Computation of invariant pairs and matrix solvents

Esteban Segura Ugalde

UNIVERSITÉ DE LIMOGES

ÉCOLE DOCTORALE Sciences et Ingénierie pour l'Information
FACULTÉ DES SCIENCES ET TECHNIQUES
Département de Mathématiques et Informatique
Laboratoire XLIM (UMR 7252)

Thèse

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline : Mathématiques et ses applications

présentée et soutenue par

Esteban SEGURA UGALDE

le 1 juillet 2015 à 14h

Computation of Invariant Pairs and Matrix Solvents

Thèse dirigée par Moulay A. BARKATOU et codirigée par Paola BOITO

JURY :

Bernard MOURRAIN	Directeur de Recherche, INRIA-Sophia Antipolis	President
Bernhard BECKERMANN	Professeur, Université de Lille 1	Rapporteur
Françoise TISSEUR	Professeur, Université de Manchester	Rapporteur
Tetsuya SAKURAI	Professeur, Université de Tsukuba	Rapporteur
Nicolas BRISEBARRE	Chargé de Recherche, INRIA	Examinateur
Moulay A. BARKATOU	Professeur, Université de Limoges	Directeur
Paola BOITO	Maître de Conférences, Université de Limoges	Codirecteur

A mis padres y hermanos,
A Noelia,
A mis amigos.

Acknowledgements

First of all, I would like to thank my supervisors, Paola Boito and Moulay Barkatou, for their friendship, support, guidance and patience throughout my years as a PhD student at the University of Limoges. Their outstanding efforts have made this thesis possible and I will always be infinitely grateful!

During the review and defending process of my doctoral dissertation, I had an excellent jury. I appreciate their availability and all their comments to improve my final work. In particular, I would like to thank Prof. Françoise Tisseur, Prof. Tetsuya Sakurai and Prof. Bernhard Beckermann for their time and support and, also, for the invitation and welcoming at their universities to help me in my thesis.

Among the friends and colleagues, special thanks go to: Reda Chakib, Riadh Omheni, Youssed El Jazouli, Anthony de Wyse, Nicolas Lauron, Jean Charles Chevron, Suzy Maddah, Richard Bezin, Achref Jalouli, Tibaire Munsch, Nicolas Pavie, Carole El Bacha, Benoit Crespin, Thomas Cluzeau, Jacques-Arthur Weil, Leonardo Robol, Yuji Nakatsukasa, Javier Pérez, Javier González, Prof. Froilán Dopico, Prof. Ion Zaballa, Prof. Wolf-Juergen Beyn, Fabien Brossard, the people of Boisseuil F.C. and of PANAFUTSAL. All these people, and many others, made my stay abroad very happy and pleasant.
In Costa Rica, I am grateful with my friends and professors of the University of Costa Rica. In particular, I have no words to express my gratitude to Prof. Javier Trejos, which presented to me the opportunity to do my studies in Limoges and, in addition, was always willing to help me in everything I needed. Moreover, I appreciate the support through my years as a student given by Prof. Eduardo Piza, Prof. William Alvarado, Prof. Eugenio Chinchilla and Prof. Mario Villalobos.

I would also like to thank the University of Limoges and the institute XLIM for welcoming me into their master and doctoral programs and for always supporting me. In particular, I thank all the people envolved in the Master ACSYON, the group of Calcul Formal, Prof. Michel Théra, Prof. Samir Adly, Odile Duval, Annie Nicolas and Yolande Vieceli.

I am also grateful with the University of Costa Rica, for its excellent academical preparation, for the financial support during my studies in Europe and, also, for reserved me a place of work for my return.

Finally, I want to thank my family, specially to my father Eugenio Segura, for all their effort and support over the years. And the last but not least, the most important person during this time: my beautiful Noelia, which was always by my side giving me all the support I needed, making me laugh and living this adventure with me.

Esteban Segura Ugalde

Contents

Notation 5

Introduction 7

Chapter 1 : Matrix Polynomials and the Eigenvalue Problem 11
1.1 Matrix Polynomials and the Eigenvalue Problem 12
1.1.1 Systems of Ordinary Differential Equations 13
1.1.2 Smith Normal Form 15
1.1.3 Solving PEPs via Linearization 15
1.1.4 Direct Methods to Solve PEPs 16
1.2 Scaling of Generalized and Polynomial Eigenvalue Problems 17
1.2.1 Balancing Technique for Matrices 18
1.2.2 Balancing Technique for Generalized and PEPs 18
1.3 Shifting Technique for $P(\lambda)$ 19
1.3.1 Meini's Shifting Formulation 20
1.3.2 Generalization of Shifting Technique 21

Chapter 2 : Invariant Pairs: Theory, Conditioning and Backward Error 28
2.1 Introduction 29
2.2 Previous Work on Invariant Pairs 29
2.3 Definition and Theory 30
2.3.1 Particular Case: Jordan Pairs 32
2.4 Formulation of the Invariant Pair Problem Using the Contour Integral . . . 32
2.5 Linearized Matrix Equation and Fréchet Derivative 33
2.6 Condition Number and Backward Error for the Invariant Pair Problem . . 34
2.6.1 Condition Number for $P(X, S)$ 34
2.6.2 Backward Error for $P(X, S)$ 38

Chapter 3 : Matrix Solvents 42
3.1 Matrix Solvents: Definition and Theory 43
3.1.1 Existence of Solvents 43
3.2 Condition Number and Backward Error for the Matrix Solvent Problem . . 45
3.2.1 Condition Number of $P(S)$ 45
3.2.2 Backward Error of $P(S)$ 47
3.3 Computation of Solvents 48
3.3.1 A Computational Approach 48
3.3.2 Matrix p-th Root 49
3.4 Solvents and Triangularized Matrix Polynomials 51
3.4.1 Triangularizing Matrix Polynomials 51
3.4.2 Procedure to Triangularize a Quadratic Matrix Polynomial 52
3.4.3 A Problem with an Infinite Number of Solvents 59

Chapter 4 : Moments, Hankel Pencils and Computation of Invariant Pairs 61
4.1 Introduction 62

4.2 Toeplitz and Hankel Matrices . . . 62
4.3 The Moment Method and Eigenvalues . . . 63
4.3.1 Computing Invariant Pairs via Moment Pencils . . . 70
4.3.2 The Block Moment Method . . . 73
4.4 Choosing the Contour . . . 77
4.5 Numerical Approximation: Trapezoid Rule for Moments . . . 79
4.5.1 Error Analysis of the Trapezoid Rule for Moments . . . 79

Chapter 5 : Iterative Refinement of Invariant Pairs and Matrix Solvents . . . 86
5.1 Iterative Refinement of Invariant Pairs and Matrix Solvents . . . 87
5.2 Newton's Method . . . 87
5.2.1 Algorithm . . . 88
5.3 Incorporating Line Search into Newton's Method . . . 89
5.3.1 Case of Invariant Pairs . . . 89
5.3.2 Case of Matrix Solvents . . . 91
5.3.3 Algorithm . . . 93
5.4 Šamanskii's Technique . . . 93
5.4.1 Algorithm . . . 93
5.5 Solution of the Correction Equation . . . 94
5.5.1 Using the Kronecker Product . . . 95
5.5.2 Using Forward Substitution . . . 96
5.6 Numerical Results . . . 99
5.7 Functions Implemented in MATLAB and Maple . . . 99

Chapter 6 : Conclusions and Future Work . . . 105

Chapter A : Appendix . . . 108
A.1 Kronecker Product . . . 109
A.1.1 Kronecker Product Properties . . . 109
A.2 Vectorization . . . 109
A.2.1 Compatibility with Kronecker Products . . . 109
A.3 Vector and Matrix Norm . . . 109
A.3.1 Vector Norm . . . 109
A.3.2 Matrix Norm . . . 110
A.4 Pseudo-inverse of a Matrix . . . 111
A.5 The Generalized Schur Decomposition . . . 111
A.6 Composite Trapezoidal Rule . . . 111

Index . . . 112

References . . . 114

Notation

$\mathbb{N}$	Set of nonnegative integers
$\mathbb{N}^*$	Set of positive integers
$\mathbb{Z}$	Ring of integers
$\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{n\times n}$	Set of real numbers, n-dimensional (column) real vectors and $n \times n$ real matrices
$\mathbb{C}$, $\mathbb{C}^n$, $\mathbb{C}^{n\times n}$	Set of complex numbers, n-dimensional (column) complex vectors and $n \times n$ complex matrices
$\mathbb{K}[\lambda]$	Ring of polynomials in λ over a ring $\mathbb{K}$
$\mathbb{A}^{m\times n}$	Additive group of $m \times n$ matrices with entries in a ring $\mathbb{A}$
$\mathbb{A}^{n\times n}$	Ring of $n \times n$ matrices with entries in a ring $\mathbb{A}$
$\mathbb{A}^n$ resp. $A^{1\times n}$	Additive group of $n-$dimensional (column) vectors, resp. row vectors, with entries in a ring $\mathbb{A}$
$\deg(p)$	Degree of the polynomial p
0_n	Square zero matrix of size n
I_n	Identity matrix of size n
A^T	Transpose of a matrix/vector A
A^*	Conjugate transpose of a matrix A
A^{-1}	Inverse of a matrix A
A^+	Pseudo-inverse of a matrix A
$\sigma(A)$	Spectrum of a square matrix A
$\operatorname{rank}(A)$	Rank of a matrix A
$\det(A)$	Determinant of a matrix A
$\operatorname{tr}(A)$, $\operatorname{trace}(A)$	Trace of a matrix A
$A \otimes B$	Kronecker product of the matrices A and B
$\operatorname{diag}(A_1, A_2 \ldots, A_n)$	The (block) diagonal matrix $\begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_n \end{bmatrix}$
a_{ij}, $A(i,j)$, $[A]_{ij}$	The (i,j)th entry of a matrix A
$A(i,:)$	The ith row of a matrix A
$A(:,j)$	The jth column of a matrix A
$\|x\|$	Norm of a vector x (see A.3.1)
$\|A\|$	Norm of a matrix A (see A.3.2)
$\operatorname{vec}(A)$	Vectorization of a matrix A (see A.2)

Introduction

Invariant pairs, introduced and analyzed in [17], [20], [42] and [120], are a generalization of eigenpairs for matrix polynomials. Let $P(\lambda) = \sum_{j=0}^{\ell} A_j \lambda^j$ be an $n \times n$ matrix polynomial: the polynomial eigenvalue problem consists in computing a scalar λ and a nonzero vector x such that $P(\lambda)x = 0$. Now, choose a positive integer k. The matrices X, S of sizes $n \times k$ and $k \times k$, respectively, are said to form an invariant pair of size k for $P(\lambda)$ if $X \neq 0$ and:

$$P(X, S) := \sum_{j=0}^{\ell} A_j X S^j = 0. \tag{1}$$

Note that the eigenvalues of the matrix S are also eigenvalues of $P(\lambda)$.

Invariant pairs offer a unified theoretical perspective on the problem of computing several eigenvalue-eigenvector pairs for a given matrix polynomial. From a numerical point of view, moreover, the computation of an invariant pair tends to be more stable than the computation of single eigenpairs, particularly in the case of multiple or tightly clustered eigenvalues. The notion of invariant pairs can also be applied to more general nonlinear problems, although here we will limit our presentation to matrix polynomials.

How to compute invariant pairs? Beyn and Thümmler ([20]) adopt a continuation method of predictor-corrector type. Betcke and Kressner ([17]), on the other hand, establish a correspondence between invariant pairs of a given matrix polynomial and of its linearizations. Invariant pairs for $P(\lambda)$ are extracted from invariant pairs of a linearized form and then refined via Newton's method.

The approach we take in this work to compute invariant pairs is based on contour integrals. Being able to specify the contour Γ allows us to select invariant pairs that have eigenvalues in a prescribed part of the complex plane. Contour integrals play an important role in the definition and computation of moments, which form a Hankel matrix pencil yielding the eigenvalues of the given matrix polynomial that belong to the prescribed contour. The use of Hankel pencils of moment matrices is widespread in several applications such as control theory, signal processing or shape reconstruction, but nonlinear eigenvalue-eigenvector problems can also be tackled through this approach, as suggested for instance in [7] and [18]. E. Polizzi's FEAST algorithm [112] is also an interesting example of contour-integral based eigensolver applied to large-scale electronic structure computations.

In this work, we adapt such methods to the computation of invariant pairs. We study, in particular, the scalar moment method and its relation with the multiplicity structure of the eigenvalues, but we also explore the behavior of the block version.

These results on invariant pairs can be applied to the particular case of matrix solvents,

that is, to the matrix equation:

$$P(S) := \sum_{j=0}^{\ell} A_j S^j = 0.$$

The matrix solvent problem has received remarkable attention in the literature since Sylvester's work [119] in the 1880s. The relation between the Riccati and the quadratic matrix equation is highlighted in [21] whereas a study on the existence of solvents can be found in [38]. Several works address the problem of computing a numerical approximation for the solution of the quadratic matrix equation: an approach to compute, when possible, the dominant solvent is proposed in [37]. Newton's method and some variations are also used to approximate solvents numerically: see, for example, [34], [80], [63], [88]. The work in [58] uses interval arithmetic to compute an interval matrix containing the exact solution to the quadratic matrix equation. For the case of the general matrix solvent problem, we can also cite [26], [111] and [81].

We exhibit computable formulations for the condition number and backward error of the general matrix solvent problem, thus generalizing previous work on the quadratic matrix equation. Moreover, we propose an adaptation of the moment method to the computation of solvents. Finally, we build on existing work on triangularization of matrix polynomials (see [124] and [121]) and explore the relationship between solvents of matrix polynomials in general and in triangularized form.

This thesis is organized as follows. Chapter 1 introduces preliminary notions, definitions and notation concerning matrix polynomials and their applications, the relation with systems of ordinary differential equations, the generalized eigenvalue problem and some solution methods. The last section discusses eigenvalue shifting for matrix polynomials and presents a generalization of the shifting technique proposed in [101]. Eigenvalue shifting consists in moving one or several eigenvalues of a given matrix polynomial to prescribed positions in the complex plane (or to infinity). It can be useful, for instance, as a preliminary modification of the polynomial before applying methods that require a particular eigenvalue distribution.

Chapter 2 introduces the general theory of the invariant pair problem, along with an alternative formulation based on the contour integral. The original contribution of this chapter consists in new formulations for the condition number and the backward error of the invariant pair problem. These formulas are obtained using the definition (1) recalled above.

Chapter 3 is devoted to matrix solvents. A quick review of the general theory, of applications, and of an alternative formulation based on the contour integral are given. Next, we specialize the results of Chapter 2 to solvents, and we compute new characterizations

for the condition number and the backward error of the matrix solvent problem. Also, motivated by the results in [121] and [124], we study in Section 3.4 the relation between solvents of general and triangularized matrix polynomials.

Chapter 4 focuses on the computation of eigenvalues and invariant pairs through moments and Hankel pencils. Our starting point here is the Sakurai-Sugiura method ([115], [7]) for computing eigenvalues of a matrix polynomial $P(\lambda)$. The main idea consists in defining a complex function $f(z)$ whose poles are the eigenvalues of $P(\lambda)$ inside a given contour Γ, and in computing a few moments of $f(z)$ via contour integrals. The moments are then arranged to form a Hankel matrix pencil, whose generalized eigenvalues are the eigenvalues of $P(\lambda)$ that belong to the interior of Γ. We generalize and adapt this approach to the computation of invariant pairs with eigenvalues belonging to a prescribed region of the complex plane. We also discuss the effectiveness of the scalar and block versions of the method in presence of multiple eigenvalues. In particular, we show that the scalar method cannot capture some eigenvalue multiplicity structures, for which the block version is needed. Our main results here consist in Theorem 14, Corollary 2 and Theorem 16. Moreover, Section 4.4 addresses some questions on the choice of the contour Γ; its content is not new, but it may offer a useful complement to the topics of this thesis, particularly, regarding the estimation of the number of eigenvalues in a given contour. Finally, Section 4.5.1 presents a theoretical and experimental error analysis for the trapezoid rule applied to our quadrature problems.

The techniques presented in Chapter 4 – as well as other direct approaches to the computation of invariant pairs – can be used either alone or in combination with iterative refinement methods. Motivated by the work in [17], in Chapter 5 we propose and compare some variants of Newton's method applied to the numerical refinement of invariant pairs. In particular, we experiment with line search strategies and with Šamanskii's acceleration technique [126].

Finally, Chapter 6 presents some conclusions and ideas for future work.

The Maple and MATLAB implementations of the symbolic and numeric methods presented in this thesis are available online at the URL
`http://www.unilim.fr/pages_perso/esteban.segura/software.html`

Chapter 1 :

Matrix Polynomials and the Eigenvalue Problem

1.1 Matrix Polynomials and the Eigenvalue Problem

Let $\mathbb{K}$ be an arbitrary field. An $n \times n$ matrix polynomial is an $n \times n$ matrix whose entries are polynomials in $\mathbb{K}[\lambda]$. Any matrix polynomial $P(\lambda)$ can be written in the following form:

Definition 1. *An $n \times n$ matrix polynomial $P(\lambda)$ is defined as:*

$$P(\lambda) = A_0 + A_1\lambda + A_2\lambda^2 + \cdots + A_\ell\lambda^\ell = \sum_{i=0}^{\ell} A_i\lambda^i, \tag{1.1}$$

where $\ell \in \mathbb{N}$ is the degree of the matrix polynomial and $A_0, A_1, ..., A_\ell \in \mathbb{K}^{n\times n}$.

A_ℓ is called the *leading coefficient matrix* and A_0 is called the *trailing coefficient matrix* of $P(\lambda)$. When $A_\ell = I_n$, the matrix polynomial is said to be *monic*.

The *rank* of a matrix polynomial $P(\lambda)$ is defined as:

$$\operatorname{rank}(P(\lambda)) = \max\{\operatorname{rank}(P(\lambda_0)) : \lambda_0 \in \bar{K}\}.$$

Definition 2. *An $n \times n$ matrix polynomial $P(\lambda)$ is said to be regular if rank$(P(\lambda)) = n$, or, equivalently, if its determinant det$(P(\lambda))$ does not vanish identically. Otherwise, it is said to be singular.*

In this work, we assume that $P(\lambda)$ is regular.

Definition 3. *The reversal of the matrix polynomial $P(\lambda)$ is:*

$$rev(P(\lambda)) := \lambda^\ell P(1/\lambda) = \lambda^\ell A_0 + \lambda^{\ell-1}A_1 + \lambda^{\ell-2}A_2 + \cdots + A_\ell \tag{1.2}$$

In this work, $\mathbb{K}$ denotes either the field of complex numbers $\mathbb{C}$ or the field of real numbers $\mathbb{R}$.

Definition 4. *The polynomial eigenvalue problem (PEP) consists in determining right eigenvalue-eigenvector pairs $(\lambda, x) \in \mathbb{C} \times \mathbb{C}^n$, with $x \neq 0$, such that*

$$P(\lambda)x = 0,$$

or left eigenvalue-eigenvector pairs $(\lambda, y) \in \mathbb{C} \times \mathbb{C}^n$, with $y \neq 0$, such that

$$y^*P(\lambda) = 0.$$

Remark 1.

- *The homogeneous formulation of equation* (1.1)*:*

$$P(\alpha,\beta) = \beta^{\ell}A_0 + \alpha\beta^{\ell-1}A_1 + \cdots + \alpha^{\ell}A_{\ell} = \sum_{i=0}^{\ell}\alpha^{i}\beta^{\ell-i}A_i,$$

  for $\lambda = \alpha/\beta$ *and* $(\alpha,\beta) \neq (0,0)$*, allows the simultaneous treatment of finite and infinite eigenvalues (see [4], [64], [67]).*

- *Infinite eigenvalues can still be covered using the reversal matrix polynomial* (1.2)*. Infinite eigenvalues of the matrix polynomial* $P(\lambda)$ *are zero eigenvalues of the reversal matrix polynomial* $rev(P(\lambda))$*.*

A particular case of special interest is the *quadratic eigenvalue problem* (QEP), where $\ell = 2$:

$$Q(\lambda)x = (A_0 + A_1\lambda + A_2\lambda^2)x = 0, \quad x \neq 0. \tag{1.3}$$

Typical applications of the QEP include the vibration analysis of buildings, machines and vehicles (see [52], [84], [123]). A considerable amount of work has been done on the theoretical and computational study of the QEP: see, for instance, [123].

Remark 2. *The case when* $\ell = 1$ *for the PEP corresponds to the generalized eigenvalue problem (GEP) for matrix pencils:*

$$Ax = \lambda Bx, \tag{1.4}$$

and if, moreover, we have that $B = I$*, we obtain the standard eigenvalue problem:*

$$Ax = \lambda x.$$

1.1.1 Systems of Ordinary Differential Equations

Matrix polynomials play an important role in the study of ordinary differential equations (ODEs). A system of ODEs of order $\ell > 1$, with constant coefficients, takes the form (see, e.g., [51]):

$$\sum_{i=0}^{\ell} A_i \left(\frac{d}{dt}\right)^{i} u(t) = 0. \tag{1.5}$$

Suppose we are looking for solutions of the form $u(t) = x_0 e^{\lambda_0 t}$, where x_0 and λ_0 are independent of t, then (1.5) leads to the polynomial eigenvalue problem $P(\lambda_0)x_0 = 0$, where $P(\lambda) = \sum_{i=0}^{\ell} A_i\lambda^i$.

More generally, the function:

$$u(t) = \left\{ \frac{t^k}{k!}x_0 + \cdots + \frac{t}{1!}x_{k-1} + x_k \right\} e^{\lambda_0 t}$$

is a solution of the differential equation if and only if the set of vectors $x_0, x_1, \ldots, x_k$, with $x_0 \neq 0$, satisfies the relations:

$$\sum_{i=0}^{j} \frac{1}{i!} P^{(i)}(\lambda_0) x_{j-i} = 0, \quad j = 0, 1, \ldots, k.$$

This set of vectors $x_0, x_1, \ldots, x_k$ is called a Jordan chain of length $k+1$ associated with the eigenvalue λ_0 and the eigenvector x_0.

In particular, when $\ell = 2$, consider the linear second order differential equation:

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = f(t), \tag{1.6}$$

where M, C, K are $n \times n$ matrices and $f(t)$ is a vector. Such equations arise in many engineering applications, for instance, when studying mechanical and electrical oscillation (see [84], [123]).

The homogeneous equation of (1.6) leads to the QEP:

$$(\lambda^2 M + \lambda C + K)x = 0$$

for the solutions of the form $q(t) = e^{\lambda t}x$.

Several examples of real life problems related to (1.6) have been studied in the literature. For instance, in [1], [5], [33], [40], [89], [123] the wobbling of the Millennium bridge over the river Thames in London was discussed. On its opening in 2000, this footbridge started to wobble. It has generally been thought that the Millennium Bridge wobble was due to pedestrians synchronizing their footsteps with the bridge motion. However, this is not supported by measurements of the phenomenon on other bridges. In [89], a simple model of human balance strategy for normal walking on a stationary surface was considered. This model showed that pedestrian can effectively act as a negative (or positive) damper to the bridge motion and hence inadvertently feed energy into bridge oscillations.

Two days after the opening, the bridge was closed for almost two years while modifications were made to eliminate the wobble entirely. It reopened in 2002.

Another interesting real life example is the project of the company SFE GmbH in Berlin, which investigated the noise in rail traffic that is caused by high speed trains (see [71], [77], [91]).

1.1.2 Smith Normal Form

Definition 5. *A matrix polynomial $E(\lambda)$ is unimodular if its determinant is a nonzero constant, independent of λ.*

Definition 6. *Two matrix polynomials $A(\lambda)$ and $B(\lambda)$ of the same size are called equivalent ($A(\lambda) \sim B(\lambda)$) if there exist two unimodular matrix polynomials $E(\lambda)$ and $F(\lambda)$, such that:*

$$A(\lambda) = E(\lambda)B(\lambda)F(\lambda).$$

A crucial and well-known property of matrix polynomials is the existence of the Smith form (see, e.g., [51]).

Theorem 1. *[51] Every $n \times n$ matrix polynomial $P(\lambda)$ is equivalent to a matrix polynomial $D(\lambda)$ of the form*

$$D(\lambda) = \mathrm{diag}(d_1(\lambda), \ldots, d_n(\lambda)), \tag{1.7}$$

where the d_i's are monic polynomials such that $d_i(\lambda)$ is divisible by $d_{i-1}(\lambda)$.

Definition 7. *The matrix polynomial $D(\lambda)$, given by (1.7), is called the Smith normal form of $P(\lambda)$, and the monic scalar polynomials $d_i(\lambda)$ are called the invariant polynomials of $P(\lambda)$.*

1.1.3 Solving PEPs via Linearization

Linearization is a well established method for computing eigenvalues and eigenvectors of polynomial eigenvalue problems of moderate size.

Given a polynomial eigenvalue problem (1.1) of degree $\ell \geq 2$, the linearization method approach consists in converting $P(\lambda)$ to a linear $\ell n \times \ell n$ pencil $L(\lambda) = A + \lambda B$, having the same spectrum as $P(\lambda)$. This linear eigenvalue problem can then be solved by standard methods, e.g., the QZ algorithm (see A.5).

Software libraries as LAPACK provide routines, among others, for solving systems of linear equations (see [6]). On the other hand, the library ARPACK uses the implicitly restarted Arnoldi method or, in the case of symmetric matrices, the corresponding variant of the Lanczos algorithm to compute a few eigenvalues and corresponding eigenvectors of large sparse or structured matrices (see[86]).

We introduce the notion of linearization following [52] and [93].

Definition 8. *Let $P(\lambda)$ be an $n \times n$ matrix polynomial of degree $\ell \geq 1$. A pencil $L(\lambda) = A + \lambda B$ with $A, B \in \mathbb{C}^{\ell n \times \ell n}$ is called linearization of $P(\lambda)$ if there exist unimodular matrix polynomials $E(\lambda), F(\lambda) \in \mathbb{C}^{\ell n \times \ell n}$, such that:*

$$F(\lambda)L(\lambda)E(\lambda) = \begin{bmatrix} P(\lambda) & 0 \\ 0 & I_{(\ell-1)n} \end{bmatrix}. \tag{1.8}$$

Remark 3. *The linearization* (1.8) *is not unique (see [93], [123]).*

Most of the linearizations used in practice are of the first companion form (see [52], [91]):

$$C_1(\lambda) = \begin{bmatrix} A_{\ell-1} & A_{\ell-2} & \cdots & A_0 \\ -I_n & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -I_n & 0 \end{bmatrix} + \lambda \begin{bmatrix} A_\ell & 0 & \cdots & 0 \\ 0 & I_n & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I_n \end{bmatrix}$$

or the second companion form:

$$C_2(\lambda) = \begin{bmatrix} A_{\ell-1} & -I_n & \cdots & 0 \\ A_{\ell-2} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & -I_n \\ A_0 & 0 & \cdots & 0 \end{bmatrix} + \lambda \begin{bmatrix} A_\ell & 0 & \cdots & 0 \\ 0 & I_n & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I_n \end{bmatrix}.$$

It is well known that using this approach to solve the PEP via linearization can have some drawbacks. For instance:

- The linearization approach transforms the original $n \times n$ matrix polynomial of degree ℓ into a larger $\ell n \times \ell n$ linear eigenvalue problem.
- The conditioning of the linearized problem can depend on the type of linearization used and may be significantly worse than the conditioning of the original problem (see [69], [123]).
- If there is a special structure in the matrix coefficients A_i, such as symmetry, sparsity pattern, palindromicity, the linearization may modify it. In that case, special linearizations can be chosen to exploit the structure (see [2], [3], [14], [29], [35], [36], [49], [66], [67],[68],[90], [92], [93], [94], [95], [96], [97], [98], [99]).

1.1.4 Direct Methods to Solve PEPs

In the previous section, we have listed some drawbacks of the linearization method for solving PEPs. We pointed out that this approach not only increases the dimension of the problem, but may also be very ill-conditioned.

We recall now some approaches which offer the possibility to handle the PEPs directly without linearization.

- Ehrlich-Aberth iteration: In [22], the authors present an effective approach based on the Ehrlich-Aberth iteration to approximate eigenvalues of matrix polynomials. The

main points addressed in this work are the choice of the starting approximations, the computation of the Newton correction, the halting criterion and the case of (multiple) eigenvalues at zero and at infinity.

- Jacobi-Davidson method: This method computes selected eigenvalues and associated eigenvectors of a matrix polynomial. It iteratively constructs approximations of certain eigenvectors by solving a projected problem. The method finds the approximate eigenvector as "best" approximation in some search subspace (see [116], [117]).
  This approach has been used for the efficient solution of quadratic eigenproblems associated with acoustic problems with damping (see [125]).

- A second-order Arnoldi method for the solution of the quadratic eigenvalue problem (SOAR): This method for solving large-scale QEPs generates an orthonormal basis and then applies the standard Rayleigh–Ritz orthogonal projection technique to approximate eigenvalues and eigenvectors (see [8]).

- Arnoldi and Lanczos-type methods: These processes are developed to construct projections of the QEP. The convergence of these methods is usually slower than a Krylov subspace method applied to the equivalent linear eigenvalue problem (see [73]).

- A subspace approximation method: In [74], the authors use perturbation subspaces for block eigenvector matrices to reduce a modified problem to a sequence of problems of smaller dimension. They show that this method converges at least as fast as the corresponding Taylor series, and that Rayleigh quotient iteration can be used for acceleration.

- Contour integral based methods: In [7], [18] and [114], the authors use contour integral formulations to find all the eigenvalues of PEPs, which are inside a closed contour in the complex plane.

1.2 Scaling of Generalized and Polynomial Eigenvalue Problems

The performance of an algorithm may depend crucially on how the problem is formulated. Balancing is a preprocessing technique that aims to avoid large differences in magnitude among matrix entries, which may cause a poor numerical performance. A matrix with a norm that is several orders of magnitude larger than the modulus of its eigenvalues typically has eigenvalues that are sensitive to perturbations in the entries.

The process of balancing produces a matrix, which is diagonally similar to the given matrix and reduces the matrix norm. As a consequence, the eigenvalues do not change, but their sensitivity can significantly be reduced. Such a diagonal scaling is, therefore, typically used before running any eigenvalue algorithm.

1.2.1 Balancing Technique for Matrices

In the case of a $n \times n$ matrix A, balancing consists in finding a diagonal matrix D such that DAD^{-1} is a well-scaled matrix (see [41], [108], [109]). For instance, consider the badly scaled matrix A defined by:

$$A = \begin{bmatrix} 0 & 2^{-20} & 2^{-5} \\ 2^{20} & 1 & 0 \\ 2^{6} & 2^{-15} & 0 \end{bmatrix}.$$

If we balance A using the diagonal matrix $D = \mathrm{diag}(2^{-20}, 1, 2^{-15})$, we obtain:

$$D^{-1}AD = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \end{bmatrix}.$$

Let us compute the eigenvalues of A and of $D^{-1}AD$ using the MATLAB command `eig`

`eig`(A,'nobalance')	`eig`($D^{-1}AD$)
2.170086486**35809**	2.170086486626034
-1.481194304**285063**	-1.481194304092016
0.311107817**428706**	0.311107817465982

If we compute the condition number for the eigenvalues of A and of $D^{-1}AD$ using MATLAB command `condeig`, we obtain:

`condeig`(A)	`condeig`($D^{-1}AD$)
335275.9044	1.1585
237384.3761	1.0392
98402.2850	1.1197

Remark 4. *There are also cases in which balancing can lead to a catastrophic increase of the errors in the computed eigenvalues (see [128]).*

1.2.2 Balancing Technique for Generalized and PEPs

In the case of the generalized eigenvalue problems $Ax = \lambda Bx$, the authors of [127] introduce a balancing technique which aims to find diagonal matrices D_1 and D_2 such that

the elements of D_1AD_2 and D_1BD_2 are scaled as equal in magnitude as possible (see also [72]).

A different approach for the scaling of GEPs is proposed in [87]. A linearly convergent iteration provides matrices D_1 and D_2 consisting of powers of 2 that approximately satisfy:

$$\|D_1AD_2e_j\|_2^2 + \|D_1BD_2e_j\|_2^2 = \|e_i^*D_1AD_2\|_2^2 + \|e_i^*D_1BD_2\|_2^2 = 1, \quad i,j = 1,\ldots,n.$$

In [15], besides studying optimal balancing of GEPs and polynomial eigenvalue problems, it is noted that this iteration can easily be extended to weighted scaling of matrix polynomials $P(\lambda)$ by:

$$\sum_{k=0}^{\ell} \omega^{2k}\|D_1A_kD_2e_i\|_2^2 = 1, \quad \sum_{k=0}^{\ell} \omega^{2k}\|e_j^*D_1A_kD_2\|_2^2 = 1, \quad i,j = 1,\ldots,n$$

for some $\omega > 0$ that is chosen to be close in magnitude to the desired eigenvalues.

In the specific case of the quadratic matrix polynomial $Q(\lambda)$, Fan, Lin and Van Dooren [45] suggest that a good scaling strategy for the QEP is to scale the coefficients A_2, A_1 and A_0 so that their 2-norms are all close to 1. They consider modifying $Q(\lambda)$ to:

$$\tilde{Q}(\mu) \equiv \beta Q(\lambda) = \mu^2\tilde{A}_2 + \mu\tilde{A}_1 + \tilde{A}_0,$$

where:

$$\lambda = \alpha\mu, \quad \tilde{A}_2 = \alpha^2\beta A_2, \quad \tilde{A}_1 = \alpha\beta A_1, \quad \tilde{A}_0 = \beta A_0,$$
$$\alpha = \sqrt{\frac{a_0}{a_2}}, \quad \beta = \frac{2}{(a_0 + a_1\alpha)}$$

and

$$a_2 = \|A_2\|_2, \quad a_1 = \|A_1\|_2, \quad a_0 = \|A_0\|_2.$$

Note that the eigenvalues of $Q(\lambda)$ can be recovered from those of $\tilde{Q}(\mu)$ by $\lambda = \alpha\mu$. Moreover, this scaling approach does not affect any sparsity of A_2, A_1 and A_0.

In [57], it is presented an eigensolver for the complete solution of QEPs (function `quadeig` in MATLAB), which uses this scaling technique.

A different approach to balance matrix polynomials can be found in [31].

1.3 Shifting Technique for $P(\lambda)$

In [101], B. Meini describes a technique to shift two eigenvalues λ_1 and λ_2 of an $n \times n$ quadratic matrix polynomial $P(\lambda)$. This method requires the knowledge of a right and of

a left eigenvector associated with λ_1 and λ_2, respectively.
The idea is to shift λ_1 to 0 and λ_2 to ∞ and then deflate those values obtaining as a result a new $(n-1)\times(n-1)$ quadratic matrix polynomial, which shares with $P(\lambda)$ all its eigenvalues, except for λ_1 and λ_2.

As we pointed out before, this approach requires the knowledge of a right and of a left eigenvector associated with the eigenvalues λ_1 and λ_2. Here, we present a formulation that allow us to shift to infinity several eigenvalues at the same time, assuming only the knowledge of a right eigenvector associated with those eigenvalues to shift.

Let us first recall some results from [101].

1.3.1 Meini's Shifting Formulation

Given a $n\times n$ matrix polynomial $P(\lambda)$ of degree $\ell\geq 1$, we have:

Theorem 2. *[Thm. 1, [101]] Let $\lambda_1\in\mathbb{C}$ and $v\in\mathbb{C}^n$, $v\neq 0$, such that $P(\lambda_1)v=0$. Then for any $\eta\in\mathbb{C}$ and for any vector x such that $x^*v=1$, the following properties hold:*

1. *The function*
$$\tilde{P}(\lambda)=P(\lambda)\left(I+\frac{\lambda_1-\eta}{\lambda-\lambda_1}vx^*\right)$$
is a matrix polynomial of degree ℓ;

2. $\det\tilde{P}(\lambda)=\frac{\lambda-\eta}{\lambda-\lambda_1}\det P(\lambda)$;

3. *$\tilde{P}(\eta)v=0$, i.e., the value η, where we moved the original eigenvalue λ_1, is an eigenvalue for the shifted matrix polynomial. Note that the eigenvector v is the same for λ_1 and η.*

4. *If $\sigma\notin\{\lambda_1,\eta\}$ is such that $P(\sigma)w=0$, then $\tilde{P}(\sigma)\tilde{w}=0$, where*
$$\tilde{w}=\left(I-\frac{\lambda_1-\eta}{\sigma-\eta}vx^*\right)w.$$

5. *Moreover, by setting:*
$$\tilde{P}(\lambda)=\sum_{i=0}^{\ell}\lambda^i\tilde{A}_i,$$
one has
$$\begin{aligned}\tilde{A}_\ell&=A_\ell,\\ \tilde{A}_i&=A_i+(\lambda_1-\eta)\sum_{j=i+1}^{\ell}\lambda^{j-i-1}A_jvx^*,\quad i=0,\ldots,\ell-1.\end{aligned}\tag{1.9}$$

Equations (1.9) give us an easy way to compute the coefficients of the matrix polynomial whose eigenvalue λ has been shifted.

Proposition 1. *[Prop. 2, [101]] Let $\lambda_1 \in \mathbb{C}$, $v \in \mathbb{C}^n$, $v \neq 0$, such that $P(\lambda_1)v = 0$ and let $x \in \mathbb{C}^n$ such that $x^*v = 1$. Then the following properties hold:*

1. *The matrix polynomial*

$$\tilde{P}(\lambda) = P(\lambda)\left(I + \frac{\lambda_1}{\lambda - \lambda_1} vx^*\right)$$

*is such that $\det \tilde{P}(\lambda) = \det P(\lambda)\frac{\lambda}{\lambda-\lambda_1}$ and $\tilde{P}(0)v = 0$, i.e., the eigenvalue λ_1 is **shifted to zero**, while **keeping the same right eigenvector** v;*

2. *if $\lambda_1 \neq 0$, the matrix polynomial*

$$\hat{P}(\lambda) = P(\lambda)\left(I + \frac{\lambda}{\lambda_1 - \lambda} vx^*\right)$$

*is such that $\det \hat{P}(\lambda) = \det P(\lambda)\frac{\lambda_1}{\lambda_1-\lambda}$ and $\hat{P}(\infty)v = 0$, i.e., the eigenvalue λ_1 is **shifted to infinity**, while **keeping the same right eigenvector** v.*

1.3.2 Generalization of Shifting Technique

The results presented in [101] allow us to shift a single eigenvalue of a matrix polynomial. Our goal in this section is to present a formulation to shift several eigenvalues at the same time. We start by shifting the eigenvalue λ_k to infinity.

Proposition 2. *Let $P(\lambda)$ be a matrix polynomial, $\lambda_k \neq 0 \in \mathbb{C}$ and $v \neq 0 \in \mathbb{C}^n$, such that $P(\lambda_k)v = 0$ and let $x \in \mathbb{C}^n$ be such that $x^*v = 1$. Consider the matrix polynomial $\hat{P}(\lambda) = \sum_{i=0}^{\ell} \lambda^i \hat{A}_i$ with coefficients:*

$$\begin{aligned} \hat{A}_0 &= A_0, \\ \hat{A}_i &= A_i + \lambda_k^{-1} \sum_{j=0}^{i-1} \lambda_k^{-j} A_{i-j-1} vx^*, \quad i = 1, \ldots, \ell. \end{aligned} \tag{1.10}$$

Then the following properties hold:

1. *$\hat{P}(\infty)v = 0$, i.e., the eigenvalue λ_k is shifted to infinity, while keeping the same right eigenvector v.*

2. *If $\lambda_r \notin \{\lambda_k, \infty\}$ and $w \neq 0$ are such that $P(\lambda_r)w = 0$, then $\hat{w} \neq 0$ such that*

$\hat{P}(\lambda_r)\hat{w} = 0$ *can be computed by:*

$$\hat{w} = \left(I - \frac{\lambda_r}{\lambda_k} v x^*\right) w. \tag{1.11}$$

Proof. The shift of λ_k to ∞ can be proved by applying Theorem 2 to the reversed matrix polynomial $rev(P(\lambda))$, by shifting λ_k^{-1} to 0. The matrix polynomial $\hat{P}(\lambda)$ is the reversed polynomial of

$$\tilde{P}(\lambda) = rev(P(\lambda)) \left(I + \frac{\lambda_k^{-1}}{\lambda - \lambda_k^{-1}} v x^*\right).$$

Then, we have:

$$\tilde{P}(\lambda) = \sum_{i=0}^{\ell} \lambda^i A_{\ell-i} \left(I + \frac{\lambda_k^{-1}}{\lambda - \lambda_k^{-1}} v x^*\right).$$

Adapting (1.9) to last $\tilde{P}(\lambda)$, we obtain:

$$\begin{aligned}
\tilde{A}_\ell &= A_{\ell-\ell} = A_0, \\
\tilde{A}_i &= A_{\ell-i} + \lambda_k^{-1} \sum_{j=i+1}^{\ell} \frac{1}{\lambda_k^{j-i-1}} A_{\ell-j} v x^*, \quad i = 0, \ldots, \ell - 1.
\end{aligned}$$

As $\hat{P}(\lambda) = rev(\tilde{P}(\lambda))$, we get:

$$\begin{aligned}
\hat{A}_0 &= A_0, \\
\hat{A}_i &= A_i + \lambda_k^{-1} \sum_{j=0}^{i-1} \lambda_k^{-j} A_{i-j-1} v x^*, \quad i = 1, \ldots, \ell.
\end{aligned}$$

Now, to prove the second part of the Theorem note that:

$$\begin{aligned}
\hat{P}(\lambda_r)\hat{w} &= P(\lambda_r) \left(I + \frac{\lambda_r}{\lambda_k - \lambda_r} v x^*\right) \left(I - \frac{\lambda_r}{\lambda_k} v x^*\right) w \\
&= P(\lambda_r) \left(I + \frac{\lambda_r}{\lambda_k} v x^* + \frac{\lambda_r}{\lambda_k - \lambda_r} v x^* - \frac{\lambda_r^2}{\lambda_k(\lambda_k - \lambda_r)} v x^*\right) w \\
&= P(\lambda_r) \left(I + \frac{-\lambda_r\lambda_k + \lambda_r^2 + \lambda_r\lambda_k - \lambda_r^2}{\lambda_k(\lambda_k - \lambda_r)} v x^*\right) w \\
&= P(\lambda_r) w = 0.
\end{aligned}$$

□

Remark 5. *The new matrix polynomial $\hat{P}(\lambda)$ has the same eigenvalues as $P(\lambda)$, except for λ_k that is shifted to ∞.*

Example 1. *Consider the following quadratic matrix polynomial discussed in [123]:*

$$P(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0 = \lambda^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} -2 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where the eigenvalues of $P(\lambda)$ *are:* $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = -1, \lambda_5 = \infty$ *and* $\lambda_6 = -\infty$. *Let's shift to infinity the eigenvalue* λ_1.
Note that if $v = [1; 0; 0]$ *is an eigenvector associated with* λ_1, *then the vector* $x = v$ *is such that* $x^* v = 1$.
To compute the coefficients of the shifted matrix polynomial $\hat{P}(\lambda)$, *we use* (1.10):

$$\begin{aligned} \hat{A}_0 &= A_0, \\ \hat{A}_1 &= A_1 + \lambda_1^{-1} \sum_{j=0}^{0} \lambda^{-j} A_{-j} v x^* = A_1 + \lambda_1^{-1} A_0 v x^*, \\ \hat{A}_2 &= A_2 + \lambda_1^{-1} \sum_{j=0}^{1} \lambda_1^{-j} A_{1-j} v x^* = A_2 + \lambda_1^{-1} \left(A_1 + \lambda_1^{-1} A_0 \right) v x^*. \end{aligned}$$

Then we obtain:

$$\hat{P}(\lambda) = \lambda^2 \hat{A}_2 + \lambda \hat{A}_1 + \hat{A}_0 = \lambda^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where the eigenvalues of $\hat{P}(\lambda)$ *are:* $\hat{\lambda}_1 = \infty, \hat{\lambda}_2 = 1, \hat{\lambda}_3 = -1, \hat{\lambda}_4 = 1, \hat{\lambda}_5 = \infty$ *and* $\hat{\lambda}_6 = -\infty$. *The associated eigenvector to the shifted eigenvalue remains equal, i.e.,* $\hat{P}(\infty)\hat{v} = 0$, *where* $\hat{v} = [1; 0; 0]$.
Moreover, suppose we want to compute the eigenvector $\hat{v}_4$ *such that* $\hat{P}(-1)\hat{v}_4 = 0$. *From formula* (1.11), *we have:*

$$\hat{v}_4 = \left(I - \frac{\lambda_4}{\lambda_1} v x^* \right) v_4 = \left(I + v x^* \right) v_4 = [0; -1; 0],$$

where $v_4 = [0; -1; 0]$ *is such that* $P(-1)v_4 = 0$.

In general, to shift m eigenvalues to ∞, we have the following more general result.

Theorem 3. *Let* $P(\lambda)$ *be an* $n \times n$ *matrix polynomial and its eigenvalues* $\lambda_i \neq 0 \in \mathbb{C}$ *with associated eigenvectors* $v_i \neq 0 \in \mathbb{C}^n$, *for* $i = 1, \ldots, m$. *Consider the* $n \times n$ *matrix*

polynomial $\hat{P}(\lambda) = \sum_{j=0}^{\ell} \lambda^j \overset{m}{A}_j$, *with coefficients:*

$$\begin{aligned} \overset{m}{A}_0 &= \overset{m-1}{A}_0 = \cdots = \overset{0}{A}_0 = A_0, \\ \overset{m}{A}_i &= \overset{m-1}{A}_i + \lambda_m^{-1} \left(\sum_{j=0}^{i-1} \lambda_m^{-j} \overset{m-1}{A}_{i-j-1} \right) \overset{m-1}{v}_m \overset{m-1}{x}_m^*, \quad i = 1, \ldots, \ell, \end{aligned} \tag{1.12}$$

where $\overset{0}{A}_i = A_i$.

Then the following properties hold:

- $\hat{P}(\infty) v_i = 0$, *i.e., the eigenvalues* λ_i *are shifted to infinity, while keeping the same right eigenvectors* v_i, *for* $i = 1, \ldots, m$.
- *If* $P(\lambda_{k+1}) v_{k+1} = 0$, *then the vector* $\overset{k}{v}_{k+1} \neq 0 \in \mathbb{C}^n$ *such that* $\hat{P}(\lambda_{k+1}) \overset{k}{v}_{k+1} = 0$, *can be computed by:*

$$\overset{k}{v}_{k+1} = \left[\prod_{j=0}^{k-1} \left(I - \frac{\lambda_{k+1}}{\lambda_{k-j}} \overset{k-j-1}{v}_{k-j} \overset{k-j-1}{x}_{k-j}^* \right) \right] v_{k+1}, \quad k = 1, \ldots, m-1, \tag{1.13}$$

where the vectors $\overset{k-j-1}{x}_{k-j}^* \in \mathbb{C}^n$ *satisfy* $\overset{k-j-1}{x}_{k-j}^* \overset{k-j-1}{v}_{k-j} = 1$ *and* $\overset{0}{v}_1 = v_1$, $\overset{0}{x}_1 = x_1$.

Proof. To get the coefficients $\overset{m}{A}_i$, for $i = 0, \ldots, \ell$, note that we just must apply m times the Proposition 2, i.e., the matrix polynomial $\hat{P}(\lambda)$ is:

$$\hat{P}(\lambda) = P(\lambda) \left(I + \frac{\lambda}{\lambda_1 - \lambda} v_1 x_1^* \right) \left(I + \frac{\lambda}{\lambda_2 - \lambda} v_2 x_2^* \right) \cdots \left(I + \frac{\lambda}{\lambda_m - \lambda} v_m x_m^* \right) w.$$

Note that:

$$\begin{aligned} \overset{2}{P}(\lambda) &= P(\lambda) \left(I + \frac{\lambda}{\lambda_1 - \lambda} v_1 x_1^* \right) \left(I + \frac{\lambda}{\lambda_2 - \lambda} v_2 x_2^* \right) \\ &= \overset{1}{P}(\lambda) \left(I + \frac{\lambda}{\lambda_2 - \lambda} v_2 x_2^* \right), \end{aligned}$$

where the coefficients of $\overset{1}{P}(\lambda)$ are:

$$\begin{aligned} \overset{1}{A}_0 &= A_0, \\ \overset{1}{A}_i &= A_i + \lambda_1^{-1} \sum_{j=0}^{i-1} \lambda_1^{-j} A_{i-j-1} v_1 x_1^*, \quad i = 1, \ldots, \ell. \end{aligned}$$

Then, the coefficients for $\overset{2}{P}(\lambda)$ are given by:

$$\begin{aligned}
&\overset{2}{A}_0 = \overset{1}{A}_0 = A_0,\\
&\overset{2}{A}_i = \overset{1}{A}_i + \lambda_2^{-1}\left(\sum_{j=0}^{i-1}\lambda_2^{-j}\overset{1}{A}_{i-j-1}\right)\overset{1}{v}_2\ \overset{1}{x}_2^*, \quad i=1,\ldots,\ell.
\end{aligned}$$

Continuing this m times we obtain that the coefficients for $\hat{P}(\lambda)$ are given by:

$$\begin{aligned}
&\overset{m}{A}_0 = \overset{m-1}{A}_0 = \cdots = \overset{0}{A}_0 = A_0,\\
&\overset{m}{A}_i = \overset{m-1}{A}_i + \lambda_m^{-1}\left(\sum_{j=0}^{i-1}\lambda_m^{-j}\overset{m-1}{A}_{i-j-1}\right)\overset{m-1}{v}_m\ \overset{m-1}{x}_m^*, \quad i=1,\ldots,\ell,
\end{aligned}$$

for

$$\overset{k}{v}_{k+1} = \left[\prod_{j=0}^{k-1}\left(I - \frac{\lambda_{k+1}}{\lambda_{k-j}}\overset{k-j-1}{v}_{k-j}\ \overset{k-j-1}{x}_{k-j}^*\right)\right]v_{k+1}, \quad k=1,\ldots,m-1,$$

where the vectors $\overset{k-j-1}{x}_{k-j}^* \in \mathbb{C}^n$ satisfy $\overset{k-j-1}{x}_{k-j}^*\ \overset{k-j-1}{v}_{k-j} = 1$ and $\overset{0}{v}_1 = v_1,\ \overset{0}{x}_1 = x_1$. □

Example 2. *Consider the quadratic matrix polynomial $P(\lambda)$ of Example 1 with eigenvalues: $\lambda_1 = 1, \lambda_2 = -1, \lambda_3 = 1, \lambda_4 = 1, \lambda_5 = \infty$ and $\lambda_6 = -\infty$.*

Let's shift to infinity the eigenvalues $\lambda_1 = 1$ and $\lambda_2 = -1$ with corresponding associated eigenvectors $v_1 = [1;0;0]$ and $v_2 = [0;-1;0]$. Consider the vectors: $x_1 = v_1$ and $x_2 = v_2$ such that $x_i^* v_i = 1$, for $i = 1, 2$.

To compute the coefficients of the shifted matrix polynomial $\hat{P}(\lambda)$, we use (1.12)*:*

$$\begin{aligned}
&\overset{2}{A}_0 = \overset{1}{A}_0 = \overset{0}{A}_0 = A_0,\\
&\overset{2}{A}_1 = \overset{1}{A}_1 + \lambda_2^{-1}\overset{1}{A}_0\overset{1}{v}_2\overset{1}{x}_2^* = \overset{0}{A}_1 + \lambda_1^{-1}\overset{0}{A}_0\overset{0}{v}_1\overset{0}{x}_1^* + \lambda_2^{-1}A_0\overset{1}{v}_2\overset{1}{x}_2^*\\
&\quad = A_1 + \lambda_1^{-1}A_0v_1x_1^* + \lambda_2^{-1}A_0\overset{1}{v}_2\overset{1}{x}_2^*,\\
&\overset{2}{A}_2 = \overset{1}{A}_2 + \lambda_2^{-1}\left(\sum_{j=0}^{1}\lambda_2^{-j}\overset{1}{A}_{1-j}\right)\overset{1}{v}_2\overset{1}{x}_2^* = \overset{1}{A}_2 + \lambda_2^{-1}\left(\overset{1}{A}_1 + \lambda_2^{-1}\overset{1}{A}_0\right)\overset{1}{v}_2\overset{1}{x}_2^*\\
&\quad = \overset{0}{A}_2 + \lambda_1^{-1}\left(\sum_{j=0}^{1}\lambda_1^{-j}\overset{0}{A}_{1-j}\right)\overset{0}{v}_1\overset{0}{x}_1^* + \lambda_2^{-1}\left(\overset{1}{A}_1 + \lambda_2^{-1}\overset{1}{A}_0\right)\overset{1}{v}_2\overset{1}{x}_2^*\\
&\quad = A_2 + \lambda_1^{-1}\left(A_1 + \lambda_1^{-1}A_0\right)v_1x_1^* + \lambda_2^{-1}\left(\overset{1}{A}_1 + \lambda_2^{-1}A_0\right)\overset{1}{v}_2\overset{1}{x}_2^*\\
&\quad = A_2 + \lambda_1^{-1}\left(A_1 + \lambda_1^{-1}A_0\right)v_1x_1^* + \lambda_2^{-1}\left(A_1 + \lambda_1^{-1}A_0v_1x_1^* + \lambda_2^{-1}A_0\right)\overset{1}{v}_2\overset{1}{x}_2^*.
\end{aligned}$$

Now, we need to compute $\overset{1}{v}_2 \in \mathbb{C}^n$. *Using* (1.13), *we obtain:*

$$\overset{1}{v}_2 = \left(I - \frac{\lambda_2}{\lambda_1} v_1 x_1^*\right) v_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}.$$

Taking $\overset{1}{x}_2 = \overset{1}{v}_2$, *satisfying* $\overset{1}{x^*}_2 \overset{1}{v}_2 = 1$, *we obtain:*

$$\hat{P}(\lambda) = \lambda^2 \overset{2}{A}_2 + \lambda \overset{2}{A}_1 + \overset{2}{A}_0 = \lambda^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that $\hat{P}(\lambda)$ *has the eigenvalues:* $\hat{\lambda}_1 = \infty, \hat{\lambda}_2 = \infty, \hat{\lambda}_3 = 1, \hat{\lambda}_4 = 1, \hat{\lambda}_5 = \infty$ *and* $\hat{\lambda}_6 = -\infty$.

The following result is a generalization of Theorem 2. It shows how to shift m eigenvalues λ_i to the values $\eta_1, \ldots, \eta_m$, with $\eta_i \neq \infty$, for $i = 1, \ldots, m$.

Theorem 4. *Let* $P(\lambda)$ *be an* $n \times n$ *matrix polynomial and its eigenvalues* $\lambda_i \neq 0 \in \mathbb{C}$ *with associated eigenvectors* $v_i \neq 0 \in \mathbb{C}^n$, *for* $i = 1, \ldots, m$. *Consider the* $n \times n$ *matrix polynomial* $\tilde{P}(\lambda) = \sum_{j=0}^{\ell} \lambda^j \overset{m}{A}_j$, *with coefficients:*

$$\begin{aligned} &\overset{m}{A}_\ell = \overset{m-1}{A}_\ell = \cdots = \overset{0}{A}_\ell = A_\ell, \\ &\overset{m}{A}_i = \overset{m-1}{A}_i + (\lambda_m - \eta_m) \left(\sum_{j=i+1}^{\ell} \lambda_m^{j-i-1} \overset{m-1}{A}_j \right) \overset{m-1}{v}_m \overset{m-1}{x}_m^*, \quad i = 0, \ldots, \ell - 1, \end{aligned} \tag{1.14}$$

where $\overset{0}{A}_i = A_i$.

Then the following properties hold:

- $\tilde{P}(\eta_i) v_i = 0$, *i.e., the eigenvalues* λ_i *are shifted to* η_i, *while keeping the same right eigenvectors* v_i, *for* $i = 1, \ldots, m$.
- *If* $P(\lambda_{k+1}) v_{k+1} = 0$, *then the vector* $\overset{k}{v}_{k+1} \neq 0 \in \mathbb{C}^n$ *such that* $\tilde{P}(\lambda_{k+1}) \overset{k}{v}_{k+1} = 0$, *can be computed by:*

$$\overset{k}{v}_{k+1} = \left[\prod_{j=0}^{k-1} \left(I - \frac{\lambda_{k-j} - \eta_{k-j}}{\lambda_{k+1} - \eta_{k-j}} \overset{k-j-1}{v}_{k-j} \overset{k-j-1}{x}_{k-j}^* \right) \right] v_{k+1}, \quad k = 1, \ldots, m-1, \tag{1.15}$$

where the vectors $\overset{k-j-1}{x}_{k-j}^* \in \mathbb{C}^n$ *satisfy* $\overset{k-j-1}{x}_{k-j}^* \overset{k-j-1}{v}_{k-j} = 1$ *and* $\overset{0}{v}_1 = v_1$, $\overset{0}{x}_1 = x_1$.

Proof. For the proof of this Theorem just note that we can apply Theorem 3 to the reversed matrix polynomial of $\hat{P}(\lambda)$ and by shifting λ_i to η_i. □

Example 3. *Consider again the quadratic matrix polynomial $P(\lambda)$ of Example 1 with eigenvalues: $\lambda_1 = 1, \lambda_2 = -1, \lambda_3 = 1, \lambda_4 = 1, \lambda_5 = \infty$ and $\lambda_6 = -\infty$.*

Let's shift the eigenvalues λ_1, λ_2 and λ_3 to the values: $\eta_1 = -5$, $\eta_2 = 0$ and $\eta_3 = 10$, respectively. The corresponding associated eigenvectors with the λ_i's are $v_1 = [1; 0; 0]$, $v_2 = [0; -1; 0]$ and $v_3 = [0; 1; 0]$.

Consider the vectors: $x_1 = v_1$ and $x_2 = v_2$ such that $x_i^* v_i = 1$, for $i = 1, 2, 3$. Then, using (1.14) and (1.15), we find the coefficients of the shifted matrix polynomial $\tilde{P}(\tilde{\lambda})$:

$$\tilde{P}(\tilde{\lambda}) = \tilde{\lambda}^2 \tilde{A}_2 + \tilde{\lambda}\tilde{A}_1 + \tilde{A}_0 = \tilde{\lambda}^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \tilde{\lambda} \begin{bmatrix} 4 & 0 & 1 \\ 0 & -10 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The eigenvalues of $\tilde{P}(\lambda)$ are: $\tilde{\lambda}_1 = -5, \tilde{\lambda}_2 = 0, \tilde{\lambda}_3 = 10, \tilde{\lambda}_4 = 1, \tilde{\lambda}_5 = \infty$ and $\tilde{\lambda}_6 = -\infty$.

Chapter 2 :

Invariant Pairs: Theory, Conditioning and Backward Error

2.1 Introduction

Invariant pairs, first introduced in [20] and further developed in [82] and [17], are a generalization of the notion of eigenpair for matrix polynomials. The notion of invariant pair offers a unified theoretical perspective on the problem of computing several eigenvalue-eigenvector pairs for a given matrix polynomial and extends the well-known concepts of standard pair [51] and null pair [9] (or Jordan pair [51]). As noted in the next section, the notion of invariant pairs can also be applied to more general nonlinear problems, but in this thesis, we will limit our presentation to matrix polynomials.

In the following section, we present a brief description of the existing theory on this topic found in some of the references cited above.

2.2 Previous Work on Invariant Pairs

The notion of invariant pairs for a quadratic matrix polynomial $Q(\lambda)$ is introduced in [20]. In this work, W.-J. Beyn and V. Thümmler consider the quadratic matrix polynomials that depend on the parameter $s \in \mathbb{R}$:

$$Q(\lambda, s) = A_2(s)\lambda^2 + A_1(s)\lambda + A_0(s), \quad \lambda \in \mathbb{C}$$

where it is assumed that $A_2(s)$, $A_1(s)$ and $A_0(s)$ are real square matrices depending smoothly on s. The authors investigate under which conditions smooth solution branches $(X, \Lambda) = (X(s), \Lambda(s))$ of the equation:

$$Q(\Lambda, s)X = A_2(s)X\Lambda^2 + A_1(s)X\Lambda + A_0(s)X = 0 \tag{2.1}$$

exist and how to compute them in an efficient way.

For the computation of $(X(s), \Lambda(s))$, the work presents a continuation method of predictor-corrector type (see [39], [79]) applied directly on (2.1), thus avoiding the linearization of the problem. Moreover, in the correction step of the method, a Newton-like process to generate the sequence (X_i, Λ_i, s_i) is studied. Finally, the method is demonstrated on several numerical examples such as: a homotopy between random matrices, a model of fluid conveying pipe problem and a traveling wave of a damped wave equation.

In [82], D. Kressner, inspired by the work in [20], studies a generalization of the notion of invariant pairs. To this end, consider the nonlinear eigenvalue problem:

$$(f_1(\lambda)A_1 + f_2(\lambda)A_2 + \cdots + f_m(\lambda)A_m)x = 0 \tag{2.2}$$

for holomorphic functions $f_1, \ldots, f_m : \Omega \to \mathbb{C}$ (where $\Omega \subseteq \mathbb{C}$ is an open set) and constant

matrices $A_1, \ldots, A_m \in \mathbb{C}^{n\times n}$.

In this case, an invariant pair $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ of (2.2) is defined by:

$$A_1 X f_1(S) + A_2 X f_2(S) + \cdots + A_m X f_m(S) = 0,$$

where the eigenvalues of S are contained in Ω.

Moreover, the work presents a block Newton method to compute such invariant pairs. An inverse iteration approach is used for finding the initial pair (X_0, S_0). It is explained that this method inherits the disadvantages of similar methods for solving linear eigenvalue problems: for instance, its global convergence may be erratic and a single slowly converging eigenvalue contained in S will hinder the convergence of the entire pair.

Finally, T. Betcke and D. Kressner focus on the invariant pair problem for matrix polynomials in [17]. They study the behavior of invariant pairs under perturbations of the matrix polynomial, and a first-order perturbation expansion is given. From a computational point of view, different ways to extract invariant pairs from a linearization of the matrix polynomial are analyzed. Moreover, the authors describe a refinement procedure based on Newton's method and applied directly on the polynomial formulation. This computational approach is tested in some numerical experiments.

Later work on invariant pairs can be found in [42], [43], [44], [78], [83], [96], [120].

2.3 Definition and Theory

Definition 9. *Let $P(\lambda)$ be an $n \times n$ matrix polynomial. A pair $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$, $X \neq 0$, is called an invariant pair if it satisfies the relation:*

$$P(X, S) := A_\ell X S^\ell + \cdots + A_2 X S^2 + A_1 X S + A_0 X = 0, \tag{2.3}$$

where $A_i \in \mathbb{C}^{n\times n}$, $i = 0, \ldots, \ell$, and k is an integer between 1 and $n\ell$.

Example 4. *Consider again the matrix polynomial of Example 4:*

$$P(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} -2 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then (X, S), with:

$$X = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

is an invariant pair for $P(\lambda)$.

The eigenvalues of $P(\lambda)$ *are:* $\lambda_0 = 1$ *with multiplicity 3,* $\lambda_1 = -1$, $\lambda_2 = \infty$ *and* $\lambda_3 = -\infty$. *Note that the eigenvalues of* S *are* $\lambda_S = 1$ *with multiplicity 3.*

Remark 6.

- *Infinite eigenvalues can still be covered by defining invariant pairs for the reversal polynomial* (1.2). *If a polynomial has zero and infinite eigenvalues, they have to be handled by separate invariant pairs, one for the original and one for the reverse polynomial.*

- *The definition of an invariant pair is independent of the choice of basis. Indeed, let* $T \in \mathbb{C}^{k\times k}$ *be an invertible matrix and consider* $\tilde{X} = XT$, $\tilde{S} = T^{-1}ST$. *Then, by multiplying* (2.3) *by* T *from the right, we obtain:*

$$\begin{aligned} &A_0XT + A_1XST + A_2XS^2T + \cdots + A_\ell XS^\ell T = \\ &A_0\tilde{X} + A_1\tilde{X}T^{-1}ST + A_2\tilde{X}T^{-1}S^2T + \cdots + A_\ell\tilde{X}T^{-1}S^\ell T = \\ &A_0\tilde{X} + A_1\tilde{X}\tilde{S} + A_2\tilde{X}\tilde{S}^2 + \cdots + A_\ell\tilde{X}\tilde{S}^\ell = 0. \end{aligned} \tag{2.4}$$

  Hence, $(\tilde{X}, \tilde{S})$ *is also an invariant pair.*

  Note that if S *is diagonalizable, then* T *can be chosen such that:* $\tilde{S} = T^{-1}ST = diag(\lambda_1, \ldots, \lambda_k)$. *More generally,* S *can always be chosen in Schur (upper triangular) form.*

- *The relation* (2.4) *implies that the columns* $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k$ *of the transformed basis* $\tilde{X}$ *are eigenvectors of* $P(\lambda)$: $P(\lambda_i)\tilde{x}_i = 0$, $\tilde{x}_i \neq 0$. *In particular,* S *can be transformed into Jordan form: it is then easy to see that the eigenvalues of* S *form a subset of the eigenvalues of* $P(\lambda)$.

The following definitions proposed in [17] and [52] will be helpful for our work, for instance, to allow for rank deficiency in X.

Definition 10. *A pair* $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ *is called minimal if there is* $m \in \mathbb{N}^*$ *such that:*

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

has full rank. The smallest such m *is called minimality index of* (X, S).

Definition 11. *An invariant pair (X, S) for a regular matrix polynomial $P(\lambda)$ of degree ℓ is called simple if (X, S) is minimal and the algebraic multiplicities of the eigenvalues of S are identical to the algebraic multiplicities of the corresponding eigenvalues of $P(\lambda)$.*

It is well known that eigenvectors associated with a multiple eigenvalue are unstable under perturbations, meaning that an arbitrarily small change in the matrix may cause some of the eigenvectors disappear. In contrast, the notion of invariant pairs offers a theoretical perspective and a numerically more stable approach to the task of computing several eigenpairs of a matrix polynomial.

In particular, simple invariant pairs play an important role when using a linearization approach as in [17], and ensure local quadratic convergence of Newton's method, as shown in [82]; see also [120].

2.3.1 Particular Case: Jordan Pairs

Invariant pairs are closely related to the theory of standard pairs presented in [52], and, in particular, to *Jordan pairs*. If (X, S) is a simple invariant pair and S is in Jordan form, then (X, S) is a Jordan pair.

As an example, consider the quadratic matrix polynomial of Example 4:

$$P(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} -2 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and its eigenvalues: $\lambda_0 = 1$, with algebraic multiplicity 3, and $\lambda_1 = -1$, with algebraic multiplicity 1. A corresponding Jordan pair (X, J) is given by:

$$X = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; J = \operatorname{diag}\left(-1, 1, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\right).$$

2.4 Formulation of the Invariant Pair Problem Using the Contour Integral

Polynomial eigenpairs and invariant pairs can also be defined in terms of a contour integral. Indeed, an equivalent representation for (2.3) is the following.

Proposition 3. *[42] A pair $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ is an invariant pair if and only if satisfies the relation:*

$$P(X, S) := \frac{1}{2\pi i} \oint_{\Gamma} P(\lambda)X(\lambda I - S)^{-1}d\lambda = 0, \tag{2.5}$$

where $\Gamma \subseteq \mathbb{C}$ is a contour with the spectrum of S in its interior.

This formulation allows us to choose the contour Γ to compute S with eigenvalues lying in a particular region of the complex plane. See [7], [10], [18], [19], [43], [50] for applications of the contour integral formulation.

2.5 Linearized Matrix Equation and Fréchet Derivative

In this chapter, we analyze the condition number for $P(X, S)$. To this end, we need to study the matrix equation:

$$(P + \Delta P)(X + \Delta X, S + \Delta S) = 0, \tag{2.6}$$

where $\Delta P(\lambda)$ is the perturbation of the matrix polynomial $P(\lambda)$ defined as:

$$\Delta P(\lambda) = \Delta A_0 + \Delta A_1\lambda + \Delta A_2\lambda^2 + \cdots + \Delta A_\ell\lambda^\ell.$$

Consider $\|\Delta X\| < \epsilon$, $\|\Delta S\| < \epsilon$, $\|\Delta A_i\| < \epsilon$ for some sufficiently small $\epsilon > 0$. Omitting terms of order $\mathcal{O}(\epsilon^2)$ as $\epsilon \to 0$, we have the linearized system:

$$\mathbb{L}_P(\Delta X, \Delta S) = -\Delta P(X, S) \tag{2.7}$$

with:

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

where $\mathbb{D}S^j$ denotes the Fréchet derivative of the map $S \mapsto S^j$:

$$\mathbb{D}S^j : \Delta S \mapsto \sum_{i=0}^{j-1} S^i \Delta S S^{j-i-1} \tag{2.8}$$

For example, for $\ell = 2$, we have:

$$\mathbb{L}_P(\Delta X, \Delta S) = A_0\Delta X + A_1\Delta X S + A_2\Delta X S^2 + A_1 X\Delta S + A_2 X(\Delta S S + S\Delta S).$$

2.6 Condition Number and Backward Error for the Invariant Pair Problem

In the following sections, we present formulations of the backward error and condition number for an invariant pair (X, S) of the matrix equation (2.3). We follow the ideas presented in the articles [122] and [63], which give expressions for backward errors and condition numbers for the polynomial eigenvalue problem and for a solvent of the quadratic matrix equation.

2.6.1 Condition Number for $P(X, S)$

A normwise condition number of the invariant pair (X, S) can be defined as:

$$\kappa(X,S) = \limsup_{\epsilon\to 0}\left\{ \frac{1}{\epsilon}\frac{\left\|\begin{bmatrix}\Delta X\\ \Delta S\end{bmatrix}\right\|_F}{\left\|\begin{bmatrix}X\\ S\end{bmatrix}\right\|_F} : (P+\Delta P)(X+\Delta X, S+\Delta S) = 0, \right.$$
$$\left. \|\Delta A_i\|_F \le \epsilon\alpha_i, i = 0,\dots,\ell \right\} \tag{2.9}$$

The α_i are nonnegative weights that provide flexibility in how the perturbations are measured. A common choice is $\alpha_i = \|A_i\|_F$; however, if some coefficients are to be left unperturbed, ΔA_i can be forced to zero by setting $\alpha_i = 0$.

Theorem 5. *The normwise condition number of the simple invariant pair (X, S) is given by:*

$$\kappa(X,S) = \frac{\left\|\begin{bmatrix}B_X & B_S\end{bmatrix}^{+} B_A\right\|_2}{\left\|\begin{bmatrix}X\\ S\end{bmatrix}\right\|_F}, \tag{2.10}$$

where

$$B_X = \sum_{j=0}^{\ell}\left(\left[(S^j)^T\otimes A_j\right]\right), \quad B_S = \sum_{j=1}^{\ell}\sum_{i=0}^{j-1}\left((S^{j-i-1})^T\otimes A_j X S^i\right),$$
$$B_A = \begin{bmatrix}\alpha_\ell (XS^\ell)^T\otimes I_n & \cdots & \alpha_0 X^T\otimes I_n\end{bmatrix}.$$

Proof. By expanding the first constraint in (2.9) and keeping only the first order terms,

we get:

$$\sum_{j=0}^{\ell} \Delta A_j X S^j + \sum_{j=0}^{\ell} A_j \Delta X S^j + \sum_{j=1}^{\ell} A_j X \mathbb{D}S^j(\Delta S) = O(\epsilon^2), \tag{2.11}$$

where $\mathbb{D}S^j$ denotes the Fréchet derivative (2.8). Using on equation (2.11) the vec operator (see A.2), we obtain:

- $\operatorname{vec}(\Delta P(X,S)) = \operatorname{vec}\left(\sum_{j=0}^{\ell} \Delta A_j X S^j\right) = \sum_{j=0}^{\ell} \operatorname{vec}\left(\Delta A_j X S^j\right) = \sum_{j=0}^{\ell} \left(\left[(XS^j)^T \otimes I_n\right] \operatorname{vec}(\Delta A_j)\right) =$

  $$= \begin{bmatrix} \alpha_\ell (XS^\ell)^T \otimes I_n & \cdots & \alpha_0 X^T \otimes I_n \end{bmatrix} \begin{bmatrix} \operatorname{vec}(\Delta A_\ell)/\alpha_\ell \\ \vdots \\ \operatorname{vec}(\Delta A_0)/\alpha_0 \end{bmatrix} =: B_A \operatorname{vec}(\Delta A),$$

- $\operatorname{vec}(P(\Delta X,S)) = \operatorname{vec}\left(\sum_{j=0}^{\ell} A_j \Delta X S^j\right) = \sum_{j=0}^{\ell} \operatorname{vec}\left(A_j \Delta X S^j\right) = \sum_{j=0}^{\ell} \left(\left[(S^j)^T \otimes A_j\right]\right) \operatorname{vec}(\Delta X) =$

  $=: B_X \operatorname{vec}(\Delta X),$

- $\operatorname{vec}\left(\sum_{j=1}^{\ell} A_j X \mathbb{D}S^j(\Delta S)\right) = \operatorname{vec}\left(\sum_{j=1}^{\ell} A_j X \sum_{i=0}^{j-1} S^i \Delta S S^{j-i-1}\right) = \sum_{j=1}^{\ell} \sum_{i=0}^{j-1} \operatorname{vec}\left(A_j X S^i \Delta S S^{j-i-1}\right) =$

  $$= \sum_{j=1}^{\ell} \sum_{i=0}^{j-1} \left((S^{j-i-1})^T \otimes A_j X S^i\right) \operatorname{vec}(\Delta S) =: B_S \operatorname{vec}(\Delta S).$$

Then, we have:

$$\begin{bmatrix} B_X & B_S \end{bmatrix} y = -B_A x + O(\epsilon^2),$$

where

$$y = \begin{bmatrix} \operatorname{vec}(\Delta X) \\ \operatorname{vec}(\Delta S) \end{bmatrix}, \quad \textit{and} \quad x = \begin{bmatrix} \operatorname{vec}(\Delta A_\ell)/\alpha_\ell \\ \vdots \\ \operatorname{vec}(\Delta A_0)/\alpha_0 \end{bmatrix}$$

and therefore

$$\|y\|_2 = \left\| \begin{bmatrix} \operatorname{vec}(\Delta X) \\ \operatorname{vec}(\Delta S) \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} \Delta X \\ \Delta S \end{bmatrix} \right\|_F.$$

So we have that the definition (2.9) is equivalent to the following

$$\limsup_{\epsilon \to 0} \left\{ \frac{1}{\epsilon} \frac{\|y\|_2}{\left\| \begin{bmatrix} X \\ S \end{bmatrix} \right\|_F} : \begin{bmatrix} B_X & B_S \end{bmatrix} y = -B_A x + O(\epsilon^2), \|x\|_2 \le \epsilon \right\} = \frac{\left\| \begin{bmatrix} B_X & B_S \end{bmatrix}^+ B_A \right\|_2}{\left\| \begin{bmatrix} X \\ S \end{bmatrix} \right\|_F},$$

where the matrix $\begin{bmatrix} B_X & B_S \end{bmatrix}$ has full rank if the invariant pair (X,S) is simple (see [Thm. 7, [17]]). □

Case k=1

In order to better illustrate Theorem 5, let us consider the particular case $k = 1$. When $k = 1$, invariant pairs (X, S) coincide with eigenpairs (x, λ). In this case, the matrices B_X, B_S and B_A in (2.10) are:

$$\begin{aligned}
B_X &= \sum_{j=0}^{\ell} \left(\left[(\lambda^j)^T \otimes A_j \right] \right) = \sum_{j=0}^{\ell} \left(\lambda^j A_j \right) = P(\lambda), \\
B_S &= \sum_{j=1}^{\ell} \sum_{i=0}^{j-1} \left((\lambda^{j-i-1})^T \otimes A_j x \lambda^i \right) = \sum_{j=1}^{\ell} \sum_{i=0}^{j-1} \left(\lambda^{j-1} A_j x \right) = P'(\lambda) x, \\
B_A &= \begin{bmatrix} \alpha_\ell \lambda^\ell x^T \otimes I_n & \alpha_{\ell-1} \lambda^{\ell-1} x^T \otimes I_n & \cdots & \alpha_0 x^T \otimes I_n \end{bmatrix}
\end{aligned}$$

Note that:

$$\begin{aligned}
B_A x &= \begin{bmatrix} \alpha_\ell \lambda^\ell x^T \otimes I_n & \cdots & \alpha_0 x^T \otimes I_n \end{bmatrix} \begin{bmatrix} \operatorname{vec}(\Delta A_\ell)/\alpha_\ell \\ \vdots \\ \operatorname{vec}(\Delta A_0)/\alpha_0 \end{bmatrix} = \\
&= \operatorname{vec}(\lambda^\ell \Delta A_\ell x + \cdots + \Delta A_0 x) = \operatorname{vec}(\Delta P(\lambda) x) = \Delta P(\lambda) x.
\end{aligned}$$

Therefore, we obtain:

$$\begin{aligned}
&\begin{bmatrix} B_X & B_S \end{bmatrix} y = -B_A x + O(\epsilon^2) \Leftrightarrow \begin{bmatrix} P(\lambda) & P'(\lambda) x \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = -\Delta P(\lambda) x + O(\epsilon^2) \\
&\Leftrightarrow P(\lambda)\Delta x + P'(\lambda) x \Delta\lambda + \Delta P(\lambda) x = O(\epsilon^2).
\end{aligned}$$

The last equation is consistent with the first part of the computation of the condition number for a nonzero simple eigenvalue λ of $P(\lambda)$ presented in [Thm. 5, [122]]. The second part differs, because here we are estimating $\left\| \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} \right\|_F$, whereas classical condition numbers for eigenvalue problems typically take into account angles between left and right eigenvectors. Of course, it would also be interesting to formalize a similar approach for invariant pairs, based on angles between suitable matrix manifolds (such as partially developed in [17]).

Remark 7. *The nonnegative parameters α_i are scaling parameters, which can be chosen accordingly to the problem. In practice, we choose them as: $\alpha_i = \|A_i\|_F$, for $i = 0, \ldots, \ell$.*

Numerical Examples

Example 5. *Consider Example 4 and the corresponding invariant pair given by:*

$$X = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \; S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Using the values $\alpha_2 = \|A_2\|_F$, $\alpha_1 = \|A_1\|_F$ *and* $\alpha_0 = \|A_0\|_F$, *the condition number for this problem is:*

$$\kappa(X, S) = 3.8057.$$

Example 6. *In [82], the following matrix polynomial was discussed:*

$$T(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} -1 & -6 \\ 2 & -9 \end{bmatrix} + \begin{bmatrix} 0 & 12 \\ -2 & 14 \end{bmatrix}. \tag{2.12}$$

It has eigenvalues $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 3$ *and* $\lambda_4 = 4$. *An invariant pair is given by:*

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \; S = \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}.$$

Using the values $\alpha_2 = \|A_2\|_F$, $\alpha_1 = \|A_1\|_F$ *and* $\alpha_0 = \|A_0\|_F$, *the condition number for this problem is:*

$$\kappa(X, S) = 49.1339.$$

Example 7. *Consider the quadratic matrix polynomial* $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ *with coefficients (see [80]):*

$$A_2 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \quad A_1 = A_2 - \tau \begin{bmatrix} 1+\tau & 0 \\ 0 & 1 \end{bmatrix}, \quad A_0 = -(A_2 + A_1). \tag{2.13}$$

Note that if $\tau = 0$ *and* $S = I_2$, *we have:*

$$A_2 X S^2 + A_1 X S + A_0 X = A_2 X + A_2 X + -2A_2 X = 0.$$

Then, any choice of $X \in \mathbb{C}^{2\times 2}$ *will form an invariant pair* (X, I_2). *Note also that the matrix* $P = \begin{bmatrix} B_X & B_S \end{bmatrix}$ *in the formula for the condition number* (2.10) *is:*

$$\begin{aligned} P &= \begin{bmatrix} ((S^2)^T \otimes A_2 + S^T \otimes A_1 + I_k \otimes A_0) & (I_k \otimes A_2 X S + S^T \otimes A_2 X + I_k \otimes A_1 X) \end{bmatrix} = \\ &= \begin{bmatrix} (I_2 \otimes A_2 + I_2 \otimes A_2 + I_2 \otimes -(A_2 + A_2)) & (I_2 \otimes A_2 X + I_2 \otimes A_2 X + I_2 \otimes A_2 X) \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 3I_2 \otimes A_2 X \end{bmatrix} \in \mathbb{C}^{4\times 8}. \end{aligned}$$

Here, the matrix P has rank 2, i.e., it is rank deficient and thus if $\tau = 0$, then we have:

$$\kappa(X, S) = \infty.$$

Now, we choose $X = I_2$, $\alpha_2 = \|A_2\|_F$, $\alpha_1 = \|A_1\|_F$, $\alpha_0 = \|A_0\|_F$ and let us compute $\kappa(X, S)$ for different values of τ. We obtain:

τ	10^{-1}	10^{-2}	10^{-3}	10^{-6}	10^{-10}
$\kappa(X, S)$	413.5617	42320.1986	4.2416e+006	4.2432e+012	4.7717e+015

Table 2.1: Condition numbers for matrix polynomial (2.13) using different τ

We show that $\kappa(X, S)$ tends to ∞ as $\tau \to 0$.

2.6.2 Backward Error for $P(X, S)$

Let α_i, for $i = 0, \ldots, \ell$, be nonnegative weights as in Section 2.6.1. The backward error of a computed solution $(\tilde{X}, \tilde{S}) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ to (2.3) can be defined as:

$$\eta(\tilde{X}, \tilde{S}) = \min\{\epsilon : (P + \Delta P)(\tilde{X}, \tilde{S}) = 0, \|\Delta A_i\|_F \leq \epsilon\alpha_i, i = 0, \ldots, \ell\} \tag{2.14}$$

By expanding the first constraint in (2.14) we get:

$$-P(\tilde{X}, \tilde{S}) = \Delta A_\ell \tilde{X}\tilde{S}^\ell + \cdots + \Delta A_0 \tilde{X}. \tag{2.15}$$

Then, we have

$$-P(\tilde{X}, \tilde{S}) = \begin{bmatrix} \alpha_\ell^{-1}\Delta A_\ell & \ldots & \alpha_1^{-1}\Delta A_1 & \alpha_0^{-1}\Delta A_0 \end{bmatrix} \begin{bmatrix} \alpha_\ell \tilde{X}\tilde{S}^\ell \\ \vdots \\ \alpha_1 \tilde{X}\tilde{S} \\ \alpha_0 \tilde{X} \end{bmatrix}$$

Taking the Frobenius norm, we obtain the lower bound for the backward error:

$$\eta(\tilde{X}, \tilde{S}) \geq \frac{\|P(\tilde{X}, \tilde{S})\|_F}{(\alpha_\ell^2\|\tilde{X}\tilde{S}^\ell\|_F^2 + \cdots + \alpha_1^2\|\tilde{X}\tilde{S}\|_F^2 + \alpha_0^2\|\tilde{X}\|_F^2)^{1/2}}.$$

Consider again equation (2.15). Using the vec operator (see A.2), we obtain:

$$\begin{aligned} -\operatorname{vec}(P(\tilde{X},\tilde{S})) &= ((\tilde{X}\tilde{S}^{\ell})^T \otimes I_n)\operatorname{vec}(\Delta A_\ell) + \cdots + (\tilde{X}^T \otimes I_n)\operatorname{vec}(\Delta A_0) \\ &= \begin{bmatrix} \alpha_\ell(\tilde{X}\tilde{S}^{\ell})^T \otimes I_n & \dots & \alpha_0\tilde{X}^T \otimes I_n \end{bmatrix} \begin{bmatrix} \operatorname{vec}(\Delta A_\ell)/\alpha_\ell \\ \vdots \\ \operatorname{vec}(\Delta A_0)/\alpha_0 \end{bmatrix}, \end{aligned}$$

which can be written as:

$$Hz = r,\ H \in \mathbb{C}^{nk\times(\ell+1)n^2} \tag{2.16}$$

Here we assume that H is full rank , to guarantee that (2.16) has a solution (backward error is finite). Then the backward error is the minimum 2-norm solution to:

$$\eta(\tilde{X},\tilde{S}) = \|H^+ r\|_2, \tag{2.17}$$

where H^+ denotes the Moore-Penrose pseudoinverse of H^+.
Eq. (2.17) yields an upper bound for $\eta(\tilde{X},\tilde{S})$:

$$\eta(\tilde{X},\tilde{S}) \le \|H^+\|_2\|r\|_2 = \frac{\|r\|_2}{\sigma_{\min}(H)},$$

where $\sigma_{\min}$ denotes the smallest singular value, which is nonzero by assumption. Note that:

$$\begin{aligned} \sigma_{\min}(H)^2 = \lambda_{\min}(HH^*) = \lambda_{\min}(\alpha_\ell^2(\tilde{X}\tilde{S}^{\ell})^T\overline{\tilde{X}\tilde{S}^{\ell}} \otimes I_n + \cdots + \alpha_0^2\tilde{X}^T\overline{X} \otimes I_n) \ge \\ \ge \alpha_\ell^2\sigma_{\min}(\tilde{X}\tilde{S}^{\ell})^2 + \cdots + \alpha_1^2\sigma_{\min}(\tilde{X}\tilde{S})^2 + \alpha_0^2\sigma_{\min}(\tilde{X})^2. \end{aligned}$$

Thus we obtain the upper bound for $\eta(\tilde{X},\tilde{S})$:

$$\eta(\tilde{X},\tilde{S}) \le \frac{\|P(\tilde{X},\tilde{S})\|_F}{(\alpha_\ell^2\sigma_{\min}(\tilde{X}\tilde{S}^{\ell})^2 + \cdots + \alpha_1^2\sigma_{\min}(\tilde{X}\tilde{S})^2 + \alpha_0^2\sigma_{\min}(\tilde{X})^2)^{1/2}}.$$

Case k=1:

In the particular case $k = 1$, the approximate invariant pair $(\tilde{X},\tilde{S})$ coincides with an approximate eigenpair $(\tilde{x},\tilde{\lambda})$. In this case, the definition (2.14) becomes:

$$\eta(\tilde{x},\tilde{\lambda}) = \min\{\epsilon : (P+\Delta P)(\tilde{x},\tilde{\lambda}) = 0, \|\Delta A_i\|_F \le \epsilon\alpha_i, i = 0,\dots,\ell\},$$

which is the definition of the normwise backward error of an approximate eigenpair $(\tilde{x},\tilde{\lambda})$ for $P(\lambda)x = 0$, presented in [(2.2), [122]].

Example 8. *Let us consider the* power plant problem *presented in [16] and in [123]. This is a real symmetric QEP, with $P(\lambda)$ of size 8×8, which describes the dynamic behaviour*

of a nuclear power plant simplified into an eight-degrees-of-freedom system. The problem is ill-conditioned due to the bad scaling of the matrix coefficients.

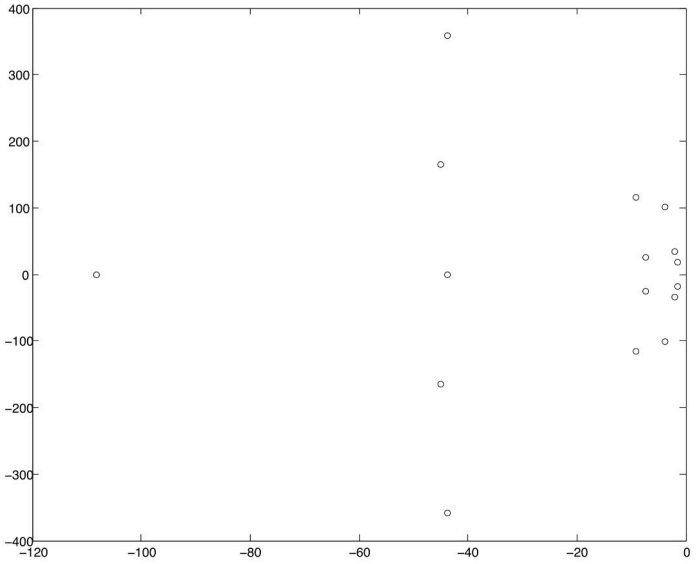

Figure 2.1: Location of the eigenvalues of the power plant problem

The maximum condition number for the eigenvalues of $P(\lambda)$, computed by the MATLAB function `polyeig`*, is:*

$$\kappa_{max} = \max_{\lambda \in \Lambda} condeig_{\lambda} = 1.0086e{+}008.$$

Using the method that will be presented in Section 4.3.1 and Section 4.5, we compute an invariant pair (X, S) associated with the 11 eigenvalues with largest condition number inside the contour $\Gamma = \gamma + \rho e^{i\theta}$ ($\gamma = 80 + 10i$, $\rho = 170$). The condition number and backward error for (X, S) are

$$\kappa(X, S) = 565.6746 \qquad and \qquad \eta(X, S) = 4.4548e - 017.$$

Observe that $\kappa(X, S)$ is significantly smaller than κ_{max}.

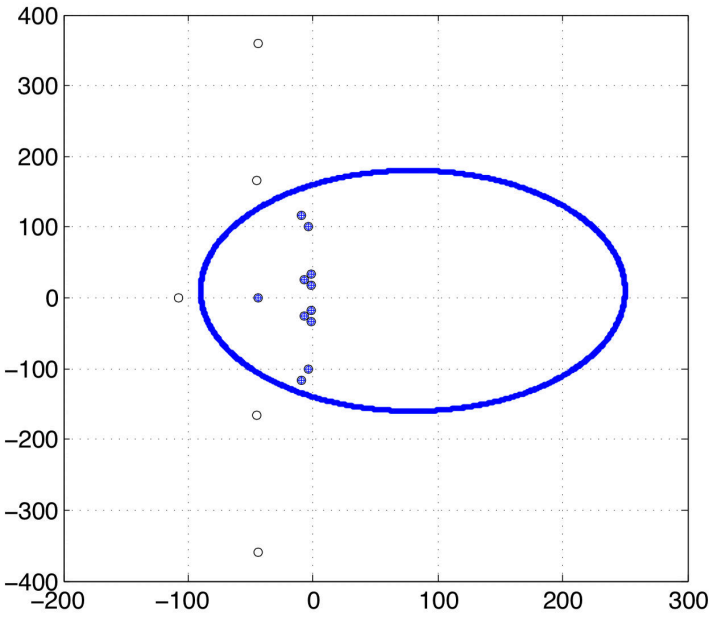

Figure 2.2: Eigenvalues (blue crosses) inside contour $\Gamma = \gamma + \rho e^{i\theta}$ ($\gamma = 80 + 10i$, $\rho = 170$)

Chapter 3 :

Matrix Solvents

3.1 Matrix Solvents: Definition and Theory

In this section we study the matrix solvent problem as a particular case of the invariant pair problem, and we apply to solvents some results we have obtained for invariant pairs.

Definition 12. *Let $P(\lambda)$ be an $n \times n$ matrix polynomial. A matrix $S \in \mathbb{C}^{n\times n}$ is called a (right) solvent for $P(S)$ if satisfies the relation:*

$$P(S) := A_\ell S^\ell + \cdots + A_2 S^2 + A_1 S + A_0 = 0. \tag{3.1}$$

A special case is, for $\ell = 2$, the quadratic matrix equation:

$$Q(S) := A_2 S^2 + A_1 S + A_0 = 0,$$

which has received considerable attention in the literature. For instance, in [62] and [63] the authors find formulations for the condition number and the backward error. They also propose functional iteration approaches based on Bernoulli's method and Newton's method with line search to compute the solution numerically.

The relation between eigenvalues of $P(\lambda)$ and solvents is highlighted in [84]: a corollary of the generalized Bézout theorem states that if S is a solvent of $P(S)$, then:

$$P(\lambda) = L(\lambda)(\lambda I - S),$$

where $L(\lambda)$ is a matrix polynomial of degree $\ell - 1$. Then any eigenpair of the solvent S is an eigenpair of $P(\lambda)$.

An equivalent representation for (3.1) that uses the contour integral is as follows.

Proposition 4. *$S \in \mathbb{C}^{n\times n}$ is a solvent if and only if*

$$P(S) := \frac{1}{2\pi\imath} \oint_\Gamma P(\lambda)(\lambda I - S)^{-1} d\lambda = 0, \tag{3.2}$$

for any closed contour $\Gamma \subseteq \mathbb{C}$ with the spectrum of S in its interior.

As for invariant pairs, this formulation allows us to choose the contour Γ to compute S with specific eigenvalues lying in a particular region of the complex plane.

3.1.1 Existence of Solvents

Let us recall some results that will be needed later. The next result is a generalization of a theorem presented in [63] which gives information about the number of solvents of $P(S)$.

Theorem 6. *Suppose $P(\lambda)$ has p distinct eigenvalues $\{\lambda_i\}_{i=1}^{p}$, with $n \leq p \leq \ell n$, and that the corresponding set of p eigenvectors $\{v_i\}_{i=1}^{p}$ satisfies the Haar condition (every subset of n of them is linearly independent). Then there are at least $\binom{p}{n}$ different solvents of $P(\lambda)$, and exactly this many if $p = \ell n$, which are given by*

$$S = W\,diag(\mu_i)W^{-1}, \quad W = \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix},$$

where the eigenpairs $(\mu_i, w_i)_{i=1}^{n}$ are chosen among the eigenpairs $(\lambda_i, v_i)_{i=1}^{p}$ of P.

Note that if we have that $p = n$ in Theorem 6, the distinctness of the eigenvalues is not needed, and then we have a sufficient condition for the existence of a solvent.

Corollary 1. *If $P(\lambda)$ has n linearly independent eigenvectors $v_1, v_2, \ldots, v_n$, then $P(S)$ has a solvent.*

An example which illustrates this last result is the following. Consider the quadratic matrix solvent problem (see [38], [63])

$$Q(S) = S^2 + \begin{bmatrix} -1 & -6 \\ 2 & -9 \end{bmatrix} S + \begin{bmatrix} 0 & 12 \\ -2 & 14 \end{bmatrix}.$$

$Q(\lambda)$ has eigenpairs (λ_i, v_i):

$$(\lambda_1, v_1) = \left(1, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right), \qquad (\lambda_2, v_2) = \left(2, \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right),$$
$$(\lambda_3, v_3) = \left(3, \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right), \qquad (\lambda_4, v_4) = \left(4, \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right).$$

Consider the subsets of eigenvectors: $\{v_1, v_2\}$, $\{v_1, v_3\}$, $\{v_1, v_4\}$, $\{v_2, v_3\}$ and $\{v_2, v_4\}$. Each subset consists of vectors that are linearly independent. Therefore, the complete set of solvents is:

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}, \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix} \text{ and } \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix}.$$

Note that we cannot construct a solvent whose eigenvalues are 3 and 4 because the associated eigenvectors are linearly dependent.

3.2 Condition Number and Backward Error for the Matrix Solvent Problem

An analysis and a computable formulation for the condition number and backward error of a solvent can be found in [62] and [63]. These results are given for the case of the quadratic solvent problem $Q(S)$.

This section presents computable expressions for the condition number and backward error for the general matrix solvent problem (3.1). We follow the ideas presented in [62], [63] and [122].

3.2.1 Condition Number of $P(S)$

We perform here a similar analysis as we did in Section 2.6.1. For self-consistency of the chapter, we will add all the details of the calculations.

A normwise condition number of the solvent S can be defined by:

$$\kappa(S) = \limsup_{\epsilon \to 0} \left\{ \frac{1}{\epsilon} \frac{||\Delta S||_F}{||S||_F} : (P + \Delta P)(S + \Delta S) = 0,\ \|\Delta A_i\|_F \leq \epsilon \alpha_i, i = 0 : \ell \right\}, \quad (3.3)$$

where $\Delta P(\lambda) = \sum_{i=0}^{\ell} \lambda^i \Delta A_i$. The α_i are nonnegative weights; in particular, ΔA_i can be forced to zero by setting $\alpha_i = 0$.

Theorem 7. *The normwise condition number of the solvent S is given by:*

$$\kappa(S) = \frac{\left|\left| \hat{B}_S^{-1} \hat{B}_A \right|\right|_2}{||S||_F}, \quad (3.4)$$

where

$$\hat{B}_S = \sum_{j=1}^{\ell} \sum_{i=0}^{j-1} \left((S^{j-i-1})^T \otimes A_j S^i \right),$$

and

$$\hat{B}_A = \begin{bmatrix} \alpha_\ell (S^\ell)^T \otimes I_n & \alpha_{\ell-1} (S^{\ell-1})^T \otimes I_n & \cdots & \alpha_0 I_{n^2} \end{bmatrix}.$$

Proof. By expanding the first constraint in (3.3) and keeping only the first order terms, we get:

$$\sum_{j=0}^{\ell} \Delta A_j S^j + \sum_{j=1}^{\ell} A_j \mathbb{D}S^j(\Delta S) = O(\epsilon^2), \quad (3.5)$$

where $\mathbb{D}S^j$ denotes the Fréchet derivative (2.8). Using on equation (3.5) the vec operator

(see A.2), we obtain:

$$\bullet\, \mathrm{vec}\left(\Delta P(S)\right) = \mathrm{vec}\left(\sum_{j=0}^{\ell} \Delta A_j S^j\right) = \sum_{j=0}^{\ell} \mathrm{vec}\left(\Delta A_j S^j\right) = \sum_{j=0}^{\ell} \left(\left[(S^j)^T \otimes I_n\right] \mathrm{vec}\left(\Delta A_j\right)\right) =$$

$$= \begin{bmatrix} \alpha_\ell (S^\ell)^T \otimes I_n & \alpha_{\ell-1}(S^{\ell-1})^T \otimes I_n & \cdots & \alpha_0 I_{n^2} \end{bmatrix} \begin{bmatrix} \mathrm{vec}\left(\Delta A_\ell\right)/\alpha_\ell \\ \mathrm{vec}\left(\Delta A_{\ell-1}\right)/\alpha_{\ell-1} \\ \vdots \\ \mathrm{vec}\left(\Delta A_0\right)/\alpha_0 \end{bmatrix} = \hat{B}_A \,\mathrm{vec}\left(\Delta A\right),$$

$$\bullet\, \mathrm{vec}\left(\sum_{j=1}^{\ell} A_j \mathbb{D} S^j(\Delta S)\right) = \mathrm{vec}\left(\sum_{j=1}^{\ell} A_j \sum_{i=0}^{j-1} S^i \Delta S S^{j-i-1}\right) = \sum_{j=1}^{\ell}\sum_{i=0}^{j-1} \mathrm{vec}\left(A_j S^i \Delta S S^{j-i-1}\right) =$$

$$= \sum_{j=1}^{\ell}\sum_{i=0}^{j-1} \left((S^{j-i-1})^T \otimes A_j S^i\right) \mathrm{vec}\left(\Delta S\right) = \hat{B}_S \,\mathrm{vec}\left(\Delta S\right).$$

Then, we have that:

$$\hat{B}_S \,\mathrm{vec}\left(\Delta S\right) = -\hat{B}_A x + O(\epsilon^2),$$

where

$$x = \begin{bmatrix} \mathrm{vec}\left(\Delta A_\ell\right) \\ \vdots \\ \mathrm{vec}\left(\Delta A_0\right) \end{bmatrix}$$

Using that $||\mathrm{vec}\left(\Delta S\right)||_2 = ||\Delta S||_F$, we have that the definition (3.3) is equivalent to the following

$$\limsup_{\epsilon\to 0} \left\{ \frac{1}{\epsilon} \frac{||\Delta S||_F}{||S||_F} : \hat{B}_S \,\mathrm{vec}\left(\Delta S\right) = -\hat{B}_A x + O(\epsilon^2), \|x\|_2 \le \epsilon \right\} = \frac{\left|\left|\hat{B}_S^{-1}\hat{B}_A\right|\right|_2}{||S||_F}.$$

□

Example 9. *Consider the quadratic matrix polynomial (see [63]):*

$$Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0 = \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ -1 & 0 \end{bmatrix},$$

with eigenvalues: $-1, 0, 0, 1$. *Note that there are three solvents for* $Q(S)$*:*

$$S_1 = \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix},\ S_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\ S_3 = \begin{bmatrix} -1 & 0 \\ -2 & 0 \end{bmatrix}.$$

Using the parameters $\alpha_2 = \|A_2\|_F$, $\alpha_1 = \|A_1\|_F$ *and* $\alpha_0 = \|A_0\|_F$, *the solvent* S_1 *has condition number:*

$$\kappa(S_1) = 3.63971.$$

Note that this condition number $\kappa(S_1)$ is equal to the one found in [63].

Remark 8. *For the solvents S_2 and S_3, the matrix $\hat{B}_S$ in Theorem 7 is singular. Then we have that $\kappa(S_2) = \infty$ and $\kappa(S_3) = \infty$.*

3.2.2 Backward Error of $P(S)$

We proceed as in Section 2.6.2 and we obtain bounds for the backward error of $P(S)$.

Let α_i, for $i = 0, \ldots, \ell$, be nonnegative weights as in Section 2.6.1. The backward error of a computed solution $\tilde{S} \in \mathbb{C}^{n\times n}$ to (3.1) can be defined as:

$$\eta(\tilde{S}) = \min\{\epsilon : (P + \Delta P)(\tilde{S}) = 0, \|\Delta A_i\|_F \leq \epsilon\alpha_i, i = 0, \ldots, \ell\} \tag{3.6}$$

By expanding the first constraint in (3.6) we get:

$$-P(\tilde{S}) = \Delta A_\ell \tilde{S}^\ell + \cdots + \Delta A_0. \tag{3.7}$$

Then, we have

$$-P(\tilde{S}) = \begin{bmatrix} \alpha_\ell^{-1}\Delta A_\ell & \alpha_{\ell-1}^{-1}\Delta A_{\ell-1} & \ldots & \alpha_1^{-1}\Delta A_1 & \alpha_0^{-1}\Delta A_0 \end{bmatrix} \begin{bmatrix} \alpha_\ell \tilde{S}^\ell \\ \alpha_{\ell-1}\tilde{S}^{\ell-1} \\ \vdots \\ \alpha_1 \tilde{S} \\ \alpha_0 \end{bmatrix}$$

Taking the Frobenius norm, we obtain the lower bound for the backward error:

$$\eta(\tilde{S}) \geq \frac{\|P(\tilde{S})\|_F}{(\alpha_\ell^2\|\tilde{S}^\ell\|_F^2 + \alpha_{\ell-1}^2\|\tilde{S}^{\ell-1}\|_F^2 + \cdots + \alpha_1^2\|\tilde{S}\|_F^2 + \alpha_0^2)^{1/2}}.$$

Consider again equation (3.7). Using the vec operator (see A.2), we obtain:

$$\begin{aligned} -\operatorname{vec}(P(\tilde{S})) &= ((\tilde{S}^\ell)^T \otimes I_n)\operatorname{vec}(\Delta A_\ell) + \cdots + (\tilde{S}^T \otimes I_n)\operatorname{vec}(\Delta A_1) + I_{n^2}\operatorname{vec}(\Delta A_0) \\ &= \begin{bmatrix} \alpha_\ell(\tilde{S}^\ell)^T \otimes I_n & \ldots & \alpha_1\tilde{S}^T \otimes I_n & \alpha_0 I_{n^2} \end{bmatrix} \begin{bmatrix} \operatorname{vec}(\Delta A_\ell)/\alpha_\ell \\ \vdots \\ \operatorname{vec}(\Delta A_1)/\alpha_1 \\ \operatorname{vec}(\Delta A_0)/\alpha_0 \end{bmatrix}, \end{aligned}$$

which can be written as:

$$Hz = r, \; H \in \mathbb{C}^{n^2\times(\ell+1)n^2} \tag{3.8}$$

Here, we assume that H has full rank to guarantee that (3.8) has a solution (backward error is finite). Then, the backward error is the minimum 2-norm solution to the least

square problem defined from (3.8), that is:

$$\eta(T) = \|H^+ r\|_2. \tag{3.9}$$

where H^+ denotes the Moore-Penrose pseudoinverse of H^+.

Eq. (3.9) yields an upper bound for $\eta(\tilde{S})$:

$$\eta(\tilde{S}) \leq \|H^+\|_2 \|r\|_2 = \frac{\|r\|_2}{\sigma_{\min}(H)},$$

where $\sigma_{\min}$ denotes the smallest singular value, which is nonzero by assumption. Note that:

$$\begin{aligned}
\sigma_{\min}(H)^2 &= \lambda_{\min}(HH^*) \\
&= \lambda_{\min}(\alpha_\ell^2 (\tilde{S}^\ell)^T \overline{\tilde{S}^\ell} \otimes I_n + \cdots + \alpha_1^2 (\tilde{S})^T \overline{\tilde{S}} \otimes I_n + \alpha_0^2 I_{n^2}) \\
&= \lambda_{\min}(\alpha_\ell^2 (\tilde{S}^\ell)^* \tilde{S}^\ell \otimes I_n + \cdots + \alpha_1^2 (\tilde{S})^* \tilde{S} \otimes I_n + \alpha_0^2 I_{n^2}) \\
&\geq \alpha_\ell^2 \sigma_{\min}(\tilde{S}^\ell)^2 + \cdots + \alpha_1^2 \sigma_{\min}(\tilde{S})^2 + \alpha_0^2.
\end{aligned}$$

Thus we obtain the upper bound for $\eta(\tilde{S})$:

$$\eta(\tilde{S}) \leq \frac{\|P(\tilde{S})\|_F}{(\alpha_\ell^2 \sigma_{\min}(\tilde{S}^\ell)^2 + \cdots + \alpha_1^2 \sigma_{\min}(\tilde{S})^2 + \alpha_0^2)^{1/2}}.$$

3.3 Computation of Solvents

The question of designing symbolic algorithms for computing solvents remains relatively unexplored. Attempts have been made to formulate the problem as a system of polynomial equations, which can be solved via standard methods. However, this approach becomes cumbersome for problems of large size (see [62]).

Motivated by applications to differential equations [27], we study an approach to the symbolic or symbolic-numeric computation of solvents based on the moment method, by specializing the results presented in Section 4.1.

3.3.1 A Computational Approach

Our approach to compute matrix solvents is based on the relation between the matrix solvent problem (3.1) and the invariant pair problem (2.3). We state this in the following result.

Theorem 8. *Let $P(\lambda)$ be a $n \times n$ matrix polynomial and consider an invariant pair $(Y, T) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ of $P(\lambda)$. If the matrix Y has size $n \times n$, i.e. $k = n$, and is*

invertible, then $S = YTY^{-1}$ *satisfies equation* (3.1)*, i.e.,* S *is a matrix solvent of* $P(\lambda)$*.*

Proof. As $(Y, T) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ is an invariant pair of $P(\lambda)$, we have:

$$A_\ell YT^\ell + \cdots + A_2YT^2 + A_1YT + A_0Y = 0.$$

Since Y is invertible, we can post-multiply by Y^{-1}. Then we get:

$$\begin{aligned} &A_\ell YT^\ell Y^{-1} + \cdots + A_2YT^2Y^{-1} + A_1YTY^{-1} + A_0 = 0 \Leftrightarrow \\ &A_\ell S^\ell + \cdots + A_2S^2 + A_1S + A_0 = 0. \end{aligned}$$

Therefore, S is a matrix solvent of $P(\lambda)$. □

Example 10. *Consider again the quadratic matrix polynomial of previous section:*

$$Q(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} -1 & -6 \\ 2 & -9 \end{bmatrix} + \begin{bmatrix} 0 & 12 \\ -2 & 14 \end{bmatrix}.$$

Invariant pairs (X_i, T_i) *for* $P(\lambda)$ *are given by:*

$$X_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \; T_1 = \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}; \qquad X_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \; T_2 = \begin{bmatrix} 0 & -2 \\ 1 & 3 \end{bmatrix};$$

$$X_3 = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}, \; T_3 = \begin{bmatrix} 0 & -6 \\ 1 & 5 \end{bmatrix}; \qquad X_4 = \begin{bmatrix} 3 & 11 \\ 4 & 12 \end{bmatrix}, \; T_4 = \begin{bmatrix} 0 & -3 \\ 1 & 4 \end{bmatrix};$$

$$X_5 = \begin{bmatrix} -1 & 2 \\ 1 & 4 \end{bmatrix}, \; T_5 = \begin{bmatrix} 0 & -4 \\ 1 & 5 \end{bmatrix}; \qquad X_6 = \begin{bmatrix} 1 & 4 \\ 2 & 6 \end{bmatrix}, \; T_6 = \begin{bmatrix} 0 & -8 \\ 1 & 6 \end{bmatrix}.$$

Taking $S_i = X_iT_iY_i^{-1}$*, for* $i = 1, \ldots, 6$*, the* S_i *satisfying* (3.1) *are:*

$$S_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \; S_3 = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}, \; S_4 = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}, \; S_5 = \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix} \text{ and } S_6 = \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix}.$$

Note that the matrix X_1 *is singular and then we can't construct the solvent* S_1 *associated to the eigenvalues 3 and 4 (the associated eigenvectors are linearly dependent).*

3.3.2 Matrix p-th Root

Consider the matrix equation

$$X^p - A = 0, \tag{3.10}$$

where $A \in \mathbb{C}^{n\times n}$. A matrix X satisfying (3.10) is called a p-th root of A (see, e.g., [55], [56], [65], [76], [118]). One application of p-th roots is in the computation of the matrix

logarithm through the relation (see [30])

$$\log A = p \log A^{\frac{1}{p}}.$$

When $p = 2$ we have the matrix equation:

$$X^2 - A = 0. \tag{3.11}$$

A solution X of (3.11) is called a matrix square root of A (see, e.g., [59], [60], [61], [75], [100]). In this case, if A is singular, the existence of a square root depends on the Jordan structure of the zero eigenvalues (see [32]). For instance, the matrix:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

has no square root, while the matrix

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

does have a square root, specifically:

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

If A is nonsingular, it always has a p-th root. The number of square roots varies from two (for a nonsingular Jordan block) to infinity (any involutory matrix is a square root of the identity matrix). Moreover, when A has no non-positive real eigenvalues, one can define the notion of principal root, denoted by $A^{\frac{1}{2}}$.

The method of reference for computing matrix square roots is to compute a Schur decomposition, compute a square root of the triangular factor, and then transform back (see [24] and [60]).

Iterative methods are alternatives for computing p-th roots. In [59], [61], [75], [76], [100], for instance, several variations of the Newton's method to approximate the (square roots) p-th roots are presented.

Note that equation (3.11) is a particular case of the monic matrix solvent problem of degree p:

$$P(S) := S^p + A_{p-1}S^{p-1} + \cdots + A_2S^2 + A_1S + A_0 = 0,$$

when the matrices A_i are zero for $i = 1, \ldots, p-1$. Therefore, we can use our (block) moment method to construct a solvent S satisfying (3.10) (see Section 4.3.2).

Example 11. *Consider the matrix equation (see [118]):*

$$X^4 = A,$$

where:

$$A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 1.3 & -1 & -1 \\ 0 & 0 & 1.7 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix}.$$

Consider the circle: $\Gamma = 1 + \frac{1}{4}e^{it}$. *Applying our moment method, which will be presented in Chapter 4, we construct the matrix solvent* $\tilde{X}$*:*

$$\tilde{X} = \begin{bmatrix} 1 & -0.2259665745747 & -0.2609342676468 & -0.3057660919094 \\ 0 & 1.067789972372 & -0.1851709326575 & -0.212512633424 \\ 0 & 0 & 1.141858345435 & -0.157829231891 \\ 0 & 0 & 0 & 1.189207115003 \end{bmatrix}.$$

The residual is:

$$res(\tilde{X}) = \frac{\|\tilde{X}^4 - A\|_2}{\|A\|_2} = 3.1024e - 13.$$

3.4 Solvents and Triangularized Matrix Polynomials

Motivated by the results in [121] and [124], where the authors analyze a method for triangularizing the matrix polynomial $P(\lambda)$, we aim here to study the relation between solvents of general and triangularized matrix polynomials.

3.4.1 Triangularizing Matrix Polynomials

For any algebraically closed field $\mathbb{F}$, any matrix polynomial $P(\lambda) \in \mathbb{F}[\lambda]^{n\times m}$, with $n \leq m$, can be reduced to triangular form via unimodular transformations. In other words, there exist matrix polynomials $E(\lambda)$ and $F(\lambda)$ with nonzero constant determinant such that:

$$T(\lambda) = E(\lambda)P(\lambda)F(\lambda),$$

where $T(\lambda)$ is monic triangular and preserves the degree and the finite and infinite elementary divisors of $P(\lambda)$ [121], [124].

Theorem 9. *[121] For an algebraically closed field $\mathbb{F}$, any $P(\lambda) \in \mathbb{F}[\lambda]^{n\times m}$ with $n \leq m$ is triangularizable.*

A continuation, we recall the procedure to triangularize a quadratic matrix polynomial $Q(\lambda)$ (see [121]).

3.4.2 Procedure to Triangularize a Quadratic Matrix Polynomial

Theorem 1.7 in [51] shows that any complex matrix polynomial of degree ℓ, with nonsingular leading coefficient, is equivalent to a monic triangular matrix polynomial of the same degree. In this section, we recall the procedure for $\ell = 2$, i.e., for the quadratic matrix polynomial $Q(\lambda)$.

1. Compute the invariant factors: $\alpha_1(\lambda) \mid \alpha_2(\lambda) \mid \cdots \mid \alpha_n(\lambda)$ of $Q(\lambda)$ and consider $D(\lambda) = \mathrm{diag}(\alpha_1(\lambda), \ldots, \alpha_n(\lambda))$.

2. If $\deg(\alpha_1) = 2$, then $\deg(\alpha_i) = 2$, $i = 2 : n$. Then $D(\lambda)$ is a monic triangular quadratic matrix polynomial equivalent to $Q(\lambda)$ and the construction is done. Otherwise, go to *Step 3*.

3. If $\ell_1 = \deg(\alpha_1) < 2$, then $\ell_n = \deg(\alpha_n) > 2$ and there is a monic polynomial $s(\lambda)$ of degree $\ell_n - 2$ such that $\alpha_{j-1}(\lambda) \mid \alpha_1(\lambda)s(\lambda) \mid \alpha_j$, for some index j, $1 < j \leq n$. The $s(\lambda)$ is the product of $\alpha_{j-1}(\lambda)/\alpha_1(\lambda)$ and some of the factors in the prime factorization of $\alpha_j(\lambda)/\alpha_{j-1}(\lambda)$.

4. Perform the following elementary transformations on $D(\lambda)$:

   (i) Add to column n the first column multiplied by $s(\lambda)$. Then add to row n the first row multiplied by $-\alpha_n(\lambda)/(\alpha_1(\lambda)s(\lambda))$ and permute columns 1 and n.

   (ii) Successively interchange rows k and $k+1$ for $k = 1, 2, \ldots, j-2$, so that rows $1, 2, \ldots, j-2, j-1$ of the new matrix are rows $2, 3, \ldots, j-1$ and 1, respectively, of the former one.

   (iii) Permute columns 1 to $j-1$ in the same way as the rows in (ii).

The resulting matrix polynomial has the form:

$$T_1(\lambda) = \begin{bmatrix} \alpha_2(\lambda) & & & & & & & \\ & \ddots & & & & & & \\ & & \alpha_{j-1}(\lambda) & & & & & \\ & & & \alpha_1(\lambda)s(\lambda) & \cdots & & \cdots & \alpha_1(\lambda) \\ & & & & \alpha_j(\lambda) & & & \vdots \\ & & & & & \ddots & & \\ & & & & & & \alpha_{n-1}(\lambda) & \vdots \\ & & & & & & & -\alpha_n(\lambda)/s(\lambda) \end{bmatrix}.$$

5. Let $D_1(\lambda) = \mathrm{diag}(\alpha_2(\lambda), \ldots, \alpha_{j-1}(\lambda), \alpha_1(\lambda)s(\lambda), \ldots, \alpha_{n-1}(\lambda))$. If $D_1(\lambda)$ is diagonal of degree 2, the construction is done; otherwise, use $D_1(\lambda)$ to choose the new index j and polynomials $s(\lambda)$ as in Step 3 and perform the elementary transformations described in Step 4 on the whole matrix $T_1(\lambda)$. Repeat these steps until all diagonal entries have degree 2.

What can we say about solvents for a given matrix polynomial $P(\lambda)$ and for the associated triangularized polynomial? A partial answer will be given in Theorem 11.

Theorem 10. *For any $\ell n \times \ell n$ monic linearization $\lambda I - A$ of $P(\lambda) \in \mathbb{C}[\lambda]^{n\times n}$ with nonsingular leading coefficient, there exists $U \in \mathbb{C}^{n\times \ell n}$ with orthogonal columns such that $M = \begin{bmatrix} U \\ UA \\ \vdots \\ UA^{\ell-1} \end{bmatrix}$ is nonsingular and $\lambda I - MAM^{-1}$ is a linearization for the polynomial $T(\lambda) = \lambda^\ell I + \lambda^{\ell-1}T_{\ell-1} + \cdots + \lambda^2 T_2 + \lambda T_1 + T_0$, which is upper triangular and equivalent to $P(\lambda)$.*

Theorem 10 is a straightforward generalization of a result found in [124]. Note that, for the time being, we assume that the leading coefficient A_ℓ is nonsingular.

Theorem 11. *Let $P(\lambda)$ be a $n \times n$ matrix polynomial and consider the linearization:*

$$A = \begin{bmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ -A_0 & -A_1 & -A_2 & \cdots & -A_{\ell-1} \end{bmatrix}.$$

Let M be as in Theorem 10 and let Y_1 be the first $n \times n$ block of the matrix $M^{-1}\begin{bmatrix} I_n \\ S_t \\ \vdots \\ S_t^{\ell-1} \end{bmatrix}$. *If Y_1 is nonsingular and S_t is a solvent for the triangularized problem, i.e.,* $T(S_t) = 0$, *then $S = Y_1 S_t Y_1^{-1}$ is a solvent for $P(S)$.*

Proof. Note that:

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} S_t - S_t \\ S_t^2 - S_t^2 \\ \vdots \\ -T_0 - T_1 S_t - T_2 S_t^2 - \cdots - S_t^\ell \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ -T_0 & -T_1 & -T_2 & \cdots & -T_{\ell-1} \end{bmatrix} \begin{bmatrix} I_n \\ S_t \\ \vdots \\ S_t^{\ell-1} \end{bmatrix} - \begin{bmatrix} S_t \\ S_t^2 \\ \vdots \\ S_t^\ell \end{bmatrix} =$$

$$= M^{-1} \begin{bmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ -T_0 & -T_1 & -T_2 & \cdots & -T_{\ell-1} \end{bmatrix} M M^{-1} \begin{bmatrix} I_n \\ S_t \\ \vdots \\ S_t^{\ell-1} \end{bmatrix} - M^{-1} \begin{bmatrix} I_n \\ S_t \\ \vdots \\ S_t^{\ell-1} \end{bmatrix} S_t \overset{(iii)}{=}$$

$$= \begin{bmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ -A_0 & -A_1 & -A_2 & \cdots & -A_{\ell-1} \end{bmatrix} M^{-1} \begin{bmatrix} I_n \\ S_t \\ \vdots \\ S_t^{\ell-1} \end{bmatrix} - M^{-1} \begin{bmatrix} I_n \\ S_t \\ \vdots \\ S_t^{\ell-1} \end{bmatrix} S_t.$$

Since $M^{-1}\begin{bmatrix} I_n \\ S_t \\ \vdots \\ S_t^{\ell-1} \end{bmatrix}$ has size $\ell n \times n$, let us partition it as $\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\ell \end{bmatrix}$, where $Y_i \in \mathbb{C}^{n \times n}$ for

$i = 1, \ldots, \ell$. Then:

$$\begin{bmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ -A_0 & -A_1 & -A_2 & \cdots & -A_{\ell-1} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\ell \end{bmatrix} - \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\ell \end{bmatrix} S_t = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Then we have:

$$Y_i = Y_1 S_t^{i-1}, \text{ for } i = 2, \ldots, \ell; \tag{3.12}$$

$$-A_0 Y_1 - A_1 Y_2 - \cdots - A_{\ell-1} Y_\ell - Y_\ell S_t = 0. \tag{3.13}$$

Substituting equations (3.12) in (3.13) we obtain:

$$0 = Y_1 S_t^\ell + A_{\ell-1} Y_1 S_t^{\ell-1} + \cdots + A_1 Y_1 S_t + A_0 Y_1.$$

If Y_1 is invertible we have:

$$0 = Y_1 S_t^\ell Y_1^{-1} + A_{\ell-1} Y_1 S_t^{\ell-1} Y_1^{-1} + \cdots + A_1 Y_1 S_t Y_1^{-1} + A_0.$$

Taking $S = Y_1 S_t Y_1^{-1}$ we have:

$$0 = S^\ell + A_{\ell-1} S^{\ell-1} + \cdots + A_2 S^2 + A_1 S + A_0 := P(S).$$

Then $S = Y_1 S_t Y_1^{-1}$ is a solvent for $P(S)$. □

Example 12. *Consider the matrix polynomial:*

$$P(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} -1 & -6 \\ 2 & -9 \end{bmatrix} + \begin{bmatrix} 0 & 12 \\ -2 & 14 \end{bmatrix}.$$

Let us triangularize $P(\lambda)$ using the procedure in Section 3.4.2. One possible result is:

$$T(\lambda) = \begin{bmatrix} (\lambda-1)(\lambda-2) & 1 \\ 0 & (\lambda-3)(\lambda-4) \end{bmatrix} = \begin{bmatrix} \lambda^2 - 3\lambda + 2 & 1 \\ 0 & \lambda^2 - 7\lambda + 12 \end{bmatrix} =$$
$$= \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} -3 & 0 \\ 0 & -7 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 0 & 12 \end{bmatrix} = \lambda^2 I_2 + \lambda T_1 + T_0.$$

Now, suppose that the solvent $S_t \in \mathbb{C}^{2\times 2}$ of the triangularized problem is in upper trian-

gular form, i.e.:

$$S_t = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix}.$$

Then, we have:

$$T(S_t) = S_t^2 + T_1 S_t + T_0 = \begin{bmatrix} a^2 - 3a + 2 & ab - 3b + bd + 1 \\ 0 & d^2 - 7d + 12 \end{bmatrix}.$$

In the task of solving the problem $T(S_t) = 0$, we see that:

$$\begin{cases} a^2 - 3a + 2 = 0 \\ ab - 3b + bd + 1 = 0 \\ d^2 - 7d + 12 = 0 \end{cases} \implies \begin{array}{c} a = 1 \text{ or } a = 2 \\ d = 3 \text{ or } d = 4 \end{array}$$

Then, we have four cases:

Case 1: If $a = 1$ and $d = 3$, then $b = -1$. We have:

$$S_{t_1} = \begin{bmatrix} 1 & -1 \\ 0 & 3 \end{bmatrix}.$$

Case 2: If $a = 1$ and $d = 4$, then $b = -\frac{1}{2}$. We have:

$$S_{t_2} = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 4 \end{bmatrix}.$$

Case 3: If $a = 2$ and $d = 3$, then $b = -\frac{1}{2}$. We have:

$$S_{t_3} = \begin{bmatrix} 2 & -\frac{1}{2} \\ 0 & 3 \end{bmatrix}.$$

Case 4: If $a = 2$ and $d = 4$, then $b = -\frac{1}{3}$. We have:

$$S_{t_4} = \begin{bmatrix} 2 & -\frac{1}{3} \\ 0 & 4 \end{bmatrix}.$$

Having the solvents S_{t_i} for the triangularized problem, we would like to find the (associated) solvents S_i for the original problem $P(\lambda)$.

Consider the linearization of $P(\lambda)$*:*

$$A = \begin{bmatrix} 0 & I_2 \\ -A_0 & -A_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -12 & 1 & 6 \\ 2 & -14 & -2 & 9 \end{bmatrix},$$

and the matrix M*:*

$$M = \begin{bmatrix} U \\ UA \end{bmatrix} = \begin{bmatrix} -20 & 15 & 8 & -7 \\ -2 & 0 & 2 & 0 \\ -14 & 2 & 2 & 0 \\ 0 & -24 & 0 & 12 \end{bmatrix}.$$

Now, compute $M^{-1}\begin{bmatrix} I_2 \\ S_{t_i} \end{bmatrix}$ *for* $i = 1, \ldots, 4$*. Taking* Y_i *as the first* 2×2 *block of this result and computing* $S_i = Y_i S_{t_i} Y_i^{-1}$*, we have:*

$$S_1 = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}, S_2 = \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix}, S_3 = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix} \text{ and } S_4 = \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix}.$$

By Theorem 11, the S_i *are solvents for* $P(S)$*, i.e., they satisfy* $P(S_i) := S_i^2 + A_1 S_i + A_0 = 0$ *for* $i = 1, \ldots, 4$*.*

Note that in the description of the procedure to triangularize a quadratic matrix polynomial in Section 3.4.2, we have the freedom when choosing the polynomial $s(\lambda)$. Therefore, the final triangularization for the matrix polynomial will change depending on our choice. Moreover, the number of solvents may also change.

To demonstrate this, consider the following example.

Example 13. *Consider again the matrix polynomial* $P(\lambda)$ *of Example 12, but now with its different triangularization:*

$$\hat{T}(\lambda) = \begin{bmatrix} (\lambda-1)(\lambda-3) & 1 \\ 0 & (\lambda-2)(\lambda-4) \end{bmatrix} = \begin{bmatrix} \lambda^2 - 4\lambda + 3 & 1 \\ 0 & \lambda^2 - 6\lambda + 8 \end{bmatrix} =$$
$$= \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} -4 & 0 \\ 0 & -6 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 0 & 8 \end{bmatrix} = \lambda^2 I_2 + \lambda T_1 + T_0.$$

Suppose that the solvent $S_t \in \mathbb{C}^{2\times 2}$ *of the triangularized problem is in upper triangular form, i.e.:*

$$S_t = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix}.$$

Then, we have:

$$\hat{T}(S_t) = S_t^2 + T_1 S_t + T_0 = \begin{bmatrix} a^2 - 4a + 3 & ab + bd - 4b + 1 \\ 0 & d^2 - 6d + 8 \end{bmatrix}.$$

When solving the problem $T(S_t) = 0$*, we see that:*

$$\begin{cases} a^2 - 4a + 3 = 0 \\ ab + bd - 4b + 1 = 0 \\ d^2 - 6d + 8 = 0 \end{cases} \implies \begin{array}{c} a = 1 \text{ or } a = 3 \\ d = 2 \text{ or } d = 4 \end{array}$$

Then, we have four cases:

Case 1: If $a = 1$ *and* $d = 2$*, then* $b = 1$*. We have:*

$$S_{t_1} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}.$$

Case 2: If $a = 1$ *and* $d = 4$*, then* $b = -1$*. We have:*

$$S_{t_2} = \begin{bmatrix} 1 & -1 \\ 0 & 4 \end{bmatrix}.$$

Case 3: If $a = 3$ *and* $d = 2$*, then* $b = -1$*. We have:*

$$S_{t_3} = \begin{bmatrix} 3 & -1 \\ 0 & 2 \end{bmatrix}.$$

Case 4: If $a = 3$ *and* $d = 4$*, then* $b = -\frac{1}{3}$*. We have:*

$$S_{t_4} = \begin{bmatrix} 3 & -\frac{1}{3} \\ 0 & 4 \end{bmatrix}.$$

Having the solvents S_{t_i} *for the triangularized problem, we would like to find the (associated) solvents* S_i *for the original problem* $P(\lambda)$*.*

Consider the linearization of $P(\lambda)$*:*

$$A = \begin{bmatrix} 0 & I_2 \\ -A_0 & -A_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -12 & 1 & 6 \\ 2 & -14 & -2 & 9 \end{bmatrix},$$

and the matrix M:

$$M = \begin{bmatrix} U \\ UA \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{13}{3} & \frac{1}{3} & -\frac{5}{3} \\ 2 & -5 & -2 & 3 \\ -\frac{10}{3} & \frac{58}{3} & \frac{13}{3} & -\frac{26}{3} \\ 6 & -18 & -6 & 10 \end{bmatrix}.$$

Now, compute $M^{-1}\begin{bmatrix} I_2 \\ S_{t_i} \end{bmatrix}$ *for* $i = 1, \ldots, 4$. *Taking* Y_i *as the first* 2×2 *block of this result and computing* $S_i = Y_i S_{t_i} Y_i^{-1}$, *we have:*

$$S_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, S_2 = \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix}, S_3 = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}.$$

Note that the matrix Y_4 *is equal to:*

$$Y_4 = \begin{bmatrix} 1 & \frac{4}{3} \\ 1 & \frac{4}{3} \end{bmatrix}.$$

This matrix is singular and then we can't construct the solvent S_4 *associated to the eigenvalues 3 and 4 (the associated eigenvectors are linearly dependent).*

By Theorem 11, the S_i *are solvents for* $P(S)$, *i.e., they satisfy* $P(S_i) := S_i^2 + A_1 S_i + A_0 = 0$ *for* $i = 1, 2, 3$.

Remark 9. *The computation of the matrix* M *in the Examples 12 and 13 was done following the proof of Theorem* 5.2 *in [121].*

3.4.3 A Problem with an Infinite Number of Solvents

What happens to the ideas outlined above when working on problems with an infinite number of solvents? Here is an example taken from [111].

Consider the matrix polynomial:

$$P(\lambda) = \lambda^2 I + \lambda \begin{bmatrix} -7 & -2 & -2 \\ \frac{3}{31} & \frac{-203}{31} & \frac{8}{31} \\ \frac{-13}{31} & \frac{-40}{31} & \frac{-231}{31} \end{bmatrix} + \begin{bmatrix} 13 & 9 & 7 \\ \frac{-21}{31} & \frac{294}{31} & \frac{-36}{31} \\ \frac{60}{31} & \frac{183}{31} & \frac{435}{31} \end{bmatrix}.$$

One possible triangularization for $P(\lambda)$ is:

$$T(\lambda) = \begin{bmatrix} (\lambda-3)(\lambda-4) & (\lambda-3) & 0 \\ 0 & (\lambda-3)^2 & 1 \\ 0 & 0 & (\lambda-4)^2 \end{bmatrix} = \lambda^2 I_2 + \lambda T_1 + T_0 =$$
$$= \lambda^2 I + \lambda \begin{bmatrix} -7 & 1 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & -8 \end{bmatrix} + \begin{bmatrix} 12 & -3 & 0 \\ 0 & 9 & 1 \\ 0 & 0 & 16 \end{bmatrix}.$$

Now, suppose that the solvent $S_t \in \mathbb{C}^{3\times 3}$ of the triangularized problem is in upper triangular form, i.e.:

$$S_t = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ 0 & x_{22} & x_{23} \\ 0 & 0 & x_{33} \end{bmatrix}.$$

Then, we have:

$$T(S_t) = S_t^2 + T_1 S_t + T_0 =$$
$$= \begin{bmatrix} (x_{11}-3)(x_{11}-4) & x_{22} - 7x_{12} + x_{11}x_{12} + x_{12}x_{22} - 3 & x_{23} - 7x_{13} + x_{11}x_{13} + x_{12}x_{23} + x_{13}x_{33} \\ 0 & (x_{22}-3)^2 & x_{22}x_{23} - 6x_{23} + x_{23}x_{33} + 1 \\ 0 & 0 & (x_{33}-4)^2 \end{bmatrix}.$$

In the task of solving the problem $T(S_t) = 0$, we see that: $x_{11} = 3$ or $x_{11} = 4$, $x_{22} = 3$ and $x_{33} = 4$. Then we have two cases:

I. If $x_{11} = 3$, $x_{22} = 3$ and $x_{33} = 4$:
Then we find that $x_{23} = -1$, $x_{12} = 0$ and $x_{12} = -1$, which is a contradiction. In this case, there is no solution and then we can't construct a solvent.

II. If $x_{11} = 4$, $x_{22} = 3$ and $x_{33} = 4$:
Then we find that $x_{23} = -1$ and $x_{13} = x_{12} + 1$. In this case, the solvent S_t has the form:
$$S_t = \begin{bmatrix} 4 & x_{12} & x_{12}+1 \\ 0 & 3 & -1 \\ 0 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 3 & -1 \\ 0 & 0 & 4 \end{bmatrix} + x_{12} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
for $x_{12} \in \mathbb{C}$.

Thus $T(\lambda)$ has an infinite number of solvents and the same holds for $P(\lambda)$.

Chapter 4 :

Moments, Hankel Pencils and Computation of Invariant Pairs

4.1 Introduction

Numerical methods based on contour integrals for the computation of eigenvalues of matrix polynomials and analytic matrix-valued functions have recently met with growing interest. Such techniques are related to the well-known method of moments, where the moments are computed by numerical quadrature (see [53]).

In this section, we explore a similar approach for computing invariant pairs. Our main reference is the Sakurai-Sugiura method (see [7] and [115]) as well as the presentation given in [18].

4.2 Toeplitz and Hankel Matrices

A Toeplitz matrix is a matrix where each descending diagonal from left to right is constant (see [54]). Formally:

Definition 13. *An $n \times n$ matrix T is Toeplitz if there exist scalars $r_{-n+1}, \ldots, r_0, \ldots, r_{n-1}$ such that $[T]_{ij} = r_{i-j}$ for $1 \leq i, j \leq n$.*

Then, an $n \times n$ Toeplitz matrix has the form:

$$T = \begin{bmatrix} r_0 & r_{-1} & r_{-2} & \cdots & \cdots & r_{-n+1} \\ r_1 & r_0 & r_{-1} & \ddots & & \vdots \\ r_2 & r_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & r_{-1} & r_{-2} \\ \vdots & & \ddots & r_1 & r_0 & r_{-1} \\ r_{n-1} & \cdots & \cdots & r_2 & r_1 & r_0 \end{bmatrix}.$$

The Toeplitz structure can be exploited to design fast algorithms for fundamental problems such as solving linear systems. For instance, when T is symmetric and positive definite, the Toeplitz system $Tx = b$ can be solved in $O(n^2)$ flops. This is due to the structure of the inverse T^{-1}.

Definition 14. *An $n \times n$ matrix H is Hankel if there exist scalars $h_0, \ldots, h_{2n-2}$ such that $[H]_{i+1,j+1} = h_{i+j}$ for $i, j = 0, 1, \ldots, n-1$.*

Then, an $n \times n$ Hankel matrix has the form:

$$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & h_2 & \cdots & h_n \\ \vdots & \vdots & & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{bmatrix}.$$

Define the "flip matrix":

$$\mathcal{E}_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Then, an $n \times n$ matrix H is Hankel if and only if it can be written as $H = T\,\mathcal{E}_n$, where T is a suitable Toeplitz matrix. Conversely, an $n \times n$ matrix T is Toeplitz if and only if $T = H\,\mathcal{E}_n$, where T is a suitable Hankel matrix.

4.3 The Moment Method and Eigenvalues

Let us begin by briefly recalling a few basic facts about the Sakurai-Sugiura moment method. Here, we essentially follow the presentation given in [7].

Let $P(\lambda)$ be an $n \times n$ matrix polynomial and let Γ be a closed contour in the complex plane and let u and v be arbitrarily given vectors in $\mathbb{C}^n$. Define the function:

$$f(\lambda) := u^H P(\lambda)^{-1} v.$$

In the following, it will be understood that no eigenvalue of $P(\lambda)$ should lie exactly on the contour Γ: each eigenvalue should be either inside or outside the contour.

The next theorem, which can be found in [7], gives a representation for $f(\lambda)$ that will be useful later on.

Theorem 12. *[Thm. 3.1, [7]] Let $D(\lambda) = \operatorname{diag}(d_1(\lambda), \ldots, d_n(\lambda))$ be the Smith form of $P(\lambda)$, and let $E(\lambda)$ and $F(\lambda)$ be as in (1.7). Let $\chi_j(\lambda) = u^H \boldsymbol{q}_j(\lambda)\boldsymbol{p}_j(\lambda)^H v$, $1 \le j \le n$. Then*

$$f(\lambda) = \sum_{j=1}^{n} \frac{\chi_j(\lambda)}{d_j(\lambda)}, \tag{4.1}$$

where $\boldsymbol{q}_j(\lambda)$ and $\boldsymbol{p}_j(\lambda)$ are the column vectors of $E(\lambda)$ and $F(\lambda)^H$, respectively.

Definition 15. *Let $k \in \mathbb{N}$. The k-th moment of $f(z)$ is:*

$$\mu_k = \frac{1}{2\pi\imath} \oint_\Gamma z^k f(z) dz. \tag{4.2}$$

For a positive integer m, define the Hankel matrices $H_0, H_1 \in \mathbb{C}^{m\times m}$ as follows:

$$H_0 = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{m-1} \\ \mu_1 & \mu_2 & \cdots & \mu_m \\ \vdots & \vdots & & \vdots \\ \mu_{m-1} & \mu_m & \cdots & \mu_{2m-2} \end{bmatrix}, \quad H_1 = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_m \\ \mu_2 & \mu_3 & \cdots & \mu_{m+1} \\ \vdots & \vdots & & \vdots \\ \mu_m & \mu_{m+1} & \cdots & \mu_{2m-1} \end{bmatrix}. \tag{4.3}$$

The eigenvalue algorithm presented in [7] relies on the following result:

Theorem 13. *[Thm. 3.4, [7]] Suppose that the polynomial $P(\lambda)$ has exactly m eigenvalues $\lambda_1, \ldots, \lambda_m$ in the interior of Γ, and that these eigenvalues are distinct, simple and non degenerated. If $\chi_n(\lambda_\ell) \neq 0$ for $1 \leq \ell \leq m$, then the eigenvalues of the pencil $H_1 - \lambda H_0$ are given by $\lambda_1, \ldots, \lambda_m$.*

So, in order to approximate $\lambda_1, \ldots, \lambda_m$, it suffices to compute by quadrature the first $2m$ moments of $f(\lambda)$ and then apply an eigensolver, such as the QZ method (see A.5), to the resulting Hankel pencil. Block versions of this approach have also been proposed; we will say more on this later.

We now wish to investigate the behavior of the above method when the hypothesis that the λ_i's are distinct is removed. In particular, we aim to generalize Theorem 13, which is based on the Vandermonde factorization of H_0 and H_1.

Theorem 14. *Suppose that $P(\lambda)$ has exactly m eigenvalues in the interior of Γ, namely, distinct eigenvalues $\lambda_0, \ldots, \lambda_s$ with algebraic multiplicities m_0, ..., m_s, respectively, such that $m = m_0 + \cdots + m_s$. Moreover, assume that no eigenvalue of $P(\lambda)$ lies exactly on the contour Γ. If the geometric multiplicity of the λ_i's, for $i = 0, \ldots, s$, is equal to one, then the matrix H_0 is nonsingular and eigenvalues of the pencil $H_1 - \lambda H_0$ are given by $\lambda_0, \ldots, \lambda_s$ with algebraic multiplicities $m_0, \ldots, m_s$.*

Proof. Suppose that, in the Smith form (1.7) of $P(\lambda)$, the matrix $D(\lambda)$ is in the form $D(\lambda) = \mathrm{diag}(d_1(\lambda), \ldots, d_{n-1}(\lambda), d_n(\lambda))$, where

$$d_n(\lambda) = (\lambda - \lambda_0)^{m_0}(\lambda - \lambda_1)^{m_1} \cdots (\lambda - \lambda_s)^{m_s} \prod_{i=s+1}^{r} (\lambda - \lambda_i)^{m_i},$$

and $\lambda_{s+1}, \ldots, \lambda_r$ are the eigenvalues of $P(\lambda)$ located outside the contour Γ, with algebraic multiplicities $m_{s+1}, \ldots, m_r$. Moreover, define $\tilde{d}_n(\lambda) = \prod_{i=0}^{s}(\lambda - \lambda_i)^{m_i}$, i.e., $\tilde{d}_n(\lambda)$ is the factor of $d_n(\lambda)$ whose roots are the eigenvalues of $P(\lambda)$ located inside Γ.

Since the geometric multiplicities of the λ_i's are all equal to one, the factors $(\lambda - \lambda_i)$, for $i = 0, \ldots, s$, do not appear in the monic scalar polynomials $d_0(\lambda), \ldots, d_{n-1}(\lambda)$.

Applying Theorem 12, we have:

$$\mu_k = \frac{1}{2\pi\imath}\oint_\Gamma z^k f(z)dz = \frac{1}{2\pi\imath}\oint_\Gamma \sum_{j=1}^{n}\frac{\chi_j(z)}{d_j(z)}z^k dz = \\ = \frac{1}{2\pi\imath}\oint_\Gamma \frac{\varphi(z)}{d_n(z)}z^k dz,$$

where

$$\varphi(z) = \sum_{j=1}^{n}\chi_j(z)h_j(z), \text{ with } h_j(z) = \frac{d_n(z)}{d_j(z)}.$$

We can introduce partial fraction decompositions and write

$$\mu_k = \frac{1}{2\pi\imath}\oint_\Gamma \frac{\varphi(z)}{d_n(z)}z^k dz = \\ = \frac{1}{2\pi\imath}\oint_\Gamma \left(\sum_{i=1}^{m_0}\frac{c_{0,i}z^k}{(z-\lambda_0)^i} + \cdots + \sum_{i=1}^{m_s}\frac{c_{s,i}z^k}{(z-\lambda_s)^i}\right)dz = \\ = \sum_{j=0}^{s}\sum_{i=1}^{m_j}\frac{1}{2\pi\imath}\oint_\Gamma \frac{c_{j,i}z^k}{(z-\lambda_j)^i}dz,$$

where $c_{j,i} \in \mathbb{C}$, for $j = 0, \ldots, s$ and $i = 1, \ldots, m_j$. Classical results on residues then yield

$$\mu_k = \sum_{j=0}^{s}\sum_{i=1}^{m_j} c_{j,i}\text{Res}\left(\frac{z^k}{(z-\lambda_j)^i}, \lambda_j\right) = \\ = \sum_{j=0}^{s}\sum_{i=1}^{m_j} c_{j,i}\frac{1}{(i-1)!}\lim_{z\to\lambda_j}\frac{d^{i-1}}{dz^{i-1}}\left((z-\lambda_j)^i\frac{z^k}{(z-\lambda_j)^i}\right) = \\ = \sum_{j=0}^{s}\sum_{i=1}^{m_j} c_{j,i}\frac{1}{(i-1)!}\lim_{z\to\lambda_j}\frac{d^{i-1}}{dz^{i-1}}\left(z^k\right) = \\ = \sum_{j=0}^{s}\sum_{i=1}^{m_j}\nu_{j,i}\lambda_j^{k-i+1}, \tag{4.4}$$

where

$$\nu_{j,i} = \begin{cases} \frac{c_{j,i}}{(i-1)!}(k-i+2)(k-i+3)\cdots k & \text{if} \quad k \geq i-1, \\ 0 \quad \text{otherwise.} \end{cases}$$

Now, consider the pencil $H_1 - \lambda H_0$ with H_0 and H_1 defined as in (4.3). Because of (4.4), and of the fact that $\lambda_0, \ldots, \lambda_s$ are roots of $\tilde{d}_n(\lambda)$, the moments μ_k satisfy a linear recurrence equation of the form:

$$\mu_k = a_{m-1}\mu_{k-1} + a_{m-2}\mu_{k-2} + \cdots + a_0\mu_{k-m}. \tag{4.5}$$

Moreover, $\tilde{d}_n(\lambda)$ is the polynomial of smallest degree that has roots $\lambda_0, \ldots, \lambda_s$ with the prescribed multiplicities $m_0, \ldots, m_s$, so the recurrence (4.5) has the shortest possible length; also, note that the a_i's in (4.5) are actually the coefficients of $\tilde{d}_n(\lambda)$. Therefore, the matrices H_0 and H_1 have full rank. The same argument shows that H_0 and H_1 are rank-deficient if taken of size larger than $m \times m$ (see [103], [[48], Vol. 2, pp. 205)] and [25]).

As a consequence of the shifted Hankel form of H_0 and H_1, we have $H_0C = H_1$, where C is a matrix in companion form

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & x_0 \\ 1 & 0 & \cdots & 0 & x_1 \\ 0 & 1 & \cdots & 0 & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & x_{m-1} \end{bmatrix},$$

and its last column is given by the solution of the linear system

$$H_0 \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-1} \end{bmatrix} = \begin{bmatrix} \mu_m \\ \mu_{m+1} \\ \vdots \\ \mu_{2m-1} \end{bmatrix}. \tag{4.6}$$

The polynomial of degree m:

$$p(\lambda) = \lambda^m - x_{m-1}\lambda^{m-1} - \cdots - x_0$$

is a scalar multiple of $\tilde{d}_n(\lambda)$, and its roots are the λ_i's generating the entries of the pencil $H_1 - \lambda H_0$. So we also have that the μ_i's satisfy the recurrence (4.5):

$$\mu_k = x_{m-1}\mu_{k-1} + x_{m-2}\mu_{k-2} + \cdots + x_0\mu_{k-m},$$

where $k = m, m+1, \ldots, 2m$.

Consider now the Jordan matrix

$$J = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_s \end{bmatrix},$$

where each block J_i, of dimension m_i, is a square matrix of the form

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix},$$

and define the confluent Vandermonde matrix

$$V = \begin{pmatrix} \mathbf{v} & J^T\mathbf{v} & \cdots & (J^T)^{r-1}\mathbf{v} \end{pmatrix},$$

where $\mathbf{v}^T = \begin{pmatrix} e_1^{[m_1]T} & \cdots & e_1^{[m_s]T} \end{pmatrix}$ is partitioned conformally with J and $e_1^{[m_\ell]T} = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^T$ is the m_ℓ-dimensional unit coordinate vector. Then we have

$$\begin{aligned} VC &= \begin{pmatrix} \mathbf{v} & J^T\mathbf{v} & \cdots & (J^T)^{r-1}\mathbf{v} \end{pmatrix} C \\ &= \begin{pmatrix} J^T\mathbf{v} & \cdots & (J^T)^{r-1}\mathbf{v} & -(x_0 I + x_1 J + \cdots + x_{m-1}J^{r-1})^T\mathbf{v} \end{pmatrix} \\ &= \begin{pmatrix} J^T\mathbf{v} & \cdots & (J^T)^{r-1}\mathbf{v} & (J^T)^r\mathbf{v} \end{pmatrix} \\ &= J^T V, \end{aligned}$$

where we have used the property $p(J) = 0$, that is, the Cayley-Hamilton theorem.

We can now introduce the Vandermonde decomposition of the Hankel matrices H_0 and H_1. From the results presented in [25] it follows that there exist block matrices $B_0 = \operatorname{diag}(D_1^{(0)}, \ldots, D_s^{(0)})$ and $B_1 = \operatorname{diag}(D_1^{(1)}, \ldots, D_s^{(1)})$, partitioned conformally with J, satisfying the conditions $B_0 J^T = J B_0$ and $B_1 J^T = J B_1$, so that

$$H_i = V^T B_i V, \quad \text{for } i = 0, 1.$$

Moreover, we can prove that $JB_0 = B_1$:

$$\begin{aligned} H_0 C = H_1 \Leftrightarrow (V^T B_0 V)C = V^T B_1 V \Leftrightarrow V^T B_0 J^T V = V^T B_1 V \\ \Leftrightarrow V^T J B_0 V = V^T B_1 V \Leftrightarrow J B_0 = B_1, \end{aligned}$$

where we used the properties $VC = J^T V$ and $B_i J^T = J B_i$.
Therefore, we have:

$$\begin{aligned} H_1 - \lambda H_0 &= V^T B_1 V - \lambda V^T B_0 V = V^T J B_0 V - \lambda V^T B_0 V \\ &= V^T (J - \lambda I) B_0 V. \end{aligned}$$

So the eigenvalues of $H_1 - \lambda H_0$ are $\lambda_0, \ldots, \lambda_s$ with respective multiplicities $m_0, \ldots, m_s$. $\square$

Example 14. *Consider the matrix polynomial:*

$$P(\lambda) = \lambda^2 I + \lambda \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -6 & 0 \\ 0 & 0 & 0 & -5 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}.$$

$P(\lambda)$ has eigenvalues: $\lambda_1 = \frac{1}{2}, \lambda_2 = 1, \lambda_3 = 2, \lambda_4 = 3$ with algebraic multiplicities: $m_1 = 2, m_2 = 2, m_3 = 1, m_4 = 3$. The associated Smith form is:

$$D(\lambda) = \operatorname{diag}\Big(\Big(1, 1, 1, (\lambda - \frac{1}{2})^2(\lambda - 1)^2(\lambda - 2)(\lambda - 3)^3\Big).$$

Choosing vectors $u = \begin{bmatrix} 2 & -2 & 1 & -1 \end{bmatrix}^T$ and $v = \begin{bmatrix} 0 & 1 & 0 & 2 \end{bmatrix}^T$, we find that:

$$H_0 = \begin{bmatrix} -3 & -7 & -9 & \frac{-21}{2} \\ -7 & -9 & \frac{-21}{2} & -12 \\ -9 & \frac{-21}{2} & -12 & \frac{-109}{8} \\ \frac{-21}{2} & -12 & \frac{-109}{8} & \frac{-123}{8} \end{bmatrix}, \qquad H_1 = \begin{bmatrix} -7 & -9 & \frac{-21}{2} & -12 \\ -9 & \frac{-21}{2} & -12 & \frac{-109}{8} \\ \frac{-21}{2} & -12 & \frac{-109}{8} & \frac{-123}{8} \\ -12 & \frac{-109}{8} & \frac{-123}{8} & \frac{-551}{32} \end{bmatrix}.$$

Then, we have:

$$C = H_0^{-1} H_1 = \begin{bmatrix} 0 & 0 & 0 & \frac{-1}{4} \\ 1 & 0 & 0 & \frac{3}{2} \\ 0 & 1 & 0 & \frac{-13}{4} \\ 0 & 0 & 1 & 3 \end{bmatrix}.$$

Note that the eigenvalues of C are $\frac{1}{2}, \frac{1}{2}, 1, 1$. Moreover, the companion matrix C is associated with the monic polynomial:

$$p(\lambda) = \lambda^4 - 3\lambda^3 + \frac{13}{4}\lambda^2 - \frac{3}{2}\lambda + \frac{1}{4},$$

whose roots are indeed $\frac{1}{2}, \frac{1}{2}, 1, 1$.

In fact, the Vandermonde factorization for H_0 and H_1 is $H_i = V^T B_i V$, $i = 0, 1$, where

$$V = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ 0 & 1 & 1 & \frac{3}{4} \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0 & -2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & -3 & -2 \\ 0 & 0 & -2 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} -2 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -5 & -2 \\ 0 & 0 & -2 & 0 \end{bmatrix},$$

and $JB_0 = B_1$, *with*

$$J = \begin{bmatrix} \frac{1}{2} & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Example 15. *Consider the quadratic matrix polynomial:*

$$P(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & -2 \end{bmatrix} + \lambda \begin{bmatrix} 0 & 0 & 0 \\ -4 & -2 & 0 \\ 2 & -2 & 4 \end{bmatrix} + \begin{bmatrix} -2 & 1 & -2 \\ 2 & 1 & 0 \\ -1 & 1 & -2 \end{bmatrix},$$

with associated Smith form:

$$D(\lambda) = \operatorname{diag}\left((d_1(\lambda), d_2(\lambda), d_3(\lambda)\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & (\lambda-1)^2 & 0 \\ 0 & 0 & (\lambda-1)^3(\lambda+1) \end{bmatrix}.$$

The Jordan matrix associated with the linearized form of $P(\lambda)$ *is:*

$$J = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix},$$

where each block J_i *is:*

$$J_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad J_2 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that we have the eigenvalue $\lambda = 1$ *in different Jordan blocks.*
Choose Γ *as the circle* $\varphi(t) = 1 + \frac{1}{10}e^{it}$, *which contains 5 eigenvalues* $\lambda = 1$, *and consider vectors* $u = \begin{bmatrix} 3 & 1 & -2 \end{bmatrix}^T$ *and* $v = \begin{bmatrix} 3 & -1 & -2 \end{bmatrix}^T$. *Theorem 14 implies that the moment method yields the eigenvalues of* $P(\lambda)$ *inside the contour, which are roots of* $d_3(\lambda)$, *i.e.,* $\lambda_0 = 1$ *with multiplicity* $m_0 = 3$. *Let us compute the matrices* H_0 *and* H_1 *of size* 3×3*:*

$$H_0 = \begin{bmatrix} 7 & 3 & 3 \\ 3 & 3 & 7 \\ 3 & 7 & 15 \end{bmatrix}, \qquad H_1 = \begin{bmatrix} 3 & 3 & 7 \\ 3 & 7 & 15 \\ 7 & 15 & 27 \end{bmatrix}.$$

The matrix H_0 is nonsingular. Then, we have:

$$C = H_0^{-1}H_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -3 \\ 0 & 1 & 3 \end{bmatrix}.$$

Note that $eig(C) = 1, 1, 1$.

As the contour contains 5 eigenvalues, we might ask what happens when taking the Hankel matrix $\hat{H}_0$ of size 5×5:

$$\hat{H}_0 = \begin{bmatrix} 7 & 3 & 3 & 7 & 15 \\ 3 & 3 & 7 & 15 & 27 \\ 3 & 7 & 15 & 27 & 43 \\ 7 & 15 & 27 & 43 & 63 \\ 15 & 27 & 43 & 63 & 63 \end{bmatrix}.$$

This matrix is singular, therefore we will not be able to find all the 5 eigenvalues inside the contour. The method miss the additional multiplicities associated with the polynomial $d_2(\lambda)$.

Remark 10. *We conclude that the scalar moment method can be used to compute the (possibly multiple) eigenvalues of $P(\lambda)$ that belong to the interior of Γ and that are roots of the polynomial $d_n(\lambda)$. The method misses the additional multiplicities associated with the polynomials $d_1(\lambda), \ldots, d_{n-1}(\lambda)$.*

The above remark is consistent with the fact that the Jordan form of a companion matrix only contains one Jordan block for each eigenvalue: it is not possible to capture multiple eigenvalues associated with several Jordan blocks.

In order to "see" the additional eigenvalues that are roots of $d_1(\lambda), \ldots, d_{n-1}(\lambda)$, the block version of the moment method may be useful (see Section 4.3.2).

4.3.1 Computing Invariant Pairs via Moment Pencils

Let Γ be a closed contour, $\lambda_1, \ldots, \lambda_m$ all the eigenvalues of $P(\lambda)$ in the interior of Γ and the matrices H_0 and H_1 defined as in (4.3).
For $k = 0, 1, \ldots, m-1$ and a nonzero vector $v \in \mathbb{C}^n$, consider the vectors:

$$s_k = \frac{1}{2\pi\imath} \oint_\Gamma z^k P(z)^{-1} v dz, \tag{4.7}$$

The method proposed in [7] for the computation of the eigenvectors of $P(\lambda)$ is based on the following result.

Theorem 15. *[Thm. 3.5, [7]] Let (λ_i, w_i), $i = 1, \ldots, m$ be eigenpairs for the matrix pencil $H_1 - \lambda H_0$, where the simple, distinct, nondegenerate eigenvalues λ_i belong to the interior of a given closed contour Γ. Let $S = [s_0, \cdots, s_{m-1}]$. Then, for $i = 1, \ldots, m$, the vector $y_i = Sw_i$ is an eigenvector of $P(\lambda)$ corresponding to the eigenvalue λ_i.*

Theorem 15 is readily applied to invariant pairs.

Corollary 2. *With the hypotheses of Theorem 15, $S = \begin{bmatrix} s_0, & s_1, & \ldots, & s_{m-1} \end{bmatrix}$ and $C = H_0^{-1}H_1$. Then the pair (S, C) satisfies $P(S, C) = 0$, i.e., (S, C) is a simple invariant pair for $P(\lambda)$.*

Proof. Note that the pair (Y, Λ), where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$ and $Y = [y_1, \ldots, y_m]$, is clearly an invariant pair for $P(\lambda)$, that is,

$$P(Y, \Lambda) = \sum_{j=0}^{\ell} A_j Y \Lambda^j = 0.$$

Moreover, we know that $C = H_0^{-1}H_1 = V^{-1}\Lambda V$, where V is the classical Vandermonde matrix associated with $\lambda_1, \ldots, \lambda_m$, and that the columns of V^{-1} are eigenvectors of $H_1 - \lambda H_0$. So we have

$$\begin{aligned} 0 = \sum_{j=0}^{\ell} A_j Y \Lambda^j = \sum_{j=0}^{\ell} A_j Y \Lambda^j V = \\ = \sum_{j=0}^{\ell} A_j Y V V^{-1} \Lambda^j V = \sum_{j=0}^{\ell} A_j S C^j = P(S, C), \end{aligned}$$

that is, (S, C) is also an invariant pair of $P(\lambda)$. □

What can we say about more general cases, where some of the hypotheses of Theorem 15 are removed? If we remove the hypothesis that the λ_i's are distinct, we can prove the following.

Theorem 16. *With the hypotheses of Theorem 14, let $S = \begin{bmatrix} s_0, & s_1, & \ldots, & s_{m-1} \end{bmatrix}$ and $C = H_0^{-1}H_1$. Then the pair (S, C) satisfies $P(S, C) = 0$, i.e., (S, C) is a simple invariant pair for $P(\lambda)$.*

Proof. Consider again the columns $q_1(\lambda), \ldots, q_n(\lambda)$ of the matrix $F(\lambda)$ in the Smith form (1.7) and the definition of s_k given in (4.7). A similar computation to (4.4) shows that

$$S = [s_0, \ldots, s_{m-1}] = \mathcal{Q}V,$$

where

$$\mathcal{Q} = [\mathcal{Q}_0, \ldots, \mathcal{Q}_s],$$
$$\mathcal{Q}_j = [\gamma_{0,j} q_n(\lambda_j),\ \gamma_{1,j} q_n'(\lambda_j), \ldots, \gamma_{m_j-1,j} q_n^{(m_j-1)}(\lambda_j)],\ \text{for}\, j = 0, \ldots, s,$$

the $\gamma_{i,j}$'s are complex coefficients and V is the confluent Vandermonde matrix defined above.

It is shown in [7] (Lemma 2.4) that, if a complex number ζ is a root of $d_j(\lambda)$ for some index $1 \leq j \leq n$, then $P(\zeta)q_j(\zeta) = 0$. In our case, this implies that the vector $q_n(\lambda)$ is a root polynomial of $P(\lambda)$ corresponding to the eigenvalue λ_j, for each $j = 0, \ldots, s$; see [51], section 1.5, for the definition and properties of root polynomials. It follows that $[q_n(\lambda_j),\ q_n'(\lambda_j), \ldots, q_n^{(m_j-1)(\lambda_j)}]$ forms a Jordan chain for the eigenvalue λ_j. So we have that $(\mathcal{Q}, J)$ is an invariant pair for $P(\lambda)$. Moreover, if $C = H_0^{-1} H_1$ as usual, we have

$$0 = \sum_{j=0}^{\ell} A_j \mathcal{Q} J^j = \sum_{j=0}^{\ell} A_j \mathcal{Q} J^j V =$$
$$= \sum_{j=0}^{\ell} A_j \mathcal{Q} V V^{-1} J^j V = \sum_{j=0}^{\ell} A_j S C^j = P(S, C),$$

Therefore, (S, C) is a simple invariant pair for $P(\lambda)$. □

Example 16. *Consider the matrix polynomial:*

$$P(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} -2 & 0 \\ 2 & -1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

which has eigenvalues $\lambda_1 = 0$, with algebraic multiplicity 1 and, $\lambda_2 = 1$, with algebraic multiplicity 3.

Suppose we are interested in the eigenvalues λ_2. Then, we can choose a contour $\Gamma(t) = z_0 + Re^{it}$, $t \in [0, 2\pi]$, where $z_0 = 1$ and $R = \frac{1}{2}$.

Choosing the vectors $u = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$ and $v = \begin{bmatrix} -1 & 1 \end{bmatrix}^T$, we find:

$$H_0 = \begin{bmatrix} -1 & -2 & -5 \\ -2 & -5 & -10 \\ -5 & -10 & -17 \end{bmatrix}, \qquad H_1 = \begin{bmatrix} -2 & -5 & -10 \\ -5 & -10 & -17 \\ -10 & -17 & -26 \end{bmatrix}.$$

Then, we have that the pair (S, C) given by Theorem 16

$$S = H_0^{-1}H_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -3 \\ 0 & 1 & 3 \end{bmatrix} \quad \textit{and} \quad C = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 1 & 3 \end{bmatrix}$$

is an invariant pair, i.e., it satisfies $P(S, C) = 0$.

Note that the companion matrix C is associated with the monic polynomial:

$$p(\lambda) = \lambda^3 - 3\lambda^2 + 3\lambda - 1,$$

which has a triple root equal to 1.

4.3.2 The Block Moment Method

Instead of the scalar version of the moment method, we can consider a Hankel pencil constructed by block moments $M_k \in \mathbb{C}^{\xi\times\xi}$, for a suitable positive integer ξ.

Definition 16. *Let k be a positive integer and $U, V \in \mathbb{C}^{n\times\xi}$ nonzero matrices with linearly independent columns. For $k = 0, 1, \ldots$, define the block moment $M_k \in \mathbb{C}^{\xi\times\xi}$ as:*

$$M_k = \frac{1}{2\pi\imath}\oint_\Gamma z^k U^H P(z)^{-1} V dz.$$

Then the block Hankel matrices $H_{\xi 0}, H_{\xi 1} \in \mathbb{C}^{\tilde{m}\xi\times\tilde{m}\xi}$ are defined as:

$$H_{\xi 0} = \begin{bmatrix} M_0 & M_1 & \cdots & M_{\tilde{m}-1} \\ M_1 & M_2 & \cdots & M_{\tilde{m}} \\ \vdots & \vdots & & \vdots \\ M_{\tilde{m}-1} & M_{\tilde{m}} & \cdots & M_{2\tilde{m}-2} \end{bmatrix}, \quad H_{\xi 1} = \begin{bmatrix} M_1 & M_2 & \cdots & M_{\tilde{m}} \\ M_2 & M_3 & \cdots & M_{\tilde{m}+1} \\ \vdots & \vdots & & \vdots \\ M_{\tilde{m}} & M_{\tilde{m}+1} & \cdots & M_{2\tilde{m}-1} \end{bmatrix}$$

Polynomial eigenvalue computation via the eigenvalues of the pencil $H_{\xi 1} - \lambda H_{\xi 0}$ is discussed in [7] and [18]. See also [85] for an application to acoustic nonlinear eigenvalue problems.

Invariant pairs can be computed from block moments by applying an approach that is similar to the one described in the previous section for the scalar version. For $k = 0, 1, \ldots, \tilde{m} - 1$, consider the matrices $S_k \in \mathbb{C}^{n\times\xi}$ defined as:

$$S_k = \frac{1}{2\pi\imath}\oint_\Gamma z^k P(z)^{-1} V dz.$$

Then, we have the following result.

Proposition 5. *Let Γ be a closed contour, let the block Hankel matrix $H_{\xi 0} \in \mathbb{C}^{\tilde{m}\xi \times \tilde{m}\xi}$ be nonsingular and m be the number of eigenvalues inside of Γ. If $\tilde{m}\xi = m$ and $Y = [S_0, \ldots, S_{\tilde{m}-1}]$, $T = H_{\xi 0}^{-1} H_{\xi 1}$, then the pair (Y, T) satisfies $P(Y, T) = 0$, i.e., (Y, T) is a simple invariant pair for $P(\lambda)$.*

Example 17. *Consider again the matrix polynomial of Example 15:*

$$P(\lambda) = \lambda^2 \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & -2 \end{bmatrix} + \lambda \begin{bmatrix} 0 & 0 & 0 \\ -4 & -2 & 0 \\ 2 & -2 & 4 \end{bmatrix} + \begin{bmatrix} -2 & 1 & -2 \\ 2 & 1 & 0 \\ -1 & 1 & -2 \end{bmatrix}.$$

with associated Smith form:

$$D(\lambda) = \operatorname{diag}\left((d_1(\lambda), d_2(\lambda), d_3(\lambda)\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & (\lambda-1)^2 & 0 \\ 0 & 0 & (\lambda-1)^3(\lambda+1) \end{bmatrix}.$$

In Example 15, we found that the scalar moment method, i.e. when $\xi = 1$, missed the additional multiplicities associated with the polynomial $d_2(\lambda)$.
Consider now $\xi = 2$ as the size of the block moments M_k, the contour $\varphi(t) = 1 + \frac{1}{10}e^{\imath t}$, containing 5 eigenvalues $\lambda = 1$, as before, and the matrices:

$$U = \begin{bmatrix} 1 & 0 \\ 5 & -3 \\ 2 & -4 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 3 \\ 0 & 1 \\ -2 & 4 \end{bmatrix}.$$

We find the block moments M_k:

$$M_0 = \begin{bmatrix} -9 & -12 \\ 9 & 12 \end{bmatrix}, \qquad M_1 = \begin{bmatrix} -1 & -22 \\ -1 & 27 \end{bmatrix}, \quad M_2 = \begin{bmatrix} -5 & -8 \\ 1 & 18 \end{bmatrix},$$
$$M_3 = \begin{bmatrix} -21 & 30 \\ 15 & -15 \end{bmatrix}, \qquad M_4 = \begin{bmatrix} -49 & 92 \\ 41 & -72 \end{bmatrix}, \quad M_5 = \begin{bmatrix} -89 & 178 \\ 79 & -153 \end{bmatrix}.$$

Then, we have the Hankel matrix H_{L0}:

$$H_{\xi 0} = \begin{bmatrix} M_0 & M_1 & M_2 \\ M_1 & M_2 & M_3 \\ M_2 & M_3 & M_4 \end{bmatrix} = \begin{bmatrix} -9 & -12 & -1 & -22 & -5 & -8 \\ 9 & 12 & -1 & 27 & 1 & 18 \\ -1 & -22 & -5 & -8 & -21 & 30 \\ -1 & 27 & 1 & 18 & 15 & -15 \\ -5 & -8 & -21 & 30 & -49 & 92 \\ 1 & 18 & 15 & -15 & 41 & -72 \end{bmatrix}.$$

The matrix $H_{\xi 0}$ is singular. This happens because there are just 5 eigenvalues inside the contour and $H_{\xi 0}$ has size 6×6. Then, we have to reduce the matrices $H_{\xi 0}$ and $H_{\xi 1}$ to match the number of eigenvalues in the contour. Therefore, we get the truncated matrices:

$$\hat{H}_{\xi 0} = \begin{bmatrix} -9 & -12 & -1 & -22 & -5 \\ 9 & 12 & -1 & 27 & 1 \\ -1 & -22 & -5 & -8 & -21 \\ -1 & 27 & 1 & 18 & 15 \\ -5 & -8 & -21 & 30 & -49 \end{bmatrix}, \; \hat{H}_{\xi 1} = \begin{bmatrix} -1 & -22 & -5 & -8 & -21 \\ -1 & 27 & 1 & 18 & 15 \\ -5 & -8 & -21 & 30 & -49 \\ 1 & 18 & 15 & -15 & 41 \\ -21 & 30 & -49 & 92 & -89 \end{bmatrix}.$$

Then, we obtain:

$$T = \hat{H}_{\xi 0}^{-1}\hat{H}_{\xi 1} = \begin{bmatrix} 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 4 & -3 \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & -2 & 3 \end{bmatrix}.$$

The eigenvalues of the matrix T are $1, 1, 1, 1, 1$, which are all the eigenvalues inside the contour.
Moreover, computing the matrix $Y = [S_0, S_1, S_2]$, using we get:

$$\hat{Y} = \begin{bmatrix} 0 & 1 & 1 & 2 & 0 \\ 0 & -2 & -2 & 0 & 0 \\ 0 & -\frac{3}{2} & -\frac{7}{2} & -3 & -4 \end{bmatrix}.$$

Then, $(\hat{Y}, T)$ is an invariant pair for $P(\lambda)$.

Experimentally, we noted that the block method allows us to better "capture" the multiplicity structure of eigenvalues, when there are several Jordan blocks per eigenvalue. Further investigation of this approach will be the topic of future work. It should be pointed out that the results in [18], and particularly Theorem 3.3, provide useful insight into a generalized block moment method and into the (good) behavior of the method in presence of multiple eigenvalues.

With the condition that the size of the block Hankel matrix $H_{\xi 0} \in \mathbb{C}^{\tilde{m}\xi \times \tilde{m}\xi}$ is equal to the number of eigenvalues inside of Γ, i.e., if $\tilde{m}\xi = m$, we get:

$$T = H_{\xi 0}^{-1}H_{\xi 1} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -X_0 \\ I & 0 & \cdots & 0 & -X_1 \\ 0 & I & \cdots & 0 & -X_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & -X_{m-1} \end{bmatrix},$$

where

$$\begin{bmatrix} -X_0 \\ -X_1 \\ \vdots \\ -X_{m-1} \end{bmatrix} = H_{\xi 0}^{-1} \begin{bmatrix} M_m \\ M_{m+1} \\ \vdots \\ M_{2m-1} \end{bmatrix}.$$

Consequently, since T has a block companion form, the problem of finding the eigenvalues $\lambda_1, \ldots, \lambda_m$ of is equivalent to the problem of finding the eigenvalues of the matrix polynomial:

$$L(\lambda) := \lambda^\ell + X_{\ell-1}\lambda^{\ell-1} + \cdots + X_1\lambda + X_0 = 0.$$

Example 18. *Consider the matrix polynomial:*

$$P(\lambda) = \lambda^4 I + \lambda^3 \begin{bmatrix} -12 & 0 & 0 \\ 0 & -12 & 0 \\ 0 & 0 & -12 \end{bmatrix} + \lambda^2 \begin{bmatrix} 41 & 0 & 0 \\ 0 & 41 & 0 \\ 0 & 0 & 41 \end{bmatrix} + \lambda \begin{bmatrix} -30 & 0 & 0 \\ 0 & -30 & 0 \\ 0 & 0 & -30 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

which has eigenvalues: 0, 0, 0, 1, 1, 1, 5, 5, 5, 6, 6, 6.
Suppose that in this example we only wish to compute the eigenvalues 0 and 1. We choose a contour Γ enclosing just those eigenvalues; for example, we can take the circle $\Gamma(t) = z_0 + Re^{it}$, where $z_0 = \frac{1}{2}$, $R = 1$. For $L = 3$, we find

$$T = H_{L0}^{-1} H_{L1} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{30} & \frac{1}{30} & \frac{1}{30} \\ 0 & 0 & 0 & \frac{-1}{30} & \frac{-1}{30} & \frac{-1}{30} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \frac{61}{60} & \frac{1}{60} & \frac{1}{60} \\ 0 & 1 & 0 & \frac{-1}{60} & \frac{59}{60} & \frac{-1}{60} \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The companion matrix T is associated with the monic quadratic matrix polynomial:

$$H(\lambda) = \lambda^2 I + \lambda X_1 + X_0,$$

where

$$X_0 = -\begin{bmatrix} \frac{1}{30} & \frac{1}{30} & \frac{1}{30} \\ \frac{-1}{30} & \frac{-1}{30} & \frac{-1}{30} \\ 0 & 0 & 0 \end{bmatrix}, \quad X_1 = -\begin{bmatrix} \frac{61}{60} & \frac{1}{60} & \frac{1}{60} \\ \frac{-1}{60} & \frac{59}{60} & \frac{-1}{60} \\ 0 & 0 & 1 \end{bmatrix},$$

which has as eigenvalues: 0,0,0,1,1,1 as sought.

In [13], it is shown that the conditioning of the generalized Hankel eigenvalue problem $(H_1 - \lambda H_0)y = 0$ grows exponentially in terms of a quantity that depends on the largest distance between eigenvalues (for conditioning on Hankel matrices see, e.g., [11], [12]).

Therefore, our computation of invariant pairs and solvents using the (block) moment method can be affected by ill conditioning. One possible solution to this problem consists in breaking the contour Γ, which contains all the eigenvalues, into smaller contours Γ_i. The small contours should be chosen so that each of them contains clustered eigenvalues. This improves the conditioning for each subproblem, and therefore for the whole problem. This strategy is consistent with the fact that invariant pairs typically present a particular interest for clustered groups of eigenvalues.
For instance, consider Figure 4.1 and suppose that the largest distance between eigenvalues λ_i is large. Then, knowing the locations of the eigenvalues, we can group them into smaller non-overlapping contours Γ_i as shown in Figure 4.2.

After computing the simple invariant pairs (X_i, S_i) associated with each contour, using the (block) moment method, we can construct a simple invariant pair (X, S) as follows:

$$X = \begin{bmatrix} X_1 & X_2 & \cdots \end{bmatrix}, \qquad S = \operatorname{diag}(S_1, S_2, \cdots).$$

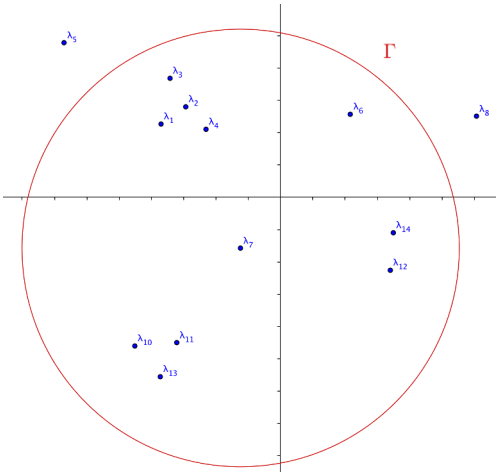

Figure 4.1: Eigenvalues inside a circle

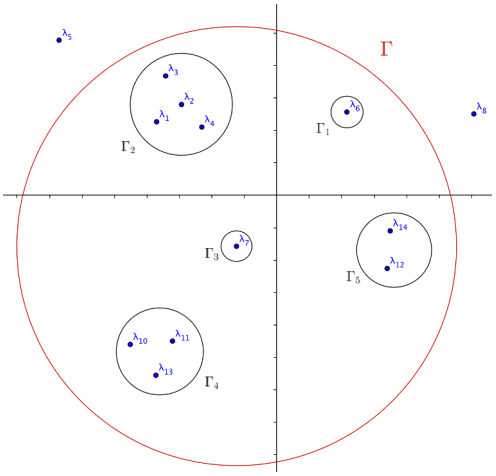

Figure 4.2: Grouping clustered eigenvalues

4.4 Choosing the Contour

The choice of Γ is, of course, a crucial step when applying the contour integral formulation of the eigenvalue or invariant pair problem. If some information about the localization of the eigenvalues of $P(\lambda)$ is available, one can choose the contour accordingly. Some works have addressed this localization problem; for instance: in [23] and [102] are presented generalizations of matrix version of Pellet's theorem to the case of matrix polynomials. In [70], N. Higham and F. Tisseur found upper and lower bounds to the moduli of the eigenvalues of a matrix polynomial. On the other hand, in [107], the authors used tropical roots as approximations to eigenvalues of matrix polynomials.

A related question is: how many eigenvalues of $P(\lambda)$ live inside a given contour? Even an approximate estimate can be useful to choose Γ and k consistently. An answer to this problem is provided in [46] and [47] and it is based on the following result.

Theorem 17. *[Thm. 2, [46]] Let $F(z)$ be an $n \times n$ regular analytic matrix function and let $tr(F(z))$ be the matrix trace of $F(z)$. In addition, let m be the number of eigenvalues, counting multiplicity, inside closed curves Γ on the complex plane for the nonlinear eigenvalue problem: $F(\lambda)\boldsymbol{x} = 0$. Then, we have:*

$$m = \oint_{\Gamma} tr\left(F(z)^{-1}\frac{dF(z)}{dz}\right)dz, \tag{4.8}$$

where $det(F(z)) \neq 0$.

The equation (4.8) can be approximated by an N-point quadrature rule by:

$$m \approx \sum_{j=0}^{N-1} \omega_j \mathrm{tr}(F(z_j)^{-1}F'(z_j)), \tag{4.9}$$

where z_j is a quadrature point and ω_j is a weight.

In the case when we use the trapezoidal rule on a circle with center C and radius R, quadrature points and weights are defined by:

$$\omega_j = \frac{R}{N}\exp\left(\frac{2\pi i}{N}\left(j+\frac{1}{2}\right)\right), \qquad z_j = C + R\exp\left(\frac{2\pi i}{N}\left(j+\frac{1}{2}\right)\right).$$

To avoid the matrix inversion in (4.9), an estimate for the trace with an unbiased estimation can be used:

$$\mathrm{tr}(F(z_j)^{-1}F'(z_j)) \approx \frac{1}{L}\sum_{i=1}^{L}\left(\mathbf{v}_i^T F(z_j)^{-1}F'(z_j)\mathbf{v}_i\right),$$

where $\mathbf{v}_i$ are the sample vectors, with entries equal to 1 or -1 with equal probability and L is the number of sample vectors.

The number m of eigenvalues inside the contour can then be estimated as:

$$m \approx \frac{1}{L}\sum_{j=0}^{N-1}\omega_j\sum_{i=1}^{L}\left(\mathbf{v}_i^T F(z_j)^{-1}F'(z_j)\mathbf{v}_i\right). \tag{4.10}$$

Remark 11. *The choice of Γ can be combined with shifting techniques for the eigenvalues of $P(\lambda)$: see for instance [101].*

4.5 Numerical Approximation: Trapezoid Rule for Moments

Consider the equation (4.2). Assuming that Γ has a 2π-periodic smooth parametrization:

$$\varphi \in C^1(\mathbb{R}, \mathbb{C}), \qquad \varphi(t + 2\pi) = \varphi(t) \quad \forall t \in \mathbb{R}.$$

Then, for $k = 0, \ldots, 2m-1$, we have:

$$\mu_k = \frac{1}{2\pi\imath} \oint_\Gamma z^k f(z) dz = \frac{1}{2\pi\imath} \int_0^{2\pi} \varphi(t)^k f(\varphi(t))\varphi'(t) dt. \tag{4.11}$$

Taking equidistant nodes $t_j = \frac{2j\pi}{N}$, $j = 0, \ldots, N$, and using the trapezoid rule (A.6), we obtain the approximation:

$$\mu_k \approx \frac{1}{\imath N} \sum_{j=0}^{N-1} \varphi(t_j)^k f(\varphi(t_j))\varphi'(t_j). \tag{4.12}$$

Note that for the special case when $\varphi(t)$ is the circle: $\varphi(t) = C + Re^{\imath t}$, we obtain the formula for (4.12):

$$\begin{aligned} \mu_k &\approx \frac{R}{N} \sum_{j=0}^{N-1} \exp\left(\frac{2\pi\imath j}{N}\right) \varphi(t_j)^k f(\varphi(t_j)) = \\ &= \frac{R}{N} \sum_{j=0}^{N-1} \exp\left(\frac{2\pi\imath j}{N}\right) \left(C + R\exp\left(\frac{2\pi\imath j}{N}\right)\right)^k f\left(C + R\exp\left(\frac{2\pi\imath j}{N}\right)\right). \end{aligned} \tag{4.13}$$

4.5.1 Error Analysis of the Trapezoid Rule for Moments

In this section, we present a explicit formulation for the error of the trapezoid rule for the moments μ_k, in the case when Γ is a closed contour in the complex plane. We follow a similar approach as W.-J. Beyn in [18].

Theorem 18. *Let $P(\lambda)$ be an $n \times n$ matrix polynomial and $h \in H(A(a_-, a_+), \mathbb{C})$ be holomorphic on the annulus:*

$$A(a_-, a_+) = \left\{ z \in \mathbb{C} : \frac{R}{a_-} < |z| < Ra_+ \right\}, \quad a_\pm > 1,$$

for some $R > 0$.

Then, the approximation error for the trapezoidal quadrature for the moments μ_k:

$$E_N(f) = \frac{1}{2\pi\imath}\oint_{|z|=R} z^k f(z)dz - \frac{1}{N}\sum_{j=0}^{N-1} f(Re^{\frac{2\imath j\pi}{N}})(Re^{\frac{2\imath j\pi}{N}})^{k+1}$$

satisfies for all $1 < \rho_- < a_-$, $1 < \rho_+ < a_+$:

$$|E_N(f)| \leq (\rho_+ R)^{k+1}\frac{M_1}{\rho_+^N - 1} + \left(\frac{R}{\rho_-}\right)^{k+1}\frac{M_2}{\rho_-^N - 1},$$

for some M_1 and M_2 such that:

$$\left|u^H P(\rho_+ Re^{\imath t})^{-1}v\right| \leq M_1, \qquad \left|u^H P\left(\frac{R}{\rho_-}e^{\imath t}\right)^{-1}v\right| \leq M_2.$$

Proof. Consider equations (4.11) and let $z = Re^{\imath t}$. Then we have:

$$\mu_k = \frac{1}{2\pi\imath}\oint_\Gamma z^k f(z)dz = \frac{1}{2\pi}\int_0^{2\pi}(Re^{\imath t})^k f(Re^{\imath t})(Re^{\imath t})dt = \frac{1}{2\pi}\int_0^{2\pi} f(Re^{\imath t})(Re^{\imath t})^{k+1}dt. \tag{4.14}$$

Now, define the function $h(z)$ as:

$$h(z) = f(z)z^{k+1}.$$

Using the Laurent expansion of h, we have:

$$h(z) = \sum_{\ell=-\infty}^{\infty} h_\ell z^\ell, \tag{4.15}$$

with coefficients:

$$\begin{aligned} h_\ell &= \frac{1}{2\pi\imath}\int_{|z|=R} h(z)z^{-\ell-1}dz = \frac{1}{2\pi\imath}\int_0^{2\pi} h(Re^{\imath t})(Re^{\imath t})^{-\ell-1}(\imath Re^{\imath t})dt = \\ &= \frac{R^{-\ell}}{2\pi}\int_0^{2\pi} h(Re^{\imath t})e^{-\imath\ell t}dt. \end{aligned} \tag{4.16}$$

From (4.14) and (4.16), we have:

$$I := \mu_k = h_0. \tag{4.17}$$

For any positive integer N, we define the trapezoidal rule approximation to μ_k in (4.14) by:

$$\mu_k \approx \frac{1}{N}\sum_{j=0}^{N-1} f(z_j)z_j^{k+1} = \frac{1}{N}\sum_{j=0}^{N-1} h(z_j) := I_N, \tag{4.18}$$

where $z_j = Re^{\imath t_j}$ for equidistant nodes $t_j = \frac{2j\pi}{N}$, $j = 0, \ldots, N-1$.

From (4.18) and (4.15), we have:

$$I_N = \frac{1}{N}\sum_{j=0}^{N-1} h(z_j) = \frac{1}{N}\sum_{\ell=-\infty}^{\infty} h_\ell \sum_{j=0}^{N-1} z_j^\ell = \\ = \frac{1}{N}\sum_{\ell=-\infty}^{\infty} h_\ell \sum_{j=0}^{N-1} (Re^{\frac{2\imath\pi j}{N}})^\ell = \frac{1}{N}\sum_{\ell=-\infty}^{\infty} R^\ell h_\ell \sum_{j=0}^{N-1} e^{\frac{2\imath\pi j\ell}{N}}.$$

If ℓ is multiple of N, the numbers $e^{\frac{2\imath\pi j\ell}{N}}$ are all equal to 1, and then, the second sum in previous formula is equal to N. On the other hand, if j is not multiple of N, then that sum is equal to 0.

Then, we have:

$$I_N = \sum_{\ell=-\infty}^{\infty} R^{\ell N} h_{\ell N} \tag{4.19}$$

Therefore, using (4.17) and (4.19), we obtain:

$$I_N - I = \sum_{\ell=-\infty}^{\infty} R^{\ell N} h_{\ell N} - h_0 = \sum_{\ell=1}^{\infty} \left(R^{\ell N} h_{\ell N} + R^{-\ell N} h_{-\ell N} \right).$$

Now, to estimate the error, we need a bound on the coefficients h_{jN}. Using Cauchy's Theorem, we can shift the contour from $|z| = R$ to $|z| = \rho_+ R$ (i.e. taking $z = \rho_+ Re^{\imath t}$). Then, we obtain:

$$|h_\ell| = \left| \frac{1}{2\pi\imath} \int_{|z|=\rho_+ R} h(z) z^{-\ell-1} dz \right| = \left| \frac{(\rho_+ R)^{-\ell}}{2\pi} \int_0^{2\pi} h(\rho_+ Re^{\imath t}) e^{-\imath\ell t} dt \right| \\ \leq \frac{(\rho_+ R)^{-\ell}}{2\pi} \int_0^{2\pi} \left| h(\rho_+ Re^{\imath t}) \right| dt,$$

where:

$$\left| h(\rho_+ Re^{\imath t}) \right| = \left| f(\rho_+ Re^{\imath t})(\rho_+ Re^{\imath t})^{k+1} \right| \leq (\rho_+ R)^{k+1} \left| f(\rho_+ Re^{\imath t}) \right| = \\ = (\rho_+ R)^{k+1} \left| u^H P(\rho_+ Re^{\imath t})^{-1} v \right| \leq (\rho_+ R)^{k+1} M_1,$$

for some M_1 such that $\left| u^H P(\rho_+ Re^{\imath t})^{-1} v \right| \leq M_1$.

Then, we have:

$$|h_\ell| \leq \frac{(\rho_+ R)^{-\ell}}{2\pi} \int_0^{2\pi} \left| h(\rho_+ Re^{\imath t}) \right| dt \leq (\rho_+ R)^{k+1-\ell} M_1.$$

Similarly, using Cauchy's Therorem, we can shift the contour from $|z| = R$ to $|z| = \frac{R}{\rho_-}$

(i.e. taking $z = \frac{R}{\rho_-}e^{\imath t}$). Then we find that:

$$|h_{-\ell}| \leq \left(\frac{R}{\rho_-}\right)^{\ell} \frac{1}{2\pi}\int_0^{2\pi} \left|h\left(\frac{R}{\rho_-}e^{\imath t}\right)\right| dt \leq \left(\frac{R}{\rho_-}\right)^{k+1+\ell} M_2,$$

for some M_2 such that $\left|u^H P\left(\frac{R}{\rho_-}e^{\imath t}\right)^{-1} v\right| \leq M_2$.

Using this, we find that:

$$\begin{aligned}
|I_N - I| &= \left|\sum_{\ell=1}^{\infty}\left(R^{\ell N}h_{\ell N} + R^{-\ell N}h_{-\ell N}\right)\right| = \sum_{\ell=1}^{\infty}\left|R^{\ell N}h_{\ell N} + R^{-\ell N}h_{-\ell N}\right| \\
&\leq \sum_{\ell=1}^{\infty} R^{\ell N}\left|h_{\ell N}\right| + \sum_{\ell=1}^{\infty} R^{-\ell N}\left|h_{-\ell N}\right| \\
&\leq \sum_{\ell=1}^{\infty}\left(R^{\ell N}(\rho_+ R)^{k+1-\ell N}M_1\right) + \sum_{\ell=1}^{\infty}\left(R^{-\ell N}\left(\frac{R}{\rho_-}\right)^{k+1+\ell N} M_2\right) \\
&= (\rho_+ R)^{k+1}M_1\sum_{\ell=1}^{\infty}\rho_+^{-\ell N} + \left(\frac{R}{\rho_-}\right)^{k+1} M_2\sum_{\ell=1}^{\infty}\rho_-^{-\ell N} \\
&= (\rho_+ R)^{k+1}M_1\frac{\rho_+^{-N}}{1-\rho_+^{-N}} + \left(\frac{R}{\rho_-}\right)^{k+1} M_2\frac{\rho_-^{-N}}{1-\rho_-^{-N}} \\
&= (\rho_+ R)^{k+1}M_1\frac{1}{\rho_+^{N}-1} + \left(\frac{R}{\rho_-}\right)^{k+1} M_2\frac{1}{\rho_-^{N}-1}.
\end{aligned}$$

Then, the error for the trapezoid sum for the moment μ_k is bounded by:

$$|I_N - I| \leq (\rho_+ R)^{k+1}\frac{M_1}{\rho_+^{N}-1} + \left(\frac{R}{\rho_-}\right)^{k+1} \frac{M_2}{\rho_-^{N}-1}.$$

□

Example 19. *Consider the matrix polynomial* (2.12) *and the annulus:*

$$A(a_-, a_+) = \left\{z \in \mathbb{C} : \frac{R}{a_-} < |z| < a_+ R\right\}, \quad a_\pm > 1,$$

for some $R > 0$. *Suppose that* $1 < \rho_- < a_-$ *and* $1 < \rho_+ < a_+$.

Now, consider in our example the circle of center $C = 0$ *and radius* $R = \frac{31}{10} = 3.1$, *i.e.* $z = \frac{31}{10}e^{\imath\pi}$. *Let* $a_+ = \frac{40}{31} \approx 1.2903$, *then* $a_+ R = 4$; *and let* $a_- = 31$, *then* $\frac{R}{a_-} = \frac{1}{10}$. *Then, we obtain the annulus:*

$$A = \left\{z \in \mathbb{C} : \frac{1}{10} < |z| < 4\right\}, \quad a_\pm > 1.$$

Note that

- *As we have that* $\rho_- < a_-$, *then we can take* $\rho_- = \frac{31}{2} = 15.5$ $(\Rightarrow \frac{R}{\rho_-} = \frac{1}{5})$.
- *Note that:* $3.1 < R\rho_+ < Ra_+ = 4$. *We will take several values of* $R\rho_+$ *to study the behavior of the error.*

Consider $u = \begin{bmatrix} -4 \\ -7 \end{bmatrix}$ *and* $v = \begin{bmatrix} -10 \\ 11 \end{bmatrix}$, *and compute the number of nodes* N_i *such that the trapezoid rule error is less than* 10^{-6}. *Then, we obtain the results presented in Tables 4.1, 4.2 and 4.3, where we fix* M_2 *and vary* M_1 *to study the behavior of the problem. Moreover, in Figures 4.3, 4.4 and 4.5 we show a log-10 plot of* $|E_N(f)|$ *for the moments* μ_k, *for* $k = 0, \ldots, 5$.

$R\rho_+$	M_1	M_2	$\lvert E_N(f)\rvert_{k=0}$	N_0	$\lvert E_N(f)\rvert_{k=1}$	N_1
3.2	3195	71.1	$\frac{10224}{1.03226^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	726	$\frac{32716.8}{1.03226^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	763
3.3	2297	71.1	$\frac{7580.1}{1.06452^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	364	$\frac{25014.33}{1.06452^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	383
3.4	1905	71.1	$\frac{6477}{1.09677^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	245	$\frac{22021.8}{1.09677^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	258
3.5	1739	71.1	$\frac{6086.5}{1.12903^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	186	$\frac{21302.75}{1.12903^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	196
3.6	1728	71.1	$\frac{6220.8}{1.1613^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	151	$\frac{22394.88002}{1.1613^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	160
3.7	1889	71.1	$\frac{6989.3}{1.19355^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	129	$\frac{25860.41}{1.19355^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	136
3.8	2378	71.1	$\frac{9036.4}{1.2258^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	113	$\frac{34338.32002}{1.2258^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	120
3.9	4062	71.1	$\frac{15841.8}{1.2581^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	103	$\frac{61783.02}{1.2581^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	109
3.98	18093	71.1	$\frac{72010.14}{1.2839^N-1} + \frac{14.22}{\left(\frac{31}{2}\right)^N-1}$	101	$\frac{286600.3574}{1.2839^N-1} + \frac{2.844}{\left(\frac{31}{2}\right)^N-1}$	106

Table 4.1: Trapezoid Error Example: μ_k for $k = 0, 1$.

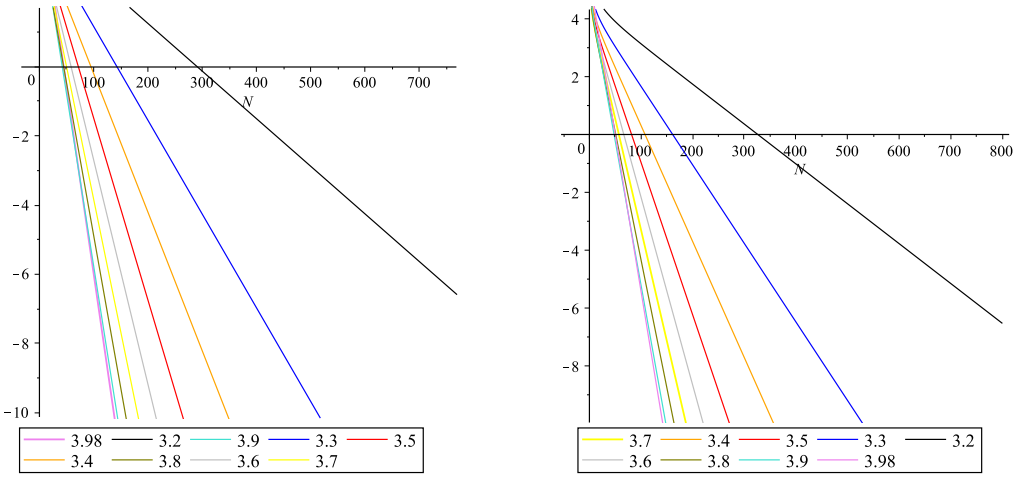

Figure 4.3: $\log|E_N(f)|$ for μ_0 and μ_1

$R\rho_+$	M_1	M_2	$\lvert E_N(f)\rvert_{k=2}$	N_2	$\lvert E_N(f)\rvert_{k=3}$	N_3
3.2	3195	71.1	$\frac{104693.7602}{1.03226^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	800	$\frac{335020.0330}{1.03226^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	836
3.3	2297	71.1	$\frac{82547.289}{1.06452^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	403	$\frac{272406.0537}{1.06452^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	422
3.4	1905	71.1	$\frac{74874.12006}{1.09677^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	272	$\frac{254572.0084}{1.09677^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	285
3.5	1739	71.1	$\frac{74559.625}{1.12903^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	207	$\frac{260958.6875}{1.12903^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	217
3.6	1728	71.1	$\frac{80621.56807}{1.1613^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	168	$\frac{290237.6451}{1.1613^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	177
3.7	1889	71.1	$\frac{95683.517}{1.19355^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	143	$\frac{354029.0129}{1.19355^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	151
3.8	2378	71.1	$\frac{130485.6161}{1.2258^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	126	$\frac{495845.3413}{1.2258^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	133
3.9	4062	71.1	$\frac{240953.778}{1.2581^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	115	$\frac{939719.7342}{1.2581^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	121
3.98	18093	71.1	$\frac{1140669.423}{1.2839^N-1}+\frac{0.5688}{\left(\frac{31}{2}\right)^N-1}$	112	$\frac{4539864.303}{1.2839^N-1}+\frac{0.11376}{\left(\frac{31}{2}\right)^N-1}$	117

Table 4.2: Trapezoid Error Example: μ_k for $k = 2, 3$.

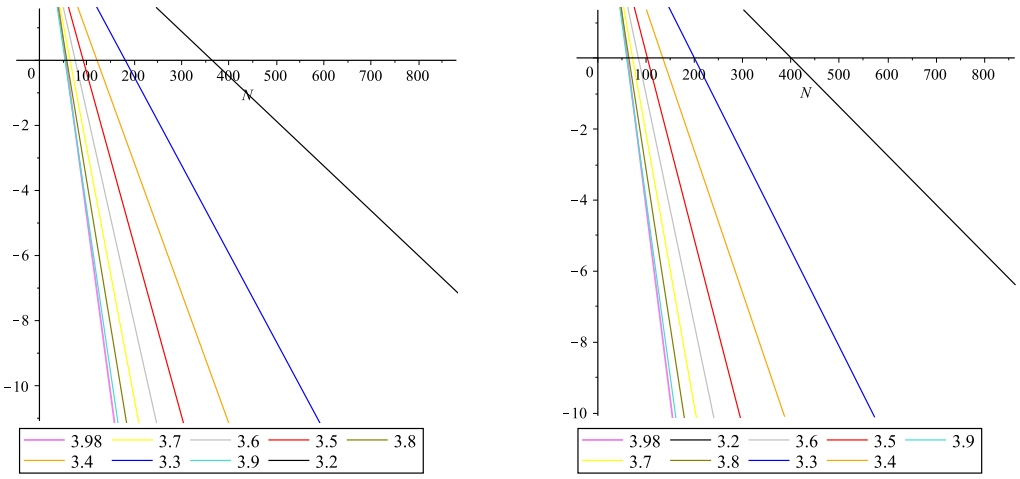

Figure 4.4: $\log|E_N(f)|$ for μ_2 and μ_3

$R\rho_+$	M_1	M_2	$\lvert E_N(f)\rvert_{k=4}$	N_4	$\lvert E_N(f)\rvert_{k=5}$	N_5
3.2	3195	71.1	$\frac{1072064.106}{1.03226^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	873	$\frac{3430605.140}{1.03226^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	910
3.3	2297	71.1	$\frac{898939.9772}{1.06452^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	441	$\frac{2966501.925}{1.06452^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	460
3.4	1905	71.1	$\frac{865544.8285}{1.09677^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	298	$\frac{2942852.418}{1.09677^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	311
3.5	1739	71.1	$\frac{913355.4062}{1.12903^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	227	$\frac{3196743.922}{1.12903^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	238
3.6	1728	71.1	$\frac{1044855.523}{1.1613^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	186	$\frac{3761479.884}{1.1613^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	194
3.7	1889	71.1	$\frac{1309907.348}{1.19355^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	158	$\frac{4846657.187}{1.19355^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	166
3.8	2378	71.1	$\frac{1884212.297}{1.2258^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	139	$\frac{7160006.733}{1.2258^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	146
3.9	4062	71.1	$\frac{3664906.963}{1.2581^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	127	$\frac{14293137.16}{1.2581^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	132
3.98	18093	71.1	$\frac{18068659.93}{1.2839^N-1}+\frac{0.022752}{\left(\frac{31}{2}\right)^N-1}$	123	$\frac{71913266.53}{1.2839^N-1}+\frac{0.0045504}{\left(\frac{31}{2}\right)^N-1}$	128

Table 4.3: Trapezoid Error Example: μ_k for $k = 4, 5$.

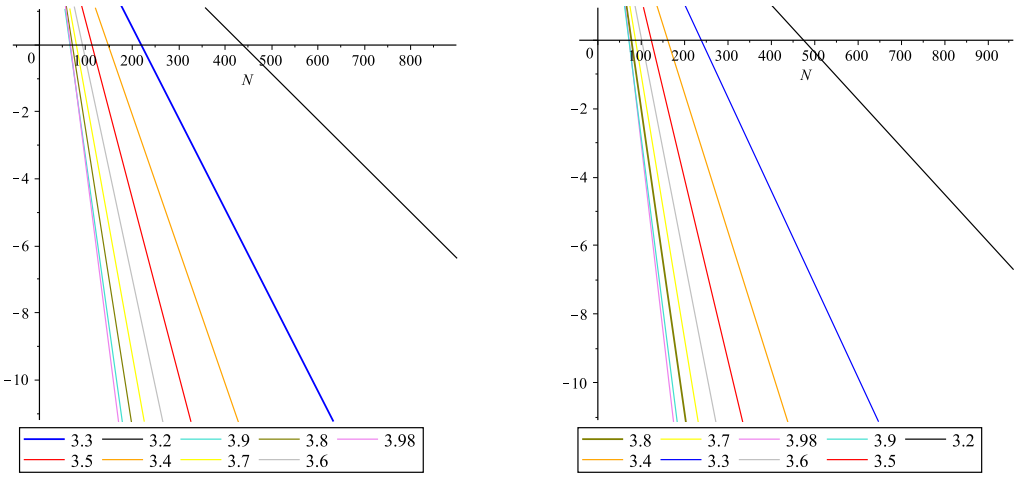

Figure 4.5: $\log|E_N(f)|$ for μ_4 and μ_5

Chapter 5 :

Iterative Refinement of Invariant Pairs and Matrix Solvents

5.1 Iterative Refinement of Invariant Pairs and Matrix Solvents

Once an invariant pair has been numerically computed or approximated, it can be refined using an iterative method such as Newton: this is, for instance, the strategy proposed in [17]. Here, we experiment with some modifications to the classical Newton's method applied to the equation $P(X, S) = 0$.

5.2 Newton's Method

Newton's method, also called the Newton-Raphson method, is a root-finding algorithm that uses the first few terms of the Taylor series of a function $f(x)$ in the vicinity of a suspected root (see [106], [113]).

Consider the Taylor series of $f(x)$ expanded about the point $x = x_0 + \epsilon$:

$$f(x_0 + \epsilon) = f(x_0) + f^{'}(x_0)\epsilon + \frac{1}{2}f^{''}(x_0)\epsilon^2 + \cdots .$$

Keeping terms only to first order, we have:

$$f(x_0 + \epsilon) \approx f(x_0) + f^{'}(x_0)\epsilon.$$

Setting $f(x_0 + \epsilon) = 0$ and solving last equation for $\epsilon = \epsilon_0$, gives:

$$\epsilon_0 = -\frac{f(x_0)}{f^{'}(x_0)},$$

which is the first-order adjustment to the root's position. Taking $x_1 = x_0 + \epsilon_0$ and calculating a new ϵ_1, and so on, the process can be repeated until it converges to a fixed point (which is precisely a root) using:

$$\epsilon_n = -\frac{f(x_n)}{f^{'}(x_n)}.$$

Then, given a good initial choice of the root, the algorithm is iteratively given by:

$$x_{k+1} = x_k - \frac{f(x_k)}{f^{'}(x_k)}, \quad \text{for } n = 1, 2, \ldots .$$

In the case of invariant pairs, Newton's method defines $(\Delta X, \Delta S)$ by

$$P(X, S) + \mathbb{D}P_{(X,S)}(\Delta X, \Delta S) = 0. \tag{5.1}$$

And for the case of matrix solvents, Newton's method defines ΔS by

$$P(S) + \mathbb{D}P_S(\Delta S) = 0. \tag{5.2}$$

As explained in [17], the definition of a simple invariant pair (X, S), i.e., a pair such that $P(X, S) = 0$, is not sufficient to characterize (X, S). Then, we must add the condition

$$W^H V_m(X, S) = I_k,$$

where $m \leq \ell$ is not smaller than the minimality index of (X, S) and the columns of $W = [W_{m-1}^H, \ldots, W_0^H]^H$ form an orthonormal basis of $\text{span}(V_m(X, S))$. In other words, the idea of the Newton's method is the following: given an initial approximation (X_0, S_0) to a simple invariant pair $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$ our goal is to compute a correction $(\Delta X_k, \Delta S_k)$, which brings(X_0, S_0) closer to (X, S). This is, the Newton's method applied to the equations:

$$P(X, S) = 0, \qquad \mathbb{V}(X, S) = 0,$$

where $\mathbb{V}(X, S) = W^H V_m(X, S) - I_k$, for some normalization matrix $W = [W_{m-1}^H, \ldots, W_0^H]^H \in \mathbb{C}^{k\times mn}$, takes the form:

$$(X_{k+1}, S_{k+1}) = (X_k, S_k) - \mathbb{L}_k^{-1} \left(P(X_k, S_k), \mathbb{V}(X_k, S_k)\right),$$

where $\mathbb{L}_k$ is the Jacobian at the current iterate (X_k, S_k):

$$\mathbb{L}_k(\Delta X, \Delta S) = \left(P(\Delta X, S_k) + \sum_{j=1}^{\ell} A_j X_k \mathbb{D}S_k^j(\Delta S), \sum_{j=1}^{m-1} W_j^H (\Delta X S_k^j + X \mathbb{D}S_k^j(\Delta S)) \right).$$

The methods for solving the correction equations will be discussed in Section 5.5. For now, we will describe the global algorithm and its variants.

5.2.1 Algorithm: Newton's Method

Given an initial approximation (X_0, S_0) to the solution of (2.5) (resp. S_0 to the solution of (3.2)), a tolerance ϵ and a contour $\Gamma \in \mathbb{C}$ (with the spectrum of S_0 in its interior), we have:

Algorithm 1.
STEP 1: Set $k = 0$
STEP 2: If $error_k < \epsilon$: ***STOP***

STEP 3: Solve for $(\Delta X_k, \Delta S_k)$ *(resp. for* ΔS_k*) the equation:*

$$\mathbb{D}P_{(X,S)}(\Delta X_k, \Delta S_k) = -(P(X_k, S_k), 0)$$
$$(resp.\ \mathbb{D}P_S(\Delta S_k) = -(P(S_k), 0))$$

STEP 4: Update:

- $X_{k+1} = X_k + \Delta X_k$, $S_{k+1} = S_k + \Delta S_k$ *(resp.* $S_{k+1} = S_k + \Delta S_k$*).*
- $k = k + 1$
- *go to STEP 2.*

Remark 12. *In the case of invariant pairs, we define* $error_k$ *as:* $error_k = \frac{\|P(X_k,S_k)\|_F}{\|X_k\|_F}$ *and in the case of matrix solvents:* $error_k = \frac{\|P(S_k)\|_F}{\|S_k\|_F}$.

5.3 Incorporating Line Search into Newton's Method

Line searches are relatively inexpensive and improve the global convergence properties of Newton's method. Each iteration of a line search method computes a search direction d_k and then decides how far to move along that direction. The iteration is given by:

$$x_{k+1} = x_k + t_k d_k,$$

where the positive scalar t_k is the step length. The success of a line search method depends on effective choices of both the direction d_k and the step length t_k (see [129]). A value $t_k = 1$ gives the original Newton iteration.

In this section, we show how to incorporate exact line searches into Newton's method to approximate invariant pairs (X, S) and solvents for the general matrix solvent problem $P(S) = 0$. For that, we use the contour integral formulations (2.5) and (3.2).

As pointed out before, the use of Newton's method incorporating line searches to find solvents is not new. For instance, in [63] and [88] this approach is used to approximate solvents for the quadratic matrix equation.

5.3.1 Case of Invariant Pairs

In the specific problem (2.5), the direction d_k is given by the solution $(\Delta X_k, \Delta S_k)$ of the correction equation (5.5) (see Section 5.5). The step length t_k on each iteration is given by the solution of the minimization problem:

$$p(t) = \|P(X + t\Delta X, S + t\Delta S)\|_F^2.$$

Using the formula for the total derivative of P at (X, S) in direction $(\Delta X, \Delta S)$:

$$\mathbb{D}P_{(X,S)}(\Delta X, \Delta S) = \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left(\Delta X + X(\lambda I - S)^{-1}\Delta S\right)(\lambda I - S)^{-1}d\lambda, \tag{5.3}$$

we have, at second order in $\|\Delta X\|$ and $\|\Delta S\|$:

$$\begin{aligned}
P(X + t\Delta X, S + t\Delta S) &= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(X + t\Delta X)(\lambda I - S - t\Delta S)^{-1}d\lambda = \\
&= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(X + t\Delta X)\left((\lambda I - S)^{-1} + (\lambda I - S)^{-1}t\Delta S(\lambda I - S)^{-1}\right. \\
&\left.+(\lambda I - S)^{-1}t\Delta S(\lambda I - S)^{-1}t\Delta S(\lambda I - S)^{-1} + \cdots\right)d\lambda \\
&\approx \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(X + t\Delta X)\left((\lambda I - S)^{-1} + t(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\right. \\
&\left.+t^2(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\right)d\lambda = \\
&= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}d\lambda \\
&+ t\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta X + X(\lambda I - S)^{-1}\Delta S\right](\lambda I - S)^{-1}d\lambda\right] + \\
&+ t^2\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta X + X(\lambda I - S)^{-1}\Delta S\right](\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right] + \\
&+ t^3\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\Delta X(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right] = \\
&= P(X, S) + t\mathbb{D}P_{(X,S)}(\Delta X, \Delta S) \\
&+ t^2\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta X + X(\lambda I - S)^{-1}\Delta S\right](\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right] + \\
&+ t^3\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\Delta X(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right].
\end{aligned}$$

Recalling that Newton's method defines $(\Delta X, \Delta S)$ by (5.1), we have:

$$\begin{aligned}
P(X + t\Delta X, S + t\Delta S) &= (1 - t)P(X, S) \\
&+ t^2\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta X + X(\lambda I - S)^{-1}\Delta S\right](\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right] \\
&+ t^3\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\Delta X(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right].
\end{aligned}$$

Thus

$$\begin{aligned}
p(t) &= (1 - t)^2\|P(X, S)\|_F^2 + t^4\|A\|_F^2 + t^6\|B\|_F^2 \\
&\quad + t^2(1 - t)\mathrm{trace}(P(X, S)^*A + A^*P(X, S)) \\
&\quad + t^3(1 - t)\mathrm{trace}(P(X, S)^*B + B^*P(X, S)) \\
&\quad + t^5\mathrm{trace}(A^*B + B^*A) \\
&\equiv (1 - t)^2\alpha + t^4\theta + t^6\varphi + t^2(1 - t)\beta + t^3(1 - t)\gamma + t^5\eta \\
&= t^6\varphi + t^5\eta + t^4(\theta - \gamma) + t^3(\gamma - \beta) + t^2(\alpha + \beta) - 2\alpha t + \alpha,
\end{aligned}$$

where:

$$A = \frac{1}{2\pi\imath}\oint_{\Gamma} P(\lambda)\left[\Delta X + X(\lambda I - S)^{-1}\Delta S\right](\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda,$$
$$B = \frac{1}{2\pi\imath}\oint_{\Gamma} P(\lambda)\Delta X(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda,$$
$$\alpha = \|P(X,S)\|_F^2,\ \theta = \|A\|_F^2,\ \varphi = \|B\|_F^2, \eta = \mathrm{trace}(A^*B + B^*A),$$
$$\beta = \mathrm{trace}(P(X,S)^*A + A^*P(X,S)), \gamma = \mathrm{trace}(P(X,S)^*B + B^*P(X,S)).$$

5.3.2 Case of Matrix Solvents

For the problem (3.2) we have the minimization problem:

$$p(t) = \|P(S + t\Delta S)\|_F^2.$$

Using the formula for the derivative of the matrix inverse:

$$\frac{dA^{-1}}{dt} = -A^{-1}\frac{dA}{dt}A^{-1},$$

we write the formula for the total derivative of P at S in direction ΔS as follows:

$$\mathbb{D}P_S(\Delta S) = \frac{1}{2\pi\imath}\oint_{\Gamma} P(\lambda)(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda. \tag{5.4}$$

Then, we have:

$$\begin{aligned}
P(S+t\Delta S) &= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda I - S - t\Delta S)^{-1}d\lambda \\
&= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[(\lambda I - S)^{-1} + (\lambda I - S)^{-1}t\Delta S(\lambda I - S)^{-1}\right. \\
&\left.+(\lambda I - S)^{-1}t\Delta S(\lambda I - S)^{-1}t\Delta S(\lambda I - S)^{-1} + \cdots\right]d\lambda \approx \\
&\approx \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[(\lambda I - S)^{-1} + t(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\right. \\
&\left.+t^2(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\right]d\lambda = \\
&= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda I - S)^{-1}d\lambda \\
&+ t\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right] + \\
&+ t^2\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right] = \\
&= P(S) + t\mathbb{D}P_S(\Delta S) \\
&+ t^2\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right].
\end{aligned}$$

Recalling that Newton's method defines ΔS by (5.2), we have:

$$P(S) + \mathbb{D}P_S(\Delta S) = 0,$$

then, we have:

$$\begin{aligned}
P(S+t\Delta S) &= (1-t)P(S) + \\
&+ t^2\left[\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda\right].
\end{aligned}$$

Thus

$$\begin{aligned}
p(t) &= (1-t)^2\|P(S)\|_F^2 + t^4\|A\|_F^2 + t^2(1-t)\mathrm{trace}(P(S)^*A + A^*P(S)) \\
&\equiv (1-t)^2\alpha + t^4\theta + t^2(1-t)\beta = \\
&= t^4\theta - t^3\beta + t^2(\alpha+\beta) - 2\alpha t + \alpha,
\end{aligned}$$

where:

$$\begin{aligned}
A &= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda, \\
\alpha &= \|P(S)\|_F^2, \quad \theta = \|A\|_F^2, \quad \beta = \mathrm{trace}(P(S)^*A + A^*P(S)).
\end{aligned}$$

5.3.3 Algorithm: Newton's Method with Line Search

Given an initial approximation (X_0, S_0) to the solution of (2.5) (resp. S_0 to the solution of (3.2)), a tolerance ϵ and a contour $\Gamma \in \mathbb{C}$ (with the spectrum of S_0 in its interior), we have:

Algorithm 2.
STEP 1: Set $k = 0$
STEP 2: If $error_k < \epsilon$: ***STOP***
STEP 3: Solve for $(\Delta X_k, \Delta S_k)$ *(resp. for* ΔS_k*) the equation:*

$$\mathbb{D}P_{(X,S)}(\Delta X_k, \Delta S_k) = -(P(X_k, S_k), 0)$$
$$(resp.\ \mathbb{D}P_S(\Delta S) = -(P(S_k), 0))$$

STEP 4: Find by exact line searches a value of t *that minimizes the function:*

$$\min_{t\in[0,2]} \|P(X + t\Delta X, S + t\Delta S)\|_F^2$$
$$(resp.\ \min_{t\in[0,2]} \|P(S + t\Delta S)\|_F^2)$$

STEP 5: Update:

- $X_{k+1} = X_k + t\Delta X_k$, $S_{k+1} = S_k + t\Delta S_k$ *(resp.* $S_{k+1} = S_k + t\Delta S_k$*).*
- $k = k + 1$
- *go to STEP 2.*

5.4 Šamanskii's Technique

Another variation on Newton's method is Šamanskii's technique, which accelerates the quadratic convergence to cubic. Therefore, we must ensure we have quadratic convergence in order to use this technique (see [126]).

5.4.1 Algorithm: Newton's Method with Line Search and Šamanskii Technique

Given an initial approximation (X_0, S_0) to the solution of (2.5) (resp. S_0 to the solution of (3.2)), tolerances ϵ and ϵ_0 and a contour $\Gamma \in \mathbb{C}$ (with the spectrum of S_0 in its interior), we have:

Algorithm 3.
STEP 1: Set $k = 0$
STEP 2: If $error_k < \epsilon$: ***STOP***
STEP 3: Solve for $(\Delta X_k, \Delta S_k)$ *(resp. for* ΔS_k*) the equation:*

$$\mathbb{D}P_{(X,S)}(\Delta X_k, \Delta S_k) = -(P(X_k, S_k), 0)$$
$$(resp.\ \mathbb{D}P_S(\Delta S) = -(P(S_k), 0))$$

STEP 4: If $error_k < \epsilon_0$*: go to STEP 7.*
STEP 5: Find by exact line searches a value of t *that minimizes the function:*

$$\min_{t\in[0,2]} \|P(X + t\Delta X, S + t\Delta S)\|_F^2$$
$$(resp.\ \min_{t\in[0,2]} \|P(S + t\Delta S)\|_F^2)$$

STEP 6: Update

- $X_{k+1} = X_k + t\Delta X_k$, $S_{k+1} = S_k + t\Delta S_k$ *(resp.* $S_{k+1} = S_k + t\Delta S_k$*).*
- $k = k + 1$ *and go to STEP 2.*

STEP 7: Update $X_{k,1} = X_k + \Delta X_k$ *and* $S_{k,1} = S_k + \Delta S_k$ *(resp.* $S_{k,1} = S_k + \Delta S_k$*).*
STEP 8: Solve for $(\Delta X_{k,1}, \Delta S_{k,1})$ *(resp.* $\Delta S_{k,1}$*) the equation:*

$$\mathbb{D}P_{(X,S)}(\Delta X_{k,1}, \Delta S_{k,1}) = -(P(X_{k,1}, S_{k,1}), 0)$$
$$(resp.\ \mathbb{D}P_S\Delta S_{k,1} = -(P(S_{k,1}), 0))$$

STEP 9: Update

- $X_{k+1} = X_{k,1} + \Delta X_{k,1}$, $S_{k+1} = S_{k,1} + \Delta S_{k,1}$ *(resp.* $S_{k+1} = S_{k,1} + \Delta S_{k,1}$*).*
- $k = k + 1$ *and go to STEP 2.*

5.5 Solution of the Correction Equation

In this section, we use Newton's Method and its variations to approximate a solution for equation (2.5) (resp. for equation(3.2)). We must find the correction $(\Delta X, \Delta S)$ (resp. ΔS) for the equations:

$$\mathbb{D}P_{(X,S)}(\Delta X_k, \Delta S_k) = -(P(X_k, S_k), 0) \tag{5.5}$$

and respectively for:

$$\mathbb{D}P_S(\Delta S) = -(P(S_k), 0).$$

In next sections, we will show two ways to find this correction. The first one uses a Kronecker product approach and the second uses forward substitution. We will focus in the case of the correction equation for invariant pairs; the corresponding equation for matrix solvents can be used in a similar way.

5.5.1 Using the Kronecker Product

Consider the equation (5.5). Vectorizing its left part and using (A.1), we obtain:

$$\begin{aligned}
&\mathrm{vec}\left(\mathbb{D}P(\Delta X, \Delta S)\right) = \\
&= \mathrm{vec}\left(\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left(\Delta X + X(\lambda I - S)^{-1}\Delta S\right)(\lambda I - S)^{-1}d\lambda\right) = \\
&= \frac{1}{2\pi\imath}\oint_\Gamma \mathrm{vec}\left(P(\lambda)\left(\Delta X + X(\lambda I - S)^{-1}\Delta S\right)(\lambda I - S)^{-1}\right)d\lambda = \\
&= \frac{1}{2\pi\imath}\oint_\Gamma \mathrm{vec}\left(P(\lambda)\Delta X(\lambda I - S)^{-1}\right)d\lambda + \\
&+ \frac{1}{2\pi\imath}\oint_\Gamma \mathrm{vec}\left(P(\lambda)X(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}\right)d\lambda = \\
&= \frac{1}{2\pi\imath}\oint_\Gamma \left((\lambda I - S)^{-T}\otimes P(\lambda)\right)d\lambda\ \mathrm{vec}\left(\Delta X\right) + \\
&+ \frac{1}{2\pi\imath}\oint_\Gamma \left((\lambda I - S)^{-T}\otimes P(\lambda)X(\lambda I - S)^{-1}\right)d\lambda\ \mathrm{vec}\left(\Delta S\right).
\end{aligned}$$

Then, we can rewrite the equation $\mathbb{D}P(\Delta X, \Delta S) = (P(X_p, S_p), 0)$ as:

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}\begin{bmatrix} \mathrm{vec}\left(\Delta X\right) \\ \mathrm{vec}\left(\Delta S\right) \end{bmatrix} = \begin{bmatrix} \mathrm{vec}\left(P(X_p, S_p)\right) \\ 0 \end{bmatrix}, \tag{5.6}$$

where

$$\begin{aligned}
K_{11} &= \frac{1}{2\pi\imath}\oint_\Gamma \left((\lambda I - S)^{-T}\otimes P(\lambda)\right)d\lambda, \\
K_{12} &= \frac{1}{2\pi\imath}\oint_\Gamma \left((\lambda I - S)^{-T}\otimes P(\lambda)X(\lambda I - S)^{-1}\right)d\lambda, \\
K_{21} &= \sum_{j=0}^{m-1}((S^j)^T\otimes W_j^H), \\
K_{22} &= \sum_{j=1}^{m-1}(I_k\otimes W_j^H X)K_{S^j},
\end{aligned}$$

where K_{S^j} denotes the Kronecker product formulation of the Fréchet derivative, i.e.:

$$K_{S^j} = \sum_{i=0}^{j-1}((S^{j-i-1})^T \otimes S^i.$$

5.5.2 Using Forward Substitution

Using the Schur decomposition of S_p and an appropriate unitary transformation of the pair (X_p, S_p), we can assume that S_p is in upper triangular form. Using this special structure of S_p, we can easily find the columns of ΔX and ΔS successively using a forward substitution process.
Note that using the upper triangular structure of S_p in the equation:

$$\mathbb{D}P(\Delta X, \Delta S) = (P(X_p, S_p), 0) \tag{5.7}$$

the computation of the first columns Δx_1 and Δs_1 of ΔX and ΔS, respectively, is simple. Assuming that S is an upper triangular matrix, then $(\lambda I - S)^{-1}$ is also upper triangular. Its diagonal entries are $(\lambda I - s_{11})^{-1}, (\lambda I - s_{22})^{-1}, \ldots, (\lambda I - s_{nn})^{-1}$, where $s_{11}, s_{22}, \ldots, s_{nn}$ are the diagonal entries of S. In order to compute Δx_1 and Δs_1, we can write:

$$\begin{aligned}\mathbb{D}P(X,S) &= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left(\Delta x_1 + X(\lambda I - S)^{-1}\Delta s_1\right)(\lambda - s_{11})^{-1}d\lambda = \\ &= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\Delta x_1(\lambda - s_{11})^{-1}d\lambda + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}\Delta s_1(\lambda - s_{11})^{-1}d\lambda.\end{aligned}$$

Then, we have:

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta s_1 \end{bmatrix} = \begin{bmatrix} r_1 \\ 0 \end{bmatrix}, \tag{5.8}$$

where r_1 denotes the first column of $Res = P(X_p, S_p)$ and:

$$\begin{aligned} B_{11} &= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)(\lambda - s_{11})^{-1}d\lambda, \\ B_{12} &= \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}(\lambda - s_{11})^{-1}d\lambda, \\ B_{21} &= \sum_{j=0}^{m-1} s_{11}^j W_j^H, \\ B_{22} &= \sum_{j=1}^{m-1} W_j^H X\left[\mathbb{D}S^j\right]_{11}, \end{aligned}$$

where $[\mathbb{D}S^j]_{11}$ denotes the Fréchet derivative of the first column of S^j with respect to the first column of S:

$$\left[\mathbb{D}S^1\right]_{11} = I_k, \quad \left[\mathbb{D}S^j\right]_{11} = s_{11}\left[\mathbb{D}S^{j-1}\right]_{11} + S^{j-1},\ j \geq 2$$

For the second columns of ΔX and ΔS, we can find an equation of the form of (5.8), but first we must update the right side of the equation (5.7). We will now describe how to do this update.
First, consider the partitions:

$$\Delta X = [\Delta x_1, \Delta X_2], \quad \Delta S = [\Delta s_1, \Delta S_2], \quad Res = [r_1, Res_2],$$
$$S = \begin{bmatrix} s_{11} & s_{12} \\ 0 & S_{22} \end{bmatrix} \text{ and } S^j \begin{bmatrix} s_{11}^j & [S^j]_{12} \\ 0 & S_{22}^j \end{bmatrix}.$$

Note also that using the formula for the inverse of a block upper triangular matrix [104], we have:

$$(\lambda I - S)^{-1} = \begin{bmatrix} (\lambda - s_{11})^{-1} & (\lambda - s_{11})^{-1} s_{12} (\lambda I - S_{22})^{-1} \\ 0 & (\lambda I - S_{22})^{-1} \end{bmatrix}.$$

Inserting this in equation (5.3), we have:

$$\begin{aligned}
\mathbb{D}P(X,S) = {} & \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\Delta X(\lambda I - S)^{-1}d\lambda + \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}\Delta S(\lambda I - S)^{-1}d\lambda = \\
& = \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta x_1, \Delta X_2\right](\lambda I - S)^{-1}d\lambda \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}\left[\Delta s_1, \Delta S_2\right](\lambda I - S)^{-1}d\lambda = \\
& = \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta x_1, 0\right](\lambda I - S)^{-1}d\lambda + \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[0, \Delta X_2\right](\lambda I - S)^{-1}d\lambda + \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}\left[\Delta s_1, 0\right](\lambda I - S)^{-1}d\lambda \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}\left[0, \Delta S_2\right](\lambda I - S)^{-1}d\lambda = \\
& = \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta x_1(\lambda - s_{11})^{-1} \quad \Delta x_1(\lambda - s_{11})^{-1}s_{12}(\lambda I - S_{22})^{-1}\right]d\lambda + \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[0 \quad \Delta X_2(\lambda I - S_{22})^{-1}\right]d\lambda + \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}\left[\Delta s_1(\lambda - s_{11})^{-1} \quad \Delta s_1(\lambda - s_{11})^{-1}s_{12}(\lambda I - S_{22})^{-1}\right]d\lambda + \\
& + \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I - S)^{-1}\left[0 \quad \Delta S_2(\lambda I - S_{22})^{-1}\right]d\lambda.
\end{aligned}$$

Multiplying last equation by $\begin{bmatrix} 0 \\ I_{k-1} \end{bmatrix}$ to consider just the last $k-1$ columns, we have:

$$\begin{aligned}
&\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\begin{bmatrix}\Delta x_1(\lambda-s_{11})^{-1} & \Delta x_1(\lambda-s_{11})^{-1}s_{12}(\lambda I-S_{22})^{-1}\end{bmatrix}\begin{bmatrix}0\\ I_{k-1}\end{bmatrix}d\lambda+\\
&+\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\begin{bmatrix}0 & \Delta X_2(\lambda I-S_{22})^{-1}\end{bmatrix}\begin{bmatrix}0\\ I_{k-1}\end{bmatrix}d\lambda+\\
&+\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I-S)^{-1}\begin{bmatrix}\Delta s_1(\lambda-s_{11})^{-1} & \Delta s_1(\lambda-s_{11})^{-1}s_{12}(\lambda I-S_{22})^{-1}\end{bmatrix}\begin{bmatrix}0\\ I_{k-1}\end{bmatrix}d\lambda+\\
&+\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I-S)^{-1}\begin{bmatrix}0 & \Delta S_2(\lambda I-S_{22})^{-1}\end{bmatrix}\begin{bmatrix}0\\ I_{k-1}\end{bmatrix}d\lambda=\\
&\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\Delta x_1(\lambda-s_{11})^{-1}s_{12}(\lambda I-S_{22})^{-1}d\lambda+\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\Delta X_2(\lambda I-S_{22})^{-1}d\lambda+\\
&+\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I-S)^{-1}\Delta s_1(\lambda-s_{11})^{-1}s_{12}(\lambda I-S_{22})^{-1}d\lambda+\\
&+\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)X(\lambda I-S)^{-1}\Delta S_2(\lambda I-S_{22})^{-1}d\lambda,
\end{aligned}$$

obtaining for the pair $(\Delta X_2, \Delta S_2) \in \mathbb{C}^{n\times(k-1)} \times \mathbb{C}^{n\times(k-1)}$ the linear matrix equation:

$$\begin{aligned}
&\frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta X_2 + X(\lambda I-S)^{-1}\Delta S_2\right](\lambda I-S_{22})^{-1}d\lambda = \widehat{Res}_2\\
&\sum_{j=0}^{m-1} W_j^H\left(\Delta X_2 S_{22}^j + X\mathbb{D}S^j([0,\Delta S_2])\begin{bmatrix}0\\ I_{k-1}\end{bmatrix}\right) = \widehat{Ort}_2,
\end{aligned}$$

where the updated right hand sides are:

$$\begin{aligned}
\widehat{Res}_2 &:= Res_2 - \frac{1}{2\pi\imath}\oint_\Gamma P(\lambda)\left[\Delta x_1 + X(\lambda I-S)^{-1}\Delta s_1\right](\lambda-s_{11})^{-1}s_{12}(\lambda I-S_{22})^{-1}d\lambda\\
\widehat{Ort}_2 &:= -\sum_{j=0}^{m-1} W_j^H\left(\Delta x_1\left[S^j\right]_{12} + X\mathbb{D}S^j([\Delta s_1,0])\begin{bmatrix}0\\ I_{k-1}\end{bmatrix}\right).
\end{aligned}$$

Taking r_2 and q_2 as the first columns of $\widehat{Res}_2$ and $\widehat{Ort}_2$, respectively, we can find the second columns Δx_2 and Δs_2 of ΔX and ΔS, respectively, solving the linear system:

$$\begin{bmatrix}B_{11} & B_{12}\\ B_{21} & B_{22}\end{bmatrix}\begin{bmatrix}\Delta x_2\\ \Delta s_2\end{bmatrix} = \begin{bmatrix}r_2\\ q_2\end{bmatrix},$$

where:

$$
\begin{aligned}
B_{11} &= \frac{1}{2\pi\imath}\oint_{\Gamma} P(\lambda)(\lambda - s_{22})^{-1}d\lambda,\\
B12 &= \frac{1}{2\pi\imath}\oint_{\Gamma} P(\lambda)X(\lambda I - S)^{-1}(\lambda - s_{22})^{-1}d\lambda,\\
B_{21} &= \sum_{j=0}^{m-1} s_{22}^{j} W_j^H,\\
B_{22} &= \sum_{j=1}^{m-1} W_j^H X \left[\mathbb{D}S^j\right]_{22}.
\end{aligned}
$$

5.6 Numerical Results

In this section we compare two methods to refine approximate invariant pairs $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$: Newton's method (N.M.) presented in [17], Newton's method with line search (N.M.L.S.), explained in Section 5.3 and Newton's method with line search and Šamanskii technique (N.M.L.S.S.), explained in Section 5.4.

We have implemented Newton's method with the above variants in MATLAB and applied it to several problems taken from the NLEVP collection (see [16]). The results are shown in Table 5.1. For each problem, an initial invariant pair (X_0, S_0) has first been approximated using the (block) moment method of Section 4.3.1 and approximating the moments μ_i in (4.2) via the trapezoid rule discussed in Section 4.5, with $N = 20$ integration nodes. Moreover, Γ is chosen for each problem as the contour enclosing the k eigenvalues with largest condition number (computed using the MATLAB function `polyeig`).

Table 5.1 shows that line search is generally effective in reducing the number of iterations and the overall computation time.

Figure 5.1 shows the convergence of the Newton's method with line search, for the Dirac problem presented in Table 5.1. Here we use as contour the circle of center $C = -0.1$ and radius $R = 1.14$, which contains the 6 eigenvalues with largest condition number.

5.7 Functions Implemented in MATLAB and Maple

In this section, we describe the Maple and MATLAB implementations used in this thesis. These implementations are available online at the URL

`http://www.unilim.fr/pages_perso/esteban.segura/software.html`

			N.M.		N.M.L.S.		N.M.L.S.S.	
Problem	**Deg P**	**Size X**	**Ite**	**Time**	**Ite**	**Time**	**Ite**	**Time**
bicycle	2	2×2	23	0.082	16	0.112	12	0.05
butterfly	4	64×5	67	3.719	22	1.567	19	1.57
cd_player	2	60×6	500	N.C.	19	1.021	19	1.15
closed_loop	2	2×2	8	0.016	7	0.02	6	0.02
damped_beam	2	200×6	28	6.109	4	0.6037	3	0.80
dirac	2	80×6	500	N.C.	41	1.965	41	2.23
hospital	2	24×24	53	5.65	51	6.21	48	6.20
metal_strip	2	9×9	500	N.C.	28	0.589	23	0.48
mobile_manipulator	2	5×2	8	0.014	7	0.030	6	0.03
pdde_stability	2	225×6	29	9.644	16	5.622	17	8.17
planar_waveguide	4	129×6	72	11.148	19	3.682	17	5.34
plasma_drift	3	128×6	69	13.059	26	5.596	23	6.33
power_plant	2	8×8	15	0.34	13	0.39	11	0.50
railtrack	2	1005×3	32	199.365	28	209.471	25	216.53

Table 5.1: Comparison of results for classical Newton, Newton with line search and Newton with line search and Šamanskii's technique.

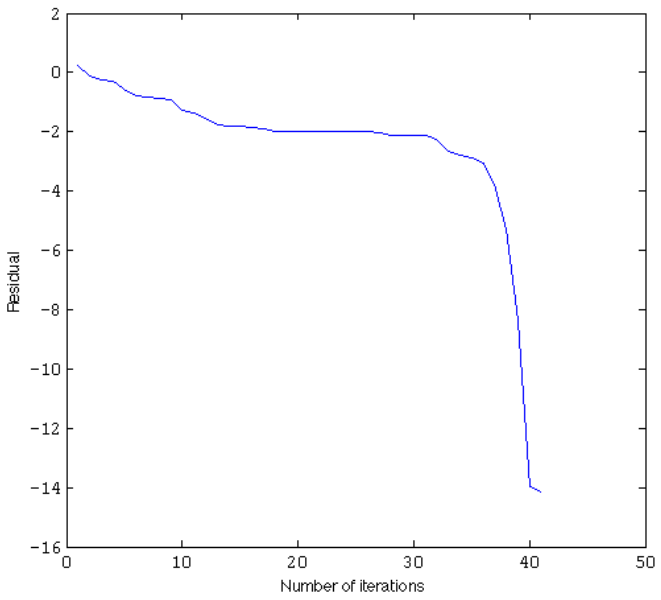

Figure 5.1: Convergence of Dirac problem using Newton's method with line search.This is a log-10 plot of the relative residual $\frac{\|P(X,S)\|_F}{\|X\|_F}$ versus the number of iterations.

MATLAB

Let $P(\lambda)$ be an $n \times n$ matrix polynomial of degree ℓ and Γ a contour (we limit our implementations to the case when Γ is a circle).

- `conditionNumInvPair.m`: This function approximates the condition number for an invariant pair (X, S) using equation (2.10). For this computation we use the values

$\alpha_i = \|A_i\|$, for $i = 0, \ldots, \ell$.
The input of this function is:

  * coeffs: cell array containing coefficients A_i of $P(\lambda)$.
  * (X, S): the invariant pair $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$.
  * C: center of the circle.
  * R: radius of the circle.
  * N: number of nodes of the trapezoid rule to approximate the contour integral.

- `condNumSolvent.m`: This function approximates the condition number for a solvent S using equation (3.4). For this computation we use the values $\alpha_i = \|A_i\|$, for $i = 0, \ldots, \ell$.
  The input of this function is:

  * coeffs: cell array containing coefficients A_i of $P(\lambda)$.
  * S: the matrix solvent $S \in \mathbb{C}^{n\times n}$.
  * C: center of the circle.
  * R: radius of the circle.
  * N: number of nodes of the trapezoid rule to approximate the contour integral.

- `numEigsContour.m`: This function approximates the number of eigenvalues of $P(\lambda)$ inside Γ using equation (4.10). The input of this function is:

  * coeffs: cell array containing coefficients A_i of $P(\lambda)$.
  * C: center of the circle.
  * R: radius of the circle.
  * N: number of nodes of the trapezoid rule to approximate the contour integral.

- `invariantPair.m`: This function approximates an invariant pair (X, S) of $P(\lambda)$. The approach used for this computation relies on the block version of the moment method presented in Section 4.3.2.
  The input of this function is:

  * coeffs: cell array containing coefficients A_i of $P(\lambda)$.
  * C: center of the circle.
  * R: radius of the circle.
  * N: number of nodes of the trapezoid rule to approximate the contour integral.

- `solvent.m`: This function approximates a solvent S of $P(\lambda)$. As explained in Section 3.3.1, the idea is to approximate first an invariant pair $(Y,T) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ and assuming that the matrix Y is invertible then we compute the solvent $S = YTY^{-1}$. The input of this function is:
  - coeffs: cell array containing coefficients A_i of $P(\lambda)$.
  - C: center of the circle.
  - R: radius of the circle.
  - N: number of nodes of the trapezoid rule to approximate the contour integral.
- `invariantPairCircles.m`: This function approximates an invariant pair (X,S) of $P(\lambda)$. The approach used for this computation relies on the block version of the moment method presented in Section 4.3.2.

  Consider the circles $\Gamma, \Gamma_1, \ldots, \Gamma_k$, where Γ_i are located inside Γ and the circles Γ_i, Γ_j do not intersect, for $i \neq j$ and $i = 1, \ldots, k$. This function computes the invariant pair (X,S), where the eigenvalues λ_r's of S are such that λ_r is inside of Γ but not inside of Γ_i, for $r = 1, \ldots$. For instance, consider Figure 5.2 and suppose we want to exclude the eigenvalues $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and λ_6. Then, we can use the circles Γ_1 and Γ_2 as in Figure 5.3, to prevent the computation of those eigenvalues.
  The input of this function is:
  - coeffs: cell array containing coefficients A_i of $P(\lambda)$.
  - C: list with centers of the circles. The first entry must contain the center of the main circle (which contains the smaller circles).
  - R: list with radius of the circles. The first entry must contain the radius of the main circle (which contains the smaller circles).
  - N: number of nodes of the trapezoid rule to approximate the contour integral.

Maple

Let $P(\lambda)$ be an $n \times n$ matrix polynomial of degree ℓ and Γ a contour (we limit our implementations to the case when Γ is a circle).

- `invariantPair.mpl`: This function computes an invariant pair (X,S) of $P(\lambda)$. The approach used for this computation relies on the block version of the moment method presented in Section 4.3.2.
  The input of this function is:
  - coeffs: list containing coefficients A_i of matrix polynomial $P(\lambda)$.

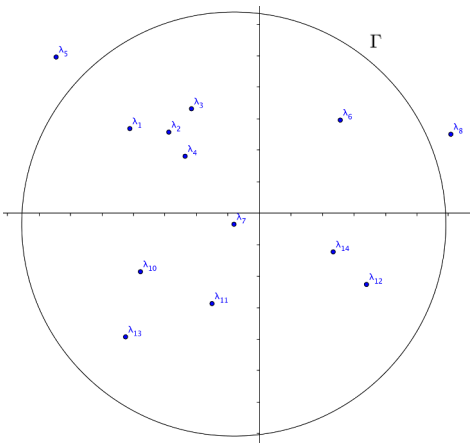

Figure 5.2: Eigenvalues inside the circle

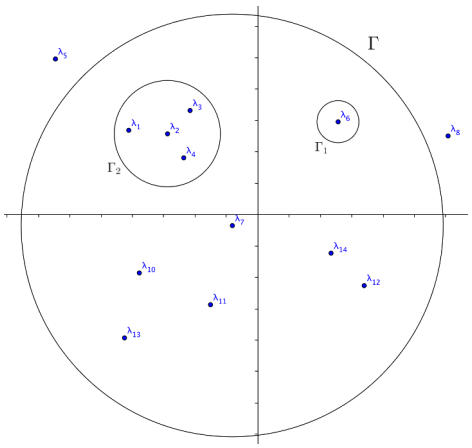

Figure 5.3: The eigenvalues that should be excluded from the computations are contained in the smaller circles

  - C: center of the circle.
  - R: radius of the circle.

- `invariantPairCircles.mpl`: This function computes an invariant pair (X, S) of $P(\lambda)$. It uses the same idea described in function `invariantPairCircles.m`.
  The input of this function is:
  - coeffs: list containing coefficients A_i of matrix polynomial $P(\lambda)$.
  - C: list with centers of the circles. The first entry must contain the center of the main circle (which contains the smaller circles).
  - R: list with radius of the circles. The first entry must contain the radius of the main circle (which contains the smaller circles).

- `solvent.mpl`: This function computes a solvent S of $P(\lambda)$. As explained in Section 3.3.1, the idea is to compute first an invariant pair $(Y, T) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ and assuming that the matrix Y is invertible then we compute the solvent $S = YTY^{-1}$.
  The input of this function is:
  - coeffs: list containing coefficients A_i of matrix polynomial $P(\lambda)$.
  - C: center of the circle.
  - R: radius of the circle.

- `condNumInvPair.mpl`: This function computes the condition number for an invariant pair (X, S) using equation (2.10). For this computation we use the values $\alpha_i = \|A_i\|$, for $i = 0, \ldots, \ell$.
  The input of this function is:

  - coeffs: list containing coefficients A_i of matrix polynomial $P(\lambda)$.
  - (X, S): the invariant pair $(X, S) \in \mathbb{C}^{n\times k} \times \mathbb{C}^{k\times k}$.
  - C: center of the circle.
  - R: radius of the circle.

- `condNumSol.mpl`: This function computes the condition number for a solvent S using equation (3.4). For this computation we use the values $\alpha_i = \|A_i\|$, for $i = 0, \ldots, \ell$.
  The input of this function is:
  - coeffs: list containing coefficients A_i of matrix polynomial $P(\lambda)$.
  - S: the matrix solvent $S \in \mathbb{C}^{n\times n}$.
  - C: center of the circle.
  - R: radius of the circle.

- `numberEigsContour.mpl`: This function computes the number of eigenvalues of $P(\lambda)$ inside Γ using equation (4.8).
  The input of this function is:
  - coeffs: list containing coefficients A_i of $P(\lambda)$.
  - C: center of the circle.
  - R: radius of the circle.

Chapter 6 :

Conclusions and Future Work

In this work, we have explored several aspects related to invariant pairs and solvents of matrix polynomials.

After recalling some basics about matrix polynomials and their applications, and the relation with systems of ordinary differential equations, we presented a generalization of a shifting method for matrix polynomials, which we believe can be useful, for instance, when applying methods that require a particular eigenvalue distribution.

We presented some key points in the general theory and applications of the invariant pair and matrix solvent problems. Additionally, we computed new formulations for the condition number and the backward error of invariant pairs and matrix solvents. In the case of solvents, this computation generalized the existent previous work on the quadratic matrix equation. Furthermore, we explored the relationship between solvents of matrix polynomials in general and in triangularized form. This could be useful when computing matrix solvents.

In order to compute invariant pairs and matrix solvents, we studied a moment Hankel pencil method in its scalar and block versions: the method lends itself to a numeric or a hybrid symbolic-numeric application. We showed that the scalar method cannot capture some eigenvalue multiplicity structures, for which the block version is needed. Moreover, we analyzed the error for the trapezoid rule applied to the approximation of the moments via contour integrals.

When computing invariant pairs and matrix solvents, the proposed moment Hankel pencil methods, and some others direct approaches as well, may need to be refined numerically using an iterative method: here, we studied and compared two variants of Newton's method, namely line-search and Šamanskii. We tested the effectiveness and robustness of these approaches on several problems, many of them taken from the NLEVP collection.

Some points in this analysis may deserve to be explored in further detail. For instance, we would like to achieve a deeper understanding of the behavior of the block moment method presented in Chapter 4: at this time, we do not have results that tell us in advance what should be the size of the blocks, depending on the structure of multiplicity of the eigenvalues.

On the other hand, we would like to present convergence theorems for Newton's method and its variants analyzed in Chapter 5. Formal convergence results would help us decide, for instance, whether the results given by direct methods are good starting points for the subsequent refinement.

Another possible generalization involves the use of numerical quadrature to approximate the contour integrals. Here we limited ourselves for simplicity to the case when the contour is a circle, but it would be interesting to extend our implementations to different types of contour (e.g. ellipses, lines, etc). This might be more advantageous for some problems with particular eigenvalue distributions.

We would also like to work more on the design and development of effective symbolic-numeric eigenvalue algorithms based on the moment method, along with extensive numerical tests.

Chapter A :

Appendix

A.1 Kronecker Product

If A is an $m \times n$ matrix and B is a $p \times q$ matrix, then the Kronecker product $A \otimes B$ is the $mp \times nq$ block matrix (see [54], pp. 27):

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{bmatrix}.$$

A.1.1 Kronecker Product Properties

- $(A \otimes B)^T = A^T \otimes B^T$.
- $(A \otimes B)(C \otimes D) = AC \otimes BD$.
- $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, for matrices A and B nonsingular.
- $A \otimes (B \otimes C) = (A \otimes B) \otimes C$.

A.2 Vectorization

The vectorization of an $m \times n$ matrix A, denoted by $\mathrm{vec}\,(A)$, is the $mn \times 1$ column vector obtained by stacking the columns of the matrix A on top of one another (see [54], pp. 28):

$$\mathrm{vec}\,(A) = \begin{bmatrix} A(:,1) \\ \vdots \\ A(:,n) \end{bmatrix}.$$

A.2.1 Compatibility with Kronecker Products

The vectorization is frequently used together with the Kronecker product to express matrix multiplication as a linear transformation on matrices. In particular:

$$\mathrm{vec}\,(ABC) = (C^T \otimes A)\,\mathrm{vec}\,(B) \tag{A.1}$$

A.3 Vector and Matrix Norm

A.3.1 Vector Norm

A general vector norm on $\mathbb{K}^n$ is a function $f : \mathbb{K}^n \to \mathbb{R}$ that satisfies the following properties (see [54], pp. 68):

- $f(x) \geq 0, \quad x \in \mathbb{K}^n, \quad (f(x) = 0, \text{ iff } x = 0).$
- $f(x+y) \leq f(x) + f(y), \quad x, y \in \mathbb{K}^n,$
- $f(\alpha x) = |\alpha| f(x), \quad \alpha \in \mathbb{K}, x \in \mathbb{K}^n.$

We denote such function with a double bar notation: $f(x) = \|x\|$. A useful class of vector norms are the $p-norms$ defined by:

$$\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}, \qquad p \geq 1.$$

The $1-$, $2-$ and $\infty-$ norms are the most important:

- $\|x\|_1 = |x_1| + \cdots + |x_n|,$
- $\|x\|_2 = (|x_1|^2 + \cdots + |x_n|^2)^{1/2} = (x^T x)^{1/2},$
- $\|x\|_\infty = \max\limits_{1 \leq i \leq n} |x_i|.$

A.3.2 Matrix Norm

$f : \mathbb{K}^{m \times n} \to \mathbb{R}$ is a matrix norm if the following properties hold (see [54], pp. 71):

- $f(A) \geq 0, \quad A \in \mathbb{K}^{m \times n}, \quad (f(A) = 0, \text{ iff } A = 0).$
- $f(A+B) \leq f(A) + f(B), \quad A, B \in \mathbb{K}^{m \times n},$
- $f(\alpha A) = |\alpha| f(A), \quad \alpha \in \mathbb{K}, A \in \mathbb{K}^{m \times n}.$

We denote such function with a double bar notation: $f(A) = \|A\|$. The most frequently used matrix norms are the Frobenius norm:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\text{trace}(A^* A)}.$$

and the $p-$norms:

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}.$$

In the case of $p = 1$ and $p = \infty$, we have:

- $\|A\|_1 = \max\limits_{1 \leq j \leq n} \sum\limits_{i=1}^{m} |a_{ij}|$, which is the maximum absolute column sum of the matrix.
- $\|A\|_\infty = \max\limits_{1 \leq i \leq n} \sum\limits_{j=1}^{n} |a_{ij}|$, which is the maximum absolute row sum of the matrix.

A.4 Pseudo-inverse of a Matrix

Given an $m \times n$ matrix A, the Moore-Penrose generalized matrix inverse is a unique $n \times m$ matrix pseudo-inverse A^+ (see [54], [110]).

The Moore-Penrose inverse satisfies:

- $AA^+A = A$,
- $A^+AA^+ = A^+$,
- $(AA^+)^* = AA^+$,
- $(A^+A)^* = A^+A$.

A.5 The Generalized Schur Decomposition

The QZ algorithm of Moler and Stewart ([105], [54]) solves the generalized eigenvalue problem of finding x and λ such that $Ax = \lambda Bx$, for $A, B \in \mathbb{K}^{n\times n}$.

Theorem 19. *If A and B are in $\mathbb{C}^{n\times n}$, then there exist unitary Q and Z such that $Q^*AZ = T$ and $Q^*BZ = S$ are upper triangular. If for some k, t_{kk} and s_{kk} are both zero, then $\sigma(A, B) = \mathbb{C}$. Otherwise:*

$$\sigma(A, B) = \left\{ \frac{t_{ii}}{s_{ii}} \ : s_{ii} \neq 0 \right\}.$$

A.6 Composite Trapezoidal Rule

An intuitive method of finding the area under the curve $y = f(x)$ over $[a, b]$ is by approximating that area with a series of trapezoids that lie above the intervals $\{[x_k, x_{k+1}]\}$ (see [28]).

Theorem 20. *Let $f \in C^2[a, b]$, $h = \frac{(b-a)}{n}$ and $x_j = a + jh$, for $j = 0, 1, \ldots, n$. There exists a $\mu \in (a, b)$ for which the Composite Trapezoidal rule for n subintervals can be written with its error term as:*

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] - \frac{b-a}{12}h^2 f''(\mu).$$

Index

Contour, 77
Number of eigenvalues inside of, 78

Invariant pair, 30
Backward error, 38
Condition number, 34
Contour integral representation, 33
Homogeneous formulation, 13
Jordan pair, 32
Minimal, 31
Minimality index, 31
Moment method, 70
Newton's method, 87
Šamanskii's technique, 93
Algorithm, 88
Line search, 89
Solution of the correction equation, 94
Simple, 32

Matrix
Balancing, 18
Fréchet derivative, 33
Hankel, 62
Kronecker product, 109
Moore-Penrose inverse, 111
Norm, 110
Toeplitz, 62
Vectorization, 109
Matrix polynomial, 12
Degree, 12
Equivalent, 15
Jordan chain, 14
Leading coefficient, 12
Linearization, 15
Monic, 12
Ordinary differential equations, 13
Rank, 12
Regular, 12
Reversal, 12
Shifting Technique, 20
Singular, 12
Smith normal form, 15
Trailing coefficient, 12
Triangularization, 51
Unimodular, 15
Matrix solvent, 43
Backward error, 47
Computation, 48
Condition number, 45
Contour integral representation, 43
Existence, 43
Infinite number of solvents, 59
Matrix p-th root, 49
Newton's method, 87
Šamanskii's technique, 93
Algorithm, 88
Line search, 91
Number of, 44
Moment method, 63
Block, 73
Moments, 63
Trapezoid rule, 79

Polynomial eigenvalue problem, 12
Balancing, 18
Generalized eigenvalue problem, 13
QZ algorithm, 111
Quadratic eigenvalue problem, 13
Scaling, 19
Standard eigenvalue problem, 13

Bibliography

[1] D. M. Abrams. *Two coupled oscillator models: the Millennium Bridge and the chimera state*. PhD thesis, Cornell University, 2006.

[2] B. Adhikari and R. Alam. On backward errors of structured polynomial eigenproblems solved by structure preserving linearizations. *Linear Algebra Appl.*, 434(9):1989–2017, 2011.

[3] B. Adhikari, R. Alam, and D. Kressner. Structured eigenvalue condition numbers and linearizations for matrix polynomials. *Linear Algebra Appl.*, 435(9):2193–2221, 2011.

[4] S. S. Ahmad and V. Mehrmann. Backward errors and pseudospectra for structured nonlinear eigenvalue problems. 2013.

[5] A. Al Jammaz. Jacobi-Davidson method for polynomial eigenvalue problems. Master's thesis, Heinrich-Heine-Universität, Germany, 2008.

[6] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, et al. *LAPACK Users' guide*, volume 9. SIAM, 1999.

[7] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura. A numerical method for polynomial eigenvalue problems using contour integral. *Jpn. J. Ind. Appl. Math.*, 27(1):73–90, 2010.

[8] Z. Bai and Y. Su. SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 26(3):640–659, 2005.

[9] J. A. Ball, I. Gohberg, L. Rodman, and Y. Z. Gohberg. *Interpolation of rational matrix functions*, volume 45. Springer, 1990.

[10] H. Bart, I. Gohberg, M. A. Kaashoek, and A. Ran. *A state space approach to canonical factorization with applications*, volume 200. Springer, 2011.

[11] B. Beckermann. *On the numerical condition of polynomial bases: estimates for the condition number of Vandermonde, Krylov and Hankel matrices*. PhD thesis, Universität Hannover, 1996.

[12] B. Beckermann. The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numer. Math.*, 85(4):553–577, 2000.

[13] B. Beckermann, G. Golub, and G. Labahn. On the numerical condition of a generalized Hankel eigenvalue problem. *Numer. Math.*, 106(1):41–68, 2007.

[14] P. Bennerý, H. Fassbender, and M. Stollü. Solving large-scale quadratic eigenvalue problems with Hamiltonian eigenstructure using a structure-preserving Krylov subspace method. *Electron. Trans. Numer. Anal.*, 29:212–229, 2008.

[15] T. Betcke. Optimal scaling of generalized and polynomial eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 30(4):1320–1338, 2008.

[16] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. NLEVP: A collection of nonlinear eigenvalue problems. *ACM T. Math. Software (TOMS)*, 39(2):7, 2013.

[17] T. Betcke and D. Kressner. Perturbation, extraction and refinement of invariant pairs for matrix polynomials. *Linear Algebra Appl.*, 435(3):514–536, 2011.

[18] W. J. Beyn. An integral method for solving nonlinear eigenvalue problems. *Linear Algebra Appl.*, 436(10):3839–3863, 2012.

[19] W.-J. Beyn, C. Effenberger, and D. Kressner. Continuation of eigenvalues and invariant pairs for parameterized nonlinear eigenvalue problems. *Numer. Math.*, 119(3):489–516, 2011.

[20] W. J. Beyn and V. Thümmler. Continuation of invariant subspaces for parameterized quadratic eigenvalue problems. *SIAM J. Matrix Anal. A.*, 31(3):1361–1381, 2009.

[21] D. A. Bini, B. Meini, and F. Poloni. Transforming algebraic Riccati equations into unilateral quadratic matrix equations. *Numer. Math.*, 116(4):553–578, 2010.

[22] D. A. Bini and V. Noferini. Solving polynomial eigenvalue problems by means of the Ehrlich-Aberth method. *Linear Algebra Appl.*, 439(4):1130–1149, 2013.

[23] D. A. Bini, V. Noferini, and M. Sharify. Locating the eigenvalues of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 34(4):1708–1727, 2013.

[24] Å. Björck and S. Hammarling. A Schur method for the square root of a matrix. *Linear algebra and its applications*, 52:127–140, 1983.

[25] D. Boley, F. Luk, and D. Vandevoorde. A general Vandermonde factorization of a Hankel matrix. In *Int'l Lin. Alg. Soc.(ILAS) Symp. on Fast Algorithms for Control, Signals and Image Processing*, 1997.

[26] I. Brás and T. de Lima. A spectral approach to polynomial matrices solvents. *Appl. Math. Lett.*, 9(4):27–33, 1996.

[27] L. Brugnano and D. Trigiante. *Solving Differential Equations by Multistep Initial and Boundary Value Methods*. CRC Press, 1998.

[28] R. L. Burden and J. Faires. Numerical analysis, 2004.

[29] R. Byers, V. Mehrmann, and H. Xu. Trimmed linearizations for structured matrix polynomials. *Linear Algebra Appl.*, 429(10):2373–2400, 2008.

[30] S. H. Cheng, N. J. Higham, C. S. Kenney, and A. J. Laub. Approximating the logarithm of a matrix to specified accuracy. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1112–1125, 2001.

[31] R. M. Corless. On a generalized companion matrix pencil for matrix polynomials expressed in the Lagrange basis. In *Symbolic-Numeric Computation*, pages 1–15. Springer, 2007.

[32] G. Cross and P. Lancaster. Square roots of complex matrices. *Linear and Multilinear Algebra*, 1(4):289–293, 1974.

[33] B. Datta. Finite element model updating and partial eigenvalue assignment in structural dynamics: recent developments on computational methods. In *Proceedings: 10th International Conference "Mathematical Modelling and Analysis*, pages 15–27, 2005.

[34] G. J. Davis. Numerical solution of a quadratic matrix equation. *SIAM J. Sci. Stat. Comp.*, 2(2):164–175, 1981.

[35] F. De Terán, F. M. Dopico, and D. Mackey. Linearizations of matrix polynomials: Sharp lower bounds for the dimension and structures. In *Actas del XXI Congreso de Ecuaciones Diferenciales y Aplicaciones/XI Congreso de Matemática Aplicada, held at Ciudad Real*, pages 21–25, 2009.

[36] F. De Terán, F. M. Dopico, and D. S. Mackey. Palindromic companion forms for matrix polynomials of odd degree. *J. Comput. Appl. Math.*, 236(6):1464–1480, 2011.

[37] J. Dennis, Jr, J. F. Traub, and R. Weber. Algorithms for solvents of matrix polynomials. *SIAM J. Numer. Anal.*, 15(3):523–533, 1978.

[38] J. E. Dennis, Jr, J. F. Traub, and R. Weber. The algebraic theory of matrix polynomials. *SIAM J. Numer. Anal.*, 13(6):831–845, 1976.

[39] P. Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer Science & Business Media, 2011.

[40] B. Eckhardt, E. Ott, S. H. Strogatz, D. M. Abrams, and A. McRobie. Modeling walker synchronization on the Millennium Bridge. *Phys. Rev. E*, 75(2):021110, 2007.

[41] A. Edelman and H. Murakami. Polynomial roots from companion matrix eigenvalues. *Math. Comp.*, 64(210):763–776, 1995.

[42] C. Effenberger. *Robust solution methods for nonlinear eigenvalue problems.* PhD thesis, École polytechnique fédérale de Lausanne, 2013.

[43] C. Effenberger. Robust successive computation of eigenpairs for nonlinear eigenvalue problems. *SIAM J. Matrix Anal. A.*, 34(3):1231–1256, 2013.

[44] C. Effenberger and D. Kressner. Chebyshev interpolation for nonlinear eigenvalue problems. *BIT Numer. Math.*, 52(4):933–951, 2012.

[45] H.-Y. Fan, W.-W. Lin, and P. Van Dooren. Normwise scaling of second order polynomial matrices. *SIAM J. Matrix Anal. Appl.*, 26(1):252–256, 2004.

[46] Y. Futamura, Y. Maeda, and T. Sakurai. Stochastic estimation method of eigenvalue density for nonlinear eigenvalue problem on the complex plane. *JSIAM letters*, 3:61–64, 2011.

[47] Y. Futamura, H. Tadano, and T. Sakurai. Parallel stochastic estimation method of eigenvalue distribution. *JSIAM Letters*, 2:127–130, 2010.

[48] F. Gantmacher. Theory of matrices I, II. *New York*, 1960.

[49] L. Gemignani and V. Noferini. The Ehrlich-Aberth method for palindromic matrix polynomials represented in the Dickson basis. *Linear Algebra Appl.*, 438(4):1645–1666, 2013.

[50] I. Gohberg, M. Kaashoek, and P. Lancaster. General theory of regular matrix polynomials and band Toeplitz operators. *Integr. Equat. Oper. Th.*, 11(6):776–882, 1988.

[51] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix polynomials.* Academic Press, New York, 1982.

[52] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix polynomials*, volume 58. SIAM, 2009.

[53] G. H. Golub and G. Meurant. *Matrices, moments and quadrature with applications.* Princeton University Press, 2009.

[54] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 4. JHU Press, 2013.

[55] C.-H. Guo. On Newton's method and Halley's method for the principal *pth* root of a matrix. *Linear algebra and its applications*, 432(8):1905–1922, 2010.

[56] C.-H. Guo and N. J. Higham. A Schur-Newton method for the matrix p-th root and its inverse. *SIAM J. Matrix Anal. Appl.*, 28(3):788–804, 2006.

[57] S. Hammarling, C. J. Munro, and F. Tisseur. An algorithm for the complete solution of quadratic eigenvalue problems. *ACM Trans. Math. Softw.*, 39(3):18, 2013.

[58] B. Hashemi and M. Dehghan. Efficient computation of enclosures for the exact solvents of a quadratic matrix equation. *Electron. J. Linear Algebra*, 20:519–536, 2010.

[59] N. J. Higham. Newton's method for the matrix square root. *Mathematics of Computation*, 46(174):537–549, 1986.

[60] N. J. Higham. Computing real square roots of a real matrix. *Linear Algebra and its applications*, 88:405–430, 1987.

[61] N. J. Higham. Stable iterations for the matrix square root. *Numerical Algorithms*, 15(2):227–242, 1997.

[62] N. J. Higham and H.-M. Kim. Numerical analysis of a quadratic matrix equation. *IMA J. Numer. Anal.*, 20(4):499–519, 2000.

[63] N. J. Higham and H.-M. Kim. Solving a quadratic matrix equation by Newton's method with exact line searches. *SIAM J. Numer. Anal. A.*, 23(2):303–316, 2001.

[64] N. J. Higham, R.-C. Li, and F. Tisseur. Backward error of polynomial eigenproblems solved by linearization. *SIAM J. Matrix Anal. Appl.*, 29(4):1218–1241, 2007.

[65] N. J. Higham and L. Lin. On *pth* roots of stochastic matrices. *Linear Algebra and its Applications*, 435(3):448–463, 2011.

[66] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur. Symmetric linearizations for matrix polynomials. *SIAM J. Matrix Anal. A.*, 29(1):143–159, 2006.

[67] N. J. Higham, D. S. Mackey, and F. Tisseur. The conditioning of linearizations of matrix polynomials. *SIAM J. Matrix Anal. A.*, 28(4):1005–1028, 2006.

[68] N. J. Higham, D. S. Mackey, and F. Tisseur. Definite matrix polynomials and their linearization by definite pencils. *SIAM J. Matrix Anal. Appl.*, 31(2):478–502, 2009.

[69] 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(3):344–360, 2008.

[70] N. J. Higham and F. Tisseur. Bounds for eigenvalues of matrix polynomials. *Linear Algebra Appl.*, 358(1):5–22, 2003.

[71] A. Hilliges, C. Mehl, and V. Mehrmann. On the solution of palindromic eigenvalue problems. In *Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS). Jyväskylä, Finland*, 2004.

[72] A. S. Hodel. Computation of system zeros with balancing. *Linear Algebra Appl.*, 188:423–436, 1993.

[73] L. Hoffnung, R.-C. Li, and Q. Ye. Krylov type subspace methods for matrix polynomials. *Linear Algebra Appl.*, 415(1):52–81, 2006.

[74] U. B. Holz, G. H. Golub, and K. H. Law. A subspace approximation method for the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 26(2):498–521, 2004.

[75] B. Iannazzo. A note on computing the matrix square root. *Calcolo*, 40(4):273–283, 2003.

[76] B. Iannazzo. On the Newton method for the matrix *pth* root. *SIAM journal on matrix analysis and applications*, 28(2):503–523, 2006.

[77] I. C. Ipsen. Accurate eigenvalues for fast trains. *SIAM News*, 37(9):1–2, 2004.

[78] E. Jarlebring, K. Meerbergen, and W. Michiels. Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 35(2):411–436, 2014.

[79] H. B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. *Applications of bifurcation theory*, pages 359–384, 1977.

[80] H.-M. Kim. *Numerical methods for solving a quadratic matrix equation.* PhD thesis, Manchester University, 2000.

[81] W. Kratz and E. Stickel. Numerical solution of matrix polynomial equations by Newton's method. *IMA J. Numer. Anal.*, 7(3):355–369, 1987.

[82] D. Kressner. A block newton method for nonlinear eigenvalue problems. *Numer. Math.*, 114(2):355–372, 2009.

[83] D. Kressner and J. E. Roman. Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis. *Numer. Linear Algebra Appl.*, 21(4):569–588, 2014.

[84] P. Lancaster. *Lambda-matrices and vibrating systems*. Courier Dover Publications, 2002.

[85] A. Leblanc and A. Lavie. Solving acoustic nonlinear eigenvalue problems with a contour integral method. *Eng. Anal. Bound. Elem.*, 37(1):162–166, 2013.

[86] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. SIAM, 1998.

[87] D. Lemonnier and P. Van Dooren. Balancing regular matrix pencils. *SIAM J. Matrix Anal. Appl.*, 28(1):253–263, 2006.

[88] J.-H. Long, X.-Y. Hu, and L. Zhang. Improved Newton's method with exact line searches to solve quadratic matrix equation. *J. Comput. Appl. Math.*, 222(2):645–654, 2008.

[89] J. H. Macdonald. Lateral excitation of bridges by balancing pedestrians. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, pages 1055–1073. The Royal Society, 2008.

[90] D. S. Mackey. *Structured linearizations for matrix polynomials*. PhD thesis, University of Manchester, 2006.

[91] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Palindromic polynomial eigenvalue problems: Good vibrations from good linearizations. In *DFG Research Center Matheon, Mathematics for*, 2005.

[92] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Structured polynomial eigenvalue problems: Good vibrations from good linearizations. *SIAM J. Matrix Anal. A.*, 28(4):1029–1051, 2006.

[93] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *SIAM J. Matrix Anal. A.*, 28(4):971–1004, 2006.

[94] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Smith forms of palindromic matrix polynomials. *Electron. J. Linear Algebra*, 22(1):4, 2011.

[95] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Skew-symmetric matrix polynomials and their smith forms. *Linear Algebra Appl.*, 438(12):4625–4653, 2013.

[96] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Möbius transformations of matrix polynomials. *Linear Algebra Appl.*, 2014.

[97] D. S. Mackey and V. Perovic. Linearizations of matrix polynomials in Bernstein basis. 2014.

[98] V. Mehrmann and D. Watkins. Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Sci. Comput.*, 22(6):1905–1925, 2001.

[99] V. Mehrmann and D. Watkins. Polynomial eigenvalue problems with Hamiltonian structure. *Electron. Trans. Numer. Anal*, 13:106–118, 2002.

[100] B. Meini. The matrix square root from a new functional perspective: theoretical results and computational issues. *SIAM journal on matrix analysis and applications*, 26(2):362–376, 2004.

[101] B. Meini. A "shift-and-deflate" technique for quadratic matrix polynomials. *Linear Algebra Appl.*, 438(4):1946–1961, 2013.

[102] A. Melman. Generalization and variations of Pellet's theorem for matrix polynomials. *Linear Algebra Appl.*, 439(5):1550–1567, 2013.

[103] G. Meurant and G. Golub. *Matrices, moments and quadrature with applications*. Princeton University Press, 2010.

[104] C. D. Meyer, Jr. Generalized inverses of block triangular matrices. *SIAM J. Math. Appl.*, 19(4):741–750, 1970.

[105] C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM J. Numer. Anal.*, 10(2):241–256, 1973.

[106] I. Newton. Methodus fluxionum et serierum infinitarum, 1664-1671.

[107] V. Noferini, M. Sharify, and F. Tisseur. Tropical roots as approximations to eigenvalues of matrix polynomials. 2014.

[108] E. Osborne. On pre-conditioning of matrices. *Journal of the ACM (JACM)*, 7(4):338–345, 1960.

[109] B. N. Parlett and C. Reinsch. Balancing a matrix for calculation of eigenvalues and eigenvectors. *Numer. Math.*, 13(4):293–304, 1969.

[110] R. Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge Univ Press, 1955.

[111] E. Pereira. On solvents of matrix polynomials. *Appl. Numer. Math.*, 47(2):197–208, 2003.

[112] E. Polizzi. Density-matrix-based algorithm for solving eigenvalue problems. *Phys. Rev. B*, 79(11):115112, 2009.

[113] J. Raphson. *Analysis aequationum universalis*. typis T.B. prostant venales apud A. & I. Churchill, 1702.

[114] T. Sakurai, Y. Futamura, and H. Tadano. Efficient parameter estimation and implementation of a contour integral-based eigensolver. *J. Algorithms Comput. Technol.*, 7(3):249–270, 2013.

[115] T. Sakurai and H. Sugiura. A projection method for generalized eigenvalue problems using numerical integration. *J. Comput. Appl. Math.*, 159(1):119–128, 2003.

[116] G. L. Sleijpen, A. G. Booten, D. R. Fokkema, and H. A. Van der Vorst. Jacobi-davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT Numer. Math.*, 36(3):595–633, 1996.

[117] G. L. Sleijpen, H. A. Van der Vorst, and M. v. Gijzen. Quadratic eigenproblems are no problem. *SIAM News*, 29(7):8–9, 2001.

[118] M. I. Smith. A Schur algorithm for computing matrix *pth* roots. *SIAM journal on matrix analysis and applications*, 24(4):971–989, 2003.

[119] J. Sylvester. On Hamilton's quadratic equation and the general unilateral equation in matrices. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 18(114):454–458, 1884.

[120] D. B. Szyld and F. Xue. Several properties of invariant pairs of nonlinear algebraic eigenvalue problems. *IMA J. Numer. Anal.*, page drt026, 2013.

[121] L. Taslaman, F. Tisseur, and I. Zaballa. Triangularizing matrix polynomials. *Linear Algebra Appl.*, 439(7):1679–1699, 2013.

[122] F. Tisseur. Backward error and condition of polynomial eigenvalue problems. *Linear Algebra Appl.*, 309(1):339–361, 2000.

[123] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Rev.*, 43(2):235–286, 2001.

[124] F. Tisseur and I. Zaballa. Triangularizing quadratic matrix polynomials. *SIAM J. Matrix Anal. A.*, 34(2):312–337, 2013.

[125] M. B. Van Gijzen. The parallel computation of the smallest eigenpair of an acoustic problem with damping. *Int. J. Numer. Meth. Eng.*, 45(6):765–777, 1999.

[126] V. Šamanskii. On a modification of the Newton method. *Ukrain. Mat.*, (19):133–138, 1967.

[127] R. C. Ward. Balancing the generalized eigenvalue problem. *SIAM J. Sci. Comput.*, 2(2):141–152, 1981.

[128] D. S. Watkins. A case where balancing is harmful. *Electron. Trans. Numer. Anal*, 23:1–4, 2006.

[129] S. Wright and J. Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

Résumé : Cette thèse porte sur certains aspects symboliques-numériques du problème des paires invariantes pour les polynômes de matrices. Les paires invariantes généralisent la définition de valeur propre / vecteur propre et correspondent à la notion de sous-espaces invariants pour le cas nonlinéaire. Elles trouvent leurs applications dans le calcul numérique de plusieurs valeurs propres d'un polynôme de matrices ; elles présentent aussi un intérêt dans le contexte des systèmes différentiels.

En utilisant une approche basée sur les intégrales de contour, nous déterminons des expressions du nombre de conditionnement et de l'erreur rétrograde pour le problème du calcul des paires invariantes. Ensuite, nous adaptons la méthode des moments de Sakurai-Sugiura au calcul des paires invariantes et nous étudions le comportement de la version scalaire et par blocs de la méthode en présence de valeurs propres multiples. Le résultats obtenus à l'aide des approches directes peuvent éventuellement être améliorés numériquement grâce à une méthode itérative : nous proposons ici une comparaison de deux variantes de la méthode de Newton appliquée aux paires invariantes.

Le problème des solvants de matrices est très proche de celui des paires invariants. Le résultats présentés ci-dessus sont donc appliqués au cas des solvants pour obtenir des expressions du nombre de conditionnement et de l'erreur, et un algorithme de calcul basé sur la méthode des moments. De plus, nous étudions le lien entre le problème des solvants et la transformation des polynômes de matrices en forme triangulaire.

Mots clés : Polynômes de matrices, paires invariantes, solvants, intégrale de contour, nombre de conditionnement, erreur rétrograde, faisceaux de matrices de Hankel, méthode de Newton.

Abstract: In this thesis, we study some symbolic-numeric aspects of the invariant pair problem for matrix polynomials. Invariant pairs extend the notion of eigenvalue-eigenvector pairs, providing a counterpart of invariant subspaces for the nonlinear case. They have applications in the numeric computation of several eigenvalues of a matrix polynomial; they also present an interest in the context of differential systems.

Here, a contour integral formulation is applied to compute condition numbers and backward errors for invariant pairs. We then adapt the Sakurai-Sugiura moment method to the computation of invariant pairs, including some classes of problems that have multiple eigenvalues, and we analyze the behavior of the scalar and block versions of the method in presence of different multiplicity patterns. Results obtained via direct approaches may need to be refined numerically using an iterative method: here we study and compare two variants of Newton's method applied to the invariant pair problem.

The matrix solvent problem is closely related to invariant pairs. Therefore, we specialize our results on invariant pairs to the case of matrix solvents, thus obtaining formulations for the condition number and backward errors, and a moment-based computational approach. Furthermore, we investigate the relation between the matrix solvent problem and the triangularization of matrix polynomials.

Keywords: Matrix polynomials, invariant pairs, solvents, contour integral, condition number, backward error, Hankel pencils, Newton's method.

LABO - UMR CNRS n° 7252
123, avenue Albert Thomas - 87060 LIMOGES