, this means that $s_2 \neq 0$, meaning that there exists more than one Jordan block for the zero eigenvalue. In this case, reduction of the matrix B to upper triangular form will not be conducted. Using the structure of the matrix A, the deflation of the first two blocks is done as in (3.66) and (3.69). For possible further deflation steps, Algorithm 3.5.1 is applied to the pencil $A_{33} - \lambda B_{33}$.

3. Both matrices are singular In this case, before any deflation process, we check the ranks of both block matrices

$$\left(\frac{Q_{K,2}^* C}{\widehat{R}_K P_K^T} \right), \quad \left(\frac{Q_{M,2}^* C}{\widehat{R}_M P_M^T} \right). \tag{3.82}$$

The ranks of these matrices determine whether there exist more than one Jordan block for the zero eigenvalue and the infinite eigenvalue, respectively. There are three possible outcomes:

3.1. Both matrices in (3.82) are regular This means that there are exactly $n - r_M$ infinite, and $n - r_K$ zero eigenvalues, which are deflated as in `quadeig` algorithm.

3.2. One of the matrices in (3.82) is singular In any case, we use the structure to deflate two Jordan blocks of eigenvalue zero, meaning that if $\left(\frac{Q_{M,2}^* C}{\widehat{R}_M P_M^T} \right)$ is singular, i.e if there are at least two Jordan block for infinite eigenvalues, the reversed problem is considered. Now, Algorithm 3.5.1 is used to compute the complete structure of the zero eigenvalue. The first two steps are as in (3.66) and (3.69), that is, the structure of original problem is used. After the deflation of the zero eigenvalue, we get new reduced pencil $\widetilde{A} - \lambda \widetilde{B}$. Now, Algorithm 3.5.1 is used to deflate the infinite eigenvalue of the generalized eigenvalue problem. We already know that the number of infinite eigenvalues is $n - r_M$, and this is used as a test when the rank of the matrix $\widetilde{B}$ is determined numerically. Namely, the rank of $\widetilde{B}$ is equal to the rank of M. We also know that only one step of Algorithm 3.5.1 is enough to deflate all infinite eigenvalues.

3.3. Both matrices in (3.82) are singular We consider the original problem, if the number of the detected zero eigenvalues is larger than the number of the detected infinite eigenvalues, and the reversed problem otherwise. That is, we want to use the structure to deflate zero eigenvalue, and we are considering either original or reversed problem, whichever has more zero eigenvalues. The first step is to deflate all zero eigenvalues, using Algorithm 3.5.1 (the first two steps are done using the structure of the matrix A). After that, we get the reduced pencil $\widetilde{\widetilde{A}} - \lambda \widetilde{\widetilde{B}}$. The next step is to deflate the infinite eigenvalues, using the Algorithm 3.5.1 on the reversed pencil. Its structure

is determined by the numbers s_i. From the previous computation we know $s_1 = n - r_M < n$, and $s_2 = n - \text{rank}(\begin{pmatrix} Q^*_{M,2}C \\ \widehat{R}_M P_M^T \end{pmatrix})$, and this is used as a test for rank determination in Algorithm 3.5.1. The decision tree for the described process is sketched in Figure 3.5

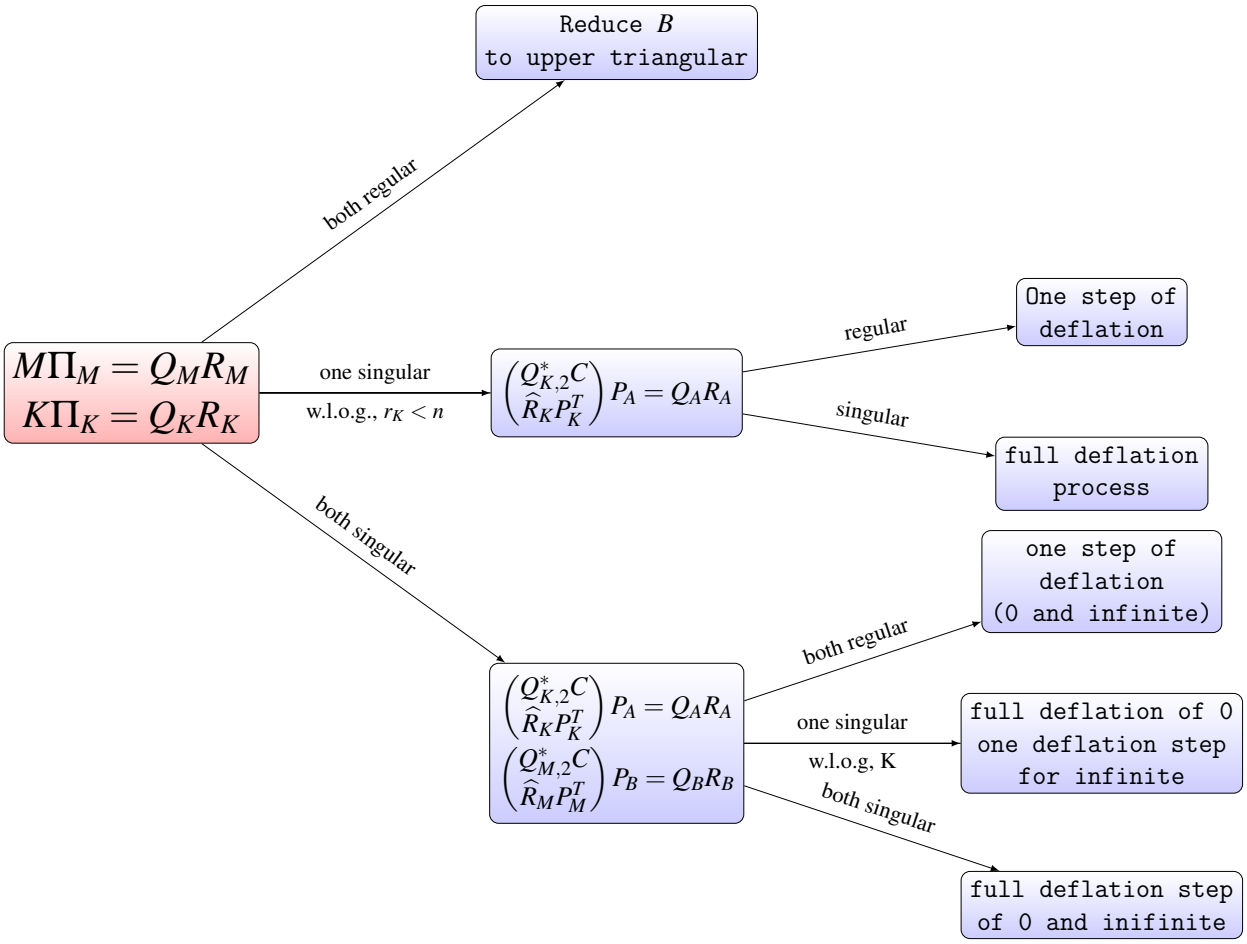


Figure 3.5: Deflation process in KVADeig – decision tree

Example 3.4 (continuation of the example intersection)**.** The deflation process described above deflates all 16 eigenvalues in 4 steps. The defects of the intermediate matrices A_{ii} are $s_1 = 7, s_2 = 6, s_3 = 2, s_4 = 1$. The computed finite eigenvalues are:

$$\begin{aligned}
\lambda_1 &= \texttt{-5.581811074974700e+008 -1.628029358197346e+009i},\\
\lambda_2 &= \texttt{-5.581811074974700e+008 +1.628029358197346e+009i},\\
\lambda_3 &= \texttt{2.476851768196167e+001 },\\
\lambda_4 &= \texttt{2.476851749893556e+001},
\end{aligned}$$

that is, two real, and a complex conjugate pair, as expected. The corresponding backward errors are given in Figure 3.6 below. The backward errors for the real eigenvalues computed by quadeig algorithm are also included. Note how this example shows that norm-wise small backward error can be completely misleading.

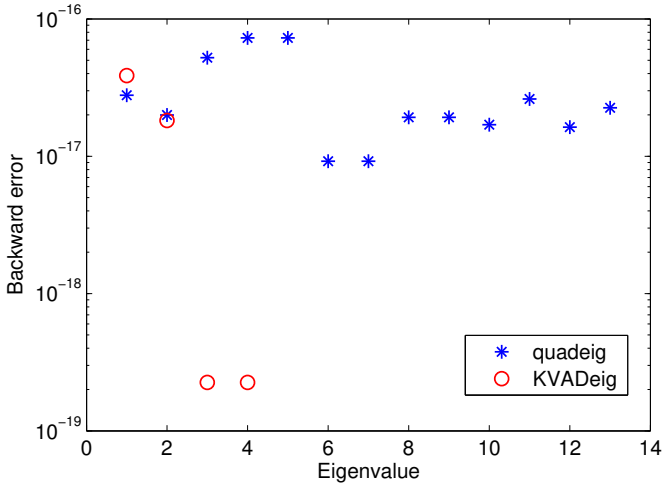


Figure 3.6: Backward errors for finite eigenvalues, sorted by magnitude, for the benchmark problem `intersection`.

Eigenvector recovery

There are two levels of the eigenvector recovery. First, we compute the eigenvectors of the transformed pencil $Q(A-\lambda B)V$, and we must recover the eigenvectors for the original pencil $A-\lambda B$. Second, we must recover the eigenvectors for the quadratic eigenvalue problem from the corresponding linearization.

The recovery of the eigenvectors in the cases when both M and K are regular, and when we have only one Jordan block to deflate for zero or/and infinite eigenvalues goes as explained in §3.3.3. In addition, we present the recovery in the case of the existence of more Jordan blocks.

Assume that more than one Jordan block is deflated for either zero or/and infinite eigenvalue. Let k be the dimension of the truncated pencil $\widetilde{A}-\lambda\widetilde{B}$. Let $z\in\mathbb{R}^{2n}$ and $w_1\in\mathbb{R}^{2n}$ be the computed right and left eigenvectors of $\widetilde{A}-\lambda\widetilde{B}$. If $k>n$, the right eigenvector is recovered as $x=Q(1:n,1:n)z(1:n)$, and if $k<n$ then $x=Q(1:n,1:k)z$.
For the left eigenvector, write the transformed pencil as

$$Q(A-\lambda B)V=\begin{pmatrix}\widetilde{A}-\lambda\widetilde{B} & X\\ \mathbf{0} & Y\end{pmatrix}.$$

Now, the left eigenvector $w\in\mathbb{R}^{2n}$ for the transformed pencil $Q(A-\lambda B)V$ is

$$w=\begin{pmatrix}w_1\\ w_2\end{pmatrix},\quad w_2=-w_1^*XY^{-1},$$

and w_1 is computed left eigenvector of $\widetilde{A}-\lambda\widetilde{B}$. For the left eigenvector we always have two choices for the original problem, and for the right eigenvector we have two choices only if K is nonsingular. By default we choose the eigenvector with smaller backward error.

3.5.5 Numerical examples

Experiment 1. `mobile_manipulator`. This example is also from the NLEVP library. It is a 5×5 quadratic matrix polynomial arising from modeling a two-dimensional three-link mobile manipulator as a time invariant descriptor control system. The matrices are of the form

$$M = \begin{pmatrix} M_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \; C = \begin{pmatrix} C_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \; K = \begin{pmatrix} K_0 & -F_0^T \\ F_0 & \mathbf{0} \end{pmatrix},$$

with

$$M_0 = \begin{pmatrix} 18.7532 & 7.94493 & 7.94494 \\ 7.94493 & 31.8182 & 26.8182 \\ 7.94494 & 26.8182 & 26.8182 \end{pmatrix}, \; C_0 = \begin{pmatrix} 1.52143 & 1.55168 & 1.55168 \\ 3.22064 & 3.28467 & 3.28467 \\ 3.22064 & 3.28467 & 3.28467 \end{pmatrix},$$
$$K_0 = \begin{pmatrix} 67.4894 & 69.2393 & 69.2393 \\ 69.8124 & 1.68624 & 1.68617 \\ 69.8123 & 1.68617 & 68.2707 \end{pmatrix}, \; F_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

This quadratic problem is known to be close to singular problem [13]. The matrix K has full rank, and the matrix M has rank $r_M = 3$. This means that there are at least $n - r_M = 2$ infinite eigenvalues amongst the total of 10 eigenvalues. We compute the eigenvalues of this problem using the `quadeig` algorithm, and these are the computed eigenvalues:

$\lambda_1 =$ `-5.1616e-002 -2.2435e-001i`	$\lambda_6 =$ `-1.0770e+006 -1.8660e+006i`
$\lambda_2 =$ `-5.1616e-002 +2.2435e-001i`	$\lambda_7 =$ `-1.0770e+006 +1.8660e+006i`
$\lambda_3 =$ `-2.7707e+005 -4.7991e+005i`	$\lambda_8 =$ `2.1551e+006`
$\lambda_4 =$ `-2.7707e+005 +4.7991e+005i`	$\lambda_9 =$ `Inf`
$\lambda_5 =$ `5.5416e+005`	$\lambda_{10} =$ `Inf`

We also compute the eigenvalues of this problem using the QZ algorithm directly on the second companion form linearization, without any prior deflation. The QZ algorithm found 8 infinite and two finite eigenvalues. Our algorithm deflated 8 zero eigenvalues from the pencil for the reversed problem. The two finite eigenvalues computed from the reduced pencil are:

$\lambda_1 =$ `-5.161621336216381e-002 -2.243476109085836e-001i`
$\lambda_2 =$ `-5.161621336216381e-002 +2.243476109085836e-001i`.

The problem in `quadeig` is in the reduction of the matrix B to the upper triangular form in the deflation process. In the QZ algorithm this step is done after the balancing algorithm [70] of the matrices A and B. Also this algorithm permutes rows and columns of matrices in order to use the sparsity structure to deflate possible zero or infinite eigenvalues before the main steps of the algorithm. The reduction to upper triangular form in `quadeig` algorithm destroys the structure

and QZ is unable to detect more infinite eigenvalues. Notice that the computed eigenvalues do not have big absolute values either. Finally, we conclude that only the scaling of the matrices M, C and K is not enough. For the interesting discussion regarding the balancing in eigenvalue computation, refer to [71].

Recall that `quadeig` works with reversed problem when the matrix M is singular, that is it deflates the zero eigenvalues. So, in this case, the algorithm deflated 2 zero eigenvalues. We computed the rank of matrix A after the deflation, and the rank was 6, meaning that there were at least two more zero eigenvalues which the QZ algorithm could not detect.

Experiment 2. Here, we present more examples from the NLEVP library where our algorithm detects more zero or/and infinite eigenvalues than `quadeig`:

Table 3.3: Number of deflated eigenvalues

	`quadeig`		`KVADeig`	
Problem	**zero**	**infinite**	**zero**	**infinite**
`bilby`	1	2	1	3(2+1)
`omnicam1`	11(8+3)	0	12(8+4)	0
`omnicam2`	14	0	23(14+9)	0
`relative_pose_6pt`	0	4	0	5(4+1)
`shaft`	0	201	0	402(201+201)

The numbers inside parentheses represent the numbers of deflated eigenvalues per deflation step. In the `quadeig` case, for the `omnicam1` problem, the QZ algorithm deflated additional 3 zero eigenvalues in addition to the 8 from the deflation process.

3.6 LU based deflation

Instead of the QR factorization, we can use the LU factorization with complete pivoting for determining the rank of the coefficient matrices in order to deflate zero and infinite eigenvalues. The transformation matrices Q and V in the deflation process now depend on the triangular matrices L and U, and on the inverse of the matrix L. However, L is triangular matrix, meaning that the inverse multiplication is actually just solution of lower triangular system of equations. Du to pivoting, it is expected to be well conditioned with respect to linear system solution.
In this section we develop a `quadeig`–type algorithm and the deflation Algorithm 3.5.1 using the LU factorization with complete pivoting as rank revealing factorization (see §3.1.4).

Let

$$L_K U_K = P_K K Q_K,\ U_K = \begin{pmatrix} \widehat{U}_K \\ \mathbf{0}_{n-r_K,n} \end{pmatrix},$$
$$L_M U_M = P_M M Q_M,\ U_M = \begin{pmatrix} \widehat{U}_M \\ \mathbf{0}_{n-r_M,n} \end{pmatrix}$$

be the LU factorizations with complete pivoting for the coefficient matrices of the quadratic pencil $\lambda^2 M + \lambda C + K$. In the following subsection we present the deflation process of one Jordan block of zero eigenvalue using the rank revealing LU factorization.

3.6.1 The case of nonsingular M

First, if $\operatorname{rank}(K) = \operatorname{rank}(M) = n$, the equivalence transformation is

$$\begin{aligned}
&\begin{pmatrix} L_M^{-1}P_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \left\{ \begin{pmatrix} C & -\mathbb{I}_n \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} Q_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \\
= &\left(\begin{array}{c|c} L_M^{-1}P_M C Q_M & -L_M^{-1}P_M \\ \hline KQ_M & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c} -L_M^{-1}P_M M Q_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array} \right) \\
= &\left(\begin{array}{c|c} L_M^{-1}P_M C Q_M & -L_M^{-1}P_M \\ \hline KQ_M & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c} -U_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array} \right).
\end{aligned} \tag{3.83}$$

If $\operatorname{rank}(K) < \operatorname{rank}(M) = n$ we have following transformation:

$$\begin{aligned}
&\begin{pmatrix} L_M^{-1}P_M & \mathbf{0} \\ \mathbf{0} & L_K^{-1}P_K \end{pmatrix} \left\{ \begin{pmatrix} C & -\mathbb{I}_n \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} Q_M & \mathbf{0} \\ \mathbf{0} & P_K^T L_K \end{pmatrix} \\
= &\left(\begin{array}{c|c} L_M^{-1}P_M C Q_M & L_M^{-1}P_M P_K^T L_K \\ \hline L_K^{-1}P_K K Q_M & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c} L_M^{-1}P_M M Q_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array} \right) \\
= &\left(\begin{array}{c|c} L_M^{-1}P_M C Q_M & L_M^{-1}P_M P_K^T L_K \\ \hline L_K^{-1}P_K K Q_M & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c} -U_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n \end{array} \right) \\
\equiv &\left(\begin{array}{c|c|c} X_{11} & X_{12} & X_{13} \\ \hline X_{21} & \mathbf{0}_{r_K,r_K} & \mathbf{0}_{r_K,n-r_K} \\ \hline \mathbf{0}_{n-r_K,n} & \mathbf{0}_{n-r_K,r_K} & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c|c} -U_M & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \multicolumn{2}{c|}{\mathbf{0}} & \begin{array}{c|c} -\mathbb{I}_{r_K} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_{n-r_K} \end{array} \end{array} \right).
\end{aligned} \tag{3.84}$$

The reduced $(n+r_K)\times(n+r_K)$ pencil is

$$A - \lambda B = \left(\begin{array}{c|c} X_{11} & X_{12} \\ \hline X_{21} & \mathbf{0}_{r_K,r_K} \end{array} \right) - \lambda \left(\begin{array}{c|c} -R_M & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_{r_K} \end{array} \right). \tag{3.85}$$

3.6.2 Rank deficient cases

In the case when $r_K \leq r_M < n$ transformation is

$$
\begin{aligned}
&\begin{pmatrix} L_M^{-1}P_M & \mathbf{0} \\ \mathbf{0} & L_K^{-1}P_K \end{pmatrix} \left\{ \begin{pmatrix} C & -\mathbb{I}_n \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & P_K^T L_K \end{pmatrix} \\
= &\left(\begin{array}{c||c} L_M^{-1}P_M C & -L_M^{-1}P_M P_K^T L_K \\ \hline\hline \widehat{U}_K Q_K^T & \mathbf{0}_{r_K,n} \\ \hline \mathbf{0}_{n-r_k,n} & \mathbf{0}_{n-r_K,n} \end{array} \right) - \lambda \left(\begin{array}{c||c} -\widehat{U}_M Q_M^T & \mathbf{0}_{r_M,n} \\ \hline \mathbf{0}_{n-r_M,n} & \mathbf{0}_{n-r_M,n} \\ \hline\hline \mathbf{0}_{n,n} & -\mathbb{I}_n \end{array} \right) \\
\equiv &\left(\begin{array}{cc||cc} (L_M^{-1}P_M C)_{11} & (L_M^{-1}P_M C)_{12} & -L_M^{-1}P_M P_K^T L_K \\ \hline (L_M^{-1}P_M C)_{21} & (L_M^{-1}P_M C)_{22} & \\ \hline\hline (\widehat{U}_K Q_K^T)_{11} & (\widehat{U}_K Q_K^T)_{12} & \mathbf{0} \;\; \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \;\; \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{cc||c} -(\widehat{U}_M Q_M^T)_{11} & -(\widehat{U}_M Q_M^T)_{12} & \mathbf{0}_n \\ \hline \mathbf{0} & \mathbf{0} & \\ \hline\hline \mathbf{0}_n & & -\mathbb{I}_n \end{array} \right) \\
\equiv &\left(\begin{array}{cc||cc} X_{11} & X_{12} & X_{13} & X_{14} \\ \hline X_{21} & X_{22} & X_{23} & X_{24} \\ \hline\hline X_{31} & X_{32} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{cc||cc} Y_{11} & Y_{12} & \mathbf{0}_{r_M,r_K} & \mathbf{0}_{r_M,n-r_K} \\ \hline \mathbf{0}_{n-r_M,r_M} & \mathbf{0}_{n-r_M,n-r_M} & \mathbf{0}_{n-r_M,r_K} & \mathbf{0}_{n-r_M,n-r_K} \\ \hline\hline \mathbf{0}_{r_K,r_M} & \mathbf{0}_{r_K,n-r_M} & -\mathbb{I}_{r_K} & \mathbf{0}_{r_K,n-r_K} \\ \hline \mathbf{0}_{n-r_K,r_M} & \mathbf{0}_{n-r_K,n-r_M} & \mathbf{0}_{n-r_K,r_K} & -\mathbb{I}_{n-r_K} \end{array} \right) \\
\equiv &\; X - \lambda Y. \qquad (3.86)
\end{aligned}
$$

In the next step, we compute the complete orthogonal decomposition (i.e. URV decomposition, using unitary matrices Q_X and Z_X)

$$
\begin{array}{c} \\ n-r_M \end{array} \begin{array}{c} \begin{array}{ccc} r_M & n-r_M & r_K \end{array} \\ \begin{pmatrix} X_{21} & X_{22} & X_{23} \end{pmatrix} \end{array} = Q_X \begin{pmatrix} R_X & \mathbf{0}_{n-r_M,r_M+r_K} \end{pmatrix} Z_X, \;\; R_X \in \mathbb{C}^{(n-r_M)\times(n-r_M)}. \qquad (3.87)
$$

The remaining steps are analogous to those in §3.3.2.

3.6.3 Eigenvectors

Assume that the right eigenvectors z and the left eigenvectors w of the transformed pencil $QC_2(\lambda)V$ are computed. The transformations are defined as in the previous sections, depending on the ranks of the matrices M and K. The corresponding right eigenvector $\widetilde{z}$ and the left eigenvector $\widetilde{w}$ for the original linearization pencil $C_2(\lambda)$ are

$$
\begin{pmatrix} \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \end{pmatrix} = \widetilde{z} = Vz,
$$

$$
\begin{pmatrix} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix} = \widetilde{w} = Q^{-1}w.
$$

Notice that, in opposite to deflation process using the QR factorization, the left eigenvector is obtained by solving the system of equations. However, this can be reduced to solving triangular systems. The candidates for the right and the left eigenvectors for the original quadratic eigenvalue problem are derived next. If there are two choices for the eigenvector, the algorithm picks the one with the smaller (e.g. norm-wise) backward error.

The right eigenvectors

The first case: $\operatorname{rank}(M) = \operatorname{rank}(K) = n$**.** The matrix K is nonsingular, and we have two choices for the right eigenvector. Let $\widetilde{z}$ be the right eigenvector for the transformed GEP. The corresponding right eigenvector for $C_2(\lambda)$ is

$$z = \begin{pmatrix} z_1 \\ \hline\hline z_2 \end{pmatrix} = \begin{pmatrix} Q_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \begin{pmatrix} \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \end{pmatrix} = \begin{pmatrix} Q_M \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \end{pmatrix}.$$

Hence, the two candidates for the eigenvector x are $Q_M \widetilde{z}_1$ and $K^{-1}\widetilde{z}_2$.

The second case: $\operatorname{rank}(K) < \operatorname{rank}(M) = n$**.** The matrix K is singular, and $n - r_K$ zero eigenvalues are deflated. The eigenvectors corresponding to those eigenvalue span the nullspace of the matrix K. The basis for the nullspace is computed using orthogonal complement of the range of K^* using the QR decomposition of the matrix $\widehat{U}_K^* Q_K$:

$$\widehat{U}_K^* Q_K = Q_{\widehat{U}_K^*} R_{\widehat{U}_K^*}.$$

The wanted vector are the last $n - r_K$ columns of the orthogonal matrix $Q_{\widehat{U}_K^*}$.
The remaining eigenvalues and eigenvectors $\widetilde{z} \in \mathbb{C}^{n+r_K}$ are computed from the $(n+r_K) \times (n+r_K)$ GEP (3.85). The corresponding eigenvector for $C_2(\lambda)$ is

$$\begin{pmatrix} z_1 \\ \hline\hline z_2 \end{pmatrix} = \begin{pmatrix} Q_M & \mathbf{0} \\ \mathbf{0} & P_K^T L_K \end{pmatrix} \begin{pmatrix} \widetilde{z}_1 \\ \hline\hline \widetilde{z}_2 \\ \mathbf{0}_{n-r_K} \end{pmatrix} = \begin{pmatrix} Q_M \widetilde{z}_1 \\ \hline\hline P_K^T L_K \begin{pmatrix} \widetilde{z}_2 \\ \mathbf{0}_{n-r_K} \end{pmatrix} \end{pmatrix}.$$

The only candidate for the right eigenvector x is $Q_M \widetilde{z}_1$.

The third case: $\operatorname{rank}(K) \leq \operatorname{rank}(M) < n$**.** Both matrices M and K are singular, and $n - r_M$ infinite and $n - r_K$ zero eigenvalues are deflated. The eigenvectors for zero eigenvalues are obtained as in the previous case, whilst the eigenvectors for infinite eigenvalues form the basis for the nullspace of the matrix M. As before, the basis is obtained as the orthogonal complement

of the range of M^* represented by the last $n - r_M$ columns of the orthogonal matrix $Q_{\widehat{U}_M^*}$

$$\widehat{U}_M^* Q_M = Q_{\widehat{U}_M^*} R_{\widehat{U}_M^*}.$$

The remaining eigenvalues and eigenvectors $\widetilde{z} \in \mathbb{C}^{r_K + r_M}$ are obtained from the $(r_K + r_M) \times (r_K + r_M)$ GEP (3.53). The corresponding eigenvector for $C_2(\lambda)$ is

$$\begin{aligned}\begin{pmatrix} z_1 \\ \hline\hline z_2 \end{pmatrix} &= \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & P_K^T L_K \end{pmatrix} \begin{pmatrix} Z_X^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_{n-r_K} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbb{I}_{n-r_M} & \mathbf{0} \\ \mathbb{I}_{r_K+r_M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{I}_{n-r_K} \end{pmatrix} \begin{pmatrix} \widetilde{z} \\ \hline\hline \mathbf{0}_{n-r_M} \\ \mathbf{0}_{n-r_K} \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \begin{pmatrix} Z_X^{-1} \begin{pmatrix} \mathbf{0}_{n-r_M} \\ \widetilde{z} \end{pmatrix} \\ \hline\hline \mathbf{0}_{n-r_K} \end{pmatrix}.\end{aligned}$$

The wanted eigenvector x is $Z_X^{-1} \begin{pmatrix} \mathbf{0}_{n-r_M} \\ \widetilde{z} \end{pmatrix} (:, 1:n)$.

The left eigenvectors

The first case: $\operatorname{rank}(M) = \operatorname{rank}(K) = n$**.** Let $\widetilde{w}$ be the left eigenvector for the transformed GEP $QC_2(\lambda)V$. The corresponding left eigenvector for the linearization $C_2(\lambda)$ is w

$$\begin{pmatrix} w_1 \\ \hline\hline w_2 \end{pmatrix} = \begin{pmatrix} P_M^T L_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \begin{pmatrix} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix} = \begin{pmatrix} P_M^T L_M \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix}.$$

The two candidates for the left eigenvector y of the quadratic eigenvalue problem are $P_M^T L_M \widetilde{w}_1$ and $\widetilde{w}_2$.

The second case: $\operatorname{rank}(K) < \operatorname{rank}(M) = n$**.** The left eigenvectors for the zero eigenvalue are the last $n - r_K$ columns of the matrix $P_K^T L_K$. Let $\begin{pmatrix} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \end{pmatrix} \in \mathbb{C}^{n+r_K}$ be the eigenvector for the deflated $n + r_K \times n + r_K$ pencil (3.85). The corresponding eigenvector for the $2n \times 2n$ pencil, before truncation, is

$$\left(\begin{array}{c|c|c} \widetilde{w}_1^* & \widetilde{w}_2^* & \widetilde{w}_3^* \end{array} \right) \left(\left(\begin{array}{c|c|c} X_{11} & X_{12} & X_{13} \\ \hline\hline X_{21} & \mathbf{0}_{r_K,r_K} & \mathbf{0}_{r_K,n-r_K} \\ \hline \mathbf{0}_{n-r_K,n} & \mathbf{0}_{n-r_K,r_K} & \mathbf{0} \end{array} \right) - \lambda \left(\begin{array}{c|c|c} -U_M & \mathbf{0} & \\ \hline\hline \mathbf{0} & -\mathbb{I}_{r_K} & \mathbf{0} \\ \hline & \mathbf{0} & -\mathbb{I}_{n-r_K} \end{array} \right) \right) =$$

$$\left(\begin{array}{c} \widetilde{w}_1^* X_{11} + \widetilde{w}_2^* X_{21} + \lambda \widetilde{w}_1^* U_M \\ \hline\hline \widetilde{w}_1^* X_{12} + \lambda \widetilde{w}_2^* \\ \widetilde{w}_1^* X_{13} + \lambda \widetilde{w}_3^* \end{array}\right) = \mathbf{0},$$

therefore, $\widetilde{w}_3 = X_{13}^* \widetilde{w}_1 / \lambda$. The vector z for $C_2(\lambda)$ is

$$\left(\begin{array}{c} w_1 \\ \hline\hline w_2 \end{array}\right) = \begin{pmatrix} P_M^T L_M & \mathbf{0} \\ \mathbf{0} & P_K^T L_K \end{pmatrix} \left(\begin{array}{c} \widetilde{w}_1 \\ \hline\hline \widetilde{w}_2 \\ \widetilde{w}_3 \end{array}\right) = \left(\begin{array}{c} P_M^T L_M \widetilde{w}_1 \\ \hline\hline P_K^T L_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_3 \end{pmatrix} \end{array}\right).$$

The left eigenvector y for the QEP is now picked between $P_M^T L_M \widetilde{w}_1$ and $P_K^T L_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_3 \end{pmatrix}$.

The third case: $\operatorname{rank}(K) \leq \operatorname{rank}(M) < n$**.** The left eigenvectors for zero eigenvalues are the last $n - r_K$ columns of $P_K^T L_K$, and for infinite eigenvalues are the last $n - r_M$ columns of $P_M^T L_M$. Let $\begin{pmatrix} \widetilde{w}_1 \\ \widetilde{w}_2 \end{pmatrix} \in \mathbb{C}^{r_K + r_M}$ be a left eigenvector for truncated $r_K + r_M \times r_K + r_M$ pencil (3.53). The corresponding eigenvector for the pencil $QC_2(\lambda)V$ is then

$$\left(\begin{array}{cc|c|cc} \widetilde{w}_1^* & \widetilde{w}_2^* & \widetilde{w}_3^* & \widetilde{w}_4^* \end{array}\right) \left(\left(\begin{array}{cc||cc} \widetilde{X}_{11} & \widehat{X}_{12} & \widehat{X}_{13} & X_{14} \\ \widetilde{X}_{21} & \widehat{X}_{22} & \widehat{X}_{23} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & R_X & \widetilde{X}_{24} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right) - \lambda \left(\begin{array}{cc||cc} \widetilde{Y}_{11} & \widehat{Y}_{12} & \widehat{Y}_{13} & \mathbf{0} \\ \widetilde{Y}_{21} & \widehat{Y}_{22} & \widehat{Y}_{23} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbb{I} \end{array}\right) \right) =$$

$$= \left(\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \hline\hline \widetilde{w}_1^* \widehat{X}_{13} + \widetilde{w}_2^* \widehat{X}_{23} + \widetilde{w}_3^* R_X - \lambda \widetilde{w}_1^* \widehat{Y}_{13} - \lambda \widetilde{w}_2^* \widehat{Y}_{23} \\ \widetilde{w}_1^* \widehat{X}_{14} + \widetilde{w}_3^* \widetilde{X}_{24} + \lambda \widetilde{w}_4^* \end{array}\right) = \mathbf{0}.$$

The components $\widetilde{w}_3^*, \widetilde{w}_4^*$ are thus computed as

$$\widetilde{w}_3^* = \left(\lambda \widetilde{w}_1^* \widehat{Y}_{13} + \lambda \widetilde{w}_2^* \widehat{Y}_{23} - \widetilde{w}_1^* \widehat{X}_{13} - \widetilde{w}_2^* \widehat{X}_{23}\right) R_X^{-1},$$
$$\widetilde{w}_4^* = \left(-\widetilde{w}_1^* \widehat{X}_{14} - \widetilde{w}_3^* \widetilde{X}_{24}\right) / \lambda.$$

The left eigenvector for $C_2(\lambda)$ is

$$w = \begin{pmatrix} P_M^T L_M & \mathbf{0} \\ \mathbf{0} & P_K^T L_K \end{pmatrix} \begin{pmatrix} \mathbb{I}_{r_M} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_X & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_{r_K} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I}_{n-r_K} \end{pmatrix} \left(\begin{array}{c} \widetilde{w}_1 \\ \widetilde{w}_2 \\ \hline\hline \widetilde{w}_3 \\ \widetilde{w}_4 \end{array}\right) = \left(\begin{array}{c} P_M^T L_M \begin{pmatrix} \widetilde{w}_1 \\ Q_X \widetilde{w}_3 \end{pmatrix} \\ \hline\hline P_K^T L_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_4 \end{pmatrix} \end{array}\right),$$

and the candidates for the left eigenvector y are $P_M^T L_M \begin{pmatrix} \widetilde{w}_1 \\ Q_X \widetilde{w}_3 \end{pmatrix}$ and $P_K^T L_K \begin{pmatrix} \widetilde{w}_2 \\ \widetilde{w}_4 \end{pmatrix}$.

3.6.4 Computing the Kronecker's Canonical form using rank revealing LU factorization

In this subsection, we derive an algorithm for deflating the eigenvalue zero, using the rank revealing LU factorization instead of the SVD or the QR factorization with column pivoting. We will describe the first step in moe detail, and then formulate the algorithm.
Let $A, B \in \mathbb{C}^{n\times n}$. Denote by n_i the size of a working matrix in step i, and s_i the defect of working matrix in step i.

1. Compute the rank revealing factorization of $A_{1,1} = A$

$$Q_A A_{1,1} P_A = L_A U_A, \tag{3.88}$$

and denote $s_1 = n_1 - \text{rank}(A) = n - \text{rank}(A)$. Now, $L_A^{-1} Q_A A_{1,1} = \left(\begin{array}{c} A_2 \\ \hline \mathbf{0}_{s_1\times n} \end{array} \right)$. Partition $L_A^{-1} Q_A B = \left(\begin{array}{c} B_2 \\ \hline B_1 \end{array} \right)$ in compatible manner. Multiply the pencil $(A - \lambda B)$ by $L_A^{-1} Q_A$ on the left to get

$$L_A^{-1} Q_A (A - \lambda B) = \left(\begin{array}{c} A_2 - \lambda B_2 \\ \hline \lambda B_1 \end{array} \right). \tag{3.89}$$

2. Compute the complete orthogonal decomposition of B_1

$$B_1 = U_B R_B V_B^*. \tag{3.90}$$

The column rank of B_1 is s_1, if the matrix pencil is regular, and $B_1 V_B = \begin{pmatrix} B_{1,1} & \mathbf{0}_{s_1, n-s_1} \end{pmatrix}$. Multiply the pencil (3.89) by V_B from the right to get

$$L_A^{-1} Q_A (A - \lambda B) V_B = \left(\begin{array}{c|c} A_{1,2} - \lambda B_{1,2} & A_{2,2} - \lambda B_{2,2} \\ \hline \lambda B_{1,1} & \mathbf{0} \end{array} \right). \tag{3.91}$$

3. Let P_B be the permutation matrix for permuting the s_1 and $n - s_1$ column blocks. Define $P_1 = L^{-1} Q_A$ and $Q_1 = Z_B P_B$. The first Jordan block for the eigenvalue 0 is deflated by the following orthogonal transformation:

$$P_1 (A - \lambda B) Q_1 = \left(\begin{array}{c|c} A_{2,2} - \lambda B_{2,2} & A_{1,2} - \lambda B_{1,2} \\ \hline \mathbf{0} & \lambda B_{1,1} \end{array} \right). \tag{3.92}$$

Complete algorithm is described below

Algorithm 3.6.1 Deflation of eigenvalue 0

1: $j = 1$; $A_{1,1} = A$; $B_{1,1} = B$; $n_1 = n$;
2: Compute rank revealing LU: $Q_A A_{1,1} P_A = L_A U_A$
3: $s_1 = n_1 - \mathrm{rank}(A_{1,1})$
4: **while** $s_j > 0$ **do**
5: Partition matrices: $\begin{pmatrix} A_{j+1} \\ \mathbf{0} \end{pmatrix} = L_A^{-1} Q_A A_{j,j}$, $\begin{pmatrix} B_{j+1} \\ B_j \end{pmatrix} = L_A^{-1} Q_A B_{j,j}$
6: Update and partition blocks in row j
7: **for** $i = 1 : j-1$ **do**
8: $\begin{pmatrix} A_{i,j+1} \\ A_{i,j} \end{pmatrix} = L_A^{-1} Q_A A_{i,j}$; $\begin{pmatrix} B_{i,j+1} \\ B_{i,j} \end{pmatrix} = L_A^{-1} Q_A B_{i,j}$;
9: **end for**
10: Compute the complete orthogonal decomposition $B_j = U_B R_B V_B^*$
11: Compress B_j to full column rank, permute and partition:
12: $\begin{pmatrix} A_{j+1,j+1} & A_{j,j+1} \end{pmatrix} = A_{j+1} V_B P_B$; $\begin{pmatrix} B_{j+1,j+1} & B_{j,j+1} \end{pmatrix} = B_{j+1} V_B P_B$;
13: $\begin{pmatrix} \mathbf{0} & B_{j,j} \end{pmatrix} = B_j V_B P_B$
14: $n_{j+1} = n_j - s_j$, $j = j+1$
15: Compute rank revealing LU: $Q_A A_{j,j} P_A = L_A U_A$
16: $s_j = n_j - \mathrm{rank}(A_{j,j})$
17: **end while**

This algorithm results in

$$P(A-\lambda B)Q = \left(\begin{array}{c|c|c|c|c} A_{\ell+1,\ell+1}-\lambda B_{\ell+1,\ell+1} & A_{\ell,\ell+1}-\lambda B_{\ell,\ell+1} & \dots & A_{2,\ell+1}-\lambda B_{2,\ell+1} & A_{1,\ell+1}-\lambda B_{1,\ell+1} \\ \hline \mathbf{0} & -\lambda B_{\ell,\ell} & \dots & A_{2,\ell}-\lambda B_{2,\ell} & A_{1,\ell}-\lambda B_{1,\ell} \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline \mathbf{0} & \mathbf{0} & \dots & -\lambda B_{2,2} & A_{1,2}-\lambda B_{1,2} \\ \hline \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & -\lambda B_{1,1} \end{array}\right). \tag{3.93}$$

The deflation of infinite eigenvalues can be done by the same algorithm, but with reversed pencil $B - \lambda A$.

Numerical examples

Experiment 1. `intersection`**.** Recall the case study example from Subsection 3.5.1. We used Algorithm 3.6.1 to compute the structure of zero eigenvalues in the reversed problem. This algorithm also deflated 16 zero eigenvalues, 7 in the first, 6 in the second, 2 in the third and 1 in the fourth step of the process, just as Algorithm 3.5.1. The computed real eigenvalues are

$$\begin{aligned}
\lambda_1 &= \texttt{2.476851749893558e+001}, \\
\lambda_2 &= \texttt{2.476851768196165e+001}, \\
\lambda_3 &= \texttt{-5.581818959997490e+008 -1.628030389374511e+009i}, \\
\lambda_4 &= \texttt{-5.581818959997490e+008 +1.628030389374511e+009i}.
\end{aligned}$$

The following figure shows the backward error for the computed finite eigenvalues for all three algorithms, `quadeig`, `KVADeig`, and LU based `KVADeig`.

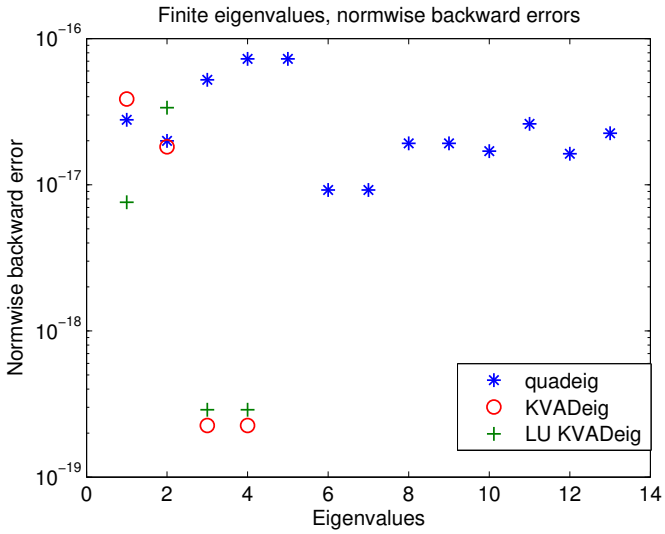


Figure 3.7: Backward errors for the finite eigenvalues sorted by magnitude for the `intersection` problem

Experiment 2. **`mobile_manipulator`.** In this example, Algorithm 3.6.1 deflated 8 zero eigenvalues in the reversed problem. There where 4 steps of deflation, and two zero eigenvalues were deflated in every step. The two finite computed eigenvalues after the deflation are

$$\lambda_1 = \texttt{-5.161621336216380e-002 -2.243476109085838e-001i},$$
$$\lambda_2 = \texttt{-5.161621336216380e-002 +2.243476109085838e-001i}.$$

Figure representing the backward errors for finite eigenvalues computed by `quadeig`, `KVADeig`, and `LU KVADeig` is presented below.

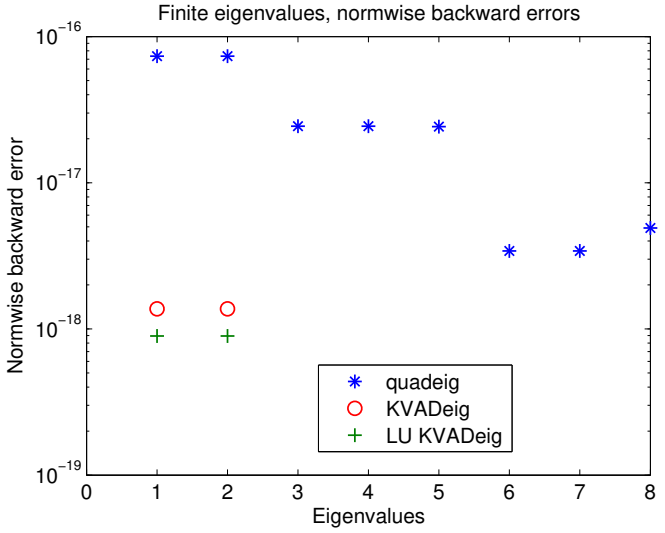


Figure 3.8: Comparison of the backward errors for the finite eigenvalues, sorted by magnitude, for the `mobile_manipulator` problem

3.7 Numerical examples. Comparison of rank revealing decompositions

The goal of this section is to present the difference in computed results using different rank revealing decompositions. The emphasize is not on the deflation process, but on the transformation of the pencil when no zero or infinite eigenvalues are detected. We will use three rank revealing decompositions:

- QR factorization with column pivoting (QR)
- QR factorization with column pivoting and initial sorting of rows so that (3.11) holds (QRrs) (default in `KVADeig`)
- LU factorization with complete pivoting (LUcp).

In addition, we will illustrate the importance of rank determination in the first step of deflation process. Our algorithm offers two types of criteria for rank determination:

1. rank of matrix A is equal to $k-1$, where k is the first index for which $R_{k,k} > \tau\|A\|_F$ holds, where $A\Pi = QR$ is rank revealing factorization (F-norm);
2. rank of matrix A is equal to $k-1$, where k is the last index for which $|R_{k,k}|/|R_{k-1,k-1}| \geq \tau$, where $A\Pi = QR$ is rank revealing factorization, and τ is prescribed threshold (drop-off).

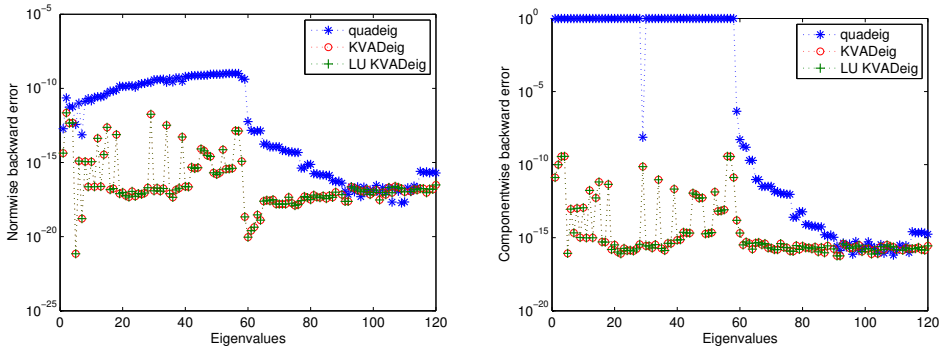
It will be clear from all examples that component-wise backward error gives better insight into the accuracy of computed solutions than the norm-wise backward error. This stresses the importance of the techniques such as parameter scaling and diagonal balancing (advocated in this chapter).

3.7.1 Example 1. `cd_player`.

This a example from NLEVP library [5]. It is a quadratic eigenvalue problem arising in the study of a CD player control task. The dimension of the problem is $n = 60$; the matrix M is the identity.

Original problem

We computed the eigenvalues for this problem using three different rank revealing decompositions in the deflation process: the QR with column pivoting, the QR with complete pivoting (presorting of rows followed by column pivoting), and the LU with complete pivoting. For the last two we used `KVADeig` implementation, and for the first one we used `quadeig`. The computed eigenvalues are sorted by magnitude in ascending order. The norm-wise and component-wise backward errors forthe eigenvalues are given in Figures 3.9a and 3.9b.

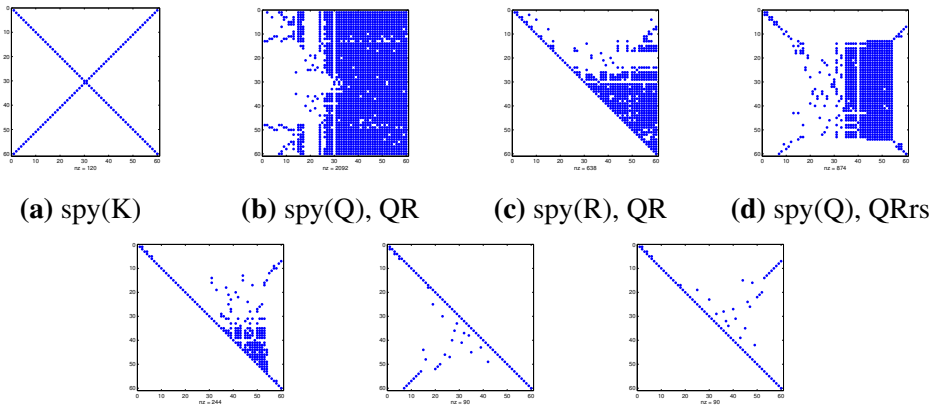


(a) Normwise backward errors **(b)** Componentwise backward errros

Figure 3.9: Comparison of the normwise and componentwise backward errors for the right eigenpair for the problem `cd_player`

From Figure 3.9a we see that the normwise backward errors for `KVADeig` and LU based `KVADeig` are similar, and the backward errors for `quadeig` are bit higher for the first 60 eigenvalues. However, the real difference is seen in the Figure 3.9b of component-wise backward errors. Precisely, for `quadeig`, the error is equal to 1 for most of the first 60 eigenvalues. We explain the reason for this below.

The matrix M is identity, so we do not have any transformation of the linearization pencil in the deflation process. However, when choosing eigenvector, we have two choices: x_1 and $K^{-1}x_2$, where $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is the corresponding eigenvector for the linear pencil. In `quadeig`, the system $K^{-1}x_2$ is solved using the computed rank revealing factorization of the matrix K. Next figures represent the structure of the matrix K and the corresponding rank revealing factorizations.



(a) spy(K) **(b)** spy(Q), QR **(c)** spy(R), QR **(d)** spy(Q), QRrs

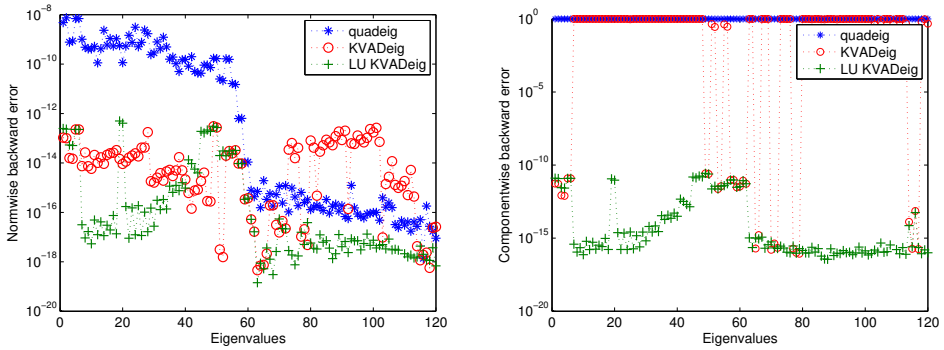
(e) spy(R), QRrs **(f)** spy(L), LUcp **(g)** spy(U), LUcp

Figure 3.10: Sparsity structure of the matrix K and the corresponding components in the rank revealing factorizations

In the case of the first 60 eigenvalues in `quadeig`, the block $K^{-1}x_2$ is chosen because its norm-wise backward error is smaller, however, solving the system using the QR factorization has bigger component-wise error. It is clear from the figures that the QR factorization does not inherit

the sparsity of the original matrix, in contrast to the LU factorization. In our algorithm, $K^{-1}x$ is computed using the LU factorization. Together with `intersection` and `mobile_manipulator` examples, this is another example where the norm-wise backward error can be misleading. In this case, `quadeig` had access to the better solution, but the criteria for choosing the approximate solution lead to the wrong one.

Reversed problem If we consider the reversed problem, the leading matrix will be K, so the first step will be the reduction to upper triangular form of the matrix B in the linearization pencil. Thereby, the rank revealing factorization from the previous paragraph will be used. Norm-wise and component-wise backward errors in this case are presented in the Figures 3.11a and 3.11b.



(a) Normwise backward error **(b)** Componentwise backward error

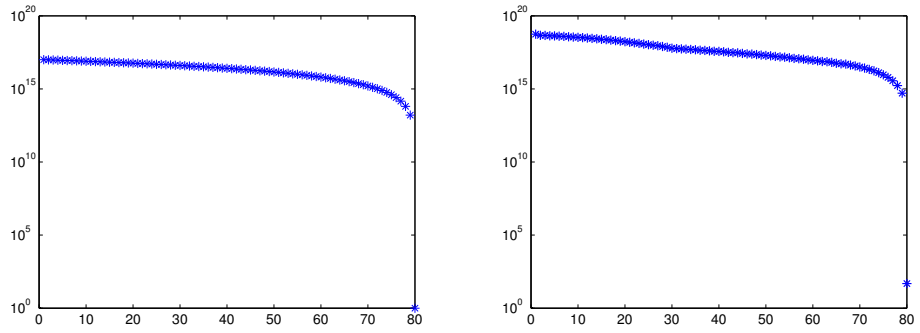
Figure 3.11: Comparison of the normwise and backward errors for the right eigenpair for the reversed `cd_player` problem

In this case, the component-wise backward errors are equally high for all eigenvalues when the QR factorizations are used, because sparsity is disturbed in the first step of the algorithm. However, when we use the LU factorization, the error is satisfactory.
There is the difference between the norm-wise backward errors as well, and this is due to computation of the block $K^{-1}x$, as in the original case.

3.7.2 Example 2. Scaled `dirac`.

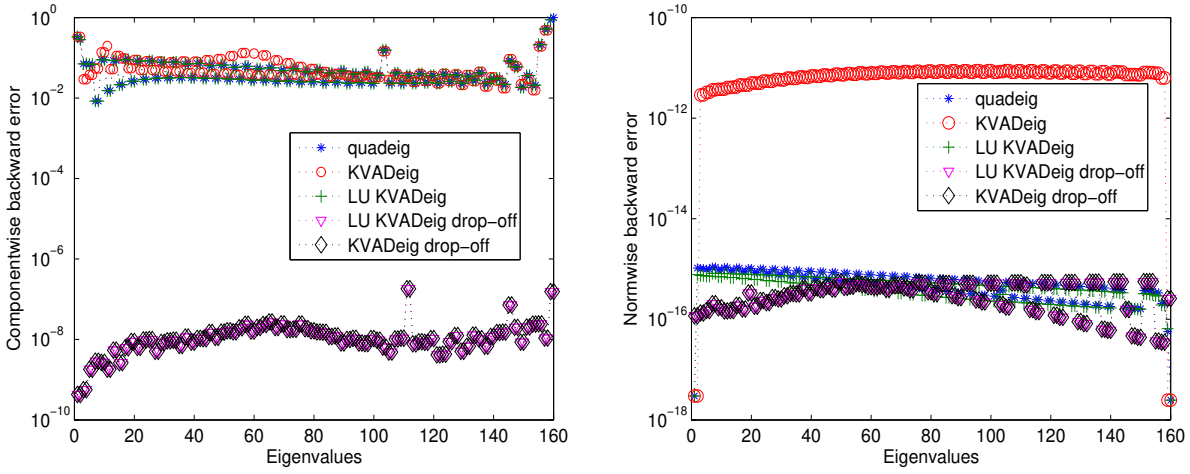
We analyze another examples from the NLEVP library. It is a quadratic eigenvalue problem that originates from the Dirac operator. The matrix M is identity, and the condition number of the matrix K is 367.4304. There is no significant difference between the methods either for original or reversed problem. However, if we scale the original problem, to increase the condition of the matrices, there is essential difference. Note that this creates a synthetic example and the goal is to illustrate the importance of scaling.

We created diagonal matrices S_L and S_R of conditions 10^8 and 10^9, respectively. The equivalent scaled quadratic problem is $(\lambda^2 S_L M S_R + \lambda S_L C S_R + S_L K S_R)x = \mathbf{0}$. The singular values for the matrices M and K are shown in Figures 3.12a, 3.12b.



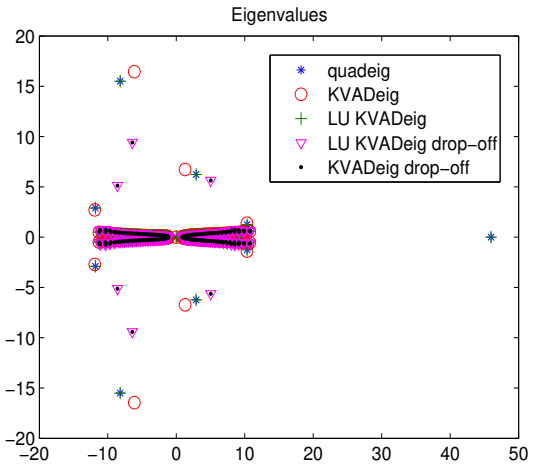
(a) Singular values of M **(b)** Singular values of K

Figure 3.12: Singular values of the coefficient matrices M and K in the scaled `dirac` example

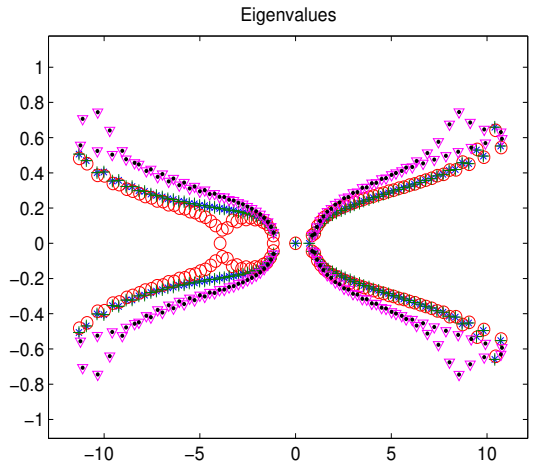


(a) Componentwise backward error

(b) Normwise backward error



(c) Eigenvalues



(d) Zoomed part of the spectrum

Figure 3.13: Comparison of the componentwise backward error, normwise backward errors, and the spectrum for the scaled `dirac` problem

There is a difference in result depending on the rank determination. In the first case, `quadeig` will deflate 1 zero and 1 infinite eigenvalue, just as LU based `KVADeig`, because it will be

determined that the matrices M and K have rank 79. In addition, `KVADeig` will deflate one more zero and infinite eigenvalue in the second step of the deflation process. This results in smaller norm-wise backward error in Figure 3.13b, compared to the other algorithms. In the second case, when we change the rank determination criteria to "drop-off", `KVADeig` and LU based `KVADeig` will not detect any zero or infinite eigenvalues. By looking just the norm-wise backward error, there is no big difference between the methods, however component-wise backward error represents the difference very well.

To see the importance of row sorting before the QR factorization, we scale only the matrix K so that the rows vary in norm, and we observe the reversed problem. There is a difference in rank determination as well. The first criterion deflates one infinite eigenvalue, that is the rank of M is declared as 79. For the second criterion, M is declared regular matrix. The singular values are presented in Figure 3.14. There is a difference in the component-wise error for rank revealing factorizations as well. This is presented in Table 3.4.

Table 3.4: Rank revealing factorization error, scaled reversed `dirac`

Method	Normwise error	Componentwise error
LU complete pivoting	`1.8958e-018`	`2.6949e-004`
QR column pivoting	`1.3928e-016`	`1.1128`
QR row sorting	`4.2503e-016`	`7.6146e-014`

The following figure shows the singular values of the scaled matrix

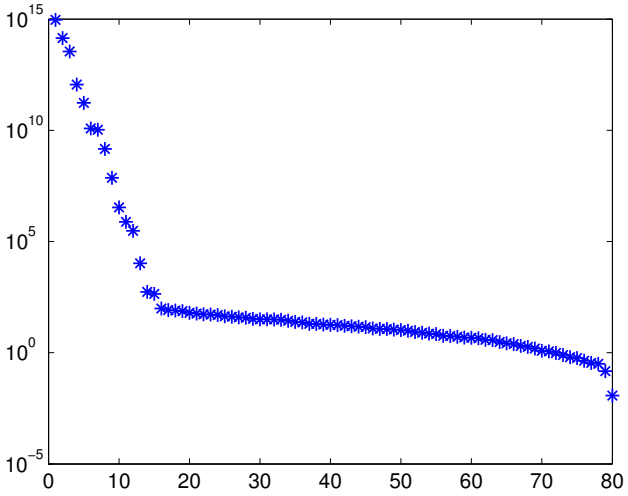
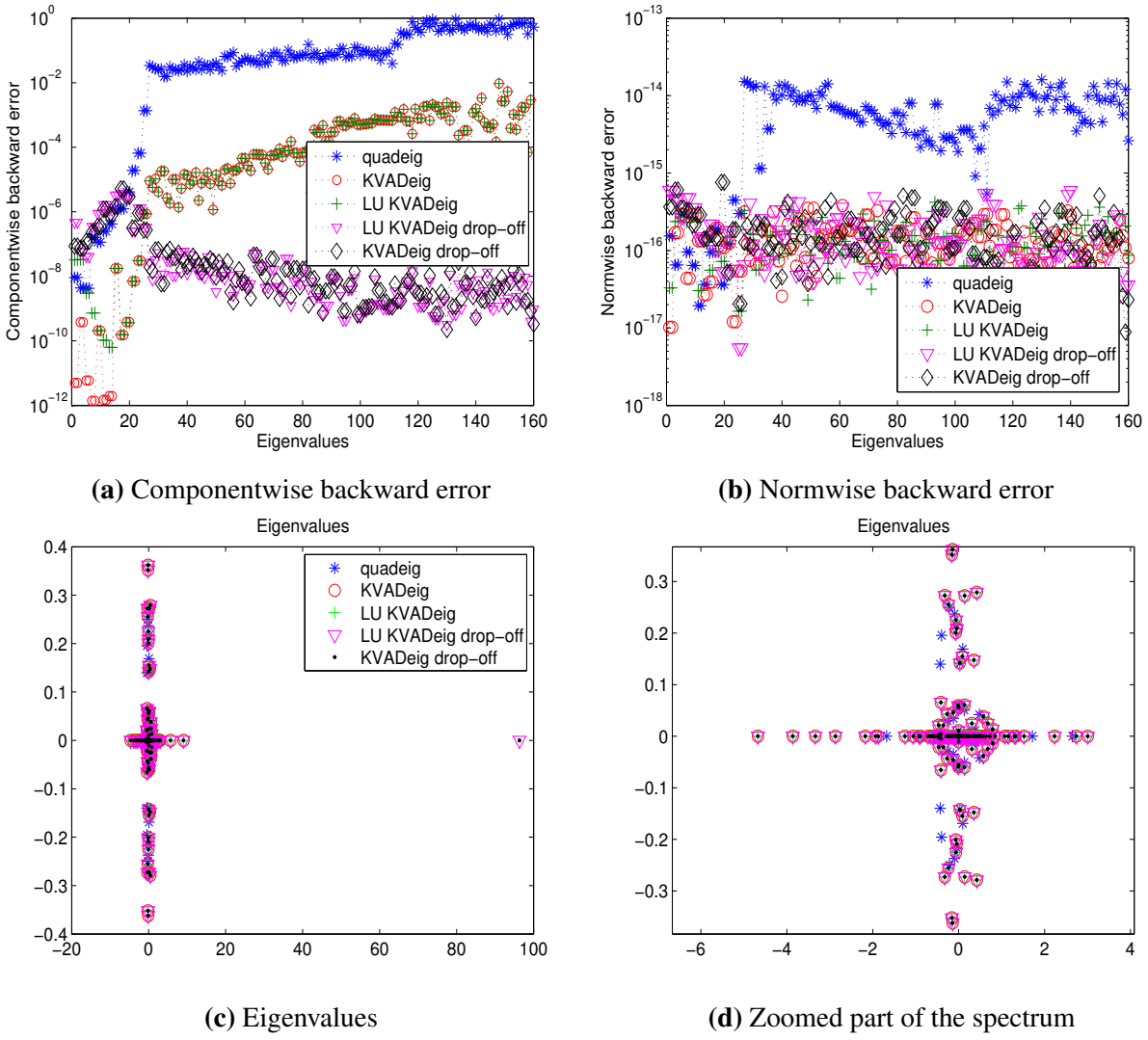


Figure 3.14: Singular values of leading coefficient matrix in scaled reversed `dirac` example

Again, the component-wise backward error in Figure 3.15a gives better insight in the difference and accuracy of the presented methods



(a) Componentwise backward error

(b) Normwise backward error

(c) Eigenvalues

(d) Zoomed part of the spectrum

Figure 3.15: Comparison of the componentwise backward error, normwise backward errors, and the spectrum for the scaled reversed `dirac` problem

3.7.3 Constrained least squares problem

Quadratically constrained least square problem

$$\min_x \|Ax-b\|_2^2, \quad \|x\|_2^2 = \delta^2, \tag{3.94}$$

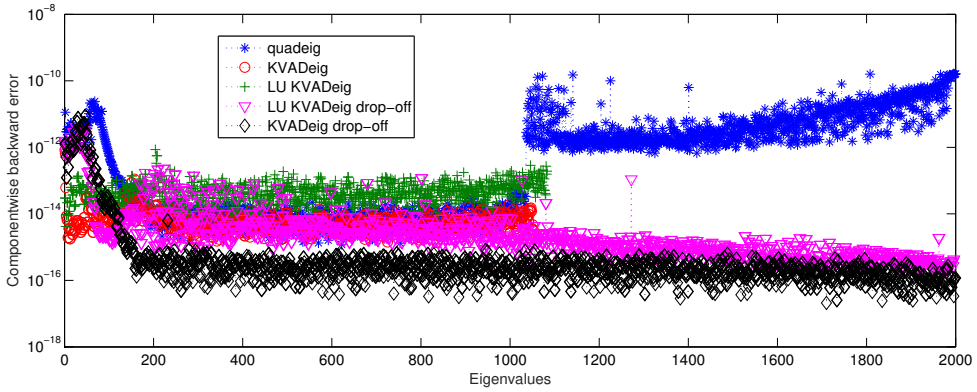
can be solved by the following quadratic eigenvalue problem

$$(\lambda^2 I + 2\lambda H + H^2 - \delta^{-2} gg^T)y = \mathbf{0}, \tag{3.95}$$

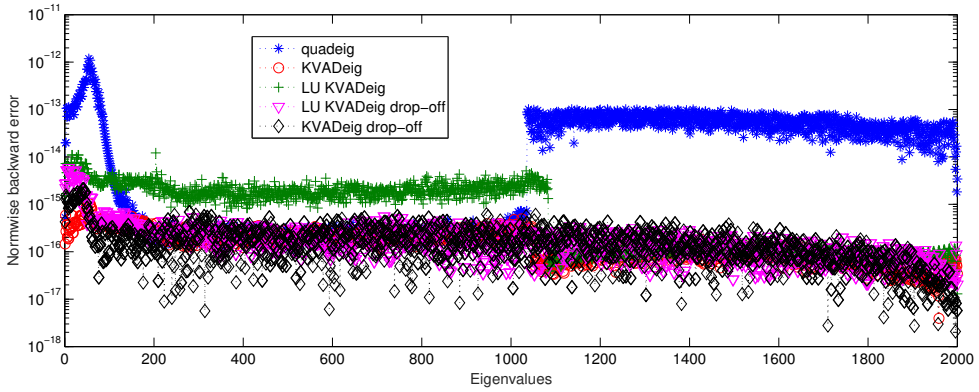
where $H = A^T A$ and $g = A^T b$. We will consider the reversed problem of order 100, so that we can compare various factorizations in the deflation process. The example `deriv2` is taken from the *Regularization Tools: A MATLAB package for Analysis and Solution of Discrete Ill-Posed Problems. Version 4.1.* In this example, the problem is the determination of the rank of

the matrix $H^2 - \delta^{-2} g g^T$. `LU KVADeig` will deflate 917 infinite eigenvalues, and `KVADeig` will deflate total $906+30+6+2+2+1 = 947$ infinite eigenvalues. If the second criterion for rank determination is used, then no infinite eigenvalue will be detected. The singular values of the leading coefficient matrix M are presented in Figure 3.17.

There is significant difference between `quadeig` and our methods, however the main difference in rank determination is detected by the componentwise backward error.



(a) Componentwise backward error



(b) Normwise backward error

Figure 3.16: Comparison of the componentwise backward errors, and normwise backward errors for the `deriv2` problem

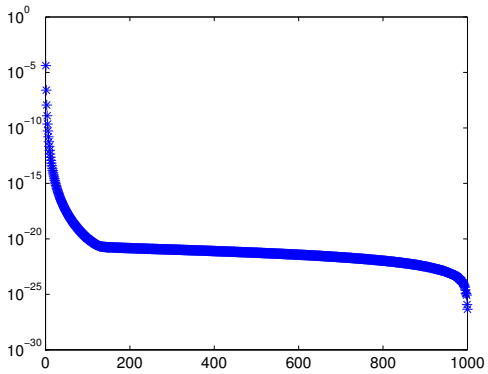


Figure 3.17: Singular values of the leading matrix coefficient in `deriv2` example

Chapter 4

Complete solution of the quartic eigenvalue problem

In this chapter we consider the polynomial eigenvalue problem of order 4, i.e. the *quartic eigenvalue problem*

$$(\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E)x = \mathbf{0}, \tag{4.1}$$

where $A, B, C, D, E \in \mathbb{C}^{n\times n}$. A important application of the quartic eigenvalue problem, as illustrated in §1.3.3, is in solving the Orr–Sommerfeld equation which appears in the analysis of the stability of the Poiseuille flow. Our goal is an efficient and numerically robust algorithm for the complete solution of the problem (4.1).

The idea is to try to use the algebraic tool of *quadratification* introduced by De Terán, Dopico and Mackey [17]. Quadratification is a equivalence relation that allows us to reduce the quartic problem (4.1) to an equivalent quadratic eigenvalue problem, which is then solved following the development from the previous chapters. Moreover, in our proposed approach, we try to use the original matrix coefficients as much as possible. Also, we develop a test for the existence of Jordan blocks for zero and infinite eigenvalues, and develop an algorithm for the complete determination of the structure of these eigenvalues.

The numerical experiments, presented in §4.5, show the power of our method in comparison to the MATLAB's function for the computation of the polynomial eigenvalue problem, `poyleig`, and to the `quadeig` as well. For instance, `polyeig` completely fails to find the solution of the quartic eigenvalue problem obtained from Orr–Sommerfeld equation of the dimension $n = 1000$, whereas our algorithm provides the solution with acceptable backward error.

4.1 Quadratification

Let us first briefly introduce the quadratification [17], and the notions of unimodulary equivalent matrix polynomials, and spectrally equivalent matrix polynomials.

Definition 4.1. *Suppose P and Q are two matrix polynomials of degrees g and h, respectively, not necessarily of the same size.*

- *P and Q are said to be extended unimodularly equivalent, denoted $P \backsim Q$, if for some $r,s \geq 0$ we have* $\operatorname{diag}(P, I_r) \sim \operatorname{diag}(Q, I_s)$.
- *P and Q are said to be spectrally equivalent, denoted $P \asymp Q$, if $P \backsim Q$ and* $\operatorname{rev} P \backsim \operatorname{rev} Q$.

Notice that unimodular equivalence corresponds to "being linearization", and spectral equivalence to "being strong linearization". This is clearer if we define these notions in the terms of the previous Definition (4.1).

Definition 4.2. *Let $P(\lambda)$ be an $m \times n$ matrix polynomial of degree g.*

- *A matrix pencil $L(\lambda)$ is said to be a linearization of $P(\lambda)$ if $L(\lambda) \backsim P(\lambda)$. A linearization is said to be strong if, in addition,* $\operatorname{rev} L(\lambda) \backsim \operatorname{rev} P(\lambda)$. *Equivalently, a pencil $L(\lambda)$ is a strong linearization for $P(\lambda)$ if*
$$L(\lambda) \asymp P(\lambda).$$
- *A quadratic matrix polynomial $Q(\lambda)$, i.e. a polynomial of degree 2, is said to be a quadratification of $P(\lambda)$ if $Q(\lambda) \backsim P(\lambda)$. A quadratification is said to be strong if, in addition,* $\operatorname{rev} Q(\lambda) \backsim \operatorname{rev} P(\lambda)$. *Equivalently, a pencil $L(\lambda)$ is a strong quadratification for $P(\lambda)$ if*
$$Q(\lambda) \asymp P(\lambda).$$

We will be interested in the strong quadratification because they preserve the structure of both finite and infinite eigenvalues (Theorem 4.1. in [17]).

4.1.1 Companion form of grade 2

Analogously to the linearization by companion form, the first and the second companion form of grade 2 are introduced in [17] as follows. First, define matrix polynomials

$$B_1(\lambda) = \lambda^2 C + \lambda D + E, \tag{4.2}$$
$$B_2(\lambda) = \lambda^2 A + \lambda B. \tag{4.3}$$

The first companion form of grade 2 is defined as

$$\begin{aligned} C_1^2(\lambda) &= \begin{pmatrix} B_2(\lambda) & B_1(\lambda) \\ -\mathbb{I}_n & \lambda^2 \mathbb{I}_n \end{pmatrix} = \begin{pmatrix} \lambda^2 A + \lambda B & \lambda^2 C + \lambda D + E \\ -\mathbb{I}_n & \lambda^2 \mathbb{I}_n \end{pmatrix} \\ &= \lambda^2 \begin{pmatrix} A & C \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} + \lambda \begin{pmatrix} B & D \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & E \\ -\mathbb{I}_n & \mathbf{0} \end{pmatrix}. \end{aligned} \tag{4.4}$$

We will use the second companion form of grade 2, because its structure is more convenient for the deflation process:

$$\begin{aligned} C_2^2(\lambda) &= \begin{pmatrix} B_2(\lambda) & -\mathbb{I}_n \\ B_1(\lambda) & \lambda^2\mathbb{I}_n \end{pmatrix} = \begin{pmatrix} \lambda^2 A + \lambda B & -\mathbb{I}_n \\ \lambda^2 C + \lambda D + E & \lambda^2\mathbb{I}_n \end{pmatrix} \\ &= \lambda^2 \begin{pmatrix} A & \mathbf{0} \\ C & \mathbb{I}_n \end{pmatrix} + \lambda \begin{pmatrix} B & \mathbf{0} \\ D & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & -\mathbb{I}_n \\ E & \mathbf{0} \end{pmatrix} \\ &= \lambda^2\mathbb{M} + \lambda\mathbb{C} + \mathbb{K}. \end{aligned} \tag{4.5}$$

It can be proved that these quadratifications are strong in the sense of Definition 4.2 (see [17]). The quadratic eigenvalue problem (4.5) can be solved by a corresponding algorithm, based on e.g. the second companion form linearization. In that case, the final matrix pencil of size $4n \times 4n$, that represents a linearization of the quartic problem 4.1, is

$$\mathbb{A} - \lambda\mathbb{B} = \left(\begin{array}{cc|cc} B & \mathbf{0}_n & -\mathbb{I}_n & \mathbf{0}_n \\ D & \mathbf{0}_n & \mathbf{0}_n & -\mathbb{I}_n \\ \hline \mathbf{0}_n & -\mathbb{I}_n & \mathbf{0}_n & \mathbf{0}_n \\ E & \mathbf{0}_n & \mathbf{0}_n & \mathbf{0}_n \end{array}\right) - \lambda \left(\begin{array}{cc|cc} -A & \mathbf{0}_n & \mathbf{0}_n & \mathbf{0}_n \\ -C & -\mathbb{I}_n & \mathbf{0}_n & \mathbf{0}_n \\ \hline \mathbf{0}_n & \mathbf{0}_n & -\mathbb{I}_n & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{0}_n & \mathbf{0}_n & -\mathbb{I}_n \end{array}\right). \tag{4.6}$$

By the same reasoning as before, we can conclude that the deflation Algorithm 3.2.1 completely determines the structure for zero and infinite eigenvalues of the quartic problem. The key is that the quadratification (4.5) is strong, meaning that the partial multiplicities for these eigenvalues are preserved. Moreover, the linearization (4.6) for the obtained quadratic problem is also strong, hence the conclusion follows by transitivity.

Theorem 4.1. *Algorithm 3.2.1 applied to pencil (4.6) completely determines the structure of eigenvalue zero for quartic eigenvalue problem* $(\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E)x = \mathbf{0}$.

4.2 Scaling

In order to equilibrate the norms for the coefficient matrices in (4.1), we propose two types of scalings, to be applied directly to the coefficient matrices A, B, C, D and E. The first one is tropical scaling, as described in §2.3.2, and the second one is a generalization of the Fan, Lin and Van Dooren's scaling for the quadratic eigenvalue problem. Here, we use the result from [4], which provides a unique minimizer of the coefficient

$$\frac{\max(1, \max_i \|A_i\|_2)^2}{\min(\|A_0\|_2, \|A_k\|_2)},$$

in the bound for the backward error of the matrix polynomial and the corresponding linearization. In addition, the parameter δ is defined as proposed in [14].

4.2.1 Tropical scaling.

The corresponding tropical polynomial for the quartic problem reads

$$tp(x) = \|A\|_2 x^4 \oplus \|B\|_2 x^3 \oplus \|C\|_2 x^2 \oplus \|D\|_2 x \oplus \|E\|_2. \tag{4.7}$$

For the computation of the tropical roots of (4.7) we use the algorithm provided in [61].

The maximal number of distinct tropical roots is 4. Every root α_i defines one set of scaling parameters

$$\gamma_i = \alpha_i, \;\; \delta_i = (p(\alpha_i))^{-1}, \;\; i = 1,2,3,4. \tag{4.8}$$

Every set of the parameters improves the backward error for certain part of the spectrum, and the other eigenvalues do not have to be computed as accurately. This is why, for this type of scaling, the complete quartic eigenvalue problem would have to be solved four times, in order to deliver all $4n$ eigenvalues with small backward errors. However, if n is large, this is not very efficient, especially because we are in fact solving the generalized eigenvalue problem of size $4n$ four times. Thus, this type of scaling is practical only in the case of problems of small dimension n.

4.2.2 Fan, Lin, Van Dooren generalization scaling.

The second option is a generalization of the Fan, Lin and Van Dooren's scaling for the quadratic eigenvalue problem. For γ, we choose

$$\gamma = \sqrt[4]{\frac{\|E\|_2}{\|A\|_2}}, \tag{4.9}$$

which is the optimal γ for minimizing the factor

$$\frac{\max(1, \|A\|_2, \|B\|_2, \|C\|_2, \|D\|_2, \|E\|_2)^2}{\min(\|E\|_2, \|A\|_2)}, \tag{4.10}$$

in the backward error ratio bounds (2.26) and (2.27).

For δ, we choose

$$\delta = \frac{4}{\|E\|_2 + \|\gamma D\|_2 + \|\gamma^2 C\|_2 + \|\gamma^3 B\|_2}. \tag{4.11}$$

This scaling is used in all our experiments.

4.3 Deflation process

If the leading coefficient matrix A has rank $r_A = \text{rank}(A) < n$, then there are at least $n - r_A$ infinite eigenvalues of the quartic eigenvalue problem (4.1). Similarly, if the coefficient matrix E has rank $r_E = \text{rank}(E) < n$ there are at least $n - r_E$ zero eigenvalues. We want to remove those

eigenvalues from the linearization pencil before calling the QZ algorithm. The first step would is determination of the numerical rank by some rank revealing factorization. Let

$$A\Pi_A = Q_A R_A,\ \ R_A = \begin{pmatrix} \widehat{R}_A \\ \mathbf{0}_{n-r_A,n} \end{pmatrix}, \tag{4.12}$$

$$E\Pi_E = Q_E R_E,\ \ R_E = \begin{pmatrix} \widehat{R}_E \\ \mathbf{0}_{n-r_E,n} \end{pmatrix}. \tag{4.13}$$

be the rank revealing QR factorizations for A and E. We would like to use the transformation matrices from the deflation process as in `KVADeig`, described in Section 3.5, applied to the quadratic problem (4.5). These include the rank revealing factorizations of matrices $\mathbb{M}$ and $\mathbb{K}$ of order $2n$.

Consider the matrix $\mathbb{M}$ of order $2n$ from the quadratification (4.5)

$$\mathbb{M} = \begin{pmatrix} A & \mathbf{0} \\ C & \mathbb{I}_n \end{pmatrix}. \tag{4.14}$$

From its structure, it is clear that $\operatorname{rank}(\mathbb{M}) = n + \operatorname{rank}(A)$, meaning that it is enough to determine the singularity of the matrix A of order n to determine the singularity of $\mathbb{M}$. Moreover, by interchanging $n \times n$ block columns and rows we get

$$\Pi M \Pi = \begin{pmatrix} \mathbf{0} & \mathbb{I}_n \\ \mathbb{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} A & \mathbf{0} \\ C & \mathbb{I}_n \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbb{I}_n \\ \mathbb{I}_n & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_n & C \\ \mathbf{0} & A \end{pmatrix}. \tag{4.15}$$

Now, we can use (4.12) to get the rank revealing decomposition of the matrix $\mathbb{M}$.

$$\begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_A^* \end{pmatrix} \Pi M \Pi \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & P_A \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbb{I}_n \\ Q_A^* & \mathbf{0} \end{pmatrix} M \begin{pmatrix} \mathbf{0} & P_A \\ \mathbb{I}_n & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_n & CP_A \\ \mathbf{0} & R_A \end{pmatrix}. \tag{4.16}$$

Finally, the rank revealing factorization of $\mathbb{M}$ is given by

$$\mathbb{M}\Pi_M = Q_M R_M,\ \ Q_M = \left(\begin{array}{c|c} \mathbf{0} & Q_A \\ \hline \mathbb{I}_n & \mathbf{0} \end{array}\right),\ \ \Pi_M = \left(\begin{array}{c|c} \mathbf{0} & P_A \\ \hline \mathbb{I}_n & \mathbf{0} \end{array}\right),\ \ R_M = \left(\begin{array}{c|c} -\mathbb{I}_n & -CP_A \\ \hline \mathbf{0} & -R_A \end{array}\right). \tag{4.17}$$

Next, consider the matrix $\mathbb{K}$ of the quadratification (4.5)

$$\mathbb{K} = \begin{pmatrix} \mathbf{0} & \mathbb{I}_n \\ E & \mathbf{0} \end{pmatrix}. \tag{4.18}$$

Again, we conclude that $\operatorname{rank}(\mathbb{K}) = n + \operatorname{rank}(E)$, that is, to check the singularity of $\mathbb{K}$ of order $2n$, it is enough to check the singularity of the $n \times n$ matrix E. If we permute the $n \times n$ column

blocks we get

$$\mathbb{K}\Pi = \begin{pmatrix} \mathbf{0} & \mathbb{I}_n \\ E & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbb{I}_n \\ \mathbb{I}_n & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & E \end{pmatrix}. \tag{4.19}$$

Hence, we can use (4.13) to determine the rank revealing decomposition of the matrix $\mathbb{K}$

$$\begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_E^* \end{pmatrix} K\Pi \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & P_E \end{pmatrix} = \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & R_E \end{pmatrix}. \tag{4.20}$$

Finally, the rank revealing factorization of the matrix $\mathbb{K}$ is

$$\mathbb{K}\Pi_K = Q_K R_K,\ Q_K = \left(\begin{array}{c|c} \mathbb{I}_n & \mathbf{0} \\ \hline \mathbf{0} & Q_E \end{array}\right),\ \Pi_K = \left(\begin{array}{c|c} \mathbf{0} & \mathbb{I}_n \\ \hline \Pi_E & \mathbf{0} \end{array}\right),\ R_K = \left(\begin{array}{c|c} \mathbb{I}_n & \mathbf{0} \\ \hline \mathbf{0} & R_E \end{array}\right). \tag{4.21}$$

However, notice that the permutation of the column blocks only ensures that the matrix R_K is upper triangular. If this structure is not important for the process, we can skip the permutation step and just make the following transformation

$$\begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_E^* \end{pmatrix} K \begin{pmatrix} \Pi_E & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbb{I}_n \\ R_E & \mathbf{0} \end{pmatrix}. \tag{4.22}$$

Now, we can use the deflation process form `KVADeig` algorithm. The first step is the determination of the ranks of the matrices A and E to determine whether there are zero and infinite eigenvalues. Of course, there can be more than one Jordan block for both of these eigenvalues, and in that case we want to deflate all of them, and not only the first block as in `quadeig`. We will have a nice characterization for the existence of the Jordan blocks in terms of the matrices of the original problem, as for the quadratic eigenvalue problem.
Again, as in the `KVADeig`, there are three standard cases: both A and E regular; only one matrix is singular; and both A and E are singular.

Both matrices A and E regular. If both matrices are regular, we can use the factorization (4.17) to reduce the matrix $\mathbb{B}$ from (4.6) to upper triangular form, since this is already the first step of the QZ algorithm.

$$\begin{aligned} &\begin{pmatrix} Q_M^* & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \left\{ \begin{pmatrix} \mathbb{C} & -\mathbb{I}_n \\ \mathbb{K} & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -\mathbb{M} & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \Pi_M & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} \\ &= \left(\begin{array}{cc|cc} \mathbf{0} & DP_A & \mathbf{0} & -\mathbb{I}_n \\ \mathbf{0} & Q_A^* B P_A & -Q_A^* & \mathbf{0} \\ \hline -\mathbb{I}_n & \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{2n}$}} \\ \mathbf{0} & EP_A & & \end{array}\right) - \lambda \left(\begin{array}{cc|c} -\mathbb{I}_n & -CP_Q & \mathbf{0}_{2n} \\ \mathbf{0} & -R_A & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{2n}} & -\mathbb{I}_{2n} \end{array}\right). \end{aligned} \tag{4.23}$$

Only one matrix is singular. Assume first that E is singular, meaning that there are at least $n - r_E$ zero eigenvalues which must to be deflated. If there is only one Jordan block of zero eigenvalues, then only one step of deflation is needed, and we can use the structure of the linearization pencil to transform the matrix $\mathbb{B}$ to upper triangular form. This is done by the same transformation matrices as in (3.40)

$$\begin{pmatrix} Q_M^* & \mathbf{0} \\ \mathbf{0} & Q_K^* \end{pmatrix} \left\{ \begin{pmatrix} \mathbb{C} & -\mathbb{I}_n \\ \mathbb{K} & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -\mathbb{M} & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \Pi_M & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix}$$
$$= \left(\begin{array}{cc|cc} \mathbf{0} & DP_A & \mathbf{0} & -Q_E \\ \mathbf{0} & Q_A^* B P_A & -Q_A^* & 0 \\ \hline -\mathbb{I}_n & \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{2n}$}} \\ \mathbf{0} & R_E P_E^* P_A & & \end{array}\right) - \lambda \left(\begin{array}{cc|c} -\mathbb{I}_n & -CP_Q & \mathbf{0}_{2n} \\ \mathbf{0} & -R_A & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{2n}} & -\mathbb{I}_{2n} \end{array}\right). \tag{4.24}$$

In order to derive the condition for the existence of multiple Jordan blocks for zero eigenvalue, we must consider different transformation, as in (3.66)

$$\begin{pmatrix} Q_K^* & \mathbf{0} \\ \mathbf{0} & Q_K^* \end{pmatrix} \left\{ \begin{pmatrix} \mathbb{C} & -\mathbb{I}_n \\ \mathbb{K} & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -\mathbb{M} & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \mathbb{I}_{2n} & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix}$$
$$= \left(\begin{array}{cc|c} B & \mathbf{0} & -\mathbb{I}_{2n} \\ Q_E^* D & \mathbf{0} & \\ \hline 0 & -\mathbb{I}_n & \multirow{2}{*}{$\mathbf{0}_{2n}$} \\ \widehat{R}_E P_E^* & \mathbf{0} & \\ \mathbf{0} & \mathbf{0} & \end{array}\right) - \lambda \left(\begin{array}{cc|c} -A & \mathbf{0} & \mathbf{0}_{2n} \\ -Q_E^* C & -Q_E^* & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{2n}} & -\mathbb{I}_{2n} \end{array}\right). \tag{4.25}$$

The deflated pencil of order $3n + r_E$ reads

$$\mathbb{A}_{22} - \lambda \mathbb{B}_{22} = \left(\begin{array}{cc|cc} B & \mathbf{0} & -\mathbb{I}_n & \\ Q_{E,1}^* D & \mathbf{0} & & -\mathbb{I}_{r_E} \\ Q_{E,2}^* D & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline 0 & -\mathbb{I}_n & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{n+r_E}$}} \\ \widehat{R}_E P_E^* & \mathbf{0} & & \end{array}\right) - \lambda \left(\begin{array}{cc|c} -A & \mathbf{0} & \mathbf{0}_{(2n)\times(n+rE)} \\ -Q_E^* C & -Q_E^* & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{(n+r_E)\times(2n)}} & -\mathbb{I}_{n+rE} \end{array}\right), \tag{4.26}$$

where $Q_{E,1}^* = Q_E^*(1 : r_E, :)$ and $Q_{E,2}^* = Q_E^*(r_E + 1 : n, :)$. The next step in the deflation process is to determine the rank of the matrix $\mathbb{A}_{22}$. From the structure of the matrix, we conclude that the rank of $\mathbb{A}_{22}$ is equal to $2n + r_E +$ *the rank of the* $n \times n$ *matrix*

$$\left(\begin{array}{c} Q_{E,2}^* D \\ \hline \widehat{R}_E P_E^* \end{array}\right). \tag{4.27}$$

Therefore, the test for the existence of Jordan blocks for the quartic problem (4.1) is to determine the rank of the $n \times n$ matrix (4.27) which is defined in terms of the coefficient matrices A and E of the original problem.

Notice that, if the matrix A is rank deficient, we can consider the reversed problem $(\mu^4 E + \mu^3 D + \mu^2 C + \mu B + A)x = \mathbf{0}$, $\mu = 1/\lambda$, and the corresponding truncated linearization pencil of order $3n + r_A$ reads

$$\mathbb{A}_{22} - \lambda \mathbb{B}_{22} = \left(\begin{array}{cc|cc} D & \mathbf{0} & -\mathbb{I}_n & \\ Q_{A,1}^* B & \mathbf{0} & & -\mathbb{I}_{r_A} \\ Q_{A,2}^* B & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline 0 & -\mathbb{I}_n & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{n+r_A}$}} \\ \widehat{R}_A P_A^* & \mathbf{0} & & \end{array}\right) - \lambda \left(\begin{array}{cc|c} -E & \mathbf{0} & \mathbf{0}_{(2n)\times(n+rA)} \\ -Q_A^* C & -Q_A^* & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{(n+r_A)\times(2n)}} & -\mathbb{I}_{n+rA} \end{array}\right), \tag{4.28}$$

and the rank of matrix $\mathbb{A}_{22}$ is now $2n + r_A +$ *the rank of the* $n \times n$ *matrix*

$$\left(\frac{Q_{A,2}^* B}{\widehat{R}_A P_A^*}\right). \tag{4.29}$$

Finally, we can prove proposition analogous to Proposition (3.4) for quadratic case

Proposition 4.1. *Assume that matrix E in the quartic pencil $\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E$ has rank* $\operatorname{rank}(E) = r_E < n$. *There exists more than one Jordan block for eigenvalue zero if*

$$(\ker(D) \cup \mathscr{X}) \cap \ker(E) \neq \{0\}, \;\; \mathscr{X} = \{y \in \mathbb{C}^n : Dy = z,\, z \in \operatorname{Im}(E)\}. \tag{4.30}$$

Analogously, if the matrix A has rank $\operatorname{rank}(A) = r_A < n$, *there are more than one Jordan block for infinite eigenvalue if*

$$(\ker(B) \cup \mathscr{Y}) \cap \ker(A) \neq \{0\}, \;\; \mathscr{Y} = \{y \in \mathbb{C}^n : By = z,\, z \in \operatorname{Im}(A)\}. \tag{4.31}$$

Proof. From Theorem 4.1 we know that the partial multiplicities, and thus the dimensions of Jordan blocks for a quartic eigenvalue problem can be obtained using Algorithm 3.2.1 for a corresponding strong linearization 4.6. The very first step of the deflation yields the pencil (4.26). Now, if $\mathbb{A}_{22}$ is singular, we will have another Jordan block for the eigenvalue zero. The rank of the matrix $\mathbb{A}_{22}$ can be determined by the rank of matrix $\left(\frac{Q_{E,2}^* D}{\widehat{R}_E P_E^T}\right)$. This matrix is rank deficient if its kernel is nontrivial, that is if $\ker\left(\frac{Q_{E,2}^* D}{\widehat{R}_E P_E^T}\right) = \ker(Q_{E,2} D) \cap \ker(\widehat{R}_E P_E^T) \neq \{0\}$. Matrix $Q_{E,2}$ represents the basis for $\ker(E^*)$, and thus

$$\ker\left(\frac{Q_{E,2}^* C}{\widehat{R}_E P_E^T}\right) = \Big(\ker(D) \cup (\operatorname{Im}(D) \cap \ker(E^*)^{\perp})\Big) \cap \ker(E). \tag{4.32}$$

□

Denote the left and the right transformation matrices from (4.25) with $\mathbf{P}_1$ and $\mathbf{Q}_1$ respectively, and the linearization pencil with $\mathbb{A}-\lambda\mathbb{B}=\mathbb{A}_{11}-\lambda\mathbb{B}_{11}$. After the first deflation step we have

$$\mathbf{P}_1(\mathbb{A}_{11}-\lambda\mathbb{B}_{11})\mathbf{Q}_1=\begin{pmatrix}\mathbb{A}_{22}-\lambda\mathbb{B}_{22} & \spadesuit\\ \mathbf{0} & -\lambda\mathbb{B}_{11}\end{pmatrix}. \tag{4.33}$$

Compute the rank revealing factorization

$$\left(\begin{array}{c}Q_{E,2}^*D\\ \hline \widehat{R}_E P_E^*\end{array}\right)\Pi_{A_{22}}=Q_{A_{22}}R_{A_{22}}. \tag{4.34}$$

If this matrix is singular, in order to deflate additional zeros, the first step is to permute the rows to get this matrix in the lower left corner of the matrix $\mathbb{A}_{22}$. This is done by the permutation $\pi=\begin{pmatrix}1:n+r_E & 2n+1:3n & n+r_E+1:2n & 3n+1:4n\end{pmatrix}$ (denote with Π the corresponding permutation matrix). Now, the transformation matrix $\widehat{P}_2$ is given by

$$\widehat{P}_2=\begin{pmatrix}\mathbb{I}_{2n+r_E} & \\ & Q_{A_{22}}^*\end{pmatrix}\Pi, \tag{4.35}$$

and the transformed pencil is

$$\widehat{P}_2\mathbb{A}_{22}=\left(\begin{array}{cc|cc}B & \mathbf{0} & -\mathbb{I}_n & \\ \hline Q_{E,1}^*D & \mathbf{0} & & -\mathbb{I}_{r_E}\\ \mathbf{0} & -\mathbb{I}_n & \mathbf{0} & \mathbf{0}\\ \hline \widehat{R}_{A_{22}}\Pi_{A_{22}}^T & \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{n\times(n+r_E)}$}}\\ \mathbf{0} & \mathbf{0} & & \end{array}\right),\quad \widehat{P}_2\mathbb{B}_{22}=\left(\begin{array}{cc|c}-A & \mathbf{0} & \mathbf{0}_{n+r_E}\\ \hline -Q_{E,1}^*C & -Q_{E,1}^* & \\ \hline \mathbf{0} & & \mathbb{I}_n \quad \mathbf{0}\\ \square & & \triangle \quad \blacktriangle\end{array}\right). \tag{4.36}$$

To be able to deflate additional zeros, we have to reduce the blocks □ and △ to zero. This is done by the complete orthogonal decomposition

$$\widehat{P}_2\mathbb{B}_{22}=U_{BB}R_{BB}V_{BB}^*, \tag{4.37}$$

so that $\widehat{P}_2\mathbb{B}_{22}V_{BB}=\begin{pmatrix}\mathbb{B}_{22} & \mathbf{0}\end{pmatrix}$. Denote by Π the permutation matrix for these column blocks. Finally, the deflated pencil is

$$\widehat{P}_2\mathbb{A}_{22}V_{BB}\Pi-\lambda\widehat{P}_2\mathbb{B}_{22}V_{BB}\Pi=\begin{pmatrix}\mathbb{A}_{33}-\lambda\mathbb{B}_{33} & \blacksquare\\ \mathbf{0} & -\lambda\mathbb{B}_{22}\end{pmatrix}. \tag{4.38}$$

Since we have lost the structure of the original linearization, the potential further deflation process is done by Algorithm 3.5.1 on the pencil $\mathbb{A}_{33}-\lambda\mathbb{B}_{33}$.

Both matrices A and E are singular. When both matrices A and E are rank deficient, and we determined that there are no more Jordan blocks for zero and infinite eigenvalues by computing the numerical rank of block matrices

$$\left(\frac{Q_{E,2}^* D}{\widehat{R}_E P_E^*}\right), \quad \left(\frac{Q_{A,2}^* B}{\widehat{R}_A P_A^*}\right), \tag{4.39}$$

we can deflate them in one step, as done in `quadeig` algorithm. The transformation matrices would be

$$\begin{aligned}
&\begin{pmatrix} Q_M^* & \mathbf{0} \\ \mathbf{0} & Q_K^* \end{pmatrix} \left\{ \begin{pmatrix} C & -\mathbb{I}_n \\ K & \mathbf{0} \end{pmatrix} - \lambda \begin{pmatrix} -M & \mathbf{0} \\ \mathbf{0} & -\mathbb{I}_n \end{pmatrix} \right\} \begin{pmatrix} \mathbb{I}_n & \mathbf{0} \\ \mathbf{0} & Q_K \end{pmatrix} \\
&= \left(\begin{array}{cc|cc} \mathbf{0} & D & \mathbf{0} & -Q_E \\ \mathbf{0} & Q_A^* B & -Q_A^* & 0 \\ \hline -\mathbb{I}_n & \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{2n}$}} \\ \mathbf{0} & R_E P_E^* & & \end{array}\right) - \lambda \left(\begin{array}{cc|c} -\mathbb{I}_n & -C & \mathbf{0}_{2n} \\ \mathbf{0} & -R_A P_A^* & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{2n}} & -\mathbb{I}_{2n} \end{array}\right).
\end{aligned} \tag{4.40}$$

Notice that, in terms of the quadratic problem, we have $r_{\mathbb{M}} = n + r_A$, and $r_{\mathbb{K}} = n + r_E$, so if we want to make the partition (3.48) as before, we will have (in previous notation):

$$\left(\begin{array}{c|c} X_{11} & X_{12} \\ \hline X_{21} & X_{22} \end{array}\right) = \left(\begin{array}{cc|c} \mathbf{0}_n & D(:,1:r_A) & D(:,r_A+1:n) \\ \mathbf{0}_{r_A,n} & Q_A^*(1:r_A,1:r_A) & Q_A^*(1:r_A,r_A+1:n) \\ \hline \mathbf{0}_{n-r_A,n} & Q_A^*(r_A+1:n,1:r_A) & Q_A^*(r_A+1:n,r_A+1:n) \end{array}\right), \tag{4.41}$$

$$\left(\begin{array}{c|c} X_{13} & X_{14} \\ \hline X_{23} & X_{24} \end{array}\right) = \left(\begin{array}{cc|c} \mathbf{0}_n & -Q_E(:,1:r_E) & -Q_E(:,r_E+1:n) \\ -Q_A^*(1:r_A,:) & \mathbf{0}_{r_A,r_E} & \mathbf{0}_{r_A,n-r_E} \\ \hline -Q_A^*(r_A+1:n,:) & \mathbf{0}_{n-r_A,r_E} & \mathbf{0}_{n-r_A,n-r_E} \end{array}\right), \tag{4.42}$$

$$\left(\begin{array}{c|c} X_{31} & X_{32} \end{array}\right) = \left(\begin{array}{cc|c} -\mathbb{I}_n & \mathbf{0}_{n,r_A} & \mathbf{0}_{n,n-r_A} \\ \mathbf{0}_{r_E,n} & \widehat{R}_E P_E(:,1:r_A) & \widehat{R}_E P_E(:,r_A+1:n) \end{array}\right), \tag{4.43}$$

$$\left(\begin{array}{c|c} Y_{11} & Y_{12} \end{array}\right) = \left(\begin{array}{cc|c} -\mathbb{I}_n & -C(:,1:r_A) & -C(:,r_A+1:n) \\ \mathbf{0}_{r_A,n} & -\widehat{R}_A P_A(:,1:r_A) & -\widehat{R}_A P_A(:,r_A+1:n) \end{array}\right). \tag{4.44}$$

The rest of the process goes as in Subsection 3.3.2.

4.3.1 Backward error analysis for the deflation process

In this section, we develop a backward error analysis for the first two steps of the deflation process, described in the previous section. The following proposition deals with the first step, that is, the deflation of the first $n - r_E$ zero eigenvalues.

Proposition 4.2. *Let*

$$\widetilde{\mathbb{A}}-\lambda\widetilde{\mathbb{B}}=\left(\begin{array}{cc|c} B & \mathbf{0} & -\mathbb{I}_n & \mathbf{0} \\ \widetilde{X}_{11} & \mathbf{0} & \mathbf{0} & -\mathbb{I}_{\widetilde{r}_E} \\ & & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbb{I}_n & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{n+\widetilde{r}_E}$}} \\ \widetilde{R}_E\widetilde{\Pi}_E^T & \mathbf{0} & & \end{array}\right)-\lambda\left(\begin{array}{cc|c} -A & \mathbf{0} & \multirow{2}{*}{$\mathbf{0}_{2n\times(n+\widetilde{r}_E)}$} \\ -\widetilde{Y}_{12} & -\widehat{Q}_E^T & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{(n+\widetilde{r}_E)\times 2n}} & -\mathbb{I}_{n+\widetilde{r}_E} \end{array}\right) \tag{4.45}$$

be the computed linearization (4.25). Then it corresponds to exact reduced linearization of a quartic pencil $\lambda^4A+\lambda^3B+\lambda^2(C+\delta C)+\lambda(D+\delta D)+(E+\delta E+\Delta E)$*, where, for all* $i=1,\ldots,n$*,*

$$\|\delta C(:,i)\|_2\leq\varepsilon_C\|C(:,i)\|_2,\ \ \|\delta D(:,i)\|_2\leq\varepsilon_D\|D(:,i)\|_2,\ \ \|\delta E(:,i)\|_2\leq\varepsilon_{qr}\|E(:,i)\|_2; \tag{4.46}$$

and the truncation error is

$$\max_{j=1:n-k}\|(\Delta E)\widetilde{\Pi}_E(:,k+j)\|_2\leq\tau\min_{i=1:k}\|(E+\delta E)\widetilde{\Pi}_E(:,i)\|_2;\ \ (\Delta E)\widetilde{\Pi}_E(:,1:k)=\mathbf{0}_{n,k}, \tag{4.47}$$

with τ *is prescribed threshold parameter.*

Proof: (i) It holds that $\widetilde{X}_{11}=computed(\widetilde{Q}_E^*D)=\widehat{Q}_E^*(D+\delta D)$. To estimate δD, we start with the fact that

$$computed(\widetilde{Q}_E^*D)=\widetilde{Q}_E^*D+\mathfrak{D},\ \ |\mathfrak{D}|\leq\varepsilon_{\mathfrak{D}}|\widetilde{Q}_E^*||D|,\ \ 0\leq\varepsilon_{\mathfrak{D}}\leq 2n\mathbf{u}$$

Since $\widetilde{Q}_E=(\mathbb{I}+\mathfrak{E})\widehat{Q}_E$, $\|\mathfrak{E}\|_2\leq\varepsilon_{qr}$, we have

$$computed(\widetilde{Q}_E^*D)=\widehat{Q}_E^*(\mathbb{I}+\mathfrak{E}^*)D+\mathfrak{D}=\widehat{Q}_E^*(D+\mathfrak{E}^*D+\widehat{Q}_E\mathfrak{D})\equiv\widehat{Q}_E^*(D+\delta D)$$

with column-wise estimates $\|\delta D(:,i)\|_2\leq(\|\mathfrak{E}^*\|_2+\varepsilon_{\mathfrak{D}}n(1+\|\mathfrak{E}^*\|_2))\|D(:,i)\|_2$ (derived as in Proposition 3.3), and (4.46) follows with $\varepsilon_D=(\varepsilon_{qr}+\varepsilon_{\mathfrak{D}}n(1+\varepsilon_{qr}))$.

(ii) By the same reasoning we get $\widetilde{Y}_{21}=\widehat{Q}_E(C+\delta C)$, where $\|\delta C(:,i)\|_2\leq\varepsilon_C\|C(:,i)\|_2$, and $\varepsilon_C=(\varepsilon_{qr}+\varepsilon_{\mathfrak{M}}\sqrt{n}(1+\varepsilon_{qr}))$.

(iii) Note that in this moment the backward error in E contains both the floating point error δE and the truncation error ΔE analogous to (3.38), i.e. $(E+\delta E+\Delta E)\widetilde{\Pi}_E=\widehat{Q}_E\widetilde{R}_E$. If we set $\Delta_\Sigma E=\delta E+\Delta E$, then we can represent the computed linearization as

$$\left(\begin{array}{cccc}\mathbb{I}_n&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\widehat{Q}_E^*&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbb{I}_n&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\widehat{Q}_E^*\end{array}\right)\left\{\left(\begin{array}{cccc}B&\mathbf{0}_n&-\mathbb{I}_n&\mathbf{0}\\ D+\delta D&\mathbf{0}_n&\mathbf{0}_n&-\mathbb{I}_n\\ 0&-\mathbb{I}_n&\mathbf{0}_n&\mathbf{0}_n\\ E+\Delta_\Sigma E&\mathbf{0}_n&\mathbf{0}_n&\mathbf{0}_n\end{array}\right)-\lambda\left(\begin{array}{cccc}-A&\mathbf{0}_n&\mathbf{0}_n&\mathbf{0}_n\\ -(C+\delta C)&-\mathbb{I}_n&\mathbf{0}_n&\mathbf{0}_n\\ \mathbf{0}_n&\mathbf{0}_n&-\mathbb{I}_n&\mathbf{0}_n\\ \mathbf{0}_n&\mathbf{0}_n&\mathbf{0}_n&-\mathbb{I}_n\end{array}\right)\right\}\left(\begin{array}{cccc}\mathbb{I}_n&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbb{I}_n&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbb{I}_n&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\widehat{Q}_E\end{array}\right).$$

■

The next step is computation of the rank revealing factorization of the block matrix

$$\begin{pmatrix} \widehat{Q}_E^*(D+\delta D) \\ \widetilde{R}_E \widetilde{\Pi}_E^T \end{pmatrix} \Pi_{A_{22}} = Q_{A_{22}} R_{A_{22}}. \tag{4.48}$$

For the computed factors $\widetilde{\Pi}_{A_{22}}, \widetilde{Q}_{A_{22}}, \widetilde{R}_{A_{22}}$ it holds that

$$\left[\begin{pmatrix} \widehat{Q}_E^*(D+\delta D) \\ \widetilde{R}_E \widetilde{\Pi}_E^T \end{pmatrix} + \begin{pmatrix} \mathfrak{D} \\ \mathfrak{E} \end{pmatrix} \right] \widetilde{\Pi}_{A_{22}} = \widehat{Q}_{A_{22}} \widetilde{R}_{A_{22}}, \tag{4.49}$$

where

$$\left\| \begin{pmatrix} \mathfrak{D} \\ \mathfrak{E} \end{pmatrix} (:,i) \right\|_2 \leq \varepsilon_{qr} \left\| \begin{pmatrix} \widehat{Q}_E^*(D+\delta D) \\ \widetilde{R}_E \widetilde{\Pi}_E^T \end{pmatrix} (:,i) \right\|_2 . \tag{4.50}$$

By an analogous procedure to the one in Subsection 3.5.2 we get the final estimate

$$\begin{aligned} \frac{\|\mathfrak{D}(:,i)\|_2}{\|D(:,i)\|_2} &\leq \varepsilon_{qr}\sqrt{2}\max\left((1+\varepsilon_D)\cos\sphericalangle(\ker(E)+\mathrm{Im}(D)), (1+\varepsilon_{qr})\frac{\|E(:,i)\|_2}{\|D(:,i)\|_2} \right), \\ \frac{\|\mathfrak{E}(:,i)\|_2}{\|E(:,i)\|_2} &\leq \varepsilon_{qr}\sqrt{2}\max\left((1+\varepsilon_D)\cos\sphericalangle(\ker(E)+\mathrm{Im}(D))\frac{\|D(:,i)\|_2}{\|E(:,i)\|_2}, (1+\varepsilon_{qr}) \right). \end{aligned}$$

4.3.2 Eigenvector recovery

The right and the left eigenvectors of the original problem (4.1) and the final linearization pencil (4.6) are related as follows. Let $z \in \mathbb{C}^{4n}$ and $w \in \mathbb{C}^{4n}$ be the right and left eigenvector for the linearization, and $x \in \mathbb{C}^n$, $y \in \mathbb{C}^n$ the right and left eigenvector for the original problem, and $\lambda \in \mathbb{C}$ the corresponding eigenvalue. If we partition $z = \begin{pmatrix} z_1^T & z_2^T & z_3^T & z_4^T \end{pmatrix}^T$ and $w = \begin{pmatrix} w_1^T & w_2^T & w_3^T & w_4^T \end{pmatrix}^T$, where $w_i, z_i \in \mathbb{C}^n, i = 1,2,3,4$, we have

$$z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \begin{pmatrix} \lambda x \\ \lambda^2(\lambda A + B)x \\ \lambda(\lambda A + B)x \\ \lambda E x \end{pmatrix}, \tag{4.51}$$

$$w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{pmatrix} \lambda^3 x \\ \lambda^2 x \\ \lambda x \\ x \end{pmatrix}. \tag{4.52}$$

For both the right and the left eigenvector there are four choices to recover x and y. Namely, for the right eigenvector we can choose $z_1, (\lambda A + B)^{-1} z_2, (\lambda A + B)^{-1} z_3$ or $E^{-1} z_4$. Notice that, for the last three choices we have to solve the system of the equations in order to compute the

wanted vector. Reconstruction of the left eigenvector is easier, though. We just choose one of the block components w_1, w_2, w_3 or w_4.
Let $\widetilde{z} \in \mathbb{C}^n$ and $\widetilde{w} \in \mathbb{C}^n$ be the computed right and left eigenvector for the linearization pencil (4.6). Both right and left eigevectors will have $4n$ elements if no deflation occurred, otherwise the number of elements will be $4n-d$, where d is the total number of zero and infinite eigenvalues deflated. $4n-d$ is also the dimension of the truncated pencil $\widetilde{\mathbb{A}} - \lambda\widetilde{\mathbb{B}} = P(\mathbb{A} - \lambda\mathbb{B})Q$ which is passed to the QZ algorithm for computation of finite nonzero eigenvalues.

No deflation occurred. The right and the left eigenvectors for the original linearization pencil are $z = Q\widetilde{z}$ and $w = P^T\widetilde{w}$. Now we choose x and y form the four choices. The criterion can be the smallest backward error.

Deflation occurred. In order to be able to recover eigenvectors we must have the full $4n$ vectors for the transformed problem. For the right eigenvector this is easy; we just add d zeros to the $\widetilde{z}$, that is $z = Q\begin{pmatrix} \widetilde{z} \\ \mathbf{0}_{d\times 1} \end{pmatrix}$. However, in the case of the deflation E and/or A is singular, so we just take the first n block as the right eigenvector of the original problem to avoid solving the system with a singular matrix.
Getting the he left eigenvector is more tricky. To obtain the full $4n$ eigenvector for the linearization, we first have to compute the missing d components of $\widetilde{w}$. Denote with $\widetilde{w}_1$ the eigenvector of the truncated problem, and let $\widetilde{w}_2$ be the missing part. From

$$\begin{pmatrix} \widetilde{w}_1^T & \widetilde{w}_2^T \end{pmatrix} P(\mathbb{A} - \lambda\mathbb{B}) = \begin{pmatrix} \widetilde{w}_1^T & \widetilde{w}_2^T \end{pmatrix} \begin{pmatrix} \widetilde{A} - \lambda\widetilde{B} & X \\ \mathbf{0} & Y \end{pmatrix} \tag{4.53}$$

we conclude that $\widetilde{w}_2 = -\widetilde{w}_1^* X Y^{-1}$. Now, $w = P^T\widetilde{w}$, and we choose one of the 4 block components as a left eigenvector for the original problem.
The right eigenvectors for zero (infinite) eigenvalues are computed as the last $n - r_E$ ($n - r_A$) columns of orthogonal matrix from the QR factorization of E^* (A^*), and the left as the last $n - r_E$ ($n - r_A$) columns of Q_E (Q_A).

Remark 4.1. Recall the Remark 3.6, where we stated that the structure of any eigenvalue α can be determined by the Algorithm 3.5.1 but with the shifted starting matrix $A_{1,1} = A - \alpha B$. Consider the linearization for the quartic eigenvalue problem (4.6). The shifted matrix $A_{1,1}$ is of form

$$A_{1,1} = \left(\begin{array}{cc|cc} B & \mathbf{0} & -\mathbb{I} & \mathbf{0} \\ D & \mathbf{0} & \mathbf{0} & -\mathbb{I} \\ \hline \mathbf{0} & -\mathbb{I} & \mathbf{0} & \mathbf{0} \\ E & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right) - \alpha \left(\begin{array}{cc|cc} -A & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -C & -\mathbb{I} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & -\mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbb{I} \end{array}\right) = \left(\begin{array}{cc|cc} B+\alpha A & \mathbf{0} & -\mathbb{I} & \mathbf{0} \\ D+\alpha C & \alpha\mathbb{I} & \mathbf{0} & -\mathbb{I} \\ \hline \mathbf{0} & -\mathbb{I} & \alpha\mathbb{I} & \mathbf{0} \\ E & \mathbf{0} & \mathbf{0} & \alpha\mathbb{I} \end{array}\right).$$

The first step in the Algorithm 3.5.1 is to determine the rank of $A_{1,1}$. Similarly as in Remark 3.6, we can conclude that the rank of the $4n \times 4n$ matrix $A_{1,1}$ can be determined by the rank of the $n \times n$ matrix $\alpha^4 A + \alpha^3 B + \alpha^2 C + \alpha D + E$ as $\operatorname{rank}(A) = 3n + \operatorname{rank}(\alpha^4 A + \alpha^3 B + \alpha^2 C + \alpha D + E)$. This follows from the following transformation

$$\left(\begin{array}{cc|cc} \alpha^3\mathbb{I} & \alpha\mathbb{I} & \alpha^2\mathbb{I} & \mathbb{I} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I} \end{array}\right) A_{1,1} = \left(\begin{array}{cc|cc} \alpha^4 A + \alpha^3 B + \alpha^2 C + \alpha D + E & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ D + \alpha C & \alpha\mathbb{I} & \mathbf{0} & -\mathbb{I} \\ \hline \mathbf{0} & -\mathbb{I} & \alpha\mathbb{I} & \mathbf{0} \\ E & \mathbf{0} & \mathbf{0} & \alpha\mathbb{I} \end{array}\right).$$

4.4 Deflation process in `KVARTeig` algorithm

We will refer to our algorithm for the complete solution of quartic eigenvalue problem (4.1) as `KVARTeig`. In this section we develop full deflation algorithm depending on the number of Jordan blocks that need to be deflated for both zero and infinite eigenvalues.
The first step is rank determination for the matrices A and E. Let $r_A = \operatorname{rank}(A)$ and $r_E = \operatorname{rank}(E)$. We have three main cases

1. **Both matrices A and E are regular**, i.e. $r_A = r_E = n$. In this case there is no deflation, we just use the rank revealing factorization for $\mathbb{M}$ (4.17) to reduce the matrix $\mathbb{B}$ to upper triangular form as in (4.23).

2. **One of the matrices is singular.** First assume that $r_E < n$. Then, before any deflation step, we determine the rank of the block $n \times n$ matrix $A_{22} := \left(\begin{array}{c} Q_{E,2}^* D \\ \hline \widehat{R}_E P_E^* \end{array}\right)$. However, if the matrix A is singular, we will consider the reversed problem, and the matrix A_{22} will be $A_{22} := \left(\begin{array}{c} Q_{A,2}^* B \\ \hline \widehat{R}_A P_A^* \end{array}\right)$. Nevertheless, the next step depends on the rank of A_{22}. In the continuation of this step we will talk only about deflation of the zero eigenvalue, because the infinite eigenvalues of our problem are the zero eigenvalues of the reversed problem.

   2.1 **Regular A_{22}.** If A_{22} is regular, there is just one Jordan block of zeros, and it is deflated as in (4.24), that is we also reduce the matrix $\mathbb{B}$ to upper triangular form.

   2.2 **Singular A_{22}.** In this case there is at least one more Jordan block for the zero eigenvalue. The first two blocks are deflated using the structure of the linearization pencil, as described in (4.25) and (4.36). At this point, we cannot use the structure of the pencil any more, and thus we send the derived pencil to Algorithm 3.5.1 to check whether there are more Jordan blocks and to deflate them.

3. **Both matrices A and E are singular.** Again, before any transformations of the linearization pencil, we must check whether there exist more Jordan blocks for the zero and the

infinite eigenvalues. This is done by determining the rank of the $n \times n$ matrices

$$\left(\frac{Q_{A,2}^* B}{\widehat{R}_A P_A^*} \right), \quad \left(\frac{Q_{E,2}^* D}{\widehat{R}_E P_E^*} \right). \tag{4.54}$$

After that, there are three possible outcomes

3.1 **Both matrices in (4.54) are regular.** In this case there is just one Jordan block of both zero and infinite eigenvalues, and they are deflated by one transformation as in (4.40).

3.2 **Only one matrix in (4.54) is singular.** This means that there are more than one Jordan blocks for zero or infinite eigenvalue. In either case, we deflate two Jordan blocks for the zero eigenvalue using the structure described in (4.25) and (4.36), meaning that the reversed problem is considered if there are more Jordan blocks for the infinite eigenvalues. After that, the pencil is sent to Algorithm 3.5.1 to check whether there are more Jordan blocks of zero and to deflate them. Finally, when all zeros are deflated, we send the reversed truncated linearization pencil to Algorithm 3.5.1 to deflate one Jordan block of the infinite eigenvalues. We do not check the rank for the number of infinite eigenvalues, but we use the information that there are exactly $n - r_A$, or $n - r_E$ if reversed pencil is considered, infinite eigenvalues.

3.3 **Both matrices in (4.54) are singular.** In this case there is more than one Jordan block for both zero and infinite eigenvalues. Depending which total sum of the dimensions of the first two Jordan blocks is greater, we consider original or the reversed problem. In either case, we use the structure to deflate two Jordan blocks of zero eigenvalue. After that, the truncated pencil is sent to Algorithm 3.5.1 to deflate possible remaining Jordan blocks of zero eigenvalues. Finally, when all zeros are deflated, we send the reversed truncated linearization pencil to Algorithm 3.5.1 together with the information about the size of the first two Jordan blocks, for which we know to exist, and need to be deflated. Any additional Jordan blocks will be determined by the algorithm.

At the end, we present the diagram for the decision three of the described algorithm

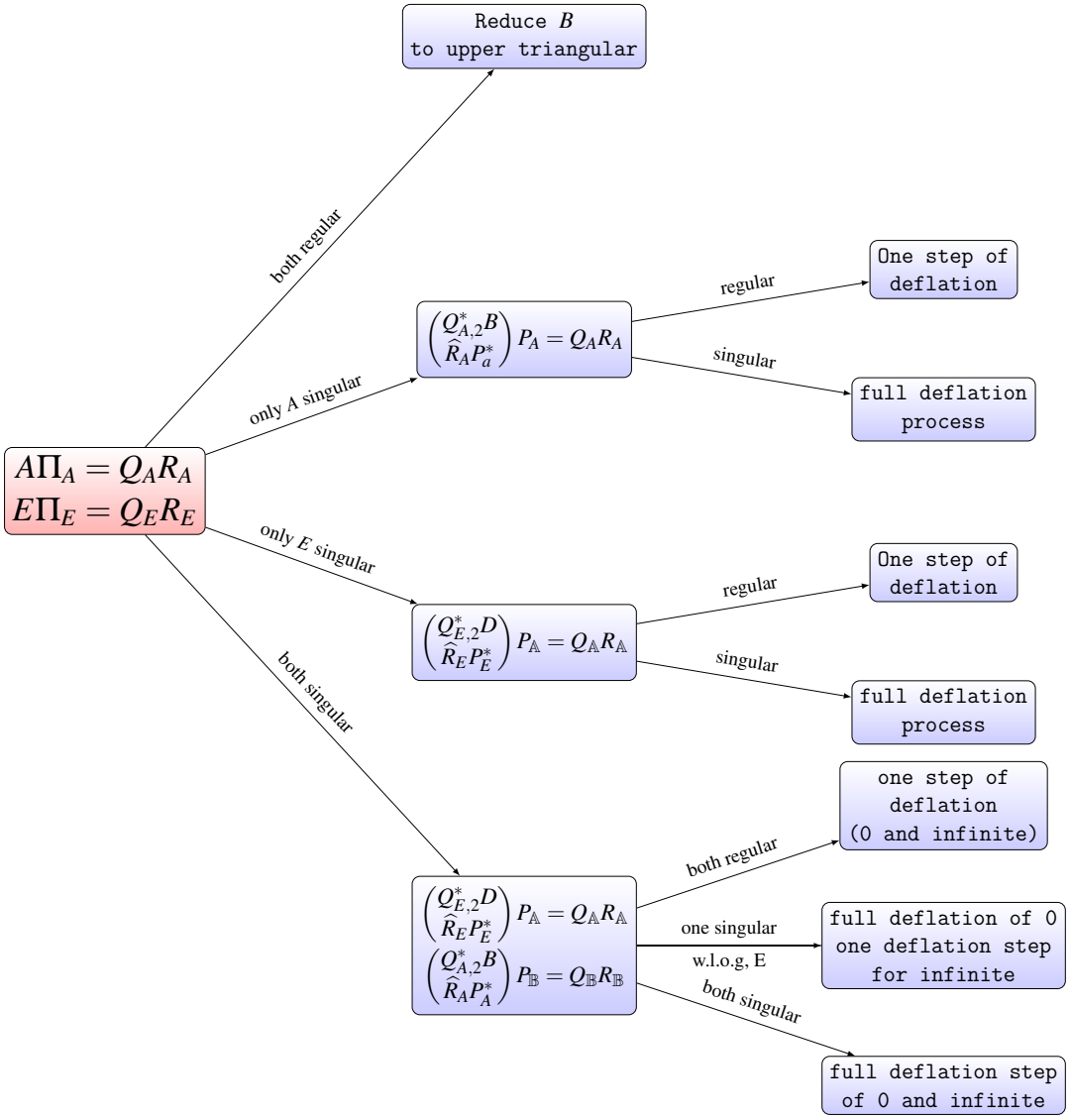


Figure 4.1: Decision tree for the deflation process in `KVARTeig`

4.5 Numerical experiments

In this section, we provide numerical examples that clearly illustrate the superiority of the new proposed algorithm, as compared with the two state of the art methods, the `polyeig` and `quadeig`.

Experiment 1. We tested our algorithm for three examples from NLEVP benchmark library for quartic eigenvalue problems: `butterfly`: $n = 64$; `orr_sommerfeld`: $n = 64$; and `planar waveguide`: $n = 129$.

We also computed the eigenvalues using the function `polyeig` from MATLAB, and the `quadeig`. The maximal backward errors are given in Table 4.1:

Table 4.1: Comparison of backward errors for `polyeig`, `quadeig` and `KVARTeig`

	`polyeig`		`quadeig`		`KVARTeig`	
Problem	$\min \eta$	$\max \eta$	$\min \eta$	$\max \eta$	$\min \eta$	$\max \eta$
`butterfly`	`2.0432e-016`	`8.6189e-016`	`2.5525e-016`	`2.0389e-015`	`5.8418e-017`	`1.1377e-015`
`orr_sommerfeld`	`1.3618e-017`	`8.0176e-006`	`2.1743e-014`	`4.0733e-004`	`6.3789e-021`	`1.7600e-015`
`planar waveguide`	`1.6060e-016`	`3.0879e-012`	`4.9977e-016`	`2.0346e-009`	`4.3288e-016`	`1.7554e-013`

From Table 4.1 we can conclude that our algorithm is convincingly better for th second problem. In other two cases is either slightly better or there is no significant difference between the methods. It is interesting to notice that `quadeig` algorithm has greater maximal backward error in every example.

Experiment 2. In this experiment we present the power of our deflation process. It is another example from NLEVP library, so called `mirror`, that originates from the calibration of cadioptric vision system. The order is $n = 9$.
Both A and E matrices are rank deficient, with the rank $r_E = r_A = 2$, which means that there are at least 7 zero and 7 infinite eigenvalues. They are deflated by the deflation process in `quadeig` algorithm. The QZ algorithm founds an additional zero eigenvalue, and two more infinite eigenvalues. `Polyeig` identified 2 zero eigenvalues, and 9 infinite eigenvalues. However, our deflation process found additional two zero and two infinite eigenvalues, making the total number of both zero and infinite eigenvalues equal to 9.
The smallest nonzero real eigenvalue computed by the `quadeig` is `-7.520795255755492e-014`. The seven smallest nonzero eigenvalues computed by the `polyeig` are

$$\begin{aligned} \lambda_1 &= \texttt{2.658653684986126e-028} & \lambda_5 &= \texttt{-8.144083812492196e-016} \\ \lambda_2 &= \texttt{-3.730521707731879e-024} & \lambda_6 &= \texttt{-1.057366058524636e-015} \\ \lambda_3 &= \texttt{4.343895348238823e-017} & \lambda_7 &= \texttt{-3.036244175050749e-014} \\ \lambda_4 &= \texttt{-4.135304334627443e-016} \end{aligned} \tag{4.55}$$

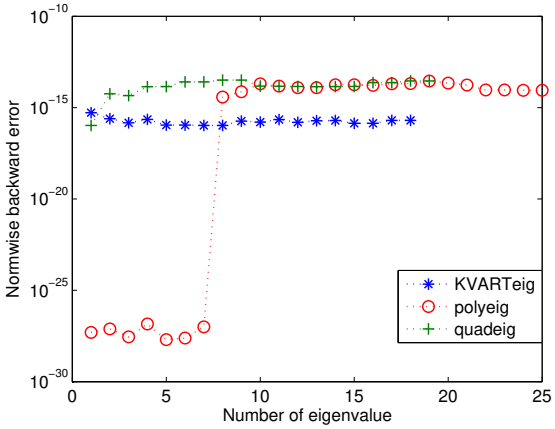


Figure 4.2: Norm-wise backward error for finite nonzero eigenvalues, `mirror`

Figure 4.2 presents (for all three algorithms) the backward errors for the finite nonzero eigenvalues, sorted by the magnitude.

The backward errors for the first seven finite eigenvalues for `polyeig` are of order 10^{-28} because they are small (4.55) and should be declared zero. It is clear form this figure that by just looking at the norm-wise backward error we cannot conclude that `polyeig` and `quadeig` did not find all zero eigenvalues because the backward errors are satisfying. Therefore this example shows the importance of checking whether there are more Jordan blocks for zero and infinite eigenvalue and then deflating them. If we look at the structure of matrices A and E for this particular problem, we see that their rank can be determined exactly because there are 7 zero columns in both matrices. On the other hand, the block matrices (4.54) which are used to determine the existence of more than one Jordan block for zero and infinite eigenvalues also have two zero columns each, and the rest 9×7 submatrices are well conditioned. Thus we can conclude that our algorithm determined the accurate number of zero and infinite eigenvalues.

Experiment 3: `orr_sommerfeld` of order 1000. Here, we specifically analyse the example `or_sommerfeld`, but now with much higher dimension, namely $n = 1000$. This means that corresponding quadratic problem has dimension 2000, and the corresponding generalized eigenvalue problem has dimension 4000. When using MATLAB function `polyeig`, all computed eigenvalues are of the form $\pm\mathrm{Inf}\pm\mathrm{Inf}\,i$. With our algorithm, the result depends on the rank determination of the matrix A, as described in Section 3.7. If we use the first criterion (F-norm), the rank is 988, meaning that 12 infinite eigenvalues are deflated. In the case of drop-off strategy, the matrix A is not rank deficient. The singular values of the matrix A are presented in Figure 4.3.

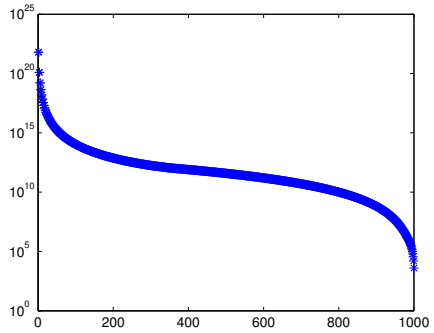


Figure 4.3: Singular values of leading matrix coefficient A, `orr_sommerfeld`

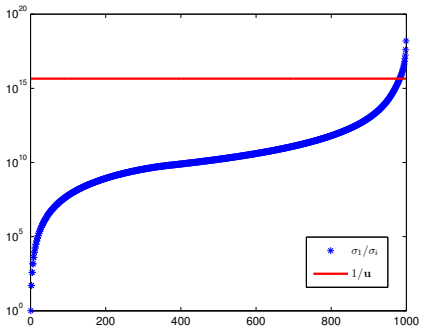
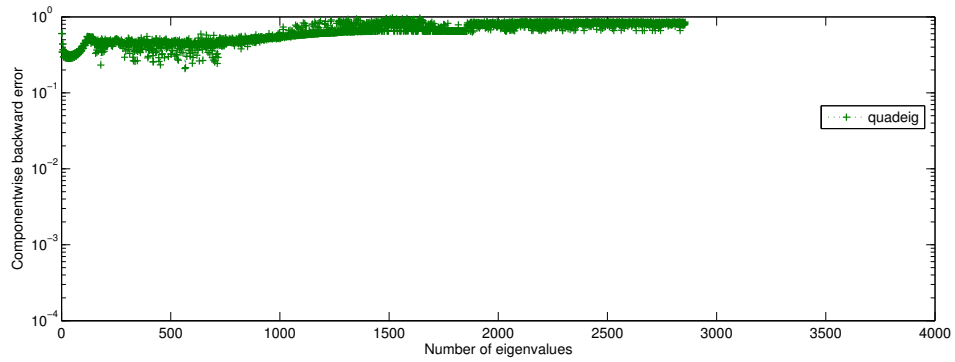
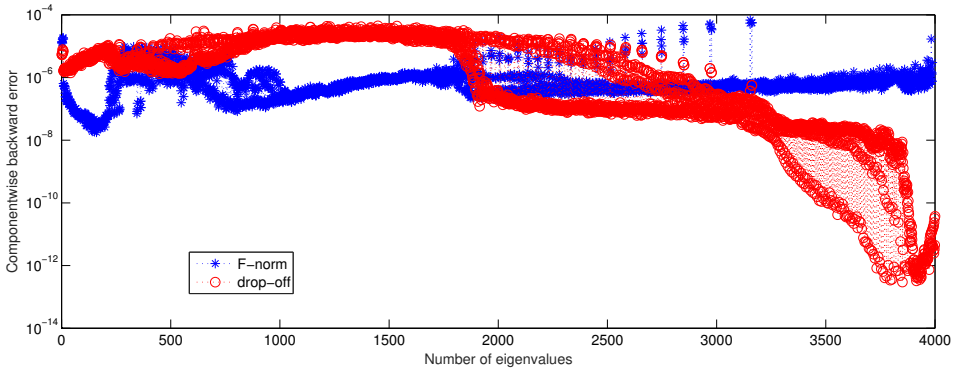


Figure 4.4: The ratio $\sigma_1(A)/\sigma_i(A)$ u–machine precision

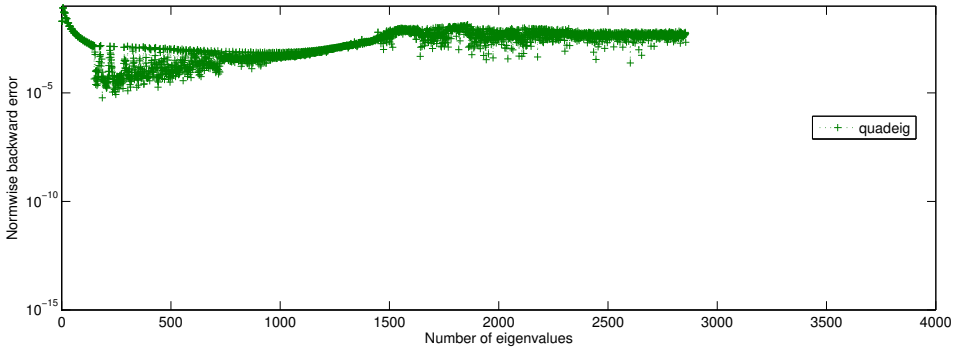
We also use `quadeig` algorithm to compute the eigenvalues of the corresponding quadratification by the second companion form. 1144 infinite eigenvalues are computed. We present the computed finite eigenvalues, and the corresponding norm-wise and component-wise backward errors in Figures 4.6b, 4.5a, 4.5c.



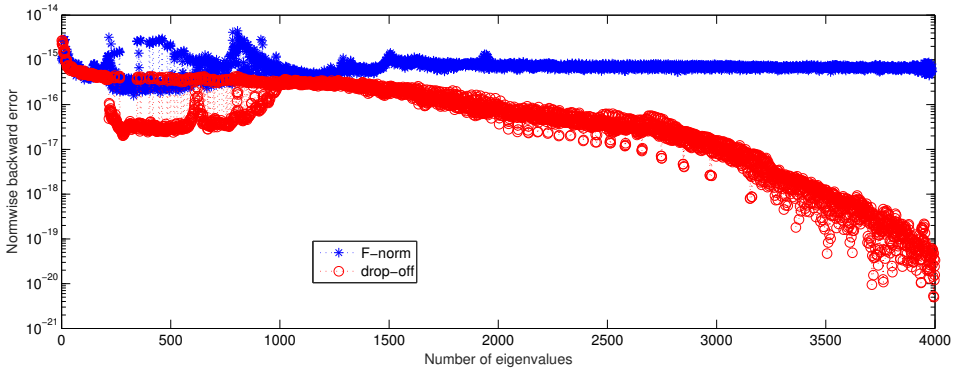
(a) Componentwise backward error in `quadeig`



(b) Componentwise backward error in `KVARTeig`



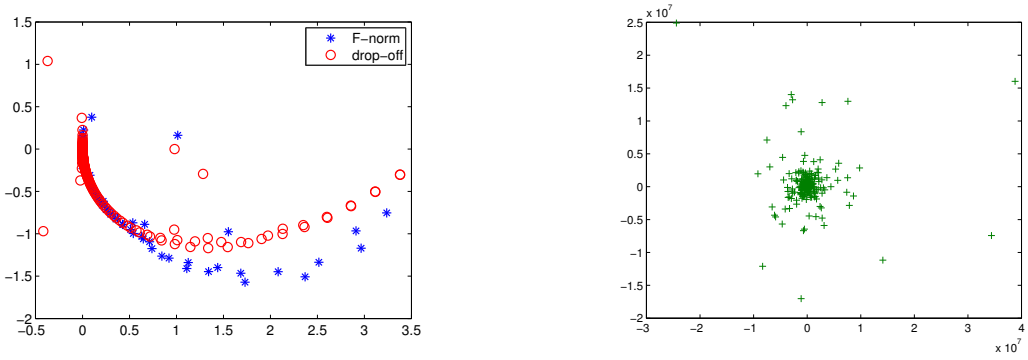
(c) Normwise backward error in `quadeig`



(d) Normwise backward error in `KVADeig`

Figure 4.5: Comparison of the normwise and componentwise backward errors for the finite right eigenpairs for `orr_sommeferld` example of order $n = 1000$

Once again we see the importance of careful rank determination. The component-wise backward error from Figure 4.5 shows that the second rank determination criterion gives better results.



(a) `KVARTeig` **(b)** `quadeig`

Figure 4.6: Computed finite eigenvalues for `orr_sommerfeld` example of order $n = 1000$

Chapter 5

Iterative methods

The objective of this section is the development of Arnoldi like methods for computation of the part of a spectrum for quadratic eigenvalue problem. In particular, we are only interested to find the prescribed number k of eigenvalues and corresponding right eigenvectors with a given property (for example, those of the largest magnitude, largest real part, closest to the real axis, etc.). Usually, the number of the wanted eigenvalues k is much smaller than the dimension of the problem n.

As we saw in the previous Chapters 3 and 4, the first step in solving the polynomial eigenvalue problems is the linearization. After that, we use well developed methods for the linear problem. However, a naive straightforward usage of these methods, without keeping in mind that the original problem is nonlinear, can produce poor results.

In this chapter, we will show, with the examples, the problem that occurs when using the Arnoldi algorithm for the quadratic eigenvalue problem. We will propose several improvements of the two level orthogonal Arnoldi algorithm. The main difference will be that the approximation for the wanted eigenpairs is obtained from the projected quadratic problem, and not projected linear problem. In addition, we will propose new shifts for restart for overdamped quadratic eigenvalue problems, and demonstrate its benefits through the numerical examples. We will propose a new selection method for starting vectors by approximating the original problem with a proportionally damped problem.

In the second part of the chapter the Krylov–Schur algorithm for the linear eigenvalue problem is introduced. We discuss the Krylov–Schur algorithm for the quadratic eigenvalue problem, and generalize the $4R$ procedure proposed in [11] when the TOAR algorithm is used to build the starting factorization. The importance of the usage of arbitrary shifts is presented with the numerical example.

5.1 Arnoldi algorithm

Consider the linear eigenvalue problem

$$Hy = \lambda y, \tag{5.1}$$

where $H \in \mathbb{C}^{N\times N}$. Instead of computing all N eigenpairs (λ, y), we want to find only $k \ll N$ eigenvalues with certain property, e.g. smallest absolute value, closest to the imaginary axis, belonging to a given $\Omega \subset \mathbf{C}$.
The idea is to find a *good* k–dimensional subspace spanned by an orthonormal $V \in \mathbb{C}^{N\times k}$, in the sense that it has a good information about the wanted part of the spectrum, and that is nearly H–invariant. Then, we compute the eigenpairs for the smaller projected problem of order k

$$\underbrace{(V^*HV)}_{\in\mathbb{C}^{k\times k}} z = \lambda z, \tag{5.2}$$

and if (λ, z) is an eigenpair for the projected problem (5.2), then (λ, Vz) is an approximate eigenpair for the original problem (5.1).
The goal of this Section is to explain, in more details, the Arnoldi type methods for finding V that nearly spans the subspace corresponding to the wanted eigenvalues. Here, V is chosen as the orthogonal basis $V_k = \begin{pmatrix} v_1 & \dots & v_k \end{pmatrix}$ of the Krylov subspace

$$\mathscr{K}_k(H, v_1) = \operatorname{span}\{v_1, Hv_1, \dots, H^{k-1}v_1\} \tag{5.3}$$

of order k. The basis is computed using the Gramm-Schmidt orthogonalization process. The algorithm is called Arnoldi algorithm and it is given below:

Algorithm 5.1.1 Arnoldi algorithm

1: $v_1 = v_1/\|v_1\|_2$
2: **for** $j = 1 : k$ **do**
3: $\quad r_j = Hv_j$
4: $\quad$ **for** $i = 1 : j$ **do**
5: $\quad\quad t_{ij} = v_i^* r_j$;
6: $\quad\quad r_j = r_j - v_i t_{ij}$
7: $\quad$ **end for**
8: $\quad t_{j+1,j} = \|r_j\|$
9: $\quad$ **if** $t_{j+1,j} = 0$ **then**
10: $\quad\quad \ell = j$; $V = [v_1, \dots, v_\ell]$; $T = (t_{ij})_{(\ell+1)\times\ell}$;
11: $\quad\quad$ STOP
12: $\quad$ **end if**
13: $\quad v_{i+1} = \frac{r_j}{t_{j+1,j}}$
14: **end for**
15: $\ell = k$; $V = [v_1, \dots, v_k]$; $T = (t_{ij})_{(k+1)\times k}$;

In a kth step of Algorithm 5.1.1, we get the so called Arnoldi factorization

$$HV_k = V_k T_k + r_k e_k^*, \tag{5.4}$$

where $T_k \in \mathbb{C}^{k\times k}$ is upper Hessenberg, and the columns of the orthonormal matrix V_k represent

an orthonormal basis for the Krylov subspace $\mathscr{K}_k(H, v_1)$.

Notice that, if the norm in the line 8 of the Algorithm 5.1.1 is zero, it means that $HV_\ell = V_\ell T_\ell$, or more precisely that V_ℓ spans an invariant subspace for H, and the eigenvalues of T_ℓ are the eigenvalues of H. Geometrically, it means that r_j is in the span of the previously computed orthogonal vectors $v_1, \ldots, v_{j-1}$. This is called the *breakdown*, and it is desirable for it to happen, because then we know that we have found an invariant subspace of H and the extracted spectral data is error-free.
However, if the breakdown did not occur, then we can only use approximate eigenpairs for H of the form $(\lambda, V_k z) =: (\lambda, y)$ where (λ, z) is a computed eigenpair for the projected matrix T_k of order k. The corresponding residual is $r = Hy - \lambda y$, and its norm is

$$\|r\|_2 = \|Hy - \lambda y\|_2 = \|(HV_k - V_k T_k)z\|_2 = \|r_k\|_2 |e_k^* z|. \tag{5.5}$$

If we define $\delta H = \frac{-ry^*}{y^* y}$ we have that (λ, y) is an exact eigenpair of the matrix $H + \delta H$. Hence, with sufficiently small residual, we can consider the computation of (λ, y) as backward stable. Moreover, this norm depends on the choice of the first vector v_1 and the following theorem says when can we expect for $\|r_k\|$ to be equal to zero.

Theorem 5.1 ([63]). *Let $HV_k - V_K T_k = r_k e_k^*$ be a k-step Arnoldi factorization of H, with T_k unreduced, i.e. $(T_k)_{i,i-1} \neq 0$, $i = 2, \ldots, k$. Then $r_k = \mathbf{0}$ if and only if $v_1 = Qy$ where $HQ = QR$ with $Q^* Q = \mathbb{I}_k$ and R upper triangular of order k.*

In essence, Theorem 5.1 states that, if the starting vector v_1 is a linear combination of k eigenvectors of H, the breakdown will occur in kth step of the Arnoldi algorithm, i.e. an invariant subspace of dimension k will be found.
Since we are interested in the specific eigenvalues, we would like the starting vector to be a linear combination of the corresponding (wanted) eigenvectors. Then, the eigenvalues of the Hessenberg matrix T_k would be exactly those that we are looking for. So the main question is, how to define a good starting vector when we do not know anything about the wanted part of the spectrum. The original idea is to use the *polynomial filters* and it was proposed by Saad in [59]. Suppose that the matrix H is diagonalizable, and let x_i, $i = 1, \ldots, N$ be the eigenbasis. Then, the starting vector v_1 is represented in this basis as

$$v_1 = \sum_{i=1}^{N} \xi_i x_i. \tag{5.6}$$

Let the eigenvalues be enumerated so that the first k represent the wanted ones. Split the sum in (5.6) in two parts

$$v_1 = \sum_{i=1}^{k} \xi_i x_i + \sum_{i=k+1}^{N} \xi_i x_i,$$

so that the first sum belongs to the wanted eigenvectors. In order to obtain the wanted invariant

subspace by using the Arnoldi algorithm, the first sum should prevail the second one (in vector norm). Saad's idea is to define a matrix function f which is large on the wanted part of the spectrum and small on the unwanted part. This matrix function is called polynomial filter. Then, if we apply it to our starting vector, we get

$$f(H)v_1 = \sum_{i=1}^{N} f(\lambda_i)\xi_i x_i = \sum_{i=1}^{k} f(\lambda_i)\xi_i x_i + \sum_{i=k+1}^{N} f(\lambda_i)\xi_i x_i. \tag{5.7}$$

Thus, if we define a new starting vector as $f(H)v_1$, where f is a polynomial filter, our starting vector will be better than the previous one.
In [59], Saad proposed to define f as polynomial p_s for which the minimum

$$\min_{p \in P_s} \max_{\lambda \in E} |p(\lambda)| \tag{5.8}$$

is achieved. This is difficult to solve for the arbitrary domain E. However, if $E = E(d,c,a)$ is an ellipse with real center d, foci $d+c$, $d-c$ and major semiaxis a, which contains the unwanted eigenvalues then the best minimax polynomial is

$$p_s(\lambda) = \frac{T_s((\lambda - d)/c)}{T_s((\lambda_1 - d)/c)}, \tag{5.9}$$

where T_s is the Chebyshev polynomial of degree s of the first kind which can be computed using the three–term recurrence

$$\begin{aligned} T_1(\lambda) = \lambda, \;\; T_0(\lambda) = 1, \\ T_{n+1}(\lambda) = 2\lambda T_n(\lambda) - T_{n-1}(\lambda), \; n \geq 1. \end{aligned}$$

The following algorithm computes $z_i = p_i(H)v_0$ which can be used to define a new starting vector in Arnoldi procedure

Algorithm 5.1.2 Chebyshev iteration

1: For given z_0, λ_1 and $E(d,c,a)$, compute $\sigma_1 = \frac{c}{\lambda_1 - d}$, $v_1 = \frac{\sigma_1}{c}(H - d\mathbb{I})z_0$
2: **for** $j = 1 : s$ **do**
3: $\quad \sigma_{j+1} = \frac{1}{2/\sigma_1 - \sigma_j}$
4: $\quad z_j = 2\frac{\sigma_{j+1}}{c}(H - d\mathbb{I})v_j - \sigma_j \sigma_{j+1} z_{j-1}$
5: **end for**

The full process is as follows:

- build the Arnoldi factorization of order $m > k$ with the starting vector v_1.
- Compute the eigenvalues of he Hessenberg matrix T_m. These are the approximations for the eigenvalues of H. Select the k wanted eigenvalues $\lambda_1, \ldots, \lambda_k$, with the corresponding eigenvectors $x_1, \ldots, x_k$.

- Find an ellipse $E(d,c,a)$ that contains the unwanted eigenvalues (of T_k). Define $z_0 = \sum_{i=1}^{k} x_i$ as a linear combination of the eigenvectors corresponding to the approximation of the wanted eigenvalues. Use Algorithm 5.1.2 to obtain z_s.
- Define new starting vector $v_1 = z_s/\|z_s\|_2$.
- Repeat these steps until convergence.

The implicit realization of this process is proposed by Sorensen in [63] and it is described in the next section.

5.1.1 Implicitly restarted Arnoldi (IRA)

Consider the linear filter

$$f(H) = H - \mu\mathbb{I}. \tag{5.10}$$

If μ is an eigenvalue of H and x_μ is the corresponding eigenvector, we have that $f(H)x_\mu = \mathbf{0}$. Moreover, if x is an arbitrary vector, $f(H)$ applied on x will remove the direction of the eigenvector x_μ from x. Recall the idea of the polynomial filter in (5.7). If, in addition, we define

$$f(H) = \prod_{i=k+1}^{N} (H - \lambda_i\mathbb{I}), \tag{5.11}$$

we get

$$f(H)v_1 = \sum_{i=1}^{k} f(H)\xi_i x_i + \mathbf{0},$$

that is, the directions of the unwanted eigenvectors in the representation of v_1 will be removed. However, we do not have any information about the spectrum of H, and thus we must use the approximations of λ_i to define the filter (5.10). The following figure illustrates one example of the filter (5.10). The goal was to determine 4 eigenvalues with the largest magnitude of the matrix of order $n = 500$ produced by MATLABs function `rand`.

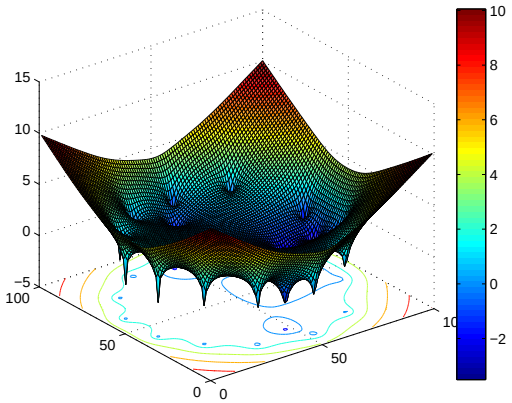


Figure 5.1: Polynomial filter in the first restart of IRA iterations

Sorensen [63] developed an implicit algorithm for applying this filter to the starting vector in the Arnoldi algorithm. The process goes as follows:

Building the starting factorization. As the first step, we build an mth order Arnoldi factorization, where m is larger than the number of the wanted eigenvalues k:

$$HV_m = V_m T_m + r_m e_m^*. \tag{5.12}$$

For example, in MATLAB's implementation of Implicitly Restarted Arnoldi (IRA) algorithm, `eigs`, the default value for m is $3k$.

Iterative part. Now, until convergence, repeat the following steps:

1. **Compute the eigenvalues of T_m:** The eigenvalues of the Hessenberg matrix T_m represent approximations for the eigenvalues of the original problem. However, among m of them we must choose those k which best correspond to the wanted ones. The remaining $m-k$ eigenvalues are then used to define the filter of the form (5.11). These are referred to as the unwanted eigenvalues. The partition in wanted and unwanted sets is done by sorting the computed eigenvalues by the prescribed criteria. For example, if we want to find the eigenvalues with the largest magnitude, we will just sort the approximations by the magnitude and choose k largest as the wanted ones, and the rest as the unwanted.
   Let $\lambda_1, \ldots, \lambda_m$ denote the eigenvalues of T_m, and assume that they are already enumerated so that $\lambda_1, \ldots, \lambda_k$ are the wanted ones.

2. **Implicit QR iterations.** The next step is an application of $p(= m-k)$ implicitly shifted QR iterations on T_m:
   $$T_m - \mu_i \mathbb{I} = Q_i R_i, \;\; i = 1, \ldots, p, \tag{5.13}$$
   resulting in $Q_m^* T_m Q_m = T_m^+$, where $Q_m = Q_p \ldots Q_1$. Since T_m is upper Hessenberg, the matrices Q_i, $i = 1, \ldots, p$ are upper Hessenberg as well, and the matrix Q_m is such that $Q_m(i, j) = 0$ for $i > j + p$, as a product of p Hessenberg matrices.
   If the matrix H is real, we want to keep the Arnoldi factorization real as well. In that case, a complex shift appear as conjugate pair $\mu_j = \alpha_j + i\beta_j$, and one uses the double shift in (5.13)
   $$(T_m - \alpha_j \mathbb{I})^2 + \beta_j^2 \mathbb{I} = Q_j R_j. \tag{5.14}$$
   If we use the unwanted Ritz values (eigenvalues of T_m) as shifts μ_i, they are called the *exact shifts*.

3. **Truncation to the factorization of order k:** Multiply the factorization (5.12) by Q_m from the right to get
   $$HV_m Q_m = V_m Q_m (Q_m^* T_m Q_m) + r_m e_m^* Q_m. \tag{5.15}$$

Now, since Q_m is product of p orthogonal Hessenberg matrices the row vector $e_m^* Q_m = \begin{pmatrix} 0 & \dots & 0 & \beta & b^* \end{pmatrix}$ has its first nonzero element on the kth position. Thus, if we equate the first k columns of the (5.15) we get the k order Arnoldi factorization

$$HV_k^+ = V_k^+ T_k^+ + r_k^+ e_k^*, \tag{5.16}$$

where $V_k^+ = V_m Q_m(:, 1:k)$, $T_k^+ = T_m^+(1:k, 1:k)$ and $r_k^+ = V_m Q_m(:, k+1) T(k+1, k) + r_m Q_m(m, k)$.
This is equivalent to Arnoldi decomposition obtained using the starting vector v_1^+

$$v_1^+ = \prod_{j=k+1}^{m} (H - \lambda_j \mathbb{I}) v_1. \tag{5.17}$$

4. **Expand to factorization of order m:** Using the Arnoldi process, without having to compute first k steps, we obtain the Arnoldi factorization of order m from (5.16).

Implicitly restarted Arnoldi algorithm is implemented ARPACK [48] which is used by the MATLABs function `eigs`.

5.2 Second Order Arnoldi (SOAR)

Suppose that we want to use the Implicitly Restarted Arnoldi (IRA) algorithm for computing a part of the spectrum for the quadratic eigenvalue problem

$$Q(\lambda)x = (\lambda^2 M + \lambda C + K)x = \mathbf{0},$$

by applying it to the first companion form linearized problem

$$Hy = \begin{pmatrix} -M^{-1}C & -M^{-1}K \\ \mathbb{I} & \mathbf{0} \end{pmatrix} y = \lambda y, \quad y = \begin{pmatrix} \lambda x \\ x \end{pmatrix}. \tag{5.18}$$

Already with the linearization the structure of the problem is lost, and, in addition, we use a small linear problem for the approximation of the large nonlinear eigenvalue problem.

Example 5.1. Consider the quadratic eigenvalue problem $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$ with the following coefficient matrices

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; C = \begin{pmatrix} 7 & -5 & 0 \\ 10 & -8 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \; K = \begin{pmatrix} 0 & 1 & 0 \\ -2 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

and the corresponding first companion form linearization

$$H = \begin{pmatrix} -7 & 5 & 0 & 0 & -1 & 0 \\ -10 & 8 & 0 & 2 & -3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{5.19}$$

Suppose we want to compute the eigenvalue with the largest magnitude. We use MATLAB's `eigs`(H,k), where $k = 1$. If we define that the maximal dimension m of the Arnoldi factorization is 3 (meaning that $p = 2$ shifts are used at restart), the algorithm fails to find the wanted eigenvalue in the first 300 restarts, producing the following error

```
??? Error using ==> eigs>processEUPDinfo at 1453 Error with ARPACK
routine dneupd: dnaupd did not find any eigenvalues to sufficient
accuracy.
```

We see that, even for the small problems the state of the art algorithm can fail.

One of the drawbacks of the direct application of the Arnoldi algorithm to the linearization is that the computed Rayleigh quotient destroys the structure of the original quadratic problem. The idea of Bai and Su in [3] is to find a good subspace, rich with the information of the wanted part of the spectrum, and then use the smaller projected quadratic problem to approximate the eigenvalues. Furthermore, if the projection is orthogonal, the structure, and therefore the specific properties of the original problem, are preserved. For example, if Q is an orthonormal basis for such a subspace, then the projected pencil is $\lambda^2(Q^*MQ) + \lambda(Q^*CQ) + (Q^*KQ)$ and if, e.g., M is Hermitian, then so is Q^*MQ as well.

The proposed wanted subspace would be the basis of the generalized Krylov subspace, which was introduced in [3]. In contrast to standard Krylov subspace, their definition depends on two matrices of the same order n and one vector.

Definition 5.1. *Let $A, B \in \mathbb{C}^n$ and $u \in \mathbb{C}^n \setminus \{0\}$. The sequence $r_0, r_1, \ldots, r_{k-1}$, where*

$$\begin{aligned} r_0 &= u, \\ r_1 &= Ar_0, \\ r_j &= Ar_{j-1} + Br_{j-2},\ j \geq 2 \end{aligned} \tag{5.20}$$

is called a second order Krylov sequence based on A, B and u. The space

$$\mathscr{G}_k(A, B; u) = \operatorname{span}\{r_0, r_1, \ldots, r_{k-1}\} \tag{5.21}$$

is called a second order Krylov subspace of order k.

Definition (5.1) is a generalization of the standard Krylov subspace definition, in the sense that $\mathscr{G}_k(A,\mathbf{0};u) = \mathscr{K}_k(A,u)$.
The algorithm for computing an orthogonal basis of (5.21) is given below.

Algorithm 5.2.1 $[P,Q,H,\ell]$ =SOAR(A,B,u,k)

1: $q_1 = u/\|u\|_2$
2: $p_1 = 0$
3: **for** $j = 1,2,\ldots,k$ **do**
4: $\quad r = Aq_j + Bp_j$
5: $\quad s = q_j$
6: $\quad$ **for** $i = 1,2,\ldots,j$ **do**
7: $\quad\quad t_{ij} = q_i^T r$
8: $\quad\quad r = r - q_i t_{ij}$
9: $\quad\quad s = s - p_i t_{ij}$
10: $\quad$ **end for**
11: $\quad t_{j+1j} = \|r\|_2$
12: $\quad$ **if** $t_{j+1j} = 0$ **then**
13: $\quad\quad \ell = j$
14: $\quad\quad P = [p_1,\ldots,p_\ell]$, $Q = [q_1,\ldots,q_\ell]$, $T = (t_{ij})_{(\ell+1)\times\ell}$
15: $\quad\quad$ STOP
16: $\quad$ **end if**
17: $\quad q_{j+1} = r/t_{j+1j}$
18: $\quad p_{j+1} = s/t_{j+1j}$
19: **end for**
20: $\ell = k$
21: $P = [p_1,\ldots,p_k]$, $Q = [q_1,\ldots,q_k]$, $T = (t_{ij})_{(k+1)\times k}$

After k steps of Algorithm 5.2.1, we get the second order Arnoldi factorization

$$AQ_k + BP_k = Q_k T_k + q_{k+1} e_k^T t_{k+1,k}, \tag{5.22}$$

$$Q_k = P_k T_k + p_{k+1} e_k^T t_{k+1,k}, \tag{5.23}$$

where Q_k has orthogonal columns and it represents the basis for the second order Krylov subspace of order k; T_k is upper Hessenberg, and P_k contains auxiliary vectors. The factorization (5.22)-(5.23) can be also written in compact form

$$H\begin{pmatrix} Q_k \\ P_k \end{pmatrix} = \begin{pmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{pmatrix}\begin{pmatrix} Q_k \\ P_k \end{pmatrix} = \begin{pmatrix} Q_k \\ P_k \end{pmatrix} T_k + \begin{pmatrix} q_{k+1} \\ p_{k+1} \end{pmatrix} e_k^T t_{k+1,k}. \tag{5.24}$$

This is similar to the Arnoldi factorization (5.4), except that the block matrix $\left(\begin{smallmatrix} Q_k \\ P_k \end{smallmatrix}\right)$ is not orthogonal.

To further explore the connection between the Arnoldi and the second order Arnoldi factorization, we make a distinction between the two key events: deflation and breakdown, which are associated with the norm in line 11 in Algorithm 5.2.1 being zero.

In the Arnoldi Algorithm 5.1.1, we concluded that breakdown means that the current vector r_j is in the span of the previously computed vectors, implying that we have found an invariant subspace. This is regarded as a good thing. However, in the SOAR procedure, the vectors r_j in (5.20), which are being orthogonalized, depend on two previous vectors. Thus, when in the jth step we get that the norm in line 11 is equal to zero, we can conclude that r_j is in the span of the previously computed vectors, i.e. $\mathscr{G}_{j-1}(A,B,u) = \mathscr{G}_j(A,B,u)$. However, this does not have

to be true for all subsequent vectors r_k, $k > j$. This means that, when this happens, we cannot say that we have found an orthogonal basis for the second order Krylov subspace. To be able to claim this, we have to check whether the block vector $\binom{r_j}{r_{j-1}}$ is in the span of the previously computed $\binom{r_i}{r_{i-1}}$, $i = 1, \ldots, j-1$. If that is the case, then we call it a breakdown and we know that we found the basis. If not, we call it deflation and the process continues. The full algorithm, which deals with the deflation phenomena is given below:

Algorithm 5.2.2 $[P, Q, H, \ell] = \text{SOAR}(A, B, u, k)$

1: $q_1 = u/\|u\|_2$
2: $p_1 = 0$
3: **for** $j = 1, 2, \ldots, k$ **do**
4: $\quad r = Aq_j + Bp_j$
5: $\quad s = q_j$
6: $\quad$ **for** $i = 1, 2, \ldots, j$ **do**
7: $\quad\quad t_{ij} = q_i^T r$
8: $\quad\quad r = r - q_i t_{ij}$
9: $\quad\quad s = s - p_i t_{ij}$
10: $\quad$ **end for**
11: $\quad t_{j+1\,j} = \|r\|_2$
12: $\quad$ **if** $t_{j+1\,j} = 0$ **then**
13: $\quad\quad$ **if** $s \in \text{span}\{p_i | i : q_i = 0, 1 \le i \le j\}$ **then**
14: $\quad\quad\quad$ breakdown
15: $\quad\quad\quad \ell = j$
16: $\quad\quad\quad P = [p_1, \ldots, p_\ell],\ Q = [q_1, \ldots, q_\ell],$ $T = (t_{ij})_{(\ell+1)\times\ell}$
17: $\quad\quad$ **else**
18: $\quad\quad\quad$ deflation
19: $\quad\quad\quad t_{j+1\,j} = 1,\ q_{j+1} = 0,\ p_{j+1} = s$
20: $\quad\quad$ **end if**
21: $\quad$ **else**
22: $\quad\quad q_{j+1} = r/t_{j+1\,j}$
23: $\quad\quad p_{j+1} = s/t_{j+1\,j}$
24: $\quad$ **end if**
25: **end for**
26: $\ell = k$
27: $P = [p_1, \ldots, p_k]$, $Q = [q_1, \ldots, q_k]$
28: $T = (t_{ij})_{(k+1)\times k}$

Now, in [3], Bai and Su proved the following theorem, which gives the connection between the SOAR and the Arnoldi algorithm.

Theorem 5.2 ([3])**.** *The SOAR procedure with the matrices A and B, and the starting vector u breaks down at a certain step j if and only if the Arnoldi procedure with the matrix* $\left(\begin{smallmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{smallmatrix}\right)$ *and the starting vector* $\binom{u}{\mathbf{0}_n}$ *breaks down at the same step j.*

It is instructive to note here that, when the breakdown occurs in the SOAR algorithm, the matrix $\binom{Q_j}{P_j}$ spans an invariant subspace for the matrix $\left(\begin{smallmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{smallmatrix}\right)$, but it is not an orthonormal basis, and we know that the computation of a nonorthonormal basis may not be a numerically stable process.

To see an application for solving the partial quadratic eigenvalue problem, we define $A = -M^{-1}C$ and $B = -M^{-1}K$. Now, the matrix $\left(\begin{smallmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{smallmatrix}\right)$ represents the first companion form linearization for the quadratic eigenvalue problem. Compute the orthogonal basis Q_k for the second order Krylov subspace $\mathscr{G}_k(A, B; u)$. To find an approximation for the eigenpairs we now compute the eigenvalues of the smaller projected problem of order k:

$$(\lambda^2 \underbrace{(Q_k^* M Q_k)}_{=:M_k} + \lambda \underbrace{(Q_k^* C Q_k)}_{=:C_k} + \underbrace{(Q_k^* K Q_k)}_{=:K_k})z = \mathbf{0}. \tag{5.25}$$

Notice that the structure of the original quadratic eigenvalue problem is preserved, for example, if M, C and K are Hermitian so are M_k, C_k and K_k as well. We will see later that this will be an important property for defining a new way of choosing the shifts for the quadratic eigenvalue problem with certain property.

5.3 Two level orthogonal Arnoldi factorization

As we mentioned before, the SOAR algorithm can be interpreted as an algorithm for finding a nonorthogonal basis for the Krylov subspace $\mathscr{K}_k(H,v)$, where H is of the form (5.24), and $u^T = \begin{pmatrix} v^T & \mathbf{0}_n \end{pmatrix}$. Therefore, the SOAR algorithm has tendency to be numerically unstable ([49]). This is why Lu, Su and Bai [49] developed the Two level Orthogonal Arnoldi (TOAR) procedure, which preserves the orthogonality of the block matrix $\begin{pmatrix} Q_k \\ P_K \end{pmatrix}$ as well.

In order to develop the TOAR procedure, a slightly modified definition of the second order Krylov subspace is introduced. Here, the second order Krylov subspace $\mathscr{G}_k(A,B;r_{-1},r_0)$ depends on two starting vectors of order n, and it is a generalization of (5.21) in the sense that in the case of Definition 5.1, r_{-1} is always a null vector.

Definition 5.2. *Let $A,B \in \mathbb{C}^{n\times n}$ and $r_{-1}, r_0 \in \mathbb{C}^n$ such that $\begin{pmatrix} r_{-1}^T & r_0^T \end{pmatrix}^T \neq 0$. Then the sequence $r_{-1}, r_0, r_1, \ldots, r_k$ with*

$$r_j = Ar_{j-1} + Br_{j-2}, \quad j \geq 1 \tag{5.26}$$

is called a second order Krylov sequence based on A, B, r_{-1} and r_0. The subspace

$$\mathscr{G}_k(A,B;r_{-1},r_0) = \operatorname{span}\{r_{-1}, r_0, \ldots, r_{k-1}\} \tag{5.27}$$

is called a second order Krylov subspace of order k.

Consider the second order Krylov subspace $\mathscr{G}_k(A,B;r_{-1},r_0)$, and let Q_k be its orthogonal basis. Furthermore, let $\mathscr{K}_k(H,v)$ be the standard Krylov subspace, with $H \in \mathbb{C}^{2n\times 2n}$ as in (5.24) and $v = \begin{pmatrix} r_0 \\ r_{-1} \end{pmatrix}$, and V_k its orthogonal basis. From the definition of the sequence (5.26) it holds

$$\mathscr{K}_k(H,v) = \operatorname{span}\{v, Hv, \ldots, H^{k-1}v\} = \operatorname{span}\left\{ \begin{pmatrix} r_0 \\ r_1 \end{pmatrix}, \begin{pmatrix} r_1 \\ r_0 \end{pmatrix}, \ldots, \begin{pmatrix} r_{k-1} \\ r_{k-2} \end{pmatrix} \right\}. \tag{5.28}$$

Now,

$$\operatorname{span}\{V_k(1:n,:)\} = \operatorname{span}\{r_0, r_1, \ldots, r_{k-1}\}, \tag{5.29}$$

$$\operatorname{span}\{V_k(n+1:2n,:)\} = \operatorname{span}\{r_{-1}, r_0, \ldots, r_{k-2}\}, \tag{5.30}$$

that is

$$\operatorname{span}\{Q_k\} = \operatorname{span}\{V_k(1:n,:), V_k(n+1:2n,:)\}. \tag{5.31}$$

This connection can be written as

$$V_k = \begin{pmatrix} V_k(1:n,:) \\ V_k(n+1:2n,:) \end{pmatrix} = \begin{pmatrix} Q_k U_{k,1} \\ Q_k U_{k,2} \end{pmatrix}. \tag{5.32}$$

Therefore, the basis Q_k can be computed from V_k using the rank revealing QR factorization of either $V_k(1:n,:)$ or $V_k(n+1:2n,:)$. Without loss of generality we can assume that $V_k \in \mathbb{C}^{2n\times k}$. However, the number of columns of $Q_k \in \mathbb{C}^{n\times \eta_k}$ can be smaller than $\eta_k < k$, which would correspond to the deflation.

Instead of building the Arnoldi factorization for the first companion form linearization and then computing the QR factorization to obtain the basis for the second order Krylov subspace, TOAR computes Q_k by maintaining the orthogonality of the basis for the standard Krylov subspace. This is why it is called two level orthogonal. One Gram-Schmidt process is used to compute orthogonal basis Q_k and another for V_k. The full algorithm is presented below.

Algorithm 5.3.1 $[Q_k, U_{k,1}, U_{k,2}, H_k]$ =TOAR(A, B, r_{-1}, r_0, k)

$\begin{pmatrix} r_{-1} & r_0 \end{pmatrix} = QX$ (Rank revealing QR factorization, η_1 is the rank)
$\gamma = \left\| \begin{pmatrix} r_0 \\ r_{-1} \end{pmatrix} \right\|_2$
$Q_1 = Q, U_{1,1} = X(:,2)/\gamma, U_{1,2} = X(:,1)/\gamma.$
for $j = 1 : k-1$ **do**
  $r = A(Q_j U_{j,1}(:,j)) + B(Q_j U_{j,2}(:,j))$
  for $i = 1 : \eta_j$ **do**
    $s_i = q_i^T r$
    $r = r - s_i q_i$
  end for
  $\alpha = \|r\|_2$
  $s = [s_1, \ldots, s_{\eta_j}]^T$, $u = U_{j,1}(:,j)$
  for $i = 1 : j$ **do**
    $t_{ij} = U_{j,1}(:,i)^T s + U_{j,2}(:,i)^T u$
    $s = s - t_{ij} U_{j,1}(:,i)$, $u = u - t_{ij} U_{j,2}(:,i)$
  end for
  $t_{j+1,j} = (\alpha^2 + \|s\|_2^2 + \|u\|_2^2)^{1/2}$
  if $t_{j+1,j} = 0$ **then**
    stop (breakdown)
  end if
  if $\alpha = 0$ **then**
    $\eta_{j+1} = \eta_j$ (deflation)
    $Q_{j+1} = Q_j$, $U_{j+1,1} = \begin{pmatrix} U_{j,1} & s/t_{j+1,j} \end{pmatrix}$, $U_{j+1,2} = \begin{pmatrix} U_{j,2} & u/t_{j+1,j} \end{pmatrix}$
  else
    $\eta_{j+1} = \eta_j + 1$
    $Q_{j+1} = \begin{pmatrix} Q_j & r/\alpha \end{pmatrix}$
    $U_{j+1,1} = \begin{pmatrix} U_{j,1} & s/t_{j+1,j} \\ 0 & \alpha/t_{j+1,j} \end{pmatrix}$
    $U_{j+1,2} = \begin{pmatrix} U_{j,2} & u/t_{j+1,j} \\ 0 & 0 \end{pmatrix}$
  end if
end for

Remark 5.1. The Arnoldi algorithm, as well as SOAR, and TOAR use the Gramm–Schmidt orthogonalization process. However, in finite precision arithmetic this procedure does not have to produce numerically orthogonal vectors. To insure the numerical orthogonality, for example in Algorithm 5.3.1, after the $\alpha = \|r\|_2$ is computed, one should check if $\alpha \leq \tau \|A(Q_j U_{j,1}(:,j)) + B(Q_j U_{j,2}(:,j))\|_2$, for the threshold parameter $\tau \leq 1$. If the inequality holds, additional orthogonalization of r against Q_j is performed. This procedure is known as the twice–is–enough algorithm [57].

After k steps of Algorithm 5.3.1 we get the TOAR factorization

$$\begin{pmatrix} A & B \\ \mathbb{I}_n & \mathbf{0}_n \end{pmatrix} \begin{pmatrix} Q_k U_{k,1} \\ Q_k U_{k,2} \end{pmatrix} = \begin{pmatrix} Q_k U_{k,1} \\ Q_k U_{k,2} \end{pmatrix} T_k + r_k e_k^T t_{k+1,k}, \tag{5.33}$$

where

$$r_k = \begin{pmatrix} Q_k(s/t_{k+1,k}) + q_{k+1}(\alpha/t_{k+1,k}) \\ Q_k(u/t_{k+1,k}) \end{pmatrix},$$

if no deflation occurred in the last step, and

$$r_k = \begin{pmatrix} Q_k(s/t_{k+1,k}) \\ Q_k(u/t_{k+1,k}) \end{pmatrix},$$

otherwise. Numerical stability of the Algorithm 5.3.1 is proved in [49].

5.3.1 Implicitly restarting the TOAR procedure

In this subsection we give a review of the implicit restarting procedure for the TOAR algorithm analogous to the implicitly restarted Arnoldi. That is, we want to apply the polynomial filter of the form

$$f(H) = \prod_{i=1}^{p} (H - \mu_i \mathbb{I}), \tag{5.34}$$

to the starting block vector $\binom{r_0}{r_{-1}}$, where μ_i, $i = 1, \ldots, p$ are the shifts which are determined in some prescribed manner. Since the factorization (5.33) is also an Arnoldi factorization for the matrix H, we can modify the process described in Subsection 5.1.1. Suppose that we have a TOAR factorization of order $m > k$, and we want to truncate it to the order k. First, we compute p steps of the shifted QR factorization on T_m with the given shifts $\mu_1, \ldots, \mu_p$ to get

$$T_m = V T_m^+ V^*. \tag{5.35}$$

T_m^+ is again upper Hessenberg, and V is orthogonal with $V_{i,j} = 0$ for $i > j + p$. Multiply the decomposition (5.33) with V from the right to get

$$\begin{pmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} Q_m U_{m,1} V \\ Q_m U_{m,2} V \end{pmatrix} = \begin{pmatrix} Q_m U_{m,1} V \\ Q_m U_{m,2} V \end{pmatrix} V^* T_m V + r_m e_m^T V t_{m+1,m}. \tag{5.36}$$

Now, the truncated factorization is

$$\begin{pmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} Q_m U_{k,1}^+ \\ Q_m U_{k,2}^+ \end{pmatrix} = \begin{pmatrix} Q_m U_{k,1}^+ \\ Q_m U_{k,2}^+ \end{pmatrix} T_k^+ + r_k^+ e_k^T t_{k+1,k}^+, \tag{5.37}$$

where $U_{k,1}^+ = U_{m,1}V(:,1:k), U_{k,2}^+ = U_{m,2}V(:,1:k), T_k^+ = T_m^+(1:k,1:k)$, and

$$s^+ = U_{m,1}V(:,k+1)T_m^+ k+1,k+st_{m+1,m}V(m,k), \tag{5.38}$$

$$\alpha^+ = \alpha t_{m+1,m}V(m,k), \tag{5.39}$$

$$u^+ = U_{m,2}V(:,k+1)T_m^+(k+1,k)+ut_{m+1,m}V(m,k), \tag{5.40}$$

$$t_{k+1,k}^+ = \sqrt{(\|u^+\|^2+\|s^+\|^2+(\alpha^+)^2)}. \tag{5.41}$$

However, we are not done yet. Notice that (5.37) is not a TOAR factorization because Q_m still has $\eta_m \leq m+1$ columns and it does not represent the basis of the blocks $Q_m U_{k,1}^+$ and $Q_m U_{k,2}^+$. To make it a legitimate TOAR factorization we compute the compact SVD factorization, as proposed in [65]

$$\left(\begin{array}{cc|cc} U_{k,1}^+ & s^+/t_{k+1,k}^+ & U_{k,2}^+ & u^+/t_{k+1,k}^+ \\ \mathbf{0} & \alpha^+/t_{k+1,k}^+ & \mathbf{0} & 0 \end{array}\right) = P\Sigma G^*, \tag{5.42}$$

$P \in \mathbb{C}^{\eta_{m+1}\times\eta_{k+1}}$, $\Sigma \in \mathbb{C}^{\eta_{k+1}\times\eta_{k+1}}$ and $G = \begin{pmatrix} G_1 & G_2 \end{pmatrix} \in \mathbb{C}^{\eta_{k+1}\times((k+1)+(k+1))}$. The rank η_{k+1} is at least $k+2$. Now, the final factorization, written in compact form, of order k is

$$\begin{pmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} Q_{k+1}^+ U_{k+1,1}^+ \\ Q_{k+1}^+ U_{k+1,2}^+ \end{pmatrix} = \begin{pmatrix} Q_{k+1}^+ U_{k+1,1}^+ \\ Q_{k+1}^+ U_{k+1,2}^+ \end{pmatrix} \widehat{T}_k^+, \tag{5.43}$$

where $Q_{k+1}^+ = Q_{m+1}P$, $U_{k+1,1}^+ = \Sigma G_1$ and $U_{k+1,2} = \Sigma G_2$. The updating algorithm is presented in Algorithm 5.3.2.

Algorithm 5.3.2 $[Q_m, U_{m,1}, U_{m,2}, H_m]$ =TOAR_Update$(A, B, Q_{k+1}, U_{k+1,1}, U_{k+1,2}, T_k, m)$

1: **for** $j = k+1 : m$ **do**
2: $r = A(Q_j U_{j,1}(:,j)) + B(Q_j U_{j,2}(:,j))$
3: **for** $i = 1 : \eta_j$ **do**
4: $s_i = q_i^T r$
5: $r = r - s_i q_i$
6: **end for**
7: $\alpha = \|r\|_2$
8: $s = [s_1, \ldots, s_{\eta_j}]^T$, $u = U_{j,1}(:,j)$
9: **for** $i = 1 : j$ **do**
10: $t_{ij} = U_{j,1}(:,i)^T s + U_{j,2}(:,i)^T u$
11: $s = s - t_{ij}U_{j,1}(:,i)$, $u = u - t_{ij}U_{j,2}(:,i)$
12: **end for**
13: $t_{j+1,j} = (\alpha^2 + \|s\|_2^2 + \|u\|_2^2)^{1/2}$
14: **if** $t_{j+1,j} = 0$ **then**
15: stop (breakdown)
16: **end if**
17: **if** $\alpha = 0$ **then**
18: $\eta_{j+1} = \eta_j$ (deflation)
19: $Q_{j+1} = Q_j$, $U_{j+1,1} = \begin{pmatrix} U_{j,1} & s/t_{j+1,j} \end{pmatrix}$, $U_{j+1,2} = \begin{pmatrix} U_{j,2} & u/t_{j+1,j} \end{pmatrix}$
20: **else**
21: $\eta_{j+1} = \eta_j + 1$
22: $Q_{j+1} = \begin{pmatrix} Q_j & r/\alpha \end{pmatrix}$
23: $U_{j+1,1} = \begin{pmatrix} U_{j,1} & s/t_{j+1,j} \\ 0 & \alpha/t_{j+1,j} \end{pmatrix}$
24: $U_{j+1,2} = \begin{pmatrix} U_{j,2} & u/t_{j+1,j} \\ 0 & 0 \end{pmatrix}$
25: **end if**
26: **end for**

5.4 TOAR revisited

In this subsection we present a new interpretation of the deflation and breakdown phenomena in the TOAR algorithm, in the terms of the invariant pair for the quadratic eigenvalue problem. We present two interpretations of the TOAR algorithm, namely as a linear solver, and as a quadratic solver. In this regard, we propose several improvements of the restarting procedure presented in §5.3.1.

5.4.1 Deflation and breakdown

Recall the defining relation for an invariant pair $(X,S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ (see Section 1.1):

$$MXS^2 + CXS + KX = \mathbf{0}. \tag{5.44}$$

Now, assume that in the kth step of the Algorithm 5.3.1 we have $t_{k+1,k} = 0$. It means that

$$AQ_kU_{k,1} + BQ_kU_{k,2} = Q_kU_{k,1}T_k, \tag{5.45}$$

$$Q_kU_{k,1} = Q_kU_{k,2}T_k. \tag{5.46}$$

If we substitute (5.46) into (5.45), and use the fact that $A = -M^{-1}C$ and $B = -M^{-1}K$ we get

$$MQ_kU_{k,2}T_k^2 + CQ_kU_{k,2}T_k + KQ_kU_{k,2} = \mathbf{0}, \tag{5.47}$$

or,

$$MQ_kU_{k,1}T_k^2 + CQ_kU_{k,1}T_k + KQ_kU_{k,1} = \mathbf{0}, \tag{5.48}$$

from $U_{k,2} = U_{k,1}T_k^{-1}$. This means that $(Q_kU_{k,1}, T_k)$ and $(Q_kU_{k,2}, T_k)$ are invariant pairs for the quadratic problem. On the other hand, $t_{k+1,k} = 0$ implies

$$H\begin{pmatrix} Q_kU_{k,1} \\ Q_kU_{k,2} \end{pmatrix} = \begin{pmatrix} Q_kU_{k,1} \\ Q_kU_{k,2} \end{pmatrix} T_k, \tag{5.49}$$

meaning that we have found an invariant pair $\left(\begin{pmatrix} Q_kU_{k,1} \\ Q_kU_{k,2} \end{pmatrix}, T_k\right)$ for the linear problem. If deflation occurred, the matrices $Q_kU_{k,1}$ and $Q_kU_{k,2}$ are not of full rank, which means that there is linear dependence between eigenvectors for the eigenvalues of T_k. However, the block matrix $\begin{pmatrix} Q_kU_{k,1} \\ Q_kU_{k,2} \end{pmatrix}$ is always orthogonal of full rank.

Remark 5.2. From the reasoning above we can conclude that the TOAR algorithm can be interpreted as an algorithm for computing the minimal invariant pair for the quadratic pencil, i.e., if the breakdown occurred at the kth step of the algorithm, $(Q_kU_{k,1}, T_k)$ and $(Q_kU_{k,2}, T_k)$ satisfy (5.44), and $\begin{pmatrix} Q_kU_{k,1} \\ Q_kU_{k,2} \end{pmatrix} = \begin{pmatrix} Q_kU_{k,2}T_k \\ Q_kU_{k,2} \end{pmatrix}$ is of full rank.

We want to use TOAR algorithm to compute a part of the spectrum of the quadratic eigenvalue problem. However, instead of approximating the eigenvalues of quadratic problem with eigenvalues of smaller linear problem we would like to use the basis Q_k for the second order Krylov subspace for defining the smaller projected problem (5.25), and then solve the smaller quadratic problem with the same structure.

There are two ways to look at the TOAR algorithm. The first way is that the TOAR procedure (5.3.1) computes a basis for the standard Krylov subspace $\mathscr{K}_k(H,v)$ and by implicitly restarting it will be closer to an invariant subspace of the matrix H. On the other hand, we use it to compute an orthogonal basis for the second order Krylov subspace $\mathscr{G}_k(A,B,r_{-1},r_0)$, and, by implicit restart, we want it to find a better subspace that will be used to project our quadratic problem. More precisely, the implicitly restarted TOAR algorithm is both a linear solver, and a quadratic solver. To construct a robust algorithm, we must keep in mind the specifics of both of these problems, and adjust our algorithm to it, always keeping in mind that the main goal is to solve the quadratic eigenvalue problem.

5.4.2 TOAR as a linear eigenvalue problem solver

The key improvement of the TOAR algorithm over SOAR is that the basis for the Krylov subspace $\mathscr{K}_k(H,v)$ remains orthogonal as well, thus making the process numerically stable.

As we explained before, the upper Hessenberg matrix represents an approximation for the invariant pairs for both the quadratic problem and the corresponding linear problem. Although we use the projected quadratic problem to compute the approximation, the procedure to obtain the basis is still done on the linear problem, and breakdown means that we have found an invariant pair for the linear problem with the matrix H. However, we already discussed that the backward error for computed eigenpairs can be sufficiently small for the linearization, but much higher for the original problem. A solution to this problem is offered in the `quadeig` algorithm, which scales the matrices before using the algorithms for the linear problem.

Remark 5.3. The scaling is also important in the TOAR algorithm, because if the norms of the coefficient matrices are not equilibrated, the breakdown will occur before we find a good enough approximation for the quadratic problem.

This is why, as a first step, we propose scaling as described in Subsection 3.3.1.

5.4.3 TOAR as a quadratic solver

Choice of the approximation for the eigenpairs. The first exploitation of the fact that we are solving quadratic eigenvalue problem is that the approximation is obtained from the projected problem

$$(\lambda^2(Q_k^*MQ_k)+\lambda(Q_k^*CQ_k)+(Q_k^*KQ_k))z=\mathbf{0}. \tag{5.50}$$

For the solution of this small QEP we use our `KVADeig` algorithm described in the Section 3.5. It is important to solve this small problem correctly for it to be a better approximation for the original problem.

Eigenvector refinement. During the restarts, the subspace spanned by Q_k can have better information, however the approximate eigenvectors do not necessarily converge. This is why Jia proposed the eigenvector refinement in [45]. Let θ be the computed eigenvalue of (5.50). Then the corresponding vector z is computed to minimize the residual:

$$z = \arg \min_{\substack{z \in \mathbb{C}^k \\ \|z\|_2 = 1}} \|(\theta^2 M + \theta C + K) Q_k z\|_2. \tag{5.51}$$

Notice that (5.51) involves the original matrices, and that $Q_k z$ represents an approximation for an eigenvector of the original problem. The proposed procedure for computing z in (5.51) is via the eigenvector of the matrix B_k

$$B_k = X_k^* X_k = (\theta^2 M Q_k + \theta C Q_k + K Q_k)^* (\theta^2 M Q_k + \theta C Q_k + K Q_k), \tag{5.52}$$

associated with the smallest eigenvalue. It is important to underline the following facts regarding this procedure. First, as the process converges, X_k becomes increasingly ill-conditioned. The condition number is $\kappa_2(B_k) = \kappa_2(X_k)^2$. Secondly, because of ill-conditioning, there is no guarantee that the eigenvalue algorithm will compute the smallest eigenvalue and the corresponding eigenvector of B_k sufficiently accurately.

As an alternative to Jia's approach, [24] proposes another procedure which does not use the matrix B_k. It uses the QR factorization

$$\begin{pmatrix} MQ_k & CQ_k & KQ_k \end{pmatrix} = QR, \;\; R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \\ 0 & 0 & 0 \end{pmatrix}, \;\; R_{ii} \in \mathbb{C}^{k \times k}, \tag{5.53}$$

and the refinement is reduced to computing the smallest singular value with the corresponding right singular vector of the $3k \times k$ matrix

$$\theta^2 \begin{pmatrix} R_{11} \\ 0 \\ 0 \end{pmatrix} + \theta \begin{pmatrix} R_{12} \\ R_{22} \\ 0 \end{pmatrix} + \begin{pmatrix} R_{13} \\ R_{23} \\ R_{33} \end{pmatrix}. \tag{5.54}$$

Shift and invert. Suppose that we want to compute the eigenvalues closest to some σ, or we have an approximation for the wanted eigenvalue and we want to use that information to improve our iterative process. Then, we can define shifted and inverted quadratic eigenvalue problem in

the following way. Let $\lambda = \xi + \sigma$. Define

$$Q_\sigma(\xi) = \xi^2 M_\sigma + \xi C_\sigma + K,\ M_\sigma = \sigma^2 M + \sigma C + K,\ C_\sigma = 2\sigma M + C,\ K_\sigma = M. \qquad (5.55)$$

Now, computing the eigenvalues with largest magnitude of QEP (5.55) corresponds to computing the eigenvalues of the original problem closest to σ. The eigenvectors are the same, and for the eigenvalues we have $\lambda = 1/(\xi + \sigma)$. This transformation is important for computing the eigenvalues close to some target σ, because they become dominant and thus more easily computed by an iterative method.

Polynomial filter. By implicit restart described in Subsection 5.3.1 the polynomial filter $f(H) = \prod_{i=1}^{p}(H - \mu_i \mathbb{I})$ is applied to the starting vector. The idea is, if the μ_i's represent the unwanted eigenvalues, then this polynomial filter will remove the directions of the unwanted eigenvectors from the starting vector.
The most used shifts in implicitly restarted Arnoldi algorithms are the eigenvalues of the Hessenberg matrix T_m which are the approximation of the unwanted eigenvalues. These shifts are referred to as exact shifts. In practice, they work well for an arbitrary linear eigenvalue problems. *However, in the quadratic eigenvalue problem, we can have two eigenvalues sharing the same eigenvector. Therefore, this can pose a problem when applying a filter. We do not want to remove the directions of the wanted eigenvalues by removing the directions of the unwanted eigenvalues.* Let us look at this situation more closely.

Example 5.2. Suppose that two eigenvalues λ_1 and λ_2 share the same eigenvector x, and suppose that we chose λ_1 as the shift. The eigenvectors for the linearization H are different for these two eigenvalues; they are $\binom{\lambda_1 x}{x}$ and $\binom{\lambda_2 x}{x}$ respectively. This suggests that, by using this shift, the direction of the wanted eigenvector will not be removed, since it is not the same eigenvector for the linearization. Let us see what happens when $f(H) = (H - \lambda_1 \mathbb{I})$ is applied to the eigenvector $\binom{\lambda_2 x}{x}$

$$\left[\begin{pmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{pmatrix} - \lambda_1 \mathbb{I}_{2n}\right]\begin{pmatrix} \lambda_2 x \\ x \end{pmatrix} = \begin{pmatrix} \lambda_2 A x + B x - \lambda_1 \lambda_2 x \\ \lambda_2 x - \lambda_1 x \end{pmatrix} = \begin{pmatrix} \lambda_2^2 x - \lambda_1 \lambda_2 x \\ (\lambda_2 - \lambda_1) x \end{pmatrix}$$
$$= (\lambda_2 - \lambda_1)\begin{pmatrix} \lambda_2 x \\ x \end{pmatrix}.$$

Hence, this polynomial filter will not remove the eigenvector $\binom{\lambda_2 x}{x}$, however the factor $(\lambda_2 - \lambda_1)$ can be e.g. very small or very big. If it is small, and λ_2 is wanted eigenvalue, then this shift will reduce the direction of this eigenvector in starting vector. On the other hand, if the factor is big, and λ_2 is also unwanted eigenvalue, this will increase the contribution of another unwanted eigenvector in the starting vector.

These are the things that need to be considered when choosing shifts in TOAR as quadratic

solver. Another thing is that we choose the shifts from the information we get from the projected problem (5.50). The number of eigenvalues is $2m$, and amongst them we must determine k approximations for the wanted eigenvalues, and p approximations for the unwanted ones. Also, some of them can be meaningless for our original problem. For example, consider the projection onto one eigenvector x, $\lambda^2 x^*Mx + \lambda x^*Cx + x^*Kx = 0$. This is a quadratic equation, which means that there are two solutions λ_1, λ_2. However, only in special cases both of these roots are eigenvalues. Usually, only one of the roots represents a valid eigenvalue. If x is a common eigenvector, then both roots are eigenvalues.

These described phenomena are nicely seen in the special class of quadratic eigenvalue problems called overdamped problems.

5.4.4 Polynomial filter for overdamped problems

We introduced the overdamped quadratic eigenvalue problems in Section 1.7. The matrices M, C and K are symmetric, M and C are positive definite, and K is positive semidefinite. The overdamping condition

$$\min_{\|x\|_2=1} \left[(x^*Cx)^2 - 4(x^*Mx)(x^*Kx) \right] > 0 \tag{5.56}$$

is satisfied. The eigenvalues are divided into two sets. The n largest eigenvalues are called primary, and the n smallest are called secondary. An important property is that the n eigenvectors corresponding to the primary eigenvalues form a linearly independent set, and the n eigenvectors corresponding to secondary eigenvalues also form a linearly independent set.

Here, we propose a new strategy for choosing the shifts for the polynomial filter in the implicitly restarted TOAR algorithm. We present numerical examples which demonstrate the power of the new proposed shift selection strategy.

Recall that, if the starting vector in the Arnoldi procedure is a linear combination of k eigenvectors, the breakdown will occur at the kth step, and the eigenvalues of the matrix T_k will match k eigenvalues of the original problem, corresponding to those eigenvectors. Suppose that the starting vector for the Arnoldi and TOAR algorithm is $\begin{pmatrix} x \\ 0 \end{pmatrix}$, where x is an eigenvector for two eigenvalues λ_1 and λ_2. Now,

$$\begin{pmatrix} x \\ 0 \end{pmatrix} = \tau \left[\begin{pmatrix} \lambda_1 x \\ x \end{pmatrix} - \begin{pmatrix} \lambda_2 x \\ x \end{pmatrix} \right] = \tau \begin{pmatrix} (\lambda_1 - \lambda_2)x \\ 0 \end{pmatrix}, \tag{5.57}$$

where τ is a normalizing factor. By the Theorem 5.1 we conclude that the breakdown will occur in the second step of the Arnoldi/TOAR procedure because the starting vector is a linear combination of two eigenvectors, corresponding to λ_1 and λ_2, and they will be the eigenvalues of the Hessenberg matrix T_2. This shows that it is natural for the eigenvalues which share the

same eigenvector to appear together, and therefore if they are unwanted, they should be used as shifts together, and if one of them is wanted, the other one cannot be used as a shift. Also, if we use both eigenvalues as shifts, we avoid the possible increase by the factor $(\lambda_1 - \lambda_2)$ of the another unwanted vector or decreasing of the wanted eigenvector.

The process of choosing the shifts we propose goes as follows:

1. Compute $2m$ eigenvalues of the projected problem

$$(\lambda^2 Q_m^* M Q_k + \lambda Q_m^* C Q_m + Q_m^* K Q_m)x = \mathbf{0}.$$

   The structure of the problem is preserved by the orthogonal projection, meaning that this problem is also overdamped.

2. The structure of the eigenvalues is as described, we have a set of m primary and a set of m secondary eigenvalues. Sort the eigenvalues by magnitude. The first m belong to the primary, and the last m to the secondary eigenvalues. Choose k eigenvalues with the largest magnitudes as approximations for the wanted eigenvalues (assuming this as the selection criterion). The eigenvectors are computed from the SVD decomposition of the matrix (5.54).

3. The number of the shifts will always be even, let us say $2p$. First, choose the p eigenvalues farthest from the wanted k eigenvalues in the primary part. We know for sure that these eigenvalues do not share eigenvectors with the wanted eigenvalues.

4. Now, if there are eigenvalues sharing the eigenvector with these p shifts, we want to choose them. If they exist, they will be the roots of the quadratic polynomial $\lambda^2 x^* M x + \lambda x^* C x + x^* K x$, where x is the eigenvector. This is why, for every eigenvalue amongst already chosen shifts from the primary part we compute the eigenvector by refinement (5.51). Then we compute the roots of the mentioned quadratic polynomial, and these roots are now the shifts. So, at the end, we have $2p$ shifts, for which we are sure that do not share the eigenvector with the wanted eigenvalues. This step can be also understood as a refinement step for computing the unwanted eigenvalues.

Here, we described how to choose shifts if the eigenvalue with the largest magnitude are of interest. This can work for any other feature prescribed for the wanted eigenvalue, we just a adjust the sorting criteria.

Tropical roots for shift and invert. In this section we propose a new selection of approximation for defining the shifted and inverted problem in order to get the better approximation for the wanted eigenvalues.

When we discussed the parameter scaling for equilibration of the backward errors for the quadratic problem and the corresponding linearization, we mentioned roots of the tropical polynomial as one of the options.

Another interesting fact is that the tropical roots can be good approximations for the moduli of the eigenvalues of the quadratic problem, as investigated in [54].

For the overdamped problems, we know that all eigenvalues are real and in the left half plane. Therefore, if we want to find eigenvalues with largest magnitudes, an upper bound on their moduli is the larger tropical root, and therefore we propose them to be used as shifts to define shifted and inverted QEP (5.55).[1]

We already said that scaling must be done before calling TOAR to avoid early breakdown, therefore the norms of matrices will be computed in any case. We can use it then to compute the larger tropical root

$$\gamma = \frac{\|C\|_2}{\|M\|_2}, \tag{5.58}$$

if $\tau = \frac{\|C\|_2}{\sqrt{\|M\|_2\|K\|_2}} > 1$. Then $\sigma = -\gamma$ is a good shift for shifted and inverted problem.

5.4.5 Numerical examples for overdamped problems

Example 1. First example is from Bai and Su's first paper on the Second Order Arnoldi algorithm [3]. The problem is of order $n = 50$ and the matrices M, C and K are defined as

$$M = 0.1 \cdot \mathbb{I}, \;\; C = \mathbb{I}, \;\; K = \text{tridiag}(-0.1, 0.2, -0.1). \tag{5.59}$$

Here, the kth largest and the kth smallest eigenvalues share the same eigenvector. In Figure 5.2, we show all 100 eigenvalues.

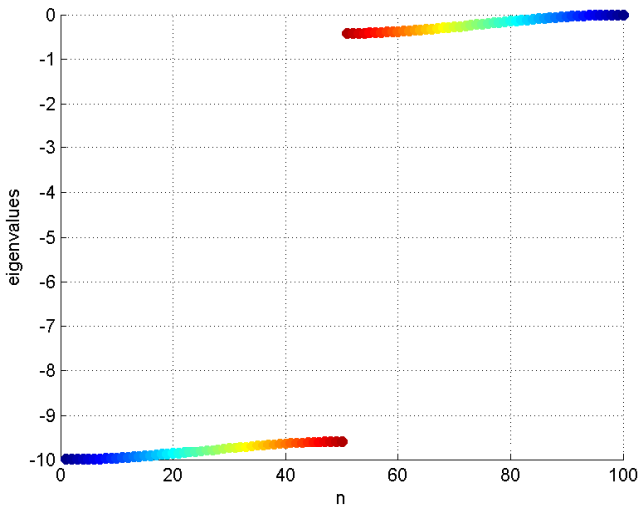


Figure 5.2: All eigenvalues of QEP (5.102)

The eigenvalues marked by the same color share the eigenvector. We compared our algorithm with MATLAB's `eigs` which is an implementation of the implicitly restarted Arnoldi algorithm.

[1] It is noted in [54] that the tropical roots are also used as the starting point in the Ehrlich–Aberth method.

The shifts are the exact shifts, that is some of the eigenvalues of the Hessenberg matrix in the Arnoldi factorization.

The goal was to find $m = 2$ eigenvalues with largest magnitudes. The number of shifts in both cases was $p = 4$, and the maximal dimension of the factorization was $k = 6$. The tolerance for the backward error was $n \times \mathbf{u}$ where $\mathbf{u}$ is the machine precision.

`eigs` did not find the requested eigenpairs for which the backward error is small enough, even after 300 restarts, producing the error message

```
??? Error using ==> eigs>processEUPDinfo at 1453 Error with ARPACK
routine dneupd: dnaupd did not find any eigenvalues to sufficient
accuracy.
```

We plotted the residuals for the first 23 restarts in Figure 5.3 for better illustration of the superior performance of our new method. After the first 23 restarts, the backward error produced by TOAR with our new filtering is already below 10^{-8}, while in `eigs` the error is around 10^{-3} and it does not improve during the remaining 277 iterations.

We also called TOAR on the shifted and inverted QEP with the shift $\sigma = -10$, which is a greater tropical root for this problem. With the same setting, approximations where found in just 3 iterations. The backward errors are present in Figure 5.4.

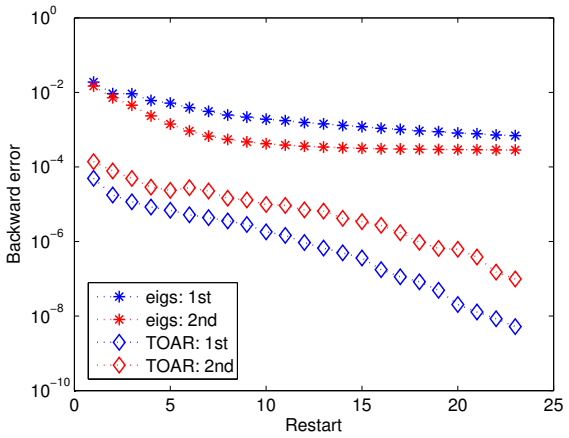


Figure 5.3: Backward errors for first 23 iterations of `eigs`

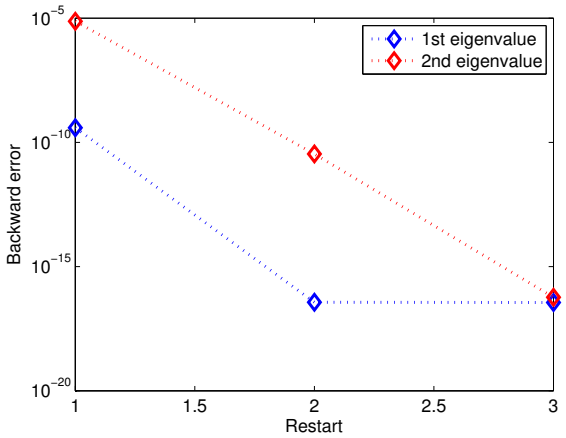


Figure 5.4: Backward errors for shift and invert with tropical root

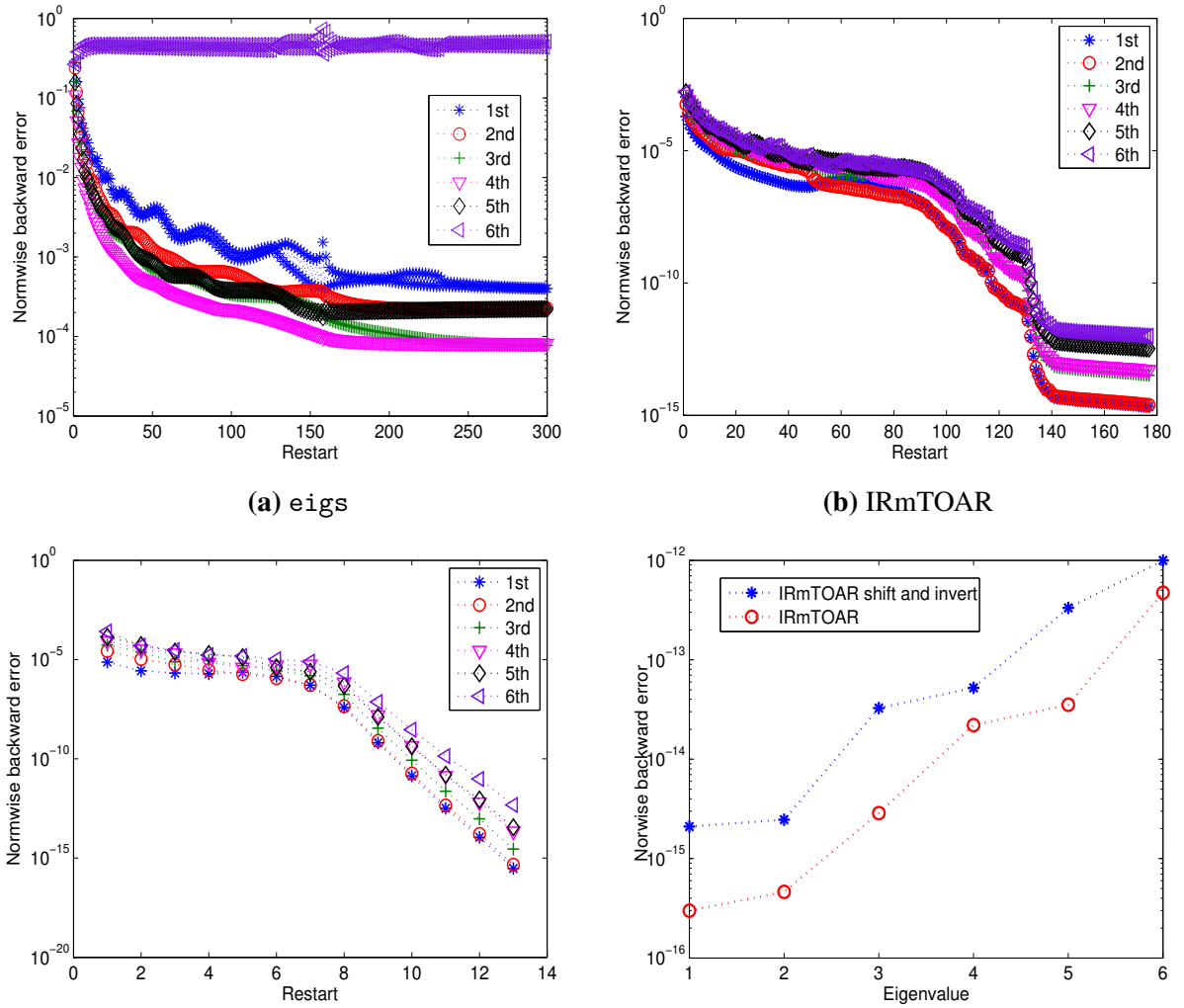
Example 2. The next example is of order $n = 400$. The matrices are

$$M = \mathbb{I}, \quad C = \text{tridiag}(-10, 30, -10), \quad K = \text{tridiag}(-5, 15, -5). \tag{5.60}$$

We want to compute $k = 6$ eigenvalues using $p = 6$ shifts with the maximal dimension $m = 12$. We used `eigs` and our new implementation of the implicitly restarted TOAR (we will refer to it as mTOAR). Depending on the starting vector, `eigs` sometimes finds good enough approximations, and sometimes not, in 300 iterations. On the other hand, mTOAR always finds

the approximations with a smaller number of restarts. For one example where `eigs` did not converge, mTOAR found satisfactory approximation in the first 177 restarts. The tolerance for the backward error was 10^{-12}.

When we used the tropical root as a shift, mTOAR needed only 13 restarts. For this example we provide figures with backward errors in every restart for every wanted eigenvalue.



(a) `eigs`

(b) IRmTOAR

(c) shifted and inverted IRmTOAR

(d) Final normwise backward errors

Figure 5.5: Normwise backward errors in every restart for all computed eigenvalues

5.5 Locking in IRA

When an element on the subdiagonal of the Hessenberg matrix T_k in the Arnoldi process is small, we know that we have found a good enough approximation for some eigenvalue of the original problem. However, a Ritz value may be close to an eigenvalue of the original problem without small elements appearing on the subdiagonal of T_k. Lechoucq and Sorenesen [47] developed the so called locking procedure, which applies a certain orthogonal change of basis so that the appropriate subdiagonal element of T_k is (close to) zero. The following lemma

is important for the derivation of this process.

Lemma 5.1 ([47]). *Let $T_k z = \theta z$ where $T_k \in \mathbb{R}^{k\times k}$ is an unreduced upper Hessenberg matrix and $\theta \in \mathbb{R}$ with $\|z\|_2 = 1$. Let W be a Householder matrix such that $Wz = e_1\tau$ where $\tau = -\operatorname{sign}(e^T z)$. Then*

$$e_k^T W = e_k^T + w^T, \tag{5.61}$$

where $\|w\| \le \sqrt{2}|e_k^T z|$ and

$$W^T H W e_1 = \theta e_1. \tag{5.62}$$

Suppose that we have the Arnoldi factorization of order k as in (5.4). Let (θ, z) be an eigenpair for T_k with $|e_k^T z|$ small enough so that the residual (5.5) for $(\theta, V_k z)$ is small enough. Define W as in Lemma 5.1, and multiply the factorization (5.4) to get

$$HV_k W = V_k W(W^T H W) + r_k e_k^T W. \tag{5.63}$$

Using (5.61) and (5.62) we get

$$HV_k W = V_k W \begin{pmatrix} \theta & \bar{t}^T \\ \mathbf{0} & \overline{T}_{k-1} \end{pmatrix} + r_k e_k^T + r_k w^T. \tag{5.64}$$

For (5.64) to be an Arnoldi factorization, the matrix $\overline{T}_{k-1}$ must be upper Hessenberg, and the term $r_k w^T$ must be dropped. When restoring $\overline{T}_{k-1}$ to Hessenberg form, we must be careful not to change the matrix $r_k e_k^T$. The transformation matrix Y is thus defined as $Y = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & Y_1 Y_2 \ldots Y_{k-3} \end{pmatrix}$ where Y_1 is such that

$$Y_1^T \overline{T}_{k-1} Y_1 = \begin{pmatrix} \overline{G} & \overline{g} \\ \overline{\beta}_k e_{k-2}^T & \gamma \end{pmatrix}, \tag{5.65}$$

and $e_{k-1}^T Y_1 e_{k-1}^T = 1$. The matrices $Y_2, \ldots, Y_{k-3}$ are defined analogously. Since $\|r_k w^T Y\|_2 = \|r_k\|_2 \|Y^T w\|_2 = \|r_k\|_2 \|w\|_2$, the size of $\|r_k w^T\|_2$ remains the same. By updating

$$V_k = V_k W Y, \;\; H_k = Y^T W^T H_k W Y, \;\; w^T = w^T Y,$$

and by discarding the term $r_k w^T$, we get a factorization in which the eigenvalue θ is locked. The following theorem shows that this process constructs the Arnoldi factorization of an nearby matrix.

Theorem 5.3 ([47]). *Let $HV_k = V_k T_k + r_k e_k^T + r_k w^T$ be an Arnoldi factorization where $T_k z = \theta z$ and $\sqrt{2}|e^T z|\|r_k\|_2 \le \varepsilon \|H\|_2$ for some $\varepsilon > 0$. Then there exists a matrix $E \in \mathbb{R}^{n\times n}$ such that*

$$(H + E)V_k = V_k T_k + r_k e_k^T, \tag{5.66}$$

where $\|E\|_2 \le \varepsilon \|A\|_2$.

The general algorithm is then as follows. Suppose that we have already locked j eigenvalues, and that the partitioned Arnoldi factorization is

$$H\begin{pmatrix} V_j & \overline{V}_{k-j}\end{pmatrix} = \begin{pmatrix} V_j & \overline{V}_{k-j}\end{pmatrix}\begin{pmatrix} T_j & G_j \\ \mathbf{0} & \overline{T}_{k-j}\end{pmatrix} + r_k e_k^T + r_k w^T. \tag{5.67}$$

The matrix $T_j \in \mathbb{R}^{j\times j}$ contains previously locked eigenvalues, and $\overline{T}_{k-j}$ is unreduced upper Hessenberg matrix. The columns of V_j represent the Schur basis for the locked invariant subspace. Let the columns of $X_i \in \mathbb{R}^{(k-j)\times i}$ represent the eigenvectors corresponding to the new i eigenvalues which we want to lock. The new factorization is obtained in the following 4 steps:

1. Compute the orthogonal factorization

$$Q\begin{pmatrix} R_i \\ \mathbf{0}_{k-j-i}\end{pmatrix} = X_i,$$

   where $Q \in \mathbb{R}^{(k-j)\times(k-j)}$.

2. Update the factorization (5.67): $\overline{T}_{k-j} = Q^T\overline{T}_{k-j}Q$, $\overline{V}_{k-j} = \overline{V}_{k-j}Q$, $G_j = G_jQ$.

3. Compute an orthogonal matrix $P \in \mathbb{R}^{(k-j-i)\times(k-j-i)}$ that restores $\overline{T}_{k-j-i}$ to Hessenberg form.

4. Update the factorization: $\overline{T}_{k-j-i} = P^T\overline{T}_{k-j-i}P, \overline{V}_{k-j-i} = \overline{V}_{k-j-i}P, G_{j+i} = G_{j+i}P$.

5.5.1 Locking in TOAR

In this subsection we develop and analyze, analogously, a locking procedure in the new implicitly restarted TOAR algorithm.

Assume that we built TOAR factorization of order m

$$AQ_mU_{m,1} + BQ_mU_{m,2} = Q_mU_{m,1}T_m + se_m^Tt_{m+1,m}, \tag{5.68}$$

$$Q_mU_{m,1} = Q_mU_{m,2}T_m + ue_m^Tt_{m+1,m}, \tag{5.69}$$

and that an eigenpair (θ, Q_mz) from the projected problem $(\theta^2Q_m^TMQ_m + \theta Q_m^TCQ_m + Q_m^TKQ_m)z = \mathbf{0}$ is a good approximation for the original problem. We would like to lock this eigenpair in the similar way to locking for the standard eigenvalue problem. That is, we want to introduce a small element onto the subdiagonal of the Hessenberg matrix T_m.

The first problem is that the eigenpair (θ, Q_mz) is obtained from the projected quadratic problem, and not from the matrix T_m. In order to proceed with locking, we first need to make sure that θ is an eigenvalue of T_m. The eigenvector for the corresponding linearization H for the eigenvalue θ is $\begin{pmatrix}\theta Q_mz \\ Q_mz\end{pmatrix}$. This means that, if θ is an eigenvalue of T_m, the corresponding

eigenvector would be

$$y = \begin{pmatrix} Q_m U_{m,1} \\ Q_m U_{m,2} \end{pmatrix}^T \begin{pmatrix} \theta Q_m z \\ Q_m z \end{pmatrix} = \begin{pmatrix} U_{m,1} \\ U_{m,2} \end{pmatrix}^T \begin{pmatrix} \theta z \\ z \end{pmatrix}. \tag{5.70}$$

Since (5.68)-(5.69) represents the Arnoldi factorization for the matrix H, the residual for the eigenpair $(\theta, \begin{pmatrix} Q_m U_{m,1} \\ Q_m U_{m,2} \end{pmatrix} y)$ (5.5) is small enough if $|e_m^T y|$ is small. Finally, we conclude that θ can be regarded as an eigenvalue of T_m if the last component of (5.70) is small. If this is the case, we can continue with locking. Suppose that W is as in Lemma 5.1. The transformed TOAR factorization is

$$AQ_m U_{m,1} W + BQ_m U_{m,2} W = Q_m U_{m,1} W W^T T_m W + s e_m^T t_{m+1,m} + s w(1:n)^T, \tag{5.71}$$

$$Q_m U_{m,1} W = Q_m U_{m,2} W W^T T_m W + u e_m^T t_{m+1,m} + u w(n+1:2n)^T, \tag{5.72}$$

and

$$W^T T_m W = \begin{pmatrix} \theta & \bar{t}^T \\ \mathbf{0} & \overline{T}_{m-1} \end{pmatrix}. \tag{5.73}$$

As described in the linear case, $W^T T_m W$ must be returned to upper Hessenberg form, making sure that we do not change the terms $s e_m^T$ and $u e_m^T$. Denote with Y the transformation matrix. By removing the terms $s w^T$ and $u w^T$ and by updating

$$U_{m,1} = U_{m,1} W Y, \;\; U_{m,2} = U_{m,2} W Y, \;\; T_m = Y^T W^T T_m W Y, \;\; w^T = w^T Y,$$

we have locked the eigenvalue θ.
However, with this procedure we did not change the matrix Q_m. And the next time we compute the approximation, we must again compute $2m$ eigenvalue from the projected problem, and thus, we will again compute the locked eigenvalue. With this locking we have only assured that the implicit restart will not affect the locked part of the Hessenberg matrix T_m in the factorization.

5.6 Rayleigh damping

Consider the quadratic eigenvalue problem $(\lambda^2 M + \lambda C + K)x = 0$ with proportional damping $C = \alpha M + \beta K$, also known as Rayleigh damping. This problem reduces to the linear pencil

$$Kx = \mu M x, \;\; \mu = -\frac{\lambda^2 + \lambda\alpha}{\lambda\beta + 1}. \tag{5.74}$$

The eigenvalues for the original quadratic problem are restored as

$$\lambda_{1,2} = \frac{-(\alpha + \beta\mu) \pm \sqrt{(\alpha+\beta\mu)^2 - 4\mu}}{2}, \;\; \text{for } \mu \text{finite and nonzero,} \tag{5.75}$$

$$\lambda_1 = 0,\ \lambda_2 = -\alpha,\ \text{for } \mu = 0, \tag{5.76}$$

$$\lambda_1 = \infty,\ \lambda_2 = -\frac{1}{\beta},\ \text{for } \mu = \infty. \tag{5.77}$$

Since the proportional damping is easier to handle numerically, we would like to exploit the information about the wanted part of the spectrum for the problems which are close to proportionally damped. Namely, for given quadratic problem $(\lambda^2 M + \lambda C + K)x = 0$ we want to determine the smallest ΔC so that $(\lambda^2 M + \lambda(C + \Delta C) + K)x = 0$ is proportionally damped. This is done by minimizing

$$\|C - (\alpha M + \beta K)\|_F \to \min,\ |\alpha|^2 + |\beta|^2 \to \min, \tag{5.78}$$

over α, β. By application of the projection theorem in $\mathbb{C}^{n\times n}$, equipped with the Frobenius inner product $\langle A, B\rangle_F = \mathrm{Tr}(B^* A)$, in [24] the following normal equations were derived

$$\begin{pmatrix} \langle M, M\rangle_F & \langle K, M\rangle_F \\ \langle M, K\rangle_F & \langle K, K\rangle_F \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \langle C, M\rangle_F \\ \langle C, K\rangle_F \end{pmatrix}. \tag{5.79}$$

Now, the algorithm for using the approximation of the quadratic problem by proportionally damped one would go as follows:

- Suppose that we want to compute k eigenvalues with largest magnitude
- Compute α, β from (5.79).
- Call implicitly restarted Arnoldi to compute k eigenpairs (λ_i, x_i) with largest magnitude for (5.74)
- Define new starting vectors $r_{-1} = \sum_{i=1}^{m} \lambda_i x_i,\ r_0 = \sum_{i=1}^{m} x_i$ and call implicitly restarted mTOAR on the original problem with these starting vectors.

We will refer to this algorithm as mTOAR NRD. The numerical examples are presented in the following subsection.

5.6.1 Numerical examples

Experiment 1. The first example is `Path crossing`, from [44]. M and K are given as `BCSSTM12` and `BCSSTK12` from the Harwell–Boeing collection [27], and C is a block combination of M and K. The matrices are of order 1473. Define $M_1 = M(1:600, 1:600)$ and $M_2 = M(540:1473, 540:1473)$, and K_1, K_2 in the same way. Then $C = [c_{ij}]$ is defined as

$$c_{ij} = \begin{cases} a_{11} m_{ij} + a_{12} k_{ij}, & \text{when } i < 540 \text{ or } j < 540, \\ (a_{11} + a_{21}) m_{ij} + (a_{12} + a_{22}) k_{ij}, & \text{when } 540 \le i, j \le 600, \\ a_{21} m_{ij} + a_{22} k_{ij}, & \text{when } i > 600 \text{ or } j > 600, \end{cases}$$

where $\binom{a_{i1}}{a_{i2}} = \frac{2\xi_i}{\omega_1+\omega_2}\binom{\omega_1\omega_2}{1}$ with $\xi_1 = 0.05$, $\xi_2 = 0.10$, and ω_1 and ω_2 are the first and tenth natural frequencies for the undamped problem $(\mu^2 M + K)x = 0$.

We want to compute $k = 6$ eigenvalues of largest magnitudes. The maximal dimension of TOAR and Arnoldi factorizations is set to $m = 18$. The number of shifts in TOAR is set to $2p = 8$, and the number of shifts in `eigs` is always $m - k = 12$.

We started TOAR and `eigs` with the same starting vectors $r_{-1} = \text{rand}(n,1)$ and $r_0 = \text{rand}(n,1)$. In addition, we called TOAR with starting vector as described in Section 5.6. More precisely, we computed $\alpha = 0.340395988262736$ and $\beta = 0.340395988262736$ so that (5.79) holds. We called `eigs` on $Kx = \mu Mx$. The tolerance on the normwise backward error was $\sqrt{\texttt{eps}}$, where `eps` is the machine precision. Algorithm found the wanted eigenvalues with prescribed tolerance in 7 restarts. The tolerance for the normwise backward error of the original problem was $n \times \texttt{eps} = 3.2707\texttt{e-013}$. The following table presents the number of restarts needed to find the eigenpairs with prescribed tolerance

Table 5.1: Number of restarts, Path crossing

Algorithm	**IRmTOAR**	**IRmTOAR NRD**	`eigs`
No. restarts	68	56 = (7+49)	118

The following figure represents the final backward errors for all 6 wanted eigenvalues obtained by all three methods

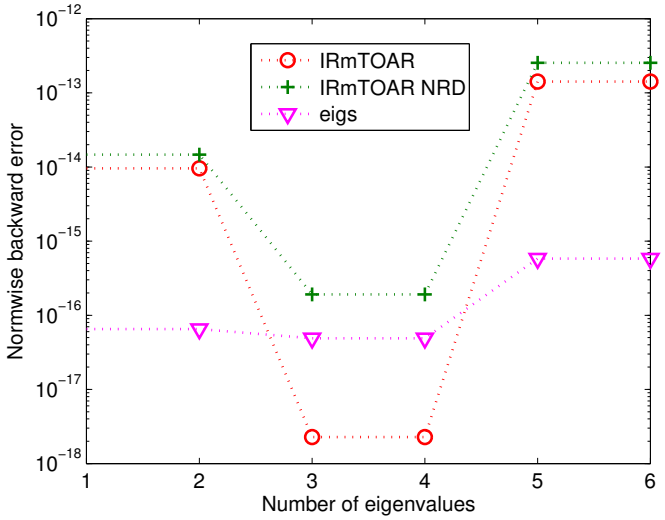


Figure 5.6: Final normwise backward errors, Path crossing

At last, we present the backward errors during the restarts for mTOAR, and mTOAR NRD for 3 complex conjugate pairs of wanted eigenvalues.

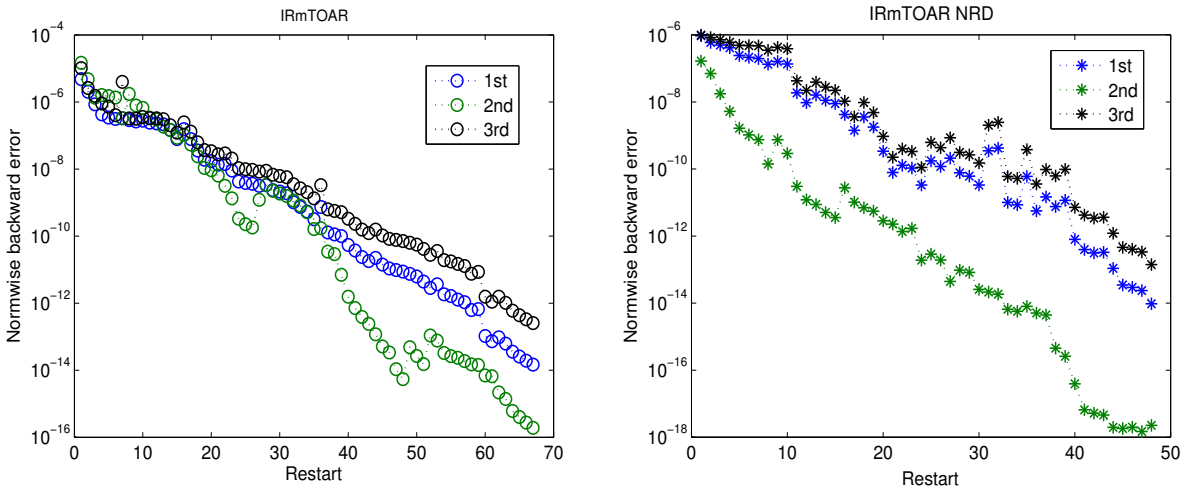


Figure 5.7: Normwise backward errors in every restart for all computed eigenvalues

Experiment 2. Next example is `cd_player` from NLEVP library. We started the algorithm with the same parameters as in previous example, except the number of wanted eigenvalues and shifts, which are $k = 4$, $m = 10$, $2p = 2$. The following table presents the number of restarts needed to find the eigenpairs with prescribed tolerance $n \times$ `eps` $=$ `1.3323e-014`

Table 5.2: Number of restarts, `cd_player`

Algorithm	**IRmTOAR**	**IRmTOAR NRD**	**`eigs`**
No. restarts	22	8 = (5+3)	23

The following figure represents the final backward errors for all 4 wanted eigenvalues obtained by all three methods

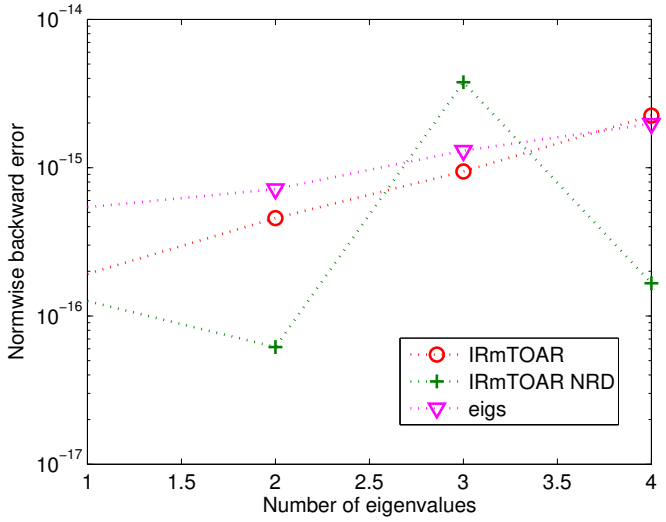


Figure 5.8: Final normwise backward errors, `cd_player`

At last, we present the backward errors during the restarts for TOAR, and TOAR with Rayleigh Damping approximation for 4 wanted eigenvalues.

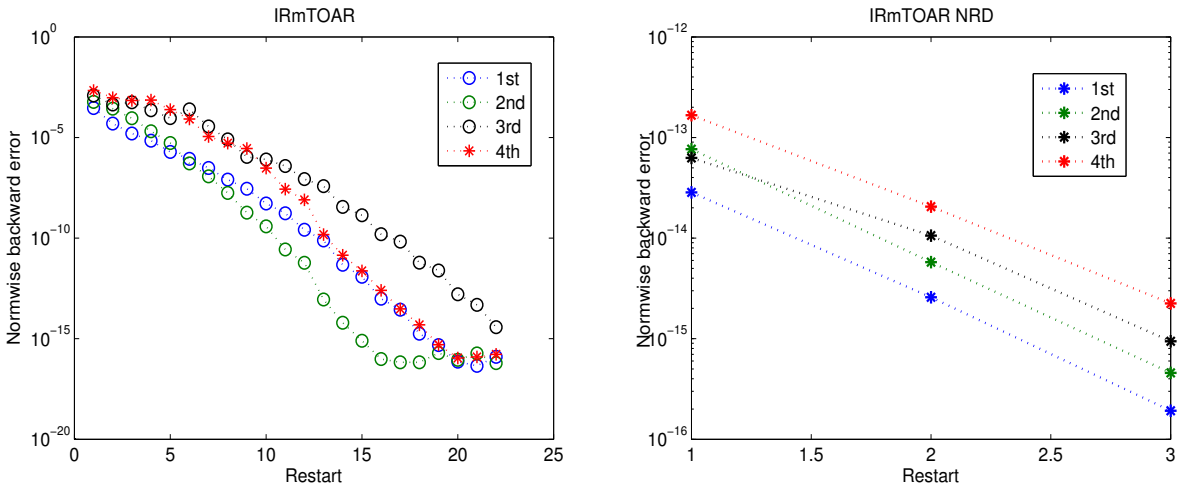


Figure 5.9: Normwise backward errors in every restart for all computed eigenvalues, `cd_player`

5.7 Krylov–Schur algorithm for the linear eigenproblem

In [64] Stewart defined the *Krylov decomposition* of order k for the an $n \times n$ matrix H as

$$HU_k = U_k B_k + u_{k+1} b_{k+1}^*, \tag{5.80}$$

where B_k is $k \times k$ matrix, $U_k \in \mathbb{C}^{n\times k}$, $u_{k+1}, b_{k+1} \in \mathbb{C}^n$, and the columns of $(U_k \;\; u_{k+1})$ are linearly independent. The idea of this decomposition is to weaken the constraints on the matrices U_k and B_k prescribed by the Arnoldi decomposition, where U_k has to be orthogonal, and B_k has to be upper Hessenberg. Due to this constraints, we always have to be careful when restarting, locking or purging Arnoldi process in order to maintain its structure.

It is proven in [64] that the Krylov decomposition is closed under translation, i.e. for $\gamma\widetilde{u}_{k+1} = u_{k+1} - U_k g$, $\gamma \neq 0$

$$HU_k = U_k(B_k + g b_{k+1}^*) + \widetilde{u}_{k+1}(\gamma b_{k+1})^*$$

is a Krylov decomposition with the same space as (5.80). Moreover, the Krylov decomposition is closed under the similarity as well, i.e. for nonsingular W

$$H(U_k W^{-1}) = (U_k W^{-1})(W B_k W^{-1}) + u_{k+1}(b_{k+1}^* W^{-1})$$

is a Krylov decomposition whose space is the same as (5.80).

This makes Krylov decomposition equivalent to Arnoldi decomposition (i.e., the Rayleigh quotients are similar). In addition, using these elementary transformations, we can reduce Krylov decomposition into a form that is the most convenient for the truncation step in the implicit restart. Namely, we can keep the columns of U_k orthonormal, and reduce B_k to Schur form. The resulting decomposition is called *Krylov–Schur decomposition*.

Implicitly restarted Krylov–Schur algorithm. Just like in the implicitly restarted Arnoldi algorithm, the Krylov–Schur method consists of the expansion phase and the contraction phase. In the expansion phase, the Krylov–Schur decomposition of order k is constructed, using the Arnoldi algorithm 5.1.1. The contraction phase purges the unwanted eigenvalues from the decomposition. An advantage of the Krylov–Schur scheme is that it can be truncated at any point. Suppose we partitioned the Kyrlov–Schur decomposition in the form

$$H\begin{pmatrix} U_1 & U_2 \end{pmatrix} = \begin{pmatrix} U_1 & U_2 \end{pmatrix}\begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} + u\begin{pmatrix} b_1^* & b_2^* \end{pmatrix}, \tag{5.81}$$

then

$$HU_1 = U_1 S_{11} + ub_1^* \tag{5.82}$$

is a Krylov–Schur decomposition of order k. Moreover, this truncation step is equivalent to applying the shifted QR to the Hessenberg matrix T_m in the implicitly restared Arnoldi algorithm in order to get a new decomposition with better starting vector. The shifts are the eigenvalues of the matrix S_{22}. This is summarized in the following theorem.

Theorem 5.4 ([11]). *Let the Krylov decomposition $HU = UB + ub^*$ be partitioned as*

$$H\begin{pmatrix} U_1 & U_2 \end{pmatrix} = \begin{pmatrix} U_1 & U_2 \end{pmatrix}\begin{pmatrix} B_{11} & B_{12} \\ \mathbf{0} & B_{22} \end{pmatrix} + u\begin{pmatrix} u_1^* & u_2^* \end{pmatrix}, \tag{5.83}$$

where $U_1 \in \mathbb{C}^{n\times k}$, $B_{11} \in \mathbb{C}^{k\times k}$, $u_1 \in \mathbb{C}^k$ and the columns of $U = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \in \mathbb{C}^{n\times m}$, $m = k+\ell$, span a Krylov subspace $\mathscr{K}_m(H,v)$ which is not H–invariant. Then,

$$\mathrm{Im}(U_1) = \mathscr{K}_k(H, \kappa_{B_{22}}(H)v), \quad \kappa_{B_{22}}(\xi) = \det(\xi\mathbb{I} - B_{22}). \tag{5.84}$$

Further, if $\mathrm{Im}(U_1) = \mathscr{K}_k(H,\pi(H)v)$, for some monic polynomial π of degree ℓ, then $\pi = \kappa_{B_{22}}$. Thus, $AU_1 = U_1B_{11} + ub_1^*$ is an implicitly restarted Krylov decomposition with $\mathrm{Im}(U_1) = \mathscr{K}_k\left(H, \prod_{i=1}^{\ell}(H-\sigma_i\mathbb{I})v\right)$ if and only if $\sigma_1,\ldots,\sigma_\ell$ are the eigenvalues of B_{22}.

Thus, in order to apply the shifts which are approximations of the unwanted eigenvalues, the eigenvalues of the matrix S_{11} must be the wanted ones. This is accomplished by using the ordered Schur form. In the ordered Schur form the cluster of eigenvalues appears in the leading elements on the diagonal of the upper triangular matrix [2].

Let (5.81) represent the desired form, that is, let the eigenvalues of S_{11} represent the approximation of wanted eigenvalues. The truncation step is illustrated in the following figure

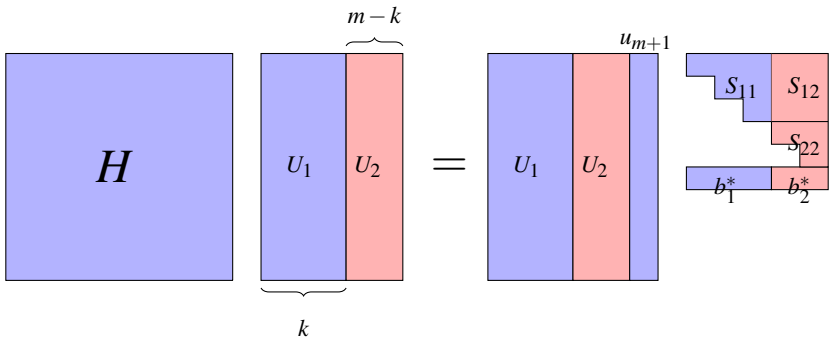


Figure 5.10: Truncation step in Krylov–Schur algorithm

It is clear that the truncation process in the Krylov–Schur algorithm is more elegant and easier that in the implicitly restarted Arnoldi algorithm, since we are not limited by the structure of the matrices. However, the flaw of this approach is that the only shifts which can be used are the exact ones, i.e., the eigenvalues of B_m in (5.80); on the other and, in the Arnoldi algorithm we can use arbitrary shifts. Since the number of iterations in the Arnoldi like algorithms depends on the shifts used in restart, it would be convenient if we could choose any shifts for the restart in the Krylov–Schur algorithm as well.

The Krylov–Schur method is implemented in the Scalable Library for Eigenvalue Problem Computations (SLEPc) [38].

5.7.1 Using the arbitrary shifts in Krylov–Schur algorithm

Bujanović and Drmač developed a new restarting procedure for Krylov–Schur algorithm using the arbitrary shifts in [11], using Theorem 5.4. We briefly outline the main steps; for more details we refer to [11].

Suppose we have an orthogonal Krylov decomposition

$$HU_m = U_m B_m + u_{m+1} b_m^*, \tag{5.85}$$

and let $\sigma_1, \ldots, \sigma_{m-k}$ be the shifts that we want to apply. We now preform the **4R-procedure** proposed in [11].

1. **Reassign**
   Apply an eigenvalue assignment algorithm to compute the vector f so that the spectrum of $B_m + f b_m^*$ contains the shifts $\sigma_1, \ldots, \sigma_{m-k}$; then use f to translate (5.85) to

$$HU_m = U_m(B_m + f b_m^*) + (u_{m+1} - U_m f) b_m^*. \tag{5.86}$$

   Re-assignment of the eigenvalues of B_m is possible if and only if the pair (B_m^*, b_m) is

controllable, i.e. if

$$\mu(B_m^*, b_m) \equiv \inf_{\zeta\in\mathbb{C}} \sigma_{min}\left((\zeta I - B_m^*,\ \ b_m)\right) \equiv \inf_{\zeta\in\mathbb{C}} \sigma_{min}\left(\begin{pmatrix}\zeta I - B_m \\ b_m^*\end{pmatrix}\right) > 0. \tag{5.87}$$

f is determined in two steps. First, compute unitary W such that $W^*B_m^*W = H$ is upper Hessenberg and $W^*b_m = \beta e_1$, $\beta = \|b_m\|_2$. This is called a reduction to Controller–Hessenberg form. The second step is computing the vector g such that $\overline{\sigma}_1, \ldots, \overline{\sigma}_{m-k}$ are eigenvalues of $H + e_1 g^*$. This can be done by using an eigenvalue assignment algorithm described in e.g. [16]. The wanted f is $f = \frac{1}{\beta} g^* W^*$.

2. **Reorder**
   In this step we compute the ordered Schur decomposition of $B_m + f b_m^*$, so that the shifts $\sigma_1, \ldots, \sigma_{m-k}$ appear as the eigenvalues of the S_{22} block in the Schur form S

$$B_m + f b_m^* = (Q_1 \ \ Q_2) \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} (Q_1 \ \ Q_2)^*. \tag{5.88}$$

3. **Restart**
   Multiply (5.86) with Q_1 to get the restarted Krylov–Schur decomposition

$$H\widehat{U}_k = \widehat{U}_k S_{11} + \widetilde{u}_{k+1} b_m^* Q_1,$$

   where $\widehat{U}_k = U_m Q_1$, $\widetilde{u}_{k+1} = u_{m+1} - U_m f$.

4. **Restore**
   Another translation is needed to restore the orthogonal Krylov decomposition. Let $\widehat{u}_{k+1} = \frac{\widetilde{u}_{k+1} - \widehat{Q}_k g_1}{\gamma}$ be the result of the Gram–Schmidt orthogonalization of the vector $\widetilde{u}_{k+1}$ against $\mathrm{Im}(\widehat{U}_k)$, with normalizing factor $\gamma = \|\widetilde{u}_{k+1} - \widehat{U}_k g_1\|_2$. Then

$$H\widehat{U}_k = \widehat{U}_k \left(S_{11} + g_1 \widehat{b}_k^*\right) + \gamma \widehat{u}_{k+1} \widehat{b}_k^*, \tag{5.89}$$

   where $\widehat{b}_k = Q_1^* b_m$.

5.8 Implicitly restarted Krylov–Schur algorithm for the QEP

Campos and Roman [14] extended the Krylov–Schur algorithm for the solution of polynomial eigenvalue problems. In order to build the starting factorization they use the TOAR Algorithm 5.3.1 with the first companion form linearization. Here, we give details of the algorithm for the quadratic eigenvalue problem $Q(\lambda) = \lambda^2 M + \lambda C + K$.

Let H be the linearization matrix

$$H = \begin{pmatrix} -M^{-1}C & -M^{-1}K \\ -\mathbb{I} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{pmatrix}. \tag{5.90}$$

Let

$$HV_m = V_m S_m + v_{m+1} b_{m+1}^* \tag{5.91}$$

be the Krylov–Schur decomposition for H, of order m, i.e. $B_m \in \mathbb{C}^{m\times m}$, and $\begin{pmatrix} V_m & v_{m+1} \end{pmatrix} \in \mathbb{C}^{2n\times m+1}$ has linearly independent columns. Partition the decomposition (5.91) to get

$$\begin{pmatrix} A & B \\ \mathbb{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} V_{m,1} \\ V_{m,2} \end{pmatrix} = \begin{pmatrix} V_{m,1} \\ V_{m,2} \end{pmatrix} S_m + v_{m+1} b_{m+1}^*. \tag{5.92}$$

Stewart proved that the decomposition (5.91) is equivalent to Arnoldi decomposition. Let

$$H\overline{V}_m = \overline{V}_m T_m + \overline{v}_{m+1} e_m^T \tag{5.93}$$

be the corresponding Arnoldi decomposition. We can thus conclude that (5.91) is also equivalent to TOAR factorization by extracting Q_{m+1} by the rank revealing decomposition of

$$\left(\begin{array}{cc|cc} \overline{V}_m(1:n,:) & \overline{v}_{m+1}(1:n) & \overline{V}_k(n+1:2n,:) & \overline{v}_{m+1}(n+1:2n) \end{array} \right).$$

Hence, we can build the Krylov decomposition for H in (5.90) using the TOAR algorithm as well

$$H \begin{pmatrix} Q_m U_{m,1} \\ Q_m U_{m,2} \end{pmatrix} = \begin{pmatrix} Q_{m+1} \begin{pmatrix} U_{m,1} & u_{m+1,1} \\ 0 & \beta_{m+1} \end{pmatrix} \\ Q_{m+1} \begin{pmatrix} U_{m,2} & u_{m+1,2} \\ 0 & 0 \end{pmatrix} \end{pmatrix} \begin{pmatrix} T_m \\ t_{m+1,m} e_m^T \end{pmatrix}. \tag{5.94}$$

The corresponding Krylov–Schur decomposition is then obtained by computing the Schur form $T_k = X S_k X^*$ and transforming

$$H \begin{pmatrix} Q_m U_{m,1} X \\ Q_m U_{m,2} X \end{pmatrix} = \begin{pmatrix} Q_{m+1} \begin{pmatrix} U_{m,1} X & u_{m+1,1} \\ 0 & \beta_{m+1} \end{pmatrix} \\ Q_{m+1} \begin{pmatrix} U_{m,2} X & u_{m+1,2} \\ 0 & 0 \end{pmatrix} \end{pmatrix} \begin{pmatrix} \overbrace{X^* T_m X}^{=S_m} \\ t_{m+1,m} e_m^T X \end{pmatrix}. \tag{5.95}$$

Now, the truncation process goes as described for the linear case, and illustrated in Figure 5.10. Let S_k be partitioned as $\begin{pmatrix} S_{11} & S_{12} \\ \mathbf{0} & S_{22} \end{pmatrix}$, and without loss of generality suppose that the eigenvalues of $S_{11} \in \mathbb{C}^{k\times k}$ approximate the wanted eigenvalues, and the eigenvalues of $S_{22} \in \mathbb{C}^{(m-k)\times(m-k)}$ approximate the unwanted eigenvalues. Partition $X = \begin{pmatrix} X_1 & X_2 \end{pmatrix}$. Then the truncated decompo-

sition (5.95) of order k is

$$H\begin{pmatrix} Q_m U_{m,1} X_1 \\ Q_m U_{m,2} X_1 \end{pmatrix} = \begin{pmatrix} Q_{m+1}\begin{pmatrix} U_{m,1}X_1 & u_{m+1,1} \\ 0 & \beta_{m+1} \end{pmatrix} \\ Q_{m+1}\begin{pmatrix} U_{m,2}X_1 & u_{m+1,2} \\ 0 & 0 \end{pmatrix} \end{pmatrix} \begin{pmatrix} S_{11} \\ t_{m+1,m} e_m^T X_1 \end{pmatrix}. \tag{5.96}$$

However, as in the case of implicitly restarted TOAR, notice that $Q_{m+1} \in \mathbb{C}^{n \times \eta_{m+1}}$ is not truncated. We solve this as in the case of TOAR, i.e., compute the compact SVD factorization of

$$\left(\begin{array}{cc|cc} U_{m,1}X_1 & u_{m+1,1} & U_{m,2}X_1 & u_{m+1,2} \\ 0 & \beta_{m+1} & 0 & 0 \end{array}\right) = P\Sigma G^*.$$

Partition $G = \begin{pmatrix} G_1 & G_2 \end{pmatrix} \in \mathbb{C}^{\eta_{k+1} \times ((k+1)+(k+1))}$ and define $Q_{k+1} = Q_{m+1}P$, $U_{k+1,1} = \Sigma G_1$ and $U_{k+1,2} = \Sigma G_2$ to obtain fully truncated decomposition of order k.

Although this procedure is more elegant and simpler in comparison to the implicitly restarted TOAR, the problem of the shifts remains, i.e., the only shifts one can use in the restarts are the eigenvalues of the Hessenberg matrix T_m. We already saw the examples in which the implicit procedure fails to find good enough approximations when the exact shifts are used. This is why we extend the idea of arbitrary shifts in Krylov–Schur algorithm derived by Bujanović and Drmač [11] for the quadratic eigenvalue problem.

5.8.1 Using arbitrary shifts in the Krylov–Schur algorithm for the quadratic eigenvalue problem

Here, we extend the 4R procedure form the Subsection 5.7.1 to the case of the quadratic eigenvalue problem.

Let

$$H\begin{pmatrix} Q_m U_{m,1} \\ Q_m U_{m,2} \end{pmatrix} = \begin{pmatrix} Q_{m+1}\begin{pmatrix} U_{m,1} & u_{m+1,1} \\ 0 & \beta_{m+1} \end{pmatrix} \\ Q_{m+1}\begin{pmatrix} U_{m,2} & u_{m+1,2} \\ 0 & 0 \end{pmatrix} \end{pmatrix} \begin{pmatrix} T_m \\ t_{m+1,m} e_m^T \end{pmatrix} \tag{5.97}$$

be the compact TOAR decomposition of order m. Let $\mu_1, \ldots, \mu_{m-k}$ be the shifts for the implicit restart. Our procedure has and additional step, and it goes as follows

1. **Reassign**

   Apply an eigenvalue assignment algorithm to compute the vector f so that the spectrum

of $T_m + f e_m^T t_{m+1,m}$ contains the shifts $\mu_1, \ldots, \mu_{m-k}$. The translated factorization (5.97) is

$$H \begin{pmatrix} Q_m U_{m,1} \\ Q_m U_{m,2} \end{pmatrix} = \begin{pmatrix} Q_{m+1} \begin{pmatrix} U_{m,1} & u_{m+1,1} - U_{m,1} f \\ 0 & \beta_{m+1} \end{pmatrix} \\ Q_{m+1} \begin{pmatrix} U_{m,2} & u_{m+1,2} - U_{m,2} f \\ 0 & 0 \end{pmatrix} \end{pmatrix} \begin{pmatrix} T_m - f e_m^T t_{m+1,m} \\ t_{m+1,m} e_m^T \end{pmatrix}. \tag{5.98}$$

THe Eigenvalue assignment is possible if and only if the pair $(T_m^*, e_m^T t_{m+1,m})$ is controllable, i.e. if (5.87) holds.

As in Subsection 5.7.1, f is determined in two steps:

- compute the Controller–Hessenberg form of $(T_m^*, e_m^T t_{m+1,m})$, i.e. compute unitary W so that $W^* T_m^* W = \widetilde{T}_m$ is upper Hessenberg, and $W^* e^T t_{m+1,m} = e_1 \widetilde{t}_{m+1,m}$.
- Compute g such that $\overline{\mu}_1, \ldots, \overline{\mu}_{m-k}$ are the eigenvalues of $\widetilde{T}_m + e_1 g^*$. The wanted vector f is $f = \frac{1}{\widetilde{t}_{m+1,m}} g^* W^*$.

2. **Reorder**
Compute the ordered Schur form of $T_m + f e_m^T t_{m+1,m}$ so that the shifts $\mu_1, \ldots, \mu_{m-k}$ appear as the eigenvalues of the $(m-k) \times (m-k)$ block S_{22}

$$T_m + f e_m^T t_{m+1,m} = \begin{pmatrix} X_1 & X_2 \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ \mathbf{0} & S_{22} \end{pmatrix} \begin{pmatrix} X_1 & X_2 \end{pmatrix}^*. \tag{5.99}$$

3. **Restart**
Multiply the decomposition (5.98) with X_1 to obtain the decomposition of order k

$$H \begin{pmatrix} Q_m U_{m,1} X_1 \\ Q_m U_{m,2} X_1 \end{pmatrix} = \begin{pmatrix} Q_{m+1} \begin{pmatrix} U_{m,1} X_1 & u_{m+1,1} - U_{m,1} f \\ 0 & \beta_{m+1} \end{pmatrix} \\ Q_{m+1} \begin{pmatrix} U_{m,2} X_1 & u_{m+1,2} - U_{m,2} f \\ 0 & 0 \end{pmatrix} \end{pmatrix} \begin{pmatrix} S_{1,1} \\ t_{m+1,m} e_m^T X_1 \end{pmatrix}. \tag{5.100}$$

Denote $\widehat{U}_{k,1} = U_{m,1} X_1, \widehat{U}_{k,2} = U_{m,2} X_1$ and $\widehat{u}_{k+1,1} = u_{m+1,1} - U_{m,1} f, \widehat{u}_{k+1,2} = u_{m+1,2} - U_{m,2} f$.

4. **Restore**
Another translation is used to restore the orthogonality of the matrix $\begin{pmatrix} \widehat{U}_{k,1} & \widehat{u}_{k+1,1} \\ 0 & \beta_{m+1} \\ \widehat{U}_{k,2} & \widehat{u}_{k+1,2} \\ 0 & 0 \end{pmatrix}$. Let

$$g = \widehat{U}_{k,1}^* \widehat{u}_{k+1,1} + \widehat{U}_{k,2}^* \widehat{u}_{k+1,2},$$

and

$$\widetilde{u}_{k+1,1} = \widehat{u}_{k+1,1} - \widehat{U}_{k,1} g, \quad \widetilde{u}_{k+1,2} = \widehat{u}_{k+1,2} - \widehat{U}_{k,2} g.$$

Compute the norm $\gamma = \sqrt{\beta_{m+1}^2 + \|\widetilde{u}_{k+1,1}\|_2^2 + \|\widetilde{u}_{k+1,2}\|_2^2}$ to get

$$\widetilde{u}_{k+1,1} = \frac{\widetilde{u}_{k+1,1}}{\gamma}, \quad \widetilde{u}_{k+1,2} = \frac{\widetilde{u}_{k+1,2}}{\gamma}, \quad \widetilde{\beta}_{k+1} = \frac{\beta_{m+1}}{\gamma}.$$

5. **Reduce**

   To get the full restarted decomposition of order k, we must truncate the orthogonal matrix Q_{m+1} as well. In the first step compute the SVD decomposition

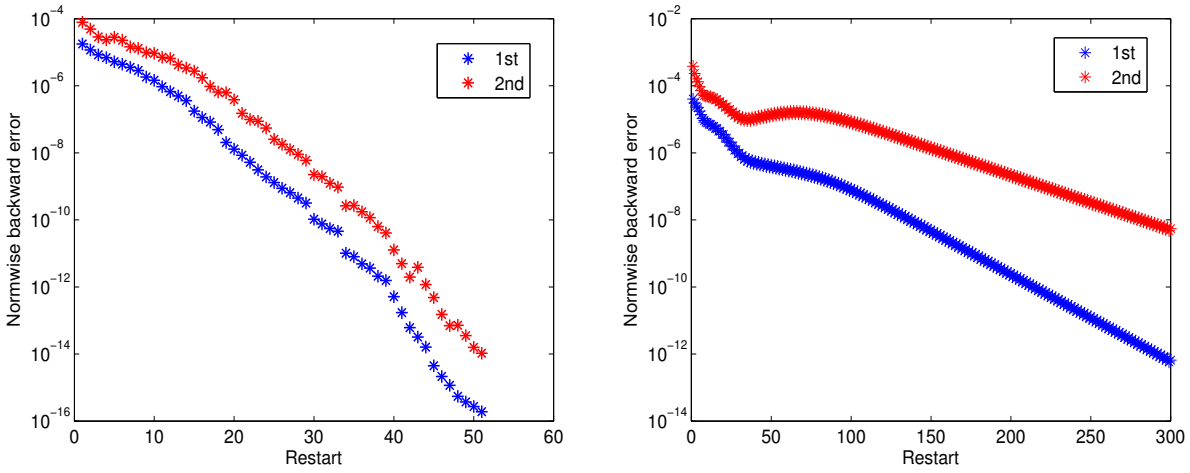
$$\left(\begin{array}{cc|cc} \widehat{U}_{k,1} & \widetilde{u}_{k+1,1} & \widehat{U}_{k,2} & \widetilde{u}_{k+1,2} \\ 0 & \widetilde{\beta}_{m+1} & 0 & 0 \end{array}\right) = P\Sigma G^*. \tag{5.101}$$

   Let η_{k+1} be the rank of the above matrix. Partition $G = \begin{pmatrix} G_1 & G_2 \end{pmatrix}$ so that $G_1, G_2 \in \mathbb{C}^{\eta_{k+1}\times(k+1)}$. The new decomposition is determined with $Q_{k+1} = Q_{m+1}P$, $U_{k+1,1} = \Sigma G_1$, $U_{k+1,1} = \Sigma G_2$.

Numerical example. Recall the quadratic eigenvalue problem $Q(\lambda) = \lambda^2 M + \lambda C + K$ with matrix coefficients

$$M = 0.1\mathbb{I}, \quad C = \mathbb{I}, \quad K = \text{tridiag}(-0.1, 0.2, -0.1). \tag{5.102}$$

from Subsection 5.4.5. We compute the $k = 2$ eigenvalues with the largest magnitude with the same parameters as in Experiment 1 of the same Subsection.



(a) New shifts **(b)** Exact shifts

Figure 5.11: Normwise backward errors for the eigenpair during the restarts

The implicitly restarted Krylov–Schur algorithm with arbitrary shifts described in Subsection 5.4.4 found the wanted eigenvalues in 51 restart. We implemented the Krylov–Schur algorithm with the exact shifts, i.e. the eigenvalues of the matrix B_m in (5.97). The eigenpairs with

the wanted normwise backward error were not found in the first 300 restarts. The normwise backward errors of the eigenvalues during the restarts are presented in Figures 5.11a and 5.11b.

Conclusion

This thesis offers new algorithms for the complete solution of the quadratic and the quartic eigenvalue problems, as well as several improvements of the implicitly restarted Arnoldi like methods for the partial solution of the quadratic eigenvalue problems. Although the basis of these methods is the solution of the equivalent linear problem, considering the particularities of the original nonlinear problem is essential for the computation of the good final approximation.

The contributions of the thesis are:

A new procedure for detecting and deflating of the zero and infinite eigenvalues of the quadratic eigenvalue problem $(\lambda^2 M + \lambda C + K)x = \mathbf{0}$, before calling the QZ algorithm for the linearized problem. It is known that the current methods, despite the prior deflation, cannot remove all zero and infinite eigenvalues; the problem is that, if there exist more Jordan blocks for these eigenvalues, current methods, such as the `quadeig`, deflate only one of them, and, in the subsequent steps, the QZ algorithm may not detect the additional zero or infinite eigenvalues. We developed a test for determining the existence of the Jordan blocks in the terms of the original quadratic problem. In addition we propose the new deflation algorithm, based on the Van Dooren's algorithm for the Kronecker canonical form of linear pencils. Moreover, we analyze different rank revealing strategies, as well as rank determination criteria, and show how they impact the output. Finally, we provide numerical experiments to illustrate the advantages of the new developed method.

An algorithm for the complete solution of the quartic eigenvalue problem $(\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E)x = \mathbf{0}$ is proposed. It follows the ideas and the guiding principles from the development of the quadratic solver. In fact, instead of the direct linearization, it uses an algebraic trick called quadratification to define an equivalent quadratic eigenvalue problem. However, the original coefficient matrices of the problem are used for the definition of scaling and development of the full deflation process of the zero and infinite eigenvalues. Numerical experiments prove that our algorithm is much better than direct application of the state of the art methods `quadeig` and `polyeig` (MATLAB).

The methods for the partial solution of the quadratic eigenvalue problems are also analyzed. New contributions to the implicit restarting of the two level orthogonal Arnoldi algorithm are developed and tested to demonstate their effectiveness in practical computations. In particular, the important class of the overdamped problems is considered in more details and a new strategy, based on tropical roots, is shown to deliver superior performance. Moreover the new starting

vectors for these methods are proposed as well. Finally, the thesis show a direction in which one can develop an efficient Krylov-Schur based method for the quadratic eigenvalue problem; for start it is shown how to enable using arbitrary shifts in a restarting procedure.

Bibliography

[1] Adnan Akay. "Acoustics of friction". In: *The Journal of the Acoustical Society of America* 111.4 (2002), pp. 1525–1548.

[2] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK Users' guide*. SIAM, 1999.

[3] Zhaojun Bai and Yangfeng Su. "SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem". In: *SIAM Journal on Matrix Analysis and Applications* 26.3 (2005), pp. 640–659.

[4] Timo Betcke. "Optimal scaling of generalized and polynomial eigenvalue problems". In: *SIAM Journal on Matrix Analysis and Applications* 30.4 (2008), pp. 1320–1338.

[5] Timo Betcke, Nicholas J. Higham, Volker Mehrmann, Christian Schröder, and Françoise Tisseur. "NLEVP: A collection of nonlinear eigenvalue problems". In: *ACM Transactions on Mathematical Software (TOMS)* 39.2 (2013), p. 7.

[6] Timo Betcke and Daniel Kressner. "Perturbation, extraction and refinement of invariant pairs for matrix polynomials". In: *Linear Algebra and its Applications* 435.3 (2011), pp. 514–536.

[7] David Bindel and Amanda Hood. "Localization theorems for nonlinear eigenvalue problems". In: *SIAM Journal on Matrix Analysis and Applications* 34.4 (2013), pp. 1728–1749.

[8] Ake Bjorck. *Numerical methods for least squares problems*. Vol. 51. Siam, 1996.

[9] Nela Bosner. "Balancing three matrices in control theory". In: *Mathematical communications* 19.3 (2014), pp. 497–516.

[10] T. J. Bridges and P. J. Morris. "Differential eigenvalue problems in which the parameter appears nonlinearly". In: *Journal of Computational Physics* 55.3 (1984), pp. 437–460.

[11] Zvonimir Bujanović and Zlatko Drmač. "A new framework for implicit restarting of the Krylov–Schur algorithm". In: *Numerical Linear Algebra with Applications* 22.2 (2015), pp. 220–232.

[12] Peter Businger and Gene H. Golub. “Linear least squares solutions by Householder transformations”. In: *Numerische Mathematik* 7.3 (1965), pp. 269–276.

[13] Ralph Byers, Chunyang He, and Volker Mehrmann. “Where is the nearest non-regular pencil?” In: *Linear algebra and its applications* 285.1-3 (1998), pp. 81–105.

[14] Carmen Campos and Jose E. Roman. “Parallel Krylov solvers for the polynomial eigenvalue problem in SLEPc”. In: *SIAM Journal on Scientific Computing* 38.5 (2016), S385–S411.

[15] Anthony J. Cox and Nicholas J. Higham. “Stability of Householder QR Factorization for Weighted Least Squares Problems”. In: *Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference*. Ed. by D. F. Griffiths, D. J. Higham, and G. A. Watson. Vol. 380. Pitman Research Notes in Mathematics. , 1998, pp. 57–73.

[16] Biswa Datta. *Numerical methods for linear control systems*. Vol. 1. Academic Press, 2004.

[17] Fernando De Terán, Froilán M Dopico, and D Steven Mackey. “Spectral equivalence of matrix polynomials and the index sum theorem”. In: *Linear Algebra and its Applications* 459 (2014), pp. 264–333.

[18] James Demmel. “Accurate singular value decompositions of structured matrices”. In: *SIAM Journal on Matrix Analysis and Applications* 21.2 (1999), pp. 562–580.

[19] James Demmel, Ming Gu, Stanley Eisenstat, Ivan Slapničar, Krešimir Veselić, and Zlatko Drmač. “Computing the singular value decomposition with high relative accuracy”. In: *Linear Algebra and its Applications* 299 (1999), pp. 21–80.

[20] James Demmel and William Kahan. “Accurate singular values of bidiagonal matrices”. In: *SIAM Journal on Scientific and Statistical Computing* 11.5 (1990), pp. 873–912.

[21] Paul Van Dooren. “The computation of Kronecker’s canonical form of a singular pencil”. In: *Linear Algebra and its Applications* 27 (1979), pp. 103–140.

[22] Zlatko Drmač. “On principal angles between subspaces of Euclidean space”. In: *SIAM Journal on Matrix Analysis and Applications* 22.1 (2000), pp. 173–194.

[23] Zlatko Drmač and Zvonimir Bujanović. “On the failure of rank revealing QR factorization software – a case study”. In: *ACM Trans. Math. Softw.* 35.2 (2008), pp. 1–28.

[24] Zlatko Drmač, Luka Grubišić, Josef Haslinger, Gunter Offner, and Danijel Pavloviıc. *Solution method for the quadratic eigenvalue problems in the structural dynamics of internal combustion engines*. Tech. rep. University of Zagreb, Faculty of Science, Department of Mathematics, 2014.

[25] Zlatko Drmač and Krešimir Veselić. “New fast and accurate Jacobi SVD algorithm: I.” In: *SIAM Journal on matrix analysis and applications* 29.4 (2008), pp. 1322–1342.

[26] Zlatko Drmač and Krešimir Veselić. “New fast and accurate Jacobi SVD algorithm: II.” In: *SIAM Journal on matrix analysis and applications* 29.4 (2008), pp. 1343–1362.

[27] Iain S. Duff, Roger G. Grimes, and John G. Lewis. “Users’ guide for the Harwell-Boeing sparse matrix collection (Release I)”. In: (1992).

[28] R. L. Duffin and R. J. Duffin. “A minimax theory for overdamped networks”. In: *Journal of Rational Mechanics and Analysis* 4 (1955), pp. 221–233.

[29] Carl Eckart and Gale Young. “The approximation of one matrix by another of lower rank”. In: *Psychometrika* 1.3 (1936), pp. 211–218.

[30] Hung-Yuan Fan, Wen-Wei Lin, and Paul Van Dooren. “Normwise scaling of second order polynomial matrices”. In: *SIAM journal on matrix analysis and applications* 26.1 (2004), pp. 252–256.

[31] Stéphane Gaubert and Meisam Sharify. “Tropical scaling of polynomial matrices”. In: *Positive systems*. Springer, 2009, pp. 291–303.

[32] Israel Gohberg, Peter Lancaster, and Leiba Rodman. *Matrix polynomials*. Springer, 2005.

[33] Gene Golub, Virginia Klema, and Gilbert W Stewart. *Rank degeneracy and least squares problems*. Tech. rep. Computer Science Department, Stanford University, 1976.

[34] Nils Gräbner, Volker Mehrmann, Sarosh Quraishi, Christian Schröder, and Utz von Wagner. “Numerical methods for parametric model reduction in the simulation of disk brake squeal”. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 96.12 (2016), pp. 1388–1405.

[35] Ming Gu, James Demmel, and Inderjit Dhillon. “Efficient computation of the singular value decomposition with applications to least squares problems”. In: (1994).

[36] Ming Gu and Stanley C Eisenstat. “Efficient algorithms for computing a strong rank-revealing QR factorization”. In: *SIAM Journal on Scientific Computing* 17.4 (1996), pp. 848–869.

[37] Sven Hammarling, Christopher J. Munro, and Françoise Tisseur. “An algorithm for the complete solution of quadratic eigenvalue problems”. In: *ACM Transactions on Mathematical Software (TOMS)* 39.3 (2013), p. 18.

[38] Vicente Hernández, Jose E Román, Andrés Tomás, and Vicente Vidal. “Krylov-schur methods in SLEPc”. In: *Universitat Politecnica de Valencia, Tech. Rep. STR-7* (2007).

[39] Desmond J. Higham and Nicholas J. Higham. “Structured backward error and condition of generalized eigenvalue problems”. In: *SIAM Journal on Matrix Analysis and Applications* 20.2 (1998), pp. 493–512.

[40] Nicholas J. Higham. “A survey of condition number estimation for triangular matrices”. In: *Siam Review* 29.4 (1987), pp. 575–596.

[41] Nicholas J. Higham. *Accuracy and stability of numerical algorithms*. Vol. 80. Siam, 2002.

[42] Nicholas J. Higham, Ren-Cang Li, and Françoise Tisseur. "Backward error of polynomial eigenproblems solved by linearization". In: *SIAM journal on matrix analysis and applications* 29.4 (2007), pp. 1218–1241.

[43] Nicholas J. Higham, D Steven Mackey, and Françoise Tisseur. "The conditioning of linearizations of matrix polynomials". In: *SIAM Journal on Matrix Analysis and Applications* 28.4 (2006), pp. 1005–1028.

[44] Urmi B. Holz, Gene H. Golub, and Kincho H. Law. "A subspace approximation method for the quadratic eigenvalue problem". In: *SIAM journal on matrix analysis and applications* 26.2 (2004), pp. 498–521.

[45] Zhongxiao Jia and Yuquan Sun. "A refined second-order Arnoldi (RSOAR) method for the quadratic eigenvalue problem and implicitly restarted algorithms". In: *arXiv preprint arXiv:1005.3947* (2010).

[46] Peter Lancaster and Ion Zaballa. "Diagonalizable quadratic eigenvalue problems". In: *Mechanical Systems and Signal Processing* 23.4 (2009), pp. 1134–1144.

[47] Richard B. Lehoucq and Danny C. Sorensen. "Deflation techniques for an implicitly restarted Arnoldi iteration". In: *SIAM Journal on Matrix Analysis and Applications* 17.4 (1996), pp. 789–821.

[48] Richard B. Lehoucq, Danny C. Sorensen, and Chao Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. Vol. 6. Siam, 1998.

[49] Ding Lu, Yangfeng Su, and Zhaojun Bai. "Stability analysis of the two-level orthogonal Arnoldi procedure". In: *SIAM Journal on Matrix Analysis and Applications* 37.1 (2016), pp. 195–214.

[50] D. Steven Mackey, Niloufer Mackey, Christian Mehl, and Volker Mehrmann. "Vector spaces of linearizations for matrix polynomials". In: *SIAM Journal on Matrix Analysis and Applications* 28.4 (2006), pp. 971–1004.

[51] Dinesh Manocha. "Solving systems of polynomial equations". In: *IEEE Computer Graphics and Applications* 14 (1994), pp. 46–55.

[52] Leon Mirsky. "Symmetric gauge functions and unitarily invariant norms". In: *The quarterly journal of mathematics* 11.1 (1960), pp. 50–59.

[53] A. P. Morgan. "Polynomial continuation and its relationship to the symbolic reduction of polynomial systems". In: *Symbolic and Numerical Computation for Artificial Intelligence* (1992), pp. 23–45.

[54] Vanni Noferini, Meisam Sharify, and Françoise Tisseur. “Tropical roots as approximations to eigenvalues of matrix polynomials”. In: *SIAM Journal on Matrix Analysis and Applications* 36.1 (2015), pp. 138–157.

[55] Steven A. Orszag. “Accurate solution of the Orr–Sommerfeld stability equation”. In: *Journal of Fluid Mechanics* 50.4 (1971), pp. 689–703.

[56] C-T Pan. “On the existence and computation of rank-revealing LU factorizations”. In: *Linear Algebra and its Applications* 316.1-3 (2000), pp. 199–222.

[57] Beresford N Parlett. *The symmetric eigenvalue problem*. Vol. 20. siam, 1998.

[58] M. J. D. Powell and J. K. Reid. “On applying Householder transformations to linear least squares problems”. In: *Information Processing 68, Proc. International Federation of Information Processing Congress, Edinburgh, 1968*. 1969, pp. 122–126.

[59] Youcef Saad. “Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems”. In: *Mathematics of Computation* 42.166 (1984), pp. 567–588.

[60] Youcef Saad. *Numerical methods for large eigenvalue problems*. Manchester University Press, 1992.

[61] Meisam Sharify. “Scaling Algorithms and Tropical Methods in Numerical Matrix Analysis: Application to the Optimal Assignment Problem and to the Accurate Computation of Eigenvalues”. PhD thesis. Ecole Polytechnique X, 2011. URL: https://pastel.archives-ouvertes.fr/pastel-00643836/document.

[62] Diana M. Sima, Sabine Van Huffel, and Gene H. Golub. “Regularized total least squares based on quadratic eigenvalue problem solvers”. In: *BIT Numerical Mathematics* 44.4 (2004), pp. 793–812.

[63] Danny C. Sorensen. “Implicit application of polynomial filters in a k-step Arnoldi method”. In: *Siam journal on matrix analysis and applications* 13.1 (1992), pp. 357–385.

[64] Gilbert W. Stewart. “A Krylov–Schur algorithm for large eigenproblems”. In: *SIAM Journal on Matrix Analysis and Applications* 23.3 (2002), pp. 601–614.

[65] Yangfeng Su, Junyi Zhang, and Zhaojun Bai. *A compact Arnoldi algorithm for polynomial eigenvalue problems*. 2008. URL: http://math.cts.nthu.edu.tw/Mathematics/RANMEP%20Slides/Yangfeng%20Su.pdf.

[66] Françoise Tisseur. “Backward error and condition of polynomial eigenvalue problems”. In: *Linear Algebra and its Applications* 309.1-3 (2000), pp. 339–361.

[67] Françoise Tisseur and Karl Meerbergen. “The quadratic eigenvalue problem”. In: *SIAM review* 43.2 (2001), pp. 235–286.

[68] Lloyd N. Trefethen. *Spectral methods in MATLAB*. Vol. 10. Siam, 2000.

[69] Lloyd N. Trefethen and Mark Embree. *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, 2005.

[70] Robert C. Ward. "Balancing the generalized eigenvalue problem". In: *SIAM Journal on Scientific and Statistical Computing* 2.2 (1981), pp. 141–152.

[71] David S Watkins. "A case where balancing is harmful". In: *Electron. Trans. Numer. Anal* 23 (2006), pp. 1–4.

[72] David S. Watkins. "Performance of the QZ algorithm in the presence of infinite eigenvalues". In: *SIAM Journal on Matrix Analysis and Applications* 22.2 (2000), pp. 364–375.

[73] Stephen J. Wright. "A collection of problems for which Gaussian elimination with partial pivoting is unstable". In: *SIAM Journal on Scientific Computing* 14.1 (1993), pp. 231–238.

[74] H. Zha. "Singular values of a classical matrix". In: *American Mathematical Monthly* 104 (1997), pp. 172–173.

List of Figures

1.1 Diagram for defining the Jordan pair 11
1.2 Poiseuille flow 25

2.1 Commutative diagram for a backward perturbation in the computation of a right eigenpair (x,λ) of the matrix polynomial $P(\lambda)=\sum_{i=0}^{k}\lambda^i A_i$. 42
2.2 Backward errors for the eigenvalue problem of the linearization (2.28) of the test example `power_plant`, and for the original problem $(\lambda^2 M+\lambda C+K)x=\mathbf{0}$. 49
2.3 Newton polygon corresponding to $\mathtt{tp}(\mathtt{x})$ 52
2.4 `speaker_box`, normwise and componentwise backward errors for all right eigenpairs 56

3.1 Comparison of the absolute values of the diagonal entries of R from (3.3) and the singular values of A. Note that the QR factorization correctly detects that A is $O(10^{-7})$ close to a matrix of rank 100. 63
3.2 Backward errors for the linearization C_2, the original problem quadratic problem and the scaled pencil $\widetilde{Q}(\lambda)$, for the test problem `power_plant`. 78
3.3 Intersection points of a sphere, a cylinder and a plane (`intersection` in NLEVP) 93
3.4 Sparsity structure of the linearization pencil before and after deflation 95
3.5 Deflation process in `KVADeig` – decision tree 105
3.6 Backward errors for finite eigenvalues, sorted by magnitude, for the benchmark problem `intersection`. 106
3.7 Backward errors for the finite eigenvalues sorted by magnitude for the `intersection` problem 116
3.8 Comparison of the backward errors for the finite eigenvalues, sorted by magnitude, for the `mobile_manipulator` problem 116
3.9 Comparison of the normwise and componentwise backward errors for the right eigenpair for the problem `cd_player` 118
3.10 Sparsity structure of the matrix K and the corresponding components in the rank revealing factorizations 118
3.11 Comparison of the normwise and backward errors for the right eigenpair for the reversed `cd_player` problem 119

3.12 Singular values of the coefficient matrices M and K in the scaled `dirac` example 120
3.13 Comparison of the componentwise backward error, normwise backward errors, and the spectrum for the scaled `dirac` problem 120
3.14 Singular values of leading coefficient matrix in scaled reversed `dirac` example 121
3.15 Comparison of the componentwise backward error, normwise backward errors, and the spectrum for the scaled reversed `dirac` problem 122
3.16 Comparison of the componentwise backward errors, and normwise backward errors for the `deriv2` problem 123
3.17 Singular values of the leading matrix coefficient in `deriv2` example 123

4.1 Decision tree for the deflation process in `KVARTeig` 140
4.2 Norm-wise backward error for finite nonzero eigenvalues, `mirror` 141
4.3 Singular values of leading matrix coefficient A, `orr_sommerfeld` 142
4.4 The ratio $\sigma_1(A)/\sigma_i(A)$ u–machine precision 142
4.5 Comparison of the normwise and componentwise backward errors for the finite right eigenpairs for `orr_sommeferld` example of order $n = 1000$ 143
4.6 Computed finite eigenvalues for `orr_sommerfeld` example of order $n = 1000$. 144

5.1 Polynomial filter in the first restart of IRA iterations 149
5.2 All eigenvalues of QEP (5.102) 165
5.3 Backward errors for first 23 iterations of `eigs` 166
5.4 Backward errors for shift and invert with tropical root 166
5.5 Normwise backward errors in every restart for all computed eigenvalues 167
5.6 Final normwise backward errors, Path crossing 172
5.7 Normwise backward errors in every restart for all computed eigenvalues 173
5.8 Final normwise backward errors, `cd_player` 173
5.9 Normwise backward errors in every restart for all computed eigenvalues, `cd_player` 174
5.10 Truncation step in Krylov–Schur algorithm 176
5.11 Normwise backward errors for the eigenpair during the restarts 181

List of Tables

3.1 Comparison of component-wise backward errors 92
3.2 Comparison of range of elements in M, C, K 92
3.3 Number of deflated eigenvalues 108
3.4 Rank revealing factorization error, scaled reversed `dirac` 121

4.1 Comparison of backward errors for `polyeig`, `quadeig` and `KVARTeig` 141

5.1 Number of restarts, Path crossing 172
5.2 Number of restarts, `cd_player` 173

Curriculum Vitae

Ivana Šain Glibić was born 14th of February 1991 in Mostar, Bosnia and Herzegovina. She finished primary school and gymnasium in Mostar. In 2009. she started undergraduate studies in Mathematics, and in 2012. she started graduate studies in Applied Mathematics, both at the University of Zagreb, Faculty of Science, Department of Mathematics. In 2014., she graduated with diploma thesis *Arnoldi algorithm for nonlinear eigenvalue problems* under the supervision of prof.dr.sc. Zlatko Drmač. At the same year she started the Doctoral Study Programme in Mathematics at University of Zagreb, Faculty of Science, Department of Mathematics.

In 2014. she received the Honours for the outstanding success in the studies by the Department of Mathematics.

From 2015. to 2017. she has been employed as an expert associate, and from 2017. as assistant on the Croatian Science Foundation Project Grant number 9345 *Mathematical modelling, analysis and computing with applications to complex mechanical systems*, at the Numerical mathematics and scientific computing division at the Department of Mathematics, Faculty of Science, University of Zagreb.

She was a teaching assistant for three courses at the Department of Mathematics: Numerical Mathematics, Computer Lab 1 and Computer Lab 2.

She attended three summer schools. She presented the work from this thesis by giving five talks at the different mathematical meetings.