8-23-2019 9:30 AM

Algebraic Companions and Linearizations

Eunice Y. S. Chan, *The University of Western Ontario*

Supervisor: Corless, Robert M., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Applied Mathematics

Abstract

In this thesis, we look at a novel way of finding roots of a scalar polynomial using eigenvalue techniques. We extended this novel method to the polynomial eigenvalue problem (PEP). PEP have been used in many science and engineering applications such vibrations of structures, computer-aided geometric design, robotics, and machine learning. This thesis explains this idea in the order which we discovered it.

In Chapter 2, a new kind of companion matrix is introduced for scalar polynomials of the form $c(\lambda) = \lambda a(\lambda)b(\lambda) + c_0$, where upper Hessenberg companions are known for the polynomials $a(\lambda)$ and $b(\lambda)$. This construction can generate companion matrices with smaller entries than the Fiedler or Frobenius forms. This generalizes Piers Lawrence's Mandelbrot companion matrix. The construction was motivated by use of Narayana-Mandelbrot polynomials.

In Chapter 3, we define Euclid polynomials $E_{k+1}(\lambda) = E_k(\lambda)(E_k(\lambda) - 1) + 1$ where $E_1(\lambda) = \lambda + 1$ in analogy to Euclid numbers $e_k = E_k(1)$. We show how to construct companion matrices E_k, so $E_k(\lambda) = \det(\lambda I - E_k)$ is of height 1 (and thus of minimal height over all integer companion matrices for $E_k(\lambda)$). We prove various properties of these objects, and give experimental confirmation of some unproved properties.

In Chapter 4, we show how to construct linearizations of matrix polynomials $z\boldsymbol{a}(z)\boldsymbol{d}_0 + \boldsymbol{c}_0$, $\boldsymbol{a}(z)\boldsymbol{b}(z)$, $\boldsymbol{a}(z) + \boldsymbol{b}(z)$ (when $\deg(\boldsymbol{b}(z)) < \deg(\boldsymbol{a}(z))$), and $z\boldsymbol{a}(z)\boldsymbol{d}_0\boldsymbol{b}(z) + \boldsymbol{c}_0$ from linearizations of the component parts, matrix polynomials $\boldsymbol{a}(z)$ and $\boldsymbol{b}(z)$. This extends the new companion matrix construction introduced in Chapter 2 to matrix polynomials.

In Chapter 5, we define "generalized standard triples" which can be used in constructing algebraic linearizations; for example, for $\boldsymbol{H}(z) = z\boldsymbol{a}(z)\boldsymbol{b}(z) + \boldsymbol{c}_0$ from linearizations for $\boldsymbol{a}(z)$ and $\boldsymbol{b}(z)$. For convenience, we tabulate generalized standard triples for orthogonal polynomial bases, the monomial basis, and Newton interpolational bases; for the Bernstein basis; for Lagrange interpolational bases; and for Hermite interpolational bases. We account for the possibility of common similarity transformations. We give explicit proofs for the less familiar bases.

In Chapter 6, we investigate the numerical stability of algebraic linearization, which re-uses linearizations of matrix polynomials $\boldsymbol{a}(\lambda)$ and $\boldsymbol{b}(\lambda)$ to make a linearization for the matrix polynomial $\boldsymbol{P}(\lambda) = \lambda\boldsymbol{a}(\lambda)\boldsymbol{b}(\lambda) + \boldsymbol{c}$. Such a re-use *seems* more likely to produce a well-conditioned linearization, and thus the implied algorithm for finding the eigenvalues of $\boldsymbol{P}(\lambda)$ seems likely to be more numerically stable than expanding out the product $\boldsymbol{a}(\lambda)\boldsymbol{b}(\lambda)$ (in whatever polynomial basis one is using). We investigate this question experimentally by using pseudospectra.

Keywords: matrix, polynomial, matrix polynomial, linearizations, pseudospectra, Bohemian, companion matrix, comrade matrix, eigenvalues

Summary for lay audience

Matrix polynomials have been used in many scientific and engineering applications, ranging from vibrations of structures including buildings and aircrafts to machine learning. This thesis studies new and effective methods for solving design problems with such models. This thesis provides new theory and new algorithms. We introduce several new ideas including minimal height companion matrices which promise more numerically stable algorithms. A numerically stable algorithm gives exact solutions to a nearby problem. This thesis extends this result to the case where the polynomial coefficients are matrices themselves i.e. matrix polynomials.

Co-Authorship Statement

This integrated-article thesis is based on 5 papers. A version of Chapter 2 [1] has been published; in this thesis, an addition has been made at the end of section 4.3. Chapters 3 [2], and 4 [3] have been published in academic journals. Chapter 5 has been submitted for publication. Chapter 6 is being prepared for publication. For these papers, Robert Corless provided assistance with the theoretical understanding of companion matrices and linearizations, and provided feedback on the papers. The initial basis for chapter 4 was developed by Robert Corless, Laureano Gonzalez-Vega, J. Rafael Sendra, and Juana Sendra. Rob provided assistance with the main proofs in Section 4.3. Leili Rafiee Sevyeri provided assistance with the Bernstein basis section of Chapter 5.

Bibliography

[1] E. Y. S. Chan and R. M. Corless, *A new kind of companion matrix*, Electronic Journal of Linear Algebra, 32 (2017), pp. 335–342.

[2] ——, *Minimal height companion matrices for Euclid polynomials*, Mathematics in Computer Science, 13 (2019), pp. 41–56.

[3] E. Y. S. Chan, R. M. Corless, L. Gonzalez-Vega, J. R. Sendra, and J. Sendra, *Algebraic linearizations for matrix polynomials*, Linear Algebra and its Applications, 563 (2019), pp. 373–399.

Acknowledgements

I am very grateful for the support that I have received from many people during my graduate studies. Below are the people that I am especially grateful for and without them, this thesis may not have been possible.

First, without my supervisor Robert Corless, this thesis most definitely would not have been possible. I would like to thank Rob for giving me the opportunity to learn and grow as a researcher from him—I would not have become the researcher I am today without his guidance. I also would like to thank Rob for his generosity, time, and encouragement while under his supervision.

Secondly, I would like to thank my parents, Mei Po Wong and Charm Fung Chan, and my brother, Sunny Chan for their constant support, especially through graduate school.

Thirdly, I would like to thank Janet Martin for allowing me the opportunity to apply my mathematical and programming knowledge to the field of public health.

Fourthly, I would like to thank my Spanish colleague and co-authors Laureano Gonzalez-Vega, J. Rafael Sendra, and Juana Sendra for their tremendous generosity during my visit in Spain during my last semester of my PhD.

Lastly, I would like to thank the professors and staff in the Department of Applied Mathematics for their help during my graduate studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For this thesis, we are interested in finding solutions to univariate scalar and matrix polynomial problems numerically. To solve for the zeros in the scalar case, we convert the polynomial problem into an eigenvalue problem, where we build a matrix whose eigenvalues are nearly identical to the solution of the polynomial problem. Such a matrix is called a companion matrix. Rather than using commonly-used constructions such as Frobenius [1] and Fiedler [2], we introduce a new construction for the companion matrix, first thought of by Piers W. Lawrence for the Mandelbrot matrices. We generalized his construction for polynomials in the form

$$p(z) = za_0(z) \cdot a_1(z) \cdots a_n(z) + c \,,$$

where n is an integer greater than 0, $a_i(z)$ are scalar polynomials with $\deg(a_i(z)) > 0$, and $c \in \mathbb{C}$. We then extended this construction to the matrix polynomials that were in the form

$$\boldsymbol{P}(\lambda) = \lambda \boldsymbol{a}(\lambda)\boldsymbol{b}(\lambda) + \boldsymbol{c} \,, \tag{1.1}$$

where $\boldsymbol{a}(\lambda)$ and $\boldsymbol{b}(\lambda)$ are matrix polynomials expressed in any polynomial basis with their coefficients of the same dimension $\mathbb{C}^{r\times r}$, and $\boldsymbol{c} \in \mathbb{C}^{r\times r}$. In this case, instead of calling the matrix a companion, we use the term linearization as it is conventional.

This chapter breaks down the scalar polynomial problem and the polynomial eigenvalue problem, separately, and then gives an outline of this thesis.

1.1 Finding solutions to a scalar polynomial

Polynomial system models are used extremely frequently in mathematics, engineering, and science. These systems have been studied extensively, such computer aided geometric design by Sederberg [7] and robotics by Sommese and Wampler [8]. In the first part of this thesis, we focus on univariate scalar polynomials.

A scalar polynomial expressed in the monomial basis is defined by

$$A(z) = \sum_{i=0}^{n} z^i a_i \,, \quad a_i \in \mathbb{C} \,, \quad a_n \neq 0 \,.$$

The article [5] surveys efficient methods to solve these polynomials, ranging from iterative methods to multi-point methods to methods based on rational approximation and more. One

method that we are particularly interested in is to solve for the roots of a scalar polynomial is to convert the problem into an eigenvalue problem. The most popular construction is the Frobenius companion. The generalized companion matrix construction is

$$\boldsymbol{M}_1(z) = z\begin{bmatrix} a_n & 0 & \cdots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} - \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & -a_0 \\ 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{1.2}$$

where $\det(\boldsymbol{M}_1(z)) = A(z)$. The permutations of equation (1.2),

$$\boldsymbol{M}_2(z) = z\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & a_n \end{bmatrix} - \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & 0 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix}$$

$$\boldsymbol{M}_3(z) = z\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & a_n \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & -a_0 \\ 1 & \ddots & \vdots & -a_1 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & -a_{n-1} \end{bmatrix}$$

$$\boldsymbol{M}_4(z) = z\begin{bmatrix} a_n & 0 & \cdots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} - \begin{bmatrix} -a_{n-1} & 1 & 0 & 0 \\ -a_{n-2} & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 \\ -a_0 & 0 & \cdots & 0 \end{bmatrix}$$

are also commonly used. Although the Frobenius construction is the most popular, there are many other companion matrix constructions, such as the Fiedler construction and the construction that we will introduce in this thesis. Companion matrices are sometimes referred to as comrade matrices or colleague matrices.

The polynomial that we are interested in is in the recursive form

$$p(z) = za(z)b(z) + c\,.$$

Our construction (Theorem 2.2.1) takes advantage of this form: if we know the companions for $a(z)$ and $b(z)$, we can simply "glue" everything together, creating the companion for $p(z)$.

We can show through a small example why it may be advantageous to use our companion matrix construction rather than the Frobenius companion matrix construction. Let us take

$$p(z) = z(z+1)(z+1) + 1 \tag{1.3}$$

$$= z^3 + 2z^2 + z + 1\,. \tag{1.4}$$

Using Maple 2017, we can find the roots exactly:

$$-\frac{1}{6}\sqrt[3]{100+12\sqrt{69}}-\frac{2}{3}\frac{1}{\sqrt[3]{100+12\sqrt{69}}}-\frac{2}{3},$$

$$\frac{1}{12}\sqrt[3]{100+12\sqrt{69}}+\frac{1}{3}\frac{1}{\sqrt[3]{100+12\sqrt{69}}}-\frac{2}{3}+\frac{i}{2}\sqrt{3}\left(-\frac{1}{6}\sqrt[3]{100+12\sqrt{69}}+\frac{2}{3}\frac{1}{\sqrt[3]{100+12\sqrt{69}}}\right),$$

$$\frac{1}{12}\sqrt[3]{100+12\sqrt{69}}+\frac{1}{3}\frac{1}{\sqrt[3]{100+12\sqrt{69}}}-\frac{2}{3}-\frac{i}{2}\sqrt{3}\left(-\frac{1}{6}\sqrt[3]{100+12\sqrt{69}}+\frac{2}{3}\frac{1}{\sqrt[3]{100+12\sqrt{69}}}\right),$$

which evaluated using 20 digit precision are

$$\begin{aligned}&-1.7548776662466927601\,,\\ &-0.12256116687665361998-0.74486176661974423660i\,,\\ &-0.12256116687665361998+0.74486176661974423660i\,,\end{aligned}$$

respectively.

The Frobenius companion matrix for this example would be

$$z\boldsymbol{I}_3-\begin{bmatrix}-2&-1&-1\\1&0&0\\0&1&0\end{bmatrix}. \tag{1.5}$$

The characteristic polynomial of equation (1.5) is

$$\begin{aligned}\begin{vmatrix}z+2&1&1\\-1&z&0\\0&-1&z\end{vmatrix}&=(z+2)\begin{vmatrix}z&0\\-1&z\end{vmatrix}+\begin{vmatrix}1&1\\-1&z\end{vmatrix}\\&=z^2(z+2)+z+1\\&=z^3+2z^2+z+1\,,\end{aligned}$$

which is equivalent to equation (1.4), which means that the eigenvalues of equation (1.5) are the solutions to equation (1.4) if computed exactly. Using Matlab R2017a's `eig` routine, the eigenvalues of equation (1.5) are

$$\begin{aligned}&-1.754877666246694+0.000000000000000i\\&-0.122561166876654+0.744861766619744i\\&-0.122561166876654-0.744861766619744i\,.\end{aligned}$$

On the other hand, our companion matrix for this example would be

$$z\boldsymbol{I}_3-\begin{bmatrix}-1&0&-1\\-1&0&0\\0&-1&-1\end{bmatrix}. \tag{1.6}$$

The characteristic polynomial of (1.6) is

$$\begin{aligned}\begin{vmatrix} z+1 & 0 & 1 \\ 1 & z & 0 \\ 0 & 1 & z+1 \end{vmatrix} &= (z+1)\begin{vmatrix} z & 0 \\ 1 & z+1 \end{vmatrix} - \begin{vmatrix} 0 & 1 \\ 1 & z+1 \end{vmatrix} \\ &= (z+1)z(z+1)+1 \\ &= z(z+1)^2+1 \\ &= z^3+2z^2+z+1\,,\end{aligned}$$

which is also equivalent to equation (1.4) meaning that equation (1.6) is also a companion of (1.4). Using MATLAB's `eig` routine, the eigenvalues of equation (1.6) are

$$\begin{aligned}&-1.754877666246691 + 0.000000000000000i \\ &-0.122561166876653 + 0.744861766619744i \\ &-0.122561166876653 - 0.744861766619744i\,.\end{aligned}$$

We can see that the eigenvalues of both companions are slightly different than the solution evaluated by Maple. This demonstrates the idea of "solving the exact solution to a nearby problem", thus resulting in nearly identical results. On another note, even though this is a small example, we can see that equation (1.6) has lower height compared to equation (1.5) since our construction does not require the multiplication between coefficients. Lower height becomes more important for larger problems by offering smaller condition number of the eigenvalues, which suggests better backward stability. We demonstrate this by using pseudospectra in chapters 3 and 6.

1.2 Polynomial eigenvalue problem

For the matrix polynomial case, polynomial eigenvalue problems (PEP) arise in many applications. In Mackey et al. [4], the authors mention applications in extreme designs, such as high speed trains, optoelectronic devices, micro-electromechanical systems and "superjumbo" jets such as the Airbus 380. This particular application presents a challenge for the computation of the resonant frequencies of these structures since these extreme designs often lead to eigenproblems with poor conditioning. Other applications of PEP can be found in survey articles such as Güttel and Tisseur [3] and Mehrmann and Voss [6].

The polynomial eigenvalue problem is to find scalars λ and nonzero vectors x and y satisfying $\boldsymbol{P}(\lambda)\boldsymbol{x} = 0$ and $\boldsymbol{y}^*\boldsymbol{P}(\lambda) = 0$ where

$$\boldsymbol{P}(\lambda) = \sum_{i=0}^{m} \phi_i(\lambda)\boldsymbol{A}_i\,, \quad \boldsymbol{A}_i \in \mathbb{C}^{n\times n}\,, \quad \boldsymbol{A}_m \neq 0$$

is a matrix polynomial of degree m and $\phi_i(\lambda)$ is any polynomial basis. If $\det \boldsymbol{P}(\lambda)$ is not a zero polynomial, then the matrix polynomial is regular. For matrix polynomials expressed in the power basis, when $\boldsymbol{A}_m$ is the identity matrix, the matrix polynomial is classified as a monic $n\times n$ matrix polynomial. On the other hand, if the leading matrix coefficient is not necessarily the

identity matrix or even invertible, but not identically equal to zero, then the matrix polynomial is classified as a non-monic $n \times n$ matrix polynomial. Chapters 4 to 6 consider these cases in various polynomial bases. Chapter 6 also gives an example where the leading coefficient of the matrix polynomial is singular.

The standard way of solving this problem is by *linearization*, where we convert the matrix polynomial $\boldsymbol{P}(\lambda)$ into a linear polynomial

$$\boldsymbol{L}(\lambda) = \lambda \boldsymbol{C}_1 + \boldsymbol{C}_0$$

where $\boldsymbol{C}_0,\ \boldsymbol{C}_1 \in \mathbb{C}^{mn \times mn}$, whose eigenvalues are nearly identical to the solution of the matrix polynomial. It is guaranteed that both $\boldsymbol{P}(\lambda)$ and $\boldsymbol{L}(\lambda)$ have the same spectrum if

$$\boldsymbol{E}(\lambda)\boldsymbol{L}(\lambda)\boldsymbol{F}(\lambda) = \begin{bmatrix} \boldsymbol{P}(\lambda) & 0 \\ 0 & \boldsymbol{I}_{(m-1)n} \end{bmatrix}$$

where $\boldsymbol{E}(\lambda)$ and $\boldsymbol{F}(\lambda)$ are unimodular. Chapter 5 presents the linearizations for bases with three-term recurrence relations, the Bernstein basis, the Lagrange interpolational basis, and the Hermite interpolational basis.

For this thesis, we are interested in matrix polynomials that are in the recursive form shown in equation (1.1). In Chapter 4, we extend the construction from Theorem 2.2.1 to the matrix polynomial case, which we call *algebraic linearizations* (Theorem 4.3.5). Due to the similarly in the construction between the scalar polynomial and matrix polynomial case, we also believe that algebraic linearizations can offer better backward stability for larger problems due to having lower height. We explore this through numerical experiments in Chapter 6.

1.3 Outline

This thesis is based on five papers. Chapters 2 and 3 discuss the construction of companion matrices for scalar polynomials. Chapters 4 extends the theorems from the previous two chapters to the the matrix polynomial case. Chapter 5 provides the standard triples for various polynomial bases which is used in the construction of the linearization introduced in Chapter 5. Lastly, Chapter 6 examines the backward stability of the linearization from Chapter 4.

Chapter 2 introduces a new construction of a companion matrix for scalar polynomials in the form

$$p(z) = za(z)b(z) + c$$

such as the Mandelbrot polynomials,

$$\begin{aligned} p_0(z) &= 0 \\ p_{n+1}(z) &= zp_n^2(z) + 1 \end{aligned}$$

and the Narayana-Mandelbrot polynomials

$$\begin{aligned} q_0(z) &= 0 \\ q_1(z) &= 1 \\ q_{n+1}(z) &= zq_n(z)q_{n-1}(z) + 1 \end{aligned}$$

for $n \geq 0$. We prove that this construction is indeed a companion matrix by using Schur factoring and Laplace expansion. We also introduce the notion of height of a matrix. In this chapter, we define height as the largest element minus the smallest element in the population. We use a different definition of height in Chapter 3.

Chapter 3 generalizes the construction of companion matrix that was introduced in the previous chapter. This generalization was successfully proven to be correct from working on a test problem that was posed by Don Knuth: the Euclid polynomials

$$\begin{aligned} E_1(\lambda) &= \lambda + 1 \\ E_{n+1}(\lambda) &= \lambda E_n(\lambda) E_{n-1}(\lambda) \cdots E_1(\lambda) + 1 \end{aligned}$$

for $n > 0$. Euclid polynomials arose from the Euclid number. In the chapter, we establish some properties for Euclid numbers and Euclid polynomials. We also looked at the conditioning of the generalized companion matrix by computing the condition number for both the evaluation of the polynomial and the companion matrix and seeing the growth as the degree of the polynomial increases. We also show through using pseudospectra that using companion matrices is the best method for computing the roots of the Euclid polynomials. We introduce the term "Bohemian" matrices in this chapter.

Chapter 4 extends the theory from Chapter 2 and 3 to matrix polynomials. Using a similar construction to the scalar polynomial case, we were successful in using this construction for matrix polynomials in the form $\boldsymbol{h}(z) = z\boldsymbol{a}(z)\boldsymbol{b}(z) + \boldsymbol{c}_0$ where $\boldsymbol{a}(z)$ and $\boldsymbol{b}(z)$ are matrix polynomials and $\boldsymbol{c}_0$ is a matrix. We call this linearization construction *algebraic linearizations*. Theorem 4.3.5 is the main theorem of the chapter, and we used the Schur complement to prove that our linearization construction is indeed a linearization. We also perform a few numerical experiments at the end of the chapter. We introduce the notion of "rhapsody" in this chapter.

Chapter 5 is an addition to Chapter 3: we define generalized standard triples $\boldsymbol{X}, z\boldsymbol{C}_1 - \boldsymbol{C}_0, \boldsymbol{Y}$ of regular matrix polynomials $\boldsymbol{P}(z) \in \mathbb{C}^{n\times n}$ in order to use the representation $\boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y} = \boldsymbol{P}^{-1}(z)$ for $z \notin \Lambda(\boldsymbol{P}(z))$ (which is needed for the construction of the algebraic linearization) in most commonly used polynomial bases. At the end of the chapter, there are numerical experiments for some of the polynomial bases to demonstrate that the standard triples provided in the chapter are correct.

Chapter 6 explores the backward stability of algebraic linearizations. We note in the chapter that the standard theory for backward stability does not applied for algebraic linearizations since the matrix polynomial in its factored for is not a linear combination of the basis elements, which is required for the standard theory to hold. We also included several numerical experiments, in which we plot the pseudospectra for both the matrix polynomials and their linearizations. From the experiments, we have seen some cases where the algebraic linearization has better backward stability in comparison to commonly-used linearizations such as the frobenius linearization.

Bibliography

[1] A. Edelman and H. Murakami, *Polynomial roots from companion matrix eigenvalues*, Mathematics of Computation, 64 (1995), pp. 763–776.

[2] M. FIEDLER, *A note on companion matrices*, Linear Algebra and its Applications, 372 (2003), pp. 325–331.

[3] S. GÜTTEL AND F. TISSEUR, *The nonlinear eigenvalue problem*, Acta Numerica, 26 (2017), pp. 1–94.

[4] D. S. MACKEY, N. MACKEY, AND F. TISSEUR, *Polynomial eigenvalue problems: Theory, computation, and structure*, in Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory, Springer, 2015, pp. 319–348.

[5] J. M. MCNAMEE AND V. Y. PAN, *Efficient polynomial root-refiners: A survey and new record efficiency estimates*, Computers & Mathematics with Applications, 63 (2012), pp. 239–254.

[6] V. MEHRMANN AND H. VOSS, *Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods*, GAMM-Mitteilungen, 27 (2004), pp. 121–152.

[7] T. SEDERBERG, *Applications to computer aided geometric design*, in Proceedings of Symposia in Applied Mathematics, vol. 53, 1998, pp. 67–89.

[8] I. C. W. WAMPLER ET AL., *The Numerical solution of systems of polynomials arising in engineering and science*, World Scientific, 2005.

Chapter 2

A new kind of companion matrix

2.1 Introduction

Recently, we generalized the Mandelbrot polynomials

$$p_{n+1} = zp_n^2 + 1 \quad p_0 = 0$$

to the Fibonacci-Mandelbrot polynomials

$$q_{n+1} = zq_nq_{n-1} + 1 \quad q_0 = 0, q_1 = 1$$

and generalized Piers Lawrence's supersparse[1] companion matrix for p_n [8] to an analogous one for q_n. See [4], [5] and [7] for details, though we summarize these constructions below.

If $p_n = \det(z\mathbf{I} - \mathbf{M}_n)$ for the Mandelbrot polynomials, the subdiagonals of $\mathbf{M}_n$ are all -1 which gives

$$\mathbf{M}_{n+1} = \begin{bmatrix} \mathbf{M}_n & & -\mathbf{c}_n\mathbf{r}_n \\ -\mathbf{r}_n & 0 & \\ & -\mathbf{c}_n & \mathbf{M}_n \end{bmatrix}, \tag{2.1}$$

where $\mathbf{r}_n = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix}$ and $\mathbf{c}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ are both of length d_n, where d_n is the degree of $p_n(z)$ or the dimension of $\mathbf{M}_n$. This is Piers Lawrence's original construction [8].These are remarkable matrices: they contain only -1 or 0, and therefore are Bohemian matrices[2]; yet the characteristic polynomial contains coefficients that grow exponentially in the degree d_n (doubly exponentially in n).

For the Fibonacci-Mandelbrot polynomials, the degree of $q_n = F_n - 1$, where F_n is the nth Fibonacci number, and the construction contains matrices of different size. We begin with

$$\mathbf{M}_3 = \begin{bmatrix} -1 \end{bmatrix}$$

and

$$\mathbf{M}_4 = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$$

[1] A matrix is supersparse if it is sparse and its nonzero elements are drawn from a small set, e.g. $\{-1, 1\}$

[2] The name "Bohemian" is an acronym for Bounded height matrix of integers. See example OEIS A272658

to construct our recursive companion matrix:

$$\mathbf{M}_{n+1} = \begin{bmatrix} \mathbf{M}_n & & (-1)^{d_{n+1}}\mathbf{c}_n\mathbf{r}_{n-1} \\ -\mathbf{r}_n & 0 & \\ & -\mathbf{c}_{n-1} & \mathbf{M}_{n-1} \end{bmatrix},$$

where $\mathbf{r}_n = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$ and $\mathbf{c}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ are, as before, the row and column vectors of length d_n. This gives a matrix of slightly greater height than (2.1) because the entries may be $\{-1, 0, 1\}$.

The surprising analogy between these two families of supersparse companions led us to conjecture and prove the following.

2.2 Main result

Theorem 2.2.1 *Suppose $a(z) = \det(z\mathbf{I} - \mathbf{A})$, $b(z) = \det(z\mathbf{I} - \mathbf{B})$, and both $\mathbf{A}$ and $\mathbf{B}$ are upper Hessenberg matrices with nonzero subdiagonal entries, and*

$$\alpha = \frac{1}{\left(\prod_{j=1}^{d_a-1} a_{j+1,j}\right)\left(\prod_{j=1}^{d_b-1} b_{j+1,j}\right)}$$

is the reciprocal of the product of the subdiagonal entries of $\mathbf{A}$ and $\mathbf{B}$, and $d_a = \deg_z a$ and $d_b = \deg_z b$, so the dimension of $\mathbf{A}$ is $d_a \times d_a$ and the dimension of $\mathbf{B}$ is $d_b \times d_b$. Suppose both d_a and d_b are at least 1. Then if

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} & & -\alpha c_0 \mathbf{c_a}\mathbf{r_b} \\ -\mathbf{r_a} & 0 & \\ & -\mathbf{c_b} & \mathbf{B} \end{bmatrix}$$

where $\mathbf{r_a} = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$ of length d_a, $\mathbf{c_b} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ of length d_b, we have

$$c(z) = \det(z\mathbf{I} - \mathbf{C}) = z \cdot a(z)b(z) + c_0.$$

Remark Proving this theorem automatically proves the validity of the constructions of the supersparse companion matrices for p_n, q_n, and r_n.

Remark Starting with a polynomial $c(z)$, we see that there are potentially many such $a(z)$ and $b(z)$. This freedom may be quite valuable or, it may be an obstacle.

Proof Partition

$$z\mathbf{I} - \mathbf{C} = \left[\begin{array}{c|c} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \hline \mathbf{C}_{21} & \mathbf{C}_{22} \end{array}\right]$$

where $\mathbf{C}_{22} = z\mathbf{I} - \mathbf{B}$ is nonsingular if z is not an eigenvalue of $\mathbf{B}$, i.e. $b(z) \neq 0$. Later we will remove this restriction. Also,

$$\mathbf{C}_{21} = \begin{bmatrix} & 1 \\ & \end{bmatrix}$$

is $d_b \times (d_a + 1)$ and has only one nonzero element, which is a 1 in the upper right corner. Next,

$$\mathbf{C}_{12} = \begin{bmatrix} & \alpha c_0 \\ & \end{bmatrix}$$

is $(1 + d_a) \times d_b$ and again has only one nonzero element, αc_0 in the upper right corner. [In fact, c_0 can be zero.] This leaves

$$\mathbf{C}_{11} = \left[\begin{array}{cc:c} & & 0 \\ & z\mathbf{I} - \mathbf{A} & \vdots \\ & & 0 \\ & & 0 \\ \hdashline & 1 & z \end{array}\right]$$

which is $d_a + 1$ by $d_a + 1$.

The Schur factoring is

$$\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{C}_{12} \\ 0 & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21} & 0 \\ \mathbf{C}_{22}^{-1}\mathbf{C}_{21} & \mathbf{I} \end{bmatrix}$$

with the computation of the Schur complement $\mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}$ going to do most of the work in the proof. The Schur determinantal formula [10, Chapter 12] is then

$$\det \mathbf{C} = \det(\mathbf{C}_{22}) \det\left(\mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}\right).$$

We have the following propositions.

1. $z\mathbf{I} - \mathbf{A}$ and $z\mathbf{I} - \mathbf{B}$ are upper Hessenberg because $\mathbf{A}$ and $\mathbf{B}$ are.
2. The first d_a columns of $\mathbf{C}_{22}^{-1}\mathbf{C}_{21}$ are zero.
3. The final column of $\mathbf{C}_{22}^{-1}\mathbf{C}_{21}$ is the solution, say $\overrightarrow{v}$, of $(z\mathbf{I} - \mathbf{B})\overrightarrow{v} = \mathbf{e}_1$. Again, $z\mathbf{I} - \mathbf{B}$ is nonsingular.
4. By Cramer's rule, the final entry in $\overrightarrow{v}$, say v, is

$$v = \frac{\det\left(\mathbf{C}_{22} \underset{d_b}{\leftarrow} \mathbf{e}_1\right)}{\det(\mathbf{C}_{22})}$$

   where the notation $\mathbf{M} \underset{k}{\leftarrow} \overrightarrow{v}$ means replace the kth column of $\mathbf{M}$ with the vector $\overrightarrow{v}$ [3].

5. Since $\mathbf{C}_{22} = z\mathbf{I} - \mathbf{B}$ is upper Hessenberg,

$$\mathbf{C}_{22} \underset{d_b}{\leftarrow} e_1 = \begin{bmatrix} * & * & * & \cdots & * & 1 \\ -b_{21} & * & * & \cdots & * & 0 \\ & -b_{32} & * & & \vdots & \vdots \\ & & -b_{43} & \ddots & & \\ & & & \ddots & & \\ & & & & * & 0 \\ & & & & -b_{d_b,d_b-1} & 0 \end{bmatrix}.$$

Laplace expansion about the final column gives

$$\begin{aligned}\det\left(\mathbf{C}_{22} \underset{d_b}{\leftarrow} \mathbf{e}_1\right) &= (-1)^{d_b-1}(-1)^{d_b-1}\prod_{j=1}^{d_b-1} b_{j+1,j} \\ &= \prod_{j=1}^{d_b-1} b_{j+1,j}.\end{aligned}$$

Therefore,

$$v = \frac{\prod_{j=1}^{d_b-1} b_{j+1,j}}{b(z)}$$

because $\det \mathbf{C}_{22} = \det(z\mathbf{I} - \mathbf{B}) = b(z)$ by hypothesis.

6. Now

$$\mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21} = \begin{bmatrix} & \alpha c_0 \\ & \end{bmatrix}\begin{bmatrix} * \\ \vdots \\ * \\ v \end{bmatrix} = \begin{bmatrix} & \alpha c_0 v \\ & \end{bmatrix}$$

is $d_a + 1$ by $d_a + 1$ and has its only nonzero entry, $\alpha c_0 v$, in the upper right corner.

7. The Schur complement is therefore

$$\left[\begin{array}{c:c} & -\alpha c_0 v \\ & 0 \\ z\mathbf{I} - \mathbf{A} & \vdots \\ & 0 \\ \hdashline 0 \quad \cdots \quad 0 \quad 1 & z \end{array}\right]$$

and we compute $\det\left(\mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}\right)$ by Laplace expansion on the last column:

$$\begin{aligned}
\det\left(\mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}\right) &= -(-1)^{d_a}\alpha c_0 v \det\begin{bmatrix} -a_{21} & * & * & \cdot & * \\ & -a_{32} & * & & * \\ & & -a_{43} & & \vdots \\ & & & \ddots & \\ & & & & -a_{d_a,d_a-1} \end{bmatrix} \\
&\quad + z\det(z\mathbf{I} - \mathbf{A}) \\
&= -(-1)^{d_a}\alpha c_0 v \prod_{j=1}^{d_a-1}\left(-a_{j+1,j}\right) + z\cdot a(z) \\
&= \alpha v \prod_{j=1}^{d_a-1} a_{j+1,j}\cdot c_0 + z\cdot a(z) \\
&= \alpha\cdot\frac{\left(\prod_{j=1}^{d_b-1} b_{j+1,j}\right)}{b(z)}\cdot\left(\prod_{j=1}^{d_a-1} a_{j+1,j}\right)\cdot c_0 + z\cdot a(z) \\
&= \frac{c_0}{b(z)} + z\cdot a(z)
\end{aligned}$$

by the definition of α.

Therefore by the Schur determinantal formula

$$\begin{aligned}
\det(z\mathbf{I} - \mathbf{C}) &= \det(\mathbf{C}_{22})\det\left(\mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}\right) \\
&= b(z)\left(\frac{c_0}{b(z)} + z\cdot a(z)\right) \\
&= z\cdot a(z)b(z) + c_0.
\end{aligned}$$

Since the left hand side is a polynomial as is the right hand side, the formula will be true even if $b(z) = 0$, by continuity.

♮

2.3 Applications and examples

Sequence A000930 of the Online Encyclopedia of Integer Sequences, Narayana's cows sequence, begins

$$1, 1, 1, 2, 3, 4, 6, 9, 13, 19, \ldots$$

and is generated by $R_n = R_{n-1} + R_{n-3}$ [13]. The connection to cows is that an ideal cow produces a calf every year, starting in its fourth year. Narayana was a mathematician in 14th century India. Various facts are known for this sequence, which is similar to the Fibonacci sequence: for instance, the generating function is $1/(1 - x - x^3)$. Many references are given in the OEIS, but see also [12].

We define the Narayana-Mandelbrot polynomials by $r_0 = 1, r_1 = r_2 = 1$ and

$$r_{n+1} = z r_n r_{n-2} + 1$$

for $n \geq 2$. We construct a recursive family of companion matrices $\mathbf{R}_n$, i.e. such that

$$r_n(z) = \det(z\mathbf{I} - \mathbf{R}_n).$$

Just as the Fibonacci-Mandelbrot polynomials, the construction contains matrices of different sizes. However, for this family, we start with

$$\mathbf{R}_3 = \begin{bmatrix} -1 \end{bmatrix},$$

$$\mathbf{R}_4 = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix},$$

and

$$\mathbf{R}_5 = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix}.$$

Our construction is then

$$\mathbf{R}_{n+1} = \begin{bmatrix} \mathbf{R}_n & & (-1)^{d_{n+1}} \mathbf{c}_n \mathbf{r}_{n-2} \\ -\mathbf{r}_n & 0 & \\ & -\mathbf{c}_{n-2} & \mathbf{R}_{n-2} \end{bmatrix},$$

where $\mathbf{r}_n = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$ and $\mathbf{c}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ are, as before, the row and column vectors of length $d_n = \deg r_n = R_{n+1} - 1$.

This construction also allows new matrix families. For instance, suppose $s_0 = 0$, $s_{n+1} = z^3 s_n^4 + 1$. Then if $\mathbf{S}_n$ is an upper Hessenberg companion for s_n (with all -1 on the subdiagonal) the matrix

$$\mathbf{S_{n+1}} = \begin{bmatrix} \mathbf{S}_n & & & & & & -\mathbf{c}_n \mathbf{r}_n \\ -\mathbf{r}_n & 0 & & & & & \\ & -\mathbf{c}_n & \mathbf{S}_n & & & & \\ & & -\mathbf{r}_n & 0 & & & \\ & & & -\mathbf{c}_n & \mathbf{S}_n & & \\ & & & & -\mathbf{r}_n & 0 & \\ & & & & & -\mathbf{c}_n & \mathbf{S}_n \end{bmatrix}$$

is an upper Hessenberg companion for s_{n+1}.

2.4 Concluding remarks

This is a genuinely new kind of companion matrix. We demonstrate this on Newton's example polynomial $x^3 - 2x - 5$. We see that $x^3 - 2x - 5 = x(x^2 - 2) - 5 = x(x - \sqrt{2})(x + \sqrt{2}) - 5$,

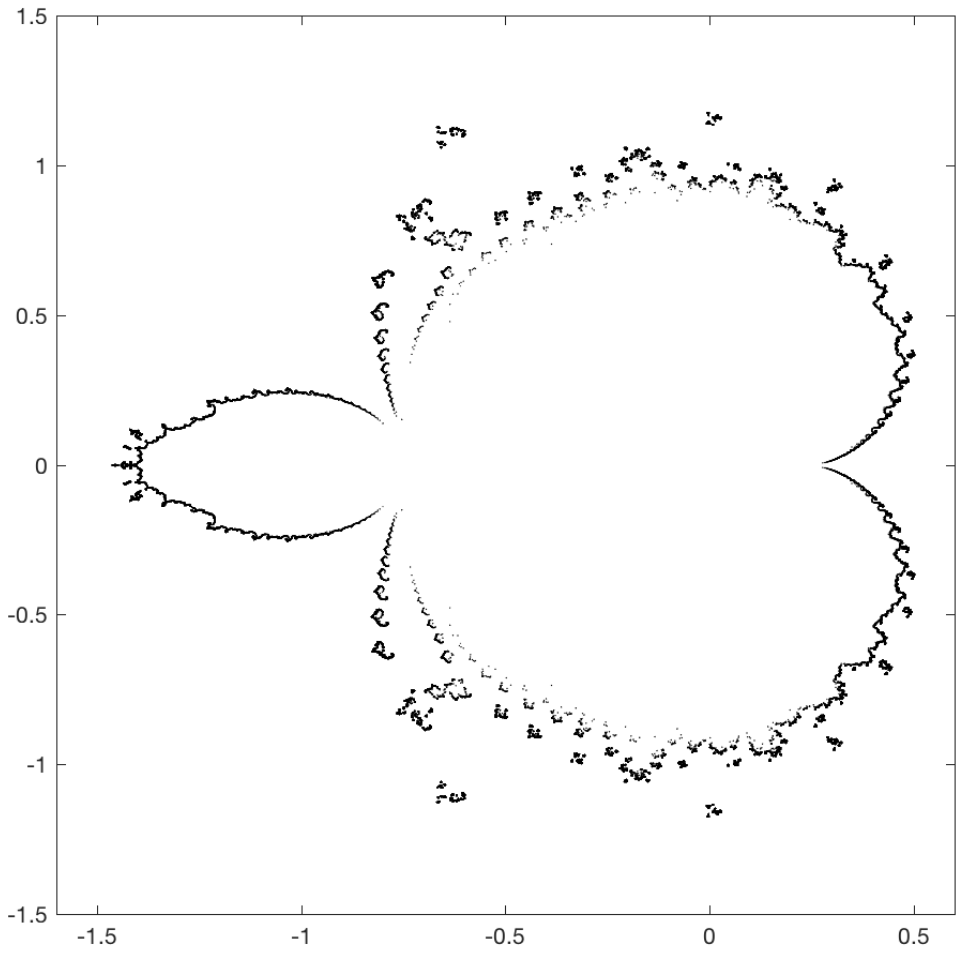


Figure 2.1: Roots of Narayana-Mandelbrot polynomial, $r_{36}(z)$. The degree of $r_{36}(z)$ is 578,948.

and companion matrices for $x - \sqrt{2}$ and $x + \sqrt{2}$ are just $[+\sqrt{2}]$ and $[-\sqrt{2}]$ respectively. Thus a companion matrix for Newton's polynomial is

$$\begin{bmatrix} \sqrt{2} & & 5 \\ -1 & & \\ & -1 & -\sqrt{2} \end{bmatrix}.$$

This matrix contains $\sqrt{2}$, unlike any previously recorded companion matrix. For unimodular polynomials, such companion matrices may be of lower height than the Frobenius or Fiedler [9] companions, and may offer better numerical condition.

We have now established that if $c(z) = z \cdot a(z)b(z) + c_0$ and $\mathbf{A}$ and $\mathbf{B}$ are upper Hessenberg companion matrices for the polynomials $a(z)$ and $b(z)$ respectively, then

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} & & -\alpha c_0 \mathbf{c_a} \mathbf{r_b} \\ -\mathbf{r_a} & 0 & \\ & -\mathbf{c_b} & \mathbf{B} \end{bmatrix}$$

is a companion matrix for $c(z)$. One wonders immediately about a corresponding linearization, $\mathbf{L_C}$, strong or otherwise, for the matrix polynomial

$$\mathbf{C}(z) = z\mathbf{A}(z)\mathbf{B}(z) + \mathbf{C_0}\,,$$

if $\mathbf{L_A}$ is a linearization for $\mathbf{A}$, $\mathbf{L_B}$ for $\mathbf{B}$. Some very preliminary experiments, where $\mathbf{L_A}$ and $\mathbf{L_B}$ were block upper Hessenberg with all blocks $\mathbf{I}$, so $\alpha = 1$, find that indeed

$$\mathbf{L_C} = \begin{bmatrix} \mathbf{L_A} & & -\mathbf{C}_0 \\ -\mathbf{I} & 0 & \\ & -\mathbf{I} & \mathbf{L_B} \end{bmatrix}$$

is a (strong) linearization for $c(z)$, in the examples we tried.

In a paper to be submitted soon, we have now proved that this construction can be extended to matrix polynomials. See [6].

A referee pointed out that Robol et al. [11] use a similar construction to linearize polynomials of the form $p(z) = a(z)b(z) + zc(z)d(z)$ to find the roots of rational functions, which can also be applied to matrix polynomials.

We leave these extensions to future work.

Acknowledgments

We thank Neil J. A. Sloane for introducing us to the Narayana's cows sequence, Dario Bini [1, 2] for teaching us about Mandelbrot polynomials and the Schur complement, and Donald E. Knuth. We thank the referees for their remarks.

Bibliography

[1] D. A. Bini and G. Fiorentino, *Design, analysis, and implementation of a multiprecision polynomial rootfinder*, Numerical Algorithms, 23 (2000), pp. 127–173.

[2] D. A. Bini and L. Robol, *Solving secular and polynomial equations: A multiprecision algorithm*, Journal of Computational and Applied Mathematics, 272 (2014), pp. 276–292.

[3] D. Carlson, C. R. Johnson, D. Lay, and A. D. Porter, *Gems of exposition in elementary linear algebra*, The College Mathematics Journal, 23 (1992), pp. 299–303.

[4] E. Y. S. Chan, *A comparison of solution methods for Mandelbrot-like polynomials*, Master's thesis, The University of Western Ontario, 2016.

[5] E. Y. S. Chan and R. M. Corless, *Fibonacci-Mandelbrot polynomials and matrices*, in ISSAC 2016, Waterloo, Ontario, 2016.

[6] E. Y. S. Chan, R. M. Corless, L. Gonzalez-Vega, J. R. Sendra, and J. Sendra, *Algebraic linearizations for matrix polynomials*, Linear Algebra and its Applications, 563 (2019), pp. 373–399.

[7] R. M. Corless and N. Fillion, *A graduate introduction to numerical methods*, AMC, 10 (2013), p. 12.

[8] R. M. Corless and P. W. Lawrence, *Mandelbrot polynomials and matrices*, In preparation.

[9] M. Fiedler, *A note on companion matrices*, Linear Algebra and its Applications, 372 (2003), pp. 325–331.

[10] L. Hogben, *Handbook of linear algebra*, CRC Press, 2006.

[11] L. Robol, R. Vandebril, and P. V. Dooren, *A framework for structured linearizations of matrix polynomials in various bases*, SIAM Journal on Matrix Analysis and Applications, 38 (2017), pp. 188–216.

[12] N. J. A. Sloane, *My favorite integer sequences*, in Sequences and their Applications, Springer, 1999, pp. 103–130.

[13] ——, *The on-line encyclopedia of integer sequences*. published electronically at `https://oeis.org`, 2016. Sequence A000930.

Chapter 3

Minimal height companion matrices for Euclid polynomials

3.1 Introduction

The sequence $e_n = 2, 3, 7, 43, 1807, \ldots$ defined by $e_1 = 2$ and the recurrence relation

$$e_{n+1} = e_n e_{n-1} \cdots e_2 e_1 + 1 = e_n(e_n - 1) + 1$$

for $n \geq 1$, is known under various names: Euclid numbers, Sylvester's sequence, or Ahmes numbers. The sequence can be found at The Online Encyclopedia of Integer Sequences as entry A000058. There, we find references to work of Erdös, Shparlinsky, Vardi, Sloane, Guy, and other well-known number theorists and analysts.

These numbers, which we will call Euclid numbers, as they are called in [7, chapter 4], have interesting properties. For instance, they are mutually relatively prime. Quoting [7],

> "Euclid's algorithm (what else?) tells us this in three short steps, because $e_n \bmod e_m = 1$ when $n > m$: $\gcd(e_n, e_m) = \gcd(1, e_m) = \gcd(1, 0) = 1$."

Euclid numbers grow *doubly exponentially*; indeed exercise 37, chapter 4 of [7] asks the reader to prove[1] that

$$e_n = \left\lfloor E^{2^n} + \frac{1}{2} \right\rfloor$$

for a number $E \approx 1.264$; here $\lfloor x \rfloor$ is the floor of x, the largest integer not greater than x.

The name "Ahmes numbers" comes from a connection to so-called *Egyptian fractions*[2]. Quoting Néstor Romeral Andrés from the A000058 entry,

> "The greedy Egyptian representation of 1 is $1 = 1/2 + 1/3 + 1/7 + 1/43 + 1/1807 + \cdots$"

and he then goes on to give a geometric dissection of a unit square (in words) proving this assertion. Algebraically, we have the following.

[1] The hint there is to write $e_{n+1} - 1/2 = (e_n - 1/2)^2 + 1/4$ and consider $2^{-n} \log(e_n - 1/2)$.

[2] Quoting Exercise 9, p. 95 from [7], "Egyptian mathematicians in 1800 BC represented rational numbers between 0 and 1 as sums of unit fractions $1/x_1 + \cdots + 1/x_k$ where the x_k were distinct positive integers."

Lemma 3.1.1 *For $n \geq 1$,*

$$1 = \sum_{k=1}^{n} \frac{1}{e_k} + \frac{1}{e_{n+1} - 1} \, .$$

Proof An easy induction: clearly $1 = {}^1\!/_2 + {}^1\!/_2 = {}^1\!/_2 + {}^1\!/_{(3-1)}$ so the statement is true for $n = 1$. Then

$$\begin{aligned} 1 &= \sum_{k=1}^{n} \frac{1}{e_k} + \frac{1}{e_{n+1} - 1} \\ &= \sum_{k=1}^{n+1} \frac{1}{e_k} + \frac{1}{e_{n+1} - 1} - \frac{1}{e_{n+1}} \\ &= \sum_{k=1}^{n+1} \frac{1}{e_k} + \frac{e_{n+1} - e_{n+1} + 1}{e_{n+1}\,(e_{n+1} - 1)} \\ &= \sum_{k=1}^{n+1} \frac{1}{e_k} + \frac{1}{e_{n+2} - 1} \, . \end{aligned}$$

♮

There are other properties too, but we hope that this is enough to whet your appetite because we want to move on to what we call[3] "Euclid polynomials." Put

$$E_1(\lambda) = \lambda + 1$$

and

$$E_{n+1}(\lambda) = \lambda E_n(\lambda) E_{n-1}(\lambda) \cdots E_1(\lambda) + 1$$

for $n \geq 1$. Then, obviously, $E_k(0) = 1$ for $k \geq 1$ and $E_k(1) = e_k$ for $k \geq 1$. Possibly these polynomials in the variable λ can shed some light on Euclid numbers. One could make $E_0(\lambda) = 1$ but this complicates later formulae to no purpose. The first few Euclid polynomials are

$$\begin{aligned} E_1 &= \lambda + 1 \\ E_2 &= \lambda^2 + \lambda + 1 \\ E_3 &= \lambda^4 + 2\lambda^3 + 2\lambda^2 + \lambda + 1 \\ E_4 &= \lambda^8 + 4\lambda^7 + 8\lambda^6 + 10\lambda^5 + 9\lambda^4 + 6\lambda^3 + 3\lambda^2 + \lambda + 1 \, . \end{aligned}$$

We will enumerate and prove some properties of these polynomials in the next section, but first we confess: we're not interested in Euclid polynomials because of their connection to Euclid numbers. We are interested because we have a new technique for finding their roots, namely by finding an equivalent eigenvalue problem (a so-called "companion matrix") that has a very interesting property of its own, namely that out of all integer matrices $\mathbf{A}_k$ having

$$E_k(\lambda) = \det(\lambda \mathbf{I} - \mathbf{A}_k)$$

the height of $\mathbf{A}_k$—that is, the largest absolute value of any entry of $\mathbf{A}_k$—is the *least* when we use our method.

[3]The polynomials $E_k(-\lambda)$ occur, not with this name, as sequence A225200 by Martin Renner.

Remark $\mathbb{H}(\mathbf{A}) = \text{Height}(\mathbf{A}) = \|\operatorname{vec}(\mathbf{A})\|_\infty$ is actually a matrix norm. It is not, however, submultiplicative:

$$\mathbb{H}(\mathbf{AB}) \not\leq \mathbb{H}(\mathbf{A})\mathbb{H}(\mathbf{B}) .$$

For example, consider

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

We will find companion matrices for $E_k(\lambda)$ of height 1, as small as possible for any integer matrix. This is to be contrasted with the size of the largest polynomial coefficient of $E_k(\lambda)$, which since

$$E_k(1) = \sum_{j=0}^{2^{k-1}} E_{j,k} = \left\lfloor E^{2^k} + \frac{1}{2} \right\rfloor$$

must at least be

$$\frac{1}{2^{k-1}+1} \left\lfloor E^{2^k} + \frac{1}{2} \right\rfloor = O\left(E^{2^k - O(k)}\right)$$

(the maximum cannot be smaller than the average). Here, we are denoting the coefficients of

$$E_k(\lambda) = \sum_{j=0}^{\deg E_k} E_{j,k} \lambda^j$$

by $E_{j,k}$ and claiming $\deg E_k(\lambda) = 2^{k-1}$, which we will prove in the next section. This massive reduction in height has important numerical consequences. The eigenvalues of this "minimal height companion matrix" will be much easier to compute in comparison to the roots of the explicit polynomial (with its doubly-exponentially large coefficients).

This minimal height companion matrix would itself just be a curiosity, except that the technique we use to generate it turns out to be quite general, and in fact can be extended to *matrix* polynomials, giving so-called *lower-height linearizations*[4]. Euclid polynomials have a special place in our hearts, though, because it was by finding their minimal height companion matrices that we realized the technique was, in fact, general.

3.2 Properties of Euclid polynomials

Proposition 3.2.1 $\deg E_k(\lambda) = 2^{k-1}$.

Proof $\deg E_1(\lambda) = \deg \lambda + 1 = 1 = 2^{1-1}$. Since

$$\begin{aligned} E_{k+1}(\lambda) &= \lambda E_k(\lambda) E_{k-1}(\lambda) \cdots E_1(\lambda) + 1 \\ &= E_k(\lambda) \left(E_k(\lambda) - 1\right) + 1 \end{aligned}$$

for $k \geq 2$, and independently for $k = 1$ when

$$\begin{aligned} E_2(\lambda) &= (1+\lambda) \cdot \lambda + 1 , \\ \deg E_{k+1}(\lambda) &= 2 \deg E_k(\lambda) . \end{aligned}$$

If $\deg E_k(\lambda) = 2^{k-1}$, $\deg E_{k+1}(\lambda) = 2^{k+1-1}$. This establishes the inductive step. ♮

[4] *Minimal* height linearizations are an open question.

Proposition 3.2.2 *If* $E_k(\lambda) = \sum_{j=0}^{2^{k-1}} E_{j,k}\lambda^j$*, then all* $E_{j,k}$ *are positive integers,*

$$E_{0,k} = E_{2^{k-1},k} = 1 \ ,$$

and

$$e_k = E_k(1) = \sum_{j=0}^{2^{k-1}} E_{j,k} \ .$$

Proof

$$\begin{aligned} E_{k+1}(\lambda) &= E_k(\lambda)\,(E_k(\lambda) - 1) + 1 \\ &= \lambda E_k(\lambda) E_{k-1}(\lambda) \cdots E_1(\lambda) + 1 \end{aligned}$$

has trailing coefficient 1 (set $\lambda = 0$) and leading coefficient 1 (the square of the leading coefficient of $E_k(\lambda)$). As for $E_{j,k} \geq 1$ being integral, the Cauchy product formula gives

$$\left[z^j\right] E_{k+1}(\lambda) = E_{j,k+1}$$

(the coefficient of z^j of E_{k+1})

$$= \sum_{\ell=0}^{j} E_{\ell,k} \hat{E}_{j-\ell,k}$$

where

$$\hat{E}_{j-\ell,k} = \begin{cases} E_{j-\ell,k} & \text{if } \ell < j \\ 0 & \text{if } \ell = j \end{cases}$$

is a sum of products of positive integers, and hence a positive integer. The statement $e_k = \sum_{j=0}^{2^{k-1}} E_{j,k}$ follows from the definition of $E_{j,k}$. ♮

Proposition 3.2.3

$$\max_{0 \leq j \leq 2^k} E_{j,k+1} \geq \left(\max_{0 \leq j \leq 2^{k-1}} E_{j,k} \right)^2 .$$

Proof From the Cauchy product in the last proposition, if j^* is the index of the largest coefficient of $E_k(\lambda)$, then for $j = 2j^*$ in $E_{k+1}(\lambda)$ the coefficient of $\left[z^j\right]$ is

$$\sum_{\ell=0}^{2j} E_{\ell,k} E_{2j-\ell,k}$$

which, for $\ell = j^*$, contains

$$E_{j^*,k} E_{j^*,k} = E_{j^*,k}^2$$

which establishes the proposition. ♮

Proposition 3.2.4 *The largest coefficient of* $E_k(\lambda)$ *grows doubly exponentially with k.*

Proof 1

$$e_k = E_k(1) = \sum_{j=0}^{2^{k-1}} E_{j,k} = \left\lfloor E^{2^k} + \frac{1}{2} \right\rfloor ,$$

then

$$\begin{aligned} \max_j E_{j,k} &\geq \frac{1}{2^{k-1}+1} \left\lfloor E^{2^k} + \frac{1}{2} \right\rfloor \\ &= E^{2^k - O(k)} . \end{aligned}$$

♮

Proof 2 By inspection, $\max_j E_{j,3} = 2$. Since $\max_j E_{j,4} = 10 > 2^2 = 2^{1/4 \cdot 2^3} = 2^{1/4 \cdot k}$, we are well on our way. Assume that $\max_j E_{j,k} = 2^{c_1 2^k}$. Then $\max_j E_{j,k+1} \geq \left(2^{c_1 \cdot 2^k}\right)^2 = 2^{c_1 2^{k+1}}$. ♮

Proposition 3.2.5 *The polynomials $E_k(\lambda)$ are all mutually relatively prime, as polynomials over $\mathbb{Z}$.*

Proof The proof is the same as that proving the e_k are relatively prime integers: $E_n(\lambda) \equiv 1 \mod E_m(\lambda)$ if $n > m \Rightarrow \gcd(E_n(\lambda), E_m(\lambda)) = \gcd(1, E_m(\lambda)) = 1$. ♮

Proposition 3.2.6 *The roots of $E_k(\lambda)$ are simple.*

Proof This is true for $E_1(\lambda)$ and $E_2(\lambda)$.

Assume to the contrary that for some k there exists a λ^* for which both

$$E_{k+1}(\lambda^*) = 0$$

and

$$E'_{k+1}(\lambda^*) = 0 .$$

Then since for any $1 \leq j \leq k$

$$E_{j+1}(\lambda) = E_j(\lambda)\left(E_j(\lambda) - 1\right) + 1 ,$$

we have

$$E'_{j+1}(\lambda) = \left(2E_j(\lambda) - 1\right) E'_j(\lambda) .$$

Therefore, either $E_k(\lambda^*) = 1/2$ (which is impossible because then $E_{k+1}(\lambda^*) = 1/2(-1/2) + 1 = 3/4 \neq 0$) or $E'_k(\lambda^*) = 0$. If there exists any $\ell < k$ for which $E'_\ell(\lambda^*) \neq 0$ while $E'_{\ell+1}(\lambda^*) = 0$, then $E_\ell(\lambda^*) = 1/2$ because $E'_{\ell+1}(\lambda) = (2E_\ell(\lambda) - 1)\, E'_\ell(\lambda)$. If $E_\ell(\lambda^*) = 1/2$, then $E_j(\lambda^*)$ for $j \geq \ell$ is rational because

$$E_{j+1}(\lambda^*) = E_j(\lambda^*)(E_j(\lambda^*) - 1)$$

is a product of rational numbers.

This gives an ultimate contradiction because

$$E_k(\lambda^*)(E_k(\lambda^*) - 1) + 1 = 0$$

only if $E_k(\lambda^*) = -1/2 \pm i\sqrt{3}/2 \notin \mathbb{Q}$. ♮

Proposition 3.2.7

$$\frac{1}{\lambda} = \sum_{k=1}^{n} \frac{1}{E_k(\lambda)} + \frac{1}{E_{n+1}(\lambda) - 1} \,. \tag{3.1}$$

Proof Identical to Lemma 3.1.1 on substituting $E_k(\lambda)$ for e_k and noting

$$\begin{aligned} \frac{1}{\lambda} &= \frac{1}{\lambda + 1} + \frac{1}{\lambda^2 + \lambda} \\ &= \frac{\lambda}{\lambda^2 + \lambda} + \frac{1}{\lambda^2 + \lambda} \\ &= \frac{\lambda + 1}{\lambda(\lambda + 1)} \\ &= \frac{1}{\lambda} \,. \end{aligned}$$

ϟ

Remark The series $\sum_{k\geq 1} E_k^{-1}(\lambda)$ converges if $\lambda > 0$ and diverges if $\lambda = -1/2$.

Conjecture 3.2.8 *There is convergence outside the "cauliflower" in Figure 3.1 and divergence inside the cauliflower.*

Definition We say that a polynomial $p(\lambda)$ is *unimodal* [9] if its coefficient vector $[a_0, a_1, \cdots, a_n]$ of positive integers has first monotonic increase to a peak (which may occur twice or more at adjacent coefficients) and then decay to $a_n = 1$. Notice that $E_1(\lambda)$, $E_2(\lambda)$, $E_3(\lambda)$ and $E_4(\lambda)$ are unimodal.

Conjecture 3.2.9 *The Euclid polynomials are unimodal.*

Remark The doubly exponential growth of the polynomial coefficients mean that the *conditioning* of the evaluation of the polynomial grows doubly exponentially in k. Note that since the degree $\deg E_k = 2^{k-1}$, this means that the conditioning grows exponentially in the degree. In contrast, we will see in section 3.5 a much better condition number of the eigenvalues of the companions $\mathbb{E}_k$, sublinear in the degree. This means that evaluation (and rootfinding) requires significantly more precision (and therefore expense) if the monomial basis is used. The following definition is used in [6] and [5]:

$$B_k(\lambda) = \sum_{j=0}^{2^{k-1}} E_{j,k} \, |\lambda|^j$$

as a "condition number for the evaluation of the polynomial $E_k(\lambda)$" for a given λ. One can show that if

$$p_k(\lambda) = \sum_{j=0}^{2^{k-1}} E_{j,k}(1 + \delta_j)\lambda^j$$

then $p_k(\lambda)$ differs from $E_k(\lambda)$ by at most

$$|p_k(\lambda) - E_k(\lambda)| \leq B_k(\lambda)\left(\max_{0\leq j\leq 2^{k-1}} |\delta_k|\right) .$$

This shows that relative errors δ_k in the coefficients produce absolute errors in the values at most $B(\lambda)\|\delta\|_\infty$. From the foregoing discussion it is evident that on $0 \leq \lambda \leq 1$

$$\begin{aligned} B_k(\lambda) &= O\left(E^{2^k}\right) \\ &= O\left(E^{2\deg E_k(\lambda)}\right) \end{aligned}$$

is exponentially large in the degree of $E_k(\lambda)$. That is, in order to ensure that numerical errors in evaluation (which, by standard backward error results are equivalent to $O(\mu)$, where μ is the unit roundoff, relative changes in the coefficients) would require that the unit roundoff to be of size

$$\mu = O\left(E^{-2\deg E_k(\lambda)}\right)$$

which in turn requires $O(2\deg E_k)$ bits of precision; this is an exponential number of bits of precision, in k. To evaluate $E_k(\lambda)$ (or to find its roots) one would need to use $O\left(2^k\right)$ bit arithmetic. This is of course possible, but the cost of multiplication of high precision number grows faster than the precision length.

Luckily, there's a better way: minimal height companion matrices.

3.3 A brief history of the technique

In 2011, Piers W. Lawrence invented a family of companion matrices for the Mandelbrot polynomials[5], defined by $p_1(\lambda) = 1$ and for $n \geq 0$

$$p_{n+1}(\lambda) = \lambda p_n^2(\lambda) + 1 .$$

[5]It can be shown that the Euclid polynomials are related to the Mandelbrot polynomials. We can rewrite the Euclid polynomials as

$$\begin{aligned} f_{n+1} &= f_n^2 + \frac{1}{4} \\ 4f_{n+1} &= \frac{1}{4}(4f_n)^2 + 1 . \end{aligned}$$

We can then let $u_n = 4f_n$, so

$$u_{n+1} = \frac{1}{4}u_n^2 + 1 ,$$

which recurrence is the same as for the Mandelbrot polynomials, except with $z = 1/4$ and

$$u_1 = 4f_1 = 4(e_1 - 1/2) = 2 ;$$

whereas $p_1 = 1$.

We have $p_2(\lambda) = \lambda + 1$ with a (trivial) companion matrix $\mathbf{M}_2 = [-1]$. Piers invented a recursive construction,

$$\mathbf{M}_{n+1} \begin{bmatrix} \mathbf{M}_n & & -\mathbf{c}_n\mathbf{r}_n \\ -\mathbf{r}_n & 0 & \\ & -\mathbf{c}_n & \mathbf{M}_n \end{bmatrix}$$

where $\mathbf{r}_n = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$ and $\mathbf{c}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$, given

$$\begin{aligned} p_{n+1}(\lambda) &= \det(\lambda\mathbf{I} - \mathbf{M}_{n+1}) \\ &= \lambda \det(\lambda\mathbf{I} - \mathbf{M}_n)^2 + 1 \,. \end{aligned}$$

In her Masters' thesis [2], Eunice Chan extended this construction to Fibonacci-Mandelbrot polynomials $q_n(\lambda)$ satisfying

$$\begin{aligned} q_0(\lambda) &= 0 \\ q_1(\lambda) &= 1 \\ q_{n+1}(\lambda) &= \lambda q_n(\lambda) q_{n-1}(\lambda) + 1 \end{aligned}$$

and Narayana-Mandelbrot polynomials $r_n(\lambda)$ satisfying

$$\begin{aligned} r_0(\lambda) &= 1 \\ r_1(\lambda) &= 1 \\ r_2(\lambda) &= 1 \\ r_{n+1}(\lambda) &= \lambda r_n(\lambda) r_{n-2}(\lambda) + 1 \,. \end{aligned}$$

Chan used these to explore the comparative efficiency of linearization (companion matrices) and homotopy methods (i.e. following paths, also called continuation methods, from roots of $p_n(\lambda)$ to roots of $p_{n+1}(\lambda)$ and similarly for the others). [Spoiler alert: homotopy wins, hands down.]

These families of polynomials all have similarities and it is not really surprising that analogues of Piers Lawrence's construction work to make companion matrices.

Donald E. Knuth suggested we look at Euclid numbers (polynomials). The fact that it worked immediately suggested that the construction was in fact general, which led to the papers [3] and [4].

We return from that generality to the Euclid polynomials, which are interesting enough in themselves to deserve further attention. In the rest of this paper, we show how this general technique of construction applies to the Euclid polynomials, how far we can push it, and what we learn in the process.

3.4 Computation of eigenvalues

Suppose $E_k = \det(\lambda\mathbf{I} - \mathbb{E}_k)$. Each identity matrix $\mathbf{I}$ is a different size, but this should be natural enough: it will be $\deg E_k$ by $\deg E_k$ if it's being used in $\lambda\mathbf{I} - \mathbb{E}_k$. Notice that this amounts to a

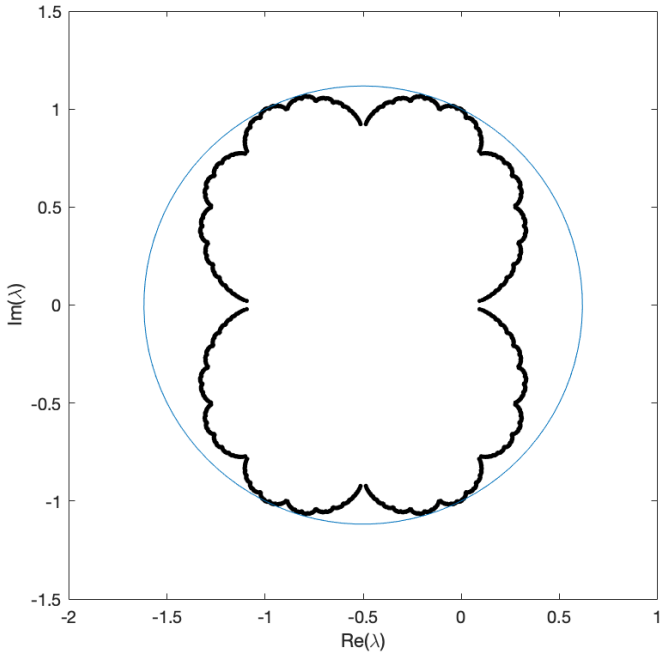


Figure 3.1: All 16,384 roots of the Euclid polynomial $E_{15}(\lambda)$ with circle of radius 1.1180, the approximate magnitude of the largest $|\lambda + {}^1\!/_2|$.

strong induction—we will need companion matrices for each prior polynomial in order to find one for E_{k+1}. Then put

$$\widetilde{\mathbb{E}}_k := \begin{bmatrix} 0 & & & & \\ -1 & \mathbb{E}_1 & & & \\ & -1 & \ddots & & \\ & & -1 & \mathbb{E}_{k-2} & \\ & & & -1 & \mathbb{E}_{k-1} \end{bmatrix} = \mathbb{E}_k - \begin{bmatrix} 0 & \cdots & 0 & 1 \\ & & & 0 \\ & & & \vdots \\ & & & 0 \end{bmatrix} .$$

Remark $\det\left(\lambda\mathbf{I} - \widetilde{\mathbb{E}}_k\right) = E_k(\lambda) - 1 = \lambda \sum_{j=1}^{k-1} E_j(\lambda)$; subtracting 1 just changes the final column of this companion (see [4]).

This is upper Hessenberg, but block lower triangular; therefore, its determinant is the product of the determinants of the blocks (see e.g. [8]) , and similarly for the resolvent [10], like so:

$$\det\left(\lambda\mathbf{I} - \widetilde{\mathbb{E}}_k\right) = \lambda E_1(\lambda)E_2(\lambda)E_3(\lambda)\cdots E_{k-1}(\lambda) .$$

Therefore, if we put a 1 in the upper right corner (we will see shortly it must be +1),

$$\mathbb{E}_{k+1} := \begin{bmatrix} \widetilde{\mathbb{E}}_k & & 1 \\ & -1 & \\ & & \mathbb{E}_k \end{bmatrix} ,$$

we will have $E_{k+1}(\lambda) = \det(\lambda\mathbf{I} - \mathbb{E}_{k+1})$ for $k \geq 2$ and $\mathbb{E}_{k+1}$ will be (irreducibly) upper Hessenberg if $\mathbb{E}_k$ is.

Explicitly, $\mathbb{E}_1 = [-1]$ and we may take

$$\mathbb{E}_2 = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$$

because $\det(\lambda\mathbf{I} - \mathbb{E}_2) = \det\begin{pmatrix} \lambda & -1 \\ 1 & \lambda+1 \end{pmatrix} = \lambda(\lambda+1) + 1 = E_2(\lambda)$. Therefore,

$$\mathbb{E}_3 = \begin{bmatrix} 0 & & 1 \\ -1 & -1 & \\ & -1 & 0 \quad 1 \\ & & -1 \quad -1 \end{bmatrix}.$$

To confirm, we form

$$\lambda\mathbf{I} - \mathbb{E}_3 = \begin{bmatrix} \lambda & 0 & 0 & -1 \\ 1 & \lambda+1 & 0 & 0 \\ & 1 & \lambda & -1 \\ & & 1 & \lambda+1 \end{bmatrix}.$$

A short computation shows

$$\begin{aligned} \det(\lambda\mathbf{I} - \mathbb{E}_3) &= \lambda(\lambda+1)(\lambda(\lambda+1)+1)+1 \\ &= \lambda E_1(\lambda)E_2(\lambda) + 1 \\ &= E_3(\lambda) \end{aligned}$$

as desired. Emboldened, we build

$$\mathbb{E}_4 = \begin{bmatrix} 0 & & & & 1 \\ -1 & -1 & & & \\ & -1 & 0 \quad 1 & & \\ & & -1 \quad -1 & & \\ & & & 0 & 1 \\ & & & -1 \quad -1 & \\ & & & & -1 \quad 0 \quad 1 \\ & & & & & -1 \quad -1 \end{bmatrix}$$

and direct computation again shows

$$\begin{aligned} \det(\lambda\mathbf{I} - \mathbb{E}_4) &= \lambda(\lambda+1)(\lambda(\lambda+1)+1)(\lambda(\lambda+1)(\lambda(\lambda+1)+1)+1)+1 \\ &= \lambda E_1(\lambda)E_2(\lambda)E_3(\lambda) + 1 \\ &= E_4(\lambda)\,. \end{aligned}$$

Theorem 3.4.1

$$E_k(\lambda) = \det(\lambda\mathbf{I} - \mathbb{E}_k)$$

where $\mathbb{E}_k$ is defined as above.

Proof This follows immediately from Theorem 4 of [4]. An easy proof follows from linearity of $(\lambda \mathbf{I} - \mathbb{E}_k)$ in its first row, and that the determinant of a block lower triangular matrix is the product of the determinants of the blocks; the 1 in the corner contributes $(-1)^{\deg(E_k(\lambda))-1} \cdot (-1)^{\deg(E_k(\lambda))-1} = +1$. ♮

Lemma 3.4.2 *The upper right corner of* $\mathbb{E}_k$ *is always* 1.

Proof As mentioned in Theorem 4 from [4], the element in the upper right corner is dependent on the degree of the polynomial, in this case $(-1)^{\deg E_k}$ for $\mathbb{E}_k$. Since the degree of the Euclid polynomials is

$$\begin{aligned} \deg E_k &= 1 + \deg(E_{k-1}) - 1 + \deg E_{k-1} \\ &= 2 \deg E_k \end{aligned}$$

and $\deg E_1 = 1$; therefore,

$$\deg E_k = 2^{k-1} ,$$

which means that $\deg E_k$ is always even for $k \geq 2$, and thus, the upper right corner of $\mathbb{E}_k$ is always 1. We get $(-1)^{\deg E_k - 1}$ from Laplace expansion and $(-1)^{\deg E_k - 1}$ from minor and therefore,

$$\left((-1)^{\deg E_k - 1}\right)^2 = +1 .$$

♮

Remark These "Bohemian" matrices[6] contain only entries that are -1, 0, or 1: the bound on that height of the entries is just $|m_{ij}| \leq 1$. But the coefficients of the Euclid polynomials $E_k(\lambda)$ are decidedly *not* bounded. This is just like the Mandelbrot polynomials, whose (polynomial coefficient) height grows exponentially with their degree $d_n = 2^{n-1} - 1$, and *doubly* exponentially with n. The eigenvalue problems we have found are considerably easier to solve than the monomial basis polynomials are!

Remark There are many choices here—these companion matrices are in no way unique. For instance, we could use any of

$$\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}, \quad \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}$$

for $\mathbb{E}_2$; and we may arrange the blocks for λ (i.e. [0]), E_1, E_2, …, E_{k-1} in any order; at this time we do not know which order is best numerically, if any.

[6] A matrix family is *Bohemian* if its entries come from a single discrete (and hence bounded) set. The name comes from "Bounded Height Matrix of Integers."

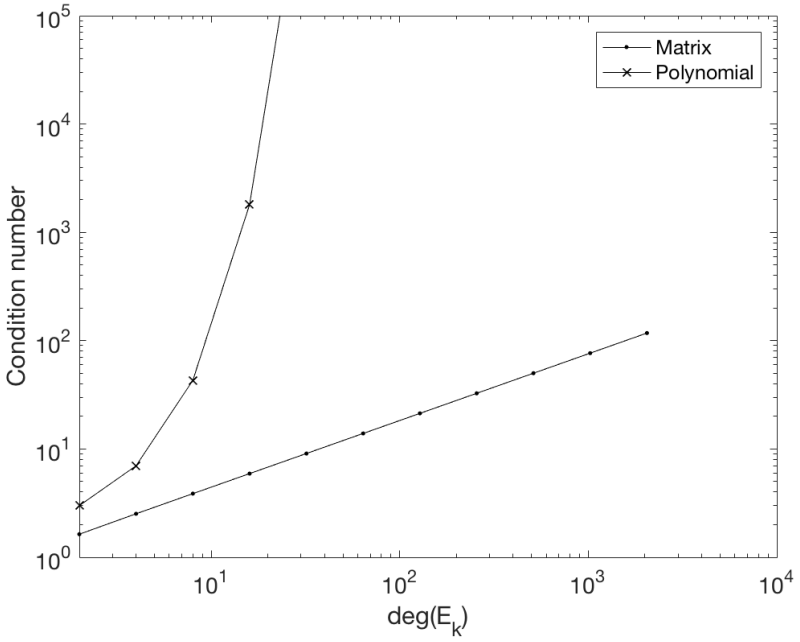


Figure 3.2: Log-log plot of condition numbers of the evaluation of the Euclid polynomials and the 2-norm condition number of their companions from $k = 2$ to $k = 12$. The computed slope for the condition number for the matrices is 0.618 giving an estimated condition number growth as $K_e \sim d^{0.618}$ which is better than the expected $O(d^2)$ behaviour [1]. The curious three digit coincidence with $(\sqrt{5}-1)/2$ is noted. The doubly exponential growth of the polynomial conditioning appears as exponential growth in this log log plot.

3.5 Conditioning of the eigenvalues of $\mathbb{E}_k$

Since the eigenvalues are all simple, $\mathbb{E}_k$ is diagonalizable and the condition number of each eigenvalue can be expressed using its unit left eigenvector y^H and unit right eigenvector x with $y^H\mathbb{E}_k = \lambda y^H$ and $\mathbb{E}_k x = \lambda x$, $\|x\| = \|y^H\| = 1$ and the condition number is

$$K_e = 1/(y^H x) \, .$$

We expect from our experience with random matrices that $K_e = O(d^2)$ where d is the dimension of the matrix, here the degree of the polynomial.

We can also look at the pseudospectra of the matrices that is, the eigenvalues of perturbed matrices [5]. Given an $\varepsilon > 0$, a pseudospectrum $\Lambda_\varepsilon(\mathbb{E}_6)$ is defined by

$$\Lambda_\varepsilon(\mathbb{E}_6) = \left\{ z \;\middle|\; \|(z\mathbf{I} - \mathbb{E}_6)^{-1}\|_2 \geq \frac{1}{\varepsilon} \right\} = \left\{ z \;\middle|\; \sigma_{\deg(E_6)}(z\mathbf{I} - \mathbb{E}_6) \leq \varepsilon \right\} \, .$$

Here $\sigma_{\deg(E_6)}$ is the smallest singular value of $z\mathbf{I} - \mathbb{E}_6$. The contour plot can then be created using

$$f(z) = \sigma_{\deg(E_6)}(z\mathbf{I} - \mathbb{E}_6) > 0 \, .$$

Figure 3.3c shows the pseudospectra of $\mathbb{E}_6$ for ten logarithmically-spaced values of ε between 10^{-2} and 10^{-1}.

To compare the conditioning of our companion matrices to the polynomials, we can also look at the pseudozeros of the polynomials. This allows us to look at the relationship between the condition number for the evaluation of polynomials and the condition number for rootfinding for polynomials [5]. The pseudozeros are defined as

$$\Lambda_\varepsilon\left(E_6(\lambda)\right) = \left\{\lambda \;\middle|\; |E_6(\lambda)| \leq \varepsilon \cdot B_6(\lambda)\right\}, \tag{3.2}$$

where $B_6(\lambda) = E_6(|\lambda|)$. Figure 3.3a shows the contour plot of the pseudozeros of $E_6(\lambda)$, and Figure 3.3b the pseudozeros of Euclid polynomial expanded about $\lambda = -1/2$, denoted as $E_6(u)$. Using the definition of the pseudozeros from equation (3.2), we can simply compute $|E_6(\lambda)|/E_6(|\lambda|)$ for various λ in the area of interest and use Matlab's `contour` function to create the contour lines.

We can see from these figures that the roots computed from the companion matrix are well-conditioned. That the spacing are similar in the two figures, when ε is so much smaller in Figure 3.3a demonstrates unequivocally that the eigenvalue problem is much better conditioned (a factor about 10^3). This factor grows exponentially, as shown in Figure 3.2. We consider that these figures are "similar" if

- there are circles around individual roots/eigenvalues,
- there are some regions surrounding merged roots/eigenvalues,
- spacing between contours in about 1% of the figure diameter.

3.6 Do we have to use matrices?

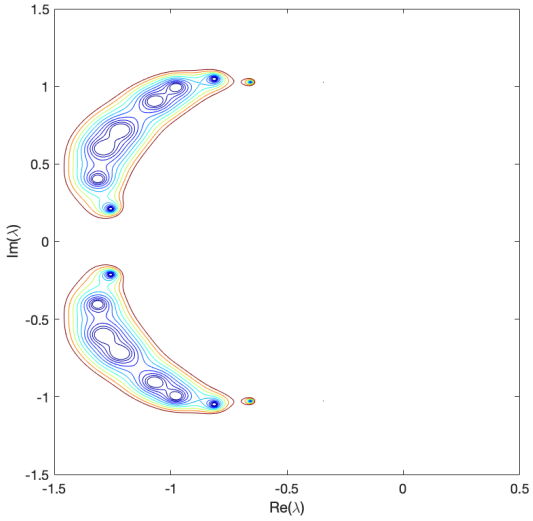
Expanding about $\lambda = -1/2$ is clearly better than expanding about $\lambda = 0$. Put $u = \lambda + 1/2$, and then

$$\begin{aligned}
E_1(\lambda) &= \lambda + 1 = u + \frac{1}{2} = E_1(u) \\
E_2(u) &= u^2 + \frac{3}{4} \\
E_3(u) &= u^4 + \frac{1}{2}u^2 + \frac{13}{16} \\
E_4(u) &= u^8 + u^6 + \frac{7}{8}u^4 + \frac{5}{16}u^2 + \frac{217}{256}
\end{aligned}$$

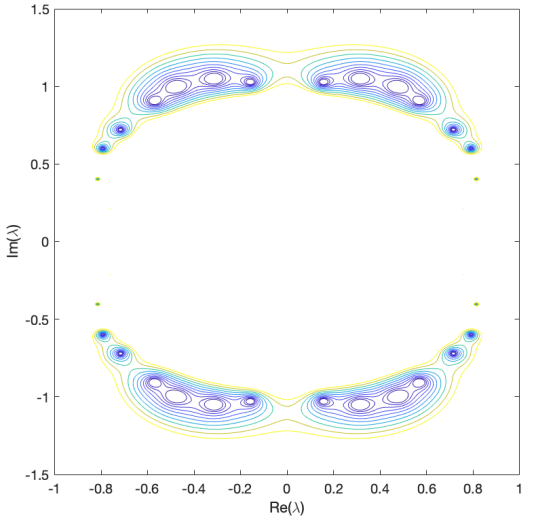
and these polynomials only have even powers (after $k = 1$); this makes the polynomials subject to only half as much rounding error because zero coefficients cannot (are not allowed to) be perturbed. More, the coefficients of the even order terms appears to grow more slowly.

However, they do still grow doubly exponentially with k (exponentially with the degree). The first polynomial to have a coefficient larger than 1 in magnitude is $E_5(u) = u^{16} + 2u^{14} + \cdots + 57073/65536$ and thereafter the repeated squaring gives runaway growth. We present the graphs of the condition numbers for the evaluation of $E_k(\lambda)$

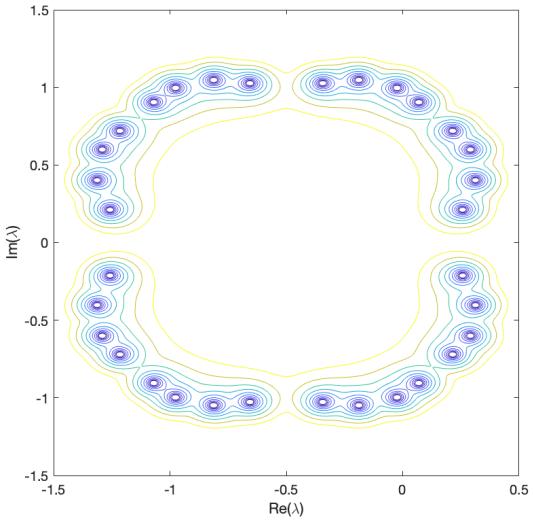
$$\widetilde{B}_k(u) = \sum_{j=0}^{\deg(E_k)} \left|v_j\right| |u|^j$$



(a) Pseudozeros of $E_6(\lambda)$ for 10 logarithmically-spaced values of ε between $10^{-9.5}$ and $10^{-8.5}$. The eigenvalues between $-1.5 \le \mathrm{Re}(\lambda) \le -0.5$ is quite ill-conditioned. We only change $E_6(\lambda)$ by $3 \times 10^{-6}\%$ at most.



(b) Pseudozeros of $E_6(u)$ for 10 logarithmically-spaced values of ε between 10^{-3} and 10^{-2}. This is substantially better-conditioned (and more symmetric) than the monomial basis (Figure 3.3a) changing $E_k(\lambda)$ by 1% at most.



(c) Pseudospectra of $\mathbb{E}_6$ for 10 logarithmically-spaced values of ε between 10^{-2} and 10^{-1}. This is the best-conditioned of the representations. This figure shows the results of changing $\mathbb{E}_6$ by 1–10%.

Figure 3.3: The similar spacings between Figures 3.3a, 3.3b and 3.3c demonstrate the superior conditioning of the companion matrix, owing to its minimal height.

on $0 \le u \le 1.1180$, a circle that contains the roots, in Figure 3.4. We see that for inside the interior of the cauliflower, this representation is well-conditioned (though uninteresting—nothing much is happening there) but near the boundary the exponential growth takes over.

We are forced to conclude that the minimal height companion matrices are exponentially better than these polynomials too.

Implicit in our discussion is the observation that the minimal height companion matrix is even more advantageous for larger k. The condition number for the evaluation of $E_k(\lambda)$ grows like E^{2^k}; the condition number for the evaluation of $E_k(u)$ grows like $E^{2^{k-1}}$ (possibly for a different E); while the condition number of $\mathbb{E}_k$'s eigenvalues grow only, as in Figure 3.2, like $\left(2^{k-1}\right)^{0.618}$. In practice, the pseudozeros/pseudospectra widths are already supporting this at $k = 6, 7, 8$, shown in Table 3.1.

k	$E_k(\lambda)$	$E_k(u)$	$\mathbb{E}_k$
6	$10^{-9.5} \ldots 10^{-8.5}$	$10^{-3} \ldots 10^{-2}$	$10^{-2} \ldots 10^{-1}$
7	$10^{-19.5} \ldots 10^{-18.5}$	$10^{-6} \ldots 10^{-5}$	$10^{-2} \ldots 10^{-1}$
8	$10^{-38.5} \ldots 10^{-37.5}$	$10^{-12} \ldots 10^{-11}$	$10^{-2} \ldots 10^{-1}$

Table 3.1: Pseudozeros/pseudospectra of $E_k(\lambda)$, $E_k(u)$ and $\mathbb{E}_k$ for $k = 6, 7, 8$. For these ε ranges, the pictures are similar to those of Figure 3.3. These pictures are available upon request.

Remark Using just the recurrence, not the polynomials, might be superior even to matrices.

3.7 Concluding remarks

For us, the Euclid polynomials showed that the construction of companion matrices by the method of Piers Lawrence was, in fact, general. This construction also gives a *minimal height* companion matrix (over the integers); trivially so, because height$(E_k) = 1$. This implies superior conditioning: already at $k = 6$, the matrix $\mathbb{E}_6$ has eigencondition about 1 while the polynomial $E_6(\lambda)$ had $B(\lambda) \sim 10^4$. But the other facts presented here show that the $E_k(\lambda)$ are themselves of interest: in particular, we're not done with the identity (for $\lambda > 0$)

$$\frac{1}{\lambda} = \sum_{k \ge 1} \frac{1}{E_k(\lambda)} \, .$$

Acknowledgment

We thank Donald E. Knuth for his interest, suggestions, and improved proof. We also thank J. Rafael Sendra, Juana Senda, and Laureano Gonzalez-Vega for their input and suggestions at the early stages of preparation of this paper. We also thank Dr. Susan Colley for her feedback. This work was supported by the Natural Sciences and Engineering Research Council of Canada and an Ontario Graduate Scholarship.

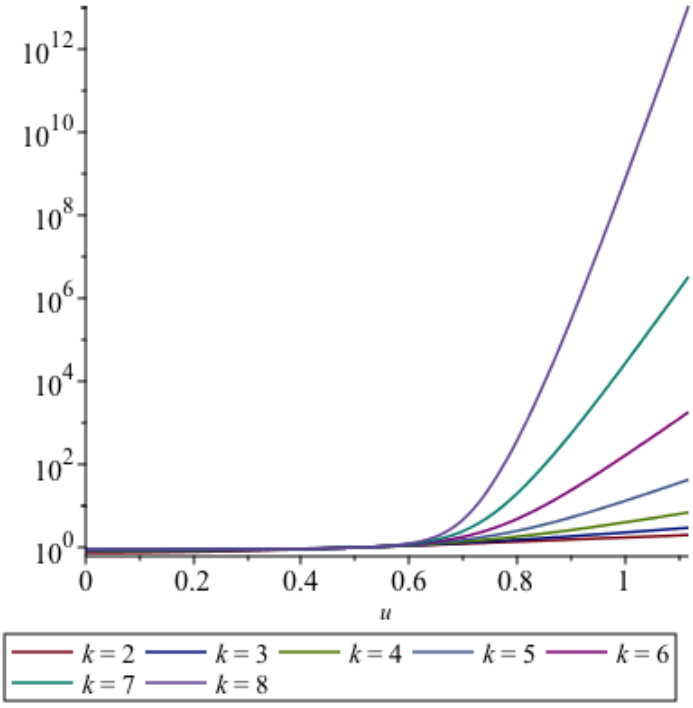


Figure 3.4: Condition numbers for the evaluation of $E_k(\lambda)\ \widetilde{B}_k(u)$ on $0 \le u \le 1.1180$ for $k = 2$ to 8.

Bibliography

[1] C. Beltrán and D. Armentano, *The polynomial eigenvalue problem is well conditioned for random inputs*, arXiv preprint arXiv:1706.06025, (2017).

[2] E. Y. S. Chan, *A comparison of solution methods for Mandelbrot-like polynomials*, Master's thesis, The University of Western Ontario, 2016.

[3] E. Y. S. Chan and R. M. Corless, *A new kind of companion matrix*, Electronic Journal of Linear Algebra, 32 (2017), pp. 335–342.

[4] E. Y. S. Chan, R. M. Corless, L. Gonzalez-Vega, J. R. Sendra, and J. Sendra, *Algebraic linearizations for matrix polynomials*, Linear Algebra and its Applications, 563 (2019), pp. 373–399.

[5] R. M. Corless and N. Fillion, *A graduate introduction to numerical methods*, AMC, 10 (2013), p. 12.

[6] R. T. Farouki and V. T. Rajan, *On the numerical condition of polynomials in Bernstein form*, Computer Aided Geometric Design, 4 (1987), pp. 191–216.

[7] R. L. Graham, D. E. Knuth, O. Patashnik, and S. Liu, *Concrete mathematics: a foundation for computer science*, Computers in Physics, 3 (1989), pp. 106–107.

[8] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge university press, 2012.

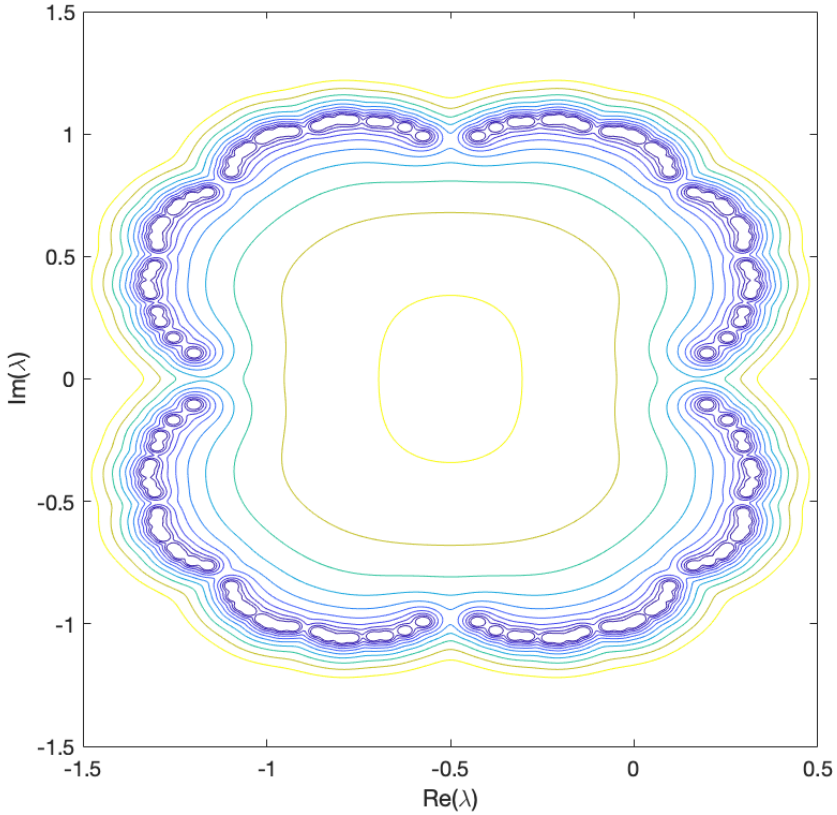


Figure 3.5: Pseudospectra of $\mathbb{E}_8$ for 10 logarithmically-spaced values of ε between 10^{-2} and 10^{-1}.

[9] G. A. KALUGIN AND D. J. JEFFREY, *Unimodal sequences show Lambert W is Bernstein*, arXiv preprint arXiv:1011.5940, (2010).

[10] C. D. MEYER, *Matrix analysis and applied linear algebra*, vol. 71, Siam, 2000.

Chapter 4

Algebraic linearizations of matrix polynomials

4.1 Introduction

Many applications require the computation or approximation of *polynomial eigenvalues*, that is, those $z \in \mathbb{C}$ for which the *matrix polynomial* $\mathbf{P}(z)$ (of degree at most s) $\in \mathbb{C}[z]^{r\times r}$ is singular. In other words, we search for z such that $\det \mathbf{P}(z) = 0$. If $s = 1$, that is $\mathbf{P}(z) = z\mathbf{B} - \mathbf{A}$, where $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{N\times N}$, where $N = r$, is degree 1 in z, *i.e. linear*, then this is "just" the generalized eigenvalue problem, which can be reliably solved numerically on many platforms using software developed over many decades by the efforts of many people. We do not here survey the state of the art of solving the generalized eigenvalue problem, *i.e.* determining z such that $\det(z\mathbf{B} - \mathbf{A}) = 0$ (provided the *pencil* $(\mathbf{A}, \mathbf{B})$ is *regular*, *i.e.* that $\det(z\mathbf{B} - \mathbf{A}) \not\equiv 0$). We do note that the so-called QZ iteration, which uses unitary transformations to simultaneously upper-triangularize $\mathbf{A}$ and $\mathbf{B}$ so that

$$\begin{aligned}\det(z\mathbf{B} - \mathbf{A}) &= \det \mathbf{Q} \det(z\mathbf{B} - \mathbf{A}) \det \mathbf{Z} \\ &= \det(z\mathbf{QBZ} - \mathbf{QAZ}) \\ &= \det(z\mathbf{T_B} - \mathbf{T_A})\end{aligned}$$

allows its eigenvalues to be read off from the corresponding diagonal entries of $\mathbf{T_B}$ and $\mathbf{T_A}$, is by now very well-developed and reliable. Research continues into making the method even faster and more reliable especially as novel architectures are invented and especially for matrix structures that arise frequently in practice. But in this paper we simply take such methods as given: we regard a linear matrix polynomial as one that is effectively solved. Thus, our task becomes one of reducing a more general matrix polynomial eigenproblem to a "mere" linear one. In this case, the dimension of the linear problem, N, is larger: $N \geq r \cdot s$ (remember the degree of $P(z)$ is at most s, and its dimension is r). This process is known as "linearization", naturally enough, although we note that the resulting problem, even if it is called "linear", is more properly considered as being of degree 2, once the unknown eigenvectors are considered: $z\mathbf{B}v = \mathbf{A}v$ is linear in the entries of v, and of z by itself, but terms like zv_1, zv_2, etc appear, which are really of degree two, in the language of computational algebra. Indeed reduction of *any* system of polynomial equations (if there are only a finite number of solutions, a situation

called "being zero-dimensional" in the literature) can always be "reduced" to a degree 2 system; this is known as the effective Nullstellensatz. Reduction to a generalized eigenproblem is a (very) practical concrete exhibition of this theorem.

Of course there are many practical details, that really matter. "In theory, there's no difference between theory and practice; but in practice, there is." One huge item of practical importance is the commonly-undertaken reduction to upper Hessenberg form, prior to beginning the QZ iteration; this can be stably done in $O(N^2)$ operations and greatly speeds up the iterations subsequently.

Other possibilities exist than linearization. Indeed there is much current research into what is called "ℓ-ification," *i.e.* reduction of a matrix polynomial of degree $m\ell$ to a (larger) matrix polynomial of degree at most ℓ (having degree at most ℓ is also called "having *grade* ℓ") [9]. But here we restrict ourselves to the case $\ell = 1$.

Surprisingly, there are still things to be said about this, in spite of many decades of work by many people. Of course, the proofs in this paper rely heavily on that work, especially that summarized in the classic [10]. But still we will see some new elements, at least for a particular class of problems.

A useful introduction to the general area can be found in [12, pages 263–281] and the references therein. Early history is discussed in [17]. Major recent works include [16] and [15].

4.2 The basic idea

The basic idea of the algebraic linearizations described here was first discovered in the context of what are called "Mandelbrot polynomials" [3, 8]. Mandelbrot polynomials are defined by $p_0 = 0$ and $p_{n+1} = zp_n^2 + 1$. Piers Lawrence found matrices $\mathbf{M}_n$, populated only by elements 0 or -1, with $p_n(z) = \det(z\mathbf{I} - \mathbf{M}_n)$. Naturally enough, these were called Mandelbrot matrices. We outline their construction below.

The first few p_n are $p_0 = 0$, $p_1 = 1$, $p_2 = z+1$, and $p_3 = z^3 + 2z^2 + z + 1$. The idea is clearest going from $\mathbf{M}_3$ to $\mathbf{M}_4$; we will build up to that. Because the only root of p_2 is $z = -1$, clearly $\mathbf{M}_2 = [-1]$, a 1×1 matrix with eigenvalue -1. To make $\mathbf{M}_3$, glue two copies of $\mathbf{M}_2$ together to make

$$\mathbf{M}_3 = \left[\begin{array}{c|c|c} -1 & 0 & -1 \\ \hline -1 & 0 & 0 \\ \hline 0 & -1 & -1 \end{array}\right]$$

which one can directly verify has

$$\begin{aligned} \det(z\mathbf{I} - \mathbf{M}_3) &= \det\begin{pmatrix} z+1 & 0 & 1 \\ 1 & z & 0 \\ & 1 & z+1 \end{pmatrix} \\ &= (z+1)\det\begin{pmatrix} z & 0 \\ 1 & z+1 \end{pmatrix} + 1 \cdot \det\begin{pmatrix} 1 & z \\ & 1 \end{pmatrix} \\ &= z(z+1)^2 + 1\,, \text{ as desired.} \end{aligned}$$

To make $\mathbf{M}_4$ we glue two copies of $\mathbf{M}_3$ together:

$$\begin{bmatrix} -1 & 0 & -1 & & & -1 \\ -1 & 0 & 0 & & & \\ & -1 & -1 & & & \\ & & -1 & 0 & & \\ & & & -1 & -1 & 0 & -1 \\ & & & & -1 & 0 & 0 \\ & & & & & -1 & -1 \end{bmatrix}$$

and at this level the "glue" and the "copies" are more distingushable. The upper Hessenberg nature of the matrix is also visible. To prove $p_4 = \det(z\mathbf{I} - \mathbf{M}_4)$ we use Knuth's idea: the determinant is linear in the first row:

$$\begin{aligned} &\det(z\mathbf{I} - \mathbf{M}_4) \\ &= \det\begin{pmatrix} z\mathbf{I} - \mathbf{M}_3 & & \\ 1 & z & \\ & 1 & z\mathbf{I} - \mathbf{M}_3 \end{pmatrix} + \det\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{pmatrix} \\ &= zp_3^2 + 1 \cdot \det\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \\ &= zp_3^2 + 1\,, \text{ as desired.} \end{aligned}$$

This gives the idea. The generalization will be Theorem 4.3.4 in the next section.

For more on Mandelbrot matrices, see [8], [3], and [4]. They and their generalizations have some interesting properties. For now, note that [4] generalized the construction to finding a companion for the scalar polynomial $c = z\mathbf{ab} + \mathbf{c}_0$ given upper Hessenberg companions for $\mathbf{a}$ and $\mathbf{b}$. It is that generalization that we turn into a linearization in the next section.

4.3 The main theorems

Theorem 4.3.1 shows how to linearize

$$\mathbf{e}_1(z) = z\mathbf{d}_0\mathbf{a}(z) + \mathbf{c}_0 \tag{4.1}$$

and

$$\mathbf{e}_2(z) = z\mathbf{a}(z)\mathbf{d}_0 + \mathbf{c}_0\,, \tag{4.2}$$

where $\mathbf{e}_1(z), \mathbf{e}_2(z) \in \mathbb{C}^{r\times r}$, once linearization for $\mathbf{a}(z)$ is available.

$$\Lambda(\mathbf{a}(z)) := \{z \mid \det(\mathbf{a}(z)) = 0\}$$

is the spectrum of the matrix polynomial $\mathbf{a}(z) \in \mathbb{C}^{r\times r}$. These z are the polynomial eigenvalues of $\mathbf{a}(z)$.

Theorem 4.3.1 *Consider* $\mathbf{e}_1(z)$ *and* $\mathbf{e}_2(z)$ *as in equations* (4.1) *and* (4.2)*, respectively. Suppose* $\mathbf{a}(z) \in \mathbb{C}[z]^{r\times r}$ *is of degree* $s \geq 1$*,* $\mathbf{c}_0$ *and* $\mathbf{d}_0 \in \mathbb{C}^{r\times r}$*, and that* $\mathbf{a}(z)$ *has the regular linearization pencil* $(\mathbf{D_A}, \mathbf{A})$ *with* $\det \mathbf{a}(z) = \det(z\mathbf{D_A} - \mathbf{A})$ *and* $z\mathbf{D_A} - \mathbf{A}$ *invertible except when* $z \in \Lambda(\mathbf{a})$ *which is a discrete set. Moreover suppose that we have the resolvent form*

$$\mathbf{a}^{-1}(z) = \mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A} \qquad z \in \mathbb{C} \notin \Lambda(\mathbf{a})$$

and $\mathbf{X_A} \in \mathbb{C}^{r\times rs}$ *and* $\mathbf{Y_A} \in \mathbb{C}^{rs\times r}$ *are known. Then if*

$$\mathbf{E}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{c}_0\mathbf{X_A} \\ -\mathbf{Y_A} & \mathbf{A} \end{bmatrix}, \quad \mathbf{D}_{\mathbf{E}_1} = \begin{bmatrix} \mathbf{d}_0 & \\ & \mathbf{D_A} \end{bmatrix}$$

and

$$\mathbf{E}_2 = \begin{bmatrix} \mathbf{A} & \mathbf{Y_A}\mathbf{c}_0 \\ -\mathbf{X_A} & \mathbf{0} \end{bmatrix}, \quad \mathbf{D}_{\mathbf{E}_2} = \begin{bmatrix} \mathbf{D_A} & \\ & \mathbf{d}_0 \end{bmatrix}$$

then $\det(z\mathbf{D}_{\mathbf{E}_1} - \mathbf{E}_1) = \det \mathbf{e}_1(z)$ *where* $\mathbf{e}_1(z) = z\mathbf{d}_0\mathbf{a}(z) + \mathbf{c}_0$ *and* $\det(z\mathbf{D}_{\mathbf{E}_2} - \mathbf{E}_2) = \det \mathbf{e}_2(z)$ *where* $\mathbf{e}_2(z) = z\mathbf{a}(z)\mathbf{d}_0 + \mathbf{c}_0$*. Moreoever*

$$\begin{bmatrix} \mathbf{0} & -\mathbf{X_A} \end{bmatrix}(z\mathbf{D}_{\mathbf{E}_1} - \mathbf{E}_1)^{-1}\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} = \mathbf{e}_1^{-1}(z)$$

and

$$\begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}(z\mathbf{D}_{\mathbf{E}_2} - \mathbf{E}_2)^{-1}\begin{bmatrix} -\mathbf{Y_A} \\ \mathbf{0} \end{bmatrix} = \mathbf{e}_2^{-1}(z)$$

give resolvent forms for the larger systems.

Proof *We use the Schur factoring [13, Chapter 12]:*

$$\begin{aligned}
&z\mathbf{D}_{\mathbf{E}_1} - \mathbf{E}_1 \\
&= \begin{bmatrix} z\mathbf{d}_0 & -\mathbf{c}_0\mathbf{X_A} \\ \mathbf{Y_A} & z\mathbf{D_A} - \mathbf{A} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{I} & -\mathbf{c}_0\mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{A})^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{S_A} & \mathbf{0} \\ \mathbf{Y_A} & z\mathbf{D_A} - \mathbf{A} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{I} & -\mathbf{c}_0\mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{A})^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\begin{bmatrix} z\mathbf{d}_0 + \mathbf{c}_0\mathbf{a}^{-1}(z) & \mathbf{0} \\ \mathbf{Y_A} & z\mathbf{D_A} - \mathbf{A} \end{bmatrix} \\
&= \mathbf{P}_1\mathbf{P}_2\,,
\end{aligned}$$

where the Schur complement $\mathbf{S_A} = z\mathbf{d}_0 + \mathbf{c}_0\mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}$*. Thus*

$$\begin{aligned}
\det(z\mathbf{D}_{\mathbf{E}_1} - \mathbf{E}_1) &= \det(\mathbf{P}_1)\det(\mathbf{P}_2) \\
&= \det\left(z\mathbf{d}_0 + \mathbf{c}_0\mathbf{a}^{-1}(z)\right)\det(z\mathbf{D_A} - \mathbf{A}) \\
&= \det\left(z\mathbf{d}_0 + \mathbf{c}_0\mathbf{a}^{-1}(z)\right)\det\mathbf{a}(z) \\
&= \det(z\mathbf{d}_0\mathbf{a}(z) + \mathbf{c}_0) \\
&= \det(\mathbf{e}_1(z))\,,
\end{aligned}$$

as desired. Moreover, $(z\mathbf{D}_{\mathbf{E}_1} - \mathbf{E}_1)^{-1} = \mathbf{P}_2^{-1}\mathbf{P}_1^{-1}$. *Let*

$$\begin{aligned}\mathbf{Q}_a &= \begin{bmatrix} \mathbf{I} & \\ & (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Y}_\mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \left(\mathbf{e}_1(z)\mathbf{a}^{-1}(z)\right)^{-1} & \\ & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{a}(z)\mathbf{e}_1^{-1}(z) & \mathbf{0} \\ -(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\mathbf{Y}_\mathbf{A}\mathbf{a}(z)\mathbf{e}_1^{-1}(z) & (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} \end{bmatrix}\end{aligned}$$

and

$$\mathbf{Q}_b = \begin{bmatrix} \mathbf{I} & \mathbf{c}_0\mathbf{X}_\mathbf{A}(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Then,

$$\begin{aligned}\mathbf{P}_2^{-1} &= \mathbf{Q}_a\mathbf{Q}_b \\ &= \begin{bmatrix} \mathbf{a}(z)\mathbf{e}_1^{-1}(z) & \mathbf{Q}_c \\ -(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\mathbf{Y}_\mathbf{A}\mathbf{a}(z)\mathbf{e}_1^{-1}(z) & \mathbf{Q}_d \end{bmatrix},\end{aligned}$$

where

$$\mathbf{Q}_c = \mathbf{a}(z)\mathbf{e}_1^{-1}(z)\mathbf{c}_0\mathbf{X}_\mathbf{A}(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}$$

and

$$\mathbf{Q}_d = (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} - (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\mathbf{Y}_\mathbf{A}\mathbf{a}(z)\mathbf{e}_1^{-1}(z)\mathbf{c}_0\mathbf{X}_\mathbf{A}(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1},$$

so

$$\begin{aligned}\begin{bmatrix} \mathbf{0} & -\mathbf{X}_\mathbf{A} \end{bmatrix}(z\mathbf{D}_{\mathbf{E}_1} - \mathbf{E}_1)^{-1}\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} &= \mathbf{X}_\mathbf{A}(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\mathbf{Y}_\mathbf{A}\mathbf{a}(z)\mathbf{e}_1^{-1}(z) \\ &= \mathbf{a}^{-1}(z)\mathbf{a}(z)\mathbf{e}_1^{-1}(z) \\ &= \mathbf{e}_1^{-1}(z)\end{aligned}$$

as claimed.

Similarly,

$$\begin{aligned}z\mathbf{D}_{\mathbf{E}_2} - \mathbf{E}_2 &= \begin{bmatrix} z\mathbf{D}_\mathbf{A} - \mathbf{A} & -\mathbf{Y}_\mathbf{A}\mathbf{c}_0 \\ \mathbf{X}_\mathbf{A} & z\mathbf{d}_0 \end{bmatrix} \\ &= \mathbf{Q}_e\mathbf{Q}_f,\end{aligned}$$

where

$$\mathbf{Q}_e = \begin{bmatrix} z\mathbf{D}_\mathbf{A} - \mathbf{A} & \mathbf{0} \\ \mathbf{X}_\mathbf{A} & z\mathbf{d}_0 + \mathbf{X}_\mathbf{A}(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\mathbf{Y}_\mathbf{A}\mathbf{c}_0 \end{bmatrix}$$

and

$$\mathbf{Q}_f = \begin{bmatrix} \mathbf{I} & -(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\mathbf{Y}_\mathbf{A}\mathbf{c}_0 \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

so

$$\begin{aligned}\det(z\mathbf{D}_{\mathbf{E}_2} - \mathbf{E}_2) &= \det(z\mathbf{D}_\mathbf{A} - \mathbf{A})\det\left(z\mathbf{d}_0 + \mathbf{a}^{-1}(z)\mathbf{c}_0\right) \\ &= \det\mathbf{a}(z)\det\left(z\mathbf{d}_0 + \mathbf{a}^{-1}(z)\mathbf{c}_0\right) \\ &= \det(z\mathbf{a}(z)\mathbf{d}_0 + \mathbf{c}_0) \\ &= \det\mathbf{e}_2(z)\end{aligned}$$

as claimed.

Moreover

$$(z\mathbf{D}_{\mathbf{E}_2} - \mathbf{E}_2)^{-1} = \mathbf{Q}_g\mathbf{Q}_h\,,$$

where

$$\begin{aligned}\mathbf{Q}_g &= \begin{bmatrix} \mathbf{I} & (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\,\mathbf{Y}_\mathbf{A}\mathbf{c}_0 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \\ & \mathbf{e}_2^{-1}(z)\mathbf{a}(z) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\,\mathbf{Y}_\mathbf{A}\mathbf{c}_0\mathbf{e}_2^{-1}(z)\mathbf{a}(z) \\ \mathbf{0} & \mathbf{e}_2^{-1}(z)\mathbf{a}(z) \end{bmatrix}\end{aligned}$$

and

$$\begin{aligned}\mathbf{Q}_h &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{X}_\mathbf{A} & \mathbf{I} \end{bmatrix}\begin{bmatrix} (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} & \\ & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} & \mathbf{0} \\ -\mathbf{X}_\mathbf{A}\,(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} & \mathbf{I} \end{bmatrix},\end{aligned}$$

which results in

$$\begin{bmatrix} \mathbf{Q}_i & (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\,\mathbf{Y}_\mathbf{A}\mathbf{c}_0\mathbf{e}_2^{-1}(z)\mathbf{a}(z) \\ -\mathbf{e}_2^{-1}(z)\mathbf{a}(z)\mathbf{X}_\mathbf{A}\,(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} & \mathbf{e}_2^{-1}(z)\mathbf{a}(z) \end{bmatrix},$$

where

$$\mathbf{Q}_i = (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} - (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\,\mathbf{Y}_\mathbf{A}\mathbf{c}_0\mathbf{e}_2^{-1}(z)\mathbf{a}(z)\mathbf{X}_\mathbf{A}\,(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\ .$$

Therefore,

$$\begin{aligned}\begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}(z\mathbf{D}_2 - \mathbf{E}_2)^{-1}\begin{bmatrix} -\mathbf{Y}_\mathbf{A} \\ \mathbf{0} \end{bmatrix} &= \mathbf{e}_2^{-1}(z)\mathbf{a}(z)\mathbf{X}_\mathbf{A}\,(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1}\,\mathbf{Y}_\mathbf{A} \\ &= \mathbf{e}_2^{-1}(z)\end{aligned}$$

as claimed. ♮

Theorem 4.3.2 shows how to linearize a product $\mathbf{a}(z)\mathbf{b}(z)$ given linearizations of each of $\mathbf{a}(z)$ and $\mathbf{b}(z)$.

Theorem 4.3.2 *Suppose* $\mathbf{a}(z)$, $\mathbf{D}_\mathbf{A}$, $\mathbf{A}$, $\mathbf{X}_\mathbf{A}$, *and* $\mathbf{Y}_\mathbf{A}$ *are as in Theorem 4.3.1, and suppose similarly that* $\mathbf{b}(z) \in \mathbb{C}[z]^{r\times r}$ *is of degree* $t \geq 1$, *has the regular linearization pencil* $(\mathbf{D}_\mathbf{B}, \mathbf{B})$ *with* $\det\mathbf{b}(z) = \det(z\mathbf{D}_\mathbf{B} - \mathbf{B})$ *and resolvent*

$$\mathbf{b}^{-1}(z) = \mathbf{X}_\mathbf{B}\,(z\mathbf{D}_\mathbf{B} - \mathbf{B})^{-1}\,\mathbf{Y}_\mathbf{B} \qquad \textit{for } z \in \mathbb{C} \notin \Lambda(\mathbf{b})$$

Then if we define

$$\mathbf{F}_1 = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{Y}_\mathbf{B}\mathbf{X}_\mathbf{A} & \mathbf{B} \end{bmatrix} \quad \text{and} \quad \mathbf{D}_{\mathbf{F}_1} = \begin{bmatrix} \mathbf{D}_\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_\mathbf{B} \end{bmatrix} \tag{4.3}$$

or similarly

$$\mathbf{F}_2 = \begin{bmatrix} \mathbf{B} & \mathbf{Y}_\mathbf{B}\mathbf{X}_\mathbf{A} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \quad \text{and} \quad \mathbf{D}_{\mathbf{F}_2} = \begin{bmatrix} \mathbf{D}_\mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_\mathbf{A} \end{bmatrix}, \tag{4.4}$$

then $z\mathbf{D}_{\mathbf{F}_1} - \mathbf{F}_1$ and $z\mathbf{D}_{\mathbf{F}_2} - \mathbf{F}_2$ are linearizations for $\mathbf{a}(z)\mathbf{b}(z)$.

Proof Consider $\mathbf{F}_1$, $\mathbf{D}_{\mathbf{F}_1}$, $\mathbf{F}_2$, and $\mathbf{D}_{\mathbf{F}_2}$ as in equations (4.3) and (4.4) shown above. Clearly $\mathbf{A}$ and $\mathbf{B}$ can be exchanged in either factor to get new but related constructions. Then

$$\begin{aligned}\det(z\mathbf{D}_{\mathbf{F}_1} - \mathbf{F}_1) &= \det\begin{pmatrix} z\mathbf{D}_\mathbf{A} - \mathbf{A} & \mathbf{0} \\ -\mathbf{Y}_\mathbf{B}\mathbf{X}_\mathbf{A} & z\mathbf{D}_\mathbf{B} - \mathbf{B} \end{pmatrix} \\ &= \det\mathbf{a}(z)\det\mathbf{b}(z) \\ &= \det\mathbf{a}(z)\mathbf{b}(z)\end{aligned}$$

and moreover

$$\begin{aligned}&(z\mathbf{D}_{\mathbf{F}_1} - \mathbf{F}_1)^{-1} = \\ &\begin{bmatrix} (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} & \mathbf{0} \\ (z\mathbf{D}_\mathbf{B} - \mathbf{B})^{-1}\mathbf{Y}_\mathbf{B}\mathbf{X}_\mathbf{A}(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} & (z\mathbf{D}_\mathbf{B} - \mathbf{B})^{-1} \end{bmatrix}\end{aligned}$$

so

$$\begin{bmatrix} \mathbf{0} & \mathbf{X}_\mathbf{B} \end{bmatrix}(z\mathbf{D}_{\mathbf{F}_1} - \mathbf{F}_1)^{-1}\begin{bmatrix} \mathbf{Y}_\mathbf{A} \\ \mathbf{0} \end{bmatrix} = \mathbf{b}^{-1}(z)\mathbf{a}^{-1}(z)\,.$$

Reversing $\mathbf{A}$ and $\mathbf{B}$ in $\mathbf{F}_1$ gives instead $\mathbf{a}^{-1}(z)\mathbf{b}^{-1}(z)$. Similarly

$$z\mathbf{D}_{\mathbf{F}_2} - \mathbf{F}_2 = \begin{bmatrix} z\mathbf{D}_\mathbf{B} - \mathbf{B} & -\mathbf{Y}_\mathbf{B}\mathbf{X}_\mathbf{A} \\ \mathbf{0} & z\mathbf{D}_\mathbf{A} - \mathbf{A} \end{bmatrix}$$

so again $\det(z\mathbf{D}_{\mathbf{F}_2} - \mathbf{F}_2) = \det\mathbf{b}(z)\det\mathbf{a}(z) = \det(\mathbf{b}(z)\mathbf{a}(z))$. Moreover

$$\begin{aligned}&(z\mathbf{D}_{\mathbf{F}_2} - \mathbf{F}_2)^{-1} = \\ &\begin{bmatrix} (z\mathbf{D}_\mathbf{B} - \mathbf{B})^{-1} & (z\mathbf{D}_\mathbf{B} - \mathbf{B})^{-1}\mathbf{Y}_\mathbf{B}\mathbf{X}_\mathbf{A}(z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} \\ \mathbf{0} & (z\mathbf{D}_\mathbf{A} - \mathbf{A})^{-1} \end{bmatrix}\end{aligned}$$

so

$$\begin{bmatrix} \mathbf{X}_\mathbf{B} & \mathbf{0} \end{bmatrix}(z\mathbf{D}_{\mathbf{F}_2} - \mathbf{F}_2)^{-1}\begin{bmatrix} \mathbf{0} \\ \mathbf{Y}_\mathbf{A} \end{bmatrix} = \mathbf{b}^{-1}(z)\mathbf{a}^{-1}(z)\,.$$

∎

Remark Theorem 4.3.2 is just Theorem 3.2 from [10, p. 85] with two minor modifications: non-monic $\mathbf{a}(z)$ is covered here, and we will use $\mathbf{F}_2$ to give a block upper Hessenberg matrix whereas they use $\mathbf{F}_1$. That seems paradoxical because $\mathbf{F}_1$ looks more likely to generate block upper Hessenberg matrices, but when used recursively the lower left triangle remains empty when $\mathbf{X}_\mathbf{A}$ and $\mathbf{Y}_\mathbf{B}$ are $e_s^T \otimes I_r$ and $f_1 \otimes I_r$. We will need the upper right block for the constant coefficient added.

Theorems 4.3.3 and 4.3.4 show how to linearize $\mathbf{a}(z) + \mathbf{c}(z)$ if $\deg(\mathbf{c}(z)) < \deg(\mathbf{a}(z))$. Theorem 4.3.3 considers the monic case for $\mathbf{a}(z)$, and Theorem 4.3.4 relaxes this restriction.

Theorem 4.3.3 (monic case) *Suppose* $\mathbf{a}(z) = z^s + \boldsymbol{\alpha}_{s-1}z^{s-1} + \cdots + \boldsymbol{\alpha}_0$ *and each* $\boldsymbol{\alpha}_k \in \mathbb{C}^{r\times r}$*, and that we have a block upper Hessenberg linearization* $\mathbf{A}$ *of* $\mathbf{a}(z)$ *with standard triple* $\mathbf{X}_\mathbf{A}$, $\mathbf{A}$, $\mathbf{Y}_\mathbf{A}$ *which means among other things that*

$$\mathbf{X}_\mathbf{A}(z\mathbf{I}_{sr} - \mathbf{A})^{-1}\mathbf{Y}_\mathbf{A} = \mathbf{a}^{-1}(z)\,.$$

Then if $\mathbf{c}(z) = \mathbf{c}_{s-1}z^{s-1} + \cdots + \mathbf{c}_1 z + \mathbf{c}_0$*, with each* $\mathbf{c}_i \in \mathbb{C}^{r\times r}$*, is of degree at most* $s-1$*, then*

$$\mathbf{G} = \mathbf{A} - \sum_{k=0}^{s-1} \mathbf{A}^k \mathbf{Y}_\mathbf{A} \mathbf{c}_k \mathbf{X}_\mathbf{A}$$

is a block upper Hessenberg linearization of $\mathbf{a}(z) + \mathbf{c}(z)$*, with*

$$\mathbf{X}_\mathbf{A} \left(z\mathbf{I} - \mathbf{G}\right)^{-1} \mathbf{Y}_\mathbf{A} = \left(\mathbf{a}(z) + \mathbf{c}(z)\right)^{-1} .$$

Proof Using the properties of a standard triple [10, see Proposition 2.1 (i), p 53] the matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{Y}_\mathbf{A} & \mathbf{A}\mathbf{Y}_\mathbf{A} & \mathbf{A}^2\mathbf{Y}_\mathbf{A} & \cdots & \mathbf{A}^{s-1}\mathbf{Y}_\mathbf{A} \end{bmatrix}$$

is nonsingular. Put $\mathbf{V}_k = \mathbf{A}^{k-1}\mathbf{Y}_\mathbf{A}$ for $1 \le k \le s$. Note each $\mathbf{V}_k$ is sr by r. Then direct computation shows

$$\mathbf{A}\begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \ldots & \mathbf{V}_s \end{bmatrix} = \begin{bmatrix} \mathbf{V}_2 & \mathbf{V}_3 & \cdots & \mathbf{V}_s & \mathbf{A}^s\mathbf{Y}_\mathbf{A} \end{bmatrix}$$

By part (iii) of the previously mentioned proposition,

$$\mathbf{A}^s\mathbf{Y}_\mathbf{A} = -\sum_{k=1}^{s} \mathbf{A}^{k-1}\mathbf{Y}_\mathbf{A}\boldsymbol{\alpha}_{k-1} ,$$

meaning that the given matrix polynomial is "solved" by its linearization times $\mathbf{Y}_\mathbf{A}$ (a generalization of the Cayley-Hamilton theorem).

Thus

$$\mathbf{A}\mathbf{V} = \mathbf{V}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -\boldsymbol{\alpha}_0 \\ \mathbf{I} & \mathbf{0} & & & -\boldsymbol{\alpha}_1 \\ & \mathbf{I} & \ddots & & \vdots \\ & & \ddots & \mathbf{0} & -\boldsymbol{\alpha}_{s-2} \\ & & & \mathbf{I} & -\boldsymbol{\alpha}_{s-1} \end{bmatrix} = \mathbf{V}\mathbf{C}_2$$

where $\mathbf{C}_2$ is the familiar "second companion linearization", making explicit the similarity $\mathbf{A} = \mathbf{V}\mathbf{C}_2\mathbf{V}^{-1}$. Quite clearly the second companion linearization of $\mathbf{a}(z) + \mathbf{c}(z)$ is

$$\begin{bmatrix} \mathbf{0} & & & -(\boldsymbol{\alpha}_0 + \mathbf{c}_0) \\ \mathbf{I} & \mathbf{0} & & -(\boldsymbol{\alpha}_1 + \mathbf{c}_1) \\ & \mathbf{I} & & \vdots \\ & & \ddots & \\ & & & \mathbf{I} - (\boldsymbol{\alpha}_{s-1} + \mathbf{c}_{s-1}) \end{bmatrix}$$

and we look for a matrix $\mathbf{W}$ such that $\mathbf{A} + \mathbf{W}$ linearizes $\mathbf{a}(z) + \mathbf{c}(z)$. Now

$$\mathbf{W}\mathbf{V} = \mathbf{V}\Delta\mathbf{C}_2 = \mathbf{V}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{c}_0 \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & -\mathbf{c}_1 \\ \vdots & \ddots & & \vdots & \vdots \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & -\mathbf{c}_{s-1} \end{bmatrix}$$

implies

$$\begin{aligned}\mathbf{W} &= \mathbf{V}\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{c}_0 \\ \vdots & & \vdots & -\mathbf{c}_1 \\ \vdots & & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{c}_{s-1} \end{bmatrix}\mathbf{V}^{-1} \\ &= \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \\ \vdots & & \vdots & -\sum_{k=1}^{s}\mathbf{V}_k\mathbf{c}_{k-1} \\ \mathbf{0} & \cdots & \mathbf{0} & \end{bmatrix}\mathbf{V}^{-1} \\ &= -\sum_{k=1}^{s}\mathbf{V}_k\mathbf{c}_{k-1}\mathbf{X}_\mathbf{A} = -\sum_{k=1}^{s}\mathbf{A}^{k-1}\mathbf{Y}_\mathbf{A}\mathbf{c}_{k-1}\mathbf{X}_\mathbf{A}\end{aligned}$$

as desired, because property (ii) of Proposition 2.1 in [10] has $\mathbf{X}_\mathbf{A}$ uniquely defined as $\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix}\cdot\mathbf{V}^{-1}$ in our notation. This proves the theorem. ♮

Theorem 4.3.4 (non-monic case) *Suppose* $\mathbf{a}(z) = \alpha_s z^s + \alpha_{s-1} + \cdots + \alpha_0$ *and* α_s *might be singular. Suppose that we have a block upper Hessenberg generalized linearization* $(\mathbf{A}, \mathbf{D}_\mathbf{A})$*—that is,* $\mathbf{A}$ *is block upper Hessenberg,* $\mathbf{D}_\mathbf{A}$ *is block diagonal, each with* $r \times r$ *blocks, and that we have the generalized standard triple,*

$$\mathbf{X}_\mathbf{A}\left(z\mathbf{D}_\mathbf{A} - \mathbf{A}\right)^{-1}\mathbf{D}_\mathbf{A}\mathbf{Y}_\mathbf{A} = \mathbf{a}^{-1}(z)\,.$$

Then if $\mathbf{c}(z) = \sum_{k=0}^{s-1}\mathbf{c}_k z^k$*, with each* $\mathbf{c}_i \in \mathbb{C}^{r\times r}$*, is of degree at most* $s-1$*, then*

$$\mathbf{G} = \mathbf{A} - \sum_{k=0}^{s-1}\mathbf{A}^k\mathbf{Y}_\mathbf{A}\mathbf{c}_k\mathbf{X}_\mathbf{A}$$

is a block upper Hessenberg linearization of $\mathbf{a}(z) + \mathbf{c}(z)$*, with*

$$\mathbf{X}_\mathbf{A}\left(z\mathbf{D}_\mathbf{A} - \mathbf{G}\right)^{-1}\mathbf{D}_\mathbf{A}\mathbf{Y}_\mathbf{A} = \left(\mathbf{a}(z) + \mathbf{c}(z)\right)^{-1}\,.$$

Proof If α_s is singular, this also means that $\mathbf{D}_\mathbf{A}$ will be singular. To find the resolvent form, we can perturb the matrix polynomial: $\mathbf{a}(z) + \varepsilon\Delta\mathbf{a}(z,\varepsilon)$, which we will define as perturbing just α_s. The generalized linearization for this new matrix polynomial is $(\mathbf{A}, \mathbf{D}_\mathbf{A} + \varepsilon\mathbf{I})$ (which defines $\Delta\mathbf{a}\,(z,\varepsilon)$ implicitly) and the standard triple is

$$\left(\mathbf{X}_\mathbf{A}, \left(\mathbf{D}_\mathbf{A} + \varepsilon\mathbf{I}\right)^{-1}\mathbf{A}, \mathbf{Y}_\mathbf{A}\right)$$

which gives the resolvent form

$$\begin{aligned}\left(\mathbf{a}(z) + \varepsilon\Delta\mathbf{a}(z,\varepsilon)\right)^{-1} &= \mathbf{X}_\mathbf{A}\left(z\mathbf{I} - \left(\mathbf{D}_\mathbf{A} + \varepsilon\mathbf{I}\right)^{-1}\mathbf{A}\right)^{-1}\mathbf{Y}_\mathbf{A} \\ &= \mathbf{X}_\mathbf{A}\left(\left(\mathbf{D}_\mathbf{A} + \varepsilon\mathbf{I}\right)^{-1}\left(z\left(\mathbf{D}_\mathbf{A} + \varepsilon\mathbf{I}\right) - \mathbf{A}\right)\right)^{-1}\mathbf{Y}_\mathbf{A} \\ &= \mathbf{X}_\mathbf{A}\left(z\left(\mathbf{D}_\mathbf{A} + \varepsilon\mathbf{I}\right) - \mathbf{A}\right)^{-1}\left(\mathbf{D}_\mathbf{A} + \varepsilon\mathbf{I}\right)\mathbf{Y}_\mathbf{A}\,.\end{aligned}$$

As $\varepsilon \to 0$,

$$\mathbf{a}^{-1}(z) = \mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{D_A}\mathbf{Y_A}\ .$$

Then using the proof from Theorem 4.3.3, we find that

$$\mathbf{G} = \mathbf{A} - \sum_{k=1}^{s} \mathbf{A}^{k-1}\mathbf{Y_A}\mathbf{c}_{k-1}\mathbf{X_A}$$

is, again, the block upper Hessenberg linearization of $\mathbf{a}(z) + \mathbf{c}(z)$ with

$$\mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{G})^{-1}\,\mathbf{D_A}\mathbf{Y_A} = (\mathbf{a}(z) + \mathbf{c}(z))^{-1}\ ,$$

as desired. ♮

We now come to the theorem that we wanted to prove, originally. The previous theorems are not used in the proof, although it seems that they could be. But because we want the $\mathbf{0}$ block between the $\mathbf{A}$ block and the $\mathbf{B}$ block, and because we want $\mathbf{c}_0$ in the upper right corner, it's better to apply the following direct proof.

Theorem 4.3.5 *Let* $\mathbf{a}(z)$, $\mathbf{A}$, $\mathbf{D_A}$, $\mathbf{b}(z)$, $\mathbf{B}$, $\mathbf{D_B}$ *and their ancillaries be as in the previous theorems. Let* $\mathbf{c}_0$, $\mathbf{d}_0 \in \mathbb{C}^{r\times r}$ *be given. Then*

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{0} & -\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B} \\ -\mathbf{X_A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{Y_B} & \mathbf{B} \end{bmatrix}$$

and

$$\mathbf{D_H} = \begin{bmatrix} \mathbf{D_A} & & \\ & \mathbf{d}_0 & \\ & & \mathbf{D_B} \end{bmatrix}$$

linearize $\mathbf{h}(z) = z\mathbf{a}(z)\mathbf{d}_0\mathbf{b}(z) + \mathbf{c}_0$*; we have*

$$\mathbf{X_H} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{X_B} \end{bmatrix} \quad \textit{and} \quad \mathbf{Y_H} = \begin{bmatrix} \mathbf{Y_A} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

making a standard triple with

$$\mathbf{X_H}\,(z\mathbf{D_H} - \mathbf{H})^{-1}\,\mathbf{Y_H} = \mathbf{h}^{-1}(z)\ .$$

An explicit formula for $(z\mathbf{D_H} - \mathbf{H})^{-1}$ will be given in the proof.

Proof We use a compound Schur factoring, *i.e.* use the Schur complement twice.

$$z\mathbf{D_H} - \mathbf{H} = \mathbf{F}_1\mathbf{F}_2$$

where

$$\mathbf{F}_1 = \begin{bmatrix} z\mathbf{D_A} - \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{X_A} & \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{tr} \end{bmatrix}$$

and

$$\mathbf{F}_2 = \begin{bmatrix} \mathbf{I}_{sr} & \mathbf{0} & (z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B} \\ \mathbf{0} & z\mathbf{d}_0 & -\mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B} \\ \mathbf{0} & \mathbf{Y_B} & z\mathbf{D_B} - \mathbf{B} \end{bmatrix} .$$

This is

$$\mathbf{F}_2 = \mathbf{F}_3\mathbf{F}_4$$

with

$$\mathbf{F}_3 = \begin{bmatrix} \mathbf{I}_{sr} & \mathbf{Q}_j & (z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B} \\ \mathbf{0} & \mathbf{Q}_k & -\mathbf{a}^{-1}\mathbf{c}_0\mathbf{X_B}\,(z\mathbf{D_B} - \mathbf{B})^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{tr} \end{bmatrix} ,$$

where

$$\mathbf{Q}_j = -\,(z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B}\,(z\mathbf{D_B} - \mathbf{B})^{-1}\,\mathbf{Y_B}$$

and

$$\mathbf{Q}_k = z\mathbf{d}_0 + \mathbf{X_A}\,(z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B}\,(z\mathbf{D_B} - \mathbf{B})^{-1}\,\mathbf{Y_B}\,,$$

and

$$\mathbf{F}_4 = \begin{bmatrix} \mathbf{I}_{sr} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{Y_B} & z\mathbf{D_B} - \mathbf{B} \end{bmatrix} .$$

Simplifying $\mathbf{F}_3$ further,

$$\mathbf{F}_3 = \begin{bmatrix} \mathbf{I}_{sr} & -\,(z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}\mathbf{c}_0\mathbf{b}^{-1} & \mathbf{Q}_\ell \\ \mathbf{0} & z\mathbf{d}_0 + \mathbf{a}^{-1}\mathbf{c}_0\mathbf{b}^{-1} & -\mathbf{a}^{-1}\mathbf{c}_0\mathbf{X_B}\,(z\mathbf{D_B} - \mathbf{B})^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{tr} \end{bmatrix} ,$$

where,

$$\mathbf{Q}_\ell = (z\mathbf{D_A} - \mathbf{A})^{-1}\,\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B}\,(z\mathbf{D_B} - \mathbf{B})^{-1}\ .$$

Since

$$\begin{aligned} \det(z\mathbf{D_H} - \mathbf{H}) &= \det\mathbf{F}_1 \cdot \det\mathbf{F}_2 \\ &= \det\mathbf{F}_1 \cdot \det\mathbf{F}_3 \cdot \det\mathbf{F}_4\,, \end{aligned}$$

we have

$$\begin{aligned} \det(z\mathbf{D_H} - \mathbf{H}) &= \det(z\mathbf{D_A} - \mathbf{A})\det\left(z\mathbf{d}_0 + \mathbf{a}^{-1}\mathbf{c}_0\mathbf{b}^{-1}\right)\det(z\mathbf{D_B} - \mathbf{B}) \\ &= \det\mathbf{a}(z)\det\left(z\mathbf{d}_0 + \mathbf{a}^{-1}\mathbf{c}_0\mathbf{b}^{-1}\right)\det\mathbf{b}(z) \\ &= \det(z\mathbf{a}(z)\mathbf{d}_0\mathbf{b}(z) + \mathbf{c}_0) \\ &= \det(\mathbf{h}(z)) \end{aligned}$$

as claimed. ♮

To find the explicit form of the resolvent inverse, we invert the factors:

$$(z\mathbf{D_H} - \mathbf{H})^{-1} = \mathbf{F}_4^{-1}\mathbf{F}_3^{-1}\mathbf{F}_1^{-1}\,.$$

$$\mathbf{F}_4^{-1} = \begin{bmatrix} \mathbf{I}_{sr} & & \\ & \mathbf{I}_r & \\ & & (z\mathbf{D_B} - \mathbf{B})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{sr} & & \\ & \mathbf{I}_r & \\ & -\mathbf{Y_B} & \mathbf{I}_{tr} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{I}_{sr} & & \\ & \mathbf{I}_r & \\ & -(z\mathbf{D_B} - \mathbf{B})^{-1}\mathbf{Y_B} & (z\mathbf{D_B} - \mathbf{B})^{-1} \end{bmatrix}.$$

Now (using α, β, γ as shorthand for the relevant blocks, where β is regular),

$$\mathbf{F}_3 = \begin{bmatrix} \mathbf{I} & \alpha & \gamma \\ & \beta & \delta \\ & & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & & \\ & \beta & \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \alpha & \gamma \\ & \mathbf{I} & \beta^{-1}\delta \\ & & \mathbf{I} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{I} & & \\ & \beta & \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & & \gamma \\ & \mathbf{I} & \beta^{-1}\delta \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \alpha & \\ & \mathbf{I} & \\ & & \mathbf{I} \end{bmatrix}$$

So

$$\mathbf{F}_3^{-1} = \begin{bmatrix} \mathbf{I} & -\alpha & \\ & \mathbf{I} & \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & & -\gamma \\ & \mathbf{I} & -\beta^{-1}\delta \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & & \\ & \beta^{-1} & \\ & & \mathbf{I} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{I} & -\alpha & -\gamma + \alpha\beta^{-1}\delta \\ & \mathbf{I} & -\beta^{-1}\delta \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & & \\ & \beta^{-1} & \\ & & \mathbf{I} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{I} & -\alpha\beta^{-1} & -\gamma + \alpha\beta^{-1}\delta \\ & \beta^{-1} & -\beta^{-1}\delta \\ & & \mathbf{I} \end{bmatrix}$$

So

$$\mathbf{F}_3^{-1} = \begin{bmatrix} \mathbf{I}_{sr} & (z\mathbf{D_A} - \mathbf{A})^{-1}\mathbf{Y_A}\mathbf{c}_0\mathbf{h}^{-1}\mathbf{a} & \mathbf{U} \\ & \mathbf{b}\mathbf{h}^{-1}\mathbf{a} & \mathbf{b}\mathbf{h}^{-1}\mathbf{c}_0\mathbf{X_B}(z\mathbf{D_B} - \mathbf{B})^{-1} \\ & & \mathbf{I}_{tr} \end{bmatrix}$$

using $z\mathbf{d}_0 + \mathbf{a}^{-1}\mathbf{c}_0\mathbf{b}^{-1} = \mathbf{a}^{-1}(z\mathbf{a}\mathbf{d}_0\mathbf{b} + \mathbf{c}_0)\mathbf{b}^{-1} = \mathbf{a}^{-1}\mathbf{h}\mathbf{b}^{-1}$ so

$$\left(z\mathbf{d}_0 + \mathbf{a}^{-1}\mathbf{c}_0\mathbf{b}^{-1}\right)^{-1} = \mathbf{b}\mathbf{h}^{-1}\mathbf{a}$$

and

$$-(z\mathbf{D_A} - \mathbf{A})^{-1}\mathbf{Y_A}\mathbf{c}_0\mathbf{X_B}(z\mathbf{D_B} - \mathbf{B})^{-1}$$
$$+ (z\mathbf{D_A} - \mathbf{A})^{-1}\mathbf{Y_A}\mathbf{c}_0\mathbf{h}^{-1}\mathbf{C}_0\mathbf{X_B}(z\mathbf{D_B} - \mathbf{B})^{-1} = \mathbf{U}$$
$$= (z\mathbf{D_A} - \mathbf{A})^{-1}\mathbf{Y_A}\left[\mathbf{c}_0\mathbf{h}^{-1}\mathbf{c}_0 - \mathbf{c}_0\right]\mathbf{X_B}(z\mathbf{D_B} - \mathbf{B})^{-1}.$$

In the next section we will use this at $z = 0$ if $\mathbf{c}_0$ is invertible to show $\mathbf{U} = \mathbf{0}$ (sr by tr block). Also,

$$\mathbf{F}_1^{-1} = \begin{bmatrix} (z\mathbf{D_A} - \mathbf{A})^{-1} & & \\ -\mathbf{X_A}(z\mathbf{D_A} - \mathbf{A})^{-1} & \mathbf{I}_r & \\ & & \mathbf{I}_{tr} \end{bmatrix}.$$

Therefore $(z\mathbf{I} - \mathbf{H})^{-1}$ is $\mathbf{F}_4^{-1}\mathbf{F}_3^{-1}\mathbf{F}_1^{-1}$. Now

$$\mathbf{F}_4^{-1}\mathbf{F}_3^{-1} = \begin{bmatrix} \mathbf{I}_{sr} & (z\mathbf{D_A} - \mathbf{A})^{-1}\mathbf{Y_A}\mathbf{c}_0\mathbf{h}^{-1}\mathbf{a} & \mathbf{U} \\ \mathbf{0} & \mathbf{b}\mathbf{h}^{-1}\mathbf{a} & \mathbf{b}\mathbf{h}^{-1}\mathbf{c}_0\mathbf{X_B}(z\mathbf{D_B} - \mathbf{B})^{-1} \\ \mathbf{0} & -(z\mathbf{D_B} - \mathbf{B})^{-1}\mathbf{Y_B}\mathbf{b}\mathbf{h}^{-1}\mathbf{a} & \mathbf{R}_{33} \end{bmatrix}$$

where

$$\mathbf{R}_{33} = (z\mathbf{D_B} - \mathbf{B})^{-1} - (z\mathbf{D_B} - \mathbf{B})^{-1}\mathbf{Y_B}\mathbf{b}\mathbf{h}^{-1}\mathbf{c}_0\mathbf{X_B}(z\mathbf{D_B} - \mathbf{B})^{-1} \ .$$

Therefore $\mathbf{F}_4^{-1}\mathbf{F}_3^{-1}\mathbf{F}_1^{-1}$ is

$$\begin{bmatrix} \mathbf{R}_{11} & (z\mathbf{D_A} - \mathbf{A})^{-1}\mathbf{Y_A}\mathbf{c}_0\mathbf{b}^{-1}\mathbf{a} & \mathbf{U} \\ \mathbf{R}_{21} & \mathbf{b}\mathbf{h}^{-1}\mathbf{a} & \mathbf{b}\mathbf{h}^{-1}\mathbf{c}_0\mathbf{X_B}(z\mathbf{D_B} - \mathbf{B})^{-1} \\ \mathbf{R}_{31} & -(z\mathbf{D_B} - \mathbf{B})^{-1}\mathbf{Y_B}\mathbf{b}\mathbf{h}^{-1}\mathbf{a} & \mathbf{R}_{33} \end{bmatrix}$$

where

$$\begin{aligned} \mathbf{R}_{11} &= (z\mathbf{D_A} - \mathbf{A})^{-1} - (z\mathbf{D_A} - \mathbf{A})^{-1}\mathbf{Y_A}\mathbf{c}_0\mathbf{h}^{-1}\mathbf{a}\mathbf{X_A}(z\mathbf{D_A} - \mathbf{A}) \ , \\ \mathbf{R}_{21} &= -\mathbf{b}\mathbf{h}^{-1}\mathbf{a}\mathbf{X_A}(z\mathbf{D_A} - \mathbf{A})^{-1} \ , \\ \mathbf{R}_{31} &= (z\mathbf{D_B} - \mathbf{B})^{-1}\mathbf{Y_B}\mathbf{b}\mathbf{h}^{-1}\mathbf{a}\mathbf{X_A}(z\mathbf{D_A} - \mathbf{A})^{-1} \ . \end{aligned}$$

Moreover,

$$\begin{aligned} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{X}_B \end{bmatrix}(z\mathbf{D_H} - \mathbf{H})^{-1}\begin{bmatrix} \mathbf{Y_A} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} &= \mathbf{b}^{-1}(z)\mathbf{b}(z)\mathbf{h}^{-1}(z)\mathbf{a}(z)\mathbf{a}^{-1}(z) \\ &= \mathbf{h}^{-1}(z) \ , \end{aligned}$$

as desired.

Additionally, we need to ensure that the linearization given in Theorem 4.3.5 has the same spectrum as $\mathbf{h}(z) = z\mathbf{a}(z)\mathbf{d}_0\mathbf{b}(z) + \mathbf{c}_0$ by showing

$$\mathbf{E}(\lambda)(\lambda\mathbf{D_H} - \mathbf{H})\mathbf{F}(\lambda) = \begin{bmatrix} \mathbf{h}(z) & 0 \\ 0 & \mathbf{I}_{(s-1)r} \end{bmatrix}$$

for some unimodular $\mathbf{E}(\lambda)$ and $\mathbf{F}(\lambda)$. To find such $\mathbf{E}(\lambda)$ and $\mathbf{F}(\lambda)$, we can take the Schur compliment of

$$\lambda\mathbf{D_H} - \mathbf{H} = \left[\begin{array}{cc|c} z\mathbf{D_A} - \mathbf{A} & \mathbf{0} & \mathbf{Y_A}\mathbf{c}_0\mathbf{X_B} \\ \hline \mathbf{X_A} & z\mathbf{d}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{Y_B} & z\mathbf{D_B} - \mathbf{B} \end{array}\right] . \tag{4.5}$$

Recall

$$\mathbf{a}(z) = \mathbf{Y_A}^{-1}(z\mathbf{D_A} - \mathbf{A})\mathbf{X_A}^{-1}$$

and

$$\mathbf{b}(z) = \mathbf{Y_B}^{-1}(z\mathbf{D_B} - \mathbf{B})\mathbf{X_B}^{-1} \ .$$

Then, the Schur compliment of equation (4.5) is

$$\begin{aligned}
\mathbf{S}(z) &= \mathbf{Y_B c_0 X_B} - \begin{bmatrix} z\mathbf{D_A} - \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X_A}^{-1} & -\mathbf{X_A} z\mathbf{d}_0 \mathbf{Y_B}^{-1} \\ \mathbf{0} & \mathbf{Y_B}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ z\mathbf{D_B} - \mathbf{B} \end{bmatrix} \\
&= \mathbf{Y_A c}_0 \mathbf{X_B} - \left[(z\mathbf{D_A} - \mathbf{A})\, \mathbf{X_A}^{-1} - (z\mathbf{D_A} - \mathbf{A})\, \mathbf{X_A}^{-1} z\mathbf{d}_0 \mathbf{Y_B}^{-1} \right] \begin{bmatrix} \mathbf{0} \\ z\mathbf{D_B} - \mathbf{B} \end{bmatrix} \\
&= \mathbf{Y_A c}_0 \mathbf{X_B} + (z\mathbf{D_A} - \mathbf{A})\, \mathbf{X_A}^{-1} z\mathbf{d}_0 \mathbf{Y_B}^{-1} (z\mathbf{D_B} - \mathbf{B}) \\
&= \mathbf{Y_A c}_0 \mathbf{X_B} + \mathbf{Y_A} \underbrace{\mathbf{Y_A}^{-1} (z\mathbf{D_A} - \mathbf{A})\, \mathbf{X_A}^{-1}}_{\mathbf{a}(z)} z\mathbf{d}_0 \underbrace{\mathbf{Y_B}^{-1} (z\mathbf{D_B} - \mathbf{B})\, \mathbf{X_B}^{-1}}_{\mathbf{b}(z)} \mathbf{X_B} \\
&= \mathbf{Y_A} \left(\mathbf{c}_0 + z\mathbf{a}(z)\mathbf{d}_0 \mathbf{b}(z) \right) \mathbf{X_B} \\
&= \mathbf{Y_A h}(z) \mathbf{X_B} \,.
\end{aligned}$$

Therefore, as long as $\mathbf{Y_A}$ and $\mathbf{X_B}$ are regular, it guarantees that there exists unimodular $\mathbf{E}(\lambda)$ and $\mathbf{F}(\lambda)$.

4.4 Implications

Consider first the Mandelbrot matrices $\mathbf{M}_n$ from Section 4.2. Here r is just 1, and we may deduce a sequence of facts, as follows.

Lemma 4.4.1 *The dimension of* $\mathbf{M}_n$ *is* $d_n \times d_n$*, where* $d_n = 2^{n-1} - 1$.

Proof Simple induction beginning with $d_2 = 1$ and $d_{n+1} = 2d_n + 1$. ♮

Lemma 4.4.2 $\mathbf{X}_n$ *and* $\mathbf{Y}_n$ *are simply* $\mathbf{e}_{d_n}^T = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}$ *and* $\mathbf{e}_1$ *where* $\mathbf{e}_1^T = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$.

Proof Again induction, beginning with $\mathbf{M}_2$:

$$p_2 = z + 1 \quad \Rightarrow \quad [1]\,(z+1)^{-1}\,[1] = p_2^{-1}$$

and

$$x_{n+1} = \begin{bmatrix} \text{zeros}(\text{size}(x_n)) & 0 & x_n \end{bmatrix}$$

while

$$y_{n+1} = \begin{bmatrix} y_n \\ 0 \\ \text{zeros}(\text{size}(y_n)) \end{bmatrix}$$

by Theorem 4.3.4. ♮

Lemma 4.4.3 *The bottom left corner of* $\mathbf{M}_n^{-1}$ *is always* -1.

Proof We have several proofs for this fact, most simply using the minor of the top right corner; but we will shortly want $\mathbf{M}_n^{-1}$ explicitly and so we compute it here. We note that $\mathbf{M}_2$ is invertible, $\mathbf{M}_2^{-1} = [-1]$, and that $\mathbf{c}_0 = 1$ is always invertible. Thus by induction $\mathbf{M}_{n+1}$ is invertible

because $\mathbf{M}_n$ is. We have, by specializing the resolvent inverse from Theorem 4.3.4, that the bottom left block of $(\mathbf{0} \cdot \mathbf{I} - \mathbf{M}_{n+1})^{-1}$ is

$$\begin{aligned}\mathbf{M}_n^{-1}\mathbf{Y}_n \cdot 1 \cdot 1^{-1} \cdot 1 \cdot \mathbf{X}_n \cdot \mathbf{M}_n^{-1} &= \mathbf{M}_n^{-1} \begin{bmatrix} 1 \\ 0 \\ \vdots \end{bmatrix} \begin{bmatrix} 0 & \cdots & 1 \end{bmatrix} \mathbf{M}_n^{-1} \\ &= C_n \mathcal{R}_n\end{aligned}$$

where C_n is the first column of $\mathbf{M}_n^{-1}$ and $\mathcal{R}_n$ is the last row of $\mathbf{M}_n^{-1}$.

By the inductive hypothesis, the bottom left corner of this block is

$$\begin{bmatrix} & \\ & \\ & \\ +1 & \end{bmatrix}$$

because $(-1)(-1) = +1$. Remember this is the bottom left block of $(-\mathbf{M}_{n+1})^{-1}$; thus if the bottom left corners of $\mathbf{M}_n^{-1}$ are -1, so is the bottom left corner of $(\mathbf{M}_{n+1})^{-1}$. ♮

Lemma 4.4.4 *The upper left block of* $\mathbf{M}_{n+1}^{-1}$ *is the same as the lower right block; both are*

$$\mathbf{M}_n^{-1} + \mathbf{M}_n^{-1}\mathbf{Y}_n \cdot \mathbf{X}_n\mathbf{M}_n^{-1} = \mathbf{M}_n^{-1} + C_n\mathcal{R}_n \, .$$

The proof is simple computation.

Lemma 4.4.5 *The first column and the last row of the blocks in Lemma 4.4.4 are zero.*

Proof The left column of $C_n\mathcal{R}_n$ is $-C_n$ because the left element of $\mathcal{R}_n$ is -1 by Lemma 4.4.3. Thus the left column of $\mathbf{M}_n^{-1}C_n\mathcal{R}_n$ is zero. Similarly the last row of $C_n\mathcal{R}_n$ is $-\mathcal{R}_n$, leading to the same conclusion. ♮

Lemma 4.4.6 *For* $n \geq 3$*, the lower left block of* $\mathbf{M}_n^{-1} + C_n\mathcal{R}_n$ *is a* $1 + d_{n-1} \times 1 + d_{n-1}$ *block of zeros, and all other blocks of* $\mathbf{M}_{n+1}^{-1}$ *are untouched.*

Proof Consider first $\mathbf{M}_3 = \begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & -1 \end{bmatrix}$ and $\mathbf{M}_3^{-1} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & -1 & -1 \\ -1 & 1 & 0 \end{bmatrix}$. Then $C_3 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$ and $\mathcal{R}_3 = \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$ so

$$C_3\mathcal{R}_3 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix}, \quad \mathbf{M}_3^{-1} + C_3\mathcal{R}_3 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Note that the first d_n entries of C_n are 0 and that the last d_n entries of $\mathcal{R}_n$ are 0. Note that as in the proof of Lemma 4.4.3, the bottom left $d_n \times d_n$ block of $\mathbf{M}_{n+1}^{-1}$ is just $C_n\mathcal{R}_n$, and by specializing

the resolvent formula the row just above that is $-\mathcal{R}_n$; similarly the column beside that block is $\mathcal{R}_n$; similarly the column beside that block is C_n. Indeed

$$\mathbf{M}_{n+1}^{-1} = \begin{bmatrix} \mathbf{M}_{n+1}^{-1} + C_n\mathcal{R}_n & C_n & \mathbf{0} \\ -\mathcal{R}_n & -1 & \mathcal{R}_n \\ -C_n\mathcal{R}_n & -C_n & \mathbf{M}_n^{-1} + C_n\mathcal{R}_n \end{bmatrix}.$$

We have established in Lemma 4.4.5 that the bottom right block has a zero last row and that the upper left block has a zero first column. Thus

$$C_{n+1} = \begin{bmatrix} 0 \\ 1 \\ C_n \end{bmatrix} \quad \text{and} \quad \mathcal{R}_{n+1} = \begin{bmatrix} \mathcal{R}_n & 1 & 0 \end{bmatrix}.$$

Therefore

$$C_{n+1}\mathcal{R}_{n+1} = \begin{bmatrix} 0 & 0 & 0 \\ \mathcal{R}_n & 1 & 0 \\ C_n\mathcal{R}_n & C_n & 0 \end{bmatrix}$$

and

$$\mathbf{M}_{n+1}^{-1} + C_{n+1}\mathcal{R}_{n+1} = \begin{bmatrix} \mathbf{M}_n^{-1}C_n\mathcal{R}_n & C_n & 0 \\ 0 & 0 & \mathcal{R}_n \\ 0 & 0 & \mathbf{M}_n^{-1} + C_n\mathcal{R}_n \end{bmatrix}$$

establishing the claim by induction. ♮

Definition A matrix family is **Bohemian** if its entries come from a single discrete (and hence bounded) set. Here the set is just $\{-1, 0, 1\}$. The name comes from "**Bo**unded **He**ight **M**atrix of Integers."

Lemma 4.4.7 *The Mandelbrot matrices are Bohemian, with height*[1] *1. Indeed the only entries are* 0 *or* -1.

Proof Induction. ♮

Definition A matrix family has **rhapsody** if it is Bohemian and its inverse is also Bohemian with the same height.

Theorem 4.4.8 *The Mandelbrot matrices have rhapsody.*

Proof By induction using the previous lemmas. Clearly the entries of C_n and $\mathcal{R}_n$ are $+1$, -1, or 0; thus $C_n\mathcal{R}_n$ has height 1. Since the contribution of $C_n\mathcal{R}_n$ to $\mathbf{M}_n^{-1}$ in $\mathbf{M}_n^{-1} + C_n\mathcal{R}_n$ was entirely removing the lower left $1 + d_{n-1}$ by $1 + d_{n-1}$ block, and did not touch the other entries, each block remains of height 1. ♮

[1] height($\mathbf{A}$) := $\|\text{vec}(\mathbf{A})\|_\infty$ is the largest entry of $|\mathbf{A}|$, where $|\mathbf{A}|$ means the matrix whose entries are the absolute values of the entries of $\mathbf{A}$.

4.5 First matrix polynomial experiments

To test these ideas we examine a family of matrix polynomials that we have artificially created for the purpose. We use the following recursive construction. Put

$$\mathbf{h}_1 = z\mathbf{I} + \mathbf{c}_0$$

and for $k \geq 0$

$$\mathbf{h}_{k+1}(z) = z\mathbf{h}_k^2(z) + \mathbf{c}_k(z) \tag{4.6}$$

where $\mathbf{c}_k(z) \in \mathbb{C}^{4\times 4}$ are nonsingular upper Hessenberg matrices with zero diagonal and entries -1 on the subdiagonal. We choose these matrices $\mathbf{c}_k$ in advance, not all the same. This gives a "Mandelbrot-like" flavour to the construction. Notice that for $k \geq 1$ $\deg \mathbf{h}_k(z) = 2^k - 1$, and its dimension is 4×4 for every k. The linearization of Theorem 4.3.4 gives matrices $\mathbf{H}_k$ of dimension $4 \cdot \left(2^k - 1\right)$ by $4 \cdot \left(2^k - 1\right)$. Our experiments covered various choices of the $\mathbf{c}_k$ and dimensions up to 16380×16380.

The matrices $\mathbf{c}_k$ that we used are

$$\begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 & -1 & -1 & 0 \\ -1 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & -1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -1 & 0 \\ -1 & 0 & 1 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix}.$$

Larger experiments are of course possible and desirable.

We exhibit the eigenvalues of one 4092×4092 ($k = 10$) matrix in figure 4.1. We compared the computed eigenvalues (computed using Maple's `LinearAlgebra:-Eigenvalues` routine, which calls an implementation of LAPACK via the NAG library) with the roots of the characteristic polynomials $p_k(z) = \det(\mathbf{h}_k(z))$ computed by Maple's built-in solver `fsolve` (refer to [5]) which is slow but quite reliable. Because the

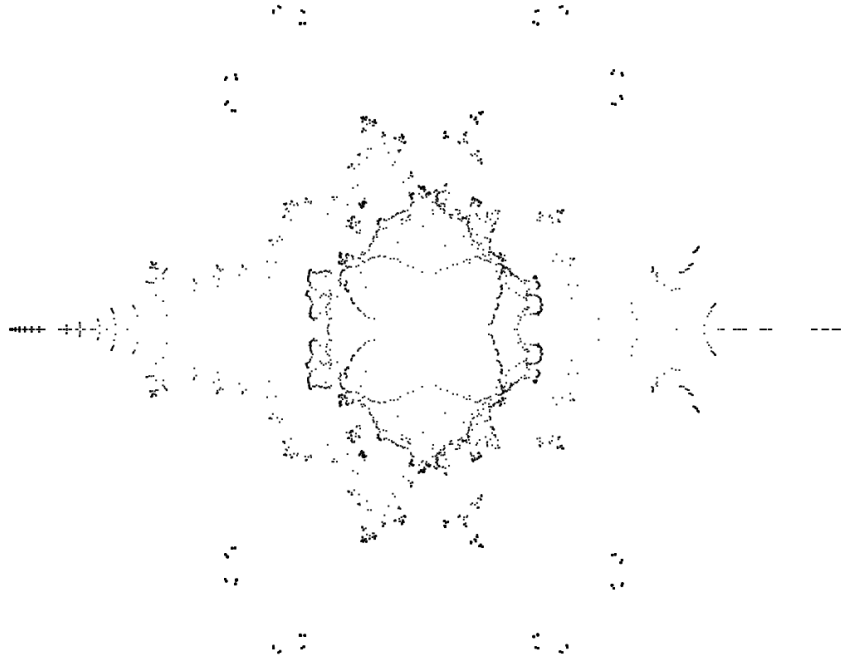


Figure 4.1: Eigenvalues of a 4092×4092 matrix. For details, refer to equation (4.6)

height of the exactly-computed characteristic polynomial reached 10^{234}, solving the polynomial using `fsolve` required multiple precision, which is slow. To compute the residual, we computed the singular values of our matrix polynomial, $\mathbf{h}_{10}(\xi_i)$ for each of the eigenvalues, ξ_i, and divided the smallest singular value by the largest singular value for each case. The residual σ_4/σ_1 of $\mathbf{h}_4(\lambda)$ in any polynomial eigenvalue λ was never more than $\sim 4 \times 10^{-13}$. Table 4.1 shows the time taken to compute the eigenvalues and time take to compute the roots using Maple's `fsolve` (using a machine with 32 GB of memory). Eigenvalue computation of the linearization was always the fastest taking only 94.782 seconds for the $k = 10$ case. Unfortunately, we had to kill the job on Maple after a week for the $k = 10$ case.

In another experiment, for a specialized example, we compared the accuracy of the eigenvalues from our companion construction and the eigenvalues from the Frobenius companion construction in MATLAB. In comparison to the previous experiment, we used a lower degree matrix polynomial

$$\mathbf{H}(z) = z\mathbf{a}(z)\mathbf{b}(z) + \mathbf{c}_0 \,,$$

k	Dimension	`Eigenvalues` (s)	`fsolve` (s)
5	124	0.047	0.531
6	252	0.125	2.313
7	508	0.640	176.203
8	1020	2.375	829.093
9	2044	13.469	80242.078
10	4092	94.782	–
11	8188	715.109	–
12	16380	6367.703	–

Table 4.1: Times of eigenvalue computation of the algebraic linearizations using Maple. The polynomial solver `fsolve` takes so long because the heights of the characteristic polynomials grow exponentially in the dimension. The eigenvalue solver has no difficulty, because the matrix height is constant.

where

$$\mathbf{a}(z) = \sum_{k=0}^{3} z^k \mathbf{A}_k \,,$$
$$\mathbf{b}(z) = \sum_{k=0}^{3} z^k \mathbf{B}_k \,,$$

and $\mathbf{c}_0 = \mathbf{I}_5$. The matrices $\mathbf{A}_k$ that we used here were chosen by calling Maple's `RandomMatrix`

function. For reference, the ones we used were

$$\mathbf{A}_0 = \begin{bmatrix} -81 & -98 & -76 & -4 & 29 \\ -38 & -77 & -72 & 27 & 44 \\ -18 & 57 & -2 & 8 & 92 \\ 87 & 27 & -32 & 69 & -31 \\ 33 & -93 & -74 & 99 & 67 \end{bmatrix},$$

$$\mathbf{A}_1 = \begin{bmatrix} 76 & 20 & 31 & 94 & -16 \\ -44 & -61 & -50 & 12 & -9 \\ 24 & -48 & -80 & -2 & -50 \\ 65 & 77 & 43 & 50 & -22 \\ 86 & 9 & 25 & 10 & 45 \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} 70 & 82 & 12 & 22 & 60 \\ -32 & 72 & -62 & 14 & -95 \\ -1 & 42 & -33 & 16 & -20 \\ 52 & 18 & -68 & 9 & -25 \\ -13 & -59 & -67 & 99 & 51 \end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} -38 & -63 & 12 & 21 & -82 \\ 91 & -26 & 45 & 90 & -70 \\ -1 & 30 & -14 & 80 & 41 \\ 63 & 10 & 60 & 19 & 91 \\ -23 & 22 & -35 & 88 & 29 \end{bmatrix}.$$

We then randomly assigned

$$\mathbf{B}_0 = \begin{bmatrix} -15 & 10 & -83 & 10 & -4 \\ 2 & -44 & 9 & -61 & 5 \\ -88 & 26 & 88 & -26 & -91 \\ 99 & -3 & 95 & -20 & -44 \\ -59 & -62 & 63 & -78 & -38 \end{bmatrix},$$

and chose the rest of the $\mathbf{B}_k$ to be

$$\begin{aligned} \mathbf{B}_3 &= \mathbf{A}_3^{-1} \\ \mathbf{B}_2 &= -\mathbf{A}_3^{-1}\mathbf{A}_2\mathbf{B}_3 \\ \mathbf{B}_1 &= -\mathbf{A}_3^{-1}\left(\mathbf{A}_1\mathbf{B}_3 + \mathbf{A}_2\mathbf{B}_2\right) \end{aligned}$$

so that some of the coefficients of $\mathbf{H}(z)$, when expressed in the monomial basis, would be $\mathbf{0}$. However, since we are computing these coefficients numerically, rounding errors would be introduced, resulting in loss of accuracy as we will see in the residuals.

In order to construct the algebraic linearization of $\mathbf{H}(z)$, we need the linearizations of both $\mathbf{a}(z)$ and $\mathbf{b}(z)$. We decided to use the Frobenius companion construction for these smaller companions, since the coefficients were readily available to use. The rest then follows the construction described in this paper. This suggests the idea that we can potentially mix different polynomial bases using our construction, which will be elaborated on in the next example.

We computed the residuals (as described in our previous example) to compare the accuracy of the two results. We found that the largest residual for the eigenvalues of the algebraic linearization is approximately 7.8×10^{-12} and the largest residual for the Frobenius companion matrix is approximately 7.0×10^{-9}, around 900 times larger. This suggests that the algebraic linearization may be more numerically stable.

For our third example, we show that one can mix different polynomial bases together. All that is needed is a standard triple for $\mathbf{a}(z)$ and another for $\mathbf{b}(z)$, like so:

$$\begin{aligned} \mathbf{X_A}(z\mathbf{A}_1 - \mathbf{A}_0)^{-1}\mathbf{Y_A} &= \mathbf{a}^{-1}(z) \\ \mathbf{X_B}(z\mathbf{B}_1 - \mathbf{B}_0)^{-1}\mathbf{Y_B} &= \mathbf{b}^{-1}(z)\,. \end{aligned}$$

For instance, suppose $\mathbf{a}(z)$ is expressed in the barycentric Lagrange basis, as follows:

$$\mathbf{a}(z) = w(z)\sum_{k=0}^{n}\frac{\beta_k\mathbf{a}_k}{z-\tau_k} \qquad \mathbf{a}_k \in \mathbb{C}^{r\times r}$$

where the τ_k are distinct nodes, the node polynomial is $w(z) = \prod_{k=0}^{n}(z-\tau_k)$, and the barycentric weights β_k come from the partial fraction decomposition

$$\frac{1}{w(z)} = \sum_{k=0}^{n}\frac{\beta_k}{z-\tau_k}\,.$$

Then there are several choices for linearizations of $\mathbf{a}(z)$ without needing to change bases. See [1] or [19]. In 2004, RMC implemented the following linearization in Maple[6]: if

$$\mathbf{A}_0 = \begin{bmatrix} -\tau_0\mathbf{I} & & & & \mathbf{a}_0^{\mathrm{T}} \\ & -\tau_1\mathbf{I} & & & \mathbf{a}_1^{\mathrm{T}} \\ & & \ddots & & \vdots \\ & & & -\tau_n\mathbf{I} & \mathbf{a}_n^{\mathrm{T}} \\ -\beta_0\mathbf{I} & -\beta_1\mathbf{I} & \cdots & -\beta_n\mathbf{I} & \mathbf{0} \end{bmatrix} \quad \mathbf{A}_1 = \begin{bmatrix} -\mathbf{I} & & & & \\ & -\mathbf{I} & & & \\ & & \ddots & & \\ & & & -\mathbf{I} & \\ & & & & \mathbf{0} \end{bmatrix}$$

then $\det(\mathbf{A}_0 - z\mathbf{A}_1) = \det\left(\mathbf{a}(z)^{\mathrm{T}}\right) = \det(\mathbf{a}(z))$. Putting the zero blocks in the lower left corner is not as numerically stable as using linearizations with the zero blocks in the upper left corner (see [14]) but we'll use the existing software. The transpose also complicates this example, but not much.

It can be shown that

$$\mathbf{X_A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

and

$$\mathbf{Y_A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} & \mathbf{0} \end{bmatrix}^{\mathbf{T}}$$

give $\mathbf{X_A}(z\mathbf{A}_1 - \mathbf{A}_0)^{-1}\mathbf{Y_A} = \mathbf{a}^{-1}(z)$ (note the sign reversal).

For $\mathbf{b}(z)$, we choose the Chebyshev basis. One could equally well choose the Legendre basis (implemented in Maple as `JacobiP(k, 0, 0, x)`) or any other bases. The generalized

companion matrix ("colleague" matrices of [11] and of [18] independently) give the linearization of $\mathbf{b}_0T_0(x) + \mathbf{b}_1T_1(x) + \mathbf{b}_2T_2(x) + \mathbf{b}_3T_3(x) + \mathbf{b}_4T_4(x) + \mathbf{b}_5T_5(x)$ as

$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{0} & \frac{1}{2}\mathbf{I} & \mathbf{0} & \mathbf{0} & -\mathbf{b}_0 \\ \mathbf{I} & \mathbf{0} & \frac{1}{2}\mathbf{I} & \mathbf{0} & -\mathbf{b}_1 \\ & \frac{1}{2}\mathbf{I} & \mathbf{0} & \frac{1}{2}\mathbf{I} & -\mathbf{b}_2 \\ & & \frac{1}{2}\mathbf{I} & \mathbf{0} & -\mathbf{b}_3 + \mathbf{b}_5 \\ & & & \frac{1}{2}\mathbf{I} & -\mathbf{b}_4 \end{bmatrix} \quad \mathbf{B}_1 = \begin{bmatrix} \mathbf{I} & & & & \\ & \mathbf{I} & & & \\ & & \mathbf{I} & & \\ & & & \mathbf{I} & \\ & & & & 2\mathbf{b}_5 \end{bmatrix}$$

with

$$\mathbf{X_B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

and

$$\mathbf{Y_B} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{\mathrm{T}}.$$

That is, $\mathbf{X_B}(z\mathbf{B}_1 - \mathbf{B}_0)^{-1}\mathbf{Y_B} = \mathbf{b}^{-1}(z)$, $z \notin \Lambda(b)$.

Specifically, we take for $\mathbf{a}(z)$ the nodes $\begin{bmatrix} -1 & -\frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$ and the barycentric weights $\beta = \begin{bmatrix} -\frac{2}{3} & \frac{4}{3} & -\frac{4}{3} & \frac{2}{3} \end{bmatrix}$. We suppose that

$$\mathbf{a}(-1) = \begin{bmatrix} -2 & -1 & -1 \\ -1 & -1 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

$$\mathbf{a}(-1/2) = \begin{bmatrix} -0.875 & -0.5 & -1.25 \\ -0.75 & -0.125 & 0.5 \\ 0 & -0.75 & -0.875 \end{bmatrix}$$

$$\mathbf{a}(1/2) = \begin{bmatrix} -1.625 & 0.5 & -0.25 \\ -1.75 & 0.125 & -0.5 \\ 0 & -1.75 & -0.625 \end{bmatrix}$$

$$\mathbf{a}(1) = \begin{bmatrix} -2 & 1 & 1 \\ -3 & 1 & -1 \\ 0 & -3 & 1 \end{bmatrix}$$

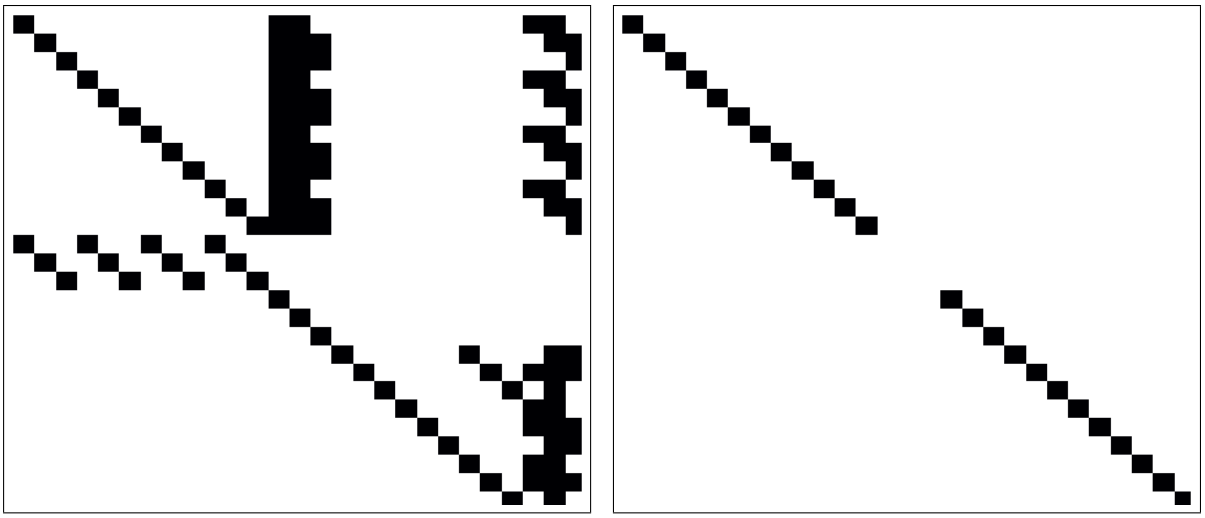
Thus $\mathbf{a}(z)$ has degree at most 3. We choose $\mathbf{b}(z)$ of degree 3, with

$$\mathbf{b}_0 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & -1 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

$$\mathbf{b}_1 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

$$\mathbf{b}_2 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\mathbf{b}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$



(a) Matrix structure of $\mathbf{H}$ (b) Matrix structure of $\mathbf{D_H}$

Figure 4.2: Matrix structure of the companion matrix $(\mathbf{H}, \mathbf{D_H})$ of $\mathbf{h}(z) = z\mathbf{a}(z)\mathbf{b}(z) + \mathbf{c}_0$. The block of zeros in $\mathbf{D_H}$ means that there are spurious infinite eigenvalues. These are numerically harmless and can be discarded.

The shape of the resulting algebraic linearization for $\mathbf{h}(z) = z\mathbf{a}(z)\mathbf{b}(z) + \mathbf{c}_0$ is shown in figure 4.2.

To find the forward error of the eigenvalues, we needed a program to find the appropriate root/eigenvalue pairings. Because the number of eigenvalues and roots in this test was modest, we wrote this "sibling finder" program in Maple. The largest forward error of this construction is approximately 8.7×10^{-15}.

While not conclusive, these experiments show that the algebraic linearization introduced this paper can be fast and accurate when computing polynomial eigenvalues.

Remark We learned to be careful not to have singular $\mathbf{c}_k$, which leads to high multiplicity zero eigenvalues of $\mathbf{h}_k(z)$ and thus of $\mathbf{H}_k$. Such high multiplicity zeros caused serious numerical artifacts. Owing to the integer nature of this family, this could perhaps be ameliorated without recourse to high precision, but we leave this for future work.

4.6 Concluding remarks

> "Almost anything will give you a strong linearization. What would be interesting would be numerical stability." — Françoise Tisseur (private communication)

There is some hope here for numerical stability of these linearizations, owing to the reduced height. Indeed, taken to extremes, a linearization of height 1 might have a characteristic equation of height exponential in the degree. This means that the polynomial evaluation condition number [7] will be C^N for some $C > 1$. However, the linearization resulting from recursive use of Theorem 4.3.4, having height 1, will have an expected condition number $O(N^2)$ [2]. Here

N is the dimension of the matrix. This means that the algorithm implied by the use of our linearizations can be (for some examples) exponentially more numerically stable.

However, not every matrix polynomial has a naturally recursive formulation. Preliminary experiments on reverse-engineering such formulations are promising and we will report on these developments later.

We have no theorems that suggest a lower-height matrix will have better-conditioned eigenvalues, only an expectation that is perhaps naive. This, too, will be reported on at a later date. Of course by "height" we mean scaled height, which needs a careful formulation; obviously $s\mathbf{A}$ has eigenvalues $s\lambda_k$ if $\mathbf{A}$ has eigenvalues λ_k, and the same eigenvectors (and thus eigenvalue condition numbers are unchanged by the scaling). Perhaps a better numerical representation of scaled height's sensitivity would be, say,

$$t = \min_{a_{ij}\neq 0} \frac{|a_{ij}|}{\text{Height}(\mathbf{A})} .$$

The smaller this number is, the more sensitive one might expect the eigenvalues to be. Again, this has yet to be explored.

Acknowledgments

We acknowledge the support of the Ontario Graduate Institution, The National Science & Engineering Research Council of Canada, the University of Alcalá, the Rotman Institute of Philosophy, the Ontario Research Centre of Computer Algebra, and Western University. Part of this work was developed while R. M. Corless was visiting the University of Alcalá, in the frame of the project Giner de los Rios. L. Gonzalez-Vega, J. R. Sendra and J. Sendra are partially supported by the Spanish Ministerio de Economía y Competitividad under the Project MTM2014-54141-P.

Bibliography

[1] A. Amiraslani, R. M. Corless, and P. Lancaster, *Linearization of matrix polynomials expressed in polynomial bases*, IMA Journal of Numerical Analysis, 29 (2008), pp. 141–157.

[2] C. Beltrán and D. Armentano, *The polynomial eigenvalue problem is well conditioned for random inputs*, arXiv preprint arXiv:1706.06025, (2017).

[3] E. Y. S. Chan, *A comparison of solution methods for Mandelbrot-like polynomials*, Master's thesis, The University of Western Ontario, 2016.

[4] E. Y. S. Chan and R. M. Corless, *A new kind of companion matrix*, Electronic Journal of Linear Algebra, 32 (2017), pp. 335–342.

[5] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt, *First leaves: a tutorial introduction to Maple V*, Springer Science & Business Media, 2012.

[6] R. M. Corless, *Generalized companion matrices in the lagrange basis*, Proceedings of EACA, (2004), pp. 317–322.

[7] R. M. Corless and N. Fillion, *A graduate introduction to numerical methods*, AMC, 10 (2013), p. 12.

[8] R. M. Corless and P. W. Lawrence, *The largest roots of the Mandelbrot polynomials*, in Computational and Analytical Mathematics, Springer, 2013, pp. 305–324.

[9] F. M. Dopico, J. Pérez, and P. Van Dooren, *Block minimal bases ℓ-ifications of matrix polynomials*, arXiv preprint arXiv:1803.06306, (2018).

[10] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix polynomials*, SIAM, 2009.

[11] I. Good, *The colleague matrix, a Chebyshev analogue of the companion matrix*, The Quarterly Journal of Mathematics, 12 (1961), pp. 61–68.

[12] N. J. Higham, M. R. Dennis, P. Glendinning, P. A. Martin, F. Santosa, and J. Tanner, *The Princeton companion to applied mathematics*, Princeton University Press, 2015.

[13] L. Hogben, *Handbook of linear algebra*, CRC Press, 2006.

[14] P. W. Lawrence and R. M. Corless, *Stability of rootfinding for barycentric Lagrange interpolants*, Numerical Algorithms, 65 (2014), pp. 447–464.

[15] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 1029–1051.

[16] ——, *Vector spaces of linearizations for matrix polynomials*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 971–1004.

[17] Q. I. Rahman and G. Schmeisser, *Analytic theory of polynomials*, no. 26, Oxford University Press, 2002.

[18] W. Specht, *Die lage der nullstellen eines polynoms*, Mathematische Nachrichten, 15 (1956), pp. 353–374.

[19] R. Van Beeumen, W. Michiels, and K. Meerbergen, *Linearization of Lagrange and Hermite interpolating matrix polynomials*, IMA Journal of Numerical Analysis, 35 (2015), pp. 909–930.

Chapter 5

Generalized standard triples for algebraic linearizations of matrix polynomials

5.1 Introduction

A *matrix polynomial* is normally defined using words such as "A matrix polynomial $\boldsymbol{P}(\lambda) \in F^{m\times n}(\lambda)$ is a polynomial in the variable λ with coefficients that are m by n matrices with entries from the field F." Typically an expression in the monomial basis λ^k is given for $\boldsymbol{P}(\lambda)$, and only *regular* matrix polynomials are considered, that is, with $m = n$ (we will use n for the dimension) and where $\det \boldsymbol{P}(\lambda)$ is not identically zero. Matrix polynomials have many applications and their study is of both classic and ongoing interest.

In this paper we wish to consider the case when *any* polynomial basis $\phi_k(\lambda)$ is used, if the set $\{\phi_k(\lambda)\}$ for $0 \le k \le \ell$ forms a basis for polynomials of degree at most ℓ. Thus, we write our regular matrix polynomial as

$$\boldsymbol{P}(\lambda) = \sum_{k=0}^{\ell} \boldsymbol{P}_k \phi_k(\lambda) \,,$$

where the matrices $\boldsymbol{P}_k \in F^{n\times n}$ are square, and the *degree* of $\boldsymbol{P}(\lambda)$ is at most ℓ. The upper bound ℓ on the degree is also called the *grade*. The notion of "grade" is useful even for the monomial basis, but it is especially useful if the basis is an interpolational basis or the Bernstein basis, when the degree of the polynomial may not be clear from the data.

Two matrix polynomials $\boldsymbol{P}_1(\lambda)$ and $\boldsymbol{P}_2(\lambda)$ are called *equivalent* if there exist unimodular matrix polynomials (that is, matrices with constant nonzero determinant) $\boldsymbol{E}(\lambda)$ and $\boldsymbol{F}(\lambda)$ with $\boldsymbol{P}_1(\lambda) = \boldsymbol{E}(\lambda)\boldsymbol{P}_2(\lambda)\boldsymbol{F}(\lambda)$. A matrix pencil $\boldsymbol{L}(\lambda) := \lambda\boldsymbol{C}_1 - \boldsymbol{C}_0$ is called a *linearization* of the matrix polynomial $\boldsymbol{P}(\lambda)$ if both $\boldsymbol{C}_1$ and $\boldsymbol{C}_0$ are of dimension $N \ge n\ell$ and $\boldsymbol{L}(\lambda)$ is equivalent to the block diagonal matrix $\operatorname{diag}(\boldsymbol{P}(\lambda), \boldsymbol{I}_{N-n})$. Two linearizations $\boldsymbol{L}_m(\lambda)$ and $\boldsymbol{L}_\phi(\lambda)$ are called *strictly* equivalent if the corresponding matrices are equivalent: $\boldsymbol{C}_{1,m} = \boldsymbol{E}\boldsymbol{C}_{1,\phi}\boldsymbol{F}$ and $\boldsymbol{C}_{0,m} = \boldsymbol{E}\boldsymbol{C}_{0,\phi}\boldsymbol{F}$, with the same unimodular matrices $\boldsymbol{E}$ and $\boldsymbol{F}$.

The *reversal*[1] of a matrix polynomial of grade ℓ is the polynomial $\operatorname{rev}\boldsymbol{P}(\lambda) = \lambda^\ell \boldsymbol{P}(\lambda^{-1})$.

[1] This definition, which is standard, is particularly appropriate for the monomial basis. The coefficients of the reversed matrix polynomial in the monomial basis are simply the same matrices in reverse order. The notion of a reversal, however, is independent of the basis used. In [8] we find a slightly different definition, appropriate for computation in a Lagrange or Hermite interpolational basis, which maps an arbitrary finite point to infinity; this

A linearization $\boldsymbol{L}(\lambda) = \lambda\boldsymbol{C}_1 - \boldsymbol{C}_0$ of $\boldsymbol{P}$ is called a *strong* linearization if also rev $\boldsymbol{L}(\lambda)$ is a linearization of rev $\boldsymbol{P}(\lambda)$.

For more information, consult [24]. See also [26], [18], and consult the seminal book [17]. Linearizations using different polynomials bases were first systematically studied in [1]. Some recent papers of interest include [4], [25], [10], [14], and [29]; this is a very active area.

5.1.1 Organization of the paper

In the remainder of this first section, we establish notation, give the definition of a generalized standard triple, and give lemmas about common similarity transformations[2]. At the end of this section, we show how to use the generalized standard triple in the construction of algebraic linearizations.

In section 5.2, we tabulate our results in detail for the generalized standard triples for many common polynomial bases in sections 5.3.2, 5.3.3, and 5.3.4. In section 5.4, we give the strict equivalence of generalized standard triples for any polynomial basis. We give specific proofs in section 5.5.

5.1.2 Notation and definition of a generalized standard triple

If $\boldsymbol{L}(z) = z\boldsymbol{C}_1 - \boldsymbol{C}_0 \in \mathbb{C}^{N\times N}$ (usually $N = n\ell$ but not always; for Lagrange and Hermite interpolational bases some constructions use $N = (n+2)\ell$ and others $N = (n+1)\ell$) is a linearization of $\boldsymbol{P}(z)$, then $\det(\boldsymbol{P}(z)) = \det(\boldsymbol{L}(z)) = \det(z\boldsymbol{C}_1 - \mathrm{C}_0)$. The eigenvalues of $\boldsymbol{P}$ are thus computable from the eigenvalues of $\boldsymbol{L}$. For instance, this can be done with `eig(C0,C1)` in Matlab, or `Eigenvalues(C[0],C[1])` in Maple if the matrix variables are defined appropriately (and are of complex floating-point type in Maple).

A *standard pair* $(\boldsymbol{X}, \boldsymbol{T})$ for a regular matrix polynomial $\boldsymbol{P}(\lambda)$ expressed in the monomial basis is defined in [17] or in [24] as having the properties $\boldsymbol{X}$ has dimension $n \times n\ell$, $\boldsymbol{T}$ has dimension $n\ell \times n\ell$, and

$$\sum_{k=0}^{\ell} \boldsymbol{P}_k \boldsymbol{X}\boldsymbol{T}^k = 0$$

and that the $n\ell$ by $n\ell$ matrix

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{X}\boldsymbol{T} \\ \vdots \\ \boldsymbol{X}\boldsymbol{T}^{\ell-1} \end{bmatrix}$$

difference allows for greater numerical stability.

[2]Transposition and flipping give altogether four common variations of companion matrix pencils. Other variations are possible (indeed, any similarity transformation will work) but these are the main variations seen in the literature. We include these variations in enough detail to help the reader with the bookkeeping.

is nonsingular. We can then define a third matrix

$$\boldsymbol{Y} = \boldsymbol{Q}^{-1} \begin{bmatrix} \boldsymbol{0}_n \\ \vdots \\ \boldsymbol{0}_n \\ \boldsymbol{I}_n \end{bmatrix}$$

and say that the triple $(\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{Y})$ is a *standard triple* for a *monic* $\boldsymbol{P}(\lambda)$. It is pointed out in [24] that monicity of $\boldsymbol{P}(\lambda)$ is not required for many of the formulæ to do with standard pairs (but is required for some). Theorem 2.6 of [17], which states that if there are matrices $\boldsymbol{X}$, $\boldsymbol{T}$, and $\boldsymbol{Y}$ of dimension $n \times n\ell$, $n\ell \times n\ell$, and $n\ell \times n$ for which

$$\boldsymbol{P}^{-1}(\lambda) = \boldsymbol{X}(\lambda \boldsymbol{I}_n - \boldsymbol{T})^{-1}\boldsymbol{Y}$$

then $(\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{Y})$ is a standard triple for $\boldsymbol{P}(\lambda)$. This construction is clearly tied to the monomial basis, and we would like to extend this to a form for other bases, and also to the non-monic case. In particular we would like the following extension of Theorem 2.6 in [17] or Theorem 12.1.4 in [16] to be available: If a matrix $\boldsymbol{X} \in \mathbb{C}^{n \times N}$, the pencil $\boldsymbol{L}(z)$, and a matrix $\boldsymbol{Y} \in \mathbb{C}^{N \times n}$ satisfies

$$\boldsymbol{P}^{-1}(z) = \boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y} \tag{5.1}$$

for $z \notin \Lambda(\boldsymbol{P})$ (the set of polynomial eigenvalues of $\boldsymbol{P}$), then $\boldsymbol{X}$, $\boldsymbol{L}(z)$, and $\boldsymbol{Y}$ form a *generalized standard triple* for $\boldsymbol{P}(\lambda)$. This obviously requires regularity of $\boldsymbol{P}$.

This new definition would allow $N > n\ell$ and not just $N = n\ell$. Note that the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ do not depend on z, but the linearization $\boldsymbol{L}(z)$ does, albeit only linearly; we could instead have chosen to use the words "standard quadruple" to mean $(\boldsymbol{X}, \boldsymbol{C}_1, \boldsymbol{C}_0, \boldsymbol{Y})$ where z does not appear of any of these matrices, but this seems to be a matter of aesthetics only.

In the case when the leading coefficient $\boldsymbol{A}_\ell$ of the matrix polynomial is nonsingular, one can reduce to the monic case in any of several ways. This is done in several places in the literature, and thus our generalized standard triple is not very new in allowing for a non-identity matrix coefficient of λ in the linearization. For various reasons we do not wish to do this here; for instance, if the polynomial basis is not degree-graded, e.g. for the Bernstein basis, then it is not clear what one should use for the leading coefficient. Similarly for the Lagrange and Hermite interpolational bases.

All we will need for the purposes of this paper is that the representation displayed in equation (5.1) holds (whether we say that it involves a "triple" or a "quadruple"). The representation itself is what is useful in the recursive construction of algebraic linearizations.

There is some risk of confusion with this definition: not all properties of standard triples may hold for "generalized standard triples", and so we should take some care.

Similarity

If $\boldsymbol{X}$, $z\boldsymbol{C}_1 - \boldsymbol{C}_0$, and $\boldsymbol{Y}$ form a generalized standard triple, then so also do $\boldsymbol{XS}$, $\boldsymbol{S}^{-1}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)\boldsymbol{S}$, and $\boldsymbol{S}^{-1}\boldsymbol{Y}$ for any nonsingular matrix $\boldsymbol{S}$ of dimension N by N.

Lemma 5.1.1 *If $\boldsymbol{S}$ is nonsingular*[3] *and $\boldsymbol{B}_1 = \boldsymbol{S}^{-1}\boldsymbol{C}_1\boldsymbol{S}$ and $\boldsymbol{B}_0 = \boldsymbol{S}^{-1}\boldsymbol{C}_0\boldsymbol{S}$ so that the pencil $z\boldsymbol{B}_1 - \boldsymbol{B}_0$ has the same generalized eigenvalues as $z\boldsymbol{C}_1 - \boldsymbol{C}_0$, then another standard triple for $\boldsymbol{P}(z)$ is $\widetilde{\boldsymbol{X}}$, $z\boldsymbol{B}_1 - \boldsymbol{B}_0$, $\widetilde{\boldsymbol{Y}}$ where $\widetilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{S}$ and $\widetilde{\boldsymbol{Y}} = \boldsymbol{S}^{-1}\boldsymbol{Y}$.*

Proof

$$\begin{aligned}\boldsymbol{P}^{-1}(z) &= \boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y} \\ &= \boldsymbol{X}\boldsymbol{S}\boldsymbol{S}^{-1}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{S}\boldsymbol{S}^{-1}\boldsymbol{Y} \\ &= (\boldsymbol{X}\boldsymbol{S})(\boldsymbol{S}^{-1}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)\boldsymbol{S})^{-1}\boldsymbol{S}^{-1}\boldsymbol{Y}\,.\end{aligned}$$

♮

Several similarities are used very frequently. For convenience we describe two of the most common explicitly here.

Lemma 5.1.2 (Flipping) *Put $\boldsymbol{J}$ = the $N \times N$ "anti-identity", also called the sip matrix, for standard involutory permutation, $\boldsymbol{J}_{i,j} = 0$ unless $i + j = N + 1$ when $\boldsymbol{J}_{i,N+1-i} = 1$. Then $\boldsymbol{J}^2 = \boldsymbol{I}$ and the "flipped" linearization $\boldsymbol{L}_{\boldsymbol{F}}(z) = \boldsymbol{J}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)\boldsymbol{J}$ has in its generalized standard triple the matrices $\boldsymbol{X}_{\boldsymbol{F}} = \boldsymbol{X}\boldsymbol{J}$ and $\boldsymbol{Y}_{\boldsymbol{F}} = \boldsymbol{J}\boldsymbol{Y}$. The paper [27] calls this matrix "$\boldsymbol{R}$".*

Proof Immediate. ♮

Remark Flipping switches both the order of the equations and the order of the variables. It obviously does not change eigenvalues. Flipping, transposition, and flipping-with-transposition give four common equivalent linearizations [31].

5.1.3 Algebraic linearizations

An *algebraic linearization*, as referred to in the title of this present note, is defined in [6] as a linearization of a matrix polynomial $\boldsymbol{H}(\lambda) = z\boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{C}$ constructed recursively from linearizations of the lower-degree component matrix polynomials $\boldsymbol{A}(\lambda)$ and $\boldsymbol{B}(\lambda)$, together with a constant matrix $\boldsymbol{C}$. Algebraic linearizations are typically strong linearizations, which not only preserve eigenvalues but also their partial multiplicities, even at infinity [6]. Algebraic linearizations offer a new, potentially more numerically stable, class of linearizations. The recursive construction of algebraic linearizations relies on the generalized standard triples of each of the component matrix polynomials, and (as does the unrelated paper [29]) allows different polynomial bases to be used for each component. This present note provides some explicit formulas for generalized standard triples in various bases, for reference. As one reviewer points out, these formulas *could* simply be obtained by reading the proofs that these linearizations are indeed linearizations; one purpose of this paper is simply convenience.

If $\boldsymbol{A}(\lambda)$ and $\boldsymbol{B}(\lambda)$ have the generalized standard triple representations $\boldsymbol{A}^{-1}(\lambda) = \boldsymbol{X}_A(\lambda\boldsymbol{D}_A - \boldsymbol{E}_A)^{-1}\boldsymbol{Y}_A$ and $\boldsymbol{B}^{-1}(\lambda) = \boldsymbol{X}_B(\lambda\boldsymbol{D}_B - \boldsymbol{E}_B)^{-1}\boldsymbol{Y}_B$, then the pencil $\lambda\boldsymbol{D}_H - \boldsymbol{E}_H$ is a linearization of

[3] We only use similarities in this section, but as a referee points out, we could also use equivalences, with different matrices $\boldsymbol{S}$ on the left and right.

$\boldsymbol{H}(\lambda) = \lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{C}$, where the matrices $\boldsymbol{D}_H$ and $\boldsymbol{E}_H$ are given as follows:

$$\boldsymbol{D}_H = \begin{bmatrix} \boldsymbol{D}_A & & \\ & \boldsymbol{I}_n & \\ & & \boldsymbol{D}_B \end{bmatrix}$$

and

$$\boldsymbol{E}_H = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0}_{N_A,n} & -\boldsymbol{Y}_A\boldsymbol{C}\boldsymbol{X}_B \\ -\boldsymbol{X}_A & \boldsymbol{0}_n & \boldsymbol{0}_{n,N_B} \\ \boldsymbol{0}_{N_B,N_A} & -\boldsymbol{Y}_B & \boldsymbol{B} \end{bmatrix}$$

For a proof and some examples, see [6].

5.2 Representation of matrix polynomials using generalized standard triples

Since the $\phi_k(x)$, $0 \le k \le \ell$ form a basis, we may express the polynomial 1 in that basis: then $1 = \sum_{k=0}^{n-1} e_k \phi_k(z)$ defines the coefficients e_k uniquely. Putting

$$\boldsymbol{X} = \begin{bmatrix} e_{n-1} & e_{n-2} & \cdots & e_1 & e_0 \end{bmatrix} \otimes \boldsymbol{I}$$

always gives our generalized standard triple $\boldsymbol{P}^{-1}(z) = \boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y}$ with $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n & \boldsymbol{0}_n & \cdots & \boldsymbol{0}_n \end{bmatrix}^T$. The proof is simple and (with appropriate modifications for the bases) universal: the change of basis matrix $\boldsymbol{\Phi}$ has as its last row of its inverse the coefficients of 1 in that basis. This is exactly the translation of the monomial standard triple element $\boldsymbol{X}$ into the new basis.

$$\begin{aligned} \boldsymbol{P}^{-1}(z) &= \boldsymbol{X}\left(z\boldsymbol{C}_1 - \boldsymbol{C}_0\right)^{-1}\boldsymbol{Y} \\ &= \boldsymbol{X}\left(z\boldsymbol{E}\boldsymbol{C}_{1,m}\boldsymbol{F} - \boldsymbol{E}\boldsymbol{C}_{0,m}\boldsymbol{F}\right)^{-1}\boldsymbol{Y} \\ &= \boldsymbol{X}\boldsymbol{\Phi}^{-1}\left(z\boldsymbol{C}_{1,m} - \boldsymbol{C}_{0,m}\right)^{-1}\boldsymbol{E}^{-1}\boldsymbol{Y}\,. \end{aligned}$$

Thus the new $\boldsymbol{X}$ is as claimed. It is a separate matter to show that $\boldsymbol{E}^{-1}[\boldsymbol{I}, 0, \ldots, 0]^T = [\boldsymbol{I}, 0, \ldots, 0]^T$ again, but it always is for the bases that we consider.

Remark There are linearizations not explicitly considered in this paper; for instance, a referee has pointed out that when a matrix polynomial is expressed in a basis where the elements satisfy a linear recurrence, then there is an automatic way to build what is called a CORK linearization. See [18] and [33] for details. Whatever the linearization, though, the previous universal theorem shows how to construct the generalized standard triple.

That proof sketch may not be convincing, in part because details are omitted. In what follows we examine specific cases in detail and supply specific proofs for each basis. Indeed, much of the utility of this paper is simply writing down those details, which will allow easier programming for the uses of these generalized standard triples.

5.3 Examples of generalized standard triples

In this section, we tabulate generalized standard triples for four classes of linearizations. We do so by examples of companion pencils, leaving the reader to do the necessary tensor products to produce linearizations. This saves some space in the presentation. In contrast, in section 5.5 where we gave proofs, we do so in full generality.

5.3.1 Companion matrices

In the special case $n = 1$, a linearization is usually called a "companion pencil" or Frobenius pencil[4]. Thus finding roots of a scalar polynomial can be done by finding generalized eigenvalues of the companion pencil. In the monic case, $\boldsymbol{C}_1$ becomes the identity matrix and the generalized eigenproblem becomes a standard eigenproblem. Kublanovskaya calls these "accompanying pencils" in [22]. For bases other than the monomial, the unfortunate nomenclature "colleague matrix" or "comrade matrix" is also used. This nomenclature hinders citation search and we prefer "generalized companion", if a distinction is needed. See [26].

Construction of a linearization from a companion pencil is a simple matter of the Kronecker (tensor) product: given $\boldsymbol{C}_1, \boldsymbol{C}_0 \in \mathbb{C}^{n\times n}$, take $\widetilde{\boldsymbol{C}_1} = \boldsymbol{C}_1 \otimes \boldsymbol{I}_n$ and then replace each block $p_k\boldsymbol{I}_n$ with the corresponding matrix coefficient $\boldsymbol{P}_k \in \mathbb{C}^{r\times r}$ (the first p_k, in $p_k\boldsymbol{I}_n$, is the symbolic coefficient from $p(z) = \sum_{k=0}^{\ell} p_k\phi_k(z)$; the matrix coefficient $\boldsymbol{P}_k \in \mathbb{C}^{r\times r}$ is from $\boldsymbol{P}(z) = \sum_{k=0}^{\ell} \boldsymbol{P}_k\phi_k(z)$.) This will be clearer by example.

5.3.2 Bases with three-term recurrence relations

The monomial basis, the shifted monomial basis, the Taylor basis, the Newton interpolational bases, and many common orthogonal polynomial bases all have three-term recurrence relations that, except for initial cases, can be written

$$z\phi_k(z) = \alpha_k\phi_{k+1}(z) + \beta_k\phi_k(z) + \gamma_k\phi_{k-1}(z)\,.$$

We give a selection in table 5.1, and refer the reader to Section 18.9 of the Digital Library of Mathematical Functions (`dlmf.nist.gov`) for more. See also [15].

[4]The *Frobenius form* of a matrix is related, but different: see for instance [30].

$\phi_k(z)$	Name	α_k	β_k	γ_k	ϕ_0	ϕ_1
z^k	monomial	1	0	0	1	z
$(z-a)^k$	shifted monomial	1	a	0	1	$z-a$
$(z-a)^k/k!$	Taylor	$n+1$	a	0	1	$z-a$
$\prod_{j=0}^{k-1}(z-\tau_j)$	Newton interpolational	1	τ_n	0	1	$z-\tau_0$
$T_k(z)=\cos\left(k\cos^{-1}(z)\right)$	Chebyshev	$1/2$	0	$1/2$	1	z
$P_k(z)$	Legendre	$(k+1)/(2k+1)$	0	$k/(2k+1)$	1	z

Table 5.1: A short list of three-term recurrence relations for some important polynomial bases. For a more comprehensive list, see The Digital Library of Mathematical Functions. These relations and others are coded in Walter Gautschi's packages OPQ and SOPQ [15] and in the `MatrixPolynomialObject` implementation package in Maple (see [19]).

For all such bases, we have the companion pencil[5]

$$\boldsymbol{C}_1 = \begin{bmatrix} \frac{p_5}{\alpha_4} & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

$$\boldsymbol{C}_0 = \begin{bmatrix} -p_4 + \frac{\beta_4}{\alpha_4}p_5 & -p_3 + \frac{\gamma_4}{\alpha_4}p_5 & -p_2 & -p_1 & -p_0 \\ \alpha_3 & \beta_3 & \gamma_3 & & \\ & \alpha_2 & \beta_2 & \gamma_2 & \\ & & \alpha_1 & \beta_1 & \gamma_1 \\ & & & \alpha_0 & \beta_0 \end{bmatrix}$$

and

$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\boldsymbol{Y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} .$$

For instance, a *flipped and transposed* pencil of this class for the Chebyshev case is[6]

$$\boldsymbol{L}(z) = \begin{bmatrix} z & -\frac{1}{2} & & & p_0 \\ -1 & z & -\frac{1}{2} & & p_1 \\ & -\frac{1}{2} & z & -\frac{1}{2} & p_2 \\ & & -\frac{1}{2} & z & p_3 + p_5 \\ & & & -\frac{1}{2} & 2zp_5 + p_4 \end{bmatrix}$$

[5]For exposition, we follow Peter Lancaster's dictum, namely that the 5×5 case almost always gives the idea.
[6]For the matrix polynomial case, each $\boldsymbol{P}_k$ would be transposed.

has flipped and transposed $\boldsymbol{X} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}$, $\boldsymbol{Y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$. As another instance, a Newton interpolational basis on the nodes $\tau_0, \tau_1, \ldots, \tau_5$ has a companion pencil

$$z \begin{bmatrix} p_5 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} - \begin{bmatrix} -p_4 + \tau_4 p_5 & -p_3 & -p_2 & -p_1 & -p_0 \\ 1 & \tau_3 & & & \\ & 1 & \tau_2 & & \\ & & 1 & \tau_1 & \\ & & & 1 & \tau_0 \end{bmatrix}.$$

The corresponding linearization is

$$z \begin{bmatrix} \boldsymbol{P}_5 & & & & \\ & \boldsymbol{I}_n & & & \\ & & \boldsymbol{I}_n & & \\ & & & \boldsymbol{I}_n & \\ & & & & \boldsymbol{I}_n \end{bmatrix} - \begin{bmatrix} -\boldsymbol{P}_4 + \tau_4 \boldsymbol{P}_5 & -\boldsymbol{P}_3 & -\boldsymbol{P}_2 & -\boldsymbol{P}_1 & -\boldsymbol{P}_0 \\ \boldsymbol{I}_n & \tau_3 \boldsymbol{I}_n & & & \\ & \boldsymbol{I}_n & \tau_2 \boldsymbol{I}_n & & \\ & & \boldsymbol{I}_n & \tau_1 \boldsymbol{I}_n & \\ & & & \boldsymbol{I}_n & \tau_0 \boldsymbol{I}_n \end{bmatrix}.$$

5.3.3 The Bernstein basis

The set of polynomials $\{B_k^\ell(z)\}_{k=0}^\ell$ is a set of $\ell + 1$ polynomials each of exact degree ℓ that forms a basis for polynomials of degree at most ℓ. They have many applications, for example in Computer Aided Geometric Design (CAGD), and many important properties including that of optimal polynomial evaluation condition number over all bases positive on $[0, 1]$. They do not satisfy a simple three term recurrence relation of the form discussed in section 5.3.2. See [13], [11], and [12] for more details of Bernstein bases.

A companion pencil for $p_5(z) = \sum_{k=0}^5 p_k B_k^5(z)$ is

$$\boldsymbol{C}_1 = \begin{bmatrix} -p_4 + \frac{1}{5} p_5 & -p_3 & -p_2 & -p_1 & -p_0 \\ 1 & \frac{2}{4} & & & \\ & 1 & \frac{3}{3} & & \\ & & 1 & \frac{4}{2} & \\ & & & 1 & \frac{5}{1} \end{bmatrix}$$

$$\boldsymbol{C}_0 = \begin{bmatrix} -p_4 & -p_3 & -p_2 & -p_1 & -p_0 \\ 1 & 0 & & & \\ & 1 & 0 & & \\ & & 1 & 0 & \\ & & & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{X} = \begin{bmatrix} \frac{1}{5} & \frac{2}{5} & \frac{3}{5} & \frac{4}{5} & \frac{5}{5} \end{bmatrix}$$

$$\boldsymbol{Y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}.$$

We have $p^{-1}(z) = \boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y}$ if $p(z) \neq 0$. This pencil was first analyzed in [20] and [21]. One of the present authors independently invented and implemented a version of this linearization in Maple (except using $\boldsymbol{P}^{\mathrm{T}}(z)$, and reversed from the above form) in about 2004. For a review of Bernstein linearization, see [27]. For a proof of its numerical stability, see the original thesis [20]. We supply a proof in section 5.5. The standard triple is, we believe, new to this paper.

5.3.4 The Lagrange interpolational basis

There are by now several Lagrange basis pencils and linearizations. The use of barycentric forms means that Lagrange interpolation is efficient and numerically stable. For many sets of nodes (Chebyshev nodes on $[-1, 1]$, or roots of unity on the unit disk) the resulting interpolant is also well-conditioned, and can even be "better than optimal" [9], see also [5]. The linearization we use here is "too large" and has (numerically harmless in our experience) spurious roots at infinity[7]; for alternative formulations see [33], [28], [32]. Then pencil is $z\boldsymbol{C}_1 - \boldsymbol{C}_0$ where

$$\boldsymbol{C}_1 = \begin{bmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

$$\boldsymbol{C}_0 = \begin{bmatrix} 0 & -\rho_0 & -\rho_1 & -\rho_2 & -\rho_3 & -\rho_4 \\ \beta_0 & \tau_0 & & & & \\ \beta_1 & & \tau_1 & & & \\ \beta_2 & & & \tau_2 & & \\ \beta_3 & & & & \tau_3 & \\ \beta_4 & & & & & \tau_4 \end{bmatrix}.$$

Then $\det(\tau_k\boldsymbol{C}_1 - \boldsymbol{C}_0) = \rho_k$, $0 \leq k \leq 4$ and $\deg(z\boldsymbol{C}_1 - \boldsymbol{C}_0) \leq 4$. Thus, $p(z) = \det(z\boldsymbol{C}_1 - \boldsymbol{C}_0)$ interpolates the given data, assuming the τ_k are distinct. Here the barycentric weights β_k are found by partial fraction expansion of $\omega(z)^{-1}$ where

$$\omega(z) = (z - \tau_0)(z - \tau_1)(z - \tau_2)(z - \tau_3)(z - \tau_4)$$

is the node polynomial. Explicitly,

$$\frac{1}{\omega(z)} = \sum_{k=0}^{5} \frac{\beta_k}{z - \tau_k}$$

[7] This numerical harmlessness needs some explanation. In brief, Lagrange basis matrix polynomial eigenvalues will be well-conditioned only in a compact region determined by the interpolation nodes, and are increasingly ill-conditioned towards infinity; in practice this means only small changes in the data are needed to perturb large finite ill-conditioned eigenvalues out to infinity. Any eigenvalues produced numerically that are well outside the region determined by the interpolation nodes are likely easily perturbed all the way to infinity, and can be safely ignored.

so

$$\beta_k = \prod_{\substack{j=0 \\ j\neq k}}^{5} (\tau_k - \tau_j)^{-1} \, .$$

The $\boldsymbol{X}$ and $\boldsymbol{Y}$ for the standard triple are

$$\boldsymbol{X} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} ,$$
$$\boldsymbol{Y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$$

Notice in this case that for the linearization $N = (\ell + 2)n$ while deg $p \leq \ell$, and therefore there are at least $2n$ eigenvalues at infinity. This can be inconvenient if n is at all large.

5.3.5 Hermite interpolational basis

The companion pencil of the previous section has been extended to Hermite interpolational bases, where some of the nodes have "flowed together," collapsing to fewer distinct nodes[8]. We suppose that at each node τ_i, there are now $s_i \geq 1$ consecutive pieces of information known, namely $\boldsymbol{P}(\tau_i)$, $\boldsymbol{P}'(\tau_i)/1!$, $\boldsymbol{P}''(\tau_i)/2!$, and so on up to the last one, the value of the $s_i - 1$-th derivative at $z = \tau_i$, namely $\boldsymbol{P}^{(s_i-1)}(\tau_i)/(s_i - 1)!$. The integer s_i is called the *confluency* of the node. The known pieces of information are the local Taylor coefficients of the polynomial fitting the data:

$$\rho_{i,j} = \frac{f^{(j)}(\tau_i)}{j!} \, , \quad 0 \leq j \leq s_i - 1 \, .$$

Note that the derivative $\boldsymbol{P}'(z)$ of a matrix polynomial is a straightforward extension to matrices of the ordinary derivative. It is isomorphic to the matrix with entries that are the ordinary derivatives of the original matrix.

The companion pencil (that is, the scalar case) of the previous section changes to the following elegant form. The matrix $\boldsymbol{C}_1$ is unchanged,

$$\boldsymbol{C}_1 = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} ,$$

being $(\ell + 2)$ by $(\ell + 2)$ as before, although now

$$\ell = -1 + \sum_{i=0}^{m} s_i$$

[8] A formal definition can be found in [7], for instance. The essential idea is that given two distinct pieces of data $(\tau_k, p(\tau_k))$ and $(\tau_{k+1}, p(\tau_{k+1}))$, we also know the forward difference $(p_{k+1} - p_k)/(\tau_{k+1} - \tau_k)$. In the limit as one node approaches (flows towards) the other, we still know two pieces of information: $p(\tau_k)$ and $p'(\tau_k)$. Hermite interpolation captures this idea.

is the grade of the resulting polynomial. The matrix $\boldsymbol{C}_0$ changes, picking up transposed Jordan-like blocks for each distinct node. For instance, suppose we have two distinct nodes, τ_0 and τ_1. Suppose further that τ_0 has confluency $s_0 = 3$ while τ_1 has confluency $s_1 = 2$. This means that we know $f(\tau_0)$, ${}^{f'(\tau_0)}/_{1!}$, ${}^{f''(\tau_0)}/_{2!}$, $f(\tau_1)$ and ${}^{f'(\tau_1)}/_{1!}$. Then,

$$\boldsymbol{C}_0 = \begin{bmatrix} 0 & -{}^{f''(\tau_0)}/_{2!} & -{}^{f'(\tau_0)}/_{1!} & -f(\tau_0) & -{}^{f'(\tau_1)}/_{1!} & -f(\tau_1) \\ \beta_{02} & \tau_0 & & & & \\ \beta_{01} & 1 & \tau_0 & & & \\ \beta_{00} & & 1 & \tau_0 & & \\ \beta_{11} & & & & \tau_1 & \\ \beta_{10} & & & & 1 & \tau_1 \end{bmatrix}$$

Note the reverse ordering of the derivative values in this formulation. The barycentric weights β_{ij} again come from the partial fraction expansion of the reciprocal of the node polynomial

$$\omega(z) = \prod_{i=0}^{m} (z - \tau_i)^{s_i} .$$

That is,

$$\frac{1}{\omega(z)} = \sum_{i=0}^{m} \sum_{j=0}^{s_i - 1} \frac{\beta_{ij}}{(z - \tau_i)^{j+1}} .$$

For the standard triple, take in the scalar case

$$\boldsymbol{Y} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}$$

but for $\boldsymbol{X}$ take the coefficients of the expansion of the polynomial 1 in this particular Hermite interpolational basis: it is equal to 1 at each node but has all derivatives zero at each node. That is, put

$$\begin{cases} \rho_{ij} = 1 & \text{if } j = 0 , \\ 0 & \text{otherwise} , \end{cases}$$

and sort them in order:

$$\boldsymbol{X} = \begin{bmatrix} 0 & \rho_{0,s_0-1} & \rho_{0,s_0-2} & \cdots & \rho_{0,0} & \rho_{1,s_1-1} & \cdots & \rho_{n,0} \end{bmatrix} .$$

For the earlier instance (two nodes, of confluency 3 and 2, respectively,

$$\boldsymbol{X} = \begin{bmatrix} 0 & \underbrace{0 \quad 0 \quad 1}_{\text{for } \tau_0} & \underbrace{0 \quad 1}_{\text{for } \tau_1} \end{bmatrix} .$$

Then

$$p^{-1}(z) = \boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y} .$$

Remark We may re-order the nodes in any fashion we like, and each ordering generates its own companion pencil (both Hermite and Lagrange). We may also find a pencil where the confluent data is ordered $p(\tau_i)$, ${}^{p'(\tau_i)}/_{1!}$, ${}^{p''(\tau_i)}/_{2!}$, etc., although we have not done so.

If there is just one node of confluency ℓ, we recover the standard Frobenius companion form (plus two infinite roots):

$$\begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & -p_{\ell-1} & -p_{\ell-2} & \cdots & -p_1 & -p_0 \\ 1 & \tau_0 & & & & \\ 0 & 1 & \tau_0 & & & \\ 0 & & 1 & \ddots & & \\ \vdots & & & \ddots & \tau_0 & \\ 0 & & & & 1 & \tau_0 \end{bmatrix}.$$

Here $p_k = {}^{p^{(k)}(\tau_0)}/_{k!}$ is the ordinary coefficient in the expansion $p(z) = \sum_{k=0}^{\ell} p_k(z-\tau_0)^k$. The numerical stability of these Hermite interpolational companions has been studied briefly [23] but much remains unknown. We confine ourselves in this paper to the study of the standard triple.

To make a linearization out of these companion pencils, take the Kronecker tensor product with $\boldsymbol{I}_n$, and insert the appropriate matrix polynomial values and derivative values.

Remark The modified linearizations of [33] also have standard triples that can be used for algebraic linearization, and arguably should be tabled here as well. They have the advantage of including fewer eigenvalues at infinity, or no spurious eigenvalues at infinity, which may lead to better algebraic linearizations. However, they are more involved, and we have less numerical experience with them. In particular we do not understand their dependence on the ordering of the nodes, and so we leave their analysis to a future study.

5.4 Strict equivalence of generalized standard triples for any polynomial basis

Theorem 5.4.1 *If $\phi_k(z)$ for $0 \le k \le \ell$ is one of the degree-graded polynomial bases (e.g. Chebyshev, Newton, Jacobi P) or a Bernstein basis although they are not degree-graded, then the companion pencil for a polynomial $p(z)$ expressed in that basis is strictly equivalent to the second companion pencil for same polynomial expressed in the monomial basis. That is, there exist unimodular matrices $\boldsymbol{E}$ and $\boldsymbol{F}$ for which $\boldsymbol{C}_{1,m} = \boldsymbol{E}\boldsymbol{C}_{1,\phi}\boldsymbol{F}$ and $\boldsymbol{C}_{0,m} = \boldsymbol{E}\boldsymbol{C}_{0,\phi}\boldsymbol{F}$. The matrix $\boldsymbol{E}$ will depend on the given polynomial $p(z)$. Here the subscript m is short for "monomial".*

Proof Denote the change-of-bases matrix for polynomials up to degree $\ell-1$ by $\boldsymbol{\Phi}$. This matrix is ℓ by ℓ. Then we have

$$\begin{bmatrix} \phi_{\ell-1}(z) \\ \phi_{\ell-2}(z) \\ \vdots \\ \phi_1(z) \\ \phi_0(z) \end{bmatrix} = \boldsymbol{\Phi} \begin{bmatrix} z^{\ell-1} \\ \vdots \\ z^2 \\ z \\ 1 \end{bmatrix}.$$

In all cases considered here, the companion matrix pencil for a polynomial $p(z)$ of exact degree ℓ has null vectors of the form

$$\boldsymbol{N} = \begin{bmatrix} \phi_{\ell-1}(\lambda) \\ \phi_{\ell-2}(\lambda) \\ \vdots \\ \phi_1(\lambda) \\ \phi_0(\lambda) \end{bmatrix}$$

where λ is a root of $p(z)$. That is,

$$\left(\lambda \boldsymbol{C}_{1,\phi} - \boldsymbol{C}_{0,\phi}\right) \boldsymbol{N} = \boldsymbol{0}\,.$$

Using the $\boldsymbol{\Phi}$ formula above, we have $\boldsymbol{F} = \boldsymbol{\Phi}$. By direct computation, we find that $\boldsymbol{E} = \boldsymbol{C}_{0,m}\boldsymbol{\Phi}^{-1}\boldsymbol{C}_{0,\phi}^{-1}$ necessarily giving $\boldsymbol{E}\boldsymbol{C}_{0,\phi}\boldsymbol{F} = \boldsymbol{C}_{0,m}$. Since $\boldsymbol{F} = \boldsymbol{\Phi}$ is unimodular and nonsingular, all that remains is to show that $\boldsymbol{E}$ is unimodular and nonsingular, and that it satisfies $\boldsymbol{E}\boldsymbol{C}_{1,\phi}\boldsymbol{F} = \boldsymbol{C}_{1,m}$ as well. Since $\boldsymbol{C}_{1,\phi}$ is the identity matrix except for the 1, 1 entry which is $a_\ell \neq 0$, the leading coefficient of the polynomial, this last is straightforward. Indeed, the action of premultiplying by $\boldsymbol{C}_{0,m}$ and postmultiplying by $\boldsymbol{C}_{0,\phi}^{-1}$ cancels the coefficient a_ℓ in the 1, 1 entry of $\boldsymbol{E}$, and thus by continuity this construction works even if $a_\ell = 0$, that is for polynomials of *grade* ℓ. Then since $\boldsymbol{\Phi}$ is upper triangular for degree-graded matrices and lower triangular for Bernstein matrices, and $\boldsymbol{E}$ has in that case zeros in the first column below the diagonal entry, $\boldsymbol{E}$ is unimodular.

To make this work for regular matrix polynomials of dimension n, we must use the tensor product $\boldsymbol{\Phi} \otimes \boldsymbol{I}$. ♮

We illustrate this proof with a four by four example in the Bernstein basis. If $p(z) = a_0 B_0^4(z) + a_1 B_1^4(z) + a_2 B_2^4(z) + a_3 B_3^4(z) + a_4 B_4^4(z)$, then

$$\boldsymbol{E} = \begin{bmatrix} 1 & 1/4\,a_0 - a_1 + 3/2\,a_2 & -1/6\,a_0 + 2/3\,a_1 & 1/4\,a_0 \\ 0 & 1/4 & 0 & 0 \\ 0 & 1/4 & 1/6 & 0 \\ 0 & 1/4 & 1/3 & 1/4 \end{bmatrix}$$

and

$$\boldsymbol{F} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ -6 & 6 & 0 & 0 \\ 4 & -8 & 4 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}.$$

Direct computation shows that both matrices are nonsingular irrespective of the values of the a_k, and that these transform the Bernstein companion pencil

$$\boldsymbol{C}_{0,\text{Bernstein}} = \begin{bmatrix} -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$\boldsymbol{C}_{1,\text{Bernstein}} = \begin{bmatrix} \frac{a_4}{4} - a_3 & -a_2 & -a_1 & -a_0 \\ 1 & 2/3 & 0 & 0 \\ 0 & 1 & 3/2 & 0 \\ 0 & 0 & 1 & 4 \end{bmatrix}$$

to

$$\boldsymbol{C}_{0,\text{monomial}} = \begin{bmatrix} -b_3 & -b_2 & -b_1 & -b_0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

with $-b_3 = 4\,a_0 - 12\,a_1 + 12\,a_2 - 4\,a_3$, $-b_2 = -6\,a_0 + 12\,a_1 - 6\,a_2$, $-b_1 = 4\,a_0 - 4\,a_1$, and $-b_0 = -a_0$. Also, $\boldsymbol{E}\boldsymbol{C}_{1,\text{Bernstein}}\boldsymbol{F}$ becomes the identity matrix except the $1,1$ entry is $b_4 = a_0 - 4\,a_1 + 6\,a_2 - 4\,a_3 + a_4$. These are the correct coefficients of the same polynomial expressed in the monomial basis.

5.4.1 The Lagrange interpolational case

As is usual we denote a Lagrange basis element on the distinct nodes $[\tau_0, \tau_1, \ldots, \tau_\ell]$ by $\ell_k(z) = \beta_k \prod_{j \neq k}(z - \tau_j)$. The use of the symbol ℓ by itself denotes an integer, namely the grade of the polynomial; the distinction is that ℓ with a subscript and a variable $\ell_j(z)$ denotes a Lagrange basis polynomial. This should not cause confusion.

Remark Many people think of "interpolation" as *meaning* the construction of a monomial basis polynomial $p(z) = a_\ell z^\ell + \cdots + a_0$ that fits the given data $p(\tau_k) = \rho_k$ for $0 \leq k \leq \ell$. This is naive. Interpolation truly means constructing a polynomial in any basis that we may use to evaluate $p(z)$ for z different to the values at the nodes. The most stable and convenient way to do this is by the barycentric form of Lagrange interpolants. Constructing an interpolant in a Newton basis by using divided differences or the monomial basis by using a Vandermonde matrix is changing the basis. Changing bases can have condition number exponential in the degree, and is usually a bad idea. In practice, we use the barycentric form [3]. For the purposes of proof of equivalence, we here occasionally use the Vandermonde matrix, and we think about the explicit construction of the monomial basis. This is not used in numerical practice.

Theorem 5.4.2 *If $\phi_k(z)$ for $0 \leq k \leq \ell$ is a Lagrange basis on distinct nodes τ_0, τ_1, …, τ_ℓ, then the $\ell + 2$ by $\ell + 2$ companion pencil for a polynomial $p(z)$ expressed in that basis is strictly equivalent to the second companion pencil for same polynomial expressed in the monomial basis but regarded as having grade $\ell+2$ (i.e. with zero coefficients padding the terms $z^{\ell+2}$ and $z^{\ell+1}$. That is, there exist unimodular matrices $\boldsymbol{E}$ and $\boldsymbol{F}$ for which $\boldsymbol{C}_{1,\phi} = \boldsymbol{E}\boldsymbol{C}_{1,m}\boldsymbol{F}$ and $\boldsymbol{C}_{0,\phi} = \boldsymbol{E}\boldsymbol{C}_{0,m}\boldsymbol{F}$, where now the second companion matrices in the monomial basis have dimension larger by two than needed for the exact degree. The matrix $\boldsymbol{E}$ will depend on the given polynomial $p(z)$.*

Proof For the Lagrange basis companion pencil, the right null vector is of the form

$$\boldsymbol{N} = \begin{bmatrix} w(\lambda) \\ \ell_0(\lambda) \\ \ell_1(\lambda) \\ \vdots \\ \ell_\ell(\lambda) \end{bmatrix}.$$

Here $w(z) = \prod_{k=0}^{\ell}(z - \tau_k)$ is of degree $\ell + 1$ and all the other entries, being elements of the Lagrange basis on $\ell + 1$ nodes, are of degree ℓ. Thus $\boldsymbol{\Phi}$ is dimension $\ell + 2$ by $\ell + 2$ and has first column e_1; that is, 1 in the first entry and zeros below it. The rest of the first row contains the coefficients of $w(z)$ expanded in the monomial basis. The remaining rows of $\boldsymbol{\Phi}$ contain the coefficients of the monomial expansions of the Lagrange basis polynomial; that is, the inverse of the transposed Vandermonde matrix, which relates the Lagrange interpolation basis to the monomial basis. Call that block $\hat{\boldsymbol{\Phi}}$. Explicitly,

$$\hat{\boldsymbol{\Phi}}^{-1} = \begin{bmatrix} \tau_0^\ell & \tau_1^\ell & \cdots & \tau_\ell^\ell \\ \tau_0^{\ell-1} & \tau_1^{\ell-1} & \cdots & \tau_\ell^{\ell-1} \\ \vdots & & & \vdots \\ \tau_0 & \tau_1 & \cdots & \tau_\ell \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{5.2}$$

Then $\boldsymbol{E}$ is straightforwardly seen to be $\mathrm{diag}(1, \hat{\boldsymbol{\Phi}}^{-1})$ and as in the degree-graded case $\boldsymbol{F} = \boldsymbol{\Phi}$. Further, $\boldsymbol{E}\boldsymbol{C}_{0,\phi}\boldsymbol{F}$ has as its first row $[0, -\rho\hat{\boldsymbol{\Phi}}]$. But $\rho\hat{\boldsymbol{\Phi}}$ is by the Vandermonde matrix simply the negative of the vector of monomial coefficients, $-[0, a_{\ell-1}, a_{\ell-2}, \ldots, a_0]$. More, the block underneath, namely $\hat{\boldsymbol{\Phi}}^{-1}\mathrm{diag}(\tau_0, \tau_1, \ldots, \tau_\ell)\hat{\boldsymbol{\Phi}}$ turns out to be simple, because

$$\hat{\boldsymbol{\Phi}}^{-1}\mathrm{diag}(\tau_0, \tau_1, \ldots, \tau_\ell) = \begin{bmatrix} \tau_0^{\ell+1} & \tau_1^{\ell+1} & \cdots & \tau_\ell^{\ell+1} \\ \tau_0^\ell & \tau_1^\ell & \cdots & \tau_\ell^\ell \\ \vdots & & & \vdots \\ \tau_0^2 & \tau_1^2 & \cdots & \tau_\ell^2 \\ \tau_0 & \tau_1 & \cdots & \tau_\ell \end{bmatrix}$$

Multiplying this by $\hat{\boldsymbol{\Phi}}$ shifts the identity matrix down one diagonal, giving the correct form for the second companion matrix $\boldsymbol{C}_{0,m}$.

As in the previous theorem, to construct $\boldsymbol{E}$ and $\boldsymbol{F}$ for n-dimensional regular matrix polynomials, we must take the tensor product $\boldsymbol{\Phi} \otimes \boldsymbol{I}$. ♮

Again we illustrate this proof with an example, this time of interpolation at the four points $[-1, -1/2, 1/2, 1]$. This will give rise to a polynomial of degree at most 3. If the values this polynomial takes at these four points are ρ_0, ρ_1, ρ_2, and ρ_3, then the equivalent polynomial expressed in the monomial basis has coefficients

$$\begin{aligned} a_0 &= -1/6\,\rho_0 + 2/3\,\rho_1 + 2/3\,\rho_2 - 1/6\,\rho_3 \\ a_1 &= 1/6\,\rho_0 - 4/3\,\rho_1 + 4/3\,\rho_2 - 1/6\,\rho_3 \\ a_2 &= 2/3\,\rho_0 - 2/3\,\rho_1 - 2/3\,\rho_2 + 2/3\,\rho_3 \\ a_3 &= 2/3\,\rho_3 - 4/3\,\rho_2 + 4/3\,\rho_1 - 2/3\,\rho_0\,. \end{aligned}$$

Expressing this as a polynomial of grade 5, that is $p(z) = 0 \cdot z^5 + 0 \cdot z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$, we get the second companion pencil

$$\boldsymbol{C}_{0,\text{monomial}} = \begin{bmatrix} 0 & -a_3 & -a_2 & -a_1 & -a_0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$\boldsymbol{C}_{1,\text{monomial}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The Lagrange basis companion matrix of [9], which admittedly has two extra infinite eigenvalues, is

$$\boldsymbol{C}_{0,\text{Lagrange}} = \begin{bmatrix} 0 & -\rho_3 & -\rho_2 & -\rho_1 & -\rho_0 \\ 2/3 & 1 & 0 & 0 & 0 \\ -4/3 & 0 & 1/2 & 0 & 0 \\ 4/3 & 0 & 0 & -1/2 & 0 \\ -2/3 & 0 & 0 & 0 & -1 \end{bmatrix}$$

and

$$\boldsymbol{C}_{1,\text{Lagrange}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

For these interpolation nodes, direct computation shows

$$\boldsymbol{E} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1/8 & -1/8 & -1 \\ 0 & 1 & 1/4 & 1/4 & 1 \\ 0 & 1 & 1/2 & -1/2 & -1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

and

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & -5/4 & 0 & 1/4 \\ 0 & 2/3 & 2/3 & -1/6 & -1/6 \\ 0 & -4/3 & -2/3 & 4/3 & 2/3 \\ 0 & 4/3 & -2/3 & -4/3 & 2/3 \\ 0 & -2/3 & 2/3 & 1/6 & -1/6 \end{bmatrix}.$$

5.5 Individual proofs

In section 5.2 we give a short universal proof of all the theorems in this section. Each individual proof in this section is therefore redundant. We include them here both for surety (giving two proofs of each theorem) and because they give insight and may be relevant to any numerical analysis. We will use the Schur Complement, in the following form: assuming a matrix $\boldsymbol{R}$ is partitioned into

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}$$

where $\boldsymbol{A} \in \mathbb{C}^{\ell\times\ell}$, $\boldsymbol{B} \in \mathbb{C}^{r\times(N-r)}$, $\boldsymbol{C} \in \mathbb{C}^{(N-r)\times r}$ and $\boldsymbol{D} \in \mathbb{C}^{(N-r)\times(N-r)}$ is assumed invertible, then

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{B}\boldsymbol{D}^{-1} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} & 0 \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}.$$

If further the Schur Complement $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$ is invertible, then

$$\boldsymbol{R}^{-1} = \begin{bmatrix} (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1} & -(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \end{bmatrix}$$

as can be verified by block multiplication of $\boldsymbol{R}$ or by $\boldsymbol{R}$. We will use $\boldsymbol{S}$ for the Schur Complement $\boldsymbol{S} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$. We will take $\boldsymbol{R} = z\boldsymbol{C}_1 - \boldsymbol{C}_0$. We may already use this to establish for each of the four classes of linearizations that

$$\det \boldsymbol{R} = \det(z\boldsymbol{C}_1 - \boldsymbol{C}_0) = \det(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}) \det \boldsymbol{D} = \det \boldsymbol{P}(z)\,. \tag{5.3}$$

Notice that the coefficients of $\boldsymbol{P}$ do not appear in the $\boldsymbol{D}$ block (in any of our linearizations). Thus the Schur Complement carries all the information particular to $\boldsymbol{P}(z)$. The computations verifying (5.3) are not obvious but in each case $\boldsymbol{D}^{-1}$ plays an important role. We will see that generically $\boldsymbol{D}^{-1}$ exists, except for isolated values of z, which we can safely ignore and recover later by continuity.
We take each case in turn.

Theorem 5.5.1 *If* $\boldsymbol{C}_1 = \operatorname{diag}\left[\dfrac{1}{\alpha_{\ell-1}}\boldsymbol{P}_\ell \quad \boldsymbol{I}_n \quad \boldsymbol{I}_n \quad \cdots \quad \boldsymbol{I}_n\right]$ *and*

$$\boldsymbol{C}_0 = \begin{bmatrix} \dfrac{\beta_{\ell-1}}{\alpha_{\ell-1}}\boldsymbol{P}_\ell - \boldsymbol{P}_{\ell-1} & \dfrac{\gamma_{\ell-1}}{\alpha_{\ell-1}}\boldsymbol{P}_\ell - \boldsymbol{P}_{\ell-2} & -\boldsymbol{P}_{\ell-1} & \cdots & -\boldsymbol{P}_0 \\ \alpha_{\ell-2}\boldsymbol{I}_\ell & \beta_{\ell-2}\boldsymbol{I}_n & \gamma_{\ell-2}\boldsymbol{I}_n & & \\ & \alpha_{\ell-3}\boldsymbol{I}_n & \beta_{\ell-3}\boldsymbol{I}_n & \gamma_{\ell-3}\boldsymbol{I}_n & \\ & & \ddots & \ddots & \gamma_1\boldsymbol{I}_n \\ & & & \alpha_0\boldsymbol{I}_n & \beta_0\boldsymbol{I}_n \end{bmatrix}$$

and $\boldsymbol{X} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \boldsymbol{I}_n \end{bmatrix}$ *and* $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I}_n & 0 & 0 & \cdots & 0 \end{bmatrix}$ *then* $\boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y} = \boldsymbol{P}^{-1}(z)$ *where* $\boldsymbol{P}(z) = \sum_{k=0}^{\ell} \boldsymbol{P}_k\phi_k(z)$ *except for such* z *that* $\det \boldsymbol{P}(z) = 0$. *As in section 5.3.2 the polynomials* $\phi_k(z)$ *satisfy* $z\phi_k = \alpha_k\phi_{k+1} + \beta_k\phi_k + \gamma_k\phi_{k-1}$, $\phi_{-1} = 0$, $\phi_0 = 1$, $\phi_1 = {}^{(z-\beta_0)}/_{\alpha_0}$. *In this theorem,* $\ell \geq 2$ *and* $N = \ell n$, *and if* $\boldsymbol{P}_\ell \neq 0_n$ *then degree* $\boldsymbol{P} = \ell$.

That this is a linearization is well-known; see e.g. [2]. We only prove $\boldsymbol{P}^{-1}(z) = \boldsymbol{X}\boldsymbol{R}^{-1}\boldsymbol{Y}$, here.

Proof We use the first block column of Schur Complement inverse formula

$$\boldsymbol{R}^{-1} = \begin{bmatrix} \boldsymbol{S}^{-1} & * \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{S}^{-1} & * \end{bmatrix}.$$

Here

$$\boldsymbol{D} = \begin{bmatrix} (z-\beta_{\ell-2})\boldsymbol{I}_n & -\gamma_{\ell-2}\boldsymbol{I}_n & & & \\ -\alpha_{\ell-3}\boldsymbol{I}_n & (z-\beta_{\ell-3})\boldsymbol{I}_n & -\gamma_{\ell-3}\boldsymbol{I}_n & & \\ & -\alpha_{\ell-4}\boldsymbol{I}_n & & \ddots & \\ & & \ddots & \ddots & -\gamma_1\boldsymbol{I}_n \\ & & & -\alpha_0\boldsymbol{I}_n & (z-\beta_0)\boldsymbol{I} \end{bmatrix}$$

is block tridiagonal, and

$$\boldsymbol{C} = \begin{bmatrix} -\alpha_{\ell-2}\boldsymbol{I}_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

By inspection $\boldsymbol{V} = -\boldsymbol{D}^{-1}\boldsymbol{C}$ is

$$\boldsymbol{V} = q\begin{bmatrix} \phi_{\ell-2}(z)\boldsymbol{I}_n \\ \vdots \\ \phi_2(z)\boldsymbol{I}_n \\ \phi_1(z)\boldsymbol{I}_n \\ \phi_0(z)\boldsymbol{I}_n \end{bmatrix}$$

for some constant q, because

$$-\alpha_k \phi_{k+1}(z) + (z - \beta_k)\phi_k(z) - \gamma_k \phi_{k-1}(z) = 0$$

for $k = 0, 1, \cdots, \ell - 3$. The constant q is obtained from

$$q \cdot (z - \beta_{\ell-2})\phi_{\ell-2}(z) - q \cdot \gamma_{\ell-2}\phi_{\ell-3}(z) = +\alpha_{\ell-2}$$

or

$$q \cdot [\phi_{\ell-1}(z)] = +1$$

So

$$q = \frac{+1}{\phi_{\ell-1}(z)} .$$

It follows that

$$\begin{aligned} \boldsymbol{S} &= \frac{z - \beta_{\ell-1}}{\alpha_{\ell-1}}\boldsymbol{P}_\ell + \boldsymbol{P}_{\ell-1} + \begin{bmatrix} \frac{-\gamma_{\ell-1}}{\alpha_{\ell-1}}\boldsymbol{P}_\ell + \boldsymbol{P}_{\ell-2} & \boldsymbol{P}_{\ell-3} & \cdots & \boldsymbol{P}_0 \end{bmatrix} \begin{bmatrix} \phi_{\ell-2}(z) \\ \phi_{\ell-3}(z) \\ \vdots \\ \phi_0(z) \end{bmatrix} \cdot \frac{1}{\phi_{\ell-1}(z)} \\ &= \frac{\frac{z - \beta_{\ell-1}}{\alpha_{\ell-1}}\phi_{\ell-1}(z)\boldsymbol{P}_\ell + \phi_{\ell-1}(z)\boldsymbol{P}_{\ell-1} - \frac{\gamma_{\ell-1}}{\alpha_{\ell-1}}\phi_{\ell-2}(z)\boldsymbol{P}_\ell + \phi_{\ell-2}(z)\boldsymbol{P}_{\ell-2} + \cdots + \phi_0(z)\boldsymbol{P}_0}{\phi_{\ell-1}(z)} \\ &= \frac{\sum_{k=0}^{\ell} \phi_k(z)\boldsymbol{P}_k}{\phi_{\ell-1}(z)} = \frac{\boldsymbol{P}(z)}{\phi_{\ell-1}(z)} . \end{aligned}$$

Thus

$$-\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{S}^{-1} = \begin{bmatrix} \phi_{\ell-2}(z)\boldsymbol{I}_n \\ \vdots \\ \phi_0(z)\boldsymbol{I}_n \end{bmatrix} \boldsymbol{P}^{-1}(z)$$

because $\frac{1}{\phi_{\ell-1}(z)}\boldsymbol{S}^{-1} = \boldsymbol{P}^{-1}(z)$. Finally, $\phi_0(z) = 1$, so the bottom block is $\boldsymbol{P}^{-1}(z)$, establishing that

$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \boldsymbol{I}_n \end{bmatrix}$$

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I}_n & 0 & \cdots & 0 & 0 \end{bmatrix}^T$$

will produce $\boldsymbol{X}\boldsymbol{R}^{-1}\boldsymbol{Y} = \boldsymbol{P}^{-1}(z)$.

Theorem 5.5.2 *Put*

$$\boldsymbol{C}_1 = \begin{bmatrix} \frac{1}{\ell}\boldsymbol{P}_\ell - \boldsymbol{P}_{\ell-1} & -\boldsymbol{P}_{\ell-2} & \cdots & -\boldsymbol{P}_1 & -\boldsymbol{P}_0 \\ \boldsymbol{I}_n & \frac{2}{\ell-1}\boldsymbol{I}_n & & & \\ & \boldsymbol{I}_n & \frac{3}{\ell-2}\boldsymbol{I}_n & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{I}_n & \frac{\ell}{1}\boldsymbol{I}_n \end{bmatrix}$$

and

$$\boldsymbol{C}_0 = \begin{bmatrix} -\boldsymbol{P}_{\ell-1} & -\boldsymbol{P}_{\ell-2} & \cdots & -\boldsymbol{P}_1 & -\boldsymbol{P}_0 \\ \boldsymbol{I}_n & 0 & & & \\ & \boldsymbol{I}_n & 0 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{I}_n & 0 \end{bmatrix}$$

and $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I}_n & 0 & \cdots & 0 & 0 \end{bmatrix}^T$ *with* $\boldsymbol{X} = \begin{bmatrix} \frac{1}{\ell}\boldsymbol{I}_n & \frac{2}{\ell}\boldsymbol{I}_n & \frac{3}{\ell}\boldsymbol{I}_n & \cdots & \frac{\ell}{\ell}\boldsymbol{I}_n \end{bmatrix}$. *Then* $\boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y} = \boldsymbol{P}^{-1}(z)$, *unless* $z \in \Lambda(\boldsymbol{P})$, *and* $\det \boldsymbol{P}(z) = \det \boldsymbol{R}(z) = \det(z\boldsymbol{C}_1 - \boldsymbol{C}_0)$.

Proof This linearization in proved e.g. in [27], but for convenience we supply one here as well. The Schur factoring is

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{B}\boldsymbol{D}^{-1} \\ 0 & \boldsymbol{I}_{N-r} \end{bmatrix} \begin{bmatrix} \boldsymbol{S} & 0 \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}$$

where $\boldsymbol{S} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$ is the Schur Complement. Here

$$A = \frac{z}{\ell}\boldsymbol{P}_\ell + (1-z)\boldsymbol{P}_{\ell-1}$$

$$\boldsymbol{B} = \begin{bmatrix} (1-z)\boldsymbol{P}_{\ell-2} & (1-z)\boldsymbol{P}_{\ell-3} & \cdots & (1-z)\boldsymbol{P}_0 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} (z-1)\boldsymbol{I}_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and

$$\boldsymbol{D} = \begin{bmatrix} \frac{2}{\ell-1}z\boldsymbol{I}_n & & & \\ (z-1)\boldsymbol{I}_n & \frac{3}{\ell-2}z\boldsymbol{I}_n & & \\ & (z-1)\boldsymbol{I}_n & \ddots & \\ & & \ddots & \\ & & (z-1)\boldsymbol{I}_n & \frac{\ell}{1}z\boldsymbol{I}_n \end{bmatrix}$$

Therefore $\boldsymbol{V} = \boldsymbol{D}^{-1}\boldsymbol{C}$ satisfies

$$\begin{bmatrix} \frac{2}{\ell-1}z\boldsymbol{I}_n & & & & \\ (z-1)\boldsymbol{I}_n & \frac{3}{\ell-2}z\boldsymbol{I}_n & & & \\ & (z-1)\boldsymbol{I}_n & \frac{4}{\ell-3}z\boldsymbol{I}_n & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & (z-1)\boldsymbol{I}_n & \frac{\ell}{1}z\boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_{\ell-1} \end{bmatrix} = \begin{bmatrix} (z-1)\boldsymbol{I}_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

So

$$\boldsymbol{v}_1 = \frac{\ell-1}{2}\left(\frac{z-1}{z}\right)\boldsymbol{I}_n = -\frac{\ell-1}{2}\left(\frac{1-z}{z}\right)\boldsymbol{I}_n$$

$$\boldsymbol{v}_2 = -\frac{\ell-2}{3}\cdot\boldsymbol{v}_1 = -\frac{\ell-2}{3}\cdot\frac{\ell-1}{2}\cdot\left(\frac{1-z}{z}\right)^2\boldsymbol{I}_n$$

$$\boldsymbol{v}_3 = -\frac{\ell-3}{4}\cdot\frac{\ell-2}{3}\cdot\frac{\ell-1}{2}\left(\frac{1-z}{z}\right)^3\boldsymbol{I}_n$$

and so on; by inspection, confirmed by a formal induction not given here,

$$\boldsymbol{v}_k = -\frac{(\ell-1)!}{(\ell-k-1)!(k+1)!}\left(\frac{1-z}{z}\right)^k\boldsymbol{I}_n = -\frac{1}{\ell}\binom{\ell}{k+1}\left(\frac{1-z}{z}\right)^k\boldsymbol{I}_n$$

for $k = 1, \cdots, \ell-1$. Thus

$$\begin{aligned}\boldsymbol{S} &= \frac{z}{\ell}\boldsymbol{P}_\ell + (1-z)\boldsymbol{P}_{\ell-1} + (1-z)\begin{bmatrix}\boldsymbol{P}_{\ell-2} & \boldsymbol{P}_{\ell-3} & \cdots & \boldsymbol{P}_0\end{bmatrix}\begin{bmatrix} \frac{1}{\ell}\binom{\ell}{2}\left(\frac{1-z}{z}\right)\boldsymbol{I}_n \\ \frac{1}{\ell}\binom{\ell}{3}\left(\frac{1-z}{z}\right)^2\boldsymbol{I}_n \\ \vdots \\ \frac{1}{\ell}\binom{\ell}{\ell}\left(\frac{1-z}{z}\right)^{\ell-1}\boldsymbol{I}_n \end{bmatrix} \\ &= \frac{1}{\ell z^{\ell-1}}\cdot\left[z^\ell\boldsymbol{P}_\ell + \ell z^{\ell-1}(1-z)\boldsymbol{P}_{\ell-1} + \binom{\ell}{2}z^{\ell-2}(1-z)^2\boldsymbol{P}_{\ell-2} + \cdots + \binom{\ell}{\ell}(1-z)^\ell\boldsymbol{P}_0\right] \\ &= \frac{\boldsymbol{P}(z)}{\ell z^{\ell-1}}\,.\end{aligned}$$

Hence

$$\begin{aligned}\det\boldsymbol{R} &= \det\boldsymbol{S}\det\boldsymbol{D} \\ &= \frac{\det\boldsymbol{P}(z)}{(\ell z^{\ell-1})^n}\cdot\left(\frac{2}{\ell-1}\cdot\frac{3}{\ell-2}\cdots\frac{\ell-1}{2}\cdot\ell\cdot z\right)^n \\ &= \det\boldsymbol{P}(z)\,.\end{aligned}$$

This establishes the linearization. Moreover,

$$\boldsymbol{S}^{-1} = \ell z^{\ell-1}\boldsymbol{P}^{-1}(z)$$

and the first column of $\boldsymbol{R}^{-1}$ is

$$\begin{bmatrix} \boldsymbol{S}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{S}^{-1} \end{bmatrix} = \begin{bmatrix} \ell z^{\ell-1}\boldsymbol{P}^{-1} \\ \ell z^{\ell-1}\cdot\frac{1}{\ell}\binom{\ell}{2}\left(\frac{1-z}{z}\right)\boldsymbol{P}^{-1} \\ \ell z^{\ell-1}\cdot\frac{1}{\ell}\binom{\ell}{3}\left(\frac{1-z}{z}\right)^2\boldsymbol{P}^{-1} \\ \ell z^{\ell-1}\cdot\frac{1}{\ell}\binom{\ell}{4}\left(\frac{1-z}{z}\right)^3\boldsymbol{P}^{-1} \\ \vdots \\ \ell z^{\ell-1}\cdot\frac{1}{\ell}\binom{\ell}{\ell}\left(\frac{1-z}{z}\right)^{\ell-1}\boldsymbol{P}^{-1} \end{bmatrix} = \begin{bmatrix} nz^{n-1}\boldsymbol{P}^{-1} \\ \binom{\ell}{2}z^{\ell-2}(1-z)\boldsymbol{P}^{-1} \\ \binom{\ell}{3}z^{\ell-3}(1-z)^2\boldsymbol{P}^{-1} \\ \vdots \\ \binom{\ell}{\ell}z^{0}(1-z)^{\ell-1}\boldsymbol{P}^{-1} \end{bmatrix}$$

We now notice that 1, expressed as a linear combination of

$$\binom{\ell}{1}z^{\ell-1}, \binom{\ell}{2}z^{\ell-2}(1-z), \cdots, \binom{\ell}{\ell}z^{0}(1-z)^{\ell-1}$$

is

$$\begin{aligned} 1 &= \frac{1}{\ell}\cdot\binom{\ell}{1}z^{\ell-1} + \frac{2}{\ell}\cdot\binom{\ell}{2}z^{\ell-2}(1-z) + \cdots + \frac{\ell}{\ell}\cdot\binom{\ell}{\ell}z^{0}(1-z)^{\ell-1} \\ &= \binom{\ell-1}{0}z^{\ell-1}(1-z)^0 + \binom{\ell-1}{1}z^{\ell-2}(1-z)^1 + \cdots + \binom{\ell-1}{\ell-1}z^{0}(1-z)^{\ell-1} \\ &= (z+1-z)^{\ell-1}\,. \end{aligned}$$

Indeed we use a degree-reduced Bernstein bases here, $\binom{\ell-1}{k}z^k(1-z)^{\ell-1-k}$, to express 1.
In any case, the coefficients of 1 give us our $\boldsymbol{X}$ vector: $\boldsymbol{X}\boldsymbol{R}^{-1}\boldsymbol{Y} = \boldsymbol{P}^{-1}(z)$. ♮

Theorem 5.5.3 (Lagrange Basis) *If $\boldsymbol{P}(z) \in \mathbb{C}^{n\times n}$ is of degree at most ℓ, and takes the values $\boldsymbol{\rho}_k \in \mathbb{C}^{n\times n}$ at the $\ell+1$ distinct nodes $z = \tau_k$, $0 \le k \le \ell$, i.e $\boldsymbol{P}(\tau_k) = \boldsymbol{\rho}_k \in \mathbb{C}^{n\times n}$, and the reciprocal of the node polynomial $\omega(z) = \prod_{k=0}^{\ell}(z-\tau_k)$ has partial fraction expansion*

$$\frac{1}{\omega(z)} = \sum_{k=0}^{\ell}\frac{\beta_k}{z-\tau_k}$$

then a linearization for $\boldsymbol{P}(z)$ is $z\boldsymbol{C}_1 - \boldsymbol{C}_0$ where $\boldsymbol{C}_1 = \operatorname{diag}(\boldsymbol{0}_n, \boldsymbol{I}_n, \boldsymbol{I}_n, \cdots, \boldsymbol{I}_n)$ with $\ell+2$ diagonal blocks, so $N = (\ell+2)r$, and

$$\boldsymbol{C}_0 = \begin{bmatrix} 0 & -\boldsymbol{\rho}_0 & -\boldsymbol{\rho}_1 & -\boldsymbol{\rho}_2 & \cdots & -\boldsymbol{\rho}_\ell \\ \beta_0\boldsymbol{I}_n & \tau_0\boldsymbol{I}_n & & & & \\ \beta_1\boldsymbol{I}_n & & \tau_1\boldsymbol{I}_n & & & \\ \beta_2\boldsymbol{I}_n & & & \tau_2\boldsymbol{I}_n & & \\ \vdots & & & & \ddots & \\ \beta_\ell\boldsymbol{I}_n & & & & & \tau_\ell\boldsymbol{I} \end{bmatrix}.$$

Moreover, if $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I}_n & 0 & 0 & \cdots & 0 \end{bmatrix}^T$ *and* $\boldsymbol{X} = \begin{bmatrix} 0_n & \boldsymbol{I}_n & \boldsymbol{I}_n & \cdots & \boldsymbol{I}_n \end{bmatrix}$ *then* $\boldsymbol{X}(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\boldsymbol{Y} = \boldsymbol{P}^{-1}(z)$ *where* $z \in \Lambda(\boldsymbol{P})$.

Proof Again we use the Schur complement: $\boldsymbol{S} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$ where here

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{0}_n \\ \boldsymbol{B} &= -\begin{bmatrix} \boldsymbol{\rho}_0 & \boldsymbol{\rho}_1 & \cdots & \boldsymbol{\rho}_\ell \end{bmatrix} \\ \boldsymbol{D}^{-1} &= \operatorname{diag}\left(\frac{1}{z-\tau_0}\boldsymbol{I}_n, \frac{1}{z-\tau_1}\boldsymbol{I}_n, \cdots, \frac{1}{z-\tau_\ell}\boldsymbol{I}_n \right) \\ \boldsymbol{C} &= \begin{bmatrix} \beta_0 \boldsymbol{I}_n \\ \beta_1 \boldsymbol{I}_n \\ \vdots \\ \beta_\ell \boldsymbol{I}_n \end{bmatrix} \end{aligned}$$

So

$$\boldsymbol{S} = \sum_{k=0}^{\ell} \frac{\beta_k}{z-\tau_k}\boldsymbol{\rho}_k = \omega(z)^{-1}\boldsymbol{P}(z) \tag{5.4}$$

from the first barycentric formula [3].
Note the first column of $\boldsymbol{R}^{-1}(z)$ is $\begin{bmatrix} \boldsymbol{S}^{-1} \\ -\boldsymbol{C}\boldsymbol{D}^{-1}\boldsymbol{S}^{-1} \end{bmatrix}$ or

$$\begin{bmatrix} \omega(z)\boldsymbol{P}^{-1}(z) \\ \left(\frac{\beta_0}{z-\tau_0}\right)\omega(z)\boldsymbol{P}^{-1}(z) \\ \left(\frac{\beta_1}{z-\tau_1}\right)\omega(z)\boldsymbol{P}^{-1}(z) \\ \vdots \\ \left(\frac{\beta_\ell}{z-\tau_\ell}\right)\omega(z)\boldsymbol{P}^{-1}(z) \end{bmatrix}$$

Note that $\sum_{k=0}^{\ell} \frac{\beta_k}{z-\tau_k} = \frac{1}{\omega(z)}$, so

$$\begin{bmatrix} 0 & \boldsymbol{I}_n & \boldsymbol{I}_n & \cdots & \boldsymbol{I}_n \end{bmatrix} \cdot \boldsymbol{R}^{-1} \begin{bmatrix} \boldsymbol{I}_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \left(\sum_{k=0}^{\ell} \frac{\beta_k}{z-\tau_k} \right) \omega(z)\boldsymbol{P}^{-1}(z) = \boldsymbol{P}^{-1}(z)$$

♮

Theorem 5.5.4 *In the Hermite interpolational bases on* $m+1$ *nodes each with coefficiency* s_i*, so the degree* ℓ *is at most* $-1 + \sum_{k=0}^{m} s_k$*, the barycentric weights are*

$$\frac{1}{\omega(z)} = \sum_{i=0}^{m} \sum_{j=0}^{s_i - 1} \frac{\beta_{ij}}{(z-\tau_i)^{j+1}}$$

As in the Lagrange case, $\boldsymbol{C}_1 = \operatorname{diag}(0, \boldsymbol{I}_n, \cdots, \boldsymbol{I}_n)$. $\boldsymbol{C}_0$ *is as below:*

$$\boldsymbol{C}_0 = \begin{bmatrix} 0 & -\hat{\boldsymbol{\rho}}_0 & -\hat{\boldsymbol{\rho}}_1 & \cdots & -\hat{\boldsymbol{\rho}}_m \\ \beta_{0,s_0-1}\boldsymbol{I}_n & \boldsymbol{J}_0^T & & & \\ \beta_{0,s_0-2}\boldsymbol{I}_n & & \boldsymbol{J}_1^T & & \\ \vdots & & & \ddots & \\ \beta_{m,s_m-1} & & & & \boldsymbol{J}_m^T \end{bmatrix}$$

where each block per node of data is collected in the $n \times m\ell$ *block matrix*

$$\hat{\boldsymbol{\rho}}_i = \begin{bmatrix} \boldsymbol{\rho}_{i,s_i-1} & \boldsymbol{\rho}_{i,s_i-2} & \cdots & \boldsymbol{\rho}_{i,0} \end{bmatrix}.$$

Each diagonal node block is a tensor product of a transposed Jordan block:

$$\boldsymbol{J}_i = \begin{bmatrix} \tau_i\boldsymbol{I}_n & & & & \\ \boldsymbol{I}_n & \tau_i\boldsymbol{I}_n & & & \\ & \boldsymbol{I}_n & \tau_i\boldsymbol{I}_n & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{I}_n & \tau_i\boldsymbol{I}_n \end{bmatrix}.$$

This form arises naturally on letting distinct Lagrange nodes flow together in a limit.

Express 1 *as a polynomial in this basis. Then* $1 \longleftrightarrow \rho_{00} = 1, \rho_{10} = 1, \cdots, \rho_{n0} = 1$ *and all other components are zero. Put*

$$\boldsymbol{X} = \begin{bmatrix} 0 & \underbrace{0 & \cdots & 0 & 1}_{S_0 \text{ entries}} & \underbrace{0 & \cdots & 0 & 1}_{S_1 \text{ entries}} & \cdots & 1 \end{bmatrix} \otimes \boldsymbol{I}_n$$

and $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I}_n & 0 & 0 & \cdots & 0 \end{bmatrix}$.

A similar but more involved computation than in theorem 5.5.3 gives

$$\boldsymbol{S} = \frac{1}{\omega(z)}\boldsymbol{P}(z) = \sum_{i=0}^{m}\sum_{j=0}^{s_i-1}\sum_{k=0}^{j}\beta_{ij}\boldsymbol{\rho}_{ik}(z-\tau_i)^{k-j-1} \tag{5.5}$$

and $\boldsymbol{D}^{-1}\boldsymbol{C}$ *contains just the correct powers of* $(z - \tau_i)$ *divided into* β_{ij} *to make the sums come out right; the inverse of the block*

$$\begin{bmatrix} (z-\tau_0)\boldsymbol{I}_n & & & & \\ -\boldsymbol{I}_n & (z-\tau_0)\boldsymbol{I}_n & & & \\ & -\boldsymbol{I}_n & \ddots & & \\ & & \ddots & & \\ & & & -\boldsymbol{I}_n & (z-\tau_0)\boldsymbol{I}_n \end{bmatrix}$$

is

$$\begin{bmatrix} \dfrac{1}{z-\tau_0}\boldsymbol{I}_n & & & & \\ \dfrac{1}{(z-\tau_0)^2}\boldsymbol{I}_n & \dfrac{1}{z-\tau_0}\boldsymbol{I}_n & & & \\ \dfrac{1}{(z-\tau_0)^3}\boldsymbol{I}_n & \dfrac{1}{(z-\tau_0)^2}\boldsymbol{I}_n & \dfrac{1}{z-\tau_0}\boldsymbol{I}_n & & \\ \vdots & & \ddots & & \\ \dfrac{1}{(z-\tau_0)^{s_0}}\boldsymbol{I}_n & & & & \dfrac{1}{z-\tau_0}\boldsymbol{I}_n \end{bmatrix}.$$

and thus each block is reminiscent of theorem 5.5.1, in fact.

Remark In every case $\boldsymbol{X}$ = [coefficients of 1] $\otimes \boldsymbol{I}$, $\boldsymbol{Y} = [1, 0, \cdots, 0] \otimes \boldsymbol{I}$. This is in agreement with our universal proof in section 5.2.

5.6 Examples

In this section, we will show some experiments done in Maple 2017 to demonstrate that standard triples introduced in section works for the different bases. We wrote our own code for constructing the linearizations rather than using Maple's built-in `CompanionMatrix` function since the result of the built-in function is the flipped and transposed version of the companion matrices compared to the structure in this paper.

For the following examples, we check the correctness of the standard triple for each of the following examples by rearranging the resolvent form

$$\begin{aligned} \boldsymbol{P}^{-1}(z) &= \boldsymbol{X}\left(z\boldsymbol{C}_1 - \boldsymbol{C}_0\right)^{-1}\boldsymbol{Y} \\ \boldsymbol{I}_n &= \boldsymbol{X}\left(z\boldsymbol{C}_1 - \boldsymbol{C}_0\right)^{-1}\boldsymbol{Y}\boldsymbol{P}(z)\,. \end{aligned}$$

Since these computations are done exactly, the result will exactly equal the identity matrix. For the Lagrange basis example, since we construct our companion matrices using τ and ρ instead of the matrix polynomial itself, $\boldsymbol{P}(z)$ is constructed using the barycentric Lagrange interpolation formula, which can be derived from equation (5.4). The Hermite interpolational basis examples are handled similarly to the Lagrange case, where $\boldsymbol{P}(z)$ is the Hermite interpolation polynomial, which can be derived from equation (5.5).

5.6.1 Bases with three-term recurrence relations

Chebyshev basis of the first kind

$$\begin{aligned} \boldsymbol{P}(z) = &\begin{bmatrix} 1/5 & 7/100 \\ -93/200 & -29/200 \end{bmatrix} T_0(z) + \begin{bmatrix} 53/300 & 7/60 \\ 2/25 & 3/50 \end{bmatrix} T_1(z) \\ &+ \begin{bmatrix} -9/80 & -13/80 \\ 57/400 & -47/400 \end{bmatrix} T_2(z) + \begin{bmatrix} -3/250 & -31/500 \\ -77/500 & 27/250 \end{bmatrix} T_3(z)\,. \end{aligned} \tag{5.6}$$

The standard triple for equation (5.6) is

$$\boldsymbol{C}_0 = \begin{bmatrix} -21/200 & -29/400 & -107/750 & -17/50 & -73/200 & 1/25 \\ -13/400 & -1/400 & -49/300 & -283/750 & -27/200 & 9/25 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} -33/250 & -9/25 & 0 & 0 & 0 & 0 \\ 3/25 & -37/250 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Then,

$$\boldsymbol{X}\,(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\,\boldsymbol{Y}\boldsymbol{P}(z) = \boldsymbol{I}_2\,.$$

which indicates that the standard triple is correct.

Newton Interpolational Basis

$$\tau = \left[\operatorname{seq}\left(\cos\left(\frac{\pi \cdot k}{3}\right), k = 0..3\right)\right] = [1, 1/2, -1/2, -1]$$

$$\begin{aligned} \boldsymbol{P}(z) = {} & \begin{bmatrix} 6 & 25 \\ -1 & 5 \end{bmatrix} \prod_{j=0}^{0}(z - \tau_j) + \begin{bmatrix} -80/3 & 25/3 \\ 43/3 & 94/3 \end{bmatrix} \prod_{j=0}^{1}(z - \tau_j) \\ & + \begin{bmatrix} 77/4 & 31/4 \\ 9/4 & -25/2 \end{bmatrix} \prod_{j=0}^{2}(z - \tau_j) + \begin{bmatrix} 86/5 & -61/5 \\ 4 & -48/5 \end{bmatrix} \prod_{j=0}^{3}(z - \tau_j) \end{aligned} \tag{5.7}$$

The standard triple for equation (5.7) is

$$\boldsymbol{C}_0 = \begin{bmatrix} -557/20 & -33/20 & 80/3 & -25/3 & -6 & -25 \\ -17/4 & 173/10 & -43/3 & -94/3 & 1 & -5 \\ 1 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} \frac{86}{5} & -\frac{61}{5} & 0 & 0 & 0 & 0 \\ 4 & -\frac{48}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

5.6.2 Bernstein basis

Non-singular leading coefficient case

$$\begin{aligned}\boldsymbol{P}(z) = &\begin{bmatrix} 4/25 & 99/100 \\ 9/100 & 3/5 \end{bmatrix} B_0^3(z) + \begin{bmatrix} -17/25 & 11/50 \\ -67/100 & 7/50 \end{bmatrix} B_1^3(z) \\ &+ \begin{bmatrix} -59/100 & -31/50 \\ 3/25 & -33/100 \end{bmatrix} B_2^3(z) + \begin{bmatrix} 41/50 & 21/50 \\ 18/25 & 9/50 \end{bmatrix} B_3^3(z)\,.\end{aligned} \tag{5.8}$$

The standard triple for equation (5.8) is

$$\boldsymbol{C}_0 = \begin{bmatrix} 59/100 & 31/50 & 17/25 & -11/50 & -4/25 & -99/100 \\ -3/25 & 33/100 & 67/100 & -7/50 & -9/100 & -3/5 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} 259/300 & 19/25 & 17/25 & -11/50 & -4/25 & -99/100 \\ 3/25 & 39/100 & 67/100 & -7/50 & -9/100 & -3/5 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 3 \end{bmatrix}$$

$$\boldsymbol{X} = \begin{bmatrix} 1/3 & 0 & 2/3 & 0 & 1 & 0 \\ 0 & 1/3 & 0 & 2/3 & 0 & 1 \end{bmatrix} \quad \boldsymbol{Y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Then,

$$\boldsymbol{X}\left(z\boldsymbol{C}_1 - \boldsymbol{C}_0\right)^{-1}\boldsymbol{Y}\boldsymbol{P}(z) = \boldsymbol{I}_2\,.$$

Singular leading coefficient case

$$\begin{aligned}\boldsymbol{P}(z) = &\begin{bmatrix} 29/100 & -8/25 \\ 7/10 & -1/100 \end{bmatrix} B_0^3(z) + \begin{bmatrix} -41/50 & 41/100 \\ -7/10 & 91/100 \end{bmatrix} B_1^3(z) \\ &+ \begin{bmatrix} 9/10 & 19/100 \\ 4/5 & 22/25 \end{bmatrix} B_2^3(z) + \begin{bmatrix} 1 & 1 \\ 9851/1980 & 0 \end{bmatrix} B_3^3(z)\,.\end{aligned} \tag{5.9}$$

Expressing equation (5.9) into the monomial basis, we have

$$\boldsymbol{P}(z) = \begin{bmatrix} 29/100 & -8/25 \\ 7/10 & -1/100 \end{bmatrix} + \begin{bmatrix} 29/100 & -8/25 \\ 7/10 & -1/100 \end{bmatrix} z + \begin{bmatrix} 849/100 & -57/20 \\ 87/10 & -57/20 \end{bmatrix} z^2 + \begin{bmatrix} -89/20 & 99/50 \\ -89/396 & 1/10 \end{bmatrix} z^3\,.$$

Taking the determinant of the leading coefficient

$$\det\left(\begin{bmatrix} -89/20 & 99/50 \\ -89/396 & 1/10 \end{bmatrix}\right) = (-89/20)\,(1/10) - (99/50)\,(-89/396) = 0\,,$$

we can observe that leading coefficient is singular, and thus, this matrix polynomial is non-monic. The standard triple for equation (5.9) is

$$\boldsymbol{C}_0 = \begin{bmatrix} -9/10 & -19/100 & 41/50 & -41/100 & -29/100 & 8/25 \\ -4/5 & -22/25 & 7/10 & -91/100 & -7/10 & 1/100 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} -17/30 & 43/300 & 41/50 & -41/100 & -29/100 & 8/25 \\ 5099/5940 & -22/25 & 7/10 & -91/100 & -7/10 & 1/100 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 3 \end{bmatrix}$$

$$\boldsymbol{X} = \begin{bmatrix} 1/3 & 0 & 2/3 & 0 & 1 & 0 \\ 0 & 1/3 & 0 & 2/3 & 0 & 1 \end{bmatrix} \quad \boldsymbol{Y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Then,

$$\boldsymbol{X}\,(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\,\boldsymbol{Y}\boldsymbol{P}(z) = \boldsymbol{I}_2\,.$$

5.6.3 Lagrange basis

Example

$$\tau = \left[\mathrm{seq}\left(\cos\left(\frac{\pi \cdot k}{2}\right), k = 0..2\right)\right] = [1, 0, -1] \tag{5.10}$$

$$\rho = [\boldsymbol{I}_2, \boldsymbol{I}_2, \boldsymbol{I}_2] = \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right]$$

$$\boldsymbol{C}_0 = \begin{bmatrix} 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 \\ 1/2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Using the barycentric Lagrange interpolation formula, we construct our matrix polynomial

$$\begin{aligned}\boldsymbol{P}(z) &= \begin{bmatrix} (z-1)\,z\,(z+1)\left(\frac{1}{2(z-1)} - \frac{1}{z} + \frac{1}{2(z+1)}\right) & 0 \\ 0 & (z-1)\,z\,(z+1)\left(\frac{1}{2(z-1)} - \frac{1}{z} + \frac{1}{2(z+1)}\right) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\end{aligned}$$

that corresponds to the given τ and ρ from equation 5.10. Therefore, $\boldsymbol{P}^{-1}(z)$

$$\boldsymbol{X}\,(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\,\boldsymbol{Y} = \boldsymbol{I}_2$$

5.6.4 Hermite interpolational basis

Polynomial case Let

$$\tau = \left[-1, -\frac{1}{2}, \frac{1}{2}, 1\right] \tag{5.11}$$

and

z	$P(z)$	$P'(z)$	$P''(z)$
$\tau_0 = -1$	1	0	0
$\tau_1 = -\frac{1}{2}$	1		
$\tau_2 = \frac{1}{2}$	1		
$\tau_3 = 1$	1	0	

Note that this polynomial is identically 1: its values at all nodes are 1, and all derivatives at all nodes are 0. This demonstrates explicitly that the degree of the polynomial is not necessarily revealed by the grade, which here is $\ell = 5$. The standard triple is then

$$\boldsymbol{C}_0 = \begin{bmatrix} 0 & 0 & -1 & -1 & -1 & 0 & 0 & -1 \\ 1/6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -25/36 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 32/27 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ -32/9 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 11/9 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 331/108 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Using the first barycentric representation, the Hermite interpolation polynomial of the given data is

$$P(z) = (z+1)^3\left(z+\frac{1}{2}\right)\left(z-\frac{1}{2}\right)(z-1)^2 \cdot \left(\frac{331}{108z+108} + \frac{11}{9(z+1)^2} + \frac{1}{3}(z+1)^{-3} - \frac{32}{9z+\frac{9}{2}} + \frac{32}{27z-\frac{27}{2}} - \frac{25}{36z-36} + \frac{1}{6}(z-1)^{-2}\right) = 1\,,$$

as discussed. Therefore the companion pencil has no finite eigenvalues, in exact arithmetic. Numerically, it can be expected to have eigenvalues around $O(1/\mu)^{1/7}$ where μ is the unit roundoff; here the exponent is 7, two more than the grade because only two of the spurious eigenvalues at infinity are detected and removed precisely [23]. Indeed that is what occurs (calculations not shown here). Returning to the example, calculating the resolvent form gives

$$\boldsymbol{X}\,(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\,\boldsymbol{Y} = 1\,,$$

and therefore (due to multiplying this by $P(z) = 1$), this shows that the standard triple for the Hermite interpolating basis is correct.

Matrix polynomial case Let

$$\tau = [0, 1]$$

and

z	$\boldsymbol{P}(z)$	$\boldsymbol{P}'(z)$
$\tau_0 = 0$	$\begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}$	
$\tau_1 = 1$	$\begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ -1 & 0 \end{bmatrix}$

Then, the standard triple is

$$\boldsymbol{C}_0 = \begin{bmatrix} 0 & 0 & -1 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{C}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

The Hermite interpolating polynomial is

$$\boldsymbol{P}(z) = \begin{bmatrix} z-1 & -2z^2+3z \\ -3z^2+5z-1 & 2z^2-4z+1 \end{bmatrix}$$

and the resolvent form is

$$\boldsymbol{X}\,(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\,\boldsymbol{Y} = \begin{bmatrix} \dfrac{-2z^2+4z-1}{6z^4-21z^3+23z^2-8z+1} & \dfrac{-2z^2+3z}{6z^4-21z^3+23z^2-8z+1} \\ \dfrac{-3z^2+5z-1}{6z^4-21z^3+23z^2-8z+1} & \dfrac{-z+1}{6z^4-21z^3+23z^2-8z+1} \end{bmatrix}.$$

Then,

$$\boldsymbol{X}\,(z\boldsymbol{C}_1 - \boldsymbol{C}_0)^{-1}\,\boldsymbol{Y}\boldsymbol{P}(z) = \boldsymbol{I}_2\,,$$

which indicates that the standard triples is correct.

5.7 Concluding remarks

The generalized standard triple (or standard quadruple, if you prefer) that we propose in this paper for convenience in algebraic linearization may have other uses. As pointed out on p. 28 of [17] many of the properties stated in that work for monic polynomials are valid for non-monic polynomials with the appropriate changes made. We have not attempted a comprehensive categorization of those changes for other purposes.

We have established that these generalized standard triples allow the resolvent representation for the matrix polynomial, equation (5.1), is useful for algebraic linearization.

Acknowledgments

We acknowledge the support of Western University, The National Science and Engineering Research Council of Canada, the Ontario Graduate Scholarship (OGS) program, the University of Alcalá, the Ontario Research Centre of Computer Algebra, and the Rotman Institute of Philosophy. Part of this work was developed while RMC was visiting the University of Alcalá, in the frame of the project Giner de los Rios. The authors would also like to thank Peter Lancaster for teaching RMC long ago the value of the 5×5 example. Similarly we thank John C. Butcher for the proper usage of the word "interpolational". In addition, thank Françoise Tisseur for her thorough comments on an earlier version of this paper.

Bibliography

[1] A. Amiraslani, R. M. Corless, and P. Lancaster, *Linearization of matrix polynomials expressed in polynomial bases*, IMA Journal of Numerical Analysis, 29 (2008), pp. 141–157.

[2] S. Barnett, *Some applications of the comrade matrix*, International Journal of Control, 21 (1975), pp. 849–855.

[3] J. Berrut and L. N. Trefethen, *Barycentric Lagrange interpolation*, SIAM Review, 46 (2004), pp. 501–517.

[4] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, ACM Transactions on Mathematical Software (TOMS), 39 (2013), p. 7.

[5] J. M. Carnicer, Y. Khiar, and J. M. Peña, *Optimal stability of the Lagrange formula and conditioning of the Newton formula*, Journal of Approximation Theory, (2017).

[6] E. Y. S. Chan, R. M. Corless, L. Gonzalez-Vega, J. R. Sendra, and J. Sendra, *Algebraic linearizations for matrix polynomials*, Linear Algebra and its Applications, 563 (2019), pp. 373–399.

[7] R. M. Corless and N. Fillion, *Polynomial and rational interpolation*, in A Graduate Introduction to Numerical Methods, Springer, 2013, pp. 331–401.

[8] R. M. Corless, N. Rezvani, and A. Amiraslani, *Pseudospectra of matrix polynomials that are expressed in alternative bases*, Mathematics in Computer Science, 1 (2007), pp. 353–374.

[9] R. M. Corless and S. M. Watt, *Bernstein bases are optimal, but, sometimes, Lagrange bases are better*, in Proceedings of SYNASC, Timisoara, 2004, pp. 141–153.

[10] F. M. Dopico, J. Pérez, and P. Van Dooren, *Block minimal bases ℓ-ifications of matrix polynomials*, arXiv preprint arXiv:1803.06306, (2018).

[11] R. Farouki and T. Goodman, *On the optimal stability of the Bernstein basis*, Mathematics of Computation of the American Mathematical Society, 65 (1996), pp. 1553–1566.

[12] R. T. Farouki, *The Bernstein polynomial basis: A centennial retrospective*, Computer Aided Geometric Design, 29 (2012), pp. 379–419.

[13] R. T. Farouki and V. T. Rajan, *On the numerical condition of polynomials in Bernstein form*, Computer Aided Geometric Design, 4 (1987), pp. 191–216.

[14] H. Fassbender and P. Saltenberger, *On vector spaces of linearizations for matrix polynomials in orthogonal bases*, Linear Algebra and its Applications, 525 (2017), pp. 59–83.

[15] W. Gautschi, *Orthogonal polynomials in MATLAB: Exercises and Solutions*, vol. 26, SIAM, 2016.

[16] I. Gohberg, P. Lancaster, and L. Rodman, *Indefinite Linear Algebra and Applications*, Springer, 2005.

[17] ——, *Matrix polynomials*, SIAM, 2009.

[18] S. Güttel and F. Tisseur, *The nonlinear eigenvalue problem*, Acta Numerica, 26 (2017), pp. 1–94.

[19] D. J. JEFFREY AND R. M. CORLESS, *Linear algebra in maple*, in Handbook of Linear Algebra, L. Hogben, ed., Chapman and Hall/CRC, 2013, ch. 89.

[20] G. F. JÓNSSON, *Eigenvalue methods for accurate solution of polynomial equations.*, PhD thesis, Center for Applied Mathematics, Cornell University, Ithaca, NY, 2001.

[21] G. F. JÓNSSON AND S. VAVASIS, *Solving polynomials with small leading coefficients*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 400–414.

[22] V. N. KUBLANOVSKAYA, *Methods and algorithms of solving spectral problems for polynomial and rational matrices*, Journal of Mathematical Sciences, 96 (1999), pp. 3085–3287.

[23] P. W. LAWRENCE AND R. M. CORLESS, *Numerical stability of barycentric Hermite root-finding*, in Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation, ACM, 2012, pp. 147–148.

[24] J. LIESEN AND C. MEHL, *Matrix polynomials*, in Handbook of Linear Algebra, L. Hogben, ed., Chapman and Hall/CRC, 2013, ch. 18.

[25] D. S. MACKEY, N. MACKEY, C. MEHL, AND V. MEHRMANN, *Vector spaces of linearizations for matrix polynomials*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 971–1004.

[26] D. S. MACKEY, N. MACKEY, AND F. TISSEUR, *Polynomial eigenvalue problems: Theory, computation, and structure*, in Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory, Springer, 2015, pp. 319–348.

[27] D. S. MACKEY AND V. PEROVIĆ, *Linearizations of matrix polynomials in Bernstein bases*, Linear Algebra and its Applications, 501 (2016), pp. 162–197.

[28] Y. NAKATSUKASA, V. NOFERINI, AND A. TOWNSEND, *Vector spaces of linearizations for matrix polynomials: a bivariate polynomial approach*, SIAM Journal on Matrix Analysis and Applications, 38 (2017), pp. 1–29.

[29] L. ROBOL, R. VANDEBRIL, AND P. V. DOOREN, *A framework for structured linearizations of matrix polynomials in various bases*, SIAM Journal on Matrix Analysis and Applications, 38 (2017), pp. 188–216.

[30] A. STORJOHANN, *An $O(n^3)$ algorithm for the Frobenius normal form*, in Proceedings of the 1998 international symposium on Symbolic and algebraic computation, ACM, 1998, pp. 101–105.

[31] O. TAUSSKY AND H. ZASSENHAUS, *On the similarity transformation between a matrix and its transpose*, Pacific Journal of Mathematics, 9 (1959), pp. 893–896.

[32] M. VAN BAREL AND F. TISSEUR, *Polynomial eigenvalue solver based on tropically scaled lagrange linearization*, Linear Algebra and its Applications, 542 (2018), pp. 186–208.

[33] R. Van Beeumen, W. Michiels, and K. Meerbergen, *Linearization of Lagrange and Hermite interpolating matrix polynomials*, IMA Journal of Numerical Analysis, 35 (2015), pp. 909–930.

Chapter 6

Numerical examples on backward stability of algebraic linearizations

6.1 Introduction

The polynomial eigenvalue problem (PEP) is to find the scalar λ and the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ to satisfy

$$\boldsymbol{P}(\lambda)x = 0 \text{ and } y^*\boldsymbol{P}(\lambda) = 0$$

where $\boldsymbol{P}(\lambda)$ is a matrix polynomial. In this article, we are interested in matrix polynomials in a particular factored form

$$\boldsymbol{P}(\lambda) = \lambda \boldsymbol{a}(\lambda)\boldsymbol{b}(\lambda) + \boldsymbol{c}\ ,$$

where $\boldsymbol{a}(\lambda), \boldsymbol{b}(\lambda) \in \mathbb{C}[\lambda]^{r\times r}$ with matrix coefficients of size $n \times n$ and $\boldsymbol{c} \in \mathbb{C}^{n\times n}$. To solve the PEP, we use a very common technique called linearization, namely the transformation of $\boldsymbol{P}$ to a linear matrix pencil; however, we use a specialized construction called *algebraic linearization*. We are interested in the backward stability of this algorithm. The algebraic linearization construction takes advantage of the factored form of the matrix polynomial by piecing together the linearization of $\boldsymbol{a}(\lambda)$ and $\boldsymbol{b}(\lambda)$ to form the linearization for $P(\lambda)$. Due to this recursive re-use, the linearizations are usually of lower height in comparison to the standard linearizations that arise on expanding $\boldsymbol{P}(\lambda)$ (in whatever polynomial basis) and thus, we hope that the algebraic linearization matrix is better conditioned (and hence the algorithm has a chance to be more numerically stable). Because of this, we believe that algebraic linearizations are more stable than other linearization constructions such as Frobenius linearizations. In this article, we explore whether our hypothesis is true by doing numerical experiments.

6.1.1 Algebraic linearizations

Algebraic linearization was first introduced in Theorem 5 of [1]. We restate the theorem below.

Theorem 6.1.1 *If $\boldsymbol{a}(z)$ and $\boldsymbol{b}(z)$ have the generalized standard triple representations $\boldsymbol{a}^{-1}(z) = \boldsymbol{X}_A(z\boldsymbol{D}_A - \boldsymbol{A})^{-1}\boldsymbol{Y}_A$ and $\boldsymbol{b}^{-1}(z) = \boldsymbol{X}_B(z\boldsymbol{D}_B - \boldsymbol{B})^{-1}\boldsymbol{Y}_B$, then the pencil $z\boldsymbol{D}_H - \boldsymbol{H}$ is a linearization*

of $\boldsymbol{h}(z) = z\boldsymbol{a}(z)\boldsymbol{b}(z) + \boldsymbol{c}_0$*, where the matrices* $\boldsymbol{D}_H$ *and* $\boldsymbol{H}$ *are given as follows:*

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & -\boldsymbol{Y}_A \boldsymbol{c}_0 \boldsymbol{X}_B \\ -\boldsymbol{X}_A & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{Y}_B & \boldsymbol{B} \end{bmatrix}$$

and

$$\boldsymbol{D}_H = \begin{bmatrix} \boldsymbol{D}_A & & \\ & \boldsymbol{I} & \\ & & \boldsymbol{D}_B \end{bmatrix}.$$

Proof See the proof from Theorem 5 from [1]. ♮

6.1.2 Backward stability of PEP

The literature of backward stability of linearizations is extremely large. Some of this literature includes Lawrence and Corless [8], Lawrence et al. [9], Higham et al. [6], [5], and Dopico et al. [3].

The normwise backward error of a finite approximate right eigenpair (λ, x) of the polynomial $\boldsymbol{P}(\lambda)$ expressed in the monomial basis is defined by

$$\eta_{\boldsymbol{P}}(x, \lambda) = \min \{\varepsilon : (\boldsymbol{P}(\lambda) + \Delta\boldsymbol{P}(\lambda))\, x = 0,\ \|\Delta A_i\|_2 \leq \varepsilon \|A_i\|_2, i = 0, \ldots, m\}\ . \tag{6.1}$$

Tisseur [10, Theorem 1] obtained the explicit formula

$$\eta_{\boldsymbol{P}}(x, \lambda) = \frac{\|\boldsymbol{P}(\lambda)x\|_2}{\left(\sum_{i=0}^{m} |\lambda^i| \|A_i\|_2\right) \|x\|_2}\ .$$

For the left eigenpair (y^*, λ), the normwise backward error is

$$\begin{aligned} \eta_{\boldsymbol{P}}(y^*, \lambda) &= \min \{\varepsilon : y^* (\boldsymbol{P}(\lambda) + \Delta\boldsymbol{P}(\lambda)) = 0, \|\Delta A_i\|_2 \leq \varepsilon \|A_i\|_2, i = 0, \ldots, m\} \\ &= \frac{\|y^* \boldsymbol{P}(\lambda)\|_2}{\left(\sum_{i=0}^{m} |\lambda^i| \|A_i\|_2\right) \|y\|_2}\ . \end{aligned}$$

Corless, Rezvani, and Amiraslani [2] have also extended the normwise backward error for the Lagrange basis, based on Amiraslani's 2006 PhD thesis. The same result was derived independently by Green and Wagenknecht in the same year [4]. For $\boldsymbol{P}(\lambda) = \sum_{i=0}^{m} A_i \ell_i(\lambda)$, where $\ell_i(\lambda)$ is the Lagrange basis, the normwise backward error is

$$\begin{aligned} \eta_{\boldsymbol{P}}(x, \lambda) &= \frac{\|\boldsymbol{P}(\lambda)x\|_2}{\left(\sum_{i=0}^{m} \|A_i\|_2 |\ell_i(\lambda)|\right) \|x\|_2} \\ \eta_{\boldsymbol{P}}(y^*, \lambda) &= \frac{\|y^* \boldsymbol{P}(\lambda)\|_2}{\left(\sum_{i=0}^{m} \|A_i\|_2 |\ell_i(\lambda)|\right) \|y\|_2}\ . \end{aligned}$$

The generalized normwise backward error for any polynomial basis[1] $\phi_i(\lambda)$ is

$$\begin{aligned} \eta_{\boldsymbol{P}}(x, \lambda) &= \frac{\|\boldsymbol{P}(\lambda)x\|_2}{\left(\sum_{i=0}^{m} \|A_i\|_2 |\phi_i(\lambda)|\right) \|x\|_2} \\ \eta_{\boldsymbol{P}}(y^*, \lambda) &= \frac{\|y^* \boldsymbol{P}(\lambda)\|_2}{\left(\sum_{i=0}^{m} \|A_i\|_2 |\phi_i(\lambda)|\right) \|y\|_2}\ . \end{aligned}$$

[1] Green and Wagenknecht noted that this theorem also holds for nonpolynomial $\phi(\lambda)$, such as $\exp(-\lambda)$, which occur for instance in delay differential equations.

Unfortunately, this type of backward error formula cannot work for $\boldsymbol{h}(\lambda) = \lambda \boldsymbol{a}(\lambda)\boldsymbol{b}(\lambda) + \boldsymbol{c}_0$ because $\boldsymbol{a}(z)\boldsymbol{b}(z)$ is not a linear combination of the basis elements. The standard theory requires that the matrix polynomial be expressed as a linear combination of the basis elements. At this time, we do not have a satisfactory theoretical framework for this. We can allow changes in $\boldsymbol{a}$ while keeping $\boldsymbol{b}$ fixed, and vice-versa; or we can do a linearized analysis, getting terms such as $\boldsymbol{a}\Delta\boldsymbol{b}$ and $\Delta\boldsymbol{a}\boldsymbol{b}$, but these are only valid in the limit of small $\Delta\boldsymbol{a}$ and $\Delta\boldsymbol{b}$.

On the other hand, we can use the established normwise backward error for algebraic linearization. Let $\boldsymbol{L}(\lambda) = \lambda\boldsymbol{X} - \boldsymbol{Y}$ be a linearization of $\boldsymbol{P}(\lambda)$. The backward errors of approximate left and right eigenpairs (u^*, λ) and (λ, v) are given by

$$\eta_{\boldsymbol{L}}(u^*, \lambda) = \frac{\|u^*\boldsymbol{L}(\lambda)\|_2}{(|\lambda|\|\boldsymbol{X}\|_2 + \|\boldsymbol{Y}\|_2)\,\|u\|_2}$$
$$\eta_{\boldsymbol{L}}(v, \lambda) = \frac{\|\boldsymbol{L}(\lambda)v\|_2}{(|\lambda|\|\boldsymbol{X}\|_2 + \|\boldsymbol{Y}\|_2)\,\|v\|_2}\,.$$

6.1.3 Pseudospectra of matrix polynomials

In the following section, we present numerical experiments to determine the backward stability of algebraic linearizations comparatively with other known linearizations, such as Frobenius linearizations and linearizations for other polynomial basis. Our method to compare the backward stability between different expressions of a matrix polynomial is to use ε-pseudospectra. Pseudospectra of matrix polynomials has been studied extensively: there are many references such as Tisseur and Higham [11] which looks at structured pseudospectra for polynomial eigenvalue problems, Lancaster and Psarrakos [7] which gives the general properties for pseudospectra of matrix polynomials, and Corless et al. [2] which looks at pseudospectra of matrix polynomials expressed in alternative polynomial bases. Tisseur and Higham [11] define the ε-pseudospectra for matrix polynomials as

$$\Lambda_\varepsilon(\boldsymbol{P}) = \{\lambda \in \mathbb{C} :\ (\boldsymbol{P}(\lambda) + \Delta\boldsymbol{P}(\lambda))\,x = 0 \text{ for some } x \neq 0 \text{ and } \Delta\boldsymbol{P}(\lambda) \text{ with } \|\Delta A_k\| \leq \varepsilon\alpha_k, k = 0, \ldots, m\}\ , \quad (6.2)$$

where

$$\Delta\boldsymbol{P}(\lambda) = \lambda^m\Delta A_m + \lambda^{m-1}\Delta A_{m-1} + \cdots + \Delta A_0\,, \quad (6.3)$$

and α_k are non negative weights that allow freedom on how the perturbations are measured.

6.1.4 Connection between pseudospectra and backward error

The pseudospectrum of a matrix has connections both with backward error and with the conditioning of its eigenvalues. The definition of the normwise backward error for an approximate eigenpair (x, λ) of $\boldsymbol{P}(\lambda)x = 0$ given in Tisseur and Higham [11] is shown in equation (6.1), but the norms $\|A_i\|_2$ are replaced by weights α_k

$$\eta(x, \lambda) := \min\{\varepsilon : (\boldsymbol{P}(\lambda) + \Delta\boldsymbol{P}(\lambda))\,x = 0, \|\Delta A_k\| \leq \varepsilon\alpha_k,\ k = 0, \ldots, m\}$$

and the backward error for an approximate eigenvalue λ is given by

$$\eta(\lambda) := \min_{k\neq 0} \eta(x, \lambda)\,.$$

Comparing equations (6.3) and (6.2), we can see that the ε-pseudospectrum can be expressed terms of backward error of λ as

$$\Lambda_\varepsilon(\boldsymbol{P}) = \{\lambda \in \mathbb{C} : \eta(\lambda) \leq \varepsilon\} \ .$$

Another way of thinking about the connection between pseudospectra and backward stability could be through the relationship between between forward and backward error:

$$\|\Delta y\| \precsim \kappa \cdot \|\Delta x\| \,, \tag{6.4}$$

where Δy and Δx are forward error and backward error, respectively, and κ is the condition number. By using the contours of the pseudospectra, we are able to compute the condition number and determine whether solving our problem using algebraic linearization is sensitive to change. An eigenvalue problem is considered well-conditioned if the condition number is relatively small, which means that the problem is insensitive to perturbations.

In figure 6.1, we show the pseudospectra contours around an eigenvalue from matrix polynomial from equation (6.5). The eigenvalue that we are looking at its sensitivity is $x = 2.23$. Then, picking a point ($x = 2.4535 + 0.03803i$) on the contour, we can estimate the condition number[2] of the PEP. The distance between the point and the eigenvalue is the forward error. In our example, the forward error is approximately 0.2267. The value of the contour, on the other hand is our backward error. In our example, the backward error is $x = 0.0499$. Using the formula from equation 6.4,

$$\kappa = \frac{\|\Delta y\|}{\|\Delta x\|} = \frac{0.2267}{0.0499} = 4.5424 \,.$$

6.1.5 Algorithms as transformations

The final and perhaps most important connection between pseudospectra, condition numbers, and numerical stability is in the notion of an algorithm being a transformation of one problem $\mathcal{P}$ to another problem $\mathcal{P}'$ which is easier to solve. Examples include transforming a pair of general matrices $(\boldsymbol{A}, \boldsymbol{B})$ to a pair of upper triangular matrices $(\boldsymbol{S}, \boldsymbol{T})$ (say, by unitary congruence) so that the generalized eigenvalues are the same, but can simply be read off from the latter pair.

The basic principle is that an algorithm that transforms a well-conditioned problem $\mathcal{P}$ to an *ill-conditioned* problem $\mathcal{P}'$ cannot, without a miracle, be a numerically stable algorithm. Thus, examining the condition number of the transformed problem $\mathcal{P}'$ gives us an indication of the potential numerical stability of the overall algorithm.

In what follows, we will use pseudospectra to examine the conditioning of the algebraic linearization (our transformed problem $\mathcal{P}'$). We will see that our algebraic linearization will usually be well-conditioned: this suggests—but does not prove—that the algorithm for solving the PEP that consists of computing and then solving an algebraic linearization has the potential to be numerically stable.

[2]The condition number of a simple eigenvalue is an estimate of how spread-apart the near-circular contours are, very close to the eigenvalue.

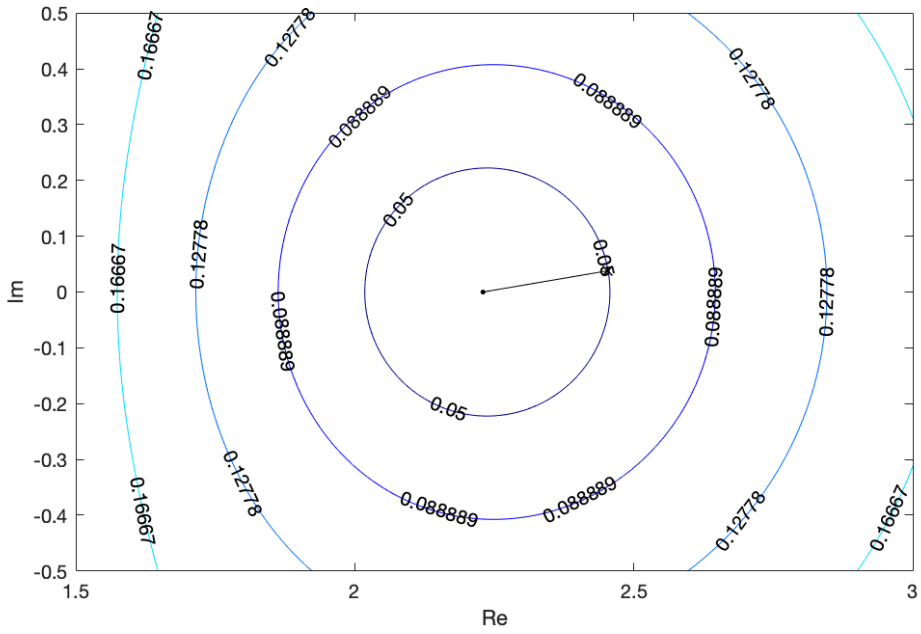


Figure 6.1: Pseudospectra of (6.5) to demonstrate the connection between pseudospectra and backward stability. The length of the line connecting the eigenvalue ($\lambda = 2.23$) and the contour ($\epsilon = 0.05$) is the forward error.

6.2 Numerical experiments

In this section, we perform some numerical experiments to gain an understanding of the backward stability of algebraic linearizations by using pseudospectra. We give a variety of numerical experiments, including using different polynomial bases, such as monomial basis, Bernstein basis and Lagrange basis, and changing the size of the problem. In the following, the pseudospectra is plotted on the complex plane. Subsection 6.2.1 shows an example of the pseudospectra of the matrix polynomials themselves in their original bases, whereas the rest are examples of the pseudospectra of the linearization.

6.2.1 Pseudospectrum of a matrix polynomial

Let

$$\boldsymbol{P}(\lambda) = \lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{I}_3 \tag{6.5}$$

where

$$\boldsymbol{A}(\lambda) = \begin{bmatrix} 0.4615 & 0.0923 & 0.8391 \\ 0.1533 & 0.4590 & 0.5133 \\ 0.9339 & 0.4703 & 0.3931 \end{bmatrix} + \begin{bmatrix} 0.0665 & 0.3310 & 0.0346 \\ 0.0251 & 0.7981 & 0.3070 \\ 0.0844 & 0.8489 & 0.2307 \end{bmatrix} \lambda + \begin{bmatrix} 0.1961 & 0.6483 & 0.6803 \\ 0.8701 & 0.4791 & 0.4333 \\ 0.8227 & 0.0951 & 0.2917 \end{bmatrix} \lambda^2$$

and

$$\boldsymbol{B}(\lambda) = \begin{bmatrix} 0.1018 & 0.8888 & 0.7178 \\ 0.4186 & 0.6231 & 0.6527 \\ 0.6902 & 0.0650 & 0.5154 \end{bmatrix} + \begin{bmatrix} 0.4264 & 0.1536 & 0.7465 \\ 0.8109 & 0.5541 & 0.5062 \\ 0.7240 & 0.3980 & 0.3853 \end{bmatrix} \lambda + \begin{bmatrix} 0.2322 & 0.4092 & 0.8949 \\ 0.8038 & 0.0554 & 0.5337 \\ 0.3854 & 0.4454 & 0.6937 \end{bmatrix} \lambda^2.$$

To save space, we rounded the printed coefficients to four places.

Figure 6.2 shows the pseudospectra of two different linearizations of the matrix polynomial in the form $\lambda \boldsymbol{a}(\lambda)\boldsymbol{b}(\lambda) + \boldsymbol{c}_0$ given in equation (6.5). On the left, we use algebraic linearization and on the right, we use Frobenius linearization. Since we do not know how to compute the condition number for a matrix polynomial expressed in the form for algebraic linearization, we used the results from subsection 6.1.4 as our condition number for the algebraic linearization problem. We can observe that the two figures look quite similar at a quick glance. Looking at the figure more carefully, some eigenvalues, especially those near the origin, for the matrix polynomial expressed in the monomial basis (right) are less sensitive to perturbation. Despite this, it seems that the matrix polynomial expressed in the algebraic linearization form looks less susceptible to changes.

We can also compute the condition number for an eigenvalue for the expanded matrix polynomial expressed in the monomial basis. For the results to be more comparable, we use the same eigenvalue as previously $x = 2.23$. Evaluating

$$\sum_{k=0}^{5} \|\boldsymbol{M}_k\|_2 \cdot |2.23|^k \doteq 321.3134\,,$$

where $\boldsymbol{M}_k$ would be the coefficients of the expanded matrix polynomial. If we normalize the matrix polynomial by dividing all the coefficients with the largest norm of the coefficients, our condition number is then $\kappa = 51.7018$. This quantitatively suggests that the matrix polynomial expressed in the factor form is more stable.

6.2.2 Monomial basis

In this subsection, we have three matrix polynomials which are expressed in the monomial basis.

Monic case

Let us use the same matrix polynomial as the example in subsection 6.2.1. In figure 6.3, we show that we calculated the pseudospectra in two ways: using the grid technique to create the contours (coloured) and perturbing the coefficients on the matrix polynomials and finding the solution to these perturbed equations (in black). For the algebraic linearization case, we perturb the coefficients of $\boldsymbol{a}(\lambda)$ and $\boldsymbol{b}(\lambda)$ and $\boldsymbol{c}_0$ by. On the other hand, for the Frobenius linearization, after expanding the matrix polynomial, we then perturb the coefficients of the matrix polynomial now expressed differently. We then take the eigenvalues of these perturbed polynomial matrices—these are what is plotted in figure 6.3. For these experiments, we performed the perturbations for the algebraic linearization problem 1000 times and the Frobenius linearization 10000 times.

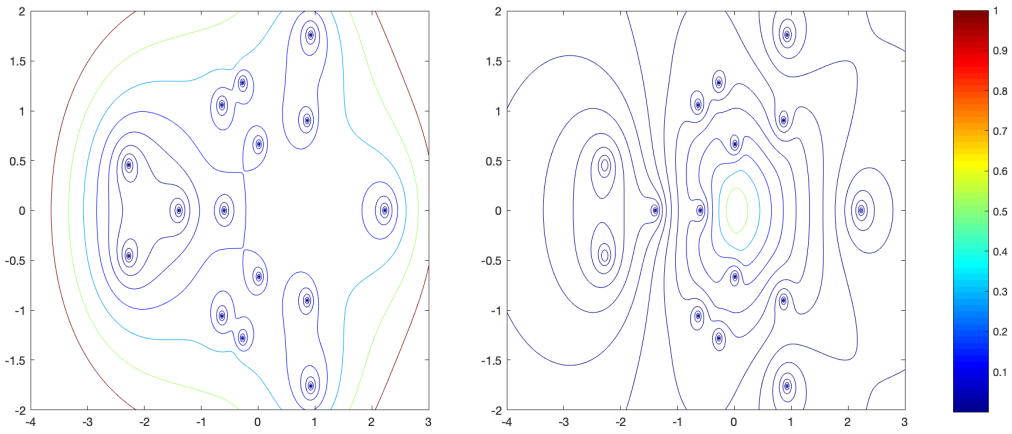


Figure 6.2: Pseudospectra of two different linearizations (left: algebraic linearization, right: frobenius linearization) of the matrix polynomial in the form $\lambda \boldsymbol{a}(\lambda)\boldsymbol{b}(\lambda) + \boldsymbol{c}_0$ given in equation (6.5).

The left figure is of the pseudospectra of the linearization of the equation constructed using algebraic linearization whereas the right figure uses Frobenius linearization. We observe that the left figure is clearly more stable according to both methods. For the grid technique, we can see that the contours, for example $\varepsilon = 0.05$, are closer to the roots in the left figure compared to the right figure. As for the perturbation technique, we also used $\varepsilon = 0.05$. We can see that distribution of the eigenvalues vary more for the Frobenius linearization, indicating that even very small perturbations result in large changes in the result. This demonstrates that for this example, the algebraic linearization is more stable in comparison to the Frobenius linearization.

Singular leading coefficient case

Let

$$\boldsymbol{P}(\lambda) = \lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c} \tag{6.6}$$

where

$$\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} 17 & 78 \\ 1 & 77 \end{bmatrix} + \begin{bmatrix} 43 & 59 \\ 6 & 18 \end{bmatrix} \lambda + \begin{bmatrix} 73 & 26 \\ 54 & 92 \end{bmatrix} \lambda^2 + \begin{bmatrix} 76 & 7 \\ 89 & 19 \end{bmatrix} \lambda^3, \\
\boldsymbol{B} &= \begin{bmatrix} 74 & 78 \\ 70 & 51 \end{bmatrix} + \begin{bmatrix} 43 & 86 \\ 62 & 68 \end{bmatrix} \lambda + \begin{bmatrix} 53 & 71 \\ 30 & 39 \end{bmatrix} \lambda^2 + \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \lambda^3 \\
\boldsymbol{c} &= \begin{bmatrix} 94 & 1 \\ 82 & 1 \end{bmatrix}.
\end{aligned}$$

Since the leading coefficient of $\boldsymbol{B}$ is singular, this means that when it is multiplied with the leading coefficient of $\boldsymbol{A}$, the result is singular, and therefore the leading coefficient of equation (6.6) is singular.

Figure 6.4 shows the pseudospectra of two different linearization construction of (6.6). The left figure uses the algebraic linearization construction whereas the right figure uses the

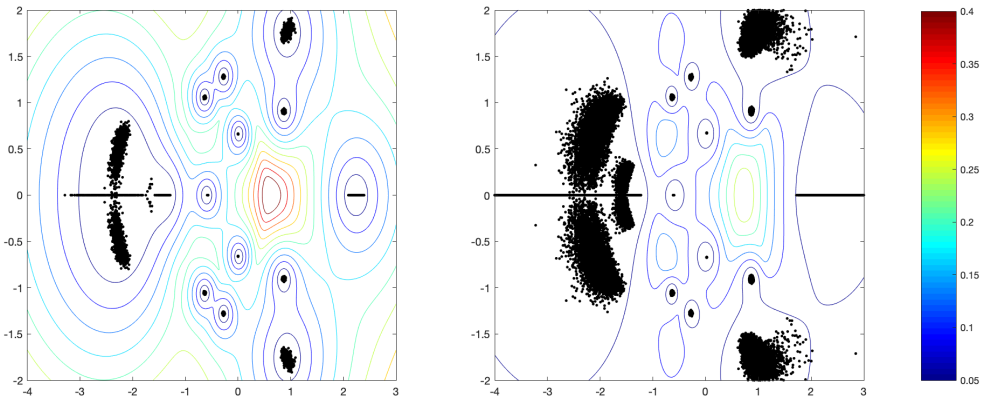


Figure 6.3: Pseudospectra of two different kinds of linearizations for equation (6.5) which is expressed in the monomial basis. The linearization constructions used are algebraic linearization (left) and Frobenius linearization (right).

Frobenius linearization. From the figure, it is clear that the algebraic linearization for this example has better stability. Although the contour $\varepsilon = 10^{-4}$ is pictorially smaller in the right figure (Frobenius linearization) compared to the left figure (algebraic linearization), we still consider that the algebraic linearization is more stable because the other contours that can be seen in these figures are comparatively much larger in the right figure compared to the left.

"Large" example

Let

$$\boldsymbol{P}(\lambda) = \lambda\left(\lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c}_0\right)\left(\lambda \boldsymbol{C}(\lambda)\boldsymbol{D}(\lambda) + \boldsymbol{c}_1\right) + \boldsymbol{c}_2 \tag{6.7}$$

where

$$\boldsymbol{A}(\lambda) = \begin{bmatrix} 0.5674 & 0.8245 \\ 0.9688 & 0.9596 \end{bmatrix} + \begin{bmatrix} 0.6463 & 0.4766 \\ 0.3796 & 0.9119 \end{bmatrix}\lambda + \begin{bmatrix} 0.0149 & 0.4716 \\ 0.1567 & 0.5430 \end{bmatrix}\lambda^2 + \begin{bmatrix} 0.0597 & 0.8896 \\ 0.6580 & 0.1096 \end{bmatrix}\lambda^3,$$

$$\boldsymbol{B}(\lambda) = \begin{bmatrix} 0.4378 & 0.9852 \\ 0.2802 & 0.6088 \end{bmatrix} + \begin{bmatrix} 0.2537 & 0.5450 \\ 0.1326 & 0.8278 \end{bmatrix}\lambda + \begin{bmatrix} 0.8370 & 0.2037 \\ 0.8333 & 0.5444 \end{bmatrix}\lambda^2 + \begin{bmatrix} 0.8749 & 0.8564 \\ 0.1210 & 0.8998 \end{bmatrix}\lambda^3,$$

$$\boldsymbol{C}(\lambda) = \begin{bmatrix} 0.2179 & 0.4742 \\ 0.0770 & 0.8350 \end{bmatrix} + \begin{bmatrix} 0.4694 & 0.5027 \\ 0.4138 & 0.1254 \end{bmatrix}\lambda + \begin{bmatrix} 0.1323 & 0.6030 \\ 0.8705 & 0.2653 \end{bmatrix}\lambda^2 + \begin{bmatrix} 0.8648 & 0.4578 \\ 0.0581 & 0.7222 \end{bmatrix}\lambda^3,$$

$$\boldsymbol{D}(\lambda) = \begin{bmatrix} 0.3390 & 0.5270 \\ 0.4012 & 0.8942 \end{bmatrix} + \begin{bmatrix} 0.7784 & 0.2788 \\ 0.0694 & 0.3794 \end{bmatrix}\lambda + \begin{bmatrix} 0.8647 & 0.2399 \\ 0.4200 & 0.5977 \end{bmatrix}\lambda^2 + \begin{bmatrix} 0.4794 & 0.9347 \\ 0.8985 & 0.8179 \end{bmatrix}\lambda^3$$

and

$$\boldsymbol{c}_0 = \begin{bmatrix} 0.7089 & 0.8997 \\ 0.7432 & 0.0652 \end{bmatrix}, \quad \boldsymbol{c}_1 = \begin{bmatrix} 0.3359 & 0.8281 \\ 0.0043 & 0.5074 \end{bmatrix}, \quad \boldsymbol{c}_2 = \begin{bmatrix} 0.3662 & 0.5348 \\ 0.2266 & 0.2895 \end{bmatrix}.$$

To save space, we have rounded the printed coefficients to four decimal places. Figure 6.5 shows pseudospectra of the eigenvalue problem being solved with different linearizations. The

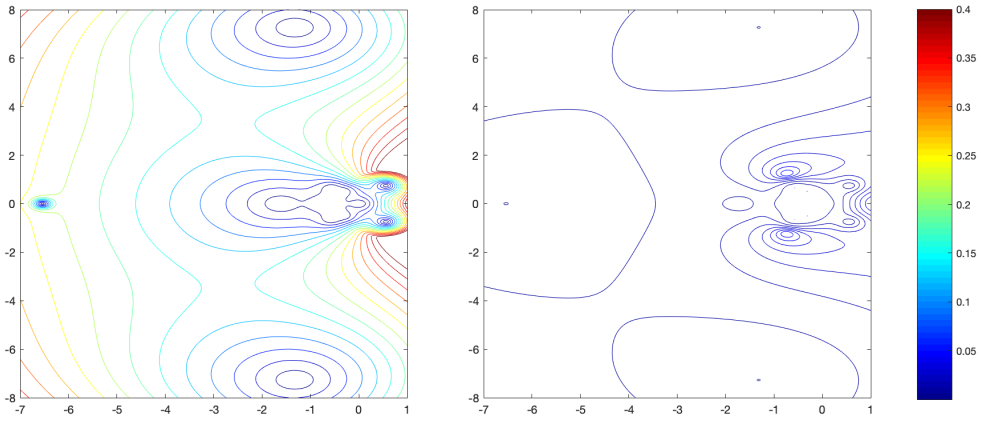


Figure 6.4: Pseudospectra of two different kinds of linearizations for equation (6.6) which is expressed in the monomial basis. The linearization constructions used are algebraic linearization (left) and Frobenius linearization (right).

linearization of the figure on the left was constructed using algebraic linearization recursively. Since $\lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c}_0$ and $\lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c}_1$ are in the appropriate form that is required for the algebraic linearization construction, we can construct the companion for these two terms using algebraic linearizations. Then, using the companions that we built, we can build the linearization for equation (6.7) using algebraic linearization again. The second way (the figure is in the center) we express the equation is by expressing $\lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c}_0$ and $\lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c}_1$ in an expanded form rather than in a factor form. By expressing these terms in the monomial basis, we then can construct linearizations for each term using the Frobenius linearization construction. Then, using the resulting linearizations, we can build the linearization for equation (6.7) using the algebraic linearization construction. Lastly, we can expand the entire matrix polynomial express it in the monomial basis. The figure for this is in the right.

From figure 6.5, we can see that the pseudospectra figure is the most stable, which is to be expected. Between the pseudospectra figures, the recursive algebraic linearization can be seen to be the most stable among the three since the contours in the algebraic linearization figure are the closest to the roots. The most unstable linearization for this example is the Frobenius linearization.

6.2.3 Bernstein basis

Let

$$\boldsymbol{P}(\lambda) = \lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c} \tag{6.8}$$

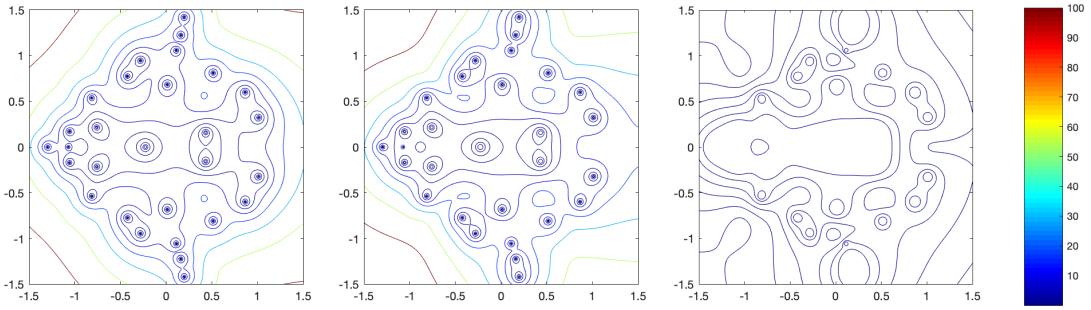


Figure 6.5: The left figure uses the form in (6.7) and uses algebraic linearizations to construct the companion. The center figure expresses the inner terms $\lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda)+\boldsymbol{c}_0$ and $\lambda \boldsymbol{C}(\lambda)\boldsymbol{D}(\lambda)+\boldsymbol{c}_1$ in the monomial basis. This is so that we use the Frobenius linearization construction for the parts expressed in the monomial basis and then use algebraic linearization to construct the final companion. Lastly, we use the expanded form of the equation and created the Frobenius linearization (right).

where matrix polynomials $\boldsymbol{A}(\lambda)$ and $\boldsymbol{B}(\lambda)$ are expressed in the Bernstein basis, $B_k^n(\lambda)$, in which the subscript k is the index and the superscript n is the degree,

$$\boldsymbol{A}(\lambda) = \begin{bmatrix} 73 & 59 \\ 61 & 44 \end{bmatrix} B_0^2(\lambda) + \begin{bmatrix} 25 & 2 \\ 43 & 61 \end{bmatrix} B_1^2(\lambda) + \begin{bmatrix} 96 & 4 \\ 10 & 89 \end{bmatrix} B_2^2(\lambda)\,,$$
$$\boldsymbol{B}(\lambda) = \begin{bmatrix} 25 & 82 \\ 1 & 15 \end{bmatrix} B_0^2(\lambda) + \begin{bmatrix} 88 & 36 \\ 10 & 60 \end{bmatrix} B_1^2(\lambda) + \begin{bmatrix} 59 & 65 \\ 67 & 44 \end{bmatrix} B_2^2(\lambda)$$

and

$$\boldsymbol{c} = \begin{bmatrix} 14 & 25 \\ 76 & 66 \end{bmatrix}.$$

Figure 6.6 shows the pseudospectra of equation (6.8) for various constructions for the linearization. From left to right, we have the algebraic linearization, linearization for matrix polynomials expressed in the Bernstein basis, and the Frobenius linearization. Here, we see that the algebraic linearization and the Frobenius linearization are quite similar (the algebraic linearization result may be only slightly better). For the Bernstein linearization, we can see that the eigenvalue $0.8639 + 0.0000i$ is very well-conditioned, whereas all the other eigenvalues are not as well-conditioned. Overall, it is difficult to determine which linearization construction offers the best stability from these results.

6.2.4 Mixed bases

Let

$$\boldsymbol{P}(\lambda) = \lambda \boldsymbol{A}(\lambda)\boldsymbol{B}(\lambda) + \boldsymbol{c} \tag{6.9}$$

where $\boldsymbol{A}(\lambda)$ is expressed in the monomial basis

$$\boldsymbol{A}(\lambda) = \begin{bmatrix} 44 & 20 \\ 18 & 62 \end{bmatrix} + \begin{bmatrix} 27 & 95 \\ 56 & 72 \end{bmatrix} \lambda + \begin{bmatrix} 68 & 78 \\ 96 & 61 \end{bmatrix} \lambda^2$$

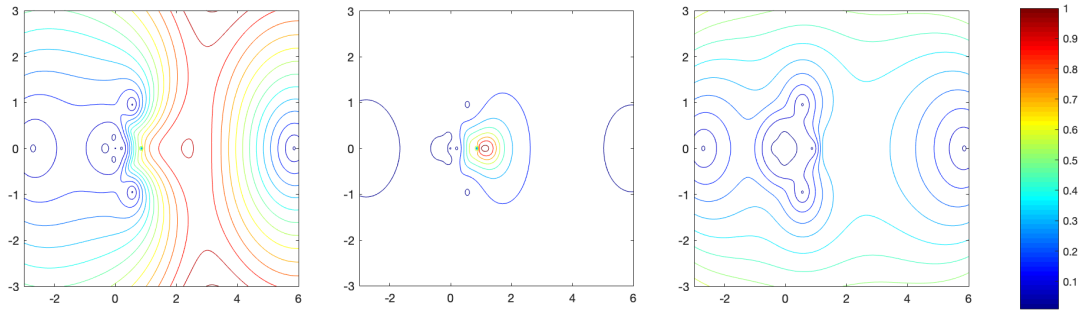


Figure 6.6: Pseudospectra of two different kinds of linearizations for equation – which is expressed in the Bernstein basis. The linearization constructions used are algebraic linearization (left), linearization for matrix polynomials expressed in the Bernstein basis (center), Frobenius linearization (right).

and $\boldsymbol{B}(\lambda)$ is expressed in the Lagrange interpolational basis with the interpolating nodes

$$\tau = \left[-1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 1\right]$$

and

$$\boldsymbol{B} = \left[\begin{bmatrix} 95 & 27 \\ 6 & 99 \end{bmatrix}, \begin{bmatrix} 78 & 69 \\ 48 & 42 \end{bmatrix}, \begin{bmatrix} 39 & 39 \\ 22 & 3 \end{bmatrix}, \begin{bmatrix} 48 & 98 \\ 34 & 56 \end{bmatrix}, \begin{bmatrix} 85 & 47 \\ 41 & 83 \end{bmatrix}\right],$$

respectively.

Figure 6.7 shows the pseudospectra of equation (6.9) for the algebraic linearization construction (left) and the Frobenius linearization construction (right). The two figures look very similar—the most obvious difference would be the color of the contours, which helps us determine which of the two linearizations are more stable. Since the largest contour ($\varepsilon = 1$) is only visible in the algebraic linearization figure, this suggests that it is more stable than the Frobenius linearization.

6.3 Concluding Remarks

In this article, we have presented some numerical experiments to show that there are many potential cases where algebraic linearizations are more backward stable. As future work, we hope to establish bounds and find an explicit formula for the normwise backward error for polynomial eigenvalue problems that are solved using algebraic linearizations.

Acknowledgments

We acknowledge the support of Western University, The National Science and Engineering Research Council of Canada, the University of Alcalá, the Ontario Research Centre of Computer

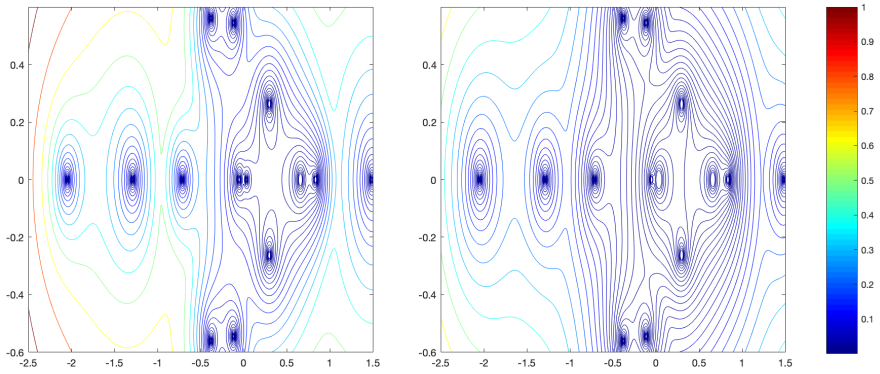


Figure 6.7: Pseudospectra for equation (6.9) which is expressed in various ways. The linearization constructions used are algebraic linearization (left) and the Frobenius linearization for matrix polynomials expressed in the monomial basis (right).

Algebra, and the Rotman Institute of Philosophy. Part of this work was developed while RMC and EYSC were visiting the University of Alcalá, in the frame of the project Giner de los Rios and Mitacs Globalink Research Award, respectively.

Bibliography

[1] E. Y. S. Chan, R. M. Corless, L. Gonzalez-Vega, J. R. Sendra, and J. Sendra, *Algebraic linearizations for matrix polynomials*, Linear Algebra and its Applications, 563 (2019), pp. 373–399.

[2] R. M. Corless, N. Rezvani, and A. Amiraslani, *Pseudospectra of matrix polynomials that are expressed in alternative bases*, Mathematics in Computer Science, 1 (2007), pp. 353–374.

[3] F. Dopico, J. Pérez, and P. Van Dooren, *Structured backward error analysis of linearized structured polynomial eigenvalue problems*, Mathematics of Computation, 88 (2019), pp. 1189–1228.

[4] K. Green and T. Wagenknecht, *Pseudospectra and delay differential equations*, Journal of computational and applied mathematics, 196 (2006), pp. 567–578.

[5] N. J. Higham, R.-C. Li, and F. Tisseur, *Backward error of polynomial eigenproblems solved by linearization*, SIAM journal on matrix analysis and applications, 29 (2007), pp. 1218–1241.

[6] N. J. Higham, D. S. Mackey, and F. Tisseur, *The conditioning of linearizations of matrix polynomials*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 1005–1028.

[7] P. LANCASTER AND P. PSARRAKOS, *On the pseudospectra of matrix polynomials*, SIAM journal on matrix analysis and applications, 27 (2005), pp. 115–129.

[8] P. W. LAWRENCE AND R. M. CORLESS, *Backward error of polynomial eigenvalue problems solved by linearization of lagrange interpolants*, SIAM Journal on Matrix Analysis and Applications, 36 (2015), pp. 1425–1442.

[9] P. W. LAWRENCE, M. VAN BAREL, AND P. VAN DOOREN, *Backward error analysis of polynomial eigenvalue problems solved by linearization*, SIAM journal on Matrix Analysis and Applications, 37 (2016), pp. 123–144.

[10] F. TISSEUR, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra and its Applications, 309 (2000), pp. 339–361.

[11] F. TISSEUR AND N. J. HIGHAM, *Structured pseudospectra for polynomial eigenvalue problems, with applications*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 187–208.

Chapter 7

Concluding remarks

This thesis has described the historical development of the idea of the recursive algebraic linearization of matrix polynomials expressible in the form $z\mathbf{a}(z)\mathbf{b}(z) + \mathbf{c}$, beginning from Piers Lawrence's idea for the Mandelbrot polynomials. We first generalized to Fibonacci-Mandelbrot and Narayana-Mandelbrot, and later to Euclid polynomials, at which time we realized that the idea must be general. At this point, the ideal of a "minimal height" companion became clearer. We then extended the idea to matrix polynomials, and studied the numerical stability, which is good, according to our experiment. Future work includes generalizing the minimal height idea to matrix polynomial linearizations.

Perhaps the most interesting future work will be to combine the Masters' Thesis [1], which used homotopy methods and singularity detection, in an attempt to create an efficient, reliable "divide and conquer" method to compute general eigenvalues using a homotopy. We will first formally specify this divide-and-conquer algorithm, prove it correct, and perform a cost analysis based on the cost of numerical solution of ODE and the approximate location of singularities, which we need to avoid. We will then test the algorithm again using millions of Bohemians, again looking for worst-cases (likely to be cases in which the eigenvalues of different blocks are close, or are identical, which leads to potential singularities) and experimentally determining the average case behaviour.

Bibliography

[1] E. Y. S. Chan, *A comparison of solution methods for Mandelbrot-like polynomials*, Master's thesis, The University of Western Ontario, 2016.

Eunice Y. S. Chan

Education

University of Western Ontario **2016–Present**
Ph.D. Applied Mathematics

University of Western Ontario **2015–2016**
M.Sc. Applied Mathematics

University of Western Ontario **2011–2015**
B.Sc. Applied Mathematics

Academic Awards

Rotman Travel Award	2019
Mitacs Globalink Research Award	2019
Ontario Graduate Scholarship	2017–2018
Distinguished Poster Award (ISSAC)	2016
Best Poster Award Winner (ACMES 2)	2016

Publications

Chan, E. Y. S., Corless, R. M., Gonzalez-Vega, L., Sendra, J. R., and Sendra, J. (2019), "Algebraic Linearizations of Matrix Polynomials." *Linear Algebra and its Applications*, Volume 563, pp. 373-399. DOI: https://doi.org/10.1016/j.laa.2018.10.028

Chan, E. Y. S., Corless, R. M. (2018), "Minimal height companion matrices for Euclid polynomials." *Mathematics in Computer Science*. DOI: https://doi.org/10.1007/s11786-018-0364-2

Chan, E. Y. S., and Corless, R. M. (2017), "A new kind of companion matrix." *Electronic Journal of Linear Algebra*, Volume 32, pp. 335-342. DOI: https://doi.org/10.13001/1081-3810.3400

Accepted Publications

Chan, E. Y. S., Corless, R. M. (2018), "A random walk through experimental mathematics." *Special Issue of the Journal of Australian Mathematical Society dedicated to Jon Borwein*, (in press).

Submitted Publications

Chan, E. Y. S.. Corless, R. M., Rafiee Sevyeri, L. (2019), "Generalized Standard Triples for Algebraic Linearizations of Matrix Polynomials." Submitted to *Electronic Journal of Linear Algebra*. Manuscript number: 4053.

Chan, E. Y. S., Corless, R. M., Gonzalez-Vega, L., Sendra, J. R., Sendra, J., and Thornton, S. E. (2018), "Upper Hessenberg and Toeplitz Bohemians." Submitted to *Linear Algebra and its Applications*. Manuscript number: LAA-D-18-01084.

Conference Presentations

"Investigation of Backward Stability of Algebraic Linearizations," presented at International Congress on Industrial and Applied Mathematics (ICIAM) (2019), University of Valencia, Valencia, Spain.

"Random Walk through Experimental Mathematics," presented at Effective Visualization in the Mathematical Sciences 3 (2018), University of Newcastle, Newcastle, Australia (through correspondence).

"Algebraic Linearizations of Matrix Polynomials," presented at Bohemian Matrices and Applications Workshop (2018), University of Manchester, Manchester, UK.

"Algebraic Linearizations of Matrix Polynomials," presented at Canadian Applied and Industrial Mathematics Society (CAIMS) (2018), Ryerson University, Toronto, Canada.

"Chaos Game Representation," presented at the Undergraduate Society of Mathematics (USAM) Conference (2018), University of Western Ontario, London, Canada (invited).

"A new kind of companion matrix," presented at Applied Mathematics, Modeling and Computational Science Conference (AMMCS) (2017), Wilfrid Laurier University, Waterloo, Canada.

"Mandelbrot polynomials and matrices," presented at USAM Conference (2017), University of Western Ontario, London, Canada (invited).

"Narayana, Mandelbrot, and a new kind of companion matrix," presented at Milestones in Computer Algebra (MICA) (2016), University of Waterloo, Waterloo, Canada.

"The Fibonacci-Mandelbrot polynomials and matrices," presented at Southern Ontario Numerical Analysis Day (SONAD) (2016), University of Waterloo, Canada.

Posters

"Investigation of Backward Stability of Algebraic Linearizations," presented at Advances in Numerical Linear Algebra (2019), University of Manchester, Manchester, UK.

"Fibonacci-Mandelbrot polynomials and matrices," presented at Computational Discovery in Mathematics: ACMES 2 (2016), University of Western Ontario, London, Canada.

"Fibonacci-Mandelbrot polynomials and matrices," presented at International Symposium on Symbolic and Algebraic Computation (ISSAC) (2016), Wilfrid Laurier University, Waterloo, Canada.

Teaching Experiences	**University of Western Ontario**	2015–2019
	Teaching Assistant, Applied Mathematics	
	Calculus with Analysis for Statistics, Applied Mathematics for Engineering II, Linear Algebra with Numerical Analysis for Engineering, Applied Mathematics for Engineers I, Open Problems in Experimental Mathematics, Intermediate/Advanced Calculus II, Introduction to Object Oriented Scientific Programming, Numerical Analysis.	
Teaching Awards	SOGS GSTA Award of Excellence	2019
	SOGS GSTA Award of Excellence	2018
	Faculty of Science Graduate Student Teaching Award	2017