Algorithms for Matrix Polynomials and Structured Matrix Problems

Munro, Christopher J.

2011

MIMS EPrint: **2011.21**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

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

ISSN 1749-9097

ALGORITHMS FOR MATRIX POLYNOMIALS AND STRUCTURED MATRIX PROBLEMS

A non-confidential version of a restricted thesis
submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2010

Christopher J. Munro
School of Mathematics

Contents

Abstract **8**

Declaration **9**

Copyright Statement **10**

Publications **12**

Advisor and Examiners **13**

Acknowledgements **14**

1 Introduction **15**

1.1 Outline and Motivation 15

1.2 Notation and Background Linear Algebra 17

1.3 Matrix Factorizations 19

1.4 Algorithms Implementation in Finite Precision 21

1.4.1 Measuring Accuracy and Stability of Computed Solutions . . 21

1.4.2 Matrix Rank Computation 23

1.5 The Polynomial Eigenvalue Problem 25

1.5.1 Structures and Properties of Matrix Polynomials 25

1.5.2 Singular Leading and Trailing Coefficients 26

1.5.3 Measuring the Accuracy of Computed Eigensolutions 27

1.5.4 Applications 29

2 Solving PEPs by Linearization **35**

2.1 Linearizations of Matrix Polynomials 35

2.2 Solving PEPs by Linearization in Finite Precision Arithmetic 37

2.3 Accuracy and Conditioning of Solutions to QEPs Solved by Linearization 38

2.3.1 Scaling Higher Degree Matrix Polynomials 44

2.3.2 Techniques to Improve Accuracy of Eigenvalues of Specific Magnitude 44

2.4 The QZ Algorithm 45

2.5 Eigenvectors of Matrix Polynomials from Linearizations 47

3 Algorithm for Quadratic Eigenproblems **49**

3.1 Introduction 49

3.2 Choice of Linearization 52

3.2.1 Backward Error and Condition Number 52

3.2.2 Companion Linearizations 54

3.3 Deflation of 0 and ∞ Eigenvalues 58

3.3.1 Rank and Nullspace Determination 59

3.3.2 Block Triangularization of $C_2(\lambda)$ 61

3.4 Left and Right Eigenvectors 64

3.4.1 Right Eigenvectors 64

3.4.2 Left Eigenvectors 66

3.5 Algorithm 66

3.6 Numerical Experiments 67

3.7 Conclusion 71

4 Structure Preserving Transformations **72**

4.1 Introduction 72

4.2 Structure Preserving Transformations 77
4.2.1 Elementary SPTs 78
4.3 Derivation of Structure Preserving Constraints 80
4.4 Computing the Vectors Defining a Class Two SPT 84
4.5 Deflation for Symmetric Quadratics 86
4.5.1 Linearly Dependent Eigenvectors 86
4.5.2 Linearly Independent Eigenvectors 89
4.6 Deflation for Nonsymmetric Quadratics 93
4.6.1 Parallel Left Eigenvectors and Parallel Right Eigenvectors . 93
4.6.2 Non Parallel Left Eigenvectors and Non Parallel Right Eigenvectors 95
4.6.3 Non Parallel Left (Right) Eigenvectors and Parallel Right (Left) Eigenvectors 98
4.7 Numerical Experiments 99
4.8 Proof of Lemma 3, Symmetric Quadratics 102
4.9 Proof of Lemma 4, Nonsymmetric Quadratics 105

5 Conclusions **113**

Bibliography **116**

Word count 28292

List of Tables

2.1 Theoretical and computed eigenpairs of first (C_1) and second (C_2) companion linearizations of quadratic Q with $\det(A_0) \neq 0$. (Finite nonzero eigenvalues λ.) 47

3.1 Quadratic eigenvalue problems from NLEVP collection. Largest backward errors of eigenpairs, and corresponding eigenvalue λ, computed by `polyeig` and `quadeig`. **D** indicates that deflation was performed by `quadeig` since the problem has singular leading or trailing coefficients. 69

3.2 Execution time in seconds for eigenvalue computation of quadratics in NLEVP with singular A_0 and/or A_2. 70

3.3 `cd_player` and `speaker_box` problems from NLEVP collection. Largest backward errors of eigenpairs computed by `quadeig` comparing two scaling types, FLV (Fan, Lin and Van Dooren scaling only) and DS, FLV (diagonal scaling, then Fan, Lin and Van Dooren scaling). . . 70

4.1 Relative magnitude of the off-diagonal elements of the deflated quadratic $Q_2(\lambda) = \lambda^2 M_2 + \lambda C_2 + K_2$ experiment 2 and condition number of the transformations. 100

4.2 Condition numbers of the SPTs T and deflating transformations G for different pairs of eigenvalues for experiment 4. 101

4.3 Scaled residuals and condition numbers for transformations in Experiment 4. 102

List of Figures

1.1 Schematic of a shaft on bearing support 31
1.2 Simply supported beam with damping 31
1.3 Spring/dashpot with Maxwell elements 34

2.1 Computed spectrum of unscaled/scaled damped beam quadratic for linearizations C_1, L_1, and L_2, (as defined in (2.1.4)–(2.2.1)) using MATLAB's `eig` to solve the linear problem 39

The University of Manchester

Christopher J. Munro
Doctor of Philosophy
Algorithms for Matrix Polynomials and Structured Matrix Problems
February 19, 2011

In this thesis we focus on algorithms for matrix polynomials and structured matrix problems.

We begin by presenting a general purpose eigensolver for dense quadratic eigenvalue problems, which incorporates recent contributions on the numerical solution of polynomial eigenvalue problems, namely a scaling of the eigenvalue parameter prior to the computation, and a choice of linearization with favourable conditioning and backward stability properties. Our algorithm includes a preprocessing step that reveals the zero and infinite eigenvalues contributed by singular leading and trailing matrix coefficients and deflates them. Numerical experiments are presented, comparing the performance of this algorithm on a collection of test problems, in terms of accuracy and stability.

We then describe structure preserving transformations for quadratic matrix polynomials. Given a pair of distinct eigenvalues (λ_1, λ_2) of an $n \times n$ quadratic matrix polynomial $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ with a nonsingular leading coefficient and their corresponding eigenvectors, we show how to transform $Q(\lambda)$ into a quadratic of the form $\begin{bmatrix} Q_d(\lambda) & 0 \\ 0 & q(\lambda) \end{bmatrix}$ having the same eigenvalues as $Q(\lambda)$, with $Q_d(\lambda)$ an $(n-1) \times (n-1)$ quadratic matrix polynomial and $q(\lambda)$ a scalar quadratic polynomial with roots λ_1 and λ_2.

Declaration

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

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property

and/or Reproductions described in it may take place is available in the University IP Policy (see http://www.campus.manchester.ac.uk/medialibrary/policies/intellectual-property.pdf), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's policy on presentation of Theses.

Publications

This thesis is based on the following publications:

- Chapter 3 is based on the technical report "A General Purpose Algorithm for Solving Quadratic Eigenproblems" [52] (with F. Tisseur and S. Hammarling)
- Chapter 4 is based on the paper "Deflating Quadratic Matrix Polynomials with Structure Preserving Transformations" [57] (with F. Tisseur, and S. Garvey), to appear in Linear Algebra and its Applications

Advisor and Examiners

Principal Advisor: Françoise Tisseur

Internal Examiner: Nick Higham

External Examiner: Karl Meerbergen (Katholieke Universiteit Leuven)

Acknowledgements

I am very grateful to my supervisor Françoise Tisseur for her supervision during the past three years. It has been a privilege to work with Sven Hammarling on the work contained in Chapter 3.

I would also like to thank:

- Younes Chahlaoui, Nick Higham and my colleagues Maha Al-Ammari, Rüdiger Borsdorf, and Lijing Lin from the linear algebra group at Manchester for many helpful comments and suggestions on work and talks during my PhD
- Chris Paul who was extremely helpful in providing quick IT support many times during my PhD
- Athena Makroglou and Graham Elliott (University of Portsmouth), and Liz Mavin and Jim Carr (formerly Fareham College) for their inspiration and encouragement
- The School of Mathematics at Manchester for financial support for the duration of my PhD, and support with costs of travelling to conferences and workshops.
- Jenny Copelton and Neil Zammit for many helpful comments on a draft of this manuscript
- my parents!

Chapter 1

Introduction

1.1 Outline and Motivation

The main theme of this thesis is developing algorithms that preserve structure in matrix problems in finite precision arithmetic. We motivate the importance of structure preservation by the following quote [62]:

> "When a problem has any significant structure, we should design and use algorithms that preserve and exploit that structure. Observation of this principle usually results in algorithms that are superior in speed and accuracy."
> David S. Watkins

A number of benefits can therefore result from developing structure preserving algorithms. Making use of the inherent structure in the problem can lead to more efficient algorithms and a reduction in storage requirements. Preserving structure can also lead to an increase in accuracy, stability, and necessarily the key qualities of the problem are preserved, for example spectral symmetries, location of eigenvalues, and physical properties such as positive definiteness.

A recent example of the importance of structure preserving methods is illustrated by a quadratic eigenvalue problem that results when modelling vibrations on railway tracks [35]. It is shown in [48] that deflation and taking into account the structure of the problem is crucial to obtaining an accurate solution. Indeed, solving the problem directly with the QZ algorithm, even in quadruple precision, returns a solution with no correct significant figures [45].

This thesis focusses on algorithms for matrix polynomials, after introducing background material in Chapter 1, we give an outline of the solution of polynomial eigenvalue problems by linearization. Chapter 3 describes theory and implementation of a general purpose algorithm `quadeig` for solving quadratic eigenvalue problems. This algorithm incorporates recent contributions on the numerical solution of polynomial eigenvalue problems, namely a scaling of the eigenvalue parameter prior to the computation, [6], [14] and a choice of linearization with favourable conditioning and backward stability properties [30], [32], [33]. Our algorithm includes a preprocessing step that reveals the zero and infinite eigenvalues contributed by singular leading and trailing matrix coefficients and deflates them. The algorithm is tested on quadratic eigenproblems from the NLEVP collection of nonlinear eigenproblems [7], illustrating the improved performance of this new algorithm `quadeig`, with the existing MATLAB routine `polyeig`, both in terms of accuracy and stability and reduced computational cost.

Chapter 4 describes a structure preserving technique for the deflation of eigenpairs from quadratic matrix polynomials (a special case of general degree matrix polynomials). Structure preserving transformations (SPTs) and associated constraints needed to determine them are defined in [17], the contribution of this thesis is to use them to construct a family of nontrivial elementary SPTs that have a specific action of practical use: that of "mapping" two linearly independent eigenvectors to a set of linearly dependent eigenvectors. Using this family of SPTs,

given two eigentriples (λ_j, x_j, y_j), $j = 1, 2$ satisfying appropriate conditions, we can decouples $Q(\lambda)$ into a quadratic $Q_d(\lambda) = \lambda^2 M_d + \lambda C_d + K_d$ of dimension $n-1$ and a scalar quadratic $q(\lambda) = \lambda^2 m + \lambda c + k = m(\lambda - \lambda_1)(\lambda - \lambda_2)$ such that (a)

$$\Lambda(Q) = \Lambda(Q_d) \cup \{\lambda_1, \lambda_2\},$$

where $\Lambda(Q)$ denotes the spectrum of Q and (b) there exist well-defined relations between the eigenvectors of $Q(\lambda)$ and those of the decoupled quadratic

$$\widetilde{Q}(\lambda) = \begin{bmatrix} Q_d(\lambda) & 0 \\ 0 & q(\lambda) \end{bmatrix}. \tag{1.1.1}$$

This procedure applies to symmetric and nonsymmetric quadratics, and when the quadratic is symmetric preserves the symmetry.

1.2 Notation and Background Linear Algebra

In this work we generally adopt the Householder convention with regard to naming variables, using the notation below.

- I_n denotes the n-by-n identity matrix.
- Matrices are denoted by capital letters: A.
- Elements of matrices by lower case letters of the respective matrix: a_{ij}.
- Vectors are denoted by lower case Latin letters: a, b, c.
- Scalars are denoted by Greek lower case letters: α, β, γ.

We adopt the MATLAB matrix notation, thus $A(i\colon j, k\colon l)$ represents the intersection of rows i to j and columns k to l, while $A(:, k)$ denotes the kth column,

the colon means to take all elements in the kth column. "T" denotes transpose, while in complex arithmetic "$*$" denotes conjugate transpose. We write the names of routines from LAPACK (linear algebra package [2]) or MATLAB [49] as for example `polyeig`.

- A vector norm is a function $\|\cdot\| : \mathbb{C}^n \to \mathbb{C}$ satisfying the following
  - $\|x\| \geqslant 0$ for all $x \in \mathbb{C}^n$ (with equality if and only if $x = 0$),
  - $\|\alpha x\| = |\alpha|\|x\|$ for all $\alpha \in \mathbb{C}, x \in \mathbb{C}^n$,
  - $\|x + y\| \leqslant \|x\| + \|y\|$ for all $x, y \in \mathbb{C}^n$.

  A matrix norm $\|\cdot\| : \mathbb{C}^{n\times n} \to \mathbb{C}$ satisfies a similar definition. Two examples are the Frobenius norm $\|A\|_F = \sqrt{\text{trace}(A^*A)}$ and the 2-norm (or spectral norm) $\|A\|_2 = \sqrt{\lambda_{\max}(A^*A)}$. Both the Frobenius and 2-norms are consistent norms ($\|AB\| \leqslant \|A\|\|B\|$), and unitarily invariant, that is if $A, Q, Z \in \mathbb{C}^{n\times n}$ with Q, Z unitary ($Z^*Z = Q^*Q = I$), then $\|QAZ\|_F = \|A\|_F$ and $\|QAZ\|_2 = \|A\|_2$.

- The spectrum or set of all eigenvalues of a matrix A is denoted by $\Lambda(A)$.

- We denote the Kronecker product by $\otimes$ and give a definition below.

  Definition 1 (Kronecker Product, see [20]). *Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$ the Kronecker product $A \otimes B \in \mathbb{C}^{mn\times mn}$ of A and B is given by*

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mm}B \end{bmatrix}.$$

- The null space of a matrix null(A) is a set of linearly independent vectors, where each vector $x \neq 0$ satisfies $Ax = 0$.
- The rank of an n-by-n matrix A is the number of linearly independent rows or columns, and we have the relation $\text{rank}(A) = n - \dim(\text{null}(A))$.

1.3 Matrix Factorizations

In this section we define the following matrix factorizations that will be used in the algorithms presented in this thesis:

- Singular value decomposition.
- QR factorization and QR factorization with column pivoting.
- Schur, generalized Schur and generalized real Schur decomposition.

Definition 2 (Singular Value Decomposition [21, Thm. 2.5.2])**.** *Any* $A \in \mathbb{R}^{m\times n}$ *can be decomposed as*

$$A = U\Sigma V^T$$

$U \in \mathbb{R}^{m\times m}$,$V \in \mathbb{R}^{n\times n}$ *are orthogonal,* $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n}$ *contains the singular values of* A*. The singular values are ordered such that* $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r = \cdots \sigma_p = 0$ *where* $\text{rank}(A) = r$ *and* $p = \min(m, n)$*.*

Computing the SVD is one possible method of computing the rank of a matrix.

Definition 3 (Schur Decomposition [21, Thm. 7.1.3])**.** *Given* $A \in \mathbb{C}^{n\times n}$ *then there exists a unitary matrix* $Q \in \mathbb{C}^{n\times n}$ *such that* $Q^*AQ = T$*, where* T *is upper triangular and* $\Lambda(A) = \text{diag}(T)$*,* Q *can be chosen such that the eigenvalues appearing on the diagonal of* T *appear in any order.*

Definition 4 (Generalized Schur Decomposition [21, Thm. 7.7.1]). *Given $A, B \in \mathbb{C}^{n\times n}$ there exist unitary matrices $Q, Z \in \mathbb{C}^{n\times n}$ such that*

$$Q^*(A - \lambda B)Z = T - \lambda S$$

where T and S are upper triangular.

If $t_{jj} = s_{jj} = 0$ for some j then $\lambda(A, B) = \mathbb{C}$ otherwise

$$\lambda(A, B) = \left\{ \frac{t_{ii}}{s_{ii}} \right\}$$

and if $s_{ii} = 0$ for some i the eigenvalue λ_i is said to be infinite.

Given a real matrix pencil, and working only in real arithmetic there is the generalized real Schur form. In this case given $A, B \in \mathbb{R}^{n\times n}$ there exist orthogonal matrices $Q, Z \in \mathbb{R}^{n\times n}$ such that

$$Q^T(A - \lambda B)Z = T - \lambda S$$

where T is quasi-upper triangular and S is upper triangular. In general $T - \lambda S$ will be quasi upper triangular. The eigenvalues of the pencil $A - \lambda B$ comprise the ratios of the diagonal elements of $T - \lambda S$ for real eigenvalues, and the eigenvalues of the blocks appearing on the diagonal of $T - \lambda S$ yield the complex eigenvalues of $A - \lambda B$.

Definition 5 (QR Factorization [21, Sec. 5.2]). *Given a matrix $A \in \mathbb{R}^{m\times n}$ with $m \geqslant n$, then its QR factorization is given by*

$$A = QR,$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times n}$ is upper triangular.

Definition 6 (QR Factorization with Column Pivoting [21, Sec. 5.4.1])**.** *Given $A \in \mathbb{R}^{m\times n}$ with $m \geqslant n$, its QR factorization with column pivoting is given by*

$$Q^T AP = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix},$$

where Q is orthogonal, P a permutation matrix, $R_{11} \in \mathbb{R}^{k\times k}$ is upper triangular, and $k = \operatorname{rank}(A)$. To define P, consider the jth stage of Householder QR factorization, at the start of which we have

$$(Q_1 \cdots Q_{j-1})^T A(P_1 \cdots P_{j-1}) = \begin{bmatrix} R_{11}^{(j-1)} & R_{12}^{(j-1)} \\ 0 & R_{22}^{(j-1)} \end{bmatrix} \tag{1.3.1}$$

with $R_{11}^{(j-1)}$ nonsingular. The next permuation matrix P_j is chosen so that the column of largest norm in $R_{22}^{(j-1)}$ is move to the lead position, then the next Householder transformation Q_j has the action of zeroing the subdiagonal components.

1.4 Algorithms Implementation in Finite Precision

In this thesis we implement algorithms in finite precision, not exact arithmetic. In this section we highlight some of the relevant details.

1.4.1 Measuring Accuracy and Stability of Computed Solutions

When considering solutions to problems in finite precision we are interested in two quantities. Firstly, when we have a problem to solve with initial sampled data,

there is the possibility that the sampled data contains errors. The *conditioning* of the data measures the sensitivity of the solution of the problem to perturbations in the data. The extent to which the problem is well conditioned is an inherent property of the problem. Secondly, given a method or algorithm for computing a solution to a problem we would like to assess the quality of the computed solution. *Backward error* is a measure of how much the problem must be perturbed for the computed solution to be an exact solution of the perturbed problem.

An important quantity involved with working in finite precision is the unit roundoff u, which characterizes the worst-case error inherent in representing real numbers as floating point numbers in finite precision arithmetic.

Theorem 1 ([29]). *If $x \in \mathbb{R}$ lies in the range of a floating point number system F (a subset of the real numbers) then*

$$fl(x) = x(1+\delta), \quad |\delta| < u,$$

where $fl(x)$ denotes x evaluated in floating point arithmetic.

When implementing algorithms in MATLAB, the inbuilt function `eps` (machine precision) can be used as a tolerance. This is not the same as the unit roundoff but characterizes spacing of floating point numbers, thus `eps` returns the distance from 1.0 to the next largest floating point number. The unit roundoff in MATLAB is $u = 2^{-53} = \texttt{eps}/2 \approx$ 1.1e-16.

When developing algorithms to work in finite precision we would ideally like to work with orthogonal transformations (U a real square matrix such that $U^TU = I$, for U complex ^{T} is replaced by conjugate transpose *). If we carry out a transformation on a matrix with errors: $\widehat{A} = A + E$ to form $U^T(A+E)U$ and take the norm, then for orthogonal/unitary matrices and a unitarily invariant matrix norm $\|\cdot\|$, $\|U^TEU\| = \|E\|$ so we do not increase error inherent in the data.

1.4.2 Matrix Rank Computation

A key stage in many of the algorithms in this thesis is computing accurately (or inferring information about) the rank of a matrix in finite precision arithmetic. Given a matrix $A \in \mathbb{R}^{n\times n}$ whose rank we wish to compute, we can take the SVD, an eigendecomposition, or compute a QR factorization with column pivoting.

Theoretically the SVD yields a factorization $A = U\Sigma V^T$ with

$$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r, \sigma_{r+1}, \ldots \sigma_n)$$

where, if the matrix is singular we have $\sigma_{r+1} \ldots \sigma_n$ equal to zero exactly. In finite precision, however, we will have the computed SVD $\widehat{U}\widehat{\Sigma}\widehat{V}^T$ where $\widehat{x}$ denotes the computed value of x. Thus $\hat{\sigma}_{r+1} \ldots \hat{\sigma}_n$ will not be exactly zero, rather some 'small' quantity. We will then have to take a rank decision and neglect (set to zero) any singular values less than a particular tolerance τ which will need to be chosen, we then call the resulting rank the *numerical rank*.

Definition 7 (Numerical Rank)**.** *Given a matrix $A \in \mathbb{C}^{n\times n}$ and a tolerance $\tau > 0$ then the numerical rank of A is the largest integer k such that $\sigma_k > \tau$.*

It is worth noting that some existing routines such as `GEQP3` in LAPACK, which computes a QR factorization with column pivoting, will only return the factors defining the factorizations and do not attempt to determine the numerical rank of the matrix within the routine. Hence when implementing algorithms we will need to use a suitable tolerance, for example $\tau = u\|A\|$ where u is the unit roundoff.

Setting to zero quantities close to the unit roundoff can be justified by the argument that doing so involves making perturbations of the same size as the error inherent in storing the data as floating point numbers.

The most accurate (although also most expensive) way to determine the numerical rank of a matrix is via the SVD [25]. A less expensive alternative to the SVD is a QR factorization with column pivoting, which is implemented robustly and efficiently in LAPACK. However, this factorization does not yield the correct numerical rank for some matrices. An example of such a matrix is the Kahan matrix (defined by the parameters $n = 90$ and $\theta = 1.2$) described in [61]. Computing a QR factorization with column pivoting in MATLAB yields an upper triangular matrix whose smallest diagonal element is 1.9039e-3 and not small (relative to the unit roundoff), but the smallest singular value is 3.9607e-15 and the matrix has numerical rank 89 based on a tolerance $\tau \approx u$. In this case QR with column pivoting has provided an overestimate of the rank. Further information on QR with column pivoting overestimating the rank of a matrix is contained in Section 3.3.1 on page 59. After computing a QR factorization the resulting R matrix is upper triangular, so it would be inexpensive to apply a condition number estimator (such as MATLAB's `condest`) to check the singularity, as the condition number estimator normally tries to computes an LU factorization which is unnecessary for upper triangular matrices. We note that, for the algorithms we later present, overestimating the rank is much more favorable than underestimating the rank. For example, in the case of preprocessing the standard eigenproblem to remove a zero eigenvalue, overestimating the rank means we fail to remove a zero eigenvalue; underestimating the rank would be much worse however, since it would mean we are essentially setting to zero an eigenvalue which is not close to zero relative to the unit roundoff.

Another option to find the numerical rank of a matrix is to compute a rank revealing QR factorization [10]—for example one of the UTV type factorizations [26]. Some of these methods are iterative however, so the cost of their computation cannot be determined a priori. They are also not currently implemented in a robust,

blocked and efficient form in a library such as LAPACK.

1.5 The Polynomial Eigenvalue Problem

The *polynomial eigenvalue problem* (PEP) is to find scalar eigenvalues λ, and associated nonzero left and right eigenvectors y, x such that

$$y^* P_\ell(\lambda) = 0, \quad P_\ell(\lambda)x = 0, \quad x, y \neq 0,$$

where

$$P_\ell(\lambda) = \lambda^\ell A_\ell + \cdots + \lambda A_1 + A_0$$

with $A_i \in \mathbb{C}^{n\times n}$, $i = 0\colon \ell$, and $A_\ell \neq 0$, and throughout this thesis we will assume that the degree ℓ matrix polynomial $P_\ell(\lambda)$ is regular, that is, $\det(P_\ell(\lambda)) \not\equiv 0$.

The most commonly occurring case in applications is the *quadratic eigenvalue problem* (QEP), a special case of the PEP with $\ell = 2$. In these applications, the quadratic matrix polynomial $Q(\lambda)$ is often written as

$$Q(\lambda) = \lambda^2 M + \lambda C + K,$$

where M is the mass matrix, C the damping matrix and K is the stiffness matrix.

1.5.1 Structures and Properties of Matrix Polynomials

A matrix polynomial may exhibit a number of structures, for example symmetric coefficients, hyperbolicity, and properties such as real or complex coefficients, and singular leading or trailing coefficients. Such structure will be exhibited in particular properties of the eigenvalues and eigenvectors. For example, when M, C, and K are symmetric, then if λ is an eigenvalue with right eigenvector x, then x is a left

eigenvector of the eigenvalue $\bar{\lambda}$. A summary of properties associated with different structures is given in the review article [58].

1.5.2 Singular Leading and Trailing Coefficients

We call the A_0 coefficient of a matrix polynomial P_ℓ the trailing coefficient, and the A_ℓ coefficient the leading coefficient. If either or both of these matrices are singular then we know the matrix polynomial will have zero or infinite eigenvalues. Specifically, if $\text{rank}(A_0) = r_0$ and $\text{rank}(A_\ell) = r_\ell$ then we have the following lower bounds:

$$\begin{aligned} &\#\text{ of zero eigenvalues} \geqslant n - r_0 \\ &\#\text{ of infinite eigenvalues} \geqslant n - r_\ell. \end{aligned}$$

Also, if $\lambda = 0$ is an eigenvalue contributed by A_0 then its corresponding eigenvector is in fact a null vector of A_0 (a null vector $x \neq 0$ of A satisfies $Ax = 0$). A similar argument applies to infinite eigenvalues with null vectors of A_ℓ. There may be additional zero or infinite eigenvalues if the leading or trailing coefficients have a nontrivial null space intersection with the coefficients $A_i,\ i = 1\colon \ell - 1$.

Infinite eigenvalues $\lambda = \infty$ are in fact zero eigenvalues of the reversal polynomial $\text{rev}(P_\ell)$. The reversal polynomial of $P_\ell(\lambda) = \lambda^2 A_\ell + \cdots + \lambda A_1 + A_0$ is given by

$$\text{rev}(P_\ell(\lambda)) := \lambda^\ell P_\ell(1/\lambda) = \lambda^\ell A_0 + \lambda^{\ell-1} A_1 + \cdots + A_\ell$$

and $\lambda = \infty$ as an eigenvalue of P_ℓ is mapped to $\lambda = 0$ as an eigenvalue of $\text{rev}(P_\ell(\lambda))$. In order to treat infinite eigenvalues more comfortably, one can work with the eigenvalue parameter written in homogeneous form, that is writing $\lambda = \alpha/\beta$, with not both of α and β zero. For real eigenvalues α and β can be normalized and

thought of as a point on the unit circle. The matrix polynomial in homogeneous form is obtained upon substituting $\lambda = \alpha/\beta$ and defining

$$P_\ell(\alpha, \beta) = \beta^\ell P_\ell(\lambda) = \alpha^\ell A_\ell + \cdots + \alpha\beta^{\ell-1} A_1 + \beta^\ell A_0.$$

Thus zero eigenvalues take the form $(\alpha, \beta) = (0, \beta)$ with $\beta \neq 0$ and infinite eigenvalues the form $(\alpha, \beta) = (\alpha, 0)$ with $\alpha \neq 0$.

1.5.3 Measuring the Accuracy of Computed Eigensolutions

In this section we describe two quantities important in measuring the accuracy of computed solutions to problems in finite precision arithmetic: backward error and condition numbers. In Chapter 3 we explain an implementation of a general purpose code to solve polynomial eigenvalue problems which will return both eigenvalue condition numbers and backward errors for computed eigenpairs. In this section we give the formulae used to compute these two quantities for the case of general degree ℓ matrix polynomials. In our algorithms we allow for the possibility of both infinite and zero eigenvalues, so to allow an equal treatment of finite, zero and infinite eigenvalues we represent the eigenvalues in homogeneous form as mentioned in the previous section.

The definition of backward error of a right eigenpair $(x; \alpha, \beta)$ of a degree ℓ matrix polynomial written in homogenous form

$$P_\ell(\alpha, \beta) = \sum_{i=0}^{\ell} \alpha^i \beta^{\ell-i} A_i$$

is given next. In this section ΔA_i denotes an unstructured perturbation to the A_i coefficient.

Definition 8 (Relative normwise backward error of an approximate right eigenpair)**.** *The relative normwise backward error of an approximate right eigenpair* $(x;\alpha,\beta)$ *of a polynomial* $P_\ell(\alpha,\beta)$ *is defined as*

$$\eta_{P_\ell}(x;\alpha,\beta) = \min\left\{\epsilon\colon (P_\ell(\alpha,\beta)+\Delta P_\ell(\alpha,\beta)\,)x=0, \quad \|\Delta A_i\|_2 \le \epsilon\,\|A_i\|_2\,,\ i=0:\ell\right\}, \tag{1.5.1}$$

where $\Delta P_\ell(\alpha,\beta) = \sum_{i=0}^{\ell}\alpha^i\beta^{\ell-i}\Delta A_i$.

An explicit expression [58] for relative backward errors of right eigenpairs $(x;\alpha,\beta)$ of degree ℓ matrix polynomials is given by

$$\eta_{P_\ell}(x;\alpha,\beta) = \frac{\|P_\ell(\alpha,\beta)x\|_2}{\left(\sum_{i=0}^{\ell}|\alpha|^i|\beta|^{\ell-i}\,\|A_i\|_2\right)\|x\|_2}. \tag{1.5.2}$$

The representation (α,β) of an eigenvalue λ is not unique, however (1.5.2) is independent of the scaling of (α,β).

Moving to condition numbers, Dedieu and Tisseur [13] present condition numbers for eigenvalues of matrix polynomials. The condition number $\kappa_{P_\ell}(\alpha,\beta)$ is defined for simple eigenvalues both finite (including zero) or infinite. It provides a bound on the angle between an exact eigenvalue (α,β) and a perturbed eigenvalue $(\tilde\alpha,\tilde\beta)$. The angle is based on viewing an eigenvalue as a line that goes through the origin in the complex plane to the point (α,β) solving $\det(P_\ell)=0$. For an eigenvalue (α,β) of a degree ℓ matrix polynomial, this condition number is defined as

$$\kappa_{P_\ell}(\alpha,\beta) = \max_{\|\Delta A\|\le 1}\frac{\|K(\alpha,\beta)\,\Delta A\|_2}{\|[\alpha,\beta]\|_2} \tag{1.5.3}$$

where $\Delta A = [\Delta A_0, \Delta A_1, \ldots, \Delta A_\ell]$. $K(\alpha,\beta) : (\mathbb{C}^{n\times n})^{(\ell+1)} \to T_{(\alpha,\beta)}\mathbb{P}_1$, $T_{(\alpha,\beta)}\mathbb{P}_1$ is a tangent space at (α,β) to $\mathbb{P}_1$ the projective space of lines through the origin in $\mathbb{C}^2$. The condition operator for the eigenvalue (α,β) is defined as the differential

of the map from the $(\ell+1)$-tuple $(A_0,\ldots,A_\ell)$ to (α,β) in projective space. The condition number can be computed using the expression given below.

Theorem 2 (see Theorem 2.3 [32]). *The normwise condition number $\kappa_{P_\ell(\alpha,\beta)}$ of a simple eigenvalue (α,β) with right eigenvector x and left eigenvector y of a degree ℓ matrix polynomial is given by*

$$\kappa_{P_\ell(\alpha,\beta)} = \frac{(\sum_{i=0}^{\ell}|\alpha|^{2i}|\beta|^{2(\ell-i)}\,\|A_i\|_2^2)^{1/2}\,\|y\|_2\,\|x\|_2}{|y^*(\bar{\beta}\mathcal{D}_\alpha P_\ell - \bar{\alpha}\mathcal{D}_\beta P_\ell)|_{\alpha,\beta}\,x|} \tag{1.5.4}$$

where $\mathcal{D}_\alpha = \frac{\partial}{\partial\alpha}$ and $\mathcal{D}_\beta = \frac{\partial}{\partial\beta}$.

An alternative condition number is $\kappa_{P_\ell}(\lambda)$ which is a direct generalization of Wilkinson's condition number [63] for the standard eigenvalue problem $Ax=\lambda x$ and measures the relative change in an eigenvalue, however it is not defined for zero or infinite eigenvalues. In Chapter 3 we describe an algorithm which allows for the possibility of zero and infinite eigenvalues, hence we use $\kappa_{P_\ell}(\alpha,\beta)$.

1.5.4 Applications

Quadratic eigenvalue problems arise in many applications, for example, dynamic analysis of mechanical systems in acoustics, structural mechanics, electrical circuit simulation, gyroscopic systems, molecular dynamics and constrained least squares. Information about many more applications can be found in the review article [58]. NLEVP [7] is a collection of nonlinear eigenvalue problems, some from applications and some constructed to have specific properties. The problems are described and the matrices defining the problems are available in a MATLAB toolbox.

A quadratic eigenvalue problem often results from vibrational/dynamic analysis of structures discretized by the finite element method. The equations of motion

are:

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = f(t), \tag{1.5.5}$$

where M, C, and $K \in \mathbb{C}^{n\times n}$ are mass, damping and stiffness matrices arising from a finite element discretization, the vector $f(t)$ is a forcing term, and $q(t), f(t)$ are n-vectors. When looking for exponential solutions, of the form $q(t) = e^{\lambda t}x$, the first step is the solution of the homogeneous equation, which arises from setting $f(t) = 0$ in (1.5.5). Then, substituting $q(t) = e^{\lambda t}x$ we obtain the QEP $(\lambda^2 M + \lambda C + K)x = 0$ with $Q(\lambda) = \lambda^2 M + \lambda C + K$. We now describe in more detail a number of applications that yield quadratic eigenvalue problems.

Shaft Problem

A finite element model of a shaft on bearing supports with a damper, modelled with the finite element package MSC/Nastran [27], yields a quadratic eigenvalue problem $Q(\lambda) = \lambda^2 M + \lambda C + K$, with $M, C, K \in \mathbb{R}^{400\times 400}$. In this example the coefficients are very sparse. The mass matrix M has rank 199 and contributes a large number of infinite eigenvalues. A schematic of the shaft can be found in Figure 1.1.

Damped Beam Problem

A model of a beam as seen in Figure 1.2, simply supported at both ends and damped at the midpoint is considered in [33].

The transverse displacement $u(x,t)$ is governed by the partial differential equation,

$$\rho A\frac{\partial^2 u}{\partial t^2} + c(x)\frac{\partial u}{\partial t} + EI\frac{\partial^4 u}{\partial x^4} = 0.$$

with associated boundary conditions: $u(x,t) = u''(x,t) = 0,\ x = 0, L$. Solving for exponential solutions of the form $u(x,t) = e^{\lambda t}v(x,\lambda)$ yields an eigenproblem for

Figure 1.1: Schematic of a shaft on bearing support

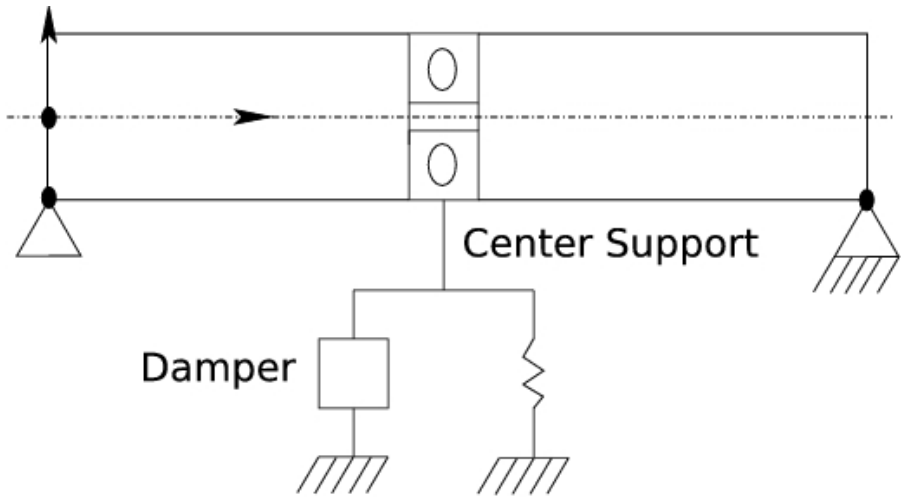
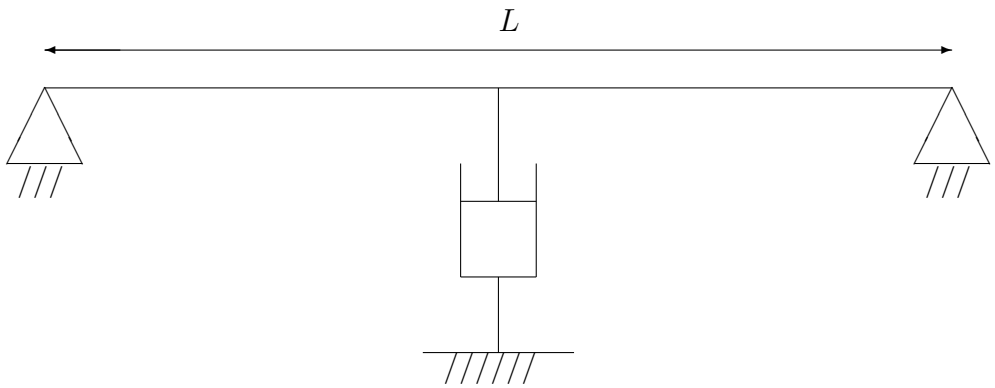


Figure 1.2: Simply supported beam with damping



free vibrations of the form

$$\lambda^2 \rho A v(x,\lambda) + \lambda c(x) v(x,\lambda) + EI \frac{\partial^4}{\partial x^4} v(x,\lambda) = 0.$$

After discretizing the PDE to obtain a finite dimensional problem, one is left with a quadratic matrix polynomial with mass, damping and stiffness matrices, M, C and K with the properties $M, K > 0$ and $C \geqslant 0$. Due to the inherent structure of the problem, it is known that the spectrum of the quadratic lies in the closed left hand half of the complex plane.

Linear Spring Dashpot with Maxwell Elements

Gotts [22] describes a quadratic eigenvalue problem arising from a finite element model of a linear spring in parallel with Maxwell elements (a Maxwell element is a spring in series with a dashpot), for a diagram see Figure 1.3. This quadratic is also included in the MATLAB toolbox NLEVP [7] under the name `spring_dashpot`. The quadratic is of the form

$$Q(\lambda) = \lambda^2 M + \lambda C + K, \quad M, C, K \in \mathbb{R}^{10\times 10},$$

where the mass matrix M is symmetric and rank deficient (and hence contributes infinite eigenvalues), the damping matrix C is rank deficient and block diagonal, and the stiffness matrix K is symmetric and exhibits "arrowhead" structure. The

form of the matrix for 4 Maxwell elements is given below

$$M = \text{diag}(\rho\widetilde{M}_{11}, 0), \quad C = \text{diag}(0, \eta_1\widetilde{K}_{11}, \cdots, \eta_4\widetilde{K}_{55}),$$

$$K = \begin{bmatrix} \alpha_\rho\widetilde{K}_{11} & & B & \\ & e_1\widetilde{K}_{22} & 0 & 0 \\ B^T & 0 & \ddots & 0 \\ & 0 & 0 & e_4\widetilde{K}_{55} \end{bmatrix},$$

where

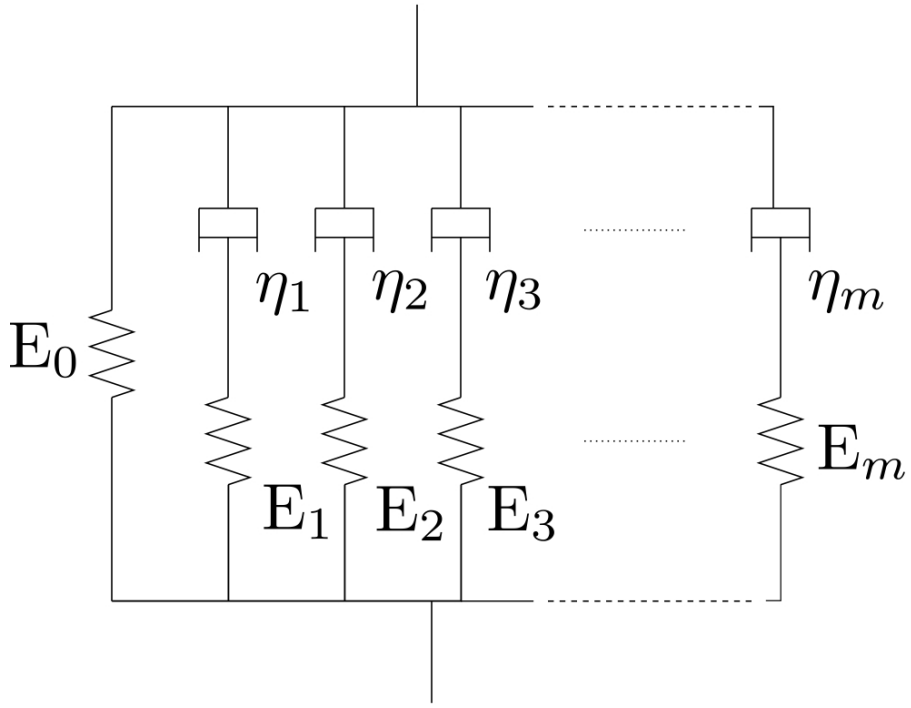
$$B = \begin{bmatrix} -\xi_1\widetilde{K}_{12}, & \ldots, & -\xi_4\widetilde{K}_{15} \end{bmatrix}.$$

$\widetilde{M}_{ij}$ and $\widetilde{K}_{ij}$ are the ijth element mass and stiffness matrices, and

$$\alpha_\rho = \sum_{k=0}^{4} \xi_k.$$

$\eta_i,\ i = 1\colon 5,\ \xi_j\ j = 0\colon 5,\ e_k,\ k = 1\colon 4$ and ρ (the material density) are scalar parameters.

Figure 1.3: Spring/dashpot with Maxwell elements



Chapter 2

Solving PEPs by Linearization

Generalized eigenvalue problems $(A - \lambda B)x = 0$ can be solved by computing the generalized Schur form. There is no extension however, of the generalized Schur form for matrix pencils to matrix polynomials of degree two or higher. The standard approach to solve PEPs both theoretically and numerically is to convert the degree ℓ matrix polynomial with n-by-n matrix coefficients to a linear matrix pencil $\lambda X+Y$ of dimension ℓn-by-ℓn, a process known as linearization. The linearized problem is a generalized eigenproblem which can be solved by computing the generalized Schur form. In this chapter we explain the linearization process, solution of the linear problem, and recovery of the solution of the polynomial problem from that of the linear problem.

2.1 Linearizations of Matrix Polynomials

The first step in solving the PEP by linearization is to find an ℓn-by-ℓn linear matrix pencil $L(\lambda)$ that is a linearization of the polynomial $P_\ell(\lambda)$ in that it satisfies the following definition.

Definition 9 (Linearization [20])**.** *The pencil $L(\lambda) = \lambda X + Y$ is a linearization of*

the degree ℓ matrix polynomial $P_\ell(\lambda)$ if

$$E(\lambda)L(\lambda)F(\lambda) = \begin{bmatrix} P_\ell(\lambda) & 0 \\ 0 & I_{n(\ell-1)} \end{bmatrix},$$

where $E(\lambda)$ and $F(\lambda)$ are matrix polynomials with constant nonzero determinants (and are said to be unimodular, and have inverses that are defined over the field of matrix polynomials).

Research on linearizations of matrix polynomials has been very active lately including generalization of its definition [42], [41], derivation of new (structured) linearizations [1], [3], [4], [31], [46], [47] and analysis of the influence of the linearization process on the accuracy and stability of computed solutions [30], [32], [33]. Factors influencing the choice of linearization include the properties of the matrix polynomial—for example structure in the coefficients, and the properties of the linearization with regard to solving the original polynomial problem.

Recent work [47] has identified vector spaces of pencils that are potential linearizations of degree ℓ matrix polynomials $P_\ell(\lambda) = \lambda^\ell A_\ell + \cdots + \lambda A_1 + A_0$. Defining $\Lambda = [\lambda^{\ell-1}, \lambda^{\ell-2}, \ldots, 1]^T$ these vector spaces, which contain an infinite number of linearizations of P_ℓ are

$$\mathbb{L}_1(P_\ell) = \left\{ L(\lambda) \colon L(\lambda)(\Lambda \otimes I_n) = v \otimes P_\ell(\lambda), \quad v \in \mathbb{C}^\ell \right\}, \tag{2.1.1}$$

$$\mathbb{L}_2(P_\ell) = \left\{ L(\lambda) \colon (\Lambda^T \otimes I_n)L(\lambda) = w^T \otimes P_\ell(\lambda), w \in \mathbb{C}^\ell \right\}, \tag{2.1.2}$$

$$\mathbb{DL}(P_\ell) = \mathbb{L}_1(P_\ell) \cap \mathbb{L}_2(P_\ell). \tag{2.1.3}$$

In practice, the most commonly used linearizations are the companion forms. For example the first companion linearization of a quadratic $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$

has the form

$$C_1(\lambda) = \lambda \begin{bmatrix} A_2 & 0 \\ 0 & I_n \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ -I_n & 0 \end{bmatrix}, \tag{2.1.4}$$

which is in the vector space $\mathbb{L}_1(Q)$ with vector $v = e_1$. An example of a symmetry preserving linearization of a real symmetric quadratic Q ($A_i = A_i^T,\ i = 0\colon 2$), with $\det(A_0) \neq 0$, is

$$L_1(\lambda) = \lambda \begin{bmatrix} A_2 & 0 \\ 0 & -A_0 \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ A_0 & 0 \end{bmatrix} \tag{2.1.5}$$

which is in the space $\mathbb{DL}(Q)$ with vector $v = e_1$. Such symmetry preserving linearizations will be relevant in Chapter 4 in the area of structure preserving transformations for quadratic matrix polynomials.

2.2 Solving PEPs by Linearization in Finite Precision Arithmetic

We begin by first considering a numerical example which illustrates the theme of this section. In finite precision arithmetic we have computed the spectrum of the damped beam quadratic [33], solving the quadratic eigenproblem by linearization (using MATLAB's `eig` function) with three different linearizations of the original quadratic: L_1 and C_1 already mentioned (equations (2.1.4) and (2.1.5)), and for $\det(A_2) \neq 0$,

$$L_2(\lambda) = \lambda \begin{bmatrix} 0 & A_2 \\ A_2 & A_1 \end{bmatrix} + \begin{bmatrix} -A_2 & 0 \\ 0 & A_0 \end{bmatrix}, \tag{2.2.1}$$

which is in the space $\mathbb{DL}(Q)$ with vector $v = e_2$. Theoretically, in exact arithmetic we know the eigenvalues of the quadratic problem are identical to those of the linearized problem, and further, the eigenvalues should be the same regardless of

which linearization is taken.

The three plots in the left hand side of Figure 2.1 show the computed spectrum of the damped beam quadratic solved using the three linearizations (2.1.4)–(2.2.1). It is shown in [33] that due to the properties of the problem, all the eigenvalues should be in the left half of the complex plane. Even visually we can see that the spectrum for the three different linearizations is different, and not all eigenvalues are in the left half of the complex plane, both in contradiction to the theory.

In the next section we discuss recent theory which explains this situation and techniques that can be used to improve the accuracy of computed eigenvalues.

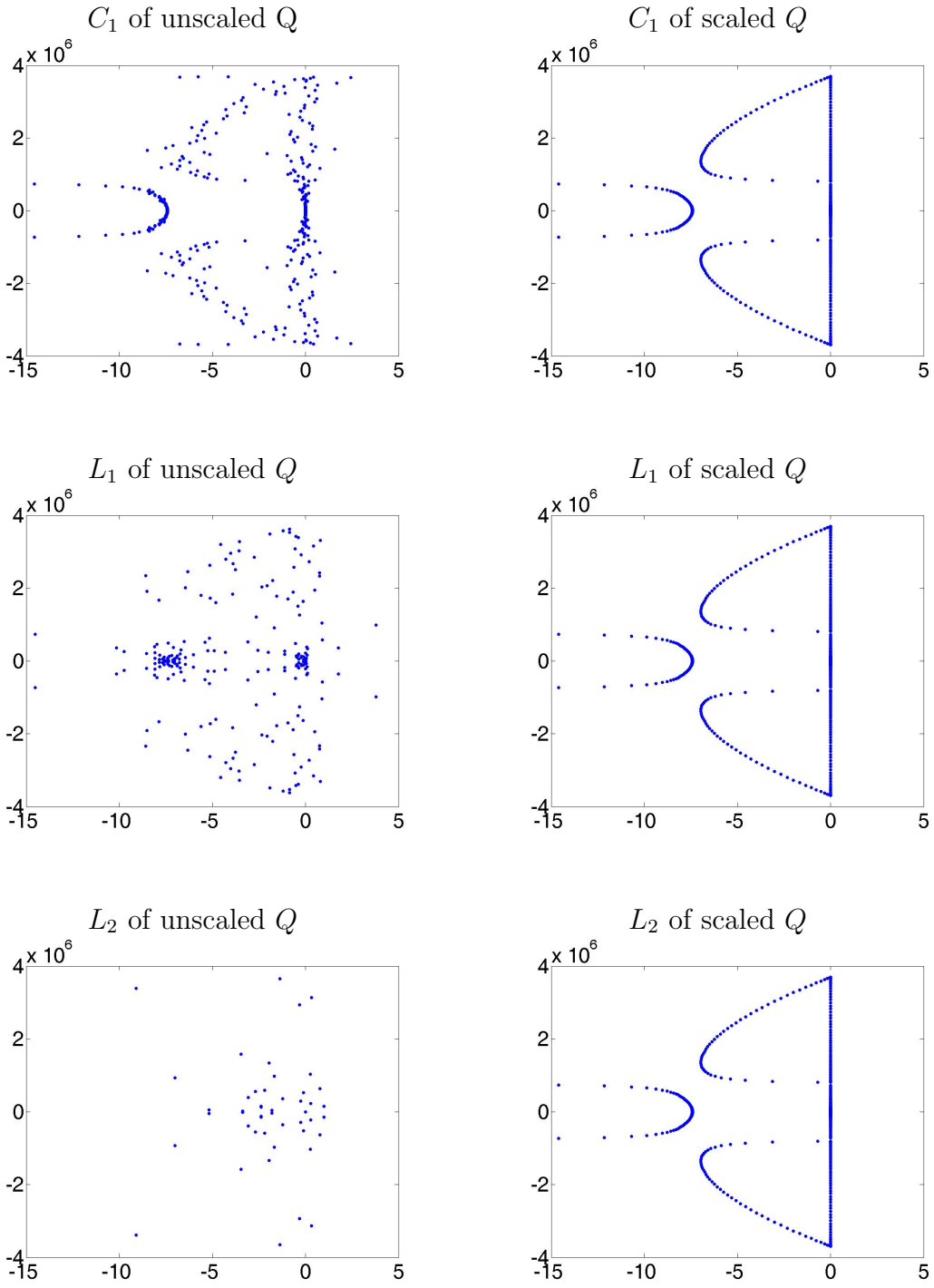
The three plots in the right hand side of Figure 2.1 show the computed spectrum when these techniques have been applied to the damped beam quadratic. We see that at least visually the computed spectrum is the same for the three linearizations.

2.3 Accuracy and Conditioning of Solutions to QEPs Solved by Linearization

In this section we discuss recent developments in the theory that can help explain the accuracy of computed eigenvalues of matrix polynomials, solved by linearization in finite precision arithmetic, and techniques that can be applied to attempt to improve the situation. What follows is phrased for quadratic matrix polynomials. In Section 2.3.1 we comment on matrix polynomials of degree higher than two.

We now define notation used in the rest of this chapter. Let $Q(\lambda)$ be the original (unscaled) matrix polynomial, and $\widetilde{Q}(\mu)$ the quadratic scaled using the Fan, Lin and Van Dooren scaling (which we will define). Let L be the linearization of the scaled quadratic $\widetilde{Q}$, where $L(\mu)z = 0$ such that the right eigenvector has the form $z = [z_1^T,\ z_2^T]^T$ where z_1 is the first and z_2 the last n components of z. We write

Figure 2.1: Computed spectrum of unscaled/scaled damped beam quadratic for linearizations C_1, L_1, and L_2, (as defined in (2.1.4)–(2.2.1)) using MATLAB's `eig` to solve the linear problem



quantities computed in finite precision as $\widehat{\mu}, \widehat{z}_1, \widehat{z}_2$ etc.

Linear problems/generalized eigenvalue problems of the form $(A - \lambda B)x = 0$ can be solved with the QZ algorithm which gives backward stable solutions for the linear problem. However, if we linearize a quadratic matrix polynomial, solve the resulting linear problem with QZ and extract a solution for the quadratic matrix polynomial, that solution will not in general be backward stable for the quadratic problem. The theorem below shows that backward stable solutions will be returned when solving by linearization with companion type linearizations, if the coefficient matrices have unit norm.

Theorem 3 ([56, Thm. 7])**.** *When solving the QEP $Q(\lambda)x = 0$ with $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$, if $\|A_2\|_2 = \|A_1\|_2 = \|A_0\|_2 = 1$ then solving the GEP using a companion type linearization, with a backward stable algorithm (e.g., the QZ algorithm) for the GEP is backward stable for the QEP.*

The scaling of Fan, Lin and Van Dooren [14] attempts to achieve the above, by rescaling the eigenvalue parameter to $\lambda = \mu\delta$ and multiplying the original quadratic by a nonzero scalar γ. This yields the scaled quadratic

$$\widetilde{Q}(\mu) = \delta Q(\mu) = \mu^2 \widetilde{A}_2 + \mu \widetilde{A}_1 + \widetilde{A}_0.$$

The coefficients of the scaled quadratic have the form

$$\widetilde{A}_2 = \gamma^2 \delta A_2, \quad \widetilde{A}_1 = \gamma \delta A_1, \quad \widetilde{A}_0 = \delta A_0$$

where $\gamma = \sqrt{\frac{\|A_0\|_2}{\|A_2\|_2}}$ and $\delta = \frac{2}{\|A_0\|_2 + \gamma \|A_1\|_2}$. This scaling has no effect on condition numbers or backward errors for eigenvalues of the quadratic, but attempts to improve the condition numbers and backward errors of eigenpairs of Q recovered from solving the linear problem $L(\mu)z = 0$ and $w^* L(\mu) = 0$ using a linearization L.

We now present the link between this scaling and recent theory that explains the impact of scaling on backward error and conditioning.

The scaling of Q can be measured by the quantity [32]

$$\rho = \frac{\max_i(\|A_i\|_2)}{\min(\|A_0\|_2, \|A_2\|_2)},$$

where for well scaled problems $\rho \approx 1$, and scaling Q generally decreases ρ.

Sufficient conditions for approximate equality between backward errors for eigenpairs of the quadratic and linearization, $\eta_Q \approx \eta_L$ are given in [30, 32]. Where there is a choice of which component of the eigenvector of the linearization (z_1 or z_2) to take as an eigenvector of the quadratic (here we focus on right eigenvectors x, results for left eigenvectors also exist), the theory says which of the first or last n components to take. We give the details for $L = C_1$ and L_1 (see [30, 32] for $L = L_2$). In Chapter 3 the second companion linearization $C_2(\lambda)$ is used, we present relevant information in that chapter.

Starting with companion linearizations for $\eta_P \approx \eta_C$ we require $\|A_1\| \leqslant \|A_2\| \approx \|A_0\|$ then if $|\lambda| \geqslant 1$ choose $x = z_1$, else choose $x = z_2$.

For the linearization L_1 the sufficient conditions depend on eigenvalue magnitude as well as the choice of eigenvector. For $|\lambda| \geqslant 1$ the condition is $\rho \approx 1$ in which case choose $x = z_1$ as the eigenvalue of Q. For $|\lambda| \leqslant 1$ the condition is $\rho \max(1, (\|A_1\| + \|A_0\|)\|A_2^{-1}\|) \approx 1$ and take $x = z_2$.

Upon proceeding from the quadratic to the linear problem, we can measure the growth of the eigenvalue condition number and the backward error of eigenpairs of $\widetilde{Q}$ recovered from the solution of the linear problem.

We now look at what happens to the backward error $\psi(\mu)$ and condition number $\phi(z_k)$ growth factors $\eta_L(\mu) = \psi_L(z_k)\kappa_Q(\mu)$, $\kappa_L(\mu) = \phi_L(\mu)\kappa_Q(\mu)$, when we scale Q using the Fan, Lin and Van Dooren scaling. We will need the quantities

[33]

$$\tau = \frac{\|A_1\|_2}{\sqrt{\|A_0\|_2 \|A_2\|_2}}, \text{ and } \omega(\mu) = \frac{1+\tau}{1+\frac{|\mu|}{1+|\mu|^2}\tau}.$$

The following expressions for the growth factors for $L = C_1, L_1$, and L_2 are presented in [33]:

$$\phi_{C_1} \approx \omega(\widehat{\mu}), \qquad \psi_{C_1}(\hat{z}_k) \approx \omega(\widehat{\mu})\frac{\|\hat{z}\|_2}{\|\hat{z}_k\|_2}, \tag{2.3.1}$$

$$\phi_{L_1} \approx \frac{1+|\widehat{\mu}|}{|\widehat{\mu}|}\,\omega(\widehat{\mu}), \qquad \psi_{L_1}(\hat{z}_k) \approx \nu(k)\,\omega(\widehat{\mu})\frac{\|\hat{z}\|_2}{\|\hat{z}_k\|_2}, \tag{2.3.2}$$

$$\phi_{L_2} \approx (1+|\widehat{\mu}|)\,\omega(\widehat{\mu}), \qquad \psi_{L_2}(\hat{z}_k) \approx \nu(k)\,\omega(\widehat{\mu})\frac{\|\hat{z}\|_2}{\|\hat{z}_k\|_2}, \tag{2.3.3}$$

where for L_1: $\nu(1) = 1$ and $\nu(2) = ||\widehat{A}_0^{-1}||_2$, and for L_2: $\nu(1) = 1$ and $\nu(2) = ||\widehat{A}_2^{-1}||_2$.

For the scaled problem it holds that

$$1 \leqslant \omega(\mu) \leqslant \min\left\{1+\tau, \frac{1+|\mu|^2}{|\mu|}\right\},$$

thus, both backward error and condition number are essentially optimal for C_1 for all λ, for L_1 if $|\hat{\mu}| > 1$ and for L_2 if $|\hat{\mu}| < 1$, if $\omega(\mu) = O(1)$, the physical interpretation of this is that for mechanical systems, this is the case for systems that are not too heavily damped, that is $\|A_1\|_2 \lesssim \sqrt{\|A_0\|_2 \|A_2\|_2}$ where $\tau < 1$. The class of elliptic quadratics fall into this category (of not too heavily damped problems), since A_2 is positive definite, and for all nonzero x it holds that $(x^*A_1x)^2 < 4(x^*A_0x)(x^*A_2x)$. Optimality also holds if $|\mu| = O(1)$.

Due to the choice of eigenvector of the quadratic from the solution of the linear problem, the expressions of backward error growth factor depend on z_k (whether the first or last n components of $\widehat{z}$ are selected as an eigenvector x of the quadratic).

Applications yielding examples of quadratics for which the scaling of Fan, Lin,

and Van Dooren does not improve the inherent scaling of the problem are available. One such example is the `cd_player` problem from NLEVP [7]. Before applying scaling we have

$$\|A_2\|_2 = 1.0, \quad \|A_1\|_2 = 1.0\text{e}7, \quad \|A_0\|_2 = 2.3\text{e}5,$$

and after scaling,

$$||\widetilde{A}_2||_2 = 9\text{e-}5, \quad ||\widetilde{A}_1||_2 = 2, \quad ||\widetilde{A}_0||_2 = 9\text{e-}5.$$

This is an example of a quadratic that is heavily damped with

$$\|A_1\|_2 \gg \max(\|A_2\|_2, \|A_0\|_2).$$

As seen in equations (2.3.1)–(2.3.3), the theory explains that eigenvalues of linearizations L of heavily damped quadratics can have large condition numbers (for L) and backward errors of the original quadratic.

Another scaling strategy is tropical scaling [19], of the same type as the scaling as Fan, Lin and Van Dooren, of the form $Q(\lambda) \leftarrow \widetilde{Q}(\mu)$. The parameters δ and γ are formed after computing the tropical roots of a scalar quadratic tropical polynomial, whose coefficients are the norms of the coefficients of Q. This yields two roots and therefore two scalings. Analysis in [52] shows that if the roots are equal this is equivalent to the scaling of Fan, Lin and Van Dooren. When the roots are unequal, one scaling attempts to improve the accuracy of small eigenvalues and the other the accuracy of large eigenvalues.

2.3.1 Scaling Higher Degree Matrix Polynomials

For higher degree matrix polynomials (cubics and above), the eigenvalue parameter scaling of Fan, Lin and Van Dooren is extended to higher degree polynomials in [6] to a scaling of the form

$$\widetilde{P}_\ell(\mu) = \gamma P_\ell(\delta\lambda). \tag{2.3.4}$$

The previously mentioned quantity ρ, which measures the scaling of the problem naturally extends to

$$\rho = \frac{\max_i(\|A_i\|_2)}{\min\left(\|A_0\|_2, \|A_\ell\|_2\right)}.$$

The choice of $\delta = \sqrt{\|A_0\|_2/\|A_\ell\|_2}$ can be shown [6] to minimize $\rho(\delta)$ for $P_\ell(\delta\lambda)$ in (2.3.4). If $\rho \approx 1$ then for a given eigenvalue there is a linearization in the space $\mathbb{DL}(P_\ell)$ such that the eigenvalue condition number for the linearization is approximately the same as the condition number for the original polynomial [32]. For companion linearizations, which are not in $\mathbb{DL}(P_\ell)$, in addition to $\rho \approx 1$ we also require [32] that $\|A_i\|_2 \approx 1,\ i = 0\colon \ell$ and γ in (2.3.4) is chosen to attempt to achieve $\|A_i\|_2 \approx 1,\ i = 0\colon \ell$. The choice of $\gamma = a^T\mathbf{1}/a^Ta$ where a is a vector with $a_i = \|A_i\|_2,\ i = 0\colon \ell$ and $\mathbf{1}$ is a vector of ones of length $\ell + 1$ minimizes $\|\gamma a - \mathbf{1}\|_2^2$ or we might choose $\gamma = \max_i(\|A_i\|_2)^{-1}$ provided $\|A_i\|_2 \neq 0$. If $\|A_0\|_2 = \|A_\ell\|_2$ then $\delta = 1$ and scalings of the form $\mu = \delta\lambda$ will not improve ρ.

2.3.2 Techniques to Improve Accuracy of Eigenvalues of Specific Magnitude

For problems that are not too heavily damped, the Fan, Lin and Van Dooren scaling yields optimal conditioning and backward error results for all eigenvalues for C_1, for all eigenvalues inside the unit circle for L_1 and for all eigenvalues outside the unit circle for L_2. If however, we are mainly interested in computing eigenvalues

of a specific magnitude $\zeta > 0$, then the technique of balancing can be attempted.

Balancing is based on the observation that for computed eigenvalues $\widehat{\lambda}$ of a single matrix A (the standard eigenvalue problem) λ is perturbed by at least $u\|A\|$ with u the unit roundoff. We can attempt to increase the accuracy of the computed eigenvalue by reducing $\|A\|$. For the standard eigenvalue problem see [54]. The technique is extended to matrix pencils in [59] and [44] (the methods differ in the cost function minimized). The method for matrix pencils in [44] is extended to matrix polynomials in [6], and involves determining diagonal scaling matrices D_1, D_2 to form a scaled matrix polynomial $\widetilde{P}_\ell(\lambda) = D_1 P_\ell(\lambda) D_2$. The matrices D_1 and D_2 aim to achieve

$$\sum_{k=0}^{\ell} \zeta^{2k} \|D_1 A_k D_2 e_i\|_2^2 = 1, \quad \sum_{k=0}^{\ell} \zeta^{2k} \left\|e_j^T D_1 A_k D_2\right\|_2^2 = 1, \ i,j = 1\colon n, \tag{2.3.5}$$

where $\zeta > 0$ is the magnitude of the desired eigenvalues. Numerical experiments are also presented, showing improvement of the accuracy of computed eigenvalues after applying the technique.

2.4 The QZ Algorithm

To solve the generalized eigenvalue problem that arises from the linearization process we use the QZ algorithm [51] as implemented robustly and efficiently in the LAPACK [2] routine `xGGEV`. For simplicity we will describe the process working with real arithmetic (`DGGEV`); `ZGGEV` is the version implemented working with complex arithmetic. The LAPACK routines are also used when the MATLAB `eig` function is called. In this section we focus on aspects of the algorithm that are relevant to later chapters in this thesis—full details can be found in [2, 51] or [21].

The QZ algorithm computes the generalized Schur decomposition of a matrix

pair (A, B) (see Definition 4) to obtain the eigenvalues of the pencil $A - \lambda B$; additional steps can then be carried out to obtain eigenvectors. The implementation of the QZ process in `DGGEV` computes the eigenvalues λ of a given matrix pencil $A - \lambda B$ with $A, B \in \mathbb{R}^{n \times n}$ and optionally the associated right and left eigenvectors $x, y \in \mathbb{C}^n$.

Algorithm 1 (QZ Algorithm, [51, 21])**.** *Given the matrix pencil $A - \lambda B$ with $A, B \in \mathbb{R}^{n \times n}$, the QZ algorithm computes orthogonal Q and Z such that $Q^T AZ = T$ is quasi upper triangular and $Q^T BZ = S$ is upper triangular. The stages can be summarized as:*

Step 1. *Attempt to permute the pencil $A - \lambda B$ to block upper triangular form, as in Equation* (2.4.1)

Step 2. *Transform B to upper triangular form*

Step 3. *Reduce to Hessenberg triangular form*

Step 4. *Apply QZ iterations to the Hessenberg triangular form (accumulate the orthogonal transformations if eigenvectors are desired).*

Step 5. *(Optional) Compute eigenvectors of the permuted pencil, taking into account the matrices that put the pencil into generalized real Schur form, then transform again to recover eigenvectors of the original unpermuted pencil.*

We briefly expand on the first two stages which will be relevant to Chapter 3.

Step 1 is implemented with `DGGBAL` which attempts to permute the pencil $A - \lambda B$ to the block upper triangular form below:

$$W_1(A - \lambda B)W_2 = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{bmatrix} - \lambda \begin{bmatrix} B_{11} & B_{12} & B_{14} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & B_{33} \end{bmatrix}, \tag{2.4.1}$$

where W_1 and W_2 are permutation matrices, $A_{11}, B_{11}, A_{33}, B_{33}$ are upper triangular, and A_{22}, B_{22} are full. If this form can be achieved then the problem decouples and the remaining spectrum can be computed from the smaller pencil $A_{22} - \lambda B_{22}$.

Step 2 starts with the matrix B and transforms it to upper triangular form by computing its QR factorization, $B = QR$, then setting $A \leftarrow Q^T A$ and $B \leftarrow R$, this is done using the routine `DGEQRF`. A general purpose algorithm for solving quadratic eigenvalue problems presented in Chapter 3 achieves this step using one of the factorizations computed for checking the rank of the leading and trailing coefficients.

2.5 Eigenvectors of Matrix Polynomials from Linearizations

In this section we briefly comment on the recovery of eigenvectors of the polynomial from eigenvectors of the linearization when working in finite precision. We consider the two linearizations C_1 and C_2 shown in Table 2.1, where we use the notation that if x is an exact quantity then $\widehat{x}$ is its computed value in finite precision.

Table 2.1: Theoretical and computed eigenpairs of first (C_1) and second (C_2) companion linearizations of quadratic Q with $\det(A_0) \neq 0$. (Finite nonzero eigenvalues λ.)

Linearization	Theoretical	Finite Precision
$C_1(\lambda) = \lambda \begin{bmatrix} A_2 & 0 \\ 0 & I_n \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ 0 & -I_n \end{bmatrix}$	$z_1 = \begin{bmatrix} \lambda x \\ x \end{bmatrix}$	$\widehat{z}_1 = \begin{bmatrix} \widehat{\lambda}\widehat{x}_1 \\ \widehat{x}_2 \end{bmatrix}$
$C_2(\lambda) = \lambda \begin{bmatrix} A_2 & 0 \\ 0 & I_n \end{bmatrix} + \begin{bmatrix} A_1 & -I_n \\ A_0 & 0 \end{bmatrix}$	$z_2 = \begin{bmatrix} \lambda x \\ -A_0 x \end{bmatrix}$	$\widehat{z}_2 = \begin{bmatrix} \widehat{\lambda}\widehat{x}_3 \\ -A_0\widehat{x}_4 \end{bmatrix}$

In finite precision we have the situation in the last column of Table 2.1 where

the eigenvectors x of the quadratic Q that appear in the eigenvectors z_1 and z_2 of the linearization are not generally equal. In theory, all the eigenvectors x of the quadratic that appear in the eigenvectors z_1 and z_2 are identical and eigenpairs (λ, x) satisfy $Q(\lambda)x = 0$. However, in finite precision, we have computed eigenpairs $(\widehat{\lambda}, \widehat{x}_i)$ that satisfy $Q(\widehat{\lambda})\widehat{x}_i = \epsilon_i \approx 0$ for $i = 1\colon 4$.

It can be seen that for the linearization C_1 we could return either $\widehat{x}_1$ or $\widehat{x}_2$, and for C_2 we either return $\widehat{x}_3$ or (when $\det(A_0) \neq 0$) solve the linear system $-A_0\widehat{x}_4 = \widehat{z}_2(n+1\colon 2n)$ for $\widehat{x}_4$. The key point is that there is a choice to be made as to how the eigenvector is returned. In practice we would like to return the most accurate solution possible. One option implemented by the MATLAB function `polyeig` is to return whichever of the possible eigenvectors yields the smallest backward error for the polynomial problem.

For a general purpose algorithm, we have the potential for zero, infinite, or finite eigenvalues. In this case, working with the homogeneous representation of an eigenvalue as $\lambda = \alpha/\beta$, the forms of the left and right eigenvectors split into different cases depending on α and β, rather than a single form for the eigenvector. For example, given the quadratic $Q(\alpha, \beta) = \alpha^2 A_2 + \alpha\beta A_1 + \beta^2 A_0$ with eigenvalue $\lambda = \alpha/\beta$ and left and right eigenvectors x and y, the second companion form $C_2(\alpha, \beta)$ of Q in homogenous form is $C_2(\alpha, \beta) = \alpha \begin{bmatrix} A_2 & 0 \\ 0 & I_n \end{bmatrix} \beta \begin{bmatrix} A_1 & -I_n \\ A_0 & 0 \end{bmatrix}$. The left and right eigenvectors (w and z) of $C_2(\alpha, \beta)$ have the form

$$\lambda = \alpha/\beta, \quad (\alpha, \beta \neq 0), \quad w = \begin{bmatrix} \alpha y \\ \beta y \end{bmatrix} \qquad z = \begin{bmatrix} \alpha x \\ -\beta A_0 x \end{bmatrix},$$

$$\lambda = 0, \quad (\alpha = 0, \quad \beta \neq 0), \quad w = \begin{bmatrix} 0 \\ y \end{bmatrix} \qquad z = \begin{bmatrix} x \\ A_1 x \end{bmatrix},$$

$$\lambda = \infty, \quad (\alpha \neq 0, \quad \beta = 0), \quad w = \begin{bmatrix} y \\ 0 \end{bmatrix} \qquad z = \begin{bmatrix} x \\ 0 \end{bmatrix}.$$

Chapter 3

A General Purpose Algorithm for Solving Quadratic Eigenvalue Problems

3.1 Introduction

Quadratic eigenvalue problems (QEPs) arise in a wide variety of science and engineering applications, such as the dynamic analysis of mechanical systems, where the eigenvalues represent vibrational frequencies. For many practical examples of QEPs, see the NLEVP collection [7] and the survey article [58].

The QEP is to find scalars λ and nonzero vectors x, y satisfying

$$\mathcal{Q}(\lambda)x = 0, \qquad y^*\mathcal{Q}(\lambda) = 0, \tag{3.1.1}$$

where

$$\mathcal{Q}(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0, \tag{3.1.2}$$

the A_j, $j = 0 : 2$ are $n \times n$ matrices and x, y are the right and left eigenvectors,

respectively, corresponding to the eigenvalue λ.

QEPs are an important class of nonlinear eigenproblems that are less routinely solved than the standard eigenvalue problem $(\mathcal{A} - \lambda I)x = 0$ or generalized eigenvalue problem $(\mathcal{A} - \lambda\mathcal{B})x = 0$. Quadratic, and more generally, polynomial eigenvalue problems are usually converted to a degree one problem of larger dimension—the process of linearization. For example the pencil

$$\mathcal{A} - \lambda\mathcal{B} = \begin{bmatrix} A_0 & 0 \\ 0 & I \end{bmatrix} - \lambda \begin{bmatrix} -A_1 & -A_2 \\ I & 0 \end{bmatrix} \tag{3.1.3}$$

has the same eigenvalues as (4.1.1) with right eigenvectors of the form $z = \left[\begin{smallmatrix} x \\ \lambda x \end{smallmatrix}\right]$ for finite eigenvalues. This is the pencil used by the MATLAB function `polyeig` for quadratics of the form (3.1.2). This conversion to linear form allows standard numerical methods (e.g., the QZ algorithm [51] or Krylov subspace methods for large sparse problems) to be applied. In doing so however, it is important to understand the influence of the linearization process on the accuracy and stability of the computed solution. Indeed Tisseur showed that solving the QEP by applying a backward stable algorithm (e.g. the QZ algorithm) to a linearization can be backward unstable [56]. Also, unless the block structure of the linearization is respected (and it is not by standard techniques), the conditioning of the solutions of the larger linear problem can be worse than those for the original quadratic (4.1.1), since the class of admissible perturbations is larger. For example, eigenvalues that are well conditioned for problem (4.1.1) may then be ill conditioned for linearizations [32], [33]. For these reasons, the numerical solution of QEPs requires special attention.

In a number of applications, such as structural mechanics [15], constrained multibody systems [9], 3D computer vision problems [39], vibration of railtracks [45], either, or both, of the leading A_2 or trailing A_0 coefficients are singular.

When both A_0 and A_2 are singular, the quadratic $\mathcal{Q}(\lambda)$ may be nonregular (i.e., $\det(\mathcal{Q}(\lambda)) \equiv 0$). In this case the QZ algorithm when applied to a linearization of $\mathcal{Q}$ may deliver meaningless results. Regular quadratics (i.e., $\det(\mathcal{Q}(\lambda)) \not\equiv 0$) with singular A_0 and/or A_2 have zero and/or infinite eigenvalues. Theoretically, the QZ algorithm handles infinite eigenvalues well [60]. However, experiments of Kågström and Kressner [36] show that if infinite eigenvalues are not extracted before starting the QZ steps, they may never be detected due to the effect of rounding errors in floating point arithmetic.

In one quadratic eigenvalue problem occurring in the vibration analysis of rail tracks under excitation arising from high speed trains [35], [45, p.18], the deflation of zero and infinite eigenvalues had a significant impact on the quality of the remaining computed finite eigenvalues.

In this work we present a general purpose eigensolver for dense QEPs, which incorporates recent contributions on the numerical solution of polynomial eigenvalue problems, namely a scaling of the eigenvalue parameter prior to the computation, [6], [14] and a choice of linearization with favourable conditioning and backward stability properties [30], [32], [33]. Our algorithm includes a preprocessing step that reveals the zero and infinite eigenvalues contributed by singular leading and trailing matrix coefficients and deflates them. The preprocessing step may also detect nonregularity (although this is not guaranteed). Our algorithm takes advantage of the block structure of the chosen linearization. We have implemented it as a MATLAB [49] function called `quadeig`, which makes use of functions from the NAG Toolbox for MATLAB [53]. Our eigensolver can in principle be extended to matrix polynomials of degree higher than two. The preprocessing step can easily be extended using the same type of linearization, merely of a higher degree matrix polynomial. For scaling of the eigenvalue parameter prior to the computation we can use the method described in Section 2.3.1 on page 44 [6], which extends the

Fan, Lin and Van Dooren scaling for matrix polynomials of degree two.

In this chapter we write $\mathcal{Q}$ to represent (in addition to matrices $\mathcal{A}, \mathcal{B}$) the quadratic matrix polynomial $\mathcal{Q}(\lambda)$ (that was previously written as $Q(\lambda)$), so we can use Q to represent a matrix transformation, for example from a QR factorization. We also write a matrix pencil as $A - \lambda B$ rather than $\lambda A + B$.

3.2 Choice of Linearization

The definition of a linearization, $\mathcal{L}(\lambda) = \mathcal{A} - \lambda\mathcal{B}$ is a of a quadratic $\mathcal{Q}(\lambda)$ was given earlier in Definition .

For a given quadratic $\mathcal{Q}$, there are an infinite number of linearizations (the pencil (3.1.3) is just one example). These linearizations can have widely varying eigenvalue condition numbers [32], and approximate eigenpairs of $\mathcal{Q}(\lambda)$ computed via linearization can have widely varying backward errors [30]. In the following subsection we define the terms backward error and condition number more precisely focusing on the particular linearization that our algorithm will employ.

3.2.1 Backward Error and Condition Number

Definitions of backward error and condition number for quadratics and linearizations are contained in Section 1.5.3, we recall only the special case for quadratics.

Explicit expressions for backward errors for $\mathcal{Q}$ and $\mathcal{L}$ are given by [30]:

$$\eta_{\mathcal{Q}}(x, \alpha, \beta) = \frac{\|\mathcal{Q}(\alpha, \beta)x\|_2}{\left(\sum_{i=0}^{2} |\alpha|^i |\beta|^{2-i} \|A_i\|_2\right) \|x\|_2}, \quad \eta_{\mathcal{L}}(z, \alpha, \beta) = \frac{\|\mathcal{L}(\alpha, \beta)z\|_2}{(|\beta|\|\mathcal{A}\|_2 + |\alpha|\|\mathcal{B}\|_2) \|z\|_2}, \tag{3.2.1}$$

The definitions and explicit expressions for the backward error $\eta_{\mathcal{Q}}(y^*, \alpha, \beta)$ and $\eta_{\mathcal{L}}(w^*, \alpha, \beta)$ of a left approximate eigenpair (y^*, α, β) and (w^*, α, β) of $\mathcal{Q}$ and $\mathcal{L}$ are analogous to those for right eigenpairs.

The eigenvalue condition number $\kappa_{\mathcal{L}}(\alpha,\beta)$ for the pencil $\mathcal{L}(\alpha,\beta) = \beta\mathcal{A} - \alpha\mathcal{B}$ is obtained by a trivial extension of a result of Dedieu and Tisseur [13, Thm. 4.2] that treats the unweighted Frobenius norm, this yields the explicit formula

$$\kappa_{\mathcal{L}}(\alpha,\beta) = \sqrt{|\beta|^2\|\mathcal{A}\|_2^2 + |\alpha|^2\|\mathcal{B}\|_2^2}\ \frac{\|w\|_2\|z\|_2}{\left|w^*(\bar{\beta}\,\mathcal{D}_\alpha\mathcal{L} - \bar{\alpha}\,\mathcal{D}_\beta\mathcal{L})|_{(\alpha,\beta)}z\right|}, \tag{3.2.2}$$

where $\mathcal{D}_\alpha \equiv \frac{\partial}{\partial\alpha}$ and $\mathcal{D}_\beta \equiv \frac{\partial}{\partial\beta}$, where z, w are right and left eigenvectors of $\mathcal{L}$ associated with (α,β). Note that the denominators of the expressions (3.2.2) is nonzero for simple eigenvalues. Also, these expressions are independent of the choice of representative of (α,β) and of the scaling of the eigenvectors. Let (α,β) and $(\tilde{\alpha},\tilde{\beta})$ be the original and perturbed simple eigenvalues, normalized such that $\|(\alpha,\beta)\|_2 = 1$ and $(\alpha,\beta)(\tilde{\alpha},\tilde{\beta})^* = 1$. Then the angle between the original and perturbed eigenvalues satisfies

$$\left|\theta\big((\alpha,\beta),(\tilde{\alpha},\tilde{\beta})\big)\right| \le \kappa_{\mathcal{Q}}(\alpha,\beta)\|\Delta A\| + o(\|\Delta A\|). \tag{3.2.3}$$

Note that $\|\Delta A\| \approx \eta_{\mathcal{Q}}(\tilde{\alpha},\tilde{\beta}) := \min_{x\neq 0}\eta_{\mathcal{Q}}(x,\tilde{\alpha},\tilde{\beta}) = \min_{y\neq 0}\eta_{\mathcal{Q}}(y^*,\tilde{\alpha},\tilde{\beta})$. Hence the product of the condition number (1.5.4) with the backward error (3.2.1) provides an approximate upper bound on the angle between the original and computed eigenvalues. The condition numbers and backward errors are optionally returned by our algorithm.

We want to use a linearization $\mathcal{L}$ that is as well conditioned as the original quadratic $\mathcal{Q}$ and for it to lead, after recovering approximate left and right eigenvectors of $\mathcal{Q}$ from those of $\mathcal{L}$, say w and z, to a backward error of the same order of magnitude as that for $\mathcal{L}$, that is, we would like

$$\kappa_{\mathcal{Q}}(\alpha,\beta) \approx \kappa_{\mathcal{L}}(\alpha,\beta), \quad \eta_{\mathcal{Q}}(x,\alpha,\beta) \approx \eta_{\mathcal{L}}(z,\alpha,\beta), \quad \eta_{\mathcal{Q}}(y^*,\alpha,\beta) \approx \eta_{\mathcal{L}}(w^*,\alpha,\beta)$$

for all eigenvalues (α, β).

3.2.2 Companion Linearizations

Companion linearizations are the most commonly used linearizations in practice. Several forms exist. The first and second companion linearization of $\mathcal{Q}$ (defined earlier in Table 2.1) are given by

$$C_1(\lambda) = \begin{bmatrix} A_1 & A_0 \\ -I_n & 0 \end{bmatrix} - \lambda \begin{bmatrix} -A_2 & 0 \\ 0 & -I_n \end{bmatrix}, \qquad C_2(\lambda) = \begin{bmatrix} A_1 & -I_n \\ A_0 & 0 \end{bmatrix} - \lambda \begin{bmatrix} -A_2 & 0 \\ 0 & -I_n \end{bmatrix}. \tag{3.2.4}$$

Note that $C_2(\lambda)$ is the block transpose of $C_1(\lambda)$. Other companion forms can be obtained, for example, by taking the reversal of the first or second companion form of $\mathrm{rev}(\mathcal{Q})$,

$$C_3(\lambda) = \begin{bmatrix} A_0 & 0 \\ 0 & I_n \end{bmatrix} - \lambda \begin{bmatrix} -A_1 & -A_2 \\ I & 0 \end{bmatrix}, \qquad C_4(\lambda) = \begin{bmatrix} A_0 & 0 \\ 0 & I \end{bmatrix} - \lambda \begin{bmatrix} -A_1 & I_n \\ -A_2 & 0 \end{bmatrix}$$

or simply by swapping the block rows or block columns of these linearizations.

Companion linearizations have a number of desirable properties:

(a) They are always linearizations even if $\mathcal{Q}(\lambda)$ is nonregular. Moreover they are strong linearizations [42]: they preserve the partial multiplicities of infinite eigenvalues.

(b) The left and right eigenvectors of $\mathcal{Q}(\lambda)$ are easily recovered from those of the companion form ([23], [30] and (3.2.5) for C_2).

(c) If the quadratic is well scaled (i.e., $\|A_i\|_2 \approx 1$, $i = 0\colon 2$), companion linearizations have good conditioning and backward stability properties (see below).

Amongst companion linearizations $C_i(\lambda) = \mathcal{A}_i - \lambda\mathcal{B}_i$ we are looking for one for which

(d) the $\mathcal{A}_i$ matrix is in block upper triangular form, thereby reducing the computational cost of the Hessenberg-triangular reduction step of the QZ algorithm,

(e) the linearization can easily be transformed to a block upper triangular form revealing zero and infinite eigenvalues, if any.

The first and second companion linearizations in (3.2.4) satisfy desideratum (d) and we will show in Section 3.3 that in the presence of singular leading and trailing matrix coefficients, desideratum (e) can easily be achieved for the second companion linearization. Hence our eigensolver will use $C_2(\lambda)$.

Concerning property (b), the second companion form $C_2(\lambda)$ in (3.2.4) has right eigenvectors z and left eigenvectors w of the form

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{cases} \begin{bmatrix} \alpha x \\ -\beta A_0 x \end{bmatrix} & \text{if } \alpha \neq 0, \\ \begin{bmatrix} \beta x \\ \beta A_1 x \end{bmatrix} & \text{if } \alpha = 0, \end{cases} \qquad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \alpha y \\ \beta y \end{bmatrix}, \tag{3.2.5}$$

where x, y are right and left eigenvectors of $\mathcal{Q}(\lambda)$ with eigenvalue $\lambda = \alpha/\beta$. The formulae in (3.2.5) show that x can be recovered from the first n entries of z or by solving $A_0 x = z_2$, whereas y can be recovered from either the n first entries or the last n entries of w.

The experiments in [30] and [32] show that for the first companion linearization in (3.2.4),

$$\kappa_{\mathcal{Q}}(\alpha,\beta) \ll \kappa_{C_1}(\alpha,\beta), \quad \eta_{\mathcal{Q}}(x,\alpha,\beta) \gg \eta_{C_1}(z,\alpha,\beta), \quad \eta_{\mathcal{Q}}(y^*,\alpha,\beta) \gg \eta_{C_1}(w^*,\alpha,\beta) \tag{3.2.6}$$

can happen when the coefficient matrices A_i, $i = 0\colon 2$ vary largely in norm, (3.2.6) also holds for the second companion linearization C_2. The scaling of Fan, Lin, and Van Dooren [14] tries to bring the 2-norms of A_0, A_1, and A_2 close to 1, in order to overcome this problem. It converts $\mathcal{Q}(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ to $\widetilde{\mathcal{Q}}(\mu) = \mu^2 \widetilde{A}_2 + \mu \widetilde{A}_1 + \widetilde{A}_0$, where

$$\lambda = \gamma\mu, \quad Q(\lambda)\delta = \mu^2(\gamma^2\delta A_2) + \mu(\gamma\delta A_1) + \delta A_0 \equiv \widetilde{\mathcal{Q}}(\mu), \tag{3.2.7}$$

$$\gamma = \sqrt{A_0/A_2}, \quad \delta = 2/(A_0 + A_1\gamma). \tag{3.2.8}$$

Note that $\eta_{\mathcal{Q}}(x, \alpha, \beta) = \eta_{\widetilde{\mathcal{Q}}}(x, \tilde{\alpha}, \tilde{\beta})$, where $\mu = \tilde{\alpha}/\tilde{\beta}$, so this scaling has no effect on the backward error for the quadratic, however $\kappa_{\mathcal{Q}}(\alpha, \beta)$ is scale-dependent.

Let (z, w, α, β) be an approximate eigentriple of the second companion linearization C_2 in (3.2.4) of the scaled quadratic $\widetilde{\mathcal{Q}}$ with $|\alpha|^2 + |\beta|^2 = 1$. Define

$$\omega = \omega(\alpha, \beta) := \frac{1 + \tau}{1 + |\alpha\beta|\tau}, \qquad \tau = \frac{\|A_1\|_2}{\sqrt{\|A_2\|_2 \, \|A_0\|_2}}. \tag{3.2.9}$$

Using the framework developed in [23], [30] we can show that

$$\frac{1}{\sqrt{2}} \leq \frac{\eta_{\widetilde{\mathcal{Q}}}(w_i^*, \alpha, \beta)}{\eta_{C_2}(w^*, \alpha, \beta)} \leq 2^{7/2}\omega \, \frac{\|w\|_2}{\|w_i\|_2}, \quad i = 1, 2, \tag{3.2.10}$$

$$\frac{1}{\sqrt{2}} \leq \frac{\eta_{\widetilde{\mathcal{Q}}}(z_1, \alpha, \beta)}{\eta_{C_2}(z, \alpha, \beta)} \leq 2^{3/2}\omega \, \frac{\|z\|_2}{\|z_1\|_2} \tag{3.2.11}$$

$$\frac{1}{2\sqrt{2}} \leq \frac{\kappa_{C_2}(\alpha, \beta)}{\kappa_{\widetilde{\mathcal{Q}}}(\alpha, \beta)} \leq 4\sqrt{3}\,\omega, \tag{3.2.12}$$

where $w_1 = w(1\colon n)$, $w_2 = w(n+1\colon 2n)$ and $z_1 = z(1\colon n)$. In interpreting these bounds recall that, for an exact left eigenvector of $C_2(\lambda)$,

$$\frac{\|w\|_2}{\|w_1\|_2} \approx 1 \quad \text{for } |\alpha| \geq |\beta|, \qquad \frac{\|w\|_2}{\|w_2\|_2} \approx 1 \quad \text{for } |\alpha| \leq |\beta| \tag{3.2.13}$$

and that for an exact right eigenvector z of $C_2(\lambda)$, $\|z\|_2/\|z_1\|_2 \approx 1$.

Hence (3.2.10)–(3.2.11) show that if $\omega = O(1)$ then $\eta_{\widetilde{\mathcal{Q}}} \approx \eta_{C_2}$ for both left and right eigenpairs. It is shown in [30] that

$$1 \le \frac{1+\tau}{1+\frac{1}{2}\tau} \le \omega \le \min\left\{1+\tau, \frac{1}{|\alpha\beta|}\right\} \le 1+\tau. \tag{3.2.14}$$

Hence, $\omega = O(1)$ if $\tau \ll 1$, or equivalently, $\|A_1\|_2 \lesssim (\|A_2\|_2\|A_0\|_2)^{1/2}$, which in the terminology of damped mechanical systems means that the problem is not too heavily damped. When $\tau \gg 1$ the penultimate inequality in (3.2.14) will still be of order 1 if $|\alpha||\beta| = |\alpha|\sqrt{1-|\alpha|^2} = O(1)$, which is the case unless $|\lambda| = |\alpha|/|\beta| = |\alpha|/\sqrt{1-|\alpha|^2}$ is small or large.

This analysis and the numerical experiments in Section 3.6 suggest applying the scaling of Fan, Lin, and Van Dooren to the original quadratic $\mathcal{Q}(\lambda)$ prior to building the second companion linearization $C_2(\lambda)$. For quadratics that are not too heavily damped, the bounds in (3.2.10)–(3.2.11) guarantee that if the eigenpairs of $C_2(\lambda)$ are computed with a small backward error (this is the case if we use the QZ algorithm) then we can recover eigenpairs for $\mathcal{Q}(\lambda)$ with a small backward error.

Algorithm 2 (Fan, Lin and Van Dooren scaling [14]). *Given $n \times n$ matrices A_2, A_1, A_0, this algorithm overwrites A_2, A_1, A_0 with scaled matrices, attempting to achieve $\|A_2\|_F \approx \|A_1\|_F \approx \|A_0\|_F \approx 1$ and returns a scalar γ such that if λ, μ are eigenvalues of the unscaled and scaled quadratic then $\lambda = \mu\gamma$. No scaling is performed when $\|A_0\|_F = 0$ or $\|A_2\|_F = 0$.*

$\gamma = 1$, $g_2 = \|A_2\|_F$, $g_0 = \|A_0\|_F$
if $g_0 \neq 0$ and $g_2 \neq 0$
  $g_1 = \|A_1\|_F$
  $\gamma = \sqrt{g_0/g_2}$
  $\delta = 2/(g_0 + g_1\gamma)$

$$A_2 = \gamma^2\delta A_2,\ A_1 = \gamma\delta A_1,\ A_0 = \delta A_0$$

end

3.3 Deflation of 0 and ∞ Eigenvalues

The eigenvalues of a regular $n \times n$ quadratic $\mathcal{Q}(\lambda)$ are the zeros of the characteristic polynomial $\det\big(\mathcal{Q}(\lambda)\big) = \det(A_2)\lambda^{2n} +$ lower order terms, so when A_2 is nonsingular, $\mathcal{Q}(\lambda)$ has $2n$ finite eigenvalues. When A_2 is singular $\mathcal{Q}(\lambda)$ has d finite eigenvalues to which we add $2n - d$ infinite eigenvalues, where d is the degree of $\det(\mathcal{Q}(\lambda))$.

Recall that λ is an eigenvalue of $\mathcal{Q}$ if and only if $1/\lambda$ is an eigenvalue of the reversal of $\mathcal{Q}$,

$$\mathrm{rev}(\mathcal{Q}(\lambda)) := \lambda^2 A_0 + \lambda A_1 + A_2$$

where 0 and ∞ are regarded as reciprocals. If $r_0 = \mathrm{rank}(A_0) < n$ then $\mathcal{Q}$ has at least $n - r_0$ zero eigenvalues and if $r_2 = \mathrm{rank}(A_2) < n$, $\mathcal{Q}$ has at least $n - r_2$ infinite eigenvalues.

As an example, the quadratic

$$\mathcal{Q}(\lambda) = \lambda^2 \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \lambda A_1 + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

with A_1 such that $\det(\mathcal{Q}(\lambda)) \not\equiv 0$ has at least one infinite eigenvalue and at least one zero eigenvalue. If $A_1 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ then the remaining eigenvalues are ∞ and -1.

Let us denote by $\mathcal{N}_r(A) = \big\{x \in \mathbb{C}^n : Ax = 0\big\}$ and $\mathcal{N}_l(A) = \big\{y \in \mathbb{C}^n : y^*A = 0\big\}$ the right and left nullspace, respectively of $A \in \mathbb{C}^{n\times n}$. Note that the right and left eigenvectors of $\mathcal{Q}$ associated with the 0 and ∞ eigenvalues generate the right and

left nullspace of A_0 and A_2, respectively.

Our algorithm checks the rank of A_0 and A_2; when one or both of them are singular, it deflates the corresponding zero and infinite eigenvalues. In the next section we describe and justify how our algorithm checks the rank of the leading and trailing coefficients using a matrix factorization, and how a basis for the nullspace can be obtained from the factorization.

3.3.1 Rank and Nullspace Determination

A QR factorization with column pivoting (see Definition 6) can be used to determine the numerical rank of an $n \times n$ matrix A.

For sufficiently small $\|E\|_2$, it is shown in [28, Thm. 5.2] that $A + E$ has the QR factorization with column pivoting

$$\bar{Q}^*(A+E)P = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} \bar{R}_{11} & \bar{R}_{12} \\ 0 & \bar{R}_{22} \end{bmatrix} \end{array}, \tag{3.3.1}$$

with

$$\frac{\|\bar{R}_{22}\|_2}{\|A\|_2} \leq \frac{\|E\|_2}{\|A\|_2}(1 + \|R_{11}^{-1}R_{12}\|_2) + O\left(\frac{\|E\|_2}{\|A\|_2}\right)^2. \tag{3.3.2}$$

The quantity $\|R_{11}^{-1}R_{12}\|_2$ can be arbitrarily large and (3.3.2) shows that even if $\|E\|_2$ is small, $\|\bar{R}_{22}\|_2$ can be much larger than the distance $\sigma_{k+1}(A+E) \leqslant \|E\|_2$ from $A + E$ to the rank k matrices. Empirical observations show however, that $\|R_{11}^{-1}R_{12}\|_2$ is usually small. Hence if $A + E$ is close to a rank k matrix then $\bar{R}_{22}$ will be small. Our algorithm sets $\bar{R}_{22}$ to zero if $\|\bar{R}_{22}\|_2 \leq nu\|A\|_2$, where u is the unit roundoff. This test can yield a numerical rank that is an overestimate of the rank but this does not affect the stability of our algorithm. Indeed we only deflate zero and infinite eigenvalues using QR factorizations with column pivoting.

Overestimating the rank results in deflating fewer eigenvalues than we could have done, had the rank been computed correctly. The QZ algorithm then has to solve a generalized eigenproblem of larger dimension.

The last $n-k$ columns of Q in (6) span the left null space of A. A basis for the right nullspace of A is obtained by postmultiplying (6) by a sequence of Householder transformations $H_1, \ldots, H_k$ that reduce R_{12} to zero. This leads to a complete orthogonal decomposition of A,

$$Q^*AZ = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} T_{11} & 0 \\ 0 & 0 \end{bmatrix} \end{array}, \tag{3.3.3}$$

where $Z = PH_1 \cdots H_r$ and Q and P are as in (6) (see [21, p. 250]). Then the last $n-k$ columns of PH span the right nullspace of A.

The LAPACK routine `xGEQP3` computes (6). In floating point arithmetic, however, `xGEQP3` computes

$$fl(Q^*AP) = \begin{array}{c} \\ r \\ n-r \end{array} \begin{array}{c} \begin{array}{cc} r & n-r \end{array} \\ \begin{bmatrix} \widehat{R}_{11} & \widehat{R}_{12} \\ 0 & \widehat{R}_{22} \end{bmatrix} \end{array}. \tag{3.3.4}$$

because of rounding errors. We set $\widehat{R}_{22}$ to zero if $\|\widehat{R}_{22}\|_2 \leq nu\|A\|_2$, where u is the unit roundoff and call r in (3.3.4) the numerical rank of A. Once (3.3.4) is computed and $\widehat{R}_{22}$ set to zero, the LAPACK routine `xTZRZF` can be used to eliminate the R_{12} block to yield a complete orthogonal decomposition.

3.3.2 Block Triangularization of $C_2(\lambda)$

Throughout this section we assume that $r_0 := \mathrm{rank}(A_0) \geq r_2 =: \mathrm{rank}(A_2)$ (if $r_0 < r_2$ we work with $\mathrm{rev}(\mathcal{Q}(\lambda))$ instead of $\mathcal{Q}(\lambda)$ and swap the factorizations (3.3.5)). Let

$$Q_i^* A_i P_i = \begin{array}{c} \\ r_0 \\ n-r_0 \end{array} \begin{array}{c} \begin{array}{cc} r_i & n-r_i \end{array} \\ \begin{bmatrix} R_{11}^{(i)} & R_{12}^{(i)} \\ 0 & 0 \end{bmatrix} \end{array} = \begin{bmatrix} R^{(i)} \\ 0 \end{bmatrix}, \quad i = 0, 2, \tag{3.3.5}$$

be QR factorizations with column pivoting of A_0 and A_2. With the help of these factorizations and another complete orthogonal decomposition when both A_0 and A_2 are singular (i.e., $r_0, r_2 < n$), we show how to transform the second companion form $C_2(\lambda) = \left[\begin{smallmatrix} A_1 & -I \\ A_0 & 0 \end{smallmatrix}\right] - \lambda \left[\begin{smallmatrix} -A_2 & 0 \\ 0 & -I \end{smallmatrix}\right]$ in (3.2.4) into block upper triangular form

$$QC_2(\lambda)V = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & 0 & 0_{n-r_0} \end{bmatrix} - \lambda \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ 0 & 0_{n-r_2} & B_{23} \\ 0 & 0 & I_{n-r_0} \end{bmatrix}, \tag{3.3.6}$$

where the $2n \times 2n$ matrices are partitioned conformably. When A_{22} is singular then $\det(\mathcal{Q}(\lambda)) = \det(C_2(\lambda)) \equiv 0$ and hence $\mathcal{Q}(\lambda)$ is nonregular. When A_{22} is nonsingular, (3.3.6) reveals $n - r_0$ zero eigenvalues and $n - r_2$ infinite eigenvalues. The remaining eigenvalues are those of the $(r_1 + r_2) \times (r_1 + r_2)$ pencil $A_{11} - \lambda B_{11}$.

We consider three cases.

(i) $r_0 = r_2 = n$. In this case there are no zero or infinite eigenvalues. We make use of the factorization of A_2 in (3.3.5), however, to reduce the leading coefficient $\left[\begin{smallmatrix} A_2 & 0 \\ 0 & I \end{smallmatrix}\right]$ of the linearization to upper triangular form, a necessary

step in the QZ algorithm. This is achieved with

$$Q = \begin{bmatrix} Q_2^* & 0 \\ 0 & I_n \end{bmatrix}, \quad V = \begin{bmatrix} P_2 & 0 \\ 0 & I_n \end{bmatrix}$$

so that

$$QC_2(\lambda)V = \begin{bmatrix} Q_2^* A_1 P_2 & -Q_2^* \\ A_0 P_2 & 0 \end{bmatrix} - \lambda \begin{bmatrix} R_2 & 0 \\ 0 & I \end{bmatrix} = \mathcal{A}_{11} - \lambda \mathcal{B}_{11}.$$

(ii) $r_0 < r_2 = n$. In this case there are at least $n - r_0$ zero eigenvalues, which we deflate with

$$Q = \begin{bmatrix} Q_2^* & 0 \\ 0 & Q_0^* \end{bmatrix}, \qquad V = \begin{bmatrix} P_2 & 0 \\ 0 & Q_0 \end{bmatrix}$$

so that

$$Q\,C_2(\lambda)V = \begin{array}{c} \\ n \\ r_0 \\ n-r_0 \end{array} \begin{array}{c} \begin{array}{ccc} n & r_0 & n-r_0 \end{array} \\ \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} - \lambda \begin{bmatrix} R^{(2)} & 0 & 0 \\ 0 & I_{r_0} & 0 \\ 0 & 0 & I_{n-r_0} \end{bmatrix} \tag{3.3.7}$$

where $X_{11} = Q_2^* A_1 P_2$, $[X_{12},\ X_{13}] = -Q_2^* Q_0$ and $X_{21} = R^{(0)} P_0^* P_2$. The pencil (3.3.7) is in the form (3.3.6) with $A_{11} = \left[\begin{smallmatrix} X_{11} & X_{12} \\ X_{21} & 0 \end{smallmatrix}\right]$ and $B_{11} = \left[\begin{smallmatrix} -R^{(2)} & 0 \\ 0 & -I_{r_0} \end{smallmatrix}\right]$ of dimension $(n + r_0) \times (n + r_0)$. As in case (i), B_{11} is upper triangular.

(iii) $r_0 \le r_2 < n$. There are at least $n - r_0$ zero eigenvalues and at least $n - r_2$ infinite eigenvalues that we deflate as follows. With

$$\widetilde{Q} = \begin{bmatrix} Q_2^* & 0 \\ 0 & Q_0^* \end{bmatrix}, \qquad \widetilde{V} = \begin{bmatrix} I_n & 0 \\ 0 & Q_0 \end{bmatrix}$$

we obtain

$$\widetilde{Q}C_2(\lambda)\widetilde{V} = \begin{array}{c} \\ r_2 \\ n-r_2 \\ r_0 \\ n-r_0 \end{array} \begin{array}{c} \begin{array}{cccc} r_2 & n-r_2 & r_0 & n-r_0 \end{array} \\ \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} - \lambda \begin{bmatrix} Y_{11} & Y_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I_{r_0} & 0 \\ 0 & 0 & 0 & -I_{n-r_0} \end{bmatrix}, \tag{3.3.8}$$

where

$$[X_{31},\ X_{32}] = R^{(0)}P_0, \quad \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = Q_2^* A_1, \quad \begin{bmatrix} X_{13} & X_{14} \\ X_{23} & X_{24} \end{bmatrix} = -Q_2^* Q_0,$$

and $[Y_{11},\ Y_{12}] = -R^{(2)}P_2$. Let

$$\begin{array}{c} \\ n-r_2 \end{array} \begin{array}{c} \begin{array}{ccc} r_2 & n-r_2 & r_0 \end{array} \\ \begin{bmatrix} X_{21} & X_{22} & X_{23} \end{bmatrix} \end{array} = Q_3 \begin{array}{c} \begin{array}{cc} n-r_2 & r_0+r_2 \end{array} \\ \begin{bmatrix} R_3 & 0 \end{bmatrix} \end{array} Z_3$$

be a complete orthogonal decomposition and let

$$Q = \begin{bmatrix} I_{r_2} & 0 & 0 & 0 \\ 0 & 0 & I_{r_0} & 0 \\ 0 & Q_3^* & 0 & 0 \\ 0 & 0 & 0 & I_{n-r_0} \end{bmatrix} \widetilde{Q}, \quad V = \widetilde{V} \begin{array}{c} \begin{array}{cc} n+r_0 & n-r_0 \end{array} \\ \begin{bmatrix} Z_3^* & 0 \\ 0 & I_{n-r_0} \end{bmatrix} \end{array} \begin{bmatrix} 0 & I_{n-r_2} & 0 \\ I_{r_2+r_0} & 0 & 0 \\ 0 & 0 & I_{n-r_0} \end{bmatrix}.$$

Then easy calculations show that $Q\,C_2(\lambda)V$ has the form (3.3.6) with $A_{22} = R_3$.

3.4 Left and Right Eigenvectors

The computation of the left and right eigenvectors differ so we consider them separately.

3.4.1 Right Eigenvectors

When either or both of A_0 and A_2 are singular, the vectors spanning their right nullspaces $\mathcal{N}_r(A_0)$ and $\mathcal{N}_r(A_2)$ are right eigenvectors associated with the 0 and ∞ eigenvalues of $\mathcal{Q}(\lambda)$. These nullspaces can be obtained from (3.3.5) by zeroing $R_{12}^{(i)}$, $i = 0, 2$ to obtain a complete orthogonal decomposition as in (3.3.3), that is,

$$Q_j^* A_j Z_j = \begin{array}{c} \\ r_j \\ n-r_j \end{array} \begin{array}{c} \begin{array}{cc} r_j & n-r_j \end{array} \\ \begin{bmatrix} T_{11}^j & 0 \\ 0 & 0 \end{bmatrix} \end{array}.$$

The last $n - r_0$ columns of Z_0 are eigenvectors of $\mathcal{Q}$ with eigenvalue 0 and the last $n - r_2$ columns of Z_2 are eigenvectors of $\mathcal{Q}$ with eigenvalue ∞.

The eigenvectors associated with the remaining eigenvalues are recovered from those of the linearization $C_2(\lambda)$. If $\widetilde{z}$ is a right eigenvector of the $(r_0+r_2)\times(r_0+r_2)$ pencil $A_{11} - \lambda B_{11}$ in (3.3.6) then

$$z = \begin{array}{c} \\ n \\ n \end{array} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = V \begin{bmatrix} \widetilde{z} \\ 0 \end{bmatrix}$$

is a right eigenvector of $C_2(\lambda)$. We also know that z must have the form displayed in (3.2.5). However, in floating point arithmetic when A_0 is nonsingular and for

finite nonzero eigenvalues,

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \alpha x_1 \\ -\beta A_0 x_2 \end{bmatrix}.$$

We can use the QR factorization of A_0 in (3.3.5) to solve the linear system $z_2 = -\beta A_0 x_2$ for x_2 efficiently. We return as approximate eigenvector of $\mathcal{Q}$ corresponding to the eigenvalue $\lambda = \alpha/\beta$ the vector x_i, $i = 1, 2$ which minimizes $\eta_{\mathcal{Q}}(x_i, \alpha, \beta)$.

Given two approximate eigenpairs (λ, x_1) and (λ, x_2), with backward errors $\eta_Q(\lambda, x_1)$ and $\eta_Q(\lambda, x_2)$, if neither of x_1 nor x_2 yields an acceptable backward error, we could attempt to obtain an improvement by determining a linear combination $x_3 = a_1 x_1 + a_2 x_2$ that solves

$$\min_{a \in \mathbb{C}^2} \frac{\|\mathcal{Q}(\lambda) X a\|_2}{\|X a\|_2},$$

where $X = [x_1,\ x_2]$. For that we take the GSVD [2, pp. 257–259] of the pair of $n \times 2$ matrices $(\mathcal{Q}(\lambda)X, X)$,

$$\mathcal{Q}(\lambda)X = UCY^{-1}, \qquad X = VSY^{-1},$$

where U, V are unitary, Y is nonsingular and $C = \mathrm{diag}(c_1, c_2)$, $S = \mathrm{diag}(s_1, s_2)$ with $c_1, c_2, s_1, s_2 \geq 0$. Thus if we let $a = Yb$,

$$\min_{a \in \mathbb{C}^2} \frac{\|\mathcal{Q}(\lambda) X a\|_2^2}{\|X a\|_2^2} = \min_{b \in \mathbb{C}^2} \frac{\|Cb\|_2^2}{\|Sb\|_2^2} = \min_{b \in \mathbb{C}^2} \frac{b^* C^* C b}{b^* S^* S b}$$

which is the smallest eigenvalue of $C^*C - \lambda S^*S$. So the minimum is achieved at $b = e_i$, where $|c_i/s_i|$ is minimal. Hence $a = Ye_i$.

In finite precision, however, there can be a problem if either one or both of

$\|\mathcal{Q}(\lambda)X\|_2$ or $\|X\|_2$ are close to the unit roundoff. This can result in an ill-conditioned Y matrix which causes problems in returning an accurate eigenvector when we solve for a_1 and a_2 to yield x_3. In addition, from numerical experiments $\eta_Q(\lambda, x_3)$ is rarely significantly smaller than $\eta_Q(\lambda, x_i),\ i = 1, 2$ (if at all).

3.4.2 Left Eigenvectors

When A_0 is singular, the last $n - r_0$ columns of Q_0 in (3.3.5) are eigenvectors of $\mathcal{Q}$ associated with the $n - r_0$ deflated zero eigenvalues and when A_2 is singular, the last $n - r_2$ columns of Q_2 in (3.3.5) are eigenvectors of $\mathcal{Q}$ associated with the deflated $n - r_2$ infinite eigenvalues.

Let w be a left eigenvector of $C_2(\lambda)$ corresponding to an eigenvalue $\lambda = \alpha/\beta$ of $\mathcal{A}_{11} - \lambda\mathcal{B}_{11}$ in (3.3.6). In exact arithmetic w has the form displayed in (3.2.5) but in floating point arithmetic, w_1 and w_2 are generally not parallel. If $\omega = O(1)$ then (3.2.10) and (3.2.12) predict optimal backward error for w_1 if $|\lambda| \geqslant 1$ and w_2 if $|\lambda| \leqslant 1$. If $\omega \gg O(1)$ then we choose whichever of w_1 or w_2 yields the smallest backward error.

3.5 Algorithm

Algorithm 3 (Quadratic Eigenvalue Solver)**.** *Given three $n\times n$ matrices A_2, A_1, A_0, and a rank tolerance* tol *this algorithm computes a vector E of length $2n$ containing the eigenvalues of $\mathcal{Q}(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$, and optionally, two $n \times 2n$ matrices X and Y containing the corresponding right and left eigenvectors.*

1. Scale A_2, A_1, A_0 using Algorithm 2 (Fan, Lin and Van Dooren scaling).
   Optionally scale using diagonal scaling with scaling parameter ζ.
2. Build block upper triangular form (3.3.6) using Algorithm 3.

If eigenvectors are desired, store transformation matrices Q and V.

3. Compute the Schur decomposition of $A_{11} - \lambda B_{11}$.

To summarize, our eigensolver performs the following steps:

1. Scaling of eigenvalue parameter using Algorithm 2.
2. Rank determination of A_2 and A_0 (see Section 3.3.1).
3. Block triangularization of second companion linearization to achieve (3.3.6).
4. Compute the Schur decomposition of $A_{11} - \lambda B_{11}$.
5. Optionally compute:
   - Right/left eigenvectors.
   - Eigenvalue condition numbers.
   - Backward errors of approximate right/left eigenpairs.

3.6 Numerical Experiments

We now describe a collection of numerical experiments designed to give insight into Algorithm 3, its performance in floating point arithmetic, and the implementation issues. Our computations were done in MATLAB 7.9.0 (R2009b) under Windows XP (SP3) with a Pentium E6850, for which $u = 2^{-53} \approx 1.1 \times 10^{-16}$.

Experiment 1. We ran our algorithm `quadeig` on some quadratic eigenvalue problems from NLEVP [7]. Table 3.1 displays for each problem the largest backward error for the right eigenpairs returned by the MATLAB function `polyeig` and the largest backward errors for the right and left eigenpairs returned by `quadeig`. For this set of problems `quadeig` returns right/left eigenpairs with backward errors close to the machine precision except for the `cd player` and `pdde stability`

problems. For these problems, the large values for the right/left backward errors are predicted by the upper bounds for the growth in backward errors of eigenpairs of the scaled quadratic from those of those of the linearization in (3.2.10)—(3.2.11). The quantity ω forms part of these growth factors and we see that $\omega \approx 10^4$ for the `cd player` problem and $\omega \approx 0.5 \times 10^2$ for the `pdde stability` problem.

Experiment 2. We tested `quadeig` against `polyeig` on QEPs with singular leading and/or trailing coefficient matrices. Table 3.2 shows that deflating speeds up the execution time. We do not see a signficant decrease in the computation time for the `spring dashpot` quadratic where $n = 1002$ and the leading coeffient M has low rank $r_2 = 2$, however as seen in Section 1.5.4 the leading coefficient is of the form $M = \text{diag}(\rho\widetilde{M}_{11}, 0)$. Due to the structure in M the routine `DGGBAL` (used in MATLAB's `eig` which is called by `polyeig` to solve the linear problem) is able to permute the linearization to peform deflation. As a result there is not as signficant a decrease in the compuation time when `quadeig` is used to solve the problem.

`polyeig` cannot cope with computing both eigenvalues and eigenvectors for the `railtrack2` problem on the machine used for these computations (hence this test was omitted from Table 3.1).

The `speaker box` problem is a 107×107 quadratic in the NLEVP collection, which comes from a finite element model of a speaker box [37, Ex. 5.5]. The stiffness matrix A_0 has rank $106 < n$ and the matrix coefficients have large variation in the norms: $\|M\|_2 = 1$, $\|C\|_2 = 5.7 \times 10^{-2}$, $\|K\|_2 = 1.0 \times 10^7$. The zero eigenvalue is not detected by `polyeig` and is computed as $\pm$7.4e-2.

Experiment 3. We investigate the effect of applying the diagonal scaling of Section 2.3.2 with the scaling parameter set to $\zeta = 1$ which is the default suggested in [6] if there is no knowledge of the desired magnitude of eigenvalues. We first apply the Fan, Lin and Van Dooren scaling and then diagonal scaling. Table 3.3 contains

Table 3.1: Quadratic eigenvalue problems from NLEVP collection. Largest backward errors of eigenpairs, and corresponding eigenvalue λ, computed by `polyeig` and `quadeig`. **D** indicates that deflation was performed by `quadeig` since the problem has singular leading or trailing coefficients.

		`polyeig`		`quadeig`		
Problem	n	$\lambda = \alpha/\beta$	$\eta_Q^{\max}(x,\alpha,\beta)$	$\eta_Q^{\max}(x,\alpha,\beta)$	$\eta_Q^{\max}(y^*,\alpha,\beta)$	
`acoustic_wave_1d`	10	2.8e+000	2.2e-015	6.5e-016	6.2e-016	
`acoustic_wave_2d`	30	-2.6e+000	5.3e-016	5.1e-016	5.5e-016	
`bicycle`	2	-7.8e-001	1.3e-015	1.1e-016	4.6e-017	
`bilby`	5	-8.8e-018	2.9e-016	4.9e-016	1.9e-016	
`cd_player`	60	-1.7e+006	1.7e-010	2.2e-012	4.9e-012	
`closed_loop`	2	-1.1e+000	1.8e-016	1.5e-016	1.2e-016	
`damped_beam`	200	-8.2e+000	3.7e-009	8.6e-016	7.1e-016	
`dirac`	80	-7.0e+000	3.7e-015	1.3e-015	1.6e-015	
`gen_hyper2`	15	9.8e+000	7.8e-016	5.2e-016	6.8e-016	
`intersection`	10	2.5e+001	3.7e-017	1.3e-016	1.3e-016	
`hospital`	24	-2.5e+000	2.5e-013	1.1e-015	1.1e-015	
`metal_strip`	9	-4.5e+000	3.1e-014	4.9e-016	3.8e-016	
`mobile_manipulator`	5	-5.2e-002	1.2e-018	5.8e-017	1.5e-017	**D**
`omnicam1`	9	3.7e-001	1.8e-015	1.2e-016	4.4e-017	
`omnicam2`	15	2.6e-001	3.9e-017	1.5e-016	2.8e-016	
`pdde_stability`	225	-4.0e+001	4.5e-014	1.3e-014	1.4e-014	
`power_plant`	8	-3.2e+000	1.3e-008	4.9e-016	4.2e-017	
`qep1`	3	3.3e-001	2.0e-016	7.1e-017	3.5e-017	
`qep2`	3	0.0e+000	9.6e-017	1.2e-016	1.2e-016	
`qep3`	3	3.0e+000	9.4e-017	1.1e-016	9.0e-017	
`railtrack`	1005	9.2e+000	2.0e-008	2.3e-015	5.9e-015	**D**
`relative_pose_6pt`	10	-8.4e-003	1.8e-015	5.0e-016	1.5e-016	
`shaft`	400	1.7e-003	5.2e-008	7.2e-016	7.1e-016	**D**
`sign1`	81	-6.1e-001	2.1e-016	7.1e-016	6.9e-016	
`sign2`	81	2.7e+000	5.1e-016	1.7e-015	1.1e-015	
`sleeper`	10	-1.3e+001	1.1e-015	4.7e-016	4.7e-016	
`speaker_box`	107	-2.0e-008	1.7e-011	2.7e-016	3.0e-016	**D**
`spring`	5	-2.9e+001	2.6e-016	4.7e-016	5.6e-016	
`spring_dashpot`	10	-6.0e-003	1.1e-015	3.3e-016	1.3e-016	**D**
`spring_dashpot`	1002	-1.0e-001	7.4e-015	4.0e-015	6.3e-015	**D**
`wing`	3	9.5e-002	4.8e-015	2.1e-016	4.8e-016	
`wiresaw1`	10	1.0e-015	1.9e-014	3.4e-016	3.5e-016	
`wiresaw2`	10	-8.0e-001	3.5e-014	9.1e-016	8.3e-016	

Table 3.2: Execution time in seconds for eigenvalue computation of quadratics in NLEVP with singular A_0 and/or A_2.

Problem	n	r_0	r_2	`polyeig`	`quadeig`
`speaker_box`	107	106	107	0.20	0.10
`shaft`	400	400	199	1.94	1.68
`spring_dashpot`	1002	1002	2	15.06	13.95
`railtrack`	1005	67	67	25.76	4.69
`railtrack2`	1410	705	705	203.06	97.98

maximum backward errors for the `cd_player` and `speaker box` quadratics from the NLEVP collection, for which diagonal scaling has a significant impact on the solution compared with applying just the scaling of Fan, Lin and Van Dooren as in Table 3.1.

For the `cd_player` problem the use of diagonal scaling reduces τ and we see a significant improvement in the maximum backward error. After applying diagonal scaling to the `speaker_box` problem, however, the value of τ increases from when only the scaling of Fan, Lin and Van Dooren is applied.

Table 3.3: `cd_player` and `speaker_box` problems from NLEVP collection. Largest backward errors of eigenpairs computed by `quadeig` comparing two scaling types, FLV (Fan, Lin and Van Dooren scaling only) and DS, FLV (diagonal scaling, then Fan, Lin and Van Dooren scaling).

Problem	Scaling	$\eta_Q^{\max}(x,\alpha,\beta)$	$\eta_Q^{\max}(y^*,\alpha,\beta)$	$\max(\omega)$	τ
`cd_player`	FLV	5.2e-012	1.1e-011	2.2e+04	2.2e+04
	DS, FLV	8.7e-014	8.9e-014	4.0e+01	3.9e+01
`speaker_box`	FLV	3.5e-016	5.1e-016	1.0e+00	1.8e-05
	DS, FLV	2.1e-013	3.4e-015	6.1e+02	6.0e+02

3.7 Conclusion

We have presented a general purpose eigensolver for dense QEPs, which, in comparison to the existing MATLAB routine `polyeig` incorporates recent contributions on the numerical solution of polynomial eigenvalue problems, namely a scaling of the eigenvalue parameter prior to the computation, and a choice of linearization with favourable conditioning and backward stability properties and, if they are present, deflation of infinite and zero eigenvalues using rank revealing factorizations.

The algorithm `quadeig` has been tested on real problems from the NLEVP benchmark collections, and from the results we can see an increase in accuracy of the solution in terms of backward error. These improvements are a result of a combination of implementing the scaling of Fan, Lin and Van Dooren, and using recent theory to recover the eigenvectors, and including a preprocessing step that reveals the zero and infinite eigenvalues contributed by singular leading and trailing matrix coefficients and deflates them. For problems with singular leading or trailing coefficients, the preprocessing step can lead to a signficant decrease in the computation time, for example in the `railtrack` and `railtrack2` problems.

The use of diagonal scaling can result in an improvement in accuracy, but requires the order of magnitude of desired eigenvalues. Since we are producing a general purpose algorithm we cannot in general expect the user to specify the magnitude of the desired eigenvalues. Hence we only include diagonal scaling in our algorithm as an option.

Chapter 4

Deflating Quadratic Matrix Polynomials with Structure Preserving Transformations

4.1 Introduction

We consider the quadratic matrix polynomial $Q(\lambda) = \lambda^2 M + \lambda C + K$, where $M, C, K \in \mathbb{R}^{n\times n}$ with M nonsingular, and the associated quadratic eigenvalue problem

$$Q(\lambda)x = 0, \quad y^* Q(\lambda) = 0, \tag{4.1.1}$$

where λ is an eigenvalue, x and y are corresponding right and left eigenvectors, respectively (where if M, C and K are symmetric $x = \bar{y}$). Throughout, we use the subscript R to denote right eigenvectors or when referring to transformations applied to the right, and the subscript L for left eigenvectors and transformations applied to the left. We also denote by $\Lambda(Q)$ the spectrum of Q.

Given two eigentriples (λ_j, x_j, y_j), $j = 1, 2$ satisfying appropriate conditions,

we propose a deflation procedure that decouples $Q(\lambda)$ into a quadratic $Q_d(\lambda) = \lambda^2 M_d + \lambda C_d + K_d$ of dimension $n-1$ and a scalar quadratic $q(\lambda) = \lambda^2 m + \lambda c + k = m(\lambda - \lambda_1)(\lambda - \lambda_2)$ such that (a)

$$\Lambda(Q) = \Lambda(Q_d) \cup \{\lambda_1, \lambda_2\},$$

where $\Lambda(Q)$ denotes the spectrum of Q and (b) there exist well-defined relations between the eigenvectors of $Q(\lambda)$ and those of the decoupled quadratic

$$\widetilde{Q}(\lambda) = \begin{bmatrix} Q_d(\lambda) & 0 \\ 0 & q(\lambda) \end{bmatrix}. \tag{4.1.2}$$

This is termed "strong deflation" in the engineering community as opposed to "weak deflation" which is achieved by introducing zeros in the last rows or last columns of the matrices.

We cannot in general construct an $n \times n$ equivalence transformation with non-singular matrices P and T such $P^T Q(\lambda) T = \widetilde{Q}(\lambda)$, where $\widetilde{Q}(\lambda)$ is the decoupled quadratic in (4.1.2) [43], unlike the case for linear polynomials $A - \lambda B$. The standard way of treating quadratic matrix polynomials, both theoretically and numerically, is to convert them into equivalent linear matrix pencils of twice the dimension, a process called linearization [20], described earlier in Section 2.1. Deflation procedures for matrix pencils ignore the block structure of linearizations such as $L_2(\lambda)$. They produce a deflated pencil that is not in general a linearization of a quadratic matrix polynomial [38].

Garvey, Friswell and Prells [16] and later Chu and Xu [12] showed that for quadratics with symmetric coefficients and semisimple eigenvalues (i.e., each eigenvalue λ appears only in 1×1 Jordan blocks in a Jordan triple for Q [20]), there

exists a real nonsingular matrix $W \in \mathbb{R}^{2n\times 2n}$ such that $W^T L_2(\lambda) W = L_D(\lambda)$, where

$$L_D(\lambda) = \lambda \begin{bmatrix} 0 & D_M \\ D_M & D_C \end{bmatrix} + \begin{bmatrix} -D_M & 0 \\ 0 & D_K \end{bmatrix}, \tag{4.1.3}$$

with D_M, D_C, D_K diagonal. The pencil $L_D(\lambda)$ is a linearization of the diagonal quadratic $Q_D(\lambda) = \lambda^2 D_M + \lambda D_C + D_K$, which clearly has the same eigenvalues as $Q(\lambda)$. The proof of the diagonalization of the blocks of $L_2(\lambda)$ to achieve $L_D(\lambda)$ in (4.1.3) is constructive and requires the knowledge of all the eigenvalues and eigenvectors of Q. Most importantly it shows that by increasing the dimension of the transformations from $n \times n$ when working directly on Q to $2n \times 2n$ by working on a pencil of twice the dimension of Q, total decoupling of the underlying second order system can be achieved. The congruence in (4.1.3) is an example of a *structure preserving transformation* (SPT). More generally, we say that a pair (W_L, W_R) of $2n \times 2n$ real nonsingular matrices defines a structure preserving transformation for an $n \times n$ quadratic matrix polynomial $Q(\lambda) = \lambda^2 M + \lambda C + K$ with M nonsingular if

$$S_L^T \left(\begin{bmatrix} 0 & M \\ M & C \end{bmatrix}, \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix} \right) S_R = \left(\begin{bmatrix} 0 & M_1 \\ M_1 & C_1 \end{bmatrix}, \begin{bmatrix} -M_1 & 0 \\ 0 & K_1 \end{bmatrix} \right), \tag{4.1.4}$$

where M_1, C_1, and K_1 are $n \times n$ matrices [34] that define a new quadratic $Q_1(\lambda) = \lambda^2 M_1 + \lambda C_1 + K_1$ sharing the same eigenvalues as $Q(\lambda)$.

The distinction between the work in [17] and this chapter is that, whereas [17] attempts complete diagonalization given that all eigenvalues and eigenvectors are known, this work attempts to block diagonalize the quadratic (subject to the eigenvalues satisfying a number of constraints), knowing only two eigenvalues and

corresponding eigenvectors. This block diagonalization of Q by deflating two eigenvalues has a number of applications, one example is in the area of model updating. Model updating is the modification of an existing inaccurate model with measured data. The eigenvalue embedding problem is a special instance of model updating and can be defined as follows: consider a quadratic matrix polynomial

$$Q(\lambda) = \lambda^2 M + \lambda C + K$$

resulting from a second-order dynamical system with a few known eigenvalues λ_j, $j = 1\colon k$. Now suppose that new eigenvalues σ_j, $j = 1\colon k$ have been measured. There are several types of eigenvalue embedding problems but one of them consists of updating the quadratic $Q(\lambda)$ to a new quadratic $\widehat{Q}(\lambda)$ with eigenvalues σ_j, $j = 1\colon k$ replacing the eigenvalues λ_j, $j = 1\colon k$ of $Q(\lambda)$ while the remaining $2n - k$ eigenvalues of $\widehat{Q}(\lambda)$ are kept the same as those of the original problem $Q(\lambda)$. This is sometimes referred to as eigenvalue updating with no spill-over.

A number of solutions to this problem has been proposed often with additional constraints such as preservation of the symmetry of the coefficient matrices and preservation of the positive definiteness of the mass and stiffness matrices.

The deflation procedure in this chapter can be used to update eigenvalues of a quadratic matrix polynomial, knowing only the eigenvalues to be updated and their corresponding eigenvectors, maintaining the symmetry of the problem if the original quadratic is symmetric. Further work involves investigating the potential of this process for updating systems, its reliability and performance in finite precision arithmetic, and comparison with existing techniques.

We deflate two eigenvalues at a time, since the problem is quadratic. For a given pair of eigenvalues λ_1, λ_2 and their associated left and right eigenvectors x_{Lj}, x_{Rj}, $j = 1, 2$, we identify conditions under which there exist elementary SPTs (S_L, S_R)

which are rank-two modifications of the $2n \times 2n$ identity matrix and transform $Q(\lambda)$ into a new quadratic $Q_1(\lambda)$ for which λ_1 and λ_2 share the same left eigenvector z_L and same right eigenvector z_R, that is,

$$z_L^* Q_1(\lambda_j) = 0, \qquad Q_1(\lambda_j) z_R = 0, \qquad j = 1, 2. \tag{4.1.5}$$

In particular we find that λ_1 and λ_2 must be semisimple and distinct and that, if they are both real, they must also satisfy

$$\operatorname{sign}\left(\frac{x_{L2}^T Q'(\lambda_2) x_{R2}}{x_{L1}^T Q'(\lambda_1) x_{R1}}\right) = \operatorname{sign}\left(\frac{x_{L2}^T Q'(\lambda_1) x_{R1}}{x_{L1}^T Q'(\lambda_2) x_{R2}}\right),$$

which for symmetric quadratics Q means that λ_1 and λ_2 must have opposite type [20] (the type of a real eigenvalue λ of $Q(\lambda)$ with associated eigenvector x being the sign of $x^T Q'(\lambda) x = 2\lambda x^T M x + x^T C x$). Under these conditions we characterize a family of elementary SPTs that maps $(\lambda_j, x_{Rj}, x_{Lj})$ to (λ_j, z_R, z_L), $j = 1, 2$. Since our transformations are structure preserving we never work with the $2n \times 2n$ matrices in (4.1.4). Indeed the matrix coefficients of $Q_1(\lambda)$ are just low rank modifications of M, C and K and are therefore not expensive to compute. When (4.1.5) holds we then show how to construct two nonsingular matrices G_L, G_R such that $G_L^T Q_1(\lambda) G_R = \widetilde{Q}(\lambda)$ with $\widetilde{Q}(\lambda)$ as in (4.1.2), that is, the pair (G_L, G_R) deflates the two eigenvalues λ_1, λ_2.

This chapter is organized as follows. After some preliminary results in Section 4.2 on structure preserving transformations, in Section 4.5 we explain how to deflate eigenvalues of symmetric quadratic matrix polynomials. In the following section we then extend the symmetric deflation procedure to quadratics with non-symmetric coefficient matrices. In Section 4.7 we present some numerical examples that illustrate our deflation procedure. To the best of our knowledge, this work

is the first attempt at constructing a family of *nontrivial elementary SPTs that have a specific action of practical use: that of "mapping" two linearly independent eigenvectors to a set of linearly dependent eigenvectors.*

4.2 Structure Preserving Transformations

In this section we recall some necessary results from [17] and [34]. SPTs, defined in (4.1.4), have a number of important and useful properties that we begin by summarizing.

Lemma 1. [34] *Let (W_L, W_R) be an SPT transforming $Q(\lambda) = \lambda^2 M + \lambda C + K$ with M nonsingular into $\widetilde{Q}(\lambda) = \lambda^2 \widetilde{M} + \lambda \widetilde{C} + \widetilde{K}$. Then*

(i) *$Q(\lambda)$ and $\widetilde{Q}(\lambda)$ share the same eigenvalues.*

(ii) *$\widetilde{M}$ is nonsingular.*

(iii) *If (λ, x, y) is an eigentriple of $Q(\lambda)$ then*

$$W_R^{-1} \begin{bmatrix} \lambda x \\ x \end{bmatrix} = \begin{bmatrix} \lambda \widetilde{x} \\ \widetilde{x} \end{bmatrix}, \quad W_L^{-1} \begin{bmatrix} \bar{\lambda} y \\ y \end{bmatrix} = \begin{bmatrix} \bar{\lambda} \widetilde{y} \\ \widetilde{y} \end{bmatrix},$$

for some nonzero $\widetilde{x}, \widetilde{y} \in \mathbb{C}^n$ such that $\widetilde{Q}(\lambda)\widetilde{x} = 0$ and $\widetilde{y}^* \widetilde{Q}(\lambda) = 0$.

(iv) *Consider the vector space of pencils* [47], [31]

$$\mathbb{DL}(Q) = \left\{ \lambda \begin{bmatrix} v_1 M & v_2 M \\ v_2 M & v_2 C - v_1 K \end{bmatrix} + \begin{bmatrix} v_1 C - v_2 M & v_1 K \\ v_1 K & v_2 K \end{bmatrix} : \; v \in \mathbb{R}^2 \right\}.$$

If $L(\lambda) \in \mathbb{DL}(Q)$ with vector v then $\widetilde{L}(\lambda) = W_L^T L(\lambda) W_R \in \mathbb{DL}(\widetilde{Q})$ with vector v. In other words, the SPT (W_L, W_R) preserves the block structure of $\mathbb{DL}(Q)$. Moreover if $L(\lambda)$ is a linearization of Q then $\widetilde{L}(\lambda)$ is a linearization of $\widetilde{Q}(\lambda)$.

(v) *If $W_L = W_R$ and $Q(\lambda)$ is symmetric (i.e., M, C and K are symmetric) then $\widetilde{Q}(\lambda)$ is symmetric.*

4.2.1 Elementary SPTs

Matrix pairs (G_L, G_R) of the form

$$G_S = \begin{bmatrix} \widetilde{G}_S & 0 \\ 0 & \widetilde{G}_S \end{bmatrix}, \quad \det(\widetilde{G}_S) \neq 0, \quad S = L, R$$

always define an SPT for any quadratic Q. They have the property that if (G_L, G_R) transforms $Q(\lambda)$ into $\widetilde{Q}(\lambda)$ then $\widetilde{Q}(\lambda) = \widetilde{G}_L^T Q(\lambda) \widetilde{G}_R$. The pair (G_L, G_R) is called a *class one elementary SPT* when $\widetilde{G}_S = I - m_S n_S^T$ for some nonzero vectors m_S, n_S, $S = L, R$ [17].

The key elementary SPT used in our deflation procedure has the form

$$T_S = \begin{bmatrix} I + a_S b_S^T & a_S d_S^T \\ a_S f_S^T & I + a_S h_S^T \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \tag{4.2.1}$$

where $a_S, b_S, d_S, f_S, h_S \in \mathbb{R}^n$ with a_S, d_S, f_S nonzero. The matrix T_S differs from the identity matrix by a matrix of rank at most 2 and it is nonsingular if [8], [34]

$$\det(T_S) = (1 + a_S^T b_S)(1 + a_S^T h_S) - (a_S^T d_S)(a_S^T f_S) \neq 0.$$

With the notation

$$\alpha_M := a_L^T M a_R, \quad \alpha_C := a_L^T C a_R, \quad \alpha_K := a_L^T K a_R,$$

a pair (T_L, T_R) of nonsingular matrices with T_S, $S = L, R$, as in (4.2.1) forms a

class two elementary SPT if [17], [34]

$$\alpha_C = a_L^T C a_R \neq 0 \tag{4.2.2}$$

and

$$\begin{aligned}
\frac{1}{2}\alpha_C f_L + \alpha_M b_L &= -M a_R, && (4.2.3)\\
\alpha_K f_L + \frac{1}{2}\alpha_C (b_L + h_L) + \alpha_M d_L &= -C a_R, && (4.2.4)\\
\alpha_K h_L + \frac{1}{2}\alpha_C d_L &= -K a_R, && (4.2.5)\\
\frac{1}{2}\alpha_C f_R + \alpha_M b_R &= -M^T a_L, && (4.2.6)\\
\alpha_K f_R + \frac{1}{2}\alpha_C (b_R + h_R) + \alpha_M d_R &= -C^T a_L, && (4.2.7)\\
\alpha_K h_R + \frac{1}{2}\alpha_C d_R &= -K^T a_L. && (4.2.8)
\end{aligned}$$

The constraints (4.2.3)–(4.2.8) (see Section 4.3 for the derivation) force preservation of structure. Multiplying the constraints (4.2.3)–(4.2.5) on the left by a_L^T and the constraints (4.2.6)–(4.2.8) on the left by a_R^T allows us to rewrite the determinant of T_L and T_R as

$$\det(T_S) = \alpha_C^{-2}(1 + a_S^T b_S)(1 + a_S^T h_S)(\alpha_C^2 - 4\alpha_K \alpha_M), \quad S = L, R$$

which shows that

$$\alpha_C^2 - 4\alpha_K \alpha_M \neq 0 \tag{4.2.9}$$

is a necessary condition for (T_L, T_S) to be an SPT.

From (4.2.3)–(4.2.8) we have that if (T_L, T_R) transforms $Q(\lambda)$ to $\widetilde{Q}(\lambda)$ then

$$\begin{aligned}
\widetilde{K} &= K - \alpha_K h_L h_R^T - \alpha_C (h_L d_R^T + d_L h_R^T)/2 - \alpha_M d_L d_R^T, \\
\widetilde{C} &= C - \alpha_K (h_L f_R^T + f_L h_R^T) - \alpha_C (h_L b_R^T + b_L h_R^T + d_L f_R^T + f_L d_R^T)/2 \\
&\quad -\alpha_M (d_L b_R^T + b_L d_R^T), \\
\widetilde{M} &= M - \alpha_K f_L f_R^T - \alpha_C (b_L f_R^T + f_L b_R^T)/2 - \alpha_M b_L b_R^T
\end{aligned}$$

which shows that $\widetilde{M}, \widetilde{C}$, and $\widetilde{K}$ are low rank modifications of M, C, and K.

4.3 Derivation of Structure Preserving Constraints

We now summarize the derivation of the constraints (4.2.3)–(4.2.8), that must be satisfied in order that the pair (T_L, T_R) forms a class two elementary SPT [16, 17]. On requiring that a congruence transformation with (T_L, T_R) on the standard basis pencil coefficients preserves the block structure we obtain three equations,

$$\begin{bmatrix} I + a_L b_L^T & a_L d_L^T \\ a_L f_L^T & I + a_L h_L^T \end{bmatrix}^T \begin{bmatrix} C & K \\ K & 0 \end{bmatrix} \begin{bmatrix} I + a_R b_R^T & a_R d_R^T \\ a_R f_R^T & I + a_R h_R^T \end{bmatrix} = \begin{bmatrix} \widetilde{C} & \widetilde{K} \\ \widetilde{K} & 0 \end{bmatrix}, \quad (4.3.1)$$

$$\begin{bmatrix} I + a_L b_L^T & a_L d_L^T \\ a_L f_L^T & I + a_L h_L^T \end{bmatrix}^T \begin{bmatrix} M & 0 \\ 0 & -K \end{bmatrix} \begin{bmatrix} I + a_R b_R^T & a_R d_R^T \\ a_R f_R^T & I + a_R h_R^T \end{bmatrix} = \begin{bmatrix} \widetilde{M} & 0 \\ 0 & -\widetilde{K} \end{bmatrix}, \quad (4.3.2)$$

$$\begin{bmatrix} I + a_L b_L^T & a_L d_L^T \\ a_L f_L^T & I + a_L h_L^T \end{bmatrix}^T \begin{bmatrix} 0 & M \\ M & C \end{bmatrix} \begin{bmatrix} I + a_R b_R^T & a_R d_R^T \\ a_R f_R^T & I + a_R h_R^T \end{bmatrix} = \begin{bmatrix} 0 & \widetilde{M} \\ \widetilde{M} & \widetilde{C} \end{bmatrix}. \quad (4.3.3)$$

To obtain the structure preserving constraints, we first set the expressions for

$\widetilde{K}$ equal in (4.3.1), and simplify the result to obtain

$$(b_L - h_L)a_L^T K(I + a_R h_R^T) + \big[(I + b_L a_L^T)C + f_L a_L^T K - d_L a_L^T M\big]a_R d_R^T = 0. \quad (4.3.4)$$

Similarly, setting the $(1,2)$ and $(2,1)$ blocks in (4.3.2) equal to one of the corresponding matrices in (4.3.1) we obtain

$$d_L a_L^T\big[C(I + a_R b_R^T) + K a_R f_R^T + M a_R d_R^T\big] + (I + h_L a_L^T)K a_R (b_R - h_R)^T = 0. \quad (4.3.5)$$

Next, setting the (1,1) and (2,2) blocks of (4.3.3) and (4.3.1) to zero and simplifying we have

$$f_L a_L^T M(I + a_R b_R^T) + (I + b_L a_L^T)M a_R f_R^T + f_L a_L^T C a_R f_R^T = 0, \quad (4.3.6)$$

$$f_L a_L^T C a_R d_R^T + (I + h_L a_L^T)K a_R d_R^T + f_L a_L^T K(I + a_R h_R^T) = 0. \quad (4.3.7)$$

Finally, setting the off diagonal blocks of (4.3.2) to zero we have

$$(I + b_L a_L^T)M a_R d_R^T - f_L a_L^T K(I + a_R h_R^T) = 0, \quad (4.3.8)$$

$$d_L a_L^T M(I + a_R b_R^T) - (I + h_L a_L^T)K a_R f_R^T = 0. \quad (4.3.9)$$

Equations (4.3.8) and (4.3.9) can be satisfied by choosing

$$f_L = \beta(I + b_L a_L^T)M a_R \quad (4.3.10)$$

$$f_R = \beta(I + b_R a_R^T)M^T a_L \quad (4.3.11)$$

$$d_L = \beta(I + h_L a_L^T)K a_R \quad (4.3.12)$$

$$d_R = \beta(I + h_R a_R^T)K^T a_L \quad (4.3.13)$$

where $\beta \neq 0$ is an arbitrary constant. For convenience we take $\beta = -2/a_L^T C a_R$

and obtain

$$f_L = -\frac{2}{a_L^T C a_R}(I + b_L a_L^T) M a_R, \tag{4.3.14}$$

$$f_R = -\frac{2}{a_L^T C a_R}(I + b_R a_R^T) M^T a_L, \tag{4.3.15}$$

$$d_L = -\frac{2}{a_L^T C a_R}(I + h_L a_L^T) K a_R, \tag{4.3.16}$$

$$d_R = -\frac{2}{a_L^T C a_R}(I + h_R a_R^T) K^T a_L. \tag{4.3.17}$$

Equations (4.3.14)–(4.3.16) simplify to

$$\frac{1}{2}\alpha_C d_L + \alpha_K h_L = -K a_R, \qquad \frac{1}{2}\alpha_C d_R + \alpha_K h_R = -K^T a_L, \tag{4.3.18}$$

$$\frac{1}{2}\alpha_C f_L + \alpha_M b_L = -M a_R, \qquad \frac{1}{2}\alpha_C f_R + \alpha_M b_R = -M^T a_L. \tag{4.3.19}$$

Equations (4.3.18) and (4.3.19) form four structure preserving constraints. To obtain the final two structure preserving constraints, we rearrange (4.3.4) and (4.3.8) to

$$b_L - h_L = \frac{2}{\alpha_C}\Big((I + b_L a_L T)C + f_L a_L^T K + d_L a_L^T M\Big) a_R, \tag{4.3.20}$$

$$b_R - h_R = \frac{2}{\alpha_C}\Big((I + b_R a_R^T)C^T + f_R a_R^T K^T + d_R a_R^T M^T\Big) a_L. \tag{4.3.21}$$

Next we substitute (4.3.14) and (4.3.16) for f_L, and d_L in (4.3.20), (4.3.15) and (4.3.17) for f_R, and d_R in (4.3.21) and assuming $\alpha_K\alpha_M - (\alpha_C/2)^2 \neq 0$ we have

$$b_L + h_L = \frac{\frac{\alpha_C}{2} C a_R - \alpha_K M a_R - \alpha_M K a_R}{\alpha_K\alpha_M - (\alpha_C/2)^2}, \tag{4.3.22}$$

$$b_R + h_R = \frac{\frac{\alpha_C}{2} C^T a_L - \alpha_K M^T a_L - \alpha_M K^T a_L}{\alpha_K\alpha_M - (\alpha_C/2)^2}. \tag{4.3.23}$$

Now after multiplying (4.3.18) by α_M and (4.3.19) by α_K, we obtain two equations, the first by adding the two equations with K and M terms, and the second by adding the equations with K^T and M^T terms to obtain,

$$\alpha_M\alpha_K(b_L+h_L)+\frac{\alpha_C}{2}\alpha_M d_L+\alpha_M K a_R+\frac{\alpha_C}{2}\alpha_K f_L+\alpha_K M a_R=0, \quad (4.3.24)$$

$$\alpha_M\alpha_K(b_R+h_R)+\frac{\alpha_C}{2}\alpha_M d_R+\alpha_M K^T a_L+\frac{\alpha_C}{2}\alpha_K f_R+\alpha_K M^T a_L=0. \quad (4.3.25)$$

Finally substituting (4.3.22) and (4.3.23) for $\alpha_M K a_R+\alpha_K M a_R$ in (4.3.24) and $\alpha_M K^T a_L+\alpha_K M^T a_L$ in (4.3.25) respectively we obtain the two final structure preserving constraints,

$$\alpha_M d_L+\alpha_K f_L+\frac{1}{2}\alpha_C(b_L+h_L)+C a_R=0, \quad (4.3.26)$$

$$\alpha_M d_R+\alpha_K f_R+\frac{1}{2}\alpha_C(b_R+h_R)+C^T a_L=0. \quad (4.3.27)$$

To summarize, we now have six structure preserving constraints which the SPT (T_L,T_R) must satisfy:

$$\frac{1}{2}\alpha_C f_L+\alpha_M b_L=-M a_R, \quad (4.3.28)$$

$$\alpha_M d_L+\alpha_K f_L+\frac{1}{2}\alpha_C(b_L+h_L)=-C a_R, \quad (4.3.29)$$

$$\frac{1}{2}\alpha_C d_L+\alpha_K h_L=-K a_R, \quad (4.3.30)$$

$$\frac{1}{2}\alpha_C f_R+\alpha_M b_R=-M^T a_L, \quad (4.3.31)$$

$$\alpha_M d_R+\alpha_K f_R+\frac{1}{2}\alpha_C(b_R+h_R)=-C^T a_L, \quad (4.3.32)$$

$$\frac{1}{2}\alpha_C d_R+\alpha_K h_R=-K^T a_L. \quad (4.3.33)$$

4.4 Computing the Vectors Defining a Class Two SPT

Once the two vectors a_L and a_R are chosen such that (4.2.2) and (4.2.9) hold the structure preserving constraints (4.2.3)–(4.2.8) are linear in the remaining unknown vectors. They can be rewritten in matrix form as

$$VA = B \qquad \Longleftrightarrow \qquad V_L A = B_R, \quad V_R A = B_L, \tag{4.4.1}$$

where $A \in \mathbb{R}^{4\times 3}$ and $B \in \mathbb{R}^{2n\times 3}$ are given by

$$A = \begin{bmatrix} \alpha_M & \frac{1}{2}\alpha_C & 0 \\ 0 & \alpha_M & \frac{1}{2}\alpha_C \\ \frac{1}{2}\alpha_C & \alpha_K & 0 \\ 0 & \frac{1}{2}\alpha_C & \alpha_K \end{bmatrix}, \quad B = -\begin{bmatrix} Ma_R & Ca_R & Ka_R \\ M^T a_L & C^T a_L & K^T a_L \end{bmatrix} = \begin{bmatrix} B_R \\ B_L \end{bmatrix} \tag{4.4.2}$$

and $V = \left[\begin{smallmatrix} V_L \\ V_R \end{smallmatrix}\right] \in \mathbb{R}^{2n\times 4}$ with $V_S = \begin{bmatrix} b_S & d_S & f_S & h_S \end{bmatrix} \in \mathbb{R}^{n\times 4}$ for $S = L, R$ contains the remaining unknown vectors. Some calculations show that

$$\det(A^T A) = \frac{1}{4}(\alpha_C^2 - 4\alpha_M\alpha_K)^2(\alpha_C^2 + \alpha_M^2 + \alpha_K^2)$$

which is nonzero by (4.2.9), so that A has full rank and all solutions to (4.4.1) are given by

$$V = BA^+ + Q(I - AA^+) \quad \Longleftrightarrow \quad \begin{cases} V_L = B_R A^+ + Q_L(I - AA^+), \\ V_R = B_L A^+ + Q_R(I - AA^+), \end{cases}$$

for some arbitrary $Q = \left[\begin{smallmatrix} Q_L \\ Q_R \end{smallmatrix}\right] \in \mathbb{R}^{2n\times 4}$. Here A^+ is the pseudoinverse of A, which is given by $A^+ = (A^T A)^{-1} A^T$ since A has full rank (see Stewart and Sun [55, Sec.

3.1]).

The transformation T_S used in our deflation procedure performs a specific action: that of mapping two non parallel eigenvectors of Q associated with a pair of eigenvalues to just one eigenvector for $\widetilde{Q}$ associated to that same pair of eigenvalues. This results in an extra constraint of the form $z^T V = w^T$ for some given z and w that the solution V of (4.4.1) must satisfy. The next result will be needed for the existence and characterization of all the class two SPTs performing that specific action.

Theorem 4. *Let $A \in \mathbb{R}^{r\times k}$, $r \geq k$ have full rank, $B \in \mathbb{R}^{n\times k}$, $w \in \mathbb{R}^r$, and nonzero $z \in \mathbb{R}^n$ be given. The problem of finding $V \in \mathbb{R}^{n\times r}$ such that*

$$VA = B, \qquad z^T V = w^T, \tag{4.4.3}$$

has a solution if and only if $w^T A = z^T B$. In this case the general solution is

$$V = (I - zz^+)BA^+ + U(I - AA^+) + z(z^T z)^{-1}w^T, \tag{4.4.4}$$

where $U \in \mathbb{R}^{n\times r}$ is any matrix such that $z^T U = 0$.

Proof. If V is a solution to (4.4.3) then $z^T B = z^T VA = w^T A$. Conversely, if $z^T B = w^T A$ then since $A^+A = I$ multiplying V in (4.4.4) on the right by A yields $VA = B$ and since $z^T U = 0$ we have that $z^T V = w^T$ so that V in (4.4.4) is a solution to (4.4.3).

Every solution V to (4.4.3) can therefore be rewritten as

$$\begin{aligned} V &= (I - zz^+)VAA^+ - (I - zz^+)VAA^+ + V - zz^+V + zz^+V \\ &= (I - zz^+)VAA^+ + (I - zz^+)V(I - AA^+) + zz^+V \\ &= (I - zz^+)BA^+ + (I - zz^+)V(I - AA^+) + z(z^T z)^{-1}w^T, \end{aligned}$$

which is of the form (4.4.4) with $U := \big(I - zz^+\big)V$ satisfying $z^TU = 0$. □

4.5 Deflation for Symmetric Quadratics

Symmetric quadratics have the property that if x is a right eigenvector associated with the eigenvalue λ then $y = \overline{x}$ is the corresponding left eigenvector. If we therefore use congruence transformations to preserve the symmetry of the quadratic we need only consider the deflation of eigenpairs rather than eigentriples. We denote by (λ_1, x_1) and (λ_2, x_2) the two eigenpairs to be deflated. We use congruence transformations to preserve the symmetry of the quadratic. We begin by showing that when x_1 and x_2 are parallel there exists an $n \times n$ congruence transformation which, when applied directly to Q, deflates λ_1 and λ_2. When x_1 and x_2 are linearly independent, we show how to construct a class two SPT that transforms Q to a new quadratic Q_1 for which λ_1 and λ_2 share the same eigenvector. In other words, the SPT allows us to transform the original deflation problem into one we know how to handle.

4.5.1 Linearly Dependent Eigenvectors

We begin by treating the case where the eigenvalues λ_1 and λ_2 have a common eigenvector $z \in \mathbb{R}^n$. The next lemma is crucial to proving the existence of a congruence transformation that deflates these two eigenvalues. Some relations in this lemma have already been observed by Chu, Hwang, and Lin [11]

Lemma 2. *Consider the $n \times n$ symmetric quadratic $Q(\lambda) = \lambda^2 M + \lambda C + K$.*

(i) *If $Q(\lambda_j)z = 0$, $j = 1, 2$ with $z \in \mathbb{R}^n \setminus \{0\}$ and $\lambda_1 \neq \lambda_2$ then $Cz = c\,Mz$ and $Kz = k\,Mz$ with $c = -(\lambda_1 + \lambda_2)$ and $k = \lambda_1\lambda_2$. Moreover, $z^TMz \neq 0$ if and only if $z^TQ'(\lambda_j)z \neq 0$, $j = 1, 2$.*

(ii) *If* $Cz = c\,Mz$ *and* $Kz = k\,Mz$ *for some nonzero* $z \in \mathbb{C}^n$ *and* $c, k \in \mathbb{C}$ *then* $Q(\lambda_j)z = 0$, $j = 1, 2$ *with* $\lambda_{1,2} = -(c \pm \sqrt{c^2 - 4k})/2$.

Proof. (i) It follows from $\lambda_j^2 Mz + \lambda_j Cz + Kz = 0$, $j = 1, 2$ that when $\lambda_1 \neq \lambda_2$, $Cz = -(\lambda_1+\lambda_2)Mz = cMz$ and then $Kz = -\lambda_1^2 Mz + \lambda_1(\lambda_1+\lambda_2)Mz = \lambda_1\lambda_2 Mz = kMz$. If λ_1, λ_2 are semisimple then $0 \neq z^T Q'(\lambda_j)z = (2\lambda_j + c)z^T Mz$ ($Q'(\lambda)$ is the first derivative of Q with respect to λ, that is $Q'(\lambda) = 2\lambda M + C$), which implies that $z^T Mz \neq 0$.

(ii) If $Cz = c\,Mz$ and $Kz = k\,Mz$ then $Q(\lambda_j)z = (\lambda_j^2 + \lambda_j c + k)Mz = 0$, $j = 1, 2$, from which the formula for $\lambda_{1,2}$ follows. □

Assume there exists a nonsingular matrix G such that

$$Ge_n = z, \quad G^T(Mz) = me_n, \quad m = z^T Mz \tag{4.5.1}$$

where e_n is the last column of the n-by-n identity matrix. Since G and M are nonsingular we must have $m \neq 0$, or equivalently, $z^T Mz \neq 0$ which by Lemma 2(i) holds when λ_1 and λ_2 are distinct and semisimple. Thus we have that

$$G^T MGe_n = G^T Mz = me_n.$$

If λ_1 and λ_2 are distinct then by Lemma 2(i), $Cz = cMz$ and $Kz = kMz$, so that

$$G^T(\lambda^2 M + \lambda C + K)G = \lambda^2 \begin{bmatrix} \widetilde{M} & 0 \\ 0 & m \end{bmatrix} + \lambda \begin{bmatrix} \widetilde{C} & 0 \\ 0 & mc \end{bmatrix} + \begin{bmatrix} \widetilde{K} & 0 \\ 0 & mk \end{bmatrix}, \tag{4.5.2}$$

where $c = -(\lambda_1 + \lambda_2)$ and $k = \lambda_1\lambda_2$; thus G deflates the two eigenvalues λ_1 and λ_2. Note that if $\lambda_1 = \lambda_2$ and, Cz and Kz are multiples of Mz then, as long as $z^T Mz \neq 0$, G in (4.5.1) deflates λ_1 and λ_2 from Q. It is easily seen from (4.5.2) that in this case $\lambda_1 (= \lambda_2)$ must be a defective eigenvalue with partial multiplicity

2.

We build the matrix G in two steps. We begin by constructing a Householder reflector H [21] such that

$$H(Mz) = \|Mz\|_2 e_n.$$

We then form $L = I_n + rs^T$, where $s^T e_n = 1$ and $r = \frac{\|Mz\|_2}{m} Hz - e_n$, so that

$$Le_n = \frac{\|Mz\|_2}{m} Hz, \quad L^T e_n = e_n$$

since $r^T e_n = \frac{\|Mz\|_2}{m} z^T H e_n - 1 = \frac{z^T Mz}{m} - 1 = 0$. Hence

$$G = \frac{m}{\|Mz\|_2} HL \tag{4.5.3}$$

satisfies (4.5.1). It is shown in [18] that taking

$$s = e_n - \frac{1 + \sqrt{1 + r^T r}}{r^T r} r$$

minimizes the condition number $\kappa(L)$ of L and that with this choice,

$$\kappa_2(G)^2 = \kappa_2(L)^2 = \frac{\sqrt{1 + \|r\|_2^2} + \|r\|_2}{\sqrt{1 + \|r\|_2^2} - \|r\|_2},$$

which is reasonably small as long as $\|r\|_2$ is not much smaller than 1. Using $\|Mz\|_2 He_n = Mz$ and the definition of r we have that

$$\|r\|_2^2 = r^T r = (z^T M^2 z)(z^T z)/(z^T Mz)^2 - 1$$

showing that $\|r\|_2$ does not depend on the norm of z or M.

4.5.2 Linearly Independent Eigenvectors

When x_1 and x_2 are linearly independent there is clearly no nonsingular transformation mapping the full rank matrix $[\,x_1 \quad x_2\,]$ to the rank-one matrix $[\,e_n \quad e_n\,]$. The idea in this case is to build an SPT T that transforms $Q(\lambda)$ with eigenpairs (λ_j, x_j), $j = 1, 2$ to $Q_1(\lambda)$ with eigenpairs (λ_j, z), $j = 1, 2$ that can then be deflated using the procedure described in Section 4.5.1. We only consider the case where $\lambda_1 \neq \lambda_2$. Indeed when the two eigenvalues are equal and x_1 is not parallel to x_2, λ_1 and λ_2 belong to two distinct Jordan blocks. In this case, the decoupling (4.5.2) cannot be achieved.

Since we aim to treat the deflation of real eigenpairs together with that of complex conjugate eigenpairs, we introduce the real matrices $\Lambda \in \mathbb{R}^{2\times 2}$ and $X \in \mathbb{R}^{n\times 2}$ defined by

$$\Lambda = \begin{cases} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} & \text{if } \lambda_1 \text{ and } \lambda_2 \text{ are real,} \\ \\ \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} & \text{if } \lambda_1 = \bar{\lambda}_2 = \alpha + i\beta \text{ with } \beta \neq 0, \end{cases} \tag{4.5.4}$$

and

$$X = \begin{cases} [\,x_1 \quad x_2\,] & \text{for real eigenpairs,} \\ [\,u \quad v\,] & \text{for complex eigenpairs with } x_1 = \bar{x}_2 = u + iv. \end{cases} \tag{4.5.5}$$

We want to construct a class two elementary SPT $T = I_{2n} + \left[\begin{smallmatrix} ab^T & ad^T \\ af^T & ah^T \end{smallmatrix}\right]$ with $a, b, d, f, h \in \mathbb{R}^n$ and a nonzero vector $z \in \mathbb{R}^n$ (for simplicity we assume $\|z\|_2 = 1$)

such that

$$T^{-1}\begin{bmatrix} X\Lambda \\ X \end{bmatrix} = \begin{bmatrix} ze^T\Lambda \\ ze^T \end{bmatrix}, \tag{4.5.6}$$

where $e = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$. This constraint means that $T^{-1}\left[\begin{smallmatrix}\lambda_j x_j\\ x_j\end{smallmatrix}\right] = \left[\begin{smallmatrix}\lambda_j z\\ z\end{smallmatrix}\right]$, for $j = 1, 2$. Hence if T transforms $Q(\lambda)$ to $Q_1(\lambda)$ then by Lemma 1(iii), $Q_1(\lambda_j)z = 0$, $j = 1, 2$. We rewrite (4.5.6) in terms of the $6n$ unknown vectors a, b, d, f, h, z as

$$ze^T\Lambda + (b^Tz)ae^T\Lambda + (d^Tz)ae^T = X\Lambda, \tag{4.5.7}$$

$$ze^T + (f^Tz)ae^T\Lambda + (h^Tz)ae^T = X, \tag{4.5.8}$$

and solve (4.5.7)–(4.5.8) for a, z and the scalars b^Tz, d^Tz, f^Tz, h^Tz as follows.

Let nonzero $p, q \in \mathbb{R}^2$ be such that

$$e^Tp = 0, \quad e^T\Lambda p = 1, \qquad e^Tq = 1, \quad e^T\Lambda q = 0.$$

Since $\lambda_1 \neq \lambda_2$, it is easily seen that

$$p = \gamma(\lambda_1 - \lambda_2)^{-1}\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad q = \Lambda p - (\lambda_1 + \lambda_2)p, \qquad \Lambda q = -\lambda_1\lambda_2 p,$$

with $\gamma = 1$ for real eigenpairs and $\gamma = i$ for complex eigenpairs. Multiplying (4.5.8) on the right by p yields $(f^Tz)a = Xp$. Since the columns of X are linearly independent, we have that $f^Tz \neq 0$. Now without loss of generality, we normalize a such that $a^Ta = 1$. It follows that

$$a = (f^Tz)^{-1}Xp, \qquad f^Tz = \|Xp\|_2 \neq 0. \tag{4.5.9}$$

Multiplying (4.5.7) on the right by p yields $z + (b^Tz)a = X\Lambda p$. If we choose to

normalize z such that $e_\ell^T z = 1$, where ℓ is such that $|e_\ell^T a| = \|a\|_\infty$ then

$$b^T z = (e_\ell^T X \Lambda p - 1)/(e_\ell^T a), \qquad z = X\Lambda p - (b^T z)a. \tag{4.5.10}$$

Multiplying (4.5.7)–(4.5.8) on the right by q and on the left by e_ℓ^T gives

$$d^T z = (e_\ell^T X \Lambda q)/(e_\ell^T a), \qquad h^T z = (e_\ell^T X q - 1)/(e_\ell^T a). \tag{4.5.11}$$

What is now left is the construction of $V := [\, b \quad d \quad f \quad h\,]$ such that $z^T V = w^T$, where $w^T = [\, b^T z \quad d^T z \quad f^T z \quad h^T z\,]$, and $VA = B$, since T is structure preserving (see Section 4.2.1), where $B = -[\, Ma \quad Ca \quad Ka\,]$ and A is as in (4.4.2) with $\alpha_M = a^T M a$, $\alpha_C = a^T C a \neq 0$ and $\alpha_K = a^T K a$. We know from Theorem 4 that a solution V to $VA = B$, $z^T V = w^T$ exists if and only if

$$w^T A = z^T B. \tag{4.5.12}$$

The next lemma, crucial for the deflation process, provides a necessary and sufficient condition on the eigenpairs (λ_j, x_j), $j = 1, 2$ for (4.5.12) to hold.

Lemma 3. *The relation $w^T A = z^T B$ holds if and only if the eigenpairs (λ_1, x_1) and (λ_2, x_2) of $Q(\lambda)$ satisfy*

$$x_1^T Q'(\lambda_1) x_1 = \epsilon x_2^T Q'(\lambda_2) x_2 \tag{4.5.13}$$

with $\epsilon = -1$ for real eigenpairs and $\epsilon = 1$ for complex conjugate eigenpairs.

Proof. Tedious calculations left to Section 4.8 show that the row vector $g^T = w^T A - z^T B$ has the form

$$g^T = \gamma\big(x_1^T Q'(\lambda_1) x_1 - \epsilon x_2^T Q'(\lambda_2) x_2\big)[\, 1 \quad c \quad k\,],$$

where γ is a nonzero scalar, $c = -(\lambda_1 + \lambda_2)$, $k = \lambda_1\lambda_2$, $\epsilon = -1$ for real eigenpairs and $\epsilon = 1$ for complex eigenpairs. □

For real eigenpairs, the condition (4.5.13) implies that λ_1 and λ_2 must have opposite type, (the type of a real eigenvalue λ of $Q(\lambda)$ with associated eigenvector x being the sign of $x^TQ'(\lambda)x = 2\lambda x^TMx + x^TCx$). Note that this is to be expected from the theory of Hermitian matrix polynomials since for a symmetric quadratic with $2r$ distinct real eigenvalues, r of them are of positive type and r of them are of negative type (see [20] or [40, Appendix]). Hence when deflating two real eigenpairs, one must be of positive type and the other of negative type. Under this condition, (4.5.13) is achieved with the scaling

$$x_1 \leftarrow x_1/\sqrt{|x_1^TQ'(\lambda_1)x_1|}, \qquad x_2 \leftarrow x_2/\sqrt{|x_2^TQ'(\lambda_2)x_2|}$$

as long as both λ_1 and λ_2 are semisimple, so that $x_j^TQ'(\lambda_j)x_j \neq 0$, $j = 1, 2$.

For complex conjugate eigenpairs, (4.5.13) is achieved with the scaling

$$x_1 \leftarrow x_1/\sqrt{x_1^TQ'(\lambda_1)x_1}, \quad x_2 = \bar{x}_1$$

if $x_1^TQ'(\lambda_1)x_1 \neq 0$ and no scaling otherwise. (Note here the use of "T" rather than "$*$".)

With the above scaling, Lemma 3 together with Theorem 4 tells us that the equations $VA = B$ and $z^TV = w^T$ have the solutions

$$V = \Big(I - \frac{zz^T}{z^Tz}\Big)BA^+ + U(I - AA^+) + \frac{z}{z^Tz}w^T, \tag{4.5.14}$$

where $U \in \mathbb{R}^{n\times 4}$ is any matrix such that $z^TU = 0$. It follows that (4.5.9)–(4.5.11) and (4.5.14) define a family of class two elementary SPTs T transforming $Q(\lambda)$ with eigenpairs (λ_j, x_j) to $Q_1(\lambda)$ with eigenpairs (λ_j, z), $j = 1, 2$. Identifying which

solution minimizes the condition number $\kappa_2(T) = \|T\|_2\|T^{-1}\|_2$ remains an open problem.

4.6 Deflation for Nonsymmetric Quadratics

The deflation procedure described in Section 4.5 extends to the case where M, C, and K are nonsymmetric. We denote by $(\lambda_j, x_{Rj}, x_{Lj})$, $j = 1, 2$ the two eigentriples to be deflated from $Q(\lambda)$ with $(\lambda_2, x_{R2}, x_{L2}) = (\bar{\lambda}_1, \bar{x}_{R1}, \bar{x}_{L1})$ when $\mathrm{Im}(\lambda_1) \neq 0$. In contrast with the symmetric deflation procedure we use equivalence transformations rather than congruence transformations since we do not need to preserve symmetry. Three situations must be considered.

4.6.1 Parallel Left Eigenvectors and Parallel Right Eigenvectors

Without loss of generality let us assume in this case that $x_{L1} = x_{L2} \equiv z_L$ and $x_{R1} = x_{R2} \equiv z_R$ with $z_L, z_R \in \mathbb{R}^n$ so that

$$z_L^T Q_1(\lambda_j) = 0, \qquad Q_1(\lambda_j) z_R = 0, \qquad j = 1, 2, \tag{4.6.1}$$

since both the left and right eigenvectors are parallel, $Q_1 = Q$. As in Lemma 2 it is easily shown that if (4.6.1) holds with $\lambda_1 \neq \lambda_2$ then

$$C_1 z_R = c M_1 z_R, \qquad K_1 z_R = k M_1 z_R, \tag{4.6.2}$$

$$z_L^T C_1 = c z_L^T M_1, \qquad z_L^T K_1 = k z_L^T M_1, \tag{4.6.3}$$

where $c = -(\lambda_1 + \lambda_2)$ and $k = \lambda_1\lambda_2$. Moreover if λ_1 and λ_2 are semisimple then $z_L^T M z_R \neq 0$. Suppose there exist nonsingular matrices G_L and G_R such that

$$G_L^T M z_R = m e_n, \qquad G_L e_n = z_L, \tag{4.6.4}$$

$$G_R^T M^T z_L = m e_n, \qquad G_R e_n = z_R, \tag{4.6.5}$$

where $m = z_L^T M z_R$. (The left (right) transformation G_L (G_R) depends on the right (left) eigenvector.) Since M, G_L, and G_R are nonsingular we must have $m \neq 0$ which is guaranteed when λ_1 and λ_2 are distinct and semisimple. With G_L and G_R satisfying (4.6.4) and (4.6.5) we have

$$G_L^T M G_R e_n = G_L^T M z_R = m e_n, \quad e_n^T G_L^T M G_R = z_L^T M G_R = m e_n^T$$

and on using (4.6.2)–(4.6.5) it follows that

$$G_L^T(M, C, K)G_R = \left(\begin{bmatrix} \widetilde{M} & 0 \\ 0 & m \end{bmatrix}, \begin{bmatrix} \widetilde{C} & 0 \\ 0 & mc \end{bmatrix}, \begin{bmatrix} \widetilde{K} & 0 \\ 0 & mk \end{bmatrix} \right). \tag{4.6.6}$$

If we let $u_L = M z_R$ and $u_R = M^T z_L$, the matrices G_L and G_R can be taken in the form

$$G_S = \frac{m}{\|u_S\|_2} H_S L_S, \quad S = L, R,$$

where H_S is a Householder reflector such that $H_S u_S = \|u_S\|_2 e_n$ and $L_S = I_n - r_S s_S^T$ with

$$r_S = \frac{\|u_S\|_2}{m}\, H_S z_S - e_n, \qquad s_S = e_n - \frac{1 + \sqrt{1 + r_S^T r_S}}{r_S^T r_S} r_S$$

so that

$$L_S e_n = \frac{\|u_S\|_2}{m} H_S z_S, \qquad L_S^T e_n = e_n.$$

Thus it is easy to check that the pair (G_L, G_R) satisfies (4.6.2) and (4.6.3) and therefore deflates λ_1 and λ_2 from Q.

4.6.2 Non Parallel Left Eigenvectors and Non Parallel Right Eigenvectors

Our aim, as for the symmetric case, is to build a class two elementary SPT (T_L, T_R), with T_L not necessarily equal to T_R, that transforms $Q(\lambda)$ to a new quadratic $Q_1(\lambda)$ for which λ_1 and λ_2 share the same left eigenvector z_L and the same right eigenvector z_R. In order to apply the deflation process of Section 4.6.1, we assume that λ_1 and λ_2 are semisimple and distinct. When $\lambda_1 = \lambda_2$ with linearly independent eigenvectors then λ_1 and λ_2 belong to two distinct Jordan blocks and the decoupling (4.6.6) cannot be achieved.

Let T_S be such that

$$T_S^{-1} \begin{bmatrix} X_S \Lambda_S \\ X_S \end{bmatrix} = \begin{bmatrix} z_S e^T \Lambda_S \\ z_S e^T \end{bmatrix}, \tag{4.6.7}$$

with $\Lambda_L = \Lambda^T$ and $\Lambda_R = \Lambda$ where Λ, X_L and X_R are formed as in (4.5.4) and (4.5.5), and $e = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. If the pair (T_L, T_R) is structure preserving and transforms $Q(\lambda)$ to $Q_1(\lambda)$ then the constraint (4.6.7) for $S = L$ and $S = R$ together with Lemma 1(iv) implies that $z_L^T Q_1(\lambda_j) = 0$ and $Q_1(\lambda_j) z_R = 0$, $j = 1, 2$.

Now if we choose T_S to have the form (4.2.1) then with the following normalizations of a_S and z_S,

$$a_S^T a_S = 1, \qquad e_{\ell_S}^T z_S = 1, \qquad |e_{\ell_S}^T a_S| = \|a_S\|_\infty, \tag{4.6.8}$$

we obtain in a similar way to the symmetric case described in Section 4.5.2, that

under the constraint (4.6.7),

$$\begin{aligned} f_S^T z_S &= \|X_S p_S\|_2 \neq 0, & a_S &= (f_S^T z_S)^{-1} X_S p_S, \\ b_S^T z_S &= (e_{\ell_S}^T X_S \Lambda_S p_S - 1)/(e_{\ell_S}^T a_S), & z_S &= X_S \Lambda_S p_S - (b_S^T z_S) a_S, \\ d_S^T z_S &= (e_{\ell_S}^T X_S \Lambda_S q_S)/(e_{\ell_S}^T a_S), & h_S^T z_S &= (e_{\ell_S}^T X_S q_S - 1)/(e_{\ell_S}^T a_S), \end{aligned} \tag{4.6.9}$$

where $p_S, q_S \in \mathbb{R}^2$ are such that

$$e^T p_S = 0, \qquad e^T \Lambda_S p_S = 1, \qquad e^T q_S = 1, \qquad e^T \Lambda_S q_S = 0.$$

Assuming that $a_L^T C a_R \neq 0$, the class two elementary SPT (T_L, T_R) is completely determined if we can find two matrices $V_L, V_R \in \mathbb{R}^{n\times 4}$ of the form $[\, b_S \quad d_S \quad f_S \quad h_S \,]$ with $S = L, R$ such that

$$V_L A = B_R, \qquad z_L^T V_L = w_L^T, \tag{4.6.10}$$

$$V_R A = B_L, \qquad z_R^T V_R = w_R^T, \tag{4.6.11}$$

where $A \in \mathbb{R}^{4\times 3}$ and $B \in \mathbb{R}^{2n\times 3}$ are as in (4.4.2) and $w_S = [\, b_S^T z_S \quad d_S^T z_S \quad f_S^T z_S \quad h_S^T z_S \,]$, $S = L, R$. From Theorem 4, a solution V_L to (4.6.10) and a solution V_R to (4.6.11) exist if and only if $w_L^T A = Z_L^T B_R$ and $w_R^T A = Z_R^T B_L$.

Lemma 4. *The relations*

$$w_L^T A - Z_L^T B_R = 0, \qquad w_R^T A - Z_R^T B_L = 0$$

hold if and only if the eigentriples $(\lambda_1, x_{R1}, x_{L1})$ *and* $(\lambda_2, x_{R2}, x_{L2})$ *of* $Q(\lambda)$ *satisfy*

$$x_{L1}^T Q'(\lambda_1) x_{R1} = \epsilon x_{L2}^T Q'(\lambda_2) x_{R2}, \qquad x_{L1}^T Q'(\lambda_2) x_{R2} = \epsilon x_{L2}^T Q'(\lambda_1) x_{R1}, \tag{4.6.12}$$

with $\epsilon = -1$ *for real eigentriples and* $\epsilon = 1$ *for complex conjugate eigentriples.*

Proof. Let $g_L^T = w_L^T A - Z_L^T B_R$ and $g_R^T = w_R^T A - Z_R^T B_L$. Calculations along the same lines as those presented in Section 4.8 for the symmetric case show that for real eigentriples,

$$\begin{aligned} g_L^T &= \gamma_L \left(\xi_1 + \xi_2 - \xi_3 - \xi_4\right) [1,\ c,\ k], \\ g_R^T &= \gamma_R \left(\xi_1 + \xi_2 - \xi_5 - \xi_6\right) [1,\ c,\ k], \end{aligned}$$

where γ_L and γ_R are nonzero scalars, $c = -(\lambda_1 + \lambda_2)$, $k = \lambda_1\lambda_2$ and

$$\begin{aligned} &\xi_1 = x_{L1}^T Q'(\lambda_1) x_{R1}, \quad \xi_3 = x_{L1}^T Q'(\lambda_1) x_{R2}, \quad \xi_5 = x_{L1}^T Q'(\lambda_2) x_{R2}, \\ &\xi_2 = x_{L2}^T Q'(\lambda_2) x_{R2}, \quad \xi_4 = x_{L2}^T Q'(\lambda_2) x_{R1}, \quad \xi_6 = x_{L2}^T Q'(\lambda_1) x_{R1}. \end{aligned} \tag{4.6.13}$$

From $x_{L1}^T Q(\lambda_j) x_{R2} = 0,\ j = 1, 2$ we find that $x_{L1}^T C x_{R2} = -(\lambda_1 + \lambda_2) x_{L1}^T M x_{R2}$, from which it follows that $x_{L1}^T Q'(\lambda_1) x_{R2} = -x_{L1}^T Q'(\lambda_2) x_{R2}$, that is, $\xi_3 = -\xi_5$. In an analogous way we find that $x_{L2}^T Q'(\lambda_1) x_{R1} = -x_{L2}^T Q'(\lambda_2) x_{R1}$, that is, $\xi_4 = -\xi_6$. Hence, $g_L = g_R = 0$ if and only if $\xi_1 + \xi_2 = 0$ and $\xi_5 + \xi_6 = 0$.

For complex conjugate eigentriples, we find that

$$\begin{aligned} g_L^T &= \widetilde{\gamma}_L \left(i\xi_7 - i\xi_8 + \xi_5 + \xi_6\right) [1,\ c,\ k], \\ g_R^T &= \widetilde{\gamma}_R \left(i\xi_1 - i\xi_2 + \xi_5 + \xi_6\right) [1,\ c,\ k], \end{aligned}$$

where $\widetilde{\gamma}_L$ and $\widetilde{\gamma}_R$ are nonzero complex scalars, ξ_j, $j = 1, 2, 5, 6$ are defined in (4.6.13) and $\xi_7 = x_{L1}^T Q'(\lambda_2) x_{R1}$, $\xi_8 = x_{L2}^T Q'(\lambda_1) x_{R2}$. Using $x_{L1}^* Q(\lambda_j) x_{R2} = 0$, $j = 1, 2$ it is easily shown that $x_{L1}^* Q'(\lambda_1) x_{R2} = -x_{L1}^* Q'(\lambda_2) x_{R2}$ which, by taking the conjugate, becomes $\xi_7 = -\xi_1$. We show similarly that $\xi_8 = -\xi_2$. Hence, $g_L = g_R = 0$ if and only if $\xi_1 - \xi_2 = 0$ and $\xi_5 + \xi_6 = 0$ which completes the proof. □

The assumption that λ_1 and λ_2 are semisimple implies that the terms on the left-hand side for real eigentriples and the terms on the right-hand side relation

in (4.6.12) for complex conjugate eigentriples are nonzero. $x_{Lj}^T Q'(\lambda_j) x_{Rj} = 0$ or $x_{Lj}^T Q'(\lambda_k) x_{Rk} = 0$, $j \neq k$, then a scaling similar to that described after Lemma 3 can be applied to ensure that (4.6.12) holds. When both $x_{L1}^T Q'(\lambda_1) x_{R1}$ and $x_{L1}^T Q'(\lambda_2) x_{R2}$ are nonzero, we let

$$\rho_1 = \frac{x_{L2}^T Q'(\lambda_2) x_{R2}}{x_{L1}^T Q'(\lambda_1) x_{R1}}, \qquad \rho_2 = \frac{x_{L2}^T Q'(\lambda_1) x_{R1}}{x_{L1}^T Q'(\lambda_2) x_{R2}}.$$

Thus for real eigentriples, (4.6.12) can be achieved for an appropriate scaling of the eigenvectors only if $\text{sign}(\rho_1) = \text{sign}(\rho_2)$, in which case we can apply the scaling

$$\begin{aligned} x_{L1} &\leftarrow |\rho_1|^{1/2} x_{L1}, \quad x_{R1} \leftarrow |\rho_1|^{1/2} x_{R1}, \\ x_{L2} &\leftarrow |\rho_2|^{-1/2} x_{L2}, \quad x_{R2} \leftarrow |\rho_2|^{1/2} x_{R2}. \end{aligned} \tag{4.6.14}$$

When the left and right eigenvectors are scaled so that (4.6.12) holds, Lemma 4 and Theorem 4 tell us that the set of solutions to (4.6.10) and (4.6.11) is given by

$$\begin{aligned} V_L &= \left(I - \frac{z_L z_L^T}{z_L^T z_L} \right) B_R A^+ + U_L (I - AA^+) + \frac{z_L}{z_L^T z_L} w_L^T, \\ V_R &= \left(I - \frac{z_R z_R^T}{z_R^T z_R} \right) B_L A^+ + U_R (I - AA^+) + \frac{z_R}{z_R^T z_R} w_R^T, \end{aligned}$$

where $U_L, U_R \in \mathbb{R}^{n \times m}$ are any matrices such that $z_S^T U_S = 0$, $S = L, R$.

The matrices V_L and V_R together with a_L and a_R in (4.6.9) define an SPT (T_L, T_R) that transforms $Q(\lambda)$ into $Q_1(\lambda)$ such that (4.6.1) holds.

4.6.3 Non Parallel Left (Right) Eigenvectors and Parallel Right (Left) Eigenvectors

When for example $\text{rank}([\, x_{L1}, x_{L2} \,]) = 1$ and $\text{rank}([\, x_{R1}, x_{R2} \,]) = 2$ we might want to look for an SPT of the form (I_{2n}, T_R) with T_R a class two elementary SPT,

since the left eigenvectors are already parallel to each other. Unfortunately, the pair (I_{2n}, T_R) is not structure preserving. We can however still use the procedure described in Section 4.6.2 to map $(\lambda_j, x_{Rj}, x_{Lj})$ to (λ_j, z_R, z_L), $j = 1, 2$ as long as we make sure that after the scaling (4.6.14), the vector $X_L p_L$ is nonzero so that a_L in (4.6.9) is defined. If $X_L p_L = 0$ then we replace x_{L1} by γx_{L1} and x_{R1} by γx_{R1}, where $\gamma = -1$ for real eigentriples and $\gamma = i$ for complex conjugate eigenpairs so that (4.6.14) still holds but $X_L p_L$ is nonzero.

4.7 Numerical Experiments

We now describe some numerical experiments designed to give insight into our deflation procedure. It is not our aim to investigate the numerical stability properties of the procedure. This is a separate issue that will be addressed in future work.

In all our experiments we take $U = 0$ in (4.4.4).

Experiment 1. Our first example is a 2×2 quadratic $Q(\lambda) = \lambda^2 M + \lambda C + K$ defined by

$$M = \begin{bmatrix} 2 & -1 \\ -1 & 3 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad K = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \tag{4.7.1}$$

with $\Lambda(Q) = \{-0.34 \pm 1.84i, 0.14 \pm 0.51i\}$ to two decimal places. Note that $M^{-1}C$ does not commute with $M^{-1}K$ thus $Q(\lambda)$ is not proportionally damped. Therefore the system cannot be decoupled by a 2×2 congruence transformation directly applied to $Q(\lambda)$.

Given the pair of complex conjugate eigenvalues $\lambda_{1,2} = -0.34 \pm 1.84i$ and their associated eigenvectors our symmetric deflation procedure, decouples $Q(\lambda)$ into

$$\lambda^2 \begin{bmatrix} 5.6 & 2.0\text{e}{-16} \\ 2.0\text{e}{-16} & -1.4\text{e}{-1} \end{bmatrix} + \lambda \begin{bmatrix} -1.6 & -9.4\text{e}{-16} \\ -9.4\text{e}{-16} & -9.3\text{e}{-2} \end{bmatrix} + \begin{bmatrix} 1.6 & -9.8\text{e}{-17} \\ -9.8\text{e}{-17} & -4.8\text{e}{-1} \end{bmatrix}$$

Table 4.1: Relative magnitude of the off-diagonal elements of the deflated quadratic $Q_2(\lambda) = \lambda^2 M_2 + \lambda C_2 + K_2$ experiment 2 and condition number of the transformations.

Deflated eigenvalues	off(M_2)	off(C_2)	off(K_2)	$\kappa_2(T_L)$	$\kappa_2(T_R)$	$\kappa_2(G_L)$	$\kappa_2(G_R)$
Real	3.0e-15	1.7e-13	1.6e-13	6.0e+5	2.0e+2	3.6e+1	3.3e+0
Complex	2.0e-16	1.4e-14	5.6e-14	1.8e+3	4.5e+1	1.0	1.1

to two significant digits with $\kappa_2(T) = 7.9$ and $\kappa_2(G) \approx 1$.

Experiment 2. Our second example is a 2×2 quadratic matrix polynomial arising in the study of the dynamic behaviour of a bicycle [50]. The coefficient matrices are nonsymmetric. They can be generated using the NLEVP MATLAB toolbox [7] via `nlevp('bicycle')`. This quadratic has two real eigenvalues, $\lambda_1 = -0.32$ and $\lambda_2 \approx -14$ and two complex conjugate eigenvalues $-0.78 \pm 4.5i$. Table 4.1 shows that the left and right transformations corresponding to the deflation of the complex conjugate eigentriples have a smaller condition number than that used for the deflation of the real eigentriples. The large condition number of T_L in the real case affects the size of the off-diagonal elements of the deflated quadratic. Here $\text{off}(E) = \|E - \text{diag}(E)\|_2 / \|E\|_2$, $E = M_2, C_2, K_2$.

Experiment 3. Our next example is a 4×4 hyperbolic symmetric quadratic eigenvalue problem generated as in [24, Sec. 6]. The eigenvalues, real since the quadratic is hyperbolic, are uniformly distributed between 1 and 8. Since this problem is overdamped, the eigenvalues are real and if we order them increasingly then $\lambda_1, \ldots, \lambda_4$ have negative type and $\lambda_5, \ldots \lambda_8$ have positive type [5, Proof of Thm. 1]. Any pairs (λ_j, λ_k) with $1 \le j \le 4$ and $5 \le k \le 8$ can be deflated from the quadratic. Table 4.2 displays the condition numbers of the SPT T and deflating transformation G for different pairings. It shows that the choice of pairings affects the conditioning of the transformations.

Table 4.2: Condition numbers of the SPTs T and deflating transformations G for different pairs of eigenvalues for experiment 4.

	(λ_1, λ_5)	(λ_1, λ_6)	(λ_1, λ_7)	(λ_1, λ_8)
$\kappa_2(T)$	4.62e+1	1.43e+3	4.41e+2	7.15e+1
$\kappa_2(L)$	2.09e+0	6.41e+0	1.61e+0	4.61e+0

Experiment 4. We now consider a symmetric quadratic eigenvalue problem coming from a model describing the motion of a beam simply supported at both ends and damped at the midpoint. This quadratic can be generated via the command `nlevp('damped_beam',nele)`, where `nele` is the number of finite elements. It is shown in [33, Thm. A1] that the damped problem $Q(\lambda) = \lambda^2 M + \lambda C + K$ and the undamped problem $Q_u(\lambda) = \lambda^2 M + K$ have n eigenvalues and n eigenvectors in common: those corresponding to the anti-symmetric modes. Because M and K are positive definite, the eigenvalues of $Q_u(\lambda)$ are pure imaginary; they come in pairs $(\lambda, \bar{\lambda})$, each pair sharing the same eigenvector.

We computed the n eigenpairs corresponding to the anti-symmetric modes of $Q_u(\lambda)$ using MATLAB function `eig` with the option `'chol'` and deflated all of them from $Q(\lambda)$ using the procedure described in section 4.5.1. Let

$$\widetilde{Q}(\lambda) = G_{acc}^T Q(\lambda) G_{acc} = \lambda^2 \widetilde{M} + \lambda \widetilde{C} + \widetilde{K}$$

be the deflated quadratic, where G_{acc} is the matrix which accumulates the product of the $n/2$ deflating transformations of the form (4.5.3) and $\widetilde{M}, \widetilde{C}, \widetilde{K}$ are block 2×2 diagonal with $(n/2) \times (n/2)$ blocks, the lower block being diagonal. Table 4.3 displays the scaled residuals $\mathrm{res}(M), \mathrm{res}(C)$, and $\mathrm{res}(K)$, where

$$\mathrm{res}(E) = \frac{\|G_{acc}^T E G_{acc} - \widetilde{E}\|_2}{\|G_{acc}\|_2^2 \|E\|_2 + \|\widetilde{E}\|_2},$$

and the 2-norm condition numbers $\kappa_2(G_{acc})$ for different values of $n = 2 \times$ `nele`.

Table 4.3: Scaled residuals and condition numbers for transformations in Experiment 4.

n	res(M)	res(C)	res(K)	$\kappa_2(G_{acc})$	$\kappa_2(U)$
8	3.07e-15	4.63e-18	3.90e-16	1.69e+1	1.52e+1
16	5.52e-15	5.08e-17	3.59e-15	4.47e+1	3.79e+1
32	1.34e-13	3.15e-16	1.68e-14	9.57e+1	7.84e+1
64	3.22e-12	6.09e-15	3.56e-14	1.95e+2	1.57e+2

The quadratic of the beam problem can be block diagonalized as (see [33, Appendix A1])

$$U^T Q(\lambda) U = \left[\begin{array}{c|c} \lambda^2 M_1 + \lambda D_1 + K_1 & 0 \\ \hline 0 & \lambda^2 M_2 + K_2 \end{array}\right],$$

where U is orthogonal, M_2 and K_2 are both symmetric positive definite and $\lambda^2 M_2 + K_2$ contains the anti-symmetric modes. The last column of Table 4.3 displays the condition number of the transformation U that block diagonalizes $\lambda^2 M_2 + K_2$. As a comparison, we note that $\kappa_2(G_{acc})$ is not much larger than $\kappa_2(U)$.

4.8 Proof of Lemma 3, Symmetric Quadratics

In this section we give the proof of Lemma 3. We start by recalling the notation. Let (λ_1, x_1) and (λ_2, x_2) be two eigenpairs of a symmetric quadratic $Q(\lambda) = \lambda^2 M + \lambda C + K$ such that $\lambda_1 \neq \lambda_2$. For real eigenpairs let $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2)$ and let $X = [x_1\ x_2]$. For complex conjugate eigenpairs let $\Lambda = \left[\begin{smallmatrix} \alpha & \beta \\ -\beta & \alpha \end{smallmatrix}\right]$ and $X = [u\ v]$, where $\lambda_1 = \bar{\lambda}_2 = \alpha + i\beta$, $\beta \neq 0$ and $x_1 = \bar{x}_2 = u + iv$. Let

$$p = \gamma(\lambda_1 - \lambda_2)^{-1} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad q = \Lambda p - (\lambda_1 + \lambda_2) p$$

with $\gamma = 1$ for real eigenpairs and $\gamma = i$ for complex eigenpairs and let

$$\begin{aligned} f^T z &= \|Xp\|_2 \neq 0, & a &= (f^T z)^{-1} Xp, \\ b^T z &= (e_\ell^T X \Lambda p - 1)/(e_\ell^T a), & z &= X\Lambda p - (b^T z)a, \\ d^T z &= (e_\ell^T X \Lambda q)/(e_\ell^T a), & h^T z &= (e_\ell^T X q - 1)/(e_\ell^T a), \end{aligned}$$

where ℓ is such that $a_\ell = e_\ell^T a \neq 0$. Define

$$A = \begin{bmatrix} \alpha_M & \frac{1}{2}\alpha_C & 0 \\ 0 & \alpha_M & \frac{1}{2}\alpha_C \\ \frac{1}{2}\alpha_C & \alpha_K & 0 \\ 0 & \frac{1}{2}\alpha_C & \alpha_K \end{bmatrix}, \quad \begin{aligned} B &= -\begin{bmatrix} Ma & Ca & Ka \end{bmatrix}, \\ V &= \begin{bmatrix} b & d & f & h \end{bmatrix}, \\ w^T &= \begin{bmatrix} b^T z & d^T z & f^T z & h^T z \end{bmatrix}, \end{aligned}$$

where $\alpha_M = a^T M a$, $\alpha_C = a^T C a$ and $\alpha_K = a^T K a$. The next lemma contains useful relations.

Lemma 5. *The following relations hold.*

$$x_1^T C x_2 = c\, x_1^T M x_2, \tag{4.8.1}$$

$$x_1^T K x_2 = k\, x_1^T M x_2, \tag{4.8.2}$$

$$d^T z = -k\, f^T z, \tag{4.8.3}$$

$$h^T z - b^T z = c\, f^T z, \tag{4.8.4}$$

where $c = -(\lambda_1 + \lambda_2)$ and $k = \lambda_1 \lambda_2$. In addition, for any symmetric matrix E we have

$$a^T E a = \alpha_E = (f^T z)^{-2} p^T X^T E X p, \tag{4.8.5}$$

$$z^T E a = (f^T z)^{-1} p^T \Lambda^T X^T E X p - (b^T z)(f^T z)^{-2} p^T X^T E X p, \tag{4.8.6}$$

with

$$p^T X^T E X p = \begin{cases} \mu(x_1^T E x_1 + x_2^T E x_2 - 2x_1^T E x_2) & \textit{for real eigenpairs,} \\ \frac{\mu}{4}(ix_1^T E x_1 - ix_2^T E x_2 + 2x_1^T E x_2) & \textit{otherwise,} \end{cases} \tag{4.8.7}$$

$$p^T \Lambda^T X^T E X p = \begin{cases} \mu(\lambda_1 x_1^T E x_1 + \lambda_2 x_2^T E x_2 + c x_2^T E x_1) & \textit{for real eigenpairs,} \\ \frac{\mu}{4}(i\lambda_1 x_1^T E x_1 - i\lambda_2 x_2^T E x_2 - c x_2^T E x_1) & \textit{otherwise,} \end{cases} \tag{4.8.8}$$

where $\mu = (\lambda_1 - \lambda_2)^{-2} \neq 0$ *is defined since* $\lambda_1 \neq \lambda_2$.

Proof. The relations (4.8.1) and (4.8.2) follow from $x_1^T Q(\lambda_1) x_2 = x_2^T Q(\lambda_1) x_1 = 0$ and $x_1^T Q(\lambda_2) x_2 = 0$. The relations (4.8.3)–(4.8.6) follow from the definition of p, q, a and z and (4.8.7)–(4.8.8) follow from the definition of Λ and X and p. □

With these relations in hand we can now prove the formula for $g^T = w^T A - z^T B$ in Lemma 3. From the definition of A, B w and z we find that

$$g = \begin{bmatrix} (b^T z)\alpha_M + \frac{1}{2}(f^T z)\alpha_C + z^T M a \\ \frac{1}{2}(b^T z)\alpha_C + (d^T z)\alpha_M + (f^T z)\alpha_k + \frac{1}{2}(h^T z)\alpha_C + z^T C a \\ \frac{1}{2}(d^T z)\alpha_C + \alpha_K h^T z + z^T K a \end{bmatrix}.$$

Using (4.8.5) with $E = M$ and $E = C$ and (4.8.6) with $E = M$ we obtain that the first component of g satisfies

$$2(f^T z) g_1 = p^T X^T C X p + 2 p^T \Lambda^T X^T M X p. \tag{4.8.9}$$

In a similar way we find that the other components of g satisfy

$$\begin{aligned} 2(f^T z) g_2 &= c p^T X C X p - 2k p^T X M X p + 2 p^T \Lambda^T X^T C X p + 2 p^T X K X p, \\ 2(f^T z) g_3 &= -k p^T X^T C X p + 2c p^T X^T K X p + 2 p^T \Lambda^T X^T K X p. \end{aligned}$$

Using (4.8.7) and (4.8.8) with $E = M, C$ and K and the relations (4.8.1)–(4.8.4) we find that for real eigenpairs,

$$2(f^T z)g^T = \mu\big(x_1^T Q'(\lambda_1)x_1 + x_2^T Q'(\lambda_2)x_2\big)\begin{bmatrix}1 & c & k\end{bmatrix}$$

and that for complex conjugate eigenpairs,

$$2(f^T z)g^T = \frac{i}{4}\mu\big(x_1^T Q'(\lambda_1)x_1 - x_2^T Q'(\lambda_2)x_2\big)\begin{bmatrix}1 & c & k\end{bmatrix}.$$

4.9 Proof of Lemma 4, Nonsymmetric Quadratics

We start by recalling the notation. In the real case $(\lambda_1, x_{L1}, x_{R1})$ and $(\lambda_2, x_{L2}, x_{R2})$ are two real eigentriples of a nonsymmetric quadratic $Q(\lambda) = \lambda^2 M + \lambda C + K$ such that $\lambda_1 \neq \lambda_2$. Since the eigenpairs are real let $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2)$ and let $X_L = [x_{L1}\ x_{L2}]$, and $X_R = [x_{R1}\ x_{R2}]$.

In the complex case we have two complex conjugate eigentriples $(\lambda_1, x_{L1}, x_{R1})$ and $(\lambda_2, x_{L2}, x_{R2})$ (where $\lambda_2 = \bar{\lambda}_1$) and let $\Lambda = \left[\begin{smallmatrix}\alpha & \beta\\ -\beta & \alpha\end{smallmatrix}\right]$ and $X_S = [u_S\ v_S],\ S = L, R$, where $\lambda_1 = \bar{\lambda}_2 = \alpha + i\beta,\ \beta \neq 0$ and

$$X_S = \begin{bmatrix}x_{S1}\\ x_{S2}\end{bmatrix} = \begin{bmatrix}u_S + iv_S\\ u_S - iv_S\end{bmatrix},\ S = L, R$$

$x_1 = \bar{x}_2 = u + iv$. Let

$$p = \gamma(\lambda_1 - \lambda_2)^{-1}\begin{bmatrix}1\\ -1\end{bmatrix} = p_L, \qquad q = \Lambda p - (\lambda_1 + \lambda_2)p$$

with $\gamma = 1$ for real eigenpairs $p = p_R = p_L$)and $\gamma = i$ for complex eigenpairs (with $p_R = -p_L$) and let

$$\begin{aligned} f_S^T z_S &= \|X_S p_S\|_2 \neq 0, & a_S &= (f_S^T z_S)^{-1} X_S p_S, \\ b_S^T z_S &= (e_{\ell_S}^T X_S \Lambda_S p_S - 1)/(e_{\ell_S}^T a_S), & z_S &= X_S \Lambda_S p_S - (b_S^T z_S) a_S, \\ d_S^T z_S &= (e_{\ell_S}^T X_S \Lambda_S q_S)/(e_{\ell_S}^T a_S), & h_S^T z_S &= (e_{\ell_S}^T X_S q_S - 1)/(e_{\ell_S}^T a_S), \end{aligned} \tag{4.9.1}$$

where

$$a_S^T a_S = 1, \qquad e_{\ell_S}^T z_S = 1, \qquad |e_{\ell_S}^T a_S| = \|a_S\|_\infty, \tag{4.9.2}$$

and $p_S, q_S \in \mathbb{R}^2$ are such that

$$e^T p_S = 0, \qquad e^T \Lambda_S p_S = 1, \qquad e^T q_S = 1, \qquad e^T \Lambda_S q_S = 0.$$

Define

$$A = \begin{bmatrix} \alpha_M & \frac{1}{2}\alpha_C & 0 \\ 0 & \alpha_M & \frac{1}{2}\alpha_C \\ \frac{1}{2}\alpha_C & \alpha_K & 0 \\ 0 & \frac{1}{2}\alpha_C & \alpha_K \end{bmatrix}, \qquad \begin{aligned} B_R &= -\begin{bmatrix} Ma_R & Ca_R & Ka_R \end{bmatrix}, \\ B_L &= -\begin{bmatrix} M^T a_L & C^T a_L & K^T a_L \end{bmatrix}, \\ w_R^T &= \begin{bmatrix} b_R^T z_R & d_R^T z_R & f_R^T z_R & h_R^T z_R \end{bmatrix}, \\ w_L^T &= \begin{bmatrix} b_L^T z_L & d_L^T z_L & f_L^T z_L & h_L^T z_L \end{bmatrix}, \\ V_S &= \begin{bmatrix} b_S & d_S & f_S & h_S \end{bmatrix}, \quad S = L, R, \end{aligned}$$

where $\alpha_M = a_L^T M a_R$, $\alpha_C = a_L^T C a_R$ and $\alpha_K = a_L^T K a_R$. The next lemma contains useful relations.

Lemma 6. *The following relations hold,*

$$x_{L1}^T K x_{R1} = k x_{L1}^T M x_{R2}, \quad x_{L2}^T K x_{R1} = k x_{L2}^T M x_{R1} \tag{4.9.3}$$

$$x_{L1}^T C x_{R2} = c x_{L1}^T M x_{R2}, \quad x_{L2}^T C x_{R1} = c x_{L2}^T M x_{R1} \tag{4.9.4}$$

$$x_{Lj}^T K x_{Rj} = -\lambda_j^2 x_{Lj}^T M x_{Rj} - \lambda_j x_{Lj}^T C x_{Rj}, \text{ for } j = 1, 2 \tag{4.9.5}$$

and

$$d_S^T z_S = -k\, f_S^T z_S, \tag{4.9.6}$$

$$h_S^T z_S - b_S^T z_S = c\, f_S^T z_S, \tag{4.9.7}$$

where $S = L, R$, $c = -(\lambda_1 + \lambda_2)$ *and* $k = \lambda_1\lambda_2$. *Also for any matrix* E *we have*

$$a_L^T E a_R = \alpha_E = (f_L^T z_L)^{-1}(f_R^T z_R)^{-1} p_L^T X_L^T E X_R\, p_R \tag{4.9.8}$$

$$z_R^T E a_L = (f_L^T z_L)^{-1} p_R^T \Lambda_R X_R^T E X_L\, p_L \tag{4.9.9}$$

$$- \quad (b_R^T z_R)(f_L^T z_L)^{-1}(f_R^T z_R)^{-1} p_R^T X_R^T E X_L\, p_L \tag{4.9.10}$$

$$z_L^T E a_R = (f_R^T z_R)^{-1} p_L^T \Lambda_R X_L^T E X_R\, p_R \tag{4.9.11}$$

$$- \quad (b_L^T z_L)(f_L^T z_L)^{-1}(f_R^T z_R)^{-1} p_L^T X_L^T E X_R\, p_R \tag{4.9.12}$$

where for real eigenpairs replace p_R *and* p_L *by* p *(since* $p = p_L = p_R$*) and for*

$$p_L^T X_L^T E X_R\, p_R = \tag{4.9.13}$$

$$\begin{cases} \mu(x_{L1}^T E x_{R1} - x_{L1}^T E x_{R2} - x_{L2}^T E x_{R1} + x_{L2}^T E x_{R2}) & \textit{real case,} \\ \frac{\mu}{2}(i x_{L1}^T E x_{R1} + x_{L1}^T E x_{R2} + x_{L2}^T E x_{R1} - i x_{L2}^T E x_{R2}) & \textit{otherwise,} \end{cases} \tag{4.9.14}$$

$$p_L^T X_L^T E X_R \Lambda p_R = \tag{4.9.15}$$

$$\begin{cases} \mu(\lambda_1 x_{L1}^T E x_{R1} - \lambda_2 x_{L1}^T E x_{R2} - \lambda_1 x_{L2}^T E x_{R1} + \lambda_2 x_{L2}^T E x_{R2}) & \textit{real case,} \\ \frac{\mu}{2}(i\lambda_1 x_{L1}^T E x_{R1} + \lambda_2 x_{L1}^T E x_{R2} + \lambda_1 x_{L2}^T E x_{R1} - i\lambda_2 x_{L2}^T E x_{R2}) & \textit{otherwise,} \end{cases} \tag{4.9.16}$$

where $\mu = (\lambda_1 - \lambda_2)^{-2} \neq 0$ *is defined since* $\lambda_1 \neq \lambda_2$.

Proof. The relations (4.9.3) and (4.9.4) follow from $x_{L2}^T Q(\lambda_1) x_{R1} = x_{L2}^T Q(\lambda_2) x_{R1} = 0$. The relations (4.9.6)–(4.9.12) follow from the definition of p, q, a and z and (4.9.14)–(4.9.16) follow from the definition of Λ and X and p.

For the complex nonsymmetric case, we have the two eigenvalues $\lambda_1 = \lambda$ and $\lambda_2 = \bar{\lambda}_2$ with left eigenvectors $x_{L1} = y = u_L + iv_L$, $x_{L2} = \bar{y} = u_L - iv_L$ and right eigenvectors $x_{R1} = x = u_R + iv_R$ and $x_{R2} = \bar{x} = u_R - iv_R$, where $X_S =$

$[u_S,\, v_S],\ S = L, R.$

Relations (4.9.14)—(4.9.16) for the complex case follow from substituting the expressions below

$$\Lambda_R \leftarrow W^* \mathrm{diag}(\lambda_1, \lambda_2) W, \quad X_R \leftarrow \tfrac{1}{2}[x,\, \bar{x}]W, \quad p_R = \frac{-i}{(\lambda_1 - \lambda_2)} \begin{bmatrix} 1 \\ & -1 \end{bmatrix} W, \qquad (4.9.17)$$

$$\Lambda_L \leftarrow \Lambda_R^T, \quad X_L \leftarrow \tfrac{1}{2}[y,\, \bar{y}]W \quad p_L = \frac{i}{(\lambda_1 - \lambda_2)} \begin{bmatrix} 1 \\ & -1 \end{bmatrix} W, \qquad (4.9.18)$$

where $W = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix}$ with $W^*W = I$. □

With these relations in hand we can now prove the formula for $W_R^T A - z_R^T B_L = g_R^T$ in Lemma 4.6.12. We omit the proof of the formula for $W_L^T A - z_L^T B_R = g_L^T$ which is almost identical.

For the real nonsymmetric case, we show that

$$g_R^T = \gamma_R(\xi_1 + \xi_2 - \xi_5 - \xi_6)[1, c, k]$$

where $\xi_1 = x_{L1}^T(2\lambda_1 M + C)x_{R1}$, $\xi_2 = x_{L2}^T(2\lambda_2 M + C)x_{R2}$, $\xi_5 = x_{L1}^T(2\lambda_2 M + C)x_{R2}$, $\xi_6 = x_{L2}^T(2\lambda_1 M + C)x_{R1}$ and $\gamma_R = \frac{1}{2}(f_L^T z_L)^{-1}$. Let $g_R^T = [g_{R1},\, g_{R2},\, g_{R3}]^T$ and

$$\begin{aligned} g_{R1} &= b_R^T z_R \alpha_M + \frac{1}{2}\alpha_C f_R^T z_R + z^T M^T a_L \\ g_{R2} &= b_R^T z_R \alpha_C + d_R^T z_R \alpha_M + f_R^T z_R \alpha_K + \frac{1}{2} h_R^T z_R \alpha_C + z_R^T C^T a_L \\ g_{R3} &= \frac{1}{2} d_R^T z_R \alpha_C + h_R^T z_R \alpha_K + z_R^T K^T a_L. \end{aligned}$$

For the first element of g_R, g_{R1} we use (4.9.8) and (4.9.12)

$$\begin{aligned} g_{R1} &= b_R^T z_R \alpha_M + \frac{1}{2}\alpha_C f_R^T z_R + z^T M^T a_L \\ 2(f_L^T z_L) g_{R1} &= p^T X_L^T C X_R p + 2p^T \Lambda^T X_L^T M X_R p \end{aligned}$$

then applying (4.9.14) and (4.9.16)

$$\begin{aligned} 2(f_L^T z_L) g_{R1} = \mu(&x_{L1}^T C x_{R1} + x_{L2}^T C x_{R2} - x_{L1}^T C x_{R2} - x_{L2}^T C x_{R1}) \\ &+ 2\mu(\lambda_1 x_{L1}^T M x_{R1} - \lambda_1 x_{L1}^T M x_{R2} - \lambda_2 x_{L2}^T M x_{R1} + \lambda_2 x_{L2}^T M x_{R2}) \end{aligned} \tag{4.9.19}$$

which simplifies to

$$\begin{aligned} &= \mu(x_{L1}^T (2\lambda_1 M + C) x_{R1} + x_{L2}^T (2\lambda_2 M + C) x_{R2} \\ &- x_{L2}^T (2\lambda_1 M + C) x_{R1} - x_{L1}^T (2\lambda_2 M + C) x_{R2}) \\ &= \mu(x_{L1}^T Q'(\lambda_1) x_{R1} + x_{L2}^T Q'(\lambda_2) x_{R2} \\ &- x_{L2}^T Q'(\lambda_1) x_{R1} - x_{L1}^T Q'(\lambda_2) x_{R2}) \\ 2(f_L^T z_L) g_{R1} &= \mu(\xi_1 + \xi_2 - \xi_5 - \xi_6). \end{aligned} \tag{4.9.20}$$

For the second element of g_R, g_{R2} we first use (4.9.6)–(4.9.7), (4.9.8) and (4.9.12)

$$\begin{aligned} g_{R2} &= b_R^T z_R \alpha_C + \alpha_M d_R^T z_R + \alpha_K f_R^T z_R + \frac{1}{2}\alpha_C h_R^T z_R + z^T M^T a_L \\ 2(f_L^T z_L) g_{R2} &= c p^T X_L^T C X_R p + 2p^T X_L^T K X_R p - 2k p^T X_L^T M X_R p + 2p^T \Lambda^T X_L^T M X_R p \end{aligned} \tag{4.9.21}$$

then, applying (4.9.14) and (4.9.16) we obtain

$$\begin{aligned}
2(f_L^T z_L)g_{R2} &= \mu\Big(-(\lambda_1+\lambda_2)(x_{L1}^T Cx_{R1} - x_{L1}^T Cx_{R2} - x_{L2}^T Cx_{R1} + x_{L2}^T Cx_{R2}) \\
&\quad + 2(x_{L1}^T Kx_{R1} - x_{L1}^T Kx_{R2} - x_{L2}^T Kx_{R1} + x_{L2}^T Kx_{R2}) \\
&\quad - 2\lambda_1\lambda_2(x_{L1}^T Mx_{R1} - x_{L1}^T Mx_{R2} - x_{L2}^T Mx_{R1} + x_{L2}^T Mx_{R2}) \\
&\quad + 2(\lambda_1 x_{L1}^T Cx_{R1} - \lambda_1 x_{L1}^T Cx_{R2} - \lambda_2 x_{L2}^T Cx_{R1} + \lambda_2 x_{L2}^T Cx_{R2})\Big) \\
&= \mu\Big(-(\lambda_1+\lambda_2)(x_{L1}^T Cx_{R1} - x_{L1}^T Cx_{R2} - x_{L2}^T Cx_{R1} + x_{L2}^T Cx_{R2}) \\
&\quad + 2(-\lambda_1^2 x_{L1}^T Mx_{R1} - \lambda_1 x_{L1}^T Cx_{R1} - kx_{L1}^T Mx_{R2} - kx_{L2}^T Mx_{R1} \\
&\quad - \lambda_2^2 x_{L2}^T Mx_{R2} - \lambda_2 x_{L2}^T Cx_{R2}) \\
&\quad - 2k(x_{L1}^T Mx_{R1} - x_{L1}^T Mx_{R2} - x_{L2}^T Mx_{R1} + x_{L2}^T Mx_{R2}) \\
&\quad + 2(\lambda_1 x_{L1}^T Cx_{R1} - \lambda_2 x_{L1}^T Cx_{R2} - \lambda_1 x_{L2}^T Cx_{R1} + \lambda_2 x_{L2}^T Cx_{R2})\Big).
\end{aligned} \tag{4.9.22}$$

Finally using (4.9.3)—(4.9.5) we have

$$\begin{aligned}
2(f_L^T z_L)g_{R2} &= \mu\Big(cx_{L1}^T Q'(\lambda_1)x_{R1} - cx_{L1}^T Q'(\lambda_2)x_{R2} - cx_{L2}^T Q'(\lambda_1)x_{R1} + cx_{L2}^T Q'(\lambda_2)x_{R2}\Big) \\
&= \mu c(\xi_1 + \xi_2 - \xi_5 - \xi_6).
\end{aligned} \tag{4.9.23}$$

For the final element g_{R3}, we first use (4.9.6)—(4.9.12),

$$\begin{aligned}
g_{R3} &= d_R^T z_R \frac{1}{2}\alpha_C + h_R^T z_R \alpha_K + z_R^T K^T a_L \\
2(f_L^T z_L)g_{R3} &= cp^T X_L^T K X_R p + 2p^T \Lambda^T X_R K^T X_L p - 2kp^T X_L^T C X_R p
\end{aligned} \tag{4.9.24}$$

applying (4.9.14) and (4.9.16) yields

$$\begin{aligned}
&= \mu\Big(-(\lambda_1+\lambda_2)(x_{L1}^T K x_{R1} + x_{L2}^T K x_{R2} - x_{L2}^T K x_{R1} - x_{L1}^T K x_{R2}) \\
&+ 2(\lambda_1 x_{L1}^T K x_{R1} + \lambda_2 x_{L2}^T K x_{R2} - \lambda_2 x_{L1}^T K x_{R2} - \lambda_1 x_{L2}^T K x_{R1}) \\
&- 2\lambda_1\lambda_2(x_{L1}^T C x_{R1} + x_{L2}^T C x_{R2} - x_{L2}^T C x_{R1} - x_{L1}^T C x_{R2}\Big) \\
2(f_L^T z_L) g_{R3} &= \mu\Big(-2\lambda_2 x_{L1}^T K x_{R1} + 2\lambda_1 x_{L1}^T K x_{R2} + 2\lambda_2 x_{L2}^T K x_{R1} - 2\lambda_1 x_{L2}^T K x_{R2} \\
&- \lambda_1\lambda_2(x_{L1}^T C x_{R1} + x_{L2}^T C x_{R2} - x_{L2}^T C x_{R1} - x_{L1}^T C x_{R2})\Big).
\end{aligned} \tag{4.9.25}$$

Next we use the relations (4.9.26)—(4.9.27) below

$$-x_{L1}^T K x_{R1} = \lambda_1^2 x_{L1}^T M x_{R1} + \lambda_1 x_{L1}^T C x_{R1}, \quad -x_{L2}^T K x_{R2} = \lambda_1^2 x_{L2}^T M x_{R2} + \lambda_1 x_{L2}^T C x_{R2}, \tag{4.9.26}$$

$$-x_{L1}^T K x_{R2} = \lambda_2^2 x_{L1}^T M x_{R2} + \lambda_2 x_{L1}^T C x_{R2}, \quad -x_{L2}^T K x_{R1} = \lambda_2^2 x_{L2}^T M x_{R1} + \lambda_2 x_{L2}^T C x_{R1}, \tag{4.9.27}$$

to obtain

$$\begin{aligned}
2(f_L^T z_L) g_{R3} &= \mu\Big(2\lambda_2(\lambda_1^2 x_{L1}^T M x_{R1} + \lambda_1 x_{L1}^T C x_{R1}) - 2\lambda_2(\lambda_1^2 x_{L2}^T M x_{R1} + \lambda_1 x_{L2}^T C x_{R1}) \\
&- 2\lambda_1(\lambda_2^2 x_{L1}^T M x_{R2} + \lambda_2 x_{L1}^T C x_{R2}) 2\lambda_1(\lambda_2^2 x_{L2}^T M x_{R2} + \lambda_2 x_{L2}^T C x_{R2}) \\
&- \lambda_1\lambda_2(x_{L1}^T C x_{R1} - x_{L1}^T C x_{R2} - x_{L2}^T C x_{R1} + x_{L2}^T C x_{R2})\Big) \\
&= \lambda_1\lambda_2\mu\Big(x_{L1}^T(2\lambda_1 M + C) x_{R1} + x_{L2}^T(2\lambda_2 M + C) x_{R2} \\
&- x_{L2}^T(2\lambda_1 M + C) x_{R1} - x_{L1}^T(2\lambda_2 M + C) x_{R2}\Big) \\
2(f_L^T z_L) g_{R3} &= k\mu(\xi_1 + \xi_2 - \xi_5 - \xi_6).
\end{aligned} \tag{4.9.28}$$

For the complex nonsymmetric case the same idea applies, but we substitute the

modified expressions (4.9.17)—(4.9.18), thus

$$g_{R1} = b_R^T z_R \alpha_M + \frac{1}{2}\alpha_C f_R^T z_R + z^T M^T a_L$$
$$2(f_L^T z_L) g_{R1} = p_L^T X_L^T C X_R p_R + 2 p_L^T X_L^T M X_R \Lambda p_R$$

note that ($x = x_{R1}, \bar{x} = x_{R2}$, and $y = x_{L1}, \bar{y} = x_{L2}$) then applying (4.9.14) and (4.9.16)

$$\begin{aligned} 2(f_L^T z_L) g_{R1} &= \mu\Big((2iy^T C x + 2y^T C \bar{x} + 2\bar{y}^T C x - 2i\bar{y}^T C \bar{x}) \\ &\quad + 2(i2\lambda y^T M x + 2\bar{\lambda} y^T M \bar{x} + 2\lambda \bar{y}^T M x - i\bar{\lambda} 2\bar{y}^T M \bar{x}) \Big) \\ &= \mu(i\bar{y}^T Q'(\lambda) x - i y^T Q'(\bar{\lambda}) \bar{x} + y^T Q'(\bar{\lambda}) \bar{x} + \bar{y}^T Q'(\lambda) x \Big) \\ &= \mu\Big(i x_{L1}^T Q'(\lambda_1) x_{R1} - i x_{L2}^T Q'(\lambda_2) x_{R2} \\ &\quad + x_{L2}^T Q'(\lambda_2) x_{R1} + x_{L1}^T Q'(\lambda_1) x_{R2} \Big) \end{aligned} \tag{4.9.29}$$

$$2(f_L^T z_L) g_{R1} = \mu(i\xi_1 - i\xi_2 + \xi_5 + \xi_6). \tag{4.9.30}$$

g_{R2} and g_{R3} are obtained in a similar manner to the real case, we omit the details.

Chapter 5

Conclusions

Polynomial eigenvalue problems, considered in the first half of this thesis, are an important class of nonlinear eigenproblems that are less routinely solved than the standard eigenvalue problem $(\mathcal{A} - \lambda I)x = 0$ or generalized eigenvalue problem $(\mathcal{A} - \lambda\mathcal{B})x = 0$. Quadratic, and more generally, polynomial eigenvalue problems are usually converted to a degree one problem of larger dimension—the process of linearization.

In Chapters 1—2 we explained the linearization process, solution of the linear problem, and recovery of the solution of the polynomial problem from that of the linear problem. By considering a numerical example, we saw that appropriate handling of the problem is essential to returning accurate solutions, when extracting solutions of the polynomial problem from those of the linear problem.

In Chapter 3 we presented a general purpose eigensolver for dense QEPs, which incorporates recent contributions on the numerical solution of polynomial eigenvalue problems, namely a scaling of the eigenvalue parameter prior to the computation, [6], [14] and a choice of linearization with favourable conditioning and backward stability properties [30], [32], [33]. Our algorithm includes a preprocessing step that reveals the zero and infinite eigenvalues contributed by singular

leading and trailing matrix coefficients and deflates them. The preprocessing step may also detect nonregularity (although this is not guaranteed), indeed robustly detecting nonregularity of a quadratic is nontrivial and therefore future work. Our algorithm takes advantage of the block structure of the chosen linearization. Implemented as a MATLAB [49] function called `quadeig`, it makes use of functions from the NAG Toolbox for MATLAB [53]. Our eigensolver can in principle be extended to matrix polynomials of degree higher than two. The preprocessing step can easily be extended using the same type of linearization, merely of a higher degree matrix polynomial. For scaling of the eigenvalue parameter prior to the computation we can use the method described in Section 2.3.1 on page 44 [6], which extends the Fan, Lin and Van Dooren scaling for matrix polynomials of degree two.

Numerical examples were presented, illustrating the improved performance of this new algorithm `quadeig`, with the existing MATLAB routine `polyeig`, both in terms of accuracy and stability and reduced computational cost.

In Chapter 4 we described a structure preserving deflation procedure for quadratic matrix polynomials, that given two eigentriples (λ_j, x_j, y_j), $j = 1, 2$ satisfying appropriate conditions, decouples $Q(\lambda)$ into a quadratic $Q_d(\lambda) = \lambda^2 M_d + \lambda C_d + K_d$ of dimension $n - 1$ and a scalar quadratic $q(\lambda) = \lambda^2 m + \lambda c + k = m(\lambda - \lambda_1)(\lambda - \lambda_2)$ such that (a)

$$\Lambda(Q) = \Lambda(Q_d) \cup \{\lambda_1, \lambda_2\},$$

where $\Lambda(Q)$ denotes the spectrum of Q and (b) there exist well-defined relations between the eigenvectors of $Q(\lambda)$ and those of the decoupled quadratic

$$\widetilde{Q}(\lambda) = \begin{bmatrix} Q_d(\lambda) & 0 \\ 0 & q(\lambda) \end{bmatrix}. \tag{5.0.1}$$

This procedure applies to symmetric and nonsymmetric quadratics, and when the

quadratic is symmetric preserves the symmetry.

Numerical examples that illustrate our deflation procedure were also presented. To the best of our knowledge, this work is the first attempt at constructing a family of *nontrivial elementary SPTs that have a specific action of practical use: that of "mapping" two linearly independent eigenvectors to a set of linearly dependent eigenvectors.*

This structure preserving deflation method has application in the area of model updating. Model updating is the modification of an existing inaccurate model with measured data. The eigenvalue embedding problem is a special instance of model updating and can be defined as follows: consider a quadratic matrix polynomial

$$Q(\lambda) = \lambda^2 M + \lambda C + K$$

resulting from a second-order dynamical system with a few known eigenvalues λ_j, $j = 1\colon k$. Now suppose that new eigenvalues $\widehat{\lambda}_j$, $j = 1\colon k$ have been measured. There are several types of eigenvalue embedding problems but one of them consists of updating the quadratic $Q(\lambda)$ to a new quadratic $\widehat{Q}(\lambda)$ with eigenvalues $\widehat{\lambda}_j$, $j = 1\colon k$ replacing the eigenvalues λ_j, $j = 1\colon k$ of $Q(\lambda)$ while the remaining $2n - k$ eigenvalues of $\widehat{Q}(\lambda)$ are kept the same as those of the original problem $Q(\lambda)$. This is sometimes referred to as eigenvalue updating with no spill-over.

A number of solutions to this problem has been proposed often with additional constraints such as preservation of the symmetry of the coefficient matrices and preservation of the positive definiteness of the mass and stiffness matrices.

The deflation procedure in Chapter 4 can be used to update eigenvalues of a quadratic matrix polynomial. Further work involves investigating the potential of this process for updating systems, its reliability and performance in finite precision arithmetic, and comparison with existing techniques.

Bibliography

[1] A. Amiraslani, R. M. Corless, and P. Lancaster, *Linearization of matrix polynomials expressed in polynomial bases*, IMA J. Numer. Anal., 29 (2009), pp. 141–157.

[2] E. Anderson, Z. Bai, C. H. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. J. D. Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, and D. C. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, third ed., 1999.

[3] E. N. Antoniou and S. Vologiannidis, *A new family of companion forms of polynomial matrices*, Electron. J. Linear Algebra, 11 (2004), pp. 78–87.

[4] ——, *Linearizations of polynomial matrices with symmetries and their applications*, Electron. J. Linear Algebra, 15 (2006), pp. 107–114.

[5] L. Barkwell and P. Lancaster, *Overdamped and gyroscopic vibrating systems*, Journal of Applied Mechanics, 59 (1992), pp. 176–181.

[6] T. Betcke, *Optimal scaling of generalized and polynomial eigenvalue problems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1320–1338.

[7] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and

F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, MIMS EPrint 2008.40, Manchester Institute for Mathematical Sciences, University of Manchester, Manchester, UK, April 2008. `www.mims.manchester.ac.uk/research/numerical-analysis/nlevp.html`.

[8] K. W. Brodlie, *Corrections to matrices expressed in non-symmetric product form*, J. Inst. Math. Appl., 14 (1974), pp. 141–144.

[9] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols, *Feedback design for regularizing descriptor systems.*, Linear Algebra Appl., 299 (1999), pp. 119–151.

[10] T. F. Chan, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.

[11] M. T. Chu, T.-M. Hwang, and W.-W. Lin., *A novel deflation technique for solving quadratic eigenvalue problems*, Tech. Rep. 2005-3-002, National Center for Theoretical Sciences, National Tsing Hua University, Taiwan, 2005. NCTS Preprints in Mathematics.

[12] M. T. Chu and S.-F. Xu, *Spectral decomposition of real symmetric quadratic λ-matrices and its applications*, Mathematics of Computation, 78 (2008), pp. 293–313.

[13] J.-P. Dedieu and F. Tisseur, *Perturbation theory for homogeneous polynomial eigenvalue problems*, Linear Algebra Appl., 358 (2003), pp. 71–94.

[14] H.-Y. Fan, W.-W. Lin, and P. Van Dooren, *Normwise scaling of second order polynomial matrices*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 252–256.

[15] A. Feriani, F. Perotti, and V. Simoncini, *Iterative system solvers for the frequency analysis of linear mechanical systems*, Computer Methods in Applied Mechanics and Engineering, 190 (2000), pp. 1719–1739.

[16] S. D. Garvey, M. I. Friswell, and U. Prells, *Co-ordinate transformations for second order systems. I. General transformations*, J. Sound Vibration, 258 (2002), pp. 885–909.

[17] ——, *Co-ordinate transformations for second order systems. II. Elementary structure-preserving transformations*, J. Sound Vibration, 258 (2002), pp. 911–930.

[18] S. D. Garvey, F. Tisseur, M. I. Friswell, J. E. T. Penny, and U. Prells, *Simultaneous tridiagonalization of two symmetric matrices*, Internat. J. Numer. Methods Engrg., 57 (2003), pp. 1643–1660.

[19] S. Gaubert and M. Sharify, *Tropical scaling of polynomial matrices*, in Positive systems, vol. 389 of Lecture Notes in Control and Inform. Sci., Springer, Berlin, 2009, pp. 291–303.

[20] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009. Reprinted by SIAM.

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, USA, third ed., 1996.

[22] A. Gotts, *Report regarding model reduction, model compaction research project, University of Nottingham.* Unpublished manuscript, 2005.

[23] L. Grammont, N. J. Higham, and F. Tisseur, *A framework for analyzing nonlinear eigenproblems and parametrized linear systems*, Linear Algebra Appl., (2010). To appear.

[24] C.-H. Guo, N. J. Higham, and F. Tisseur, *Detecting and solving hyperbolic quadratic eigenvalue problems*, SIAM J. Matrix Anal. Appl., 30 (2009), pp. 1593–1613.

[25] S. Hammarling, *The singular value decomposition in multivariate statistics*, SIGNUM Newsl., 20 (1985), pp. 2–25.

[26] P. C. Hansen, R. D. Fierro, and P. S. Hansen, *UTV tools: Matlab templates for rank-revealing UTV decompositions*, Numerical Algorithms, 20 (1999), pp. 165–194.

[27] D. Herting, *MSC/Nastran Advanced Dynamic Analysis User's Guide*, MSC. Software Corporation, 1997.

[28] N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix*, in Reliable Numerical Computation, M. G. Cox and S. J. Hammarling, eds., Oxford University Press, 1990, pp. 161–185.

[29] ——, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second ed., 2002.

[30] N. J. Higham, R.-C. Li, and F. Tisseur, *Backward error of polynomial eigenvalue problems solved by linearization*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1218–1241.

[31] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Symmetric linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 29 (2006/07), pp. 143–159.

[32] N. J. Higham, D. S. Mackey, and F. Tisseur, *The conditioning of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1005–1028.

[33] N. J. Higham, D. S. Mackey, F. Tisseur, and S. D. Garvey, *Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems*, Internat. J. Numer. Methods Engrg., 73 (2008), pp. 344–360.

[34] N. J. Higham and F. Tisseur, *On structure preserving transformations for quadratic matrix polynomials*, MIMS EPrint, Manchester Institute for Mathematical Science, University of Manchester, Manchester M13 9PL, England, 2008. In preparation.

[35] A. Hilliges, C. Mehl, and V. Mehrmann, *On the solution of palindromic eigenvalue problems*, in Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), Jyväskylä, Finland, P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, eds., 2004, p. 11.

[36] B. Kågström and D. Kressner, *Multishift variants of the QZ algorithm with aggressive early deflation*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 199–227.

[37] T. R. Kowalski., *Extracting a few eigenpairs of symmetric indefinite matrix pencils*, PhD thesis, Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA, 2000.

[38] V. N. Kublanovskaya, V. B. Mikhailov, and V. B. Khazanov, *Eigenvalue problem for an irregular λ-matrix*, Journal of Mathematical Sciences, 13 (1980), pp. 251–260.

[39] Z. Kukelova, M. Bujnak, and T. Pajdla., *Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems*, in BMVC 2008: Proceedings of the 19th British Machine Vision Conference, M. Everingham, C. Needham, and R. Fraile, eds., vol. 1, Malvern, UK, 2008, pp. 565–574.

[40] P. Lancaster, *Inverse spectral problems for semisimple damped vibrating systems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 279–301.

[41] ——, *Linearization of regular matrix polynomials*, Electron. J. Linear Algebra, 17 (2008), pp. 21–27.

[42] P. Lancaster and P. Psarrakos, *A note on weak and strong linearizations of regular matrix polynomials*, Numerical Analysis Report No. 470, Manchester Centre for Computational Mathematics, Manchester, England, June 2005.

[43] P. Lancaster and I. Zaballa, *Diagonalizable quadratic eigenvalue problems*, Mechanical Systems and Signal Processing, 23 (2009), pp. 1134–1144.

[44] D. Lemonnier and P. Van Dooren, *Balancing regular matrix pencils*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 253–263.

[45] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, MIMS EPrint 2006.38, Manchester Institute for Mathematical Sciences, University of Manchester, Manchester, UK, March 2006. `http://eprints.ma.man.ac.uk/190`.

[46] ——, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1029–1051.

[47] ——, *Vector spaces of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.

[48] ——, *Numerical methods for palindromic eigenvalue problems: computing the anti-triangular Schur form*, Numer. Linear Algebra Appl., 16 (2009), pp. 63–86.

[49] *MATLAB*, The MathWorks, Inc., Natick, MA, USA. `http://www.mathworks.com`.

[50] J. P. Meijaard, J. M. Papadopoulos, A. Ruina, and A. L. Schwab, *Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 463 (2007), pp. 1955–1982.

[51] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[52] C. Munro, F. Tisseur, and S. Hammarling, *A general purpose algorithm for solving quadratic eigenvalue problems*. In preparation, 2010.

[53] *NAG Toolbox for MATLAB*, The Numerical Algorithms Group Ltd., Oxford, UK. `http://www.nag.com`.

[54] B. N. Parlett and C. Reinsch, *Balancing a matrix for calculation of eigenvalues and eigenvectors*, Numer. Math., 13 (1969), pp. 293–304.

[55] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, London, 1990.

[56] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.

[57] F. Tisseur, S. D. Garvey, and C. Munro, *Deflating quadratic matrix polynomials with structure preserving transformations*, Linear Algebra Appl., (2010). To appear.

[58] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[59] R. C. Ward, *Balancing the generalized eigenvalue problem*, SIAM J. Sci. Comput., 2 (1981), pp. 141–153.

[60] D. S. Watkins, *Performance of the* QZ *algorithm in the presence of infinite eigenvalues*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 364–375.

[61] ——, *Fundamentals of Matrix Computations*, Pure and Applied Mathematics, Wiley-Interscience [John Wiley & Sons], New York, 2002. Second editon.

[62] ——, *The Matrix Eigenvalue Problem*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.

[63] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.