# On the stability of computing polynomial roots via confederate linearizations

Nakatsukasa, Yuji and Noferini, Vanni

2014

MIMS EPrint: **2014.49**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

# On the stability of computing polynomial roots via confederate linearizations

Yuji Nakatsukasa*and Vanni Noferini†

September 25, 2014

A common way of computing the roots of a polynomial is to find the eigenvalues of a linearization, such as the companion (when the polynomial is expressed in the monomial basis), colleague (Chebyshev basis) or comrade matrix (general orthogonal polynomial basis). For the monomial case, many studies exist on the stability of linearization-based rootfinding algorithms. By contrast, little seems to be known for other polynomial bases. This paper studies the stability of algorithms that compute the roots via linearization in nonmonomial bases, and has three goals. First we prove normwise stability when the polynomial is properly scaled and the QZ algorithm (as opposed to the more commonly used QR algorithm) is applied to a comrade pencil associated with a Jacobi orthogonal polynomial. Second, we extend a result by Arnold that leads to a first-order expansion of the backward error when the eigenvalues are computed via QR, which shows that the method can be unstable. Finally, we focus on the special case of Chebyshev basis, in particular the Chebfun rootfinder: we discuss its stability and describe an optional functionality, made available for improved stability, for computing the roots of a general continuous function $f(x)$, implemented in the recently updated version 5. The main message is that to guarantee backward stability QZ applied to a properly scaled pencil is necessary.

**Keywords:** polynomial, roots, Chebyshev polynomial, companion matrix, colleague matrix, comrade matrix, QZ, QR, recurrence relation, perturbation, orthogonal polynomial

**MSC classification:** 65H04, 65F15, 65G50

*Department of Mathematical Informatics, University of Tokyo, Tokyo 113-8656, Japan (`vanni.noferini@manchester.ac.uk`). Supported by JSPS Scientific Research Grant No. 26870149.
†School of Mathematics, University of Manchester, Manchester, England, M13 9PL (`vanni.noferini@manchester.ac.uk`). Supported by ERC Advanced Grant MATFUN (267526).

1

# 1. Introduction

Let $p(x) \in \mathbb{R}[x]_n$ be a nonzero polynomial of degree at most $n$ with real coefficients. The rootfinding quest for the set of the solutions of the equation $p(x) = 0$ can rightly be deemed one of the eldest mathematical problems that mankind has considered [15, 41]. Since the classical algebraic results by Abel, Galois and Ruffini in the 18th and 19th centuries, it has been known that, for high degree polynomials ($n \geq 5$), the search for a general algebraic method that gives the exact roots is hopeless. Hence, it is unsurprising that devising reliable numerical methods for polynomial rootfinding is a central theme in numerical analysis.

A related problem is that of finding the roots, or some roots, of a general nonlinear function. Indeed, it is not uncommon to reduce the problem to the polynomial case by approximation: for instance, a standard way to compute the real roots of a smooth function $f(x)$ on an interval, as done in Chebfun [48], is to approximate $f(x)$ by a polynomial $p(x)$ via Chebyshev interpolation, then compute the roots of $p(x) = \sum_{i=0}^{n} c_i T_i(x)$ expressed in the Chebyshev basis $\{T_i(x)\}$ by computing the eigenvalues of the linearized colleague matrix [25], [47, Ch. 18]. This process is known to work well in practice, but no analysis has been carried out to prove its numerical stability.

More generally, one practical way of finding the roots of a polynomial is to first construct either a matrix or a matrix pencil whose eigenvalues coincide, with the same algebraic and geometric multiplicities, with the roots of $p(x)$. This process is known as a *linearization* of $p(x)$. The next step is of course approximating the eigenvalues numerically: usual choices are the QR algorithm, for matrices, or the QZ algorithm, for matrix pencils. In this paper we refer to these algorithms simply as QR and QZ.

It should be noted that linearization is by no means the only option. Indeed, many alternative ideas exist: the Durand–Kerner [31], the Ehrlich–Aberth [11], or the Jenkins–Traub algorithms [29], and Weyl's method [42] to name but a few.

Some of these alternative methods are strong competitors of the linearization method, both for computational complexity and for stability (see, e.g., [7, 10, 11] for the Ehrlich–Aberth method); on the other hand, special technologies, like subdivision [14] or preservation of structures such as quasiseparable [5, 8, 9, 12, 13], can be exploited in order to reduce the complexity of the linearization method to $\mathcal{O}(n^2)$. Among the many rootfinding algorithms available, our goal here is to understand the stability of linearization-based methods, as they are often easy to implement (given a black-box eigensolver) and widely used. For instance, the MATLAB function `roots` follows precisely the above described procedure via the eigenvalues of the companion matrix.

Since the second step of computing the eigenvalues is numerical, it needs to be investigated whether the roots are computed stably. Specifically, the two standard backward stable eigensolvers, QR for a matrix $C$ or QZ for a matrix pencil $\lambda X + Y$ [24, Ch. 7,8], are known to be backward stable with respect to the matrix norms, i.e., they compute[1] the exact eigenvalues of slightly perturbed matrices $C + \Delta C$ and $\lambda(X + \Delta X) +$

---

[1]Strictly speaking, we should add: if they converge. For the nonsymmetric case, no formal proof of convergence of either QR or QZ is known to the authors. In practice, and possibly relying on randomized shifts when dealing with counterexamples cleverly conceived to embarrass one specific

$(Y + \Delta Y)$ for $\|\Delta C\| \le \varepsilon \|C\|, \|\Delta X\| \le \varepsilon \|X\|, \|\Delta Y\| \le \varepsilon \|Y\|$, where $\varepsilon = \widehat{q}(n)u$ for a fixed unit roundoff $u$ and some low-degree polynomial with moderate coefficients $\widehat{q}$, whose exact form would depend on the number of iterations before convergence and on the choice of norms in the bound. However, stability in the matrix norm does not necessarily imply stability in the polynomial. Specifically, let $\hat{x}_i$ be the computed roots. Writing

$$p(x) = \sum_{i=0}^{n} c_i \phi_i(x), \qquad \widehat{p}(x) = \alpha \prod_{i=1}^{n} (x - \hat{x}_i) = \sum_{i=0}^{n} \widehat{c}_i \phi_i(x) \tag{1.1}$$

for some scalar $\alpha \ne 0$, and defining $c = [c_0, c_1, \ldots, c_n], \widehat{c} = [\widehat{c}_0, \widehat{c}_1, \ldots, \widehat{c}_n]$ and $\Delta c = c - \widehat{c}$, we say that the algorithm performed in a backward stable manner with respect to the polynomial $p$ if the difference in the coefficients is within $\mathcal{O}(\varepsilon)$:

$$\left( \frac{\|\Delta c\|_2}{\|c\|_2} = \right) \frac{\|c - \widehat{c}\|_2}{\|c\|_2} = \mathcal{O}(\varepsilon). \tag{1.2}$$

Note that the norm depends on the basis $\{\phi_i(x)\}$ and on the scaling, i.e., $p \leftarrow \alpha p$ for a nonzero scalar $\alpha$. In practice we set $\alpha$ to $\alpha = \frac{c^T \widehat{c}}{\|\widehat{c}\|_2^2}$, which is the minimizer of $\|c - \alpha \widehat{c}\|_2$. Throughout we always denote by $\hat{x}_i$ the computed roots of $p$, and by $\widehat{p}$ the polynomial with exact roots $\hat{x}_i$ and $\widehat{c}$ its coefficients.

In this paper we are primarily interested in the normwise backward stability of the computation of the roots. That is, our goal is to give bounds for the right-hand side of (1.2). For example, the stability proof in [49] for the use of QZ to compute the roots of $p(x)$, expressed in the monomial basis, falls into this category, and we extend their result to other orthogonal polynomial bases in Section 3.

We note that some authors have discussed the more stringent componentwise backward stability, which for example takes the form $\max_i \frac{|\Delta c_i|}{|c_i|}$ [21], [33], (also [46] for matrix polynomials). However, even in the monomial case no known bound appears to guarantee componentwise backward stability. For example, the analysis in [21] for monic polynomials, i.e., $c_n = 1$ expressed in the monomials gives

$$\Delta c_{i-1} = \sum_{m=0}^{i-1} c_m \sum_{j=i+1}^{n} E_{j,j+m-i} - \sum_{m=i}^{n} c_m \sum_{j=1}^{i} E_{j,j+m-i}, \tag{1.3}$$

where $E$ is the backward error in $C$ by QR so that the computed eigenvalues are the exact eigenvalues of $C + E$. While (1.3) gives the exact backward error in $c_i$ to first order, it is generally difficult to derive individual bounds for each $i$. Often, componentwise bounds (see for instance [33] for QZ in monomials) are not much more informative than the much simpler normwise bound (obtained as a corollary of (1.3))

$$\frac{\|\Delta c\|_2}{\|c\|_2} = \mathcal{O}(\varepsilon)\|C\|_2. \tag{1.4}$$

Here the dependence on $n$ is hidden in $\mathcal{O}(\varepsilon)$, that is, $\mathcal{O}(\varepsilon)$ is a constant that has size $q(n)u$ where $q$ is a modest low-degree polynomial.

---

implementation, they do converge with no known exception.

We note that (1.4) suggests the crucial instability of QR: the backward error is proportional to $\|C\|$, meaning the computation is unstable if $\|c\|_2 \gg 1$. Note that this issue cannot be resolved by a scaling $p \leftarrow \alpha p$. A related discussion is given in [33], which also examines the effect of diagonal balancing for QR and QZ, see Section 4.4.

The authors feel a linearization-based rootfinder generally cannot achieve componentwise stability unless a special structure is present in $p(x)$ or the basis; indeed we argue further in the appendix that componentwise backward stability is not achievable by any method, at least for a generic choice of the polynomial basis and the machine number system (the monomials are a notable exception). Given the discussion above, we focus on the normwise stability.

It is worth mentioning another possible definition of stability, which concerns the individual stability for each computed root. For example, each root computed by the Ehrlich–Aberth method is known to have a small componentwise backward error [11] in the monomial case. Specifically, each computed root $\hat{x}_i$ satisfies[2] $p(\hat{x}_i) + \Delta p_i(\hat{x}_i) = 0$ with $\Delta p_i(x) = \sum_{j=0}^{n} \Delta C_j^{(i)} \phi_j^{(i)}(x)$ and $\frac{|\Delta c_j^{(i)}|}{|c_j|} = \mathcal{O}(\varepsilon)$; compare this with (1.2), and note that $\Delta p_i$ here depends on $i$. The strongest backward stability would be componentwise for the whole set of computed roots: $\frac{|\Delta c_i|}{|c_i|} = \mathcal{O}(\varepsilon)$ for $p(x) + \Delta p(x) = \kappa \prod_{i=1}^{n}(x - \hat{x}_i)$, but at present we are unaware of any polynomial rootfinder that guarantees this.

When $p(x)$ is expressed in the standard monomial basis, many studies exist on the stability, or lack thereof, of the linearization-based rootfinders [12, 20, 21, 49]. However, little seems to be known for $p(x)$ expressed in a nonmonomial basis such as Chebyshev and other orthogonal polynomials. Such bases are becoming increasingly important for numerical purposes [22, 38, 47], as particularly exemplified by the prominence of Chebyshev polynomials in the Chebfun system, in which `roots` is an important command called within various polynomial operations such as computing the maxima and the $L_1$ norm, or invoking `abs`.

In this work we focus on the normwise stability of linearization-based rootfinding methods for polynomials expressed in certain nonmonomial basis. We will review some basic notions concerning a commonly used class of linearizations in non-monomial bases in Section 2.

The paper has three main themes:

1. In Section 3, we show that if $p(x)$ is expressed in certain orthogonal polynomial bases and the eigenvalues of a linearization of $p(x)$, known as the comrade pencil, are computed by QZ after scaling the polynomial to have coeffcients $\mathcal{O}(1)$, the process is normwise backward stable. Note that in Chebfun QR, and not QZ, is used by default for a polynomial scaled to be monic in the Chebyshev basis, and this may in some circumstances lead to a colleague matrix with a large norm $\|C\|$, which is undesirable in view of (1.4). See also [12, 30] for a discussion for the monomial basis yielding the similar conclusion that QZ is preferred to QR when the leading coefficient is small.

---

[2]The weaker notion of normwise stability for each root is $(p + \Delta p_i)(\hat{x}_i) = 0$ with $\frac{|\Delta c_i|}{\|c\|_2} = \mathcal{O}(\varepsilon)$.

2. In Section 4, we discuss QR applied to a monic polynomial, i.e., the leading coefficient in the considered basis is 1, and in particular we show how some results in [21] can be extended to other degree-graded bases, including any orthogonal polynomial basis. The result reconfirms the advantage of QZ over QR, at least when balancing is not used. In practice, the technique of diagonal balancing often improves the stability of QR significantly, as we discuss in Section 4.4. However, as the evidence provided in Section 6 illustrates, even with balancing QR-based rootfinding can be normwise unstable, whereas we prove QZ-based rootfinding is stable with a simple initial normalization. Since QZ is empirically about 3 times more expensive than QR, the choice should be made based on the trade-off between guaranteed stability (QZ) and speed (QR).

3. In Section 5 we focus on the Chebyshev basis and on the problem of finding the real roots of a (possibly non-polynomial) function $f(x)$ on an interval as done in Chebfun, showing that the polynomial approximation preserves the normwise stability of the computed roots and arguing that the subdivision technique can improve the accuracy if the original function $f(x)$ is resampled. Based on our findings we suggest replacing the currently used QR with QZ, or at least providing QZ as a "safer" option when stability is more important than speed. Indeed, examples exist for which QR misses real roots that cannot be moved off the real axis by a small backward perturbation.

## 2. Preliminaries on confederate linearizations

Clearly, there are uncountably many linearizations of a polynomial $p(x)$. For example, if $C \in \mathbb{R}^{n \times n}$ is a linearization and $X \in GL(\mathbb{R}, n)$, then $XCX^{-1}$ is also a linearization. Yet, it is natural to focus only on those that can be easily constructed from its coefficient in a given basis. Even with this restriction, the number of possibilities in the literature is huge, particularly in the monomial basis. We will restrict ourselves to the case where $\{\phi_i\}$ is a degree-graded basis, that is, $\deg \phi_i = i$, and to a certain class of linearizations, that, following the nomenclature in [6], we call confederate linearizations. In the following we assume that we can express $p(x) = \sum_{i=0}^{n} c_i \phi_i(x)$. We will visualize a few confederate linearizations by depicting their version for $n = 4$.

First, we assume that $p(x)$ is expressed in the monomials and that it is monic in such a basis, i.e., $p_n = 1$ (which is no loss of generality, modulo a global multiplicative scaling). The archetype of all linearizations is the *companion matrix* of $p(x)$:

$$C = \begin{bmatrix} -c_3 & -c_2 & -c_1 & -c_0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.1}$$

In the literature, there is not a fixed convention on how to define the companion matrix. Many variants are found, for instance: the transpose of (2.1), the matrix obtained by

flipping both rows and columns of (2.1), and the transpose of the latter. Moreover, while some authors call (2.1) and its variants just "companion" matrices, others call them "Frobenius companion" matrices, thereby granting the status of "companion" to other linearizations as well, e.g., Fiedler matrices. All this is, of course, only a matter of convention: we clarify once and for all that, throughout our theoretical analysis in Sections 2, 3 and 4, we shall have no companion matrices other than (2.1). We nonetheless note that the four mathematically equivalent forms can exhibit nontrivial numerical differences, see Section 6.

The deep algebraic meaning of the companion matrix is that it is nothing but a representation, in the monomial basis, of the multiplication-by-$x$ operator in the quotient ring $\mathbb{R}[x]_n/\langle p(x)\rangle$ where $\langle p(x)\rangle$ is the ideal generated by $p(x)$. In other words, as it is immediate to verify,

$$
C \begin{bmatrix} x^{n-1} \\ x^{n-2} \\ \vdots \\ x \\ 1 \end{bmatrix} = \begin{bmatrix} x^n - p(x) \\ x^{n-1} \\ \vdots \\ x^2 \\ x \end{bmatrix} \equiv \begin{bmatrix} x^n \\ x^{n-1} \\ \vdots \\ x^2 \\ x \end{bmatrix}^T \quad \bmod p(x).
$$

For more details on this algebraic viewpoint, the reason it yields a linearization, and some generalizations, see, e.g., [6, Ch. 5], [17, Sec. 10.4], and [40, Sec. 9].

Clearly, this concept is easily generalizable to any other polynomial basis [6]: we now recall how. In particular, let $\{\phi_i\}$ be a degree-graded basis with $\phi_0 = 1$ and denote by $\kappa$ the ratio between the leading coefficients of $\phi_n$ and $\phi_{n-1}$, when they are expressed in the monomial basis. We now assume that $p(x)$ is monic in the basis $\{\phi_i\}$, i.e., $p(x) = \phi_n(x) + \sum_{i=0}^{n-1} c_i \phi_i(x)$. We consider the (unique) matrix $C_\phi$ satisfying

$$
C_\phi \begin{bmatrix} \phi_{n-1}(x) \\ \phi_{n-2}(x) \\ \vdots \\ \phi_1(x) \\ \phi_0 \end{bmatrix} = \begin{bmatrix} x\phi_{n-1}(x) - \kappa^{-1}p(x) \\ x\phi_{n-2}(x) \\ \vdots \\ x\phi_1(x) \\ x\phi_0 \end{bmatrix}.
$$

We call $C_\phi$ the *confederate matrix* of $p(x)$ in the basis $\{\phi_i\}$. Now we introduce some notation. Let $B$ be the change of basis matrix such that $\begin{bmatrix} \phi_{n-1}(x) & \phi_{n-2}(x) & \dots & \phi_1(x) & \phi_0 \end{bmatrix}^T = B \begin{bmatrix} x^{n-1} & x^{n-2} & \dots & x & 1 \end{bmatrix}^T$. In particular, since $\{\phi_i\}$ is degree-graded, $B$ is upper triangular. Moreover, let $\mathcal{FR}_n \subseteq \mathbb{R}^{n\times n}$ be the vector subspace of "first row matrices", which we define as those matrices whose rows are all zero except (possibly) the first. We state some properties of $C_\phi$. Their proof is omitted as it is not difficult, and can be found in various sources, e.g., [6, Thm. 5.3].

**Theorem 2.1** (Properties of confederate matrices)**.** *Let $p(x) = \phi_n(x) + \sum_{i=0}^{n-1} c_i \phi_i(x)$ and let $C_\phi$ be the confederate matrix of $p(x)$ in the degree-graded basis $\{\phi_i\}$, and let $B$ be*

*the change of basis matrix between the monomials and* $\{\phi_i\}$, *defined as above. Then, the following properties hold:*

- $C_\phi = BCB^{-1}$;

- *if* $p(\mu) = 0$ *then* $\mu$ *is an eigenvalue of* $C_\phi$ *of geometric multiplicity 1, and the corresponding eigenvector* $v$ *has Vandermonde structure* $v = \begin{bmatrix} \phi_{n-1}(\mu) & \dots & \phi_0(\mu) \end{bmatrix}^T$;

- $C_\phi = H_\phi + F_\phi(p)$, *where* $H_\phi$ *is upper Hessenberg and depends only on* $\{\phi_i\}$, *but not on* $p(x)$, *while* $F_\phi(p) \in \mathcal{FR}_n$ *and its first row is* $\kappa^{-1} \begin{bmatrix} -c_{n-1} & \dots & -c_1 & -c_0 \end{bmatrix}$, *where* $\kappa$ *is the ratio of the leading coefficients of* $\phi_n$ *and* $\phi_{n-1}$, *when expressed in the monomial basis.*

Note in the first item that if $C_\phi = BCB^{-1}$ is a confederate matrix for $p(x)$, then $C$ is the companion matrix for $\alpha p(x)$, where $\alpha \neq 0$ is some constant. This scaling is needed because, in general, a polynomial that is monic when represented in one degree-graded basis needs not be monic when represented in another one.

If $\{\phi_i\}$ are orthogonal polynomials, the corresponding confederate matrices are called *comrade* matrices [6], and have the additional property that $H_\phi$ is tridiagonal [6]. Among orthogonal polynomials, the Chebyshev polynomials of the first kind, traditionally denoted by $\{T_i\}$, have great importance in practical applications. The comrade matrix for the Chebyshev basis is known as the *colleague matrix* and we denote it by $C_T$. In its decomposition $C_T = H_T + F_T(p)$ as in Theorem 2.1, it holds $\kappa = 2$ and

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

So far, we have assumed that the polynomial $p(x)$ is *monic* in the basis $\{\phi_i\}$, i.e., in its expansion on the basis of choice its leading coefficient $c_n = 1$. Although, as argued above, this is no loss of generality, there are some circumstances where $c_n \neq 1$ and one might find it more convenient not to scale all the coefficients by $c_n$. However, in this case the corresponding linearizations will not be confederate matrices, but *confederate pencils*. For $n = 4$ the confederate pencil of $p(x) = \sum_{i=0}^{4} c_i \phi_i(x)$ is

$$
\begin{bmatrix} c_4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (xI_4 - C_\phi),
$$

where $C_\phi$ is the confederate matrix of $p(x)/c_4$. Analogously to the monic case, we will use the expressions, resp., comrade pencil, colleague pencil and companion pencil to refer to the confederate pencil in, resp., an orthogonal polynomials basis, the first Chebyshev basis and the monomial basis.

**Example 2.2.** *We illustrate the previous definitions with a concrete example. Let* $p(x) = x^4 + x^3 + x^2 + x + 1$. *Then its companion matrix is*

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

*and, since* $p(x)$ *is monic in the monomial basis, its companion pencil is* $xI_4 - C$. *An easy computation shows that in the Chebyshev basis* $p(x) = \frac{1}{8}T_4(x) + \frac{1}{4}T_3(x) + T_2(x) + \frac{7}{4}T_1(x) + \frac{15}{8}T_0(x)$. *Although* $p(x)$ *is not monic in this basis, we may scale it appropriately, and the colleague matrix associated with the monic (in the Chebyshev basis) polynomial* $8p(x)$ *is*

$$C_T = \begin{bmatrix} -1 & -7/2 & -7 & -15/2 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

*whereas the colleague pencil associated with* $p(x)$ *is*

$$x\begin{bmatrix} 1/8 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} -1/8 & -7/16 & -7/8 & -15/16 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

*Similarly, if* $\{\phi_i\}$ *is the Legendre basis then expanding* $\frac{35}{8}p(x) = \phi_4(x) + \frac{7}{4}\phi_3(x) + \frac{65}{12}\phi_2(x) + 7\phi_1(x) + \frac{161}{24}\phi_0(x)$ *we obtain the comrade matrix*

$$C_\phi = \begin{bmatrix} -1 & -8/3 & -4 & -23/6 \\ 3/5 & 0 & 2/5 & 0 \\ 0 & 2/3 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

*and so on.*

We conclude this introductory section with a few comments. In principle, one could consider any polynomial basis, not necessarily degree-graded. In other words, $B$ needs not be triangular. Borrowing the terminology once again from [6], one could use the name *congenial matrices* (or pencils) for this further generalization of confederate matrices (or pencils). Note that any matrix similar to the companion (that is, any linearization consisting of a matrix rather than a pencil) is a congenial matrix in *some* polynomial basis. Although some bases of practical interest, e.g., Newton, Lagrange, or Bernstein, are not degree-graded, we argue that in practice, if the QR algorithm is then used, there is not much to gain in analyzing congenial matrices that are not confederate. Indeed, generally a congenial matrix will not be upper Hessenberg, and the first task that QR performs is to reduce the matrix to Hessenberg form. One can regard this process as implicitly performing a change of basis towards a degree-graded one.

# 3. Stability of rootfinding via the QZ algorithm applied to the colleague and a certain class of comrade pencils

In the concluding remark of [49], Van Dooren and Dewilde show that QZ applied to the companion pencil for computing the roots of a scalar polynomial $p(x) = \sum c_i x^i$ such that $\max_i |c_i| = \mathcal{O}(1)$ yields a normwise backward stable rootfinder. In this section, we extend this result to the Chebyshev basis, corresponding to the colleague pencil, and to a certain class of orthogonal polynomials, corresponding to comrade pencils based on Jacobi polynomials with parameters $|\alpha|, |\beta| \leq \frac{1}{2}$.

This is the main technical result of the paper and the argument is not immediate. We first show that the backward error caused in QZ can be compressed so that the computed roots can be written as the exact roots of a polynomial with slightly perturbed coefficients in a slightly perturbed basis. We then prove that orthogonal polynomials on $[-1, 1]$ defined by a three-term recurrence have roots that are not sensitive to perturbation in the recurrence relation. We use this to conclude that the computed roots are the exact roots of a polynomial with slightly perturbed coefficients in the original basis.

For definiteness and simplicity we first derive the results for the Chebyshev basis. Later we will argue that essentially the same proof carries over more generally to Jacobi polynomials $P_n^{(\alpha,\beta)}$ [45, Ch. 4], i.e. polynomials orthogonal with respect to the weight function $(1-x)^\alpha (1+x)^\beta$ on $[-1, 1]$, in which we impose the parameters $|\alpha|, |\beta| \leq \frac{1}{2}$. Chebyshev is a special case in which $\alpha = \beta = -\frac{1}{2}$, and other important special cases include Legendre with $\alpha = \beta = 0$, Chebyshev of the second kind with $\alpha = \beta = \frac{1}{2}$, and ultraspherical polynomials with $\alpha = \beta$ (up to normalizing each $\phi_i(x)$ by constants). Extending the results to polynomials orthogonal on a general real interval $[a, b]$ is straightforward by an affine mapping.

## 3.1. Compressing backward error by QZ

Let $p(x)$ be a scalar polynomial expressed in the Chebyshev basis:

$$p(x) = \sum_{i=0}^{n} c_i T_i(x), \quad \|c\|_2 = 1, \tag{3.1}$$

where $p(x)$ is normalized so that the vector of coefficients $[c_1, \ldots, c_n]$ have norm 1, as assumed also in [49]; the essence of what follows remains valid for $\|c\|_2 = \mathcal{O}(1)$. Suppose that $p(x)$ is linearized with the colleague pencil, say, $\lambda X + Y$. Then, the QZ algorithm is applied to the latter. This ensures that the eigenvalues of the linearization $\lambda X + Y$ are computed in a backward stable manner, that is, they are the exact eigenvalues of some $\lambda \widetilde{X} + \widetilde{Y}$, which is a pencil of the form (for $n = 6$)

$$\lambda \begin{bmatrix} \tilde{c}_6 & \tilde{0} & \tilde{0} & \tilde{0} & \tilde{0} & \tilde{0} \\ \tilde{0} & \tilde{1} & \tilde{0} & \tilde{0} & \tilde{0} & \tilde{0} \\ \tilde{0} & \tilde{0} & \tilde{1} & \tilde{0} & \tilde{0} & \tilde{0} \\ \tilde{0} & \tilde{0} & \tilde{0} & \tilde{1} & \tilde{0} & \tilde{0} \\ \tilde{0} & \tilde{0} & \tilde{0} & \tilde{0} & \tilde{1} & \tilde{0} \\ \tilde{0} & \tilde{0} & \tilde{0} & \tilde{0} & \tilde{0} & \tilde{1} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} -\tilde{c}_5 & \tilde{c}_6 - \tilde{c}_4 & -\tilde{c}_3 & -\tilde{c}_2 & -\tilde{c}_1 & -\tilde{c}_0 \\ \tilde{1} & \tilde{0} & \tilde{1} & \tilde{0} & \tilde{0} & \tilde{0} \\ \tilde{0} & \tilde{1} & \tilde{0} & \tilde{1} & \tilde{0} & \tilde{0} \\ \tilde{0} & \tilde{0} & \tilde{1} & \tilde{0} & \tilde{1} & \tilde{0} \\ \tilde{0} & \tilde{0} & \tilde{0} & \tilde{1} & \tilde{0} & \tilde{1} \\ \tilde{0} & \tilde{0} & \tilde{0} & \tilde{0} & \tilde{2} & \tilde{0} \end{bmatrix}. \tag{3.2}$$

Here and below, we adopt the following notation: for any $a \in \mathbb{R}$, $\tilde{a}$ is a real number satisfying $|\tilde{a} - a| \le \varepsilon \le \hat{q}(n)u$ for some low degree polynomial $\hat{q}$, denoting by $u$ the unit roundoff. In other words, $\varepsilon$ represents the actual backward error of the outcome of QZ, measured in the max-norm, and $\hat{q}(n)u$ is some theoretical upper bound for $\varepsilon$. We now show that we can apply an equivalence transformation so that $(I + E)(\lambda \widetilde{X} + \widetilde{Y})(I + F)$ is a comrade pencil and, at the same time, a small perturbation of a colleague pencil. More specifically, it has the form

$$
\lambda \begin{bmatrix} \tilde{c}_6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} -\tilde{c}_5 & \tilde{c}_6 - \tilde{c}_4 & -\tilde{c}_3 & -\tilde{c}_2 & -\tilde{c}_1 & -\tilde{c}_0 \\ \tilde{1} & \tilde{0} & \tilde{1} & 0 & 0 & 0 \\ 0 & \tilde{1} & \tilde{0} & \tilde{1} & 0 & 0 \\ 0 & 0 & \tilde{1} & \tilde{0} & \tilde{1} & 0 \\ 0 & 0 & \tilde{0} & \tilde{1} & \tilde{0} & \tilde{1} \\ 0 & 0 & 0 & 0 & \tilde{2} & \tilde{0} \end{bmatrix}. \tag{3.3}
$$

**Lemma 3.1.** *Let $\lambda \widetilde{X} + \widetilde{Y} = \lambda(X + \Delta X) + (Y + \Delta Y)$ be a perturbed colleague pencil as in (3.2), with $|\Delta X_{ij}|, |\Delta Y_{ij}| \le \varepsilon$. Then there exist matrices $E, F$ with $\|E\|_2, \|F\|_2 = \mathcal{O}(n\varepsilon)$ such that the equivalent pencil $(I + E)(\lambda \widetilde{X} + \widetilde{Y})(I + F)$ has the form as in (3.3).*

*Proof.* The proof is constructive and algorithmic, in the same vein as [49]: we describe a procedure that generates a pencil of the form (3.3) by an equivalence transformation starting from one of the form (3.2). The matrices $E, F$ can be chosen so that $(I + E) = \prod_i (I + E_i)$ and $(I + F) = \prod_i (I + F_i)$, where $I + E_i$ (resp., $I + F_i$) represent elementary row (resp. column) operations. Observe that this immediately implies that $I + E$ and $I + F$ are nonsingular, so that the outcome $(I + E)(\lambda \widetilde{X} + \widetilde{Y})(I + F)$ is indeed equivalent to the pencil $\lambda \widetilde{X} + \widetilde{Y}$. To help the reader follow the algorithm, we first illustrate it graphically for $n = 6$: the first subscript denotes the order in which perturbed zeros are annihilated, whereas the second subscript indicates whether this is done via a row (r) or a column (c) operation.

$$
\lambda \begin{bmatrix} \tilde{c} & \tilde{0}_{10,r} & \tilde{0}_{10,r} & \tilde{0}_{10,r} & \tilde{0}_{10,r} & \tilde{0}_{10,r} \\ \tilde{0}_{1,c} & \tilde{1} & \tilde{0}_{11,r} & \tilde{0}_{11,r} & \tilde{0}_{11,r} & \tilde{0}_{11,r} \\ \tilde{0}_{1,c} & \tilde{0}_{3,c} & \tilde{1} & \tilde{0}_{13,r} & \tilde{0}_{13,r} & \tilde{0}_{13,r} \\ \tilde{0}_{1,c} & \tilde{0}_{3,c} & \tilde{0}_{5,c} & \tilde{1} & \tilde{0}_{15,r} & \tilde{0}_{15,r} \\ \tilde{0}_{1,c} & \tilde{0}_{3,c} & \tilde{0}_{5,c} & \tilde{0}_{7,c} & \tilde{1} & \tilde{0}_{17,r} \\ \tilde{0}_{1,c} & \tilde{0}_{3,c} & \tilde{0}_{5,c} & \tilde{0}_{7,c} & \tilde{0}_{9,c} & \tilde{1} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \tilde{c} & \tilde{c} & \tilde{c} & \tilde{c} & \tilde{c} & \tilde{c} \\ \tilde{1} & \tilde{0} & \tilde{1} & \tilde{0}_{12,c} & \tilde{0}_{12,c} & \tilde{0}_{12,c} \\ \tilde{0}_{2,r} & \tilde{1} & \tilde{0} & \tilde{1} & \tilde{0}_{14,c} & \tilde{0}_{14,c} \\ \tilde{0}_{2,r} & \tilde{0}_{4,r} & \tilde{1} & \tilde{0} & \tilde{1} & \tilde{0}_{16,c} \\ \tilde{0}_{2,r} & \tilde{0}_{4,r} & \tilde{0}_{6,r} & \tilde{1} & \tilde{0} & \tilde{1} \\ \tilde{0}_{2,r} & \tilde{0}_{4,r} & \tilde{0}_{6,r} & \tilde{0}_{8,r} & \tilde{2} & \tilde{0} \end{bmatrix}.
$$
$$\tag{3.4}$$

The process perturbs the coefficients terms $c$ and $1$, which we do not keep track of in (3.4), instead simply write $\tilde{c}$ and $\tilde{1}$.

Here the $\tilde{0}$ terms (without subscripts) in the second matrix do not get eliminated; they remain nonzero and $\mathcal{O}(\varepsilon)$ in absolute value.

The final step is to scale all but the first row in order to obtain $1$ (unperturbed), rather then $\tilde{1}$, in the diagonal elements in $\widetilde{X}$.

A more formal description of the algorithm for a generic $n$ is as follows:

1  For $j = 1 : n - 2$
2      For $i = (j + 1) : n$
3          Annihilate $\widetilde{X}_{ij}$ by adding a multiple of $i$th column to $j$th column.
4      end
5      For $i = (j + 2) : n$
6          Annihilate $\widetilde{Y}_{ij}$ by adding a multiple of $(j + 1)$th row to $i$th row.
7      end
8  end
9  Annihilate $\widetilde{X}_{n,n-1}$ by adding a multiple of $n$th column to $(n - 1)$th column.
10  For $j = 2 : n$
11      Annihilate $\widetilde{X}_{1j}$ by adding a multiple of $j$th row to 1st row.
12  end
13  For $i = 2 : n - 2$
14      For $j = (i + 1) : n$
15          Annihilate $\widetilde{X}_{ij}$ by adding a multiple of $j$th row to $i$th row.
16      end
17      For $j = (i + 2) : n$
18          Annihilate $\widetilde{Y}_{ij}$ by adding a multiple of $(i + 1)$th column to $j$th column.
19      end
20  end
21  Annihilate $\widetilde{X}_{n-1,n}$ by adding a mutliple of $n$th row to $(n - 1)$th row.
22  For $i = 2 : n$
23      Set $\widetilde{X}_{ii} = 1$ by scaling the $i$th row.
24  end

Now we represent each single operation in the pseudocode above either as $I + E_i$ or as $I + F_i$, according to whether it is a row or column operation. Observe the two key features of the process that (i) once a zero element is created, it is never subsequently perturbed again and (ii) to annihilate an element we always add a small multiple of $\tilde{1}$, ensuring that $E_i$ and $F_i$ are indeed small (this might not be the case if we needed to add a multiple of some $\tilde{c}$). This guarantees that eventually the form (3.3) can be obtained. Each $E_i, F_i$ has only one nonzero element, of absolute value bounded by $\varepsilon$. Moreover, examining lines 6, 11, 15 and 21 of the pseudocode we see that the nonzero elements always are in different positions in each $E_i$, and the same is true for each $F_i$.

Hence, $E = \sum_i E_i + \mathcal{O}(\varepsilon^2)$ and $F = \sum_i F_i + \mathcal{O}(\varepsilon^2)$ are both bounded, to first oder, by $\varepsilon$ in the max-norm. But in turn this implies [27, Ch. 5] that for the spectral norm we have $\|E\|_2, \|F\|_2 \leq n\varepsilon + \mathcal{O}(\varepsilon^2)$. □

**Remark 3.2.** *Although, for definiteness, we have chosen to give the statement of Lemma 3.1 in the spectral norm, similar results are as easily obtainable for other choices of norms. For example, a simple modification of the proof shows that $\|E\|, \|F\| \leq n^{1.5}\varepsilon + \mathcal{O}(\varepsilon^2)$ where $\| \cdot \|$ is any unitarily invariant norm, e.g., the Frobenius norm, or the nuclear norm [27, Sec. 5.6].*

As a consequence of Lemma 3.1, we see that QZ gives the exact roots of

$$\widehat{p}(x) = \sum_{i=0}^{n} \widetilde{c}_i \widetilde{T}_i(x), \tag{3.5}$$

which, with respect to the original $p(x)$ in (3.1), is a polynomial with slightly perturbed coefficients in a slightly perturbed basis $\widetilde{T}_i(x)$. Concerning the coefficients $\widetilde{c}_i$ in (3.5), by the assumptions on the outcome of QZ and by Lemma 3.1 one can check that they satisfy

$$\max_i |\widetilde{c}_i - c_i| = \mathcal{O}(q_1(n)u), \tag{3.6}$$

where $q_1$ is some slowly growing function $\leq \alpha n^{\frac{1}{2}+\tau}$ where $\tau = \deg \widehat{q}(n)$, the error by QZ, and $\alpha$ is a moderate constant. Note the factor $n^{\frac{1}{2}}$ instead of $\|E\|_2, \|F\|_2 \approx n$, which we obtain since only one row or column of $E$ and $F$ affect each $\widetilde{c}_i$. On the other hand, the perturbed Chebyshev polynomials $\widetilde{T}_i(x)$ satisfy the perturbed recurrence relation

$$2x\widetilde{T}_i(x) = (1 + \epsilon_{i,i+1})\widetilde{T}_{i+1}(x) + \epsilon_{i,i}\widetilde{T}_i(x) + (1 + \epsilon_{i,i-1})\widetilde{T}_{i-1}(x), \tag{3.7}$$

which is a slight perturbation of the original Chebyshev recurrence relation

$$2xT_i(x) = T_{i+1}(x) + T_{i-1}(x). \tag{3.8}$$

In (3.7), the $\epsilon_{i,j}$ terms are modest multiples of the unit roundoff $u$, satisfying $\max |\epsilon_{i,j}| \leq q_2(n)u$. Here again $q_2(n) \leq \alpha n^{\frac{1}{2}+\tau}$. To simplify notation, we also introduce a third slowly growing function $q(n) = \max(q_1(n), q_2(n))$ and define (note that we distinguish $\varepsilon$ and $\epsilon$)

$$\epsilon = |q(n)u|. \tag{3.9}$$

Observe that by definition we have $\max_i |\widetilde{c}_i - c_i| \leq \epsilon$ and $|\epsilon_{i,j}| < \epsilon$. The exact exponent of $n$ in $q(n)$ might in principle be obtained by using strict error bounds for QZ and keeping track of each step in the proof of Lemma 3.1. Such a bound might be obtainable by applying with some care the results of [26, Ch. 19], and would clearly also depend on the number of iterations[3]. However, here we will not go into such level of detail. We just assume in the following that, for a given unit roundoff $u$, $n$ is moderate so that $\epsilon = |q(n)u| \ll n^{-2}$.

The question therefore is whether the roots of $\widehat{p}(x)$ in (3.5) can be regarded as backward stably computed roots of $p(x)$. We prove this in the affirmative:

**Theorem 3.3.** *Let $\widetilde{T}_i(x)$ be perturbed Chebyshev polynomials defined by (3.7), and let $\widehat{p}(x)$ be as in (3.5) with $\max_i |c_i - \widetilde{c}_i| \leq \epsilon$. Suppose moreover that $\epsilon \ll n^{-2}$. Then we can write*

$$(\widehat{p}(x) =) \sum_{i=0}^{n} \widetilde{c}_i \widetilde{T}_i(x) = \sum_{i=0}^{n} \widehat{c}_i T_i(x) \tag{3.10}$$

*for some $\widehat{c}_i$, which satisfy $\|\widehat{c} - c\|_2 = \mathcal{O}(n^{2.5}\epsilon)$, where $\epsilon$ is defined by (3.9).*

The proof of Theorem 3.3 needs some tools from the theory of orthogonal polynomials. These tools are used in Section 3.2 to develop some intermediate results: building on these, we will then prove Theorem 3.3 in Section 3.3.

---

[3]It is commonly observed that, in practice, the number of iterations is $\approx 2n$. Again, no rigourous proof of this fact for a general pencil is, to our knowledge, currently available.

## 3.2. Orthogonal polynomials with perturbed recurrence relation

For definiteness we will first focus on the Chebyshev polynomial of the first kind. We will then discuss how to extend the results to certain Jacobi orthogonal polynomials [45, Ch. 4].

Let $\{\widetilde{T}_i(x)\}_{i\leq n}$ be the perturbed Chebyshev polynomials satisfying (3.7). Letting $K = [-1, 1]$, we show that

$$\|\widetilde{T}_k - T_k\|_K = \mathcal{O}(k^2\epsilon). \tag{3.11}$$

Here and below $\|p\|_K := \max_{x\in K}|p(x)|$ denotes the $L^\infty$ norm of a continuous function $p(x)$ on the compact interval $K$. We prove (3.11) via the following steps:

1. Prove the roots of $\widetilde{T}_k(x)$ are within $\mathcal{O}(\epsilon)$ of those of $T_k(x)$.

2. Prove the polynomial value $\widetilde{T}_k(x)$ is stable under perturbation in the roots: $\|\widetilde{T}_k - T_k\|_K = \mathcal{O}(k^2\epsilon)$.

Our results extend previous studies on perturbed orthogonal polynomials [28, 34, 35] in that we consider perturbation in any of the recurrence relations.

### 3.2.1. Roots of orthogonal polynomials are insensitive to perturbation in the recurrence relations

For the orthogonal polynomials $\{\widetilde{T}_i(x)\}$ defined by the recurrence relation (3.7) the following statement holds [23, 51]: for all $j = 0, \ldots, n-1$, the roots of $\widetilde{T}_{j+1}(x)$ are the eigenvalues of the $(j+1) \times (j+1)$ Jacobi matrix

$$\frac{1}{2}\begin{bmatrix} -2\epsilon_{j,j} & 1+\epsilon_{j,j-1} & & & \\ 1+\epsilon_{j-1,j} & -2\epsilon_{j-1,j-1} & 1+\epsilon_{j-1,j-2} & & \\ & \ddots & \ddots & \ddots & \\ & & 1+\epsilon_{1,2} & \ddots & 1+\epsilon_{1,0} \\ & & & 2+2\epsilon_{0,1} & -2\epsilon_{0,0} \end{bmatrix}, \tag{3.12}$$

where $\epsilon_{ij}$ are the same as in (3.7). Let $\epsilon$ be as in (3.9). Then (3.12) can be easily transformed into a symmetric tridiagonal matrix plus an $\mathcal{O}(\epsilon)$ perturbation via a diagonal similarity transformation defined by the matrix $D = \mathrm{diag}(I_{n-1}, \sqrt{\frac{1}{2}})$. It is known that simple eigenvalues of a symmetric matrix are well-conditioned, even under nonsymmetric perturbation [44, Sec. IV.5.1]. Specifically, the perturbation in the roots is linear, with constant $\gamma$ of magnitude $\approx 1$, in the spectral norm of the perturbation matrix $\Delta T$. The latter is $\mathcal{O}(\epsilon)$, due to the tridiagonal structure, as can be verified via Gerschgorin's theorem applied to $(\Delta T)^T \Delta T$. To summarize, labelling the roots of $T_j(x)$ (resp. $\widetilde{T}_j(x)$) as $r_i^{(j)}$ (resp. $\widetilde{r}_i^{(j)}$), we can write

$$\max_{j=1,\ldots,n} \max_{i=1,\ldots,j} |r_i^{(j)} - \widetilde{r}_i^{(j)}| := \eta \leq \rho\epsilon \tag{3.13}$$

where $\rho$ is a small constant independent of $n$. Furthermore, if the perturbation is real, as always happens if $p$ has real coefficients, then the roots will stay real unless the perturbation is large enough to make the roots collide, which does not happen under our assumption $\epsilon = |q(n)u| \ll n^{-2}$, because the roots of $T_j(x)$ are separated by at least a distance $\mathcal{O}(n^{-2})$ [45, Thm. 6.21.2]. We note that for Chebyshev polynomials $T_j(x)$ whose roots are $\cos(\frac{i\pi}{2j})$ a simple argument improves the separation to $\frac{4}{n^2}$, as can be seen by adapting (B.2) in the appendix to the specific case of Chebyshev polynomials. This allows for a slightly larger $\epsilon$ (i.e. the results are applicable to larger $n$).

### 3.2.2. Values of Chebyshev polynomials are insensitive to perturbation in the roots

As a preliminary, we need to prove a technical lemma involving the digamma function [1, Sec. 6.3].

**Lemma 3.4.** *Let $\psi(y)$ be the digamma function and consider the smooth function*

$$y \in [0,1] \mapsto \begin{cases} f(y) = -\psi(y)\sin(\pi y) & \text{if } y > 0, \\ f(0) = \pi. \end{cases}$$

*The following properties hold:*

1. *For any $y \in [0, 1/4]$, $f(y) < 3.2222$;*

2. *For any $y \in [1/4, 1]$, $f(y)$ is decreasing.*

*Proof.* Note first that $f(y) > 0$ for all $y \in [0,1]$, being the product of two positive functions. It is immediate that $f(y)$ is decreasing in $[1/2, 1]$, because it is the product of two positive decreasing functions. We claim, and will prove later, that $f(y)$ is concave on $[0, 1/2]$. Observe that by the expansion [1, eq. (6.3.5),(6.3.14)]

$$-\psi(y) = \frac{1}{y} + \gamma - \sum_{n=2}^{\infty} (-1)^n \zeta(n) y^{n-1},$$

where $\gamma \approx 0.5772$ is the Euler-Mascheroni constant and $\zeta(x)$ is Riemann's zeta function, it is straightforward to show that $f'(0) = \gamma > 0$. Similarly, from [1, eq. (6.4.4)] one can prove $f'(1/2) = -\pi^2/2 < 0$. Therefore, $f(y)$ has a unique maximum in $(0, 1/2)$. Both $f(y)$ and $f'(y)$ can be reliably evaluated numerically [3, 37], and hence, we may approximate the maximum and its value with arbitrary precision, e.g., by finding the unique root of $f'(y)$ by the bisection method. With this technique, it is readily estimated that $f'(y_*) = 0$ for $y_* \simeq 0.089638 < 1/4$. Therefore $f$ is decreasing for any $y > y_*$, and a fortiori for any $y > 1/4$. To conclude the proof, from a numerical approximation of $f(y_*)$ up to the desired precision we obtain $|f(y)| \leq f(y_*) < 3.2222$.

It remains to prove that $f(y)$ is concave on $[0, 1/2]$. By [1, eq. (6.3.16)],

$$f(y) = -\psi(y)\sin(\pi y) = \gamma\sin(\pi y) - \sin(\pi y)\sum_{n=1}^{\infty} \frac{y-1}{n(n+y-1)},$$

Writing $g_n(\tilde{y}) = \sin(\pi y) \frac{y-1}{n(n+y-1)}$, we can show that $g_n(y)$ is convex for $n \geq 2$. Indeed, we have

$$g_n''(y) = \frac{2n\pi(n+y-1)\cos(\pi y) + \left(-2n\left(1 + \pi^2(1-y)^2\right) + n^2\pi^2(1-y) + \pi^2(1-y)^3\right)\sin(\pi y)}{n(n+y-1)^3}$$

Since $y \in [0, 1/2]$, the numerator is larger than $\sin(\pi y)h_n(1-y)$ where

$$h_n(z) = n^2\pi^2 z - 2n\left(1 + \pi^2 z^2\right) + \pi^2 z^3,$$

having performed the simple change of variable $1 - y =: z \in [1/2, 1]$. We get

$$h_{n+1} - h_n = (2n+1)\pi^2 z - 2n\left(1 + \pi^2 z^2\right) \geq 0 \quad \forall n, \quad \forall z \in [1/2, 1],$$

as is clear because the above is a concave function of $z$ that takes the values, resp., $n\pi^2 - 2 > 0$ and $(2n-1)\pi^2 - 2 > 0$ at, resp., $z = 1/2$ and $z = 1$. Hence together with $h_2(1-y) > 0$ for $y \in [0, 1/2]$ (which can be verified easily) we conclude that $h_n(1-y) > 0$ for all $n \geq 2$ and $y \in [0, 1/2]$. Since the first term in the sum $\sin(\pi y)\sum_{n=1}^\infty \frac{y-1}{n(n+y-1)}$ is not convex, it remains to prove that for a partial sum $\sin(\pi y)\left(\sum_{n=1}^k \frac{y-1}{n(n+y-1)} - \gamma\right)$ is convex on $[0, 1/2]$ for some $k$. This is true for $k = 2$, as can be verified by direct calculation, and hence, the claim is proved. $\qquad\square$

Now, for any interval $I$, we denote the $L^\infty(I)$ norm of a continuous function by $\|\cdot\|_I$; also, recall $K := [-1, 1]$. Let $T_n(x)$ be the Chebyshev polynomial of degree $n$; recall that $\|T_n\|_K = 1$. Observe that we can write $T_n(x) = \alpha \prod_i (x - r_i)$ where $\alpha > 0$ depends on $n$ (more precisely $\alpha = 1$ for $n \leq 1$ and $\alpha = 2^{n-1}$ otherwise). Here and below, $r_i$ are the roots of the $n$th Chebyshev polynomial. Suppose each root of $T_n(x)$ is perturbed by $\eta_i$, and let $\widetilde{T}_n(x) = \alpha \prod_i (x - r_i - \eta_i)$. We denote $\vec{r} = [r_1, \ldots, r_n] \in \mathbb{R}^n$, $\vec{\eta} = [\eta_1, \ldots, \eta_n] \in \mathbb{R}^n$, $\eta = \|\vec{\eta}\|_\infty$.

Our goal is to prove that the perturbation in the polynomial value $\|T_n - \widetilde{T}_n\|_K / \|T_n\|_K$ is insensitive to perturbation in the roots. In particular, we will prove the following

**Theorem 3.5.** *Let $T_n \in \mathbb{R}[x]$ be the Chebyshev polynomial of degree $n$, and $\widetilde{T}_n$ defined as above. $\|\widetilde{T}_n - T_n\|_K \leq \eta n^2 \|T_n\|_K + \mathcal{O}(\eta^2)$. Moreover, the constant is tight, i.e., for any $c < 1$ there exists a choice of $\vec{\eta}$ such that $\|\vec{\eta}\|_\infty \leq \eta$ and $\|\widetilde{T}_n - T_n\|_K > c\eta n^2 \|T_n\|_K$.*

*Proof.* Let us consider the function $f(\vec{y}; x) = \alpha \prod_i (x - y_i) : \mathbb{R}^n \to \mathbb{R}$. Noting that $T_n(x) = f(\vec{r}; x)$ and expanding $f$ in a Taylor series around $\vec{y} = \vec{r}$ we get

$$\widetilde{T}_n(x) - T_n(x) = \nabla f(\vec{r}; x) \cdot \vec{\eta} + \mathcal{O}(\eta^2).$$

Observe that $\left|\frac{\partial f}{\partial y_i}\right| = \left|\frac{f}{x-y_i}\right|$. Hence, to first order in $\eta$, $\|T_n - \widetilde{T}_n\|_K \leq \eta \sum_i \|\frac{T_n(x)}{x-r_i}\|_K$. Hence, we only need to estimate the function

$$\mathcal{P}(x) := \sum_i \frac{|T_n(x)|}{|x - r_i|}.$$

15

This function satisfies the property that $\mathcal{P}(r_i) = |T_n'(r_i)|$ for all $i$, and furthermore $\mathcal{P}(\pm 1) = |T_n'(\pm 1)|$. Observe further that $\mathcal{P}(x)$ is even, so it suffices to study it for $0 \le x \le 1$.

Clearly, since the involved absolute values can be resolved with a unique choice of signs in each interval of the form $[r_{j+1}, r_j]$, we have that $\mathcal{P}(x)$ is piecewise polynomial and equal to $T_n'(x)$ for $x \ge r_1$. Furthermore it is equal to

$$(-1)^j \left( T_n'(x) + 2 \sum_{i=1}^{j} \frac{T_n(x)}{r_i - x} \right)$$

for all $r_{j+1} \le x \le r_j$. We may limit our study to $j \le n/2$, by symmetry.

Moreover note that for $n = 1, 2$ we have, resp., $\mathcal{P}(x) = 1$ and $2\mathcal{P}(x) = |x - \frac{1}{\sqrt{2}}| + |x + \frac{1}{\sqrt{2}}| \le 2$, and hence, we may assume $n \ge 3$ in the following.

Now let us parametrize $x = \cos(\frac{2j-1+2y}{2n}\pi)$ for $y \in [0,1]$, except if $n$ is even and $j = n/2$: in this case $y \in [0, 1/2]$. Then, recall $T_n'(x) = nU_{n-1}(x)$. We can use $T_n(x) = \cos(n\,\mathrm{acos}(x)) = \cos(n(\frac{2j-1+2y}{2n}\pi)) = \cos(\frac{2j-1+2y}{2}\pi) = (-1)^j \sin(\pi y)$ and similarly, using $U_{n-1}(x) = \frac{\sin(n\,\mathrm{acos}(x))}{\sin(\mathrm{acos}(x))}$ we have $\sin(\frac{2j-1+2y}{2n}\pi)U_{n-1}(x) = \sin(\frac{2j-1+2y}{2}\pi) = (-1)^{j+1}\cos(\pi y)$. Thus we have

$$\mathcal{P}(x) = -n\frac{\cos(\pi y)}{\sin(\frac{2j-1+2y}{2n}\pi)} + \sum_{i=1}^{j} \frac{\sin(\pi y)}{\sin(\frac{i+j-1+y}{2n}\pi)\sin(\frac{j-i+y}{2n}\pi)}. \qquad (3.14)$$

We distinguish the cases $j = 1$ and $j > 1$. Let first $j = 1$, then

$$\mathcal{P}(x) = -n\frac{\cos(\pi y)}{\sin(\frac{1+2y}{2n}\pi)} + \frac{\sin(\pi y)}{\sin(\frac{1+y}{2n}\pi)\sin(\frac{y}{2n}\pi)}.$$

Suppose first $y \le 1/2$. Then $\frac{1+y}{2n} \le \frac{1}{4}$ implying $\sin(\frac{1+y}{2n}\pi) \ge \frac{\sqrt{2}(1+y)}{n}$, and similarly from $\frac{y}{2n} \le \frac{1}{12}$ we see that $\sin(\frac{y}{2n}\pi) \ge \frac{3(\sqrt{3}-1)y}{\sqrt{2}n}$. Therefore we get the upper bound

$$\mathcal{P}(x) \le n^2 \left( \frac{\pi}{3(\sqrt{3}-1)(1+y)} - \frac{2\cos(\pi y)}{\pi(1+2y)} \right),$$

and studying the function between brackets it is revealed that its maximum in $[0, 1/2]$ is achieved at $y = 1/2$ and therefore we have the upper bound

$$\mathcal{P}(x) \le \frac{2\pi}{9(\sqrt{3}-1)}n^2 \lesssim 0.954n^2 < n^2.$$

Now suppose $1/2 \le y \le 1$. In this case $\frac{1+y}{2n}\pi \le \frac{1}{3}\pi$ so we can estimate $\sin(\frac{1+y}{2n}\pi) \ge \frac{3\sqrt{3}(1+y)}{4n}$, $\sin(\frac{y}{2n}\pi) \ge \frac{3y}{2n}$, and $\sin(\frac{1+2y}{2n}\pi) \ge \frac{1+2y}{n}$, yielding the upper bound

$$\mathcal{P}(x) \le n^2 \left( \frac{8\sin(\pi y)}{9\sqrt{3}y(1+y)} - \frac{\cos(\pi y)}{1+2y} \right),$$

and once again studying this function we see that its maximum is achieved at $y = 1/2$, where it is
$$\mathcal{P}(x) \leq \frac{32}{27\sqrt{3}} n^2 \lesssim 0.685 n^2,$$
concluding the analysis for $j = 1$.

For $j \geq 2$, defining $\delta = j - i$ we estimate the second term in $\mathcal{P}(x)$ by
$$\sum_{i=1}^{j} \frac{\sin(\pi y)}{\sin(\frac{i+j-1+y}{2n}\pi) \sin(\frac{j-i+y}{2n}\pi)} \leq n^2 \frac{\sin(\pi y)}{\sqrt{2}} \sum_{\delta=0}^{j-1} \frac{1}{2j-1-\delta+y} \frac{1}{\delta+y}.$$

Recall that the digamma function $\psi(z)$, satisfies [1, eq. (6.3.5),(6.3.6)]
$$\psi(j + z) - \psi(z) = \sum_{\delta=0}^{j-1} \frac{1}{\delta + z}, \tag{3.15}$$
and note that we can write
$$(2j + 2y - 1) \sum_{\delta=0}^{j-1} \frac{1}{2j-1-\delta+y} \frac{1}{\delta+y} = \sum_{\delta=0}^{j-1} \frac{1}{\delta+y} + \sum_{\delta=0}^{j-1} \frac{1}{j+y+\delta},$$
where in the second summation we have relabelled $\delta \to j - 1 - \delta$. Thus, using (3.15) twice, we obtain
$$\sum_{\delta=0}^{j-1} \frac{1}{2j-1-\delta+y} \frac{1}{\delta+y} = \frac{\psi(2j+y) - \psi(y)}{2j-1+2y}. \tag{3.16}$$

Hence
$$\sum_{i=1}^{j} \frac{\sin(\pi y)}{\sin(\frac{i+j-1+y}{2n}\pi) \sin(\frac{j-i+y}{2n}\pi)} \leq n^2 \frac{\sin(\pi y)}{\sqrt{2}} \frac{\psi(2j+y) - \psi(y)}{2j-1+2y}.$$

From [1, eq. (6.3.2)] we see that $\frac{\psi(2j+y)}{2j+2y-1} \leq \frac{H_{2j-1}-\gamma}{2j-1}$, with $\gamma$ the Euler-Mascheroni constant and $H_j$ the $j$th harmonic number. It is easy to verify that $\frac{H_{2j-1}-\gamma}{2j-1}$ is a decreasing function of $j$, and so using $j \geq 2$ we get
$$\frac{\psi(2j+y)}{2j+2y-1} \leq \frac{H_3 - \gamma}{3} = \frac{11 - 6\gamma}{18}.$$

By Lemma 3.4, $-\psi(y)\sin(\pi y) < 3.2222$ on $y \in [0, 1]$. Hence for any $y \in [0, 1/4]$, since $\sin(\pi y) \leq \frac{1}{\sqrt{2}}$, the summation (3.16) is bounded by
$$n^2 \left( \frac{3.2222}{6} + \frac{11 - 6\gamma}{36} \right) < 0.74644 n^2 < n^2,$$
yielding a bound for $\mathcal{P}(x)$ in (3.14) because the cosine term is negative. If $1/4 \leq y \leq 1/2$, we cannot bound the sine other than with 1, but we can use the better

17

bounds $-\psi(y)\sin(\pi y) \le -\psi(1/4)\sin(\pi/4) = \frac{\pi+6\log 2+2\gamma}{2\sqrt{2}}$, by Lemma 3.4, and $\frac{\psi(2j+y)}{2j+2y-1} \le 2\frac{H_3-\gamma}{7} = \frac{11-6\gamma}{21}$. These yield the upper bound

$$\mathcal{P}(x) \le n^2\left(\frac{\pi+6\log 2+2\gamma}{14} + \frac{11-6\gamma}{21\sqrt{2}}\right) < 0.9n^2 < n^2.$$

Finally, when $y \ge 1/2$ we must also control the cosine term, since it is positive. This is easily done since $2j-1+2y \ge 2j \ge 4$ and hence $\sin(\frac{2j-1+2y}{2n}\pi) \ge \frac{4}{n}$, yielding $-n\cos(\pi n)\csc(\frac{2j-1+2y}{2n}\pi) \le |\cos(\pi y)|\frac{n^2}{4}$. Note the improved bounds $-\psi(y)\sin(\pi y) \le -\psi(1/2)\sin(\pi/2) = 2\log 2+\gamma$ and $\frac{\psi(2j+y)}{2j-1+2y} \le \frac{H_{2j-1}-\gamma}{2j} \le \frac{11-6\gamma}{24}$. Hence we get the bound

$$n^2\left(\frac{1}{4} + \frac{\gamma+2\log 2}{4\sqrt{2}} + \frac{11-6\gamma}{24\sqrt{2}}\right) < 0.81922n^2,$$

and we are done.

Finally, for the tightness, since $\|T_n'\|_K = n^2\|T_n\|_K = n^2$, and this is achieved at $x = 1$ [36, Ch. 2] this is the best bound we can get. Note that it is realizable by choosing $\eta_i = \eta$ in the definition of $\widehat{p}$. $\qquad\square$

Finally, we must slightly weaken one assumption in Theorem 3.5. Indeed, $\widetilde{T}_k(x)$ might not have the same leading coefficient as $T_k(x)$. Let $\widetilde{c}_k$ and $c_k$ denote the respective leading coefficients. By (3.7) we deduce that $\widetilde{c}_k \le (1+\epsilon)^k$, hence $\widetilde{c}_k \le (1+k\epsilon)c_k + \mathcal{O}(\epsilon^2)$. Therefore

$$\|\widetilde{T}_k - T_k\|_K \le \|\widehat{T}_k - T_k\|_K + \|(\frac{\widetilde{c}_k}{c_k}-1)\widehat{T}_k\|_K$$

$$\le \|\widehat{T}_k - T_k\|_K + k\epsilon(\|\widehat{T}_k - T_k\|_K + \|T_k\|_K)$$

$$\le k^2\eta + k\epsilon + \mathcal{O}(\epsilon^2) \le 2\rho k^2\epsilon,$$

recalling from the discussion in Section 3.2.1 that $\eta = \rho\epsilon$ for a moderate constant $\rho$.

## 3.3. Proof of Theorem 3.3

Now we prove (3.10). We have

$$\widehat{p}(x) = \sum_{i=0}^{n}\widetilde{c}_i\widetilde{T}_i(x) = \sum_{i=0}^{n}\widetilde{c}_iT_i(x) + \sum_{i=0}^{n}\widetilde{c}_i(\widetilde{T}_i(x) - T_i(x)) =: p_1(x) + p_2(x),$$

where $p_2(x)$ is a polynomial of degree $n$ or lower. To bound $\|p_2\|_K$ we use Theorem 3.5 together with $\|c\|_2 = 1 + \mathcal{O}(\epsilon)$ to obtain $\|p_2\|_K \le \sum_{k=0}^{n}2|\widetilde{c}_i|\rho k^2\epsilon \le \rho n^{2.5}\epsilon + \mathcal{O}(\epsilon^2)$ where $\rho'$ is some moderate constant. Therefore by Chebyshev interpolation we can write

$$p_2(x) = \sum_{i=0}^{n}d_iT_i(x).$$

Denote by $\|p\|_{L^2}$ the $L^2(K)$ norm with weight function $\frac{1}{\sqrt{1-x^2}}$, $\sqrt{\frac{2}{\pi}\int_{-1}^{1}|p(x)|^2\frac{dx}{\sqrt{1-x^2}}}$. Then by the orthogonality of the Chebyshev polynomials it is immediate that $\|p_2\|_{L^2}^2 =$

$2d_0^2 + \sum_{i=1}^n d_i^2 \geq d_0^2 + \sum_{i=1}^n d_i^2 = \|d\|_2^2$ where $d = [d_0, d_1, \ldots, d_n]$. By standard $L^p$ norms embedding inequalities [2, Thm. 2.14], [50, Thm. 2] we then have $\|d\|_2 \leq \|p_2\|_{L^2} \leq \sqrt{2}\|p_2\|_K \leq \sqrt{2}\rho' n^{2.5}\epsilon$. We conclude that

$$\widehat{p}(x) = \sum_{i=0}^n (\widetilde{c}_i + d_i)T_i(x) = \sum_{i=0}^n \widehat{c}_i T_i(x),$$

where $\widehat{c}_i := \widetilde{c}_i + d_i$ satisfies $\|\widehat{c} - c\|_2 \leq \|\widehat{c} - \widetilde{c}\|_2 + \|\widetilde{c} - c\|_2 = \|d\|_2 + \|\widetilde{c} - c\|_2 \leq \sqrt{2}\rho' n^{2.5}\epsilon + \sqrt{n}\epsilon + \mathcal{O}(\epsilon^2) = \mathcal{O}(n^{2.5}\epsilon)$.

This completes the proof of Theorem 3.3.

### 3.4. Generalizing the argument to Jacobi orthogonal polynomials

Extending the results more generally to comrade pencils based on Jacobi orthogonal polynomials $\{P_i^{(\alpha,\beta)}(x)\}$ with $|\alpha|, |\beta| \leq \frac{1}{2}$ can be done essentially by following the same argument as above. The key properties that we need in $P_i^{(\alpha,\beta)}(x)$ are

(i) $\|P_n^{(\alpha,\beta)}\|_K = \mathcal{O}(1)$ or a slowly growing function of $n$; indeed we have $\|P_n^{(\alpha,\beta)}\|_K \leq n^{\max(\alpha,\beta)}$ [45, Thm. 7.32.1] for Jacobi (due to the difference in normalization, the bound is 1 and $n$ for Chebyshev polynomials of the first and second kinds respectively).

(ii) $\{P_i^{(\alpha,\beta)}\}$ is defined by a three-term recurrence $xP_i^{(\alpha,\beta)}(x) = a_i P_{i+1}^{(\alpha,\beta)}(x) + b_i P_i^{(\alpha,\beta)}(x) + d_i P_{i-1}^{(\alpha,\beta)}(x)$, with $a_i, b_i, d_i$ all being $\mathcal{O}(1)$.

(iii) The roots of $P_n^{(\alpha,\beta)}(x)$ lie in the interval $[-1 + \mathcal{O}(\frac{1}{n^2}), 1 - \mathcal{O}(\frac{1}{n^2})]$.

Regarding (iii), indeed the $i$th root of any Jacobi polynomial of degree $n$ lies in the interval $[\cos(\frac{2i}{2n+1}\pi), \cos(\frac{2i-1}{2n+1}\pi)]$ and furthermore for the ultraspherical case $\alpha = \beta$ it lies in $[\cos(\frac{i\pi}{n+1}), \cos((i - \frac{1}{2})\frac{\pi}{n})]$ [45, Thm. 6.3.2]. Hence, one can prove an analogue of Theorem 3.5. The proof is deferred to the appendix, in Theorem B.2.

We outline the proof for comrade pencils, highlighting the differences. The elimination order stays the same in (3.4), in which the tridiagonal elements of the second matrix are replaced by $\{a_i, b_i, d_i\}$, which all remain $\mathcal{O}(1)$. The resulting perturbed polynomials (3.5) and recurrence relation (3.7) are modified respectively to $\widetilde{p}(x) = \sum_{i=0}^n \widetilde{c}_i \widetilde{P}_i^{(\alpha,\beta)}(x)$, and

$$x\widetilde{P}_i^{(\alpha,\beta)}(x) = (a_i + \epsilon_{i,i+1})\widetilde{P}_{i+1}^{(\alpha,\beta)}(x) + (b_i + \epsilon_{i,i})\widetilde{P}_i^{(\alpha,\beta)}(x) + (d_i + \epsilon_{i,i-1})\widetilde{P}_{i-1}^{(\alpha,\beta)}(x).$$

The corresponding matrix of the form (3.12) is still easily converted to a symmetric tridiagonal plus $\mathcal{O}(\epsilon)$ perturbation, so we obtain the counterpart of (3.11) as $\|\widetilde{P}_k^{(\alpha,\beta)}(x) - P_k^{(\alpha,\beta)}(x)\|_K = \mathcal{O}(k^3\epsilon)$ (where one extra power $k$ comes from $\|P_k^{(\alpha,\beta)}\|_K > 1$), thus showing

$$\widetilde{p}(x) = \sum_{i=0}^n \widetilde{c}_i \widetilde{P}_i^{(\alpha,\beta)}(x) = \sum_{i=0}^n \widehat{c}_i P_i^{(\alpha,\beta)}(x), \tag{3.17}$$

19

hence the computed roots are exact roots of a polynomial with slightly perturbed coefficients $\|\widehat{c} - \widetilde{c}\|_2 = \mathcal{O}(n^{3.5}\|c\|_2\epsilon)$ in the same basis $\{P_i^{(\alpha,\beta)}(x)\}$, proving the normwise backward stability of algorithms based on QZ applied to comrade pencils.

To summarize, a comrade pencil for polynomials $p(x)$ expressed in a Jacobi polynomial basis $\{P_i^{(\alpha,\beta)}(x)\}$ with parameters $|\alpha|, |\beta| \leq \frac{1}{2}$ gives a small backward error for the roots if $p(x)$ is scaled to have coefficients $\|c\|_2 = \mathcal{O}(1)$ in that basis and QZ is used to compute the eigenvalues.

# 4. Stability of rootfinding via the QR algorithm for confederate matrices

In this section, we analyze the stability of rootfinding via QR. To this goal we will extend to some nonmonomial basis a geometric result of Arnold [4] and its ramifications in numerical analysis [21].

## 4.1. Arnold transversality theorem for confederate matrices

We first briefly summarize Arnold's theorem.

Let $\Lambda, \mathcal{M}$ be smooth manifolds [32] and $\mathcal{N} \subset \mathcal{M}$ a smooth submanifold [32, Ch. 5]; let $A : \Lambda \to \mathcal{M}$ be a smooth mapping [32, pp. 34]. Following Arnold [4], $A$ is said to be transversal to $\mathcal{N}$ at the point $c \in \mathcal{N}$ if it holds

$$T_c\mathcal{M} = dA(T_{A^{-1}(c)}\Lambda) + T_c\mathcal{N},$$

where $T_xY$ denotes the tangent space [32, pp. 54] to $Y$ at $x$ and $dA$ is the pushforward [32, pp. 55,63] of $A$.

Now we take $\Lambda = \mathbb{R}^n$ and $\mathcal{M} = \mathbb{R}^{n\times n}$, and we let $A$ be the smooth mapping from the non-leading coefficients of a monic polynomial $\phi_n(x) + \sum_{i=0}^{n-1} c_i\phi_i(x)$ to the associated confederate matrix $C_\phi$. Thus, $\text{Im}(A)$ is the set of confederate matrices in the basis $\{\phi_i\}$. Let $\mathcal{N}$ be the orbit under similarity of $C_\phi \in \text{Im}(A)$. Then a first-order expansion shows that the tangent space of $\mathcal{N}$ at $C_\phi$ is the set of commutators $T_{C_\phi}\mathcal{N} = \{[C_\phi, X]$ for some $X \in \mathbb{R}^{n\times n}\}$. On the other hand, it is known that $T_{C_\phi}\mathcal{M}$ is isomorphic to $\mathcal{M}$ and $T_{A^{-1}(C_\phi)}\Lambda$ is isomorphic to $\Lambda$ [32, pp. 51], while by Theorem 2.1 one can see that $dA(T_{A^{-1}(C_\phi)}\Lambda) \cong \mathcal{FR}_n$. Hence, the interpretation of the statement "the mapping $A$ is transversal to $\mathcal{N}$ at $C_\phi$" is the following (see also [20, 21]): "given any matrix $Y \in \mathbb{R}^{n\times n}$ and a specific confederate matrix $C_\phi$, it is possible to find $X \in \mathbb{R}^{n\times n}$, and $F_0 \in \mathcal{FR}_n$, such that

$$Y = \alpha F_0 + \beta[C_\phi, X] \tag{4.1}$$

for some $\alpha, \beta \in \mathbb{R}$".

Arnold states this result for companion matrices [4]. A proof is not explicitly given by Arnold, but is easy to obtain constructively, e.g., by setting $\alpha = \beta = 1$ and by forming $X$ and $F_0$ given $Y$ and $C_\phi$. Specifically, we can let the last row of $X$ be zero (or arbitrary), and since $C_\phi X - XC_\phi$ and $Y$ are the same except for the first row, this determines

the $(n-i)$th row of $X$ inductively for $i = 1, \ldots, n-1$. Thus the whole matrix $X$ is determined, and $F_0$ is obtained directly by the first row of (4.1).

We now claim that the theorem holds more generally for confederate matrices. Indeed, let $M \in \mathcal{M}$ and suppose that $C_\phi$ is a confederate matrix. Denote by $B$ the upper triangular change of basis matrix from the monomials $\{x^i\}_{i=0,\ldots,n-1}$ to the basis $\{\phi_i\}_{i=0,\ldots,n-1}$. Then $C := B^{-1} C_\phi B$ is a companion matrix. Hence, there exist $F_0 \in \mathcal{FR}_n$ and $X \in \mathbb{R}^{n \times n}$ such that $B^{-1} M B = F_0 + [C, X]$. Therefore, $M = B F_0 B^{-1} + [C_\phi, B X B^{-1}]$. Note that, since $B$ is upper triangular, $G_0 := B F_0 B^{-1} \in \mathcal{FR}_n$. Formally, we can state:

**Theorem 4.1** (Arnold's transversality theorem for confederate matrices). *Let $A : \gamma_0 = \begin{bmatrix} c_{n-1} & \ldots & c_0 \end{bmatrix}^T \in \mathbb{R}^n \mapsto A(\gamma_0) = C_\phi \in \mathbb{R}^{n \times n}$, where $C_\phi$ is the confederate matrix in the basis $\{\phi_i\}$ of $p(x) = \phi_n(x) + \sum_{i=0}^{n-1} c_i \phi_i(x)$. Then $A$ is transversal to $\mathcal{N} = \{X C_\phi X^{-1} | X \in GL_n(\mathbb{R})\}$ at any point $C_\phi \in \mathrm{Im}(A)$.*

Arnold also notes that universality holds, that is, the decomposition in (4.1) is unique for the companion case once $\alpha$ and $\beta$ are fixed, and this holds also for confederate matrices.

An important consequence of Theorem 4.1 is that if $\epsilon E$ is a small perturbation, then $\epsilon E = \epsilon G_0 + \epsilon[C_\phi, X]$. Therefore, $C_\phi + \epsilon E = (I - \epsilon X)(C_\phi + \epsilon G_0)(I + \epsilon X) + \mathcal{O}(\epsilon^2)$. Observing that $C_\phi + \epsilon G_0$ is in turn a confederate matrix for another polynomial $\widehat{p}$, we deduce that a small perturbation of the confederate matrix of $p$ is similar (to first order in the norm of the perturbation) to the confederate matrix of a perturbation of $p$.

Going even further, Edelman and Murakami [21] exploit Arnold's transversality theorem to show that, if $F = F_0 + [C, X]$ for some companion matrix $C$ and $F_0 \in \mathcal{FR}_n$, then the elements of $F_0$ are affine functions of the coefficients of $p(x)$. Now we argue that this property also extends to any degree-graded basis. Write $E = G_0 + [C_\phi, Y]$. Then, $F = F_0 + [C, X]$ where $F = B^{-1} E B$, $F_0 = B^{-1} G_0 B$, $C = B^{-1} C_\phi B$. Let $\hat{B} = \begin{bmatrix} \nu & \star \\ 0 & B \end{bmatrix}$ be the change of basis matrix from $\{x^i\}_{i=0,\ldots,n}$ to $\{\phi_i\}_{i=0,\ldots,n}$; the symbol $\star$ denotes elements not relevant for the discussion. Let $p(x) = \begin{bmatrix} 1 & \mu_0 \end{bmatrix} [x^n, \ldots, x^0]^T = \nu^{-1} \begin{bmatrix} 1 & \gamma_0 \end{bmatrix} \hat{B} [x^n, \ldots, x^0]^T$, i.e., the coefficients of $p(x)$ in the monomial basis are $\begin{bmatrix} 1 & \mu_0 \end{bmatrix}$, and the coefficients of $\nu p(x)$ in the basis $\{\phi_i\}$ are $\begin{bmatrix} 1 & \gamma_0 \end{bmatrix}$. Now, $G_0 = B F_0 B^{-1}$ is an affine function of $F_0$, which by [21, Thm 2.1] is an affine function of $\mu_0$, which in turn, by construction, is an affine function of $\gamma_0$. Now it suffices to recall that the composition of affine maps is an affine map. Summarizing, we conclude that for any degree-graded basis we have the following theorem.

**Theorem 4.2** (Edelman–Murakami theorem for confederate matrices). *Let $C_\phi$ be a confederate matrix in the degree-graded basis $\{\phi_i\}$ and $E \in \mathbb{R}^{n \times n}$. Denote by $p(x)$ (resp., $\widehat{p}(x)$) the characteristic polynomial of $C_\phi$ (resp., $C_\phi + \epsilon E$). Then it holds*

$$\widehat{p}(x) - p(x) = \epsilon \sum_{i=0}^{n-1} \delta_i \phi_i(x) + \mathcal{O}(\epsilon^2),$$

*where $\delta_i$ are polynomials in the coefficients of $p(x)$ in the basis $\{\phi_i\}$, and in the $E_{ij}$. Moreover, for each $i$, $\delta_i$ is an affine function of the coefficients of $p(x)$ and, separately, of the elements of $E$.*

We have stated Theorem 4.2 in terms of the characteristic polynomial, which is usually defined to be monic in the monomial basis. Note that with this convention $p(x)$ may not be monic in the basis $\{\phi_i\}$: as explained in Section 2, the elements in the first row of $C_\phi$ are the coefficients of $\nu\kappa^{-1}p(x)$, where $\nu\kappa^{-1}$ is the leading coefficient of $\phi_{n-1}(x)$ expressed in the monomial basis. In the statement of Theorem 4.2, any scaling factor can be absorbed in the $\delta_i$. However, for our goals it is simpler to absorb this factor into $p(x)$ and $\widehat{p}(x)$: therefore, from now on we slightly modify the definition of characteristic polynomial, scaling by $\nu\kappa^{-1}$.

## 4.2. Backward error when approximating roots by QR

Suppose that a backward stable eigensolver is applied to $C_\phi$. It will compute the eigenvalues of a slightly perturbed $C_\phi + \epsilon E$, $\|E\| \le \|C_\phi\|$, for some small constant $\epsilon$. The eigenvalues of $C_\phi + \epsilon E$ are the roots of $\widehat{p}(x)$, and the previous theorem yields (recall (1.2))

$$\frac{\|\widehat{c} - c\|_2}{\|c\|_2} \le k\epsilon.$$

Here, $k$ depends on the basis, on the degree of $p$, and (linearly) on the norm of $E$, which is $\mathcal{O}(\max\{\|c\|_2, \alpha\})$, where $\alpha = \|H_\phi\|_2$ is the norm of the constant part of the confederate matrix (recall Theorem 2.1), and generally $\alpha = \mathcal{O}(1)$ in all the polynomial bases we consider. Therefore,

$$\|\widehat{c} - c\|_2 \le \epsilon k' \|c\|_2 \max\{\|c\|_2, \alpha\}. \tag{4.2}$$

Note the quadratic dependence on $\|c\|_2$ when $\|c\|_2 > \alpha$, in which case $\frac{\|\widehat{c}-c\|_2}{\|c\|_2} \le \epsilon k' \|c\|_2$, suggesting the backward error can be much larger than $\mathcal{O}(\epsilon)$ if $\|c\|_2 \gg 1$.

Observe that Theorem 4.2 does not necessarily imply that $k'$ is of moderate size. It only says that, if $p(x) = \nu\kappa^{-1}(\phi_n(x) + \sum_{i=0}^{n-1} c_i\phi_i(x))$, then $\delta_i$ is of the form $\delta_i = \sum_{i,j,\ell} \beta_{ij\ell} c_\ell E_{ij}$. In principle, it could happen that $|\beta_{ij\ell}| \gg 1$: the only way to check is to carry out the specific calculations in the given basis. From [21], we know that in the case of monomials $\max |\beta_{ij\ell}| = 1$. From the similarity argument that we used, it is possible to obtain bounds for any degree-graded basis, but they involve a factor $\mathrm{cond}(B)$, which may be large (for instance, for the Chebyshev basis $\mathrm{cond}(B) \sim 2^n$). In practice, it could happen that, by a more elaborate argument working directly in a specific basis, much better bounds may be obtained. For example, we verified algebraically that, in the Chebyshev basis and for $n = 5$, $\max |\beta_{ij\ell}| = 2$; numerical experiments corroborate the hypothesis that in this basis, just like in the monomials, $\beta_{ijk}$ do not grow with $n$.

One difficulty in bounding $|\beta_{ij\ell}|$ lies in the fact that the Horner shift matrices appearing in the backward error representation [21] can have norms growing with $n$ for bases other than monomials. It appears therefore that further work would be required to obtain sharp bounds for $|\beta_{ij\ell}|$.

However, we will not pursue a more detailed theoretical analysis because, even if $k' = \mathcal{O}(1)$, our main point is that for QR, unlike QZ, the backward error $\|\widehat{c} - c\|_2$ depends superlinearly (more precisely, quadratically) on $\|c\|_2$ when $\|c\|_2 > 1$. Note that while we regarded $E$ as an arbitrary perturbation matrix, in [21], it is advocated that a "more realistic model of errors" in which $E$ has structure such as upper Hessenberg can possibly predict a much better performance of QR. Nonetheless, even for nonmonomial bases, we verified that QR often gives accurate answers. Yet, counterexamples exist where QR performs much worse than QZ, as we illustrate in Section 6.

## 4.3. Why Theorem 4.2 does not hold for Fiedler matrices

In [20] a study appears of an extension of Arnold's theorem to Fiedler linearizations, and its relations to rootfinding stability. In principle, Fiedler matrices can be viewed as congenial matrices of $p(x)$ in a certain basis that, unlike all other basis considered in this paper, depends on $p(x)$ itself. To see this, recall the second statement in Theorem 2.1, which although stated for confederate matrices, can easily be generalized to congenial matrices: the right eigenvector of a congenial matrix takes the form $[\phi_n(x_i), \ldots, \phi_0(x_i)]^T$ for each root $x_i$. Analytic formulae for the eigenvectors of Fiedler linearizations are known [19, Sec. 7]: they involve the so-called *Horner shifts* of $p(x)$. Hence, the polynomial basis in which a Fiedler matrix is a congenial linearization is also explicitly known. Such a basis is "almost" degree-graded, in the sense that $\deg \phi_i = \sigma(i)$, where $\sigma$ is some permutation of $\{0, 1, \ldots, n-1\}$ (a proof of this fact is not difficult to obtain from the results in [19]). Hence, any Fiedler matrix is permutation similar to a confederate linearization, and it is tempting to conjecture that the arguments of the previous section could be generalized. Tempting, yet wrong: because of the presence of the Horner shifts, in this case the change of basis matrix $B$ depends on $p(x)$ itself, whereas in the derivation of Theorem 4.2 it was tacitly assumed that $B$ is independent of $p$ and $E$.

Remarkably, Arnold's transversality theorem is true for Fiedler also when they are seen as linearizations of $p(x)$ expressed in the monomial basis (as opposed to the Horner shift basis in which they are congenial linearizations), as shown in [20, Thm 5.4]. However, the authors of [20] conclude that there is a cubic, as opposed to quadratic, dependence on $\|c\|_2$, thus showing that indeed, when $B$ is not constant, nonlinearity can occur. To summarize, QR-based rootfinding is unstable also for Fiedler linearizations.

## 4.4. Diagonal balancing

Balancing is a technique to improve the accuracy of computed eigenvalues by reducing the matrix norm, which initially applies a similarity transformation

$$\widehat{C} := XCX^{-1},$$

with the hope that the eigenvalues of the resulting $\widehat{C}$ are better conditioned. Diagonal balancing [43] is employed by default in MATLAB's command `eig`.

Lemonnier and Van Dooren [33] investigate the effect of balancing the companion matrix. For QR, they show that when one allows non-diagonal balancing the optimal

balancing is the one that diagonalizes $C$ (they tacitly make the generic assumption that there are no double roots), that is, when $X$ is the eigenvector matrix. Of course in practice the eigenvector matrix is unknown, and [33] shows that diagonal balancing still attempts to find a reduction of $\|C\|$ within diagonal similarity transformations. A similar argument is given there for QZ applied to the companion pencil.

However, even if balancing is applied to the confederate matrix, $\|\widehat{C}\|$ is never smaller than the largest eigenvalue of $C$. This is true with any $X$, not necessarily diagonal. In some cases such as in Chebfun, one may be looking for roots of $p(x)$ in a certain interval and those outside are irrelevant for the application. In such cases the presence of an irrelevant but large root causes $\|\widehat{C}\|$ to be large, and this impairs the stability of the relevant roots. This suggests that polynomials with a large second leading coefficient, which is the only contributor (besides the leading coefficient) to the diagonals of confederate matrices, are problematic for QR stability. We investigate and confirm this effect in the experiments in Section 6.

## 5. Chebyshev basis and the Chebfun rootfinder

In this section we focus on the Chebyshev basis, particularly the Chebfun rootfinder.

### 5.1. Polynomial approximation preserves normwise backward stability

Chebfun is capable of dealing with a general continuous function $f(x)$ that is not necessarily a polynomial. It first approximates $f(x)$ by a polynomial $p(x)$ via Chebyshev interpolation [47]. Then approximate roots of $f(x)$ are obtained by computing the roots of $p(x)$. The second step can be done in a normwise stable manner, as we proved in Section 3.

Here we claim that provided that Chebyshev interpolation gives an accurate approximant such that on the interval of interest $K$

$$\|f - p\|_K = \epsilon_1 \|p\|_K, \tag{5.1}$$

where $\epsilon_1 = q_1(n)u$ (here and below $q_i$ denotes a modest polynomial), an algorithm that stably computes the roots of $p$ is in turn a backward stable rootfinder for $f$. This can be verified as follows: as shown in Section 3, QZ applied to a colleague pencil for a normalized polynomial in the Chebyshev basis computes the roots of a polynomial $\widehat{p}$ with

$$\|p - \widehat{p}\|_K \le \epsilon_2 \|p\|_K, \quad \epsilon_2 = q_2(n)u. \tag{5.2}$$

Together with (5.1) we obtain

$$\|f - \widehat{p}\|_K \le \|f - p\|_K + \|\widehat{p} - p\|_K = (\epsilon_1 + \epsilon_2)\|f\|_K.$$

Overall this means that Chebfun computes the roots of $f(x)$ in a backward stable manner.

## 5.2. Accuracy estimate of computed roots

So far our discussion has been on the backward stability of rootfinding algorithms. Here we turn to the forward stability; see [26, Ch. 1] for a discussion on backward and forward stability. Consider the computed approximation $\hat{x}$ to a root $x_0$ of $p(x)$ such that $p(x_0) = 0$. By the first-order expansion around $x_0$ we have

$$p(\hat{x}) = p(x_0) + p'(x_0)(\hat{x} - x_0) + \mathcal{O}(\hat{x} - x_0)^2,$$

so the absolute accuracy $\Delta x = \hat{x} - x_0$ is estimated by $|\Delta x| \approx \frac{|p(\hat{x})|}{|p'(x_0)|}$. To estimate $\frac{|p(\hat{x})|}{|p'(x_0)|}$ we first examine the value $|p(\hat{x})|$. Assuming a stable rootfinder such as QZ is used, the computed roots are exact roots of $\widehat{p}$ satisfying (5.2) so we have

$$|p(\hat{x})| = |p(\hat{x}) - \widehat{p}(\hat{x})| \leq \|\widehat{p} - p\|_K = \widehat{q}_2(n)\epsilon\|p\|_K. \tag{5.3}$$

For the denominator $|p'(x_0)|$, we use the fact that for sufficiently smooth $f(x)$, approximation in the function value (5.1) also implies approximation in the derivatives $p'(x) \approx f'(x)$ [47, Thm. 21.1]. We conclude that the accuracy $|\Delta x|$ of the computed root is

$$|\Delta x| \approx \frac{|p(\hat{x})|}{|p'(x_0)|} \lesssim \widehat{q}(n)\epsilon \frac{\|f\|_K}{|f'(x_0)|}. \tag{5.4}$$

This shows that the computed roots are accurate if the function value on the interval $\|f\|_K$ is not too large relative to the derivative $|f'(x_0)|$ at the roots. Conversely, roots at which $|f'(x_0)| \ll \|f\|_K$ may not be computed reliably. For an illustrative example, the famous Wilkinson polynomial $f(x) = \prod_{i=1}^{20}(x - i)$ makes $|f'(x_0)|$ too small for roots in the middle compared with $\max_{x \in [0,20]} |f(x)|$, thus a normwise backward stable algorithm fails to compute accurate roots. A related discussion is given in [14], under the name "dynamical range".

## 5.3. Cause for inaccurate roots and remedy by subdivision

The above observation indicates that roots $x_0$ for which $\frac{\|f\|_K}{|f'(x_0)|} \gg 1$ generally cannot be computed accurately by a polynomial rootfinder that is normwise backward stable. We next argue that sometimes the accuracy can be improved[4].

We discuss two possible remedies for this issue, besides the obvious attempt of using higher-precision arithmetic. The first idea is to attempt to reduce the value of $\|f\|_K$ on the whole interval. This can be done for example by introducing a weighting function $w(x) > 0$, and finding the roots of $g(x) := f(x)w(x)$, which has the same roots as $f(x)$. If $w(x)$ is chosen in such a way that $\frac{\|g\|_K}{|g'(x_0)|}$ is not too large at the roots, the roots can be computed accurately. Clearly the question is how to find such a $w(x)$. A suggestion is made in [14] for the Wilkinson polynomial, but in general constructing an effective $w(x)$ is nontrivial.

---

[4]We do not discuss algorithms that may achieve componentwise stability such as the Ehrlich–Aberth method in the monomial basis (for each root), but rather focus on improving the accuracy using a normwise stable algorithm.

The second remedy is to subdivide the interval into smaller pieces. Since the $\mathcal{O}(n^3)$ cost of computing the eigenvalues of the colleague matrix for a polynomial $p(x)$ is high when the degree $n$ is large, especially when compared with the $\mathcal{O}(n^2)$ cost of other rootfinders such as Ehrlich–Aberth, in Chebfun a technique called subdivision is employed. The idea, simply put, is to divide the interval of interest $[-1, 1]$ into two (or more) subintervals $[-1, \delta]$ and $[\delta, 1]$, and find the roots in each by approximating $p(x)$ by lower-degree polynomials $p_1(x), p_2(x)$ such that $p_1(x) \approx p(x)$ on $[-1, \delta]$ and $p_2(x) \approx p(x)$ on $[\delta, 1]$, then computing the roots of $p_1, p_2$ via the eigenvalues of two colleague matrices. This results in cost reduction provided that the degrees of $p_1, p_2$ are lower than $\left(\frac{1}{2}\right)^{1/3} n \approx 0.79n$, which is typically the case [14].

Here we argue that subdivision can be beneficial also for improving the accuracy of the computed roots, especially if we resample the original function $f(x)$ instead of the polynomial interpolant $p(x)$ to obtain $p_1(x), p_2(x)$, as done in Chebfun.

For definiteness, suppose the original interval is $[-1, 1]$ and after subdivision we work with the interval $[a, b]$. Then the same argument as above shows that the accuracy of a computed root is

$$|\Delta x| = \mathcal{O}\left(\epsilon \frac{\max_{x \in [a,b]} |f(x)|}{|f'(x_0)|}\right).$$

The crucial difference from (5.4) is that the interval is replaced by a smaller $[a, b]$, so the numerator $\max_{x \in [a,b]} |f(x)|$ is smaller than in (5.4), hence so is the error estimate. Clearly the difference is significant if $\max_{x \in [a,b]} |f(x)| \ll \max_{x \in [-1,1]} |f(x)|$. See Section 6.3 for an example where subdivision improves the accuracy significantly.

In practice, it may be difficult to determine a priori how to subdivide in order to achieve good accuracy. One strategy is to first find the Chebyshev interpolant to $f(x)$ on the whole interval, find the roots, and for roots $\hat{x}_i$ for which $\frac{\max_{x \in [-1,1]} |f(x)|}{|f'(x)|}$ is large, recompute the roots in intervals $[a_i, b_i] \ni \hat{x}_i$ chosen small enough so that $\frac{\max_{x \in [a_i, b_i]} |f(x)|}{|f'(x)|}$ is moderate.

We note that the whole argument assumed that the evaluation of the original function $f(x)$ can be done with high (relative) accuracy. If this is not the case, and evaluating $f(x)$ involves an error of size $\delta$, then taking $|p(\hat{x})| \approx \delta$ in (5.4) shows that the accuracy of $\hat{x}_i$ is limited by $\frac{\delta}{|f'(x_0)|}$.

In Chebfun, $p_1, p_2$ are obtained by sampling the global polynomial approximant $p$ on each interval, not the original $f$. This means that generally the accuracy of the roots cannot be improved by subdivision, because the value of $p(x)$ generally contains an error $\|p - f\|_K = \mathcal{O}(\epsilon \|p\|_K)$. Thus our result suggests a possible improvement for Chebfun when high accuracy is a priority: resample the original function $f$ instead of $p$ when subdividing.

We note that a related statement is given in [39], in which subdivision is shown to be important for accuracy when computing common roots of two bivariate functions. In that case subdivision helps even when the polynomial approximant is resampled, as the conditioning depends on the *square* of the polynomial norms.

To summarize this section: roots with small derivatives may be computed inaccurately

by a normwise stable algorithm, and one way to improve the accuracy is to subdivide the interval and work in intervals in which the functions have values comparable with the derivatives at the roots. In addition, subdivision also has the additional accuracy benefit from the reduced degree, hence reduced matrix size $n$; see the experiments in Section 6.2.

### 5.4. Improving Chebfun rootfinder?

Our discussion suggests two possible improvements for the Chebfun `roots` command, both of which should lead to a more robust rootfinder:

1. Replacing QR with QZ to guarantee normwise stability.

2. Resampling the original function $f(x)$ when subdividing to improve accuracy further.

For the first improvement, in Section 6 we show an example where QZ gives significantly better stability than Chebfun `roots`, illustrating that indeed there exist cases where QZ is certainly recommended over QR, even with the increased cost of about a factor three. However, the instability issue in QR caused by a large second-leading coefficient (recall Section 4.4) typically does not arise in Chebfun, because the higher-degree coefficients decay for polynomial approximants of smooth functions, and usually Chebfun truncates the leading coefficients if they are of size $\mathcal{O}(\sqrt{u}\|c\|_2)$. This observation explains the resounding success of the Chebfun rootfinder observed in practice, even without QZ. See [18] for an explanation on why balancing can be expected to successfully reduce the norm of the colleague matrix $\|C_T\|$ in practice. In conclusion, the choice between QR and QZ should be made based on the trade-off between guaranteed stability and speed. Following the analysis presented in this paper, the developers of Chebfun have introduced, starting from version 5, an option in the `roots` command to call QZ by the flag `roots(f,'qz')` [48]. It is worth noting that the speed loss is benign: in Chebfun the speed difference between QR and QZ is typically less than 30%, much less than the factor $\approx 3$ for computing matrix eigenvalues. This is due to the preprocessing steps before the eigensolvers are invoked, including subdivision.

Regarding the second improvement, resampling $f(x)$ would pose a major change to how Chebfun works, and defies the philosophy of Chebfun that everything should be polynomialized. Hence this technique might be best reserved for a rootfinder of $f(x)$ in a non-Chebfun environment.

## 6. Numerical experiments

All the experiments were carried out in MATLAB version R2013a on a desktop machine with Intel Core i7 Processor and 16GB RAM, using IEEE double precision arithmetic.

### 6.1. Balancing and QR vs. QZ

The discussion in Section 4.4 suggested that rootfinding based on QR applied to a comrade matrix may be unstable if the second leading coefficient is large. To illustrate

this observation we test the following linearization-based rootfinders using the Chebyshev basis:

1. QR applied to the colleague matrix $C_T$, without balancing.

2. QR applied to the colleague matrix $C_T$, with balancing.

3. Chebfun command `roots`.

4. Chebfun command `roots(p,'qz')`, which invokes the option of using QZ.

5. QZ applied to the colleague pencil $\lambda X + Y$.

The default Chebfun `roots` algorithm is based on QR for the colleague matrix $C_T$, with balancing and subdivision, along with other techniques; however, subdivision is used for reducing the complexity for polynomials of degree $\approx 100$ or higher, so it is not used for the low-degree problems presented below. Section 3 established that the bottom two algorithms are backward stable.

For comparison purposes, we also show results with the Ehrlich–Aberth method [10, 11], modified to work in the Chebyshev basis (we have coded our own implementation, which is not highly optimized), shown as ChebEA in the tables below.

**Small leading coefficient, large trace**   As a test polynomial we construct the degree eight polynomial

$$p(x) = \sum_{i=0}^{n} c_i T_i(x), \quad c = [-\frac{1}{10}, -\frac{1}{10}, -\frac{1}{10}, -\frac{1}{10}, -\frac{1}{10}, -\frac{1}{10}, 10^{-10}, 1, 10^{-20}], \quad (6.1)$$

and attempt to compute its seven roots on the interval $[-1, 1]$ by the four methods; there is another "irrelevant" root well off the interval. The construction of $p(x)$ is not too special: any coefficient vector $c$ for which the leading coefficient is small and the second leading coefficient is large would show similar behaviors.

Figure 1 plots $p(x)$ and shows the roots computed by each method. The roots computed by QR are visibly inaccurate, with or without balancing, and in fact in this case balancing appears to do more harm than good. QZ, ChebEA and Chebfun `roots` (with QR and QZ) computed all roots accurately[5].

Table 1 shows the backward errors $\frac{\|c-\widehat{c}\|_2}{\|c\|_2}$, in which $\widehat{c}$ is the coefficients of the degree eight polynomial $\widehat{p}$ whose roots are the computed $\hat{x}_i$, scaled $\widehat{p} \leftarrow \alpha\widehat{p}$ with $\alpha = \frac{c^T \widehat{c}}{\|\widehat{c}\|_2^2}$, which minimizes $\|c - \alpha\widehat{c}\|_2$. We also show the values at the computed roots $\max_i |p(\hat{x}_i)|$, which is a measure of the individual backward errors: if $p(\hat{x}) = \epsilon$ then $\widehat{p}(\hat{x}) = 0$ with $\widehat{p}(x) = \sum_{i=0}^{n} c_i T_i(x) - \epsilon T_0(x)$, so $\|\widehat{p} - p\|_K = \epsilon$. The latter is the smallest possible backward error in coefficients $\|c - \alpha\widehat{c}\|_2$ such that $\widehat{p}(\hat{x}_i) = 0$, because $|T_i(x)| \le 1$ for $x \in [-1, 1]$.

---

[5] The reason Chebfun `roots` differs from that of QR with balancing appears to be that the former first removes very small leading coefficients if present. We also note that Chebfun uses the colleague form with coefficients in the first column, not the first row.
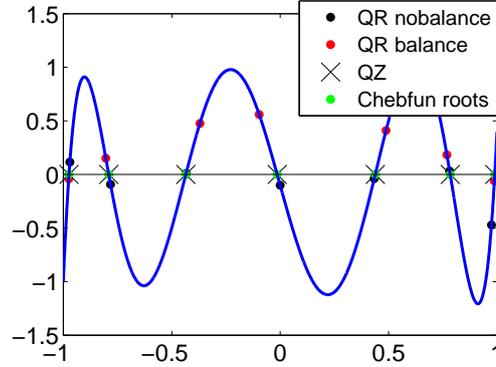
Figure 1: A degree eight polynomial $p(x)$ as in (6.1) and its computed roots by QR with/witout balancing, and QZ. QZ computes stable results while QR does not, and balancing does not help.

Table 1: Normwise backward error $\frac{\|c-\widehat{c}\|_2}{\|c\|_2}$ and $|p(\hat{x}_i)|$ for $p(x)$ in (6.1).

| method | $\frac{\|c-\widehat{c}\|_2}{\|c\|_2}$ | $|\max_i p(\hat{x}_i)|$ |
|---|---|---|
| QR no balancing | 2.6e-01 | 4.7e-01 |
| QR balancing | 3.7e-01 | 5.6e-01 |
| Chebfun `roots` | 1.1e-14 | 2.4e-14 |
| Chebfun `roots(p,'qz')` | 1.3e-14 | 1.5e-14 |
| QZ | 9.0e-15 | 1.6e-14 |
| ChebEA | 7.1e-16 | 1.0e-15 |

Recall from Section 2 that three possible variants of the colleague matrix are available: namely, $C_T^T$, $PC_T P$ and $PC_T^T P$, where $P$ is the antidiagonal permutation matrix. Although for our theoretical analysis we argued that the choice of one of these four possibilities was just a matter of convention, we now note that numerically it can have nontrivial consequences for the stability of QR. In the above example, QR for $C_T^T$ gave normwise backward error 9.2e-01 without balancing and 4.0e-05 with balancing, for $PC_T P$ the error was 1.4e+00 and 1.3e+00 respectively, and 2.4e-01 and 5.4e-04 for $(PC_T P)^T$. Generally, the form that has the coefficients in the last row never seems to be the most accurate, but among the other three, the best choice appears to depend on $p(x)$. This difference is not observed with QZ, which is stable regardless of the choice; indeed essentially the same argument as in Theorem 3.3 proves stability for each variant.

**Unstable results with Chebfun `roots`**   In the last example QR failed but the default Chebfun `roots` worked well. Although extensive experiments suggest that Chebfun `roots` usually gives backward error of size $\mathcal{O}(u)$, there are examples where QZ (or adding the optional flag `'qz'`) gives significantly better accuracy. Table 2 shows the results for a polynomial obtained by changing the leading coefficient $c_8$ from $10^{-20}$ to $10^{-10}$ and $c_6$ from $10^{-10}$ to $-10^{-20}$.

Table 2: Normwise backward error $\frac{\|c-\widehat{c}\|_2}{\|c\|_2}$ and $|p(\hat{x}_i)|$, second example.

| method | $\frac{\|c-\widehat{c}\|_2}{\|c\|_2}$ | $\max_i |p(\hat{x}_i)|$ |
|---|---|---|
| QR no balancing | 8.4e-15 | 4.9e-15 |
| QR balancing | 7.9e-09 | 9.9e-09 |
| Chebfun `roots` | 1.5e-10 | 3.6e-10 |
| Chebfun `roots(p,'qz')` | 1.1e-14 | 1.1e-14 |
| QZ | 2.3e-15 | 3.8e-15 |
| ChebEA | 8.8e-16 | 1.1e-15 |

Overall, the two QZ-based algorithms (QZ and `roots(p,'qz')`) and ChebEA performed stably in all our experiments. Among the stable methods, Ehrlich–Aberth has the advantage of typically giving slightly better accuracy and having $\mathcal{O}(n^2)$ cost, while the advantage of QZ includes its robustness and ease of implementation (and the observed cost is $\mathcal{O}(n^2)$ when subdivision is employed).

**Missed solutions with Chebfun `roots`**   We present an example where the large backward error by QR can cause a solution to be missed. We form a degree three polynomial $p(x)$ by $p = \texttt{chebfun(@(x)1e}-10*\texttt{x.}^3 + \texttt{x.}^2 - \texttt{1e}-\texttt{12})$.

$p(x)$ has two real roots near $\pm 10^{-6}$, and their condition number is such that an $\mathcal{O}(u)$ perturbation in $p(x)$ cannot move them off the real line, which means a stable algorithm should successfully find the roots. QZ for the colleague pencil does this with $|p(x)| = \mathcal{O}(u)$

at both roots. However, Chebfun `roots`, which by default looks for real roots, misses both solutions because QR applied to the colleague matrix finds two nonreal roots near 0 with imaginary parts $\mathcal{O}(\sqrt{u})$. The explanation is that the large backward error in QR caused eigenvalues to coalesce and then move off the real line. Again, with the QZ option `roots(p,'qz')` computed the two roots stably.

**Forward error**   The next example concerns the forward error. Our results show, and the above examples illustrate, that QZ is to be preferred to guarantee backward stability. In applications, however, one might be interested more in the forward errors of the computed roots $|x_i - \hat{x}_i|$. Somewhat surprisingly, sometimes QR gives smaller forward error for some roots than QZ, even with a larger backward error. For example, consider the polynomial $p$ whose exact roots are

$$x = [-1, 0.1, 1, 10^{10}, 2 \times 10^{10}, 10^{15}]. \tag{6.2}$$

Table 3 shows the results; we do not show the results with Chebfun, because due to some preprocessing step such as truncation, it computed a wrong number of roots (even with the `'all'` flag); note that a small perturbation in $p$ on $[-1, 10^{15}]$ can change the number of roots.

Table 3:   Backward error $\frac{\|c-\hat{c}\|_2}{\|c\|_2}$ and relative forward errors $\frac{|\hat{x}_i - x_i|}{|x_i|}$ for $i = 1, \ldots, 6$ for polynomial whose roots are (6.2). "Inf" means the computed root was infinity.

| method | back err. | $i = 1$ | $i = 2$ | $i = 3$ | $i = 4$ | $i = 5$ | $i = 6$ |
|---|---|---|---|---|---|---|---|
| QR no balancing | 2.8e-07 | 9.9e-11 | 2.2e-09 | 1.2e-10 | 4.0e-15 | 4.2e-15 | 1.3e-16 |
| QR balancing | 6.0e-10 | 7.6e-08 | 4.8e-07 | 5.5e-08 | 5.7e-16 | 1.9e-16 | 1.3e-16 |
| QZ | 1.2e-15 | 3.3e-16 | 0 | 0 | 3.3e-01 | Inf | Inf |
| ChebEA | 3.2e-16 | 0 | 2.8e-16 | 0 | 9.5e-16 | 5.7e-16 | 2.5e-16 |

We observe that while QZ is backward stable, the forward error can be much worse than QR for some roots, in particular for $|x_i| \gg 1$. This is unsurprising as an $\mathcal{O}(u)$ perturbation in the coefficients is enough to alter the roots by this amount, because $T_n(x)$ grows rapidly for $|x| > 1$. What is surprising is the accuracy that QR achieves for such large roots. We do not have a clear explanation to this; we conceive that the structure of the colleague matrix is playing a role. Moreover, at least in our experiments, ChebEA seems to get the best of both worlds, both backward and forward errors being small.

This is not necessarily bad news for QZ, at least for computing roots on $[-1, 1]$ as done in Chebfun; its proven backward stability, together with the fact $|T_n(x)| \leq 1$ on $[-1, 1]$, guarantee that these roots are computed with accuracy $\mathcal{O}(\frac{u}{|p'(x_i)|})$.

## 6.2. Error growth with $n$ for colleague pencil

In Section 3 we analyzed the backward stability of rootfinding algorithms based on QZ applied to the colleague pencil, and derived the bound $\mathcal{O}(n^{2.5}\epsilon)$. Clearly, the analysis accounts for the worst-case bound, which usually gives a significant overestimation.

To examine the tightness of the bound, we computed the roots of the degree $n$ Chebyshev polynomial $T_n(x)$ for varying $n$ by forming the $n \times n$ colleague pencil and computing the eigenvalues. We then compute the backward error by forming $\widehat{p}(x) = \prod_{i=1}^{n}(x - \hat{x}_i)$ and expressing it in the Chebyshev basis $\widehat{p}(x) = \alpha \sum_{i=0}^{n} \widehat{c}_i T_i(x)$ and normalizing $\alpha = \frac{c^T \widehat{c}}{\|\widehat{c}\|_2^2}$ as before, then computing the backward error $\|c - \widehat{c}\|_2$. The exact roots are $x_i = \cos(\frac{(2i-1)\pi}{2n})$ for $i = 1, \ldots, n$, and we also computed the forward error as $\max_i |x_i - \hat{x}_i|$.
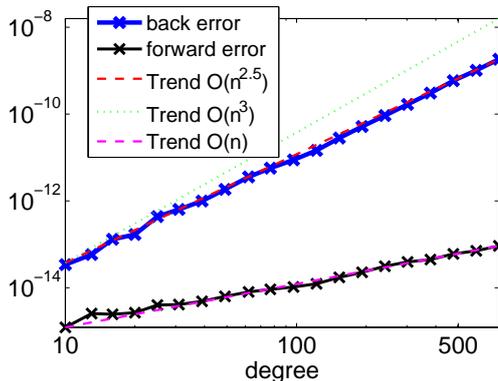


Figure 2: Backward and forward error for computing the roots of $T_n(x)$.

Figure 2 shows the resulting backward and forward errors for $n \in [10, 750]$, which illustrates that the backward error grows like $\mathcal{O}(n^{2.5})$. The forward error grew like $\mathcal{O}(n)$ in the experiment.

We note that the overall error of QZ for colleague consists of (i) the backward error resulting from QZ, (ii) error compression to preserve the colleague nonzero structure, and (iii) error from perturbation in the orthogonal polynomial basis and coefficients. We gave a quantitative bound $\mathcal{O}(n^{2.5})$ for the third error, but the contribution from the first two are, resp., $n^{\tau}$ and $n^{\frac{1}{2}}$, so altogether our analysis gives a bound $\mathcal{O}(n^{3+\tau})$.

## 6.3. Improving accuracy by subdivision

In Section 5 we argued that if $f(x)$ varies widely in magnitude on $[-1, 1]$, then the accuracy of the computed roots of $f(x)$ can be improved by subdivision and resampling $f(x)$. To verify this we consider the function

$$f(x) = xe^{20x}, \tag{6.3}$$

which clearly has one root at $x = 0$. Chebfun approximates $f(x)$ on $[-1, 1]$ by a polynomial $p(x)$ of degree 41, which satisfies $\|f - p\|_K \leq \epsilon \|f\|_K$, and computes nine roots

of $p(x)$ (i.e., eight spurious roots), the one closest to 0 being $\approx -9.9 \times 10^{-8}$. This is an issue caused by the ill-conditioning of the problem, not the stability of the algorithm, and using QZ here does not improve the accuracy.

We can resolve this inaccuracy as follows: subdivide $[-1, 1]$ into ten intervals of width 0.2 (or any width sufficiently smaller than 1 would suffice) and let Chebfun compute the roots of the polynomial approximant of $f(x)$ on each interval. This results in a single computed root at $-5 \times 10^{-16}$. Note that resampling $f(x)$ instead of the polynomial approximant is crucial: if the polynomial approximant $p(x)$ is resampled on each interval, Chebfun computed two roots, the smaller being $-7.3 \times 10^{-8}$.

### Acknowledgements

## A. Impossibility of a generic coefficientwise backward stable root-finder

In the introduction we mentioned several types of stability that we can consider for a rootfinding algorithm. This paper focused on the normwise backward stability for all the computed roots. Here we argue that the more stringent stability of requiring coefficientwise backward stability $\max_i \frac{|\Delta c_i|}{|c_i|} = \mathcal{O}(q(n)u)$ is generically not possible.

For instance, consider the Chebyshev polynomial $T_n(x)$ for $n \geq 2$, whose exact roots are $x_i = \cos(\frac{(2i-1)\pi}{2n})$ for $i = 1, \ldots, n$. Now suppose the computed roots $\hat{x}_i$ are obtained with high relative accuracy satisfying $|x_i - \hat{x}_i| = \mathcal{O}(u)|x_i|$, which (if $x_i$ are not machine representable) is the best one can hope for in finite precision arithmetic. Then the computed roots are the exact roots of the polynomial $\widehat{T}_n(x) = \alpha \prod_{i=1}^{n}(x - \hat{x}_i)$ for any constant $\alpha > 0$. Expressing $\widehat{T}_n(x)$ in the Chebyshev basis $\widehat{T}_n(x) = \sum_{i=0}^{n} \hat{c}_i T_i(x)$ necessarily involves $\hat{c}_i \neq 0$ for some $i < n$. Since the original coefficient is $c_i = 0$ for all $i < n$, this implies that no algorithm is able to compute the roots of $T_n(x)$ with coefficientwise backward stability in finite precision arithmetic.

Note that the above argument holds even when one is looking for the looser condition of coefficientwise stability for each root, because $\hat{x}_i$ is not an exact root of $(1 + \epsilon)T_n(x)$ for any scalar $\epsilon$.

The argument generalizes easily to give the conclusion that, for almost every arbitrary polynomial basis (the crux being that almost any polynomial bases will have roots that are not machine representable), there exists a polynomial for which coefficientwise backward stability cannot be obtained in finite precision arithmetic. It is worth mentioning that

nonetheless the argument fails when the polynomials in the basis have roots that are representable in finite precision arithmetic. A notable example of this kind is when the polynomial basis is the monomials.

## B. An analogue of Theorem 3.5 for Jacobi orthogonal polynomials

Here we extend Theorem 3.5 to Jacobi orthogonal polynomials with parameters $|\alpha|, |\beta| \leq \frac{1}{2}$. The constant we obtain is larger and not necessarily tight, but the main message remains valid that polynomial value is insensitive to perturbation in the roots. We start with a technical lemma applicable to any polynomial $q$.

**Lemma B.1.** *Let $q \in \mathbb{R}[x]$ be a polynomial of degree $n \geq 1$. Suppose moreover $q(r) = 0$ for some $r \in K = [-1, 1]$ and let $J = [r - n^{-2}, r + n^{-2}] \subseteq K$. Then*

$$\left\| \frac{q(x)}{x - r} \right\|_J \leq (e - 1) n^2 \|q\|_K.$$

*Proof.* Expanding $q(x)$ around $x = r$ we get

$$\frac{q(x)}{(x - r)} = \sum_{j=1}^{n} \frac{q^{(j)}(r)}{j!} (x - r)^{j-1}.$$

Hence, we need a bound for $|q^{(j)}(r)|$. To this end we invoke Markov brothers' inequality [16], which states that for $q(x) \in \mathbb{R}[x]_n$,

$$\max_{x \in [-1,1]} |q^{(j)}(x)| \leq \gamma_{j;n} \max_{x \in [-1,1]} |q(x)|, \qquad \gamma_{j;n} = \frac{n^2(n^2 - 1)(n^2 - 4) \cdots (n^2 - (j-1)^2)}{(2j - 1)!!}.$$

Observing that $\gamma_{j;n} \leq n^{2j}$, we conclude that $\|q^{(j)}\|_J \leq \|q^{(j)}\|_K \leq n^{2j} \|q\|_K$. Therefore we get

$$\left\| \frac{q(x)}{(x - r)} \right\|_{J_i} \leq \sum_{j=1}^{n} \frac{\|q^{(j)}\|_{J_i}}{j!} n^{2-2j} \leq n^2 \|q\|_K \sum_{j=1}^{n} \frac{1}{j!} \leq (e - 1) n^2 \|q\|_K.$$

$\square$

Now we specialize to the case where $p(x)$ is a Jacobi polynomial of degree $n$ with parameters $|\alpha|, |\beta| \leq \frac{1}{2}$, such that we can write $p(x) = \prod_i c(x - r_i)$ for some nonzero scalar $c$. We adopt the same notation as in Section 3.2.2: $r_i$ are the roots of the $n$th Jacobi polynomial. Suppose each root is perturbed by $\eta_i$, and let $\widetilde{p}(x) = c \prod_i (x - r_i - \eta_i)$. We denote $\vec{r} = [r_1, \ldots, r_n] \in \mathbb{R}^n$, $\vec{\eta} = [\eta_1, \ldots, \eta_n] \in \mathbb{R}^n$, $\eta = \|\vec{\eta}\|_\infty$.

**Theorem B.2.** *Let $p \in \mathbb{R}[x]$ be a Jacobi polynomial of degree $n \geq 5$ with parameters $|\alpha|, |\beta| \leq \frac{1}{2}$, and $\widetilde{p}$ defined as above. Then it holds $\|p - \widetilde{p}\|_K / \|p\|_K \leq 20.22 n^2 \eta + \mathcal{O}(\eta^2)$.*

*Proof.* For simplicity of discussion we prove the statement for the scaled variant in which $c = 1$, i.e., $p(x) := \prod_i (x - r_i)$.

As in Theorem 3.5 we consider the function $f(\vec{y}; x) = \prod_i (x - y_i) : \mathbb{R}^n \to \mathbb{R}$. Noting that $p(x) = f(\vec{r}; x)$, we have

$$\widetilde{p}(x) - p(x) = \nabla f(\vec{r}; x) \cdot \vec{\eta} + \mathcal{O}(\eta^2).$$

Hence, to first order in $\eta$, $\|p - \widetilde{p}\|_K \leq \eta \sum_i \|\frac{p(x)}{x - r_i}\|_K$, so we only need to estimate $\|\frac{p(x)}{x - r_i}\|_K$.

It is known [45, Thm. 6.3.2] that the $i$th root of any Jacobi polynomial of degree $n$ with $|\alpha|, |\beta| \leq \frac{1}{2}$ lies in the interval $[\cos(\frac{2i}{2n+1}\pi), \cos(\frac{2i-1}{2n+1}\pi)]$. Define $J_i = [r_i - n^{-2}, r_i + n^{-2}]$. We first make two simple observations. First, $i \neq j \Rightarrow J_i \cap J_j = \emptyset$. Indeed, the distance between any two roots is bounded from below by $2n^{-2}$. To see this observe that, for any $i < j$,

$$\cos(\frac{2i\pi}{2n+1}) - \cos(\frac{(2j-1)\pi}{2n+1}) = 2\sin(\frac{(i+j-\frac{1}{2})\pi}{2n+1})\sin(\frac{(j-i-\frac{1}{2})\pi}{2n+1}), \qquad \text{(B.1)}$$

so using (B.1) and $\sin(\theta) > 2\pi^{-1}\theta$ for any $\theta \in (0, \pi/2)$, we obtain the lower bound

$$|r_i - r_j| \geq 2\frac{5}{2n+1}\frac{1}{2n+1} = \frac{10}{(2n+1)^2} > \frac{2}{n^2}. \qquad \text{(B.2)}$$

For the last inequality we used the assumption $n \geq 5$. The second observation is: $J_i \subset K$ $\forall i$. By symmetry, it suffices to check this for $1 - r_1 \geq n^{-2}$. But again, $1 - \cos(\frac{\pi}{2n}) = 2\sin^2(\frac{\pi}{4n}) \geq \frac{1}{n^2}$.

Let us now fix a particular $x \in K = [-1, 1]$, and let $j \in \{0, \ldots, n\}$ be the unique index such that $r_{j+1} < x < r_j$ (if $j = 0$ or $j = n$, this condition reduces to just $x > r_1$ or $x < r_n$). From the above, we have

$$|p(x) - \widetilde{p}(x)| \leq \eta \sum_i \frac{|p(x)|}{|x - r_i|} + \mathcal{O}(\eta^2), \qquad \text{(B.3)}$$

so we see that we essentially need to bound $\sum_i |p(x)||x - r_i|^{-1}$, which is bounded by $\|p\|_K \sum_i |x - r_i|^{-1}$.

Now, there are three cases: either $x \in J_j$, or $x \in J_{j+1}$, or $x$ is in the complement of all $J_i$. We claim that this implies

$$\frac{|p(x)|}{|x - r_j|} + \frac{|p(x)|}{|x - r_{j+1}|} \leq en^2 \|p\|_K. \qquad \text{(B.4)}$$

To obtain this bound we have used Lemma B.1 and the fact that at least one of $|x - r_j|$ and $|x - r_{j+1}|$ is bounded below by $n^{-2}$, thus the sum (B.4) is bounded by $en^2 \|p\|_K$. To handle the special cases $j \in \{0, n\}$, one may just formally define $\frac{|p(x)|}{|x - r_{n+1}|} = \frac{|p(x)|}{|x - r_0|} = 0$, so that the bound clearly remains valid.

It remains to find an upper bound for

$$\sum_{i < j} \frac{1}{|x - r_i|} + \sum_{i > j+1} \frac{1}{|x - r_i|} \leq \sum_{i < j} \frac{1}{|r_i - r_j|} + \sum_{i > j+1} \frac{1}{|r_{j+1} - r_i|}.$$

To do so we use the following bounds: for $i + j - \frac{1}{2} \leq n + \frac{1}{2}$

$$|r_i - r_j| \geq 2\frac{(i + j - \frac{1}{2})(j - i - \frac{1}{2})}{(2n+1)^2} \quad \Rightarrow \quad \frac{1}{|r_i - r_j|} \leq \frac{(2n+1)^2}{2(i + j - \frac{1}{2})(j - i - \frac{1}{2})},$$

and for $i + j - \frac{1}{2} > n + \frac{1}{2}$ we have

$$|r_i - r_j| \geq 2\frac{(2n + 1 - i - j + \frac{1}{2})(j - i - \frac{1}{2})}{(2n+1)^2} \quad \Rightarrow \quad \frac{1}{|r_i - r_j|} \leq \frac{(2n+1)^2}{2(2n - i - j + \frac{3}{2})(j - i - \frac{1}{2})}.$$

Now observe that we can split the summation (if $2j < n + 3$ the second summation is empty and the first one actually stops at $i = j - 1$):

$$\sum_{i<j} \frac{1}{|r_i - r_j|} = \sum_{i=1}^{n+1-j} \frac{1}{|r_i - r_j|} + \sum_{i=n+2-j}^{j-1} \frac{1}{|r_i - r_j|},$$

and defining $\delta = j - i$ this can be bounded by

$$\sum_{\delta=2j-n-1}^{j-1} \frac{(2n+1)^2}{2} \frac{1}{\delta - \frac{1}{2}} \frac{1}{2j - \delta - \frac{1}{2}} + \sum_{\delta=1}^{2j-n-2} \frac{(2n+1)^2}{2} \frac{1}{\delta - \frac{1}{2}} \frac{1}{2n + \frac{3}{2} + \delta - 2j}$$

$$\leq \frac{(2n+1)^2}{2} \left( \sum_{\delta=1}^{j-1} \frac{1}{(j + \frac{1}{2})(\delta - \frac{1}{2})} + \sum_{\delta=1}^{\infty} \frac{1}{\delta^2 - \frac{1}{4}} \right) \leq \frac{(2n+1)^2}{2} \left( \frac{4}{5} + 2 \right)$$

$$\leq \frac{35}{4}n^2.$$

Here we have used the facts that $\sum_{\delta=1}^{j-1} \frac{1}{(j+\frac{1}{2})(\delta-\frac{1}{2})}$ is a decreasing function of $j$ on $[1, \infty)$, and $\sum_{\delta=1}^{\infty} \frac{1}{\delta^2 - \frac{1}{4}} = 2\sum_{\delta=1}^{\infty} \left( \frac{1}{2\delta-1} - \frac{1}{2\delta+1} \right) = 2(1 - \frac{1}{3} + \frac{1}{3} - \frac{1}{5} + \frac{1}{5} \cdots) = 2$. By symmetry we can bound the term $\sum_{i>j+1} \frac{1}{|r_{j+1} - r_i|}$ analogously by $\frac{35}{4}n^2$. Putting it all together we obtain

$$\sum_i \frac{1}{|x - r_i|} \leq \left( e + \frac{35}{2} \right) n^2 \leq 20.22n^2,$$

hence together with (B.3) we obtain

$$\frac{|p(x) - \widetilde{p}(x)|}{\|p\|_K} \leq 20.22n^2\eta + \mathcal{O}(\eta^2),$$

as required. $\qquad\square$

The assumption $n \geq 5$ is a technical one needed to get the separation bound $\frac{2}{n^2}$ in (B.2). For $n \geq 2$ we can obtain the bound $\frac{1}{n^2}$, with which we can proceed similarly to obtain a result with a slightly larger constant. However, little is lost in focusing on $n \geq 5$, since otherwise exact algebraic formulae for the roots are available.

# References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables.* Number 55. Courier Dover Publications, 1972.

[2] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces.* Academic Press, New York, 1975.

[3] D. E. Amos. Algorithm 610: A portable FORTRAN subroutine for derivatives of the psi function. *ACM Trans. Math. Softw.*, 9(4):494–502, December 1983.

[4] V. I. Arnold. On matrices depending on parameters. *Russian Mathematical Surveys*, 26(2):29–43, 1971.

[5] J. L. Aurentz, R. Vandebril, and D. S. Watkins. Fast computation of roots of Companion, Comrade, and related matrices. *BIT*, 54(1):85–111, 2014.

[6] S. Barnett. *Polynomials and Linear Control Systems.* Marcel Dekker Inc., 1983.

[7] D. A. Bini and G. Fiorentino. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numer. Algorithms*, 23(2–3):127–173, 2000.

[8] D. Bini, L. Gemignani, and V. Y. Pan. Fast and stable QR eigenvalue algorithms for generalized companion matrices and secular equations. *Numer. Math.*, 100:373–408, 2005.

[9] D. Bini, L. Gemignani, and V. Y. Pan. Fast QR eigenvalue algorithms for Hessenberg matrices which are rank-one perturbations of unitary matrices. *SIAM J. Matrix Anal. Appl.*, 29(2):556–585, 2007.

[10] D. A. Bini and L. Robol. Solving secular and polynomial equations: A multiprecision algorithm. *J. Comput. Appl. Math.*, 2013.

[11] D. A. Bini. Numerical computation of polynomial zeros by means of Aberth's method. *Numer. Algorithms*, 13(2):179–200, 1996.

[12] P. Boito, Y. Eidelman, and L. Gemignani. Implicit QR for companion-like pencils. *arXiv preprint 1401.5606*, 2014.

[13] P. Boito, Y. Eidelman, and L. Gemignani. Implicit QR for rank-structured matrix pencils. *BIT*, 54(1):85–111, 2014.

[14] J. P. Boyd. Computing real roots of a polynomial in Chebyshev series form through subdivision. *Appl. Numer. Math.*, 56:1077–1091, 2006.

[15] C. A. Boyer. *A History of Mathematics.* Wiley, 1968.

[16] E. W. Cheney. *Introduction to Approximation Theory.* Chelsea, New York, second edition, 1982.

[17] D. A. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer, New York, second edition, 2005.

[18] D. Day and L. Romero. Roots of polynomials expressed in terms of orthogonal polynomials. *SIAM J. Matrix Anal. Appl.*, 43(5):1969–1987, 2005.

[19] F. De Terán, F. M. Dopico, and D. S. Mackey. Fiedler companion linearizations and the recovery of minimal indices. *SIAM J. Matrix Anal. Appl.*, 31(4):2181–2204, 2009/10.

[20] F. De Terán, F. M. Dopico, and J. Pérez. Backward stability of polynomial root-finding using fiedler companion matrices. MIMS EPrint 2014.38, 2014. preprint.

[21] A. Edelman and H. Murakami. Polynomial roots from companion matrix eigenvalues. *Math. Comp.*, 64(210):763–776, 1995.

[22] C. Effenberger and D. Kressner. Chebyshev interpolation for nonlinear eigenvalue problems. *BIT*, 52:933–951, 2012.

[23] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Math. Comp.*, 23(106):221–230, 1969.

[24] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 4th edition, 2012.

[25] I. J. Good. The colleague matrix, a Chebyshev analogue of the companion matrix. *The Quarterly Journal of Mathematics*, 12(1):61–68, 1961.

[26] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 1996.

[27] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, second edition, 2012.

[28] E. K. Ifantis and P. D. Siafarikas. Perturbation of the coefficients in the recurrence relation of a class of polynomials. *J. Comput. Appl. Math.*, 57(1):163–170, 1995.

[29] M. Jenkins and J. Traub. A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration. *Numer. Math.*, 14(3):252–263, 1970.

[30] G. F. Jónsson and S. Vavasis. Solving polynomials with small leading coefficients. *SIAM J. Matrix Anal. Appl.*, 26(2):400–414, 2004.

[31] I. O. Kerner. Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer. Math.*, 8(3):290–294, 1966.

[32] J. M. Lee. *Introduction to Smooth Manifolds*. Springer, 2003.

[33] D. Lemonnier and P. Van Dooren. Optimal scaling of companion pencils for the QZ algorithm. In *Proceedings SIAM Applied Linear Algebra Conference*, 2003.

[34] E. Leopold. Perturbed recurrence relations. *Numer. Algorithms*, 33(1-4):357–366, 2003.

[35] F. Marcellán, J. S. Dehesa, and A. Ronveaux. On orthogonal polynomials with perturbed recurrence relations. *J. Comput. Appl. Math.*, 30(2):203–212, 1990.

[36] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2010.

[37] NAG Ltd. *The NAG C Library Manual, Mark 7*. The Numerical Algorithms Group, 2002. `http://www.nag.co.uk/numeric/cl/manual/pdf/G05/g05cac.pdf` and `http://www.nag.co.uk/numeric/fl/manual/pdf/G05/g05kaf.pdf`.

[38] Y. Nakatsukasa, V. Noferini, and A. Townsend. Vector spaces of linearizations for matrix polynomials: a bivariate polynomial approach. *The Mathematical Institute, University of Oxford, Eprints Archive 1638*, 2013.

[39] Y. Nakatsukasa, V. Noferini, and A. Townsend. Computing the common zeros of two bivariate functions via Bézout resultants. *Numer. Math.*, 2014. to appear.

[40] V. Noferini and F. Poloni. Duality of matrix pencils, Wong chains and linearizations. 2014. Preprint, submitted.

[41] V. Y. Pan. Solving a polynomial equation: Some history and and recent progress. *SIAM Rev.*, 39(2):187–200, 1997.

[42] V. Y. Pan. Approximating complex polynomial zeros: Modified Weyl's quadtree construction and improved Newton's iteration. *Journal of Complexity*, 16(1):213–264, 2000.

[43] B. N. Parlett and C. Reinsch. Balancing a matrix for calculation of eigenvalues and eigenvectors. *Numer. Math.*, 13(4):293–304, 1969.

[44] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory (Computer Science and Scientific Computing)*. Academic Press, 1990.

[45] G. Szegő. *Orthogonal polynomials*. AMS Colloquium Publications, 1992.

[46] F. Tisseur. Backward error and condition of polynomial eigenvalue problems. *Linear Algebra Appl.*, 309(1):339–361, 2000.

[47] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, Philadelphia, 2013.

[48] L. N. Trefethen et al. *Chebfun Version 5*. The Chebfun Development Team, 2014. `http://www.maths.ox.ac.uk/chebfun/`.

[49] P. Van Dooren and P. Dewilde. The eigenstructure of an arbitrary polynomial matrix: computational aspects. *Linear Algebra Appl.*, 50:545–579, 1983.

[50] A. Villani. Another note on the inclusion $L^p(\mu) \subset L^q(\mu)$. *The American Mathematical Monthly*, 92(7):485–487, 1985.

[51] H. S. Wilf. *Mathematics for the Physical Sciences.* Courier Dover Publications, 2013.