Min-Max Elementwise Backward Error for Roots of Polynomials and a Corresponding Backward Stable Root Finder[⋆]

Françoise Tisseur[a], Marc Van Barel[b,1,*]

[a]*Department of Mathematics, The University of Manchester, Manchester, M13 9PL, UK*
[b]*Department of Computer Science, KU Leuven, B-3001 Leuven (Heverlee), Belgium*

Abstract

A new measure called min-max elementwise backward error is introduced for approximate roots of scalar polynomials $p(z)$. Compared with the elementwise relative backward error, this new measure allows for larger relative perturbations on the coefficients of $p(z)$ that do not participate much in the overall backward error. By how much these coefficients can be perturbed is determined via an associated max-times polynomial and its tropical roots. An algorithm is designed for computing the roots of $p(z)$. It uses a companion linearization $C(z) = A - zB$ of $p(z)$ to which we added an extra zero leading coefficient, and an appropriate two-sided diagonal scaling that balances A and makes B graded in particular when there is variation in the magnitude of the coefficients of $p(z)$. An implementation of the QZ algorithm with a strict deflation criterion for eigenvalues at infinity is then used to obtain approximations to the roots of $p(z)$. Under the assumption that this implementation of the QZ algorithm exhibits a graded backward error when B is graded, we prove that our new algorithm is min-max elementwise backward stable. Several numerical experiments show the superior performance of the new algorithm compared with the MATLAB `roots` function. Extending the algorithm to polynomial eigenvalue problems leads to a new polynomial eigensolver that exhibits excellent numerical behaviour compared with other existing polynomial eigensolvers, as illustrated by many numerical tests.

Keywords: zeros of polynomials, polynomial root finder, elementwise backward error, tropical roots, polynomial eigenvalue problems, (block) companion linearization,
2000 MSC: 65F15, 65H04, 30C15, 15A22, 15A80, 15A18, 47J10

[⋆] Version of October 25, 2020.

[*] Corresponding author.

Email addresses: `francoise.tisseur@manchester.ac.uk` (Françoise Tisseur), `marc.vanbarel@cs.kuleuven.be` (Marc Van Barel)

[1] This author was partially supported by the Research Council KU Leuven, C1-project (Numerical Linear Algebra and Polynomial Computations), and by the Fund for Scientific Research–Flanders (Belgium), G.0828.14N (Multivariate polynomial and rational interpolation and approximation), and EOS Project no 30468160.

1. Introduction

We consider the problem of computing all the zeros z_k, $k = 1, 2, \ldots, d$, of the scalar polynomial $p(z)$ of degree d expressed in the monomial basis as

$$p(z) = \sum_{i=0}^{d} p_i z^i.$$

We denote by $\widehat{z}_k$ the approximate zeros computed by some algorithm whose numerical stability we want to assess. For this we consider the backward error in a *global* way, i.e., for all computed roots at the same time. The computed zeros $\widehat{z}_k$ are the exact zeros of a polynomial

$$\widetilde{p}(z) = \sum_{i=0}^{d} \widetilde{p}_i z^i = \widetilde{p}_d(z - \widehat{z}_1) \cdots (z - \widehat{z}_d) = p(z) + \Delta p(z)$$

and the backward error measures the difference between the vector of coefficients $\mathbf{p} = [p_0, p_1, \ldots, p_d]$ of the given polynomial $p(z)$ and the vector of coefficients $\widetilde{\mathbf{p}} = [\widetilde{p}_0, \widetilde{p}_1, \ldots, \widetilde{p}_d]$ of the polynomial $\widetilde{p}(z)$. Assuming that $p_d = \widetilde{p}_d$, we can consider the normwise relative backward error

$$\eta^{\text{norm}} = \frac{\|\widetilde{\mathbf{p}} - \mathbf{p}\|}{\|\mathbf{p}\|} = \frac{\|\Delta \mathbf{p}\|}{\|\mathbf{p}\|}$$

for some vector norm $\|\cdot\|$ or the elementwise relative backward error

$$\eta_{|\mathbf{p}|}^{\text{elem}} = \max_{i, p_i \neq 0} \frac{|\widetilde{p}_i - p_i|}{|p_i|} \quad \text{if } \widetilde{p}_i = 0 \text{ whenever } p_i = 0, \tag{1.1}$$

and $\eta_{|\mathbf{p}|}^{\text{elem}} = \infty$ if $\widetilde{p}_i \neq 0$ when $p_i = 0$ for some i. This elementwise backward error was studied in [5] expanding on earlier work by Van Dooren and Dewilde [15]. In fact, with the latter measure of the backward error, there is no backward stable polynomial root solver [9] while there exist several normwise backward stable algorithms, e.g., the fast polynomial root solver described in [2]. Example 1 below shows that the normwise backward error η^{norm} can be much smaller than the elementwise relative backward error $\eta_{|\mathbf{p}|}^{\text{elem}}$ and when combined with a condition number, they do not provide sharp upper bounds on the relative errors $|z_i - \widehat{z}_i|/|\widehat{z}_i|$. So we introduce in Section 2 a new measure of the backward error, denoted by $\eta_{\widetilde{\boldsymbol{\gamma}}}^{\text{elem}}$, called min-max elementwise backward error, and for which the perturbations Δp_i are measured relative to some parameters $\widetilde{\gamma}_i \geq |p_i|$. In Section 3, we show the connection between the parameters $\widetilde{\boldsymbol{\gamma}} = [\widetilde{\gamma}_0, \widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_d]$ associated with this new backward error measure and the tropical roots of the max-times polynomial $\mathtt{t}p(x) = \max_i(|p_i|x^i)$ associated with $p(z)$. In Section 4, we describe a new polynomial root finder for $p(z)$ based on a $(d+1) \times (d+1)$ companion linearization $C(z) = A - zB$ of the grade $d+1$ polynomial $0z^{d+1} + p(z)$ and an appropriate two-sided diagonal scaling of $C(z)$ that balances the matrix A and makes the matrix B graded when there are large variations in the magnitude of the tropical roots. This property of the scaled pencil is crucial for the numerical stability of our algorithm and can be difficult to achieve on a companion linearization of $p(z)$ but not for $0z^{d+1} + p(z)$. The diagonal scaling of $C(z)$ is then followed by a deflation of the artificially introduced eigenvalue at infinity. Finally, we use an implementation of the QZ algorithm with a "strict" deflation criterion for the eigenvalues at infinity to compute the finite eigenvalues of the scaled and deflated pencil, which we return as approximate roots of $p(z)$. We prove in Section 5 that this new polynomial root finder is min-max elementwise backward stable under the assumption that, when applied to a pencil $A - zB$ with A well-balanced and B graded, the QZ algorithm with strict deflation at infinity computes the exact generalized Schur form of a perturbed pencil $A + \Delta A - z(B + \Delta B)$ with $|(\Delta A)_{ij}|$ of order of the machine precision ϵ_{mach} and a ΔB that can be written as ϵ_{mach} times a graded matrix. Section 6 presents numerical experiments that illustrate the min-max elementwise backward stability of the new polynomial root finder. We explain how to extend our algorithm to the computation of eigenvalues of matrix polynomials. This leads to a new polynomial eigensolver based on a tropically scaled block companion pencil. Numerical experiments show that this new polynomial eigensolver and the eigensolver based on a tropically scaled Lagrange linearization described in [14] both compute eigenvalues with small relative normwise backward errors. An advantage of the new eigensolver over that in [14] is that it does not require the computation of "well-separated tropical roots" and is easier to implement. Section 7 gives our conclusions.

Example 1. *Let us now compute the three measures η^{norm}, $\eta^{\text{elem}}_{|\mathbf{p}|}$ and $\eta^{\text{elem}}_{\widetilde{\boldsymbol{\gamma}}}$ for the backward error when computing the roots of*

$$p(z) = z^4 - z^3 + 2 \cdot 10^{-25} z^2 + 10^{-30} z - 10^{-60} \tag{1.2}$$

using the MATLAB *function* `roots` *and the new algorithm* (*written in* MATLAB). *The results are provided in Table 1.1 together with the relative forward errors*

$$\operatorname{err}(\widehat{z}_k) := \frac{|z_k - \widehat{z}_k|}{|z_k|},$$

$k = 1, \ldots, 4$. *For this example, the new algorithm computes the roots of $p(z)$ more accurately than* `roots`. *Note that we could wrongly decide that the roots have been well computed when looking at η^{norm} when using* `roots`.

Assuming that the zeros z_k of $p(z)$ are all simple and neglecting the higher order terms, the relative forward error can be written as

$$\operatorname{err}(z_k) = \frac{|\Delta p(\widehat{z}_k)|}{|z_k|\,|p'(\widehat{z}_k)|}. \tag{1.3}$$

It will be clear from Section 2 that the following upper bounds for the numerator in (1.3) *hold:*

$$|\Delta p(\widehat{z}_k)| \le \begin{cases} \eta^{\text{norm}} \|[p_0, p_1, \ldots, p_d]\| \; \|1, \widehat{z}_k, \ldots, \widehat{z}_k^d]\|, & (1.4\text{a}) \\ \eta^{\text{elem}}_{|\mathbf{p}|} \displaystyle\sum_{i=0, p_i \neq 0}^{d} |p_i \widehat{z}_k|^i, & (1.4\text{b}) \\ \eta^{\text{elem}}_{\widetilde{\boldsymbol{\gamma}}} (d+1) \displaystyle\max_{0 \le j \le d} |p_j \widehat{z}_k^j|. & (1.4\text{c}) \end{cases}$$

Note that the small roots of $p(z)$ in (1.2) *are ill conditioned for a normwise measure of the perturbations, since $\|[p_0, p_1, \ldots, p_4]\| \; \|1, \widehat{z}_k, \ldots, \widehat{z}_k^4]\|$ is of order one but $|z_k|\,|p'(\widehat{z}_k)|$*

Table 1.1: Relative error $\mathrm{err}(\widehat{z}_k)$ for the four roots of $p(z)$ in (1.2) computed by `roots` and by the new root solver, and corresponding backward errors.

	`roots`	new algorithm
z_k	$\mathrm{err}(\widehat{z}_k)$	$\mathrm{err}(\widehat{z}_k)$
`-9.999999999000001e-16`	`1.5e-09`	`1.1e-16`
`+9.999999999999999e-31`	`5.1e-02`	`1.5e-16`
`+1.000000000100000e-15`	`1.5e-09`	`2.1e-16`
`+1.000000000000000e+00`	`0`	`2.2e-16`
	$\eta^{\mathrm{norm}} =$ `8.2e-27`	$\eta^{\mathrm{norm}} =$ `4.7e-16`
	$\eta^{\mathrm{elem}}_{\lvert\mathbf{p}\rvert} =$ `5.1e-02`	$\eta^{\mathrm{elem}}_{\lvert\mathbf{p}\rvert} =$ `1.5e-06`
	$\eta^{\mathrm{elem}}_{\widetilde{\boldsymbol{\gamma}}} =$ `5.1e-02`	$\eta^{\mathrm{elem}}_{\widetilde{\boldsymbol{\gamma}}} =$ `6.7e-16`

is very small. On the other hand, all the roots of $p(z)$ are well-conditioned when perturbations are measured elementwise, i.e., the values of $(\sum_{i=0,p_i\neq 0}^{4} |p_i||\widehat{z}_k|^i)/(|\widehat{z}_k p'(\widehat{z}_k)|$ and $\max_{0\le j\le 4} |p_j \widehat{z}_k^j|/(|\widehat{z}_k p'(\widehat{z}_k)|$ are of order 1. From (1.4c)*, it is clear that the computed zeros are the exact zeros of a polynomial with a relative error of the order of $\eta^{\mathrm{elem}}_{\widetilde{\boldsymbol{\gamma}}}$ on the coefficients of the* dominant *terms. Hence, the zeros computed by the new algorithm can be seen as the exact zeros of a polynomial whose coefficients were rounded up to the order of the machine precision since $\eta^{\mathrm{elem}}_{\widetilde{\boldsymbol{\gamma}}} = 6.7 \times 10^{-16} \approx 3\epsilon_{mach}$. This is the best we can hope for when computing in finite precision. This results in computed zeros with relative forward errors of the size of the machine precision.*

2. Min-max elementwise backward error

Without loss of generality, we can assume the zero roots have been deflated and that the resulting polynomial $p(z)$ has no root equal to zero, so that $p_0 \neq 0$. The *elementwise backward error for an approximate root* $\widehat{z}_k$ of $p(z)$ is defined as

$$\eta^{\mathrm{elem}}_{\boldsymbol{\alpha}}(\widehat{z}_k) = \min\{\epsilon :\ p(\widehat{z}_k) + \Delta p(\widehat{z}_k) = 0,\ |\Delta\mathbf{p}| \le \epsilon\boldsymbol{\alpha}\}, \tag{2.1}$$

where $\Delta p(z) = \sum_{i=0}^{d} \Delta p_i z^i$, $\Delta\mathbf{p} = [\Delta p_0, \ldots, \Delta p_d]$, the entries of $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_d]$ are nonnegative parameters, and the inequality $|\Delta\mathbf{p}| \le \epsilon\boldsymbol{\alpha}$ holds componentwise. The parameters α_i allow freedom in how perturbations are measured, for example, in an absolute sense with $\alpha_i = 1$ or in a relative sense if $\alpha_i = |p_i|$. It follows from $p(\widehat{z}_k) + \Delta p(\widehat{z}_k) = 0$ that

$$|p(\widehat{z}_k)| = |\Delta p(\widehat{z}_k)| \le \sum_{i=0}^{d} |\Delta p_i||\widehat{z}_k|^i \le \epsilon \sum_{i=0}^{d} \alpha_i |\widehat{z}_k|^i \tag{2.2}$$

so that $\eta^{\mathrm{elem}}_{\boldsymbol{\alpha}}(\widehat{z}_k) \ge |p(\widehat{z}_k)|/\sum_{i=0}^{d} \alpha_i |\widehat{z}_k|^i$. It is easy to check that the lower bound is attained for the perturbations $\Delta p_i = -(\sum_{i=0}^{d} \alpha_i |\widehat{z}_k|^i)^{-1} \alpha_i \mathrm{sign}(\widehat{z}_k^i) p(\widehat{z}_k)$, $i = 0, \ldots, d$. Hence

$$\eta^{\mathrm{elem}}_{\boldsymbol{\alpha}}(\widehat{z}_k) = \frac{|p(\widehat{z}_k)|}{\sum_{i=0}^{d} \alpha_i |\widehat{z}_k|^i}. \tag{2.3}$$

Note that the backward error (2.1) is just a particular case of the componentwise backward error for an approximate solution to a linear system $Ax = b$ with rectangular matrix A equal to the row vector $\mathbf{p} = [p_0, \dots, p_d]$ and $b = 0$ [8, Sec. 7.2]. So (2.3) is a special case of Oettli and Prager's explicit expression for the componentwise backward error for linear systems [11].

Now with the particular choice $\boldsymbol{\alpha} = |\mathbf{p}| = [|p_0|, \dots, |p_d|]$ to measure the perturbations, we obtain the *relative componentwise backward error*

$$\eta_{|\mathbf{p}|}^{\text{elem}}(\widehat{z}_k) = \frac{|p(\widehat{z}_k)|}{\sum_{i=0}^{d} |p_i \widehat{z}_k^i|}. \tag{2.4}$$

One of our contributions in this paper is another choice for the vector of parameters $\boldsymbol{\alpha}$, weaker than $\boldsymbol{\alpha} = |\mathbf{p}|$ but still meaningful. We rewrite the first upper bound in (2.2) as

$$\sum_{i=0}^{d} |\Delta p_i||\widehat{z}_k|^i = \sum_{i=0, p_i \neq 0}^{d} |\Delta p_i||\widehat{z}_k|^i + \sum_{i=0, p_i = 0}^{d} |\Delta p_i||\widehat{z}_k|^i, \tag{2.5}$$

and define

$$j := \arg\max_i |p_i||\widehat{z}_k|^i.$$

Note that for such j, $p_j \neq 0$. Now the term $|\Delta p_i||\widehat{z}_k|^i$ does not affect the order of magnitude of the sum in (2.5) when

$$|\Delta p_i||\widehat{z}_k|^i \leq |\Delta p_j||\widehat{z}_k|^j \iff \begin{cases} \dfrac{1}{\beta_i}\dfrac{|\Delta p_i|}{|p_i|} \leq \dfrac{|\Delta p_j|}{|p_j|} & \text{if } p_i \neq 0, \\ \dfrac{1}{\beta_i}|\Delta p_i| \leq \dfrac{|\Delta p_j|}{|p_j|} & \text{otherwise,} \end{cases} \tag{2.6}$$

with

$$\beta_i = \begin{cases} \dfrac{|p_j||\widehat{z}_k|^j}{|p_i||\widehat{z}_k|^i} \geq 1 & \text{if } p_i \neq 0, \\ |p_j||\widehat{z}_k|^{j-i} & \text{otherwise.} \end{cases} \tag{2.7}$$

This suggests choosing $\boldsymbol{\alpha} = \widetilde{\boldsymbol{\gamma}}$ to measure the perturbations in (2.1), where $\widetilde{\boldsymbol{\gamma}} = [\tilde{\gamma}_0, \dots, \tilde{\gamma}_d]$ with

$$\widetilde{\gamma}_i = \begin{cases} \beta_i |p_i| & \text{if } p_i \neq 0, \\ \beta_i & \text{otherwise,} \end{cases} \qquad i = 0, \dots, d. \tag{2.8}$$

With this choice of parameters,

- larger perturbations are allowed on coefficients that do not participate much to the upper bound in (2.2), i.e., on the modulus of the residual $p(\widehat{z}_k)$, but
- the sparsity structure of the problem may not be preserved, that is, if $p_i = 0$ then $\Delta p_i \neq 0$ is allowed.

Note that the entries of $\widetilde{\boldsymbol{\gamma}}$ depend on $\widehat{z}_k$. Since

$$\sum_{i=0}^{d} \widetilde{\gamma}_i |\widehat{z}_k|^i = (d+1)|p_j \widehat{z}_k^j| = (d+1) \max_i |p_i||\widehat{z}_k^i|,$$

we find that

$$\eta^{\mathrm{elem}}_{\widetilde{\boldsymbol{\gamma}}}(\widehat{z}_k) = \frac{1}{d+1} \cdot \frac{|p(\widehat{z}_k)|}{\max_i |p_i||\widehat{z}_k^i|}.$$

Hence,

$$\eta^{\mathrm{elem}}_{\widetilde{\boldsymbol{\gamma}}}(\widehat{z}_k) \leq \eta^{\mathrm{elem}}_{|\mathbf{p}|}(\widehat{z}_k) \leq (d+1)\eta^{\mathrm{elem}}_{\widetilde{\boldsymbol{\gamma}}}(\widehat{z}_k). \tag{2.9}$$

As mentioned in the Introduction, we are interested in a global way of measuring the backward error for a given (usually computed) approximation $\widehat{\mathbf{z}} = [\widehat{z}_1, \ldots, \widehat{z}_d]^T$ to all the zeros $\mathbf{z} = [z_1, \ldots, z_d]^T$ of $p(z)$. A natural extension of the definition for the backward error for a single zero provided in (2.1) leads to

$$\eta^{\mathrm{elem}}_{\boldsymbol{\alpha}}(\widehat{\mathbf{z}}) := \min\{\epsilon : \ p(\widehat{\mathbf{z}}) + \Delta p_\mu(\widehat{\mathbf{z}}) = 0, \ |\Delta \mathbf{p}_\mu| \leq \epsilon \boldsymbol{\alpha}\}, \tag{2.10}$$

where $p(\widehat{\mathbf{z}})$ denotes the vector $[p(\widehat{z}_1), \ldots, p(\widehat{z}_d)]^T$,

$$\Delta p_\mu(z) = \sum_{i=0}^{d} \Delta p_{\mu,i} z^i = \mu \widetilde{p}(z) - p(z), \quad \mu \in \mathbb{C} \setminus \{0\},$$

with

$$\widetilde{p}(z) = p_d \prod_{j=1}^{d} (z - \widehat{z}_j) = p_d z^d + \widetilde{p}_{d-1} z^{d-1} + \cdots + \widetilde{p}_1 z + \widetilde{p}_0 \tag{2.11}$$

fixed, and $\Delta \mathbf{p}_\mu = [\Delta p_{\mu,0}, \ldots, \Delta p_{\mu,d}]$ with $\Delta p_{\mu,i} = p_i - \mu \widetilde{p}_i$. If $\Delta p_{\mu,i} = 0$ when $\alpha_i = 0$, the last constraint in (2.10) can be rewritten as $\max_{i, \alpha_i \neq 0} |\Delta p_{\mu,i}|/\alpha_i \leq \epsilon$. Hence, since μ is the only free parameter, we have that

$$\eta^{\mathrm{elem}}_{\boldsymbol{\alpha}}(\widehat{\mathbf{z}}) = \begin{cases} \displaystyle\min_{\mu \in \mathbb{C} \setminus \{0\}} \max_{i, \alpha_i \neq 0} \frac{|\Delta p_{\mu,i}|}{\alpha_i} & \text{if } \Delta p_{\mu,i} = 0 \text{ when } \alpha_i = 0, \\ \infty & \text{otherwise.} \end{cases} \tag{2.12}$$

When $p(z)$ has real coefficients and the approximate roots $\widehat{\mathbf{z}}$ are symmetric with respect to the real axis, then the coefficients of $\widetilde{p}(z)$ are real and we can minimize over $\mu \in \mathbb{R} \setminus \{0\}$ in (2.12). Then, in this case, (2.12) can be rewritten as a linear programming problem that can be solved by the simplex method.

When $\boldsymbol{\alpha} = |\mathbf{p}|$, the elementwise relative backward error in (1.1) is just an upper bound on (2.12) corresponding to fixing μ to be 1.

Let us define

$$\gamma_i = \min_{|z| \geq 0} \beta_i(z), \qquad \beta_i(z) = \begin{cases} \displaystyle\max_{0 \leq j \leq d} \frac{|p_j z^j|}{|p_i z^i|} & \text{if } p_i \neq 0, \\ \displaystyle\max_{0 \leq j \leq d} \frac{|p_j z^j|}{|z|^i} & \text{if } p_i = 0, \end{cases} \qquad i = 0, \ldots, d, \tag{2.13}$$

and consider the generalization to $\widetilde{\boldsymbol{\gamma}}$ in (2.8),

$$\widetilde{\gamma}_i = \begin{cases} \gamma_i |p_i| & \text{if } p_i \neq 0, \\ \gamma_i & \text{if } p_i = 0, \end{cases} \qquad i = 0, \ldots, d. \tag{2.14}$$

Definition 1 (Min-max elementwise backward error). *The min-max elementwise backward error for the approximate zeros* $\widehat{\mathbf{z}} = [\widehat{z}_1, \ldots, \widehat{z}_d]^T$ *of* $p(z) = \sum_{i=0}^{d} p_i z^i$ *is* $\eta_{\widetilde{\boldsymbol{\gamma}}}^{\mathrm{elem}}(\widehat{\mathbf{z}})$ *in* (2.10) *with* $\widetilde{\boldsymbol{\gamma}} = [\widetilde{\gamma}_0, \ldots, \widetilde{\gamma}_d]$ *defined in* (2.14).

The naming *min-max* comes from the min-max characterization of the γ_i in (2.13). Note that the latter do not depend on the approximate zeros $\widehat{z}_k$, $k = 1, \ldots, d$. Because $\gamma_i \geq 1$ if $p_i \neq 0$, our choice for $\widetilde{\gamma}_i$ allows a larger relative error on those coefficients having a corresponding γ_i larger than one. For $p_i = 0$, we allow a certain absolute error without changing the order of magnitude of the upper bound (2.2) for any of the zeros $\widehat{z}_k$. This does not preserve sparsity in the data but unlike for the elementwise relative backward error, the min-max elementwise backward error is always finite. Indeed, we will show in the next section that for $p_i = 0$, $\gamma_i > 0$ so $\widetilde{\gamma}_i \neq 0$.

3. Connection between the γ_i and the tropical roots

Based on the polynomial $p(z)$, consider the corresponding tropical polynomial $\mathtt{t}p(x)$ in the max-times semiring $\mathbb{R}_{\max,\times}$ consisting of the set of nonnegative real numbers $\mathbb{R}^+$ with the operations $\oplus$ and $\otimes$. The $\oplus$ operation is taking the maximum value of the terms and the $\otimes$ operation is the classical multiplication. The tropical polynomial $\mathtt{t}p(x)$ based on $p(z)$ is defined as

$$\mathtt{t}p : \mathbb{R}^+ \to \mathbb{R}^+ : x \mapsto \bigoplus_{i=0}^{d} |p_i| x^i = \max_{0 \leq i \leq d} |p_i| x^i. \tag{3.1}$$

If $p_0 = p_1 = \cdots = p_{m-1} = 0$, then zero is a tropical root of $\mathtt{t}p(x)$ with multiplicity $m_0 = m$ ($m_0 = 0$ is $p_0 \neq 0$). The (nonzero) tropical roots are points x in $\mathbb{R}^+$ at which the maximum in (3.1) is attained for at least two values of i for this specific x-values [1]. Since $p_d \neq 0$, there are $t \leq d$ distinct tropical roots

$$0 < \tau_1 < \cdots < \tau_t$$

with τ_ℓ of multiplicity m_ℓ, and $\sum_{\ell=0}^{t} m_\ell = d$. These tropical roots can be computed from the Newton polygon associated with $p(z)$, i.e., the upper boundary of the convex hull of the set of points $(j, \log|p_j|)$, $j = 0, 1, \ldots, d$ resulting in the points $(k_\ell, \log|p_{k_\ell}|)$, $\ell = 1, \ldots, t$ with

$$k_0 = 0 < k_1 < \cdots < k_{t-1} < k_t = d.$$

The opposites of the slopes of the segments of this upper boundary are the logarithm of the tropical roots. Hence, if $(k_{\ell-1}, \log|p_{k_{\ell-1}}|)$ and $(k_\ell, \log|p_{k_\ell}|)$ are the two endpoints of such a segment, it follows that

$$\tau_\ell = \left(\frac{|p_{k_{\ell-1}}|}{|p_{k_\ell}|} \right)^{1/m_\ell}, \quad m_\ell = k_\ell - k_{\ell-1}, \quad \ell = 1, 2, \ldots, t. \tag{3.2}$$

These tropical roots can be computed in $\mathcal{O}(d)$ operations [6, Prop. 1]. For more details on tropical roots and corresponding applications in polynomial eigenvalue problems, we refer the interested reader to [4, 6, 7, 10, 12, 14].

When computing the zeros of a polynomial $p(z)$, we assume that $p_0 \neq 0$, i.e., the possible zero roots are already divided out exactly. Hence, all tropical roots τ_ℓ will be nonzero.

Theorem 1. *Given a polynomial $p(z) = \sum_{i=0}^{d} p_i z^i$ and its associated tropical polynomial $\mathtt{t}p(x) = \max_{0\le i\le d} |p_i|x^i = \max_{0\le i\le t} |p_{k_i}|x^{k_i}$, with $0 = k_0 < k_1 < \cdots < k_t = d$ and tropical roots τ_ℓ, $\ell = 1, 2, \ldots, t$ as in (3.2), we have that, for each $i \in \{1, \ldots, d\}$ and corresponding ℓ such that $k_{\ell-1} \le i \le k_\ell$, the parameters γ_i in (2.13) are given by*

$$\gamma_i = \begin{cases} \dfrac{\tau_\ell^{k_{\ell-1}}|p_{k_{\ell-1}}|}{\tau_\ell^i |p_i|} = \dfrac{\tau_\ell^{k_\ell}|p_{k_\ell}|}{\tau_\ell^i|p_i|} & \textit{if } p_i \neq 0, \\ \dfrac{\tau_\ell^{k_{\ell-1}}|p_{k_{\ell-1}}|}{\tau_\ell^i} = \dfrac{\tau_\ell^{k_\ell}|p_{k_\ell}|}{\tau_\ell^i} & \textit{if } p_i = 0. \end{cases}$$

Proof. The tropical polynomial $\mathtt{t}p(x) = \max_{0\le i\le t} |p_{k_i}|x^{k_i}$ with $x \in \mathbb{R}^+$ is a convex piecewise affine function such that

$$\mathtt{t}p(x) = \begin{cases} |p_{k_0}|x^{k_0} = |p_0| & \text{for } x \le \tau_0, \\ |p_{k_\ell}|x^{k_\ell} & \text{for } \tau_{\ell-1} \le x \le \tau_\ell,\ 1 \le \ell \le t\ , \\ |p_{k_t}|x^{k_t} = |p_d|x^d & \text{for } x \ge \tau_t. \end{cases}$$

Now if $p_i \neq 0$, then by (2.13), $\gamma_i = \min_{|z|>0} \beta_i(z)$ with $\beta_i(z) = \max_j |p_j z^j|/|p_i z^i|$. Hence,

$$\beta_i(\tau_\ell) = \frac{\max_j |p_j \tau_\ell^j|}{|p_i \tau_\ell^i|} = \frac{\mathtt{t}p(\tau_\ell)}{|p_i|\tau_\ell^i} = \frac{|p_{k_{\ell-1}}|\tau_\ell^{k_{\ell-1}}}{|p_i|\tau_\ell^i} = \frac{|p_{k_\ell}|\tau_\ell^{k_\ell}}{|p_i|\tau_\ell^i}.$$

Suppose that $\tau_{\ell'-1} \le |z| \le \tau_{\ell'} \le \tau_\ell$, then we get

$$\begin{aligned} \beta_i(z) &= \frac{\max_j |p_j z^j|}{|p_i z^i|} = \frac{\mathtt{t}p(|z|)}{|p_i z^i|} = \frac{|p_{k_{\ell'-1}} z^{k_{\ell'-1}}|}{|p_i z^i|} \ge \frac{|p_{k_{\ell-1}} z^{k_{\ell-1}}|}{|p_i z^i|} = \frac{|p_{k_{\ell-1}}|}{|p_i|} \frac{1}{|z|^{i-k_{\ell-1}}} \\ &\ge \frac{|p_{k_{\ell-1}}|}{|p_i|} \frac{1}{\tau_\ell^{i-k_{\ell-1}}} = \beta_i(\tau_\ell). \end{aligned}$$

A similar argument can be followed when $\tau_\ell \le \tau_{\ell'} \le |z| \le \tau_{\ell'+1}$ to prove that also in this case $\beta_i(z) \ge \beta_i(\tau_\ell)$. Hence, the minimum value for $\beta_i(z)$ is for $z = \tau_\ell$. This gives us the expression for γ_i. The expression when $p_i = 0$ is proved in a similar way. □

When $p_0 \neq 0$, there are no zero tropical roots and no indices k_ℓ from the Newton polygon such that $p_{k_\ell} = 0$ so $\gamma_i > 0$ for all i. When $p_i \neq 0$, it follows from Theorem 1 that for $\widetilde{\gamma}_i$ in (2.14), we have that $\widetilde{\gamma}_i = \gamma_i|p_i| = \tau_\ell^{k_{\ell-1}-i}|p_{k_{\ell-1}}|$ and hence, on using (3.2) and taking logarithm, we find that

$$\log \gamma_i + \log |p_i| = \frac{i - k_{\ell-1}}{k_\ell - k_{\ell-1}}(\log |p_{k_\ell}| - \log |p_{k_{\ell-1}}|) + \log |p_{k_{\ell-1}}|,$$

that is, $(i, \log \gamma_i + \log |p_i|)$ is a point on the segment between $(k_{\ell-1}, \log |p_{k_{\ell-1}}|)$ and $(k_\ell, \log |p_{k_\ell}|)$ or, in other words, it is a point on the upper boundary of the convex hull of the set of points $(j, \log |p_j|)$, $j = 0, 1, \ldots, d$. So $\log \gamma_i$ represents the distance between this point and the point $(i, \log |p_i|)$. This is illustrated in Figure 3.1 using a log scale.

Note that $\gamma_{k_\ell} = 1$, $\ell = 0, \ldots, t$, which in the log scale on Figure 3.1 corresponds to zero. The min-max backward error is equal to ϵ when each absolute error $|\Delta p_i|$ is ϵ

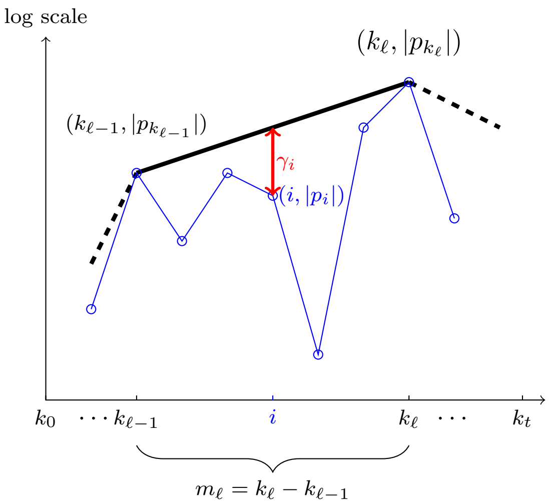


Figure 3.1: The factor γ_i as fraction of a point on the convex hull and the modulus of the corresponding polynomial coefficient $|p_i|$.

times smaller than the convex hull and when there is one or more of these absolute errors just ϵ times smaller. In Figure 3.2 (left) this is shown when the value $|p_i|$ is between the corresponding point on the convex hull and ϵ times smaller. Here Δp_i is very small so $p_i \approx \widetilde{p}_i$. Figure 3.2 (right) illustrates the case when $|p_{i'}|$ is less than ϵ times the corresponding point on the convex hull. Here p_i' is very small so $\Delta p_i' = \widetilde{p}_i'$.

Instead of computing the min-max elementwise backward error

$$\eta_{\widetilde{\boldsymbol{\gamma}}}^{\text{elem}}(\widehat{\mathbf{z}}) = \min_{\mu \in \mathbb{C} \setminus \{0\}} \max_{0 \le i \le d} \frac{|p_i - \mu \widetilde{p}_i|}{\widetilde{\gamma}_i},$$

it is easier to compute the upper bound $\eta_{\widetilde{\boldsymbol{\gamma}}}^{\text{elem}}$ given by

$$\eta_{\widetilde{\boldsymbol{\gamma}}}^{\text{elem}} = \max_{0 < i \le d} \frac{|p_i - \widetilde{p}_i|}{\widetilde{\gamma}_i} \tag{3.3}$$

for which the coefficients $\widetilde{p}_i$ are obtained by constructing $\widetilde{p}(z)$ in (2.11) using multiple precision.

Example 2 (Mastronardi and Van Dooren's example [9])**.** *Let u denote the unit roundoff and consider the polynomial*

$$p(z) = z^2 - 2\beta z - 1$$

with zeros $z_{1,2} = \beta \pm \sqrt{\beta^2 + 1}$, where $\beta = 2^{-t} + 2^{-2t}$ with $2^{-2t} \le u/2$ and $2^{-t} \approx \sqrt{u}$. These zeros are well conditioned. The approximations $\widehat{z}_{1,2} = 2^{-t} \pm 1$ to the zeros $z_{1,2}$ both have a relative error of order u. These approximations are the exact zeros of the polynomial

$$\widetilde{p}(z) = z^2 - 2^{-t+1} z + 2^{-2t} - 1.$$

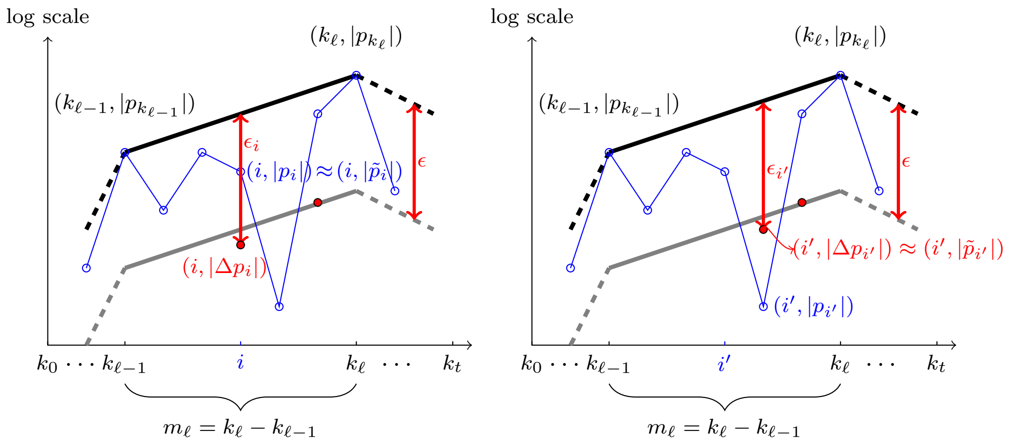


Figure 3.2: The min-max backward error is ϵ. On the left, the value of $|p_i|$ is within the band between the convex hull and ϵ times the convex hull. On the right, the value of $|p_{i'}|$ is less than ϵ times the convex hull

For the polynomial $p(z)$, it is easy to check that

$$\gamma_0 = \gamma_2 = 1 \quad \textit{and} \quad \gamma_1 = (2\beta)^{-1} = \mathcal{O}(u^{-1/2}) \gg 1.$$

This results in the backward errors

$$\eta^{\text{elem}}_{|\mathbf{p}|}(\widehat{\mathbf{z}}) = \min_{\mu \in \mathbb{R}} \max_{0 \le i \le 2} \frac{|\mu \widetilde{p}_i - p_i|}{|p_i|} \approx u^{1/2}, \quad \eta^{\text{elem}}_{\widetilde{\boldsymbol{\gamma}}}(\widehat{\mathbf{z}}) = \min_{\mu \in \mathbb{R}} \max_{0 \le i \le 2} \frac{|\mu \widetilde{p}_i - p_i|}{|p_i| \gamma_i} = 2^{-2t+1} \le u.$$

Other approximations $\widehat{z}_1$ and $\widehat{z}_2$ of the zeros with a relative error of the size of the unit roundoff lead to similar results.

4. A root solver based on companion linearization

In this section we describe an algorithm for computing all the zeros of a polynomial $p(z) = \sum_{i=0}^{d} p_i z^i$. It is based on a companion linearization of the grade $d+1$ polynomial $0 \cdot z^{d+1} + p(z)$, a suitable scaling/balancing of the linearization, and the use of a QZ algorithm with an appropriate deflation strategy for the eigenvalues at infinity. The min-max elementwise backward error of the resulting algorithm is studied in Section 5. Note that, at this point, we do not see a way to perform a scaling/balancing on a companion linearization of $p(z)$ that would result in a pencil with one of the matrix well-balanced and the other graded when there are large variations in the magnitude of the tropical roots. These properties being crucial when applying the QZ algorithm.

We transform the original problem, i.e., $p(z) = 0$ into the generalized eigenvalue problem

$$\begin{bmatrix} p_d & p_{d-1} & \cdots & p_1 & p_0 \\ 1 & -z & & & \\ & 1 & -z & & \\ & & \ddots & \ddots & \\ & & & 1 & -z \end{bmatrix} \begin{bmatrix} z^d \\ z^{d-1} \\ \vdots \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} p(z) \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

The pencil on the left of the equality, which we write $C(z) = A - zB$ with

$$A = \begin{bmatrix} p_d & p_{d-1} & \cdots & p_1 & p_0 \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \tag{4.1}$$

is the $(d+1) \times (d+1)$ companion linearization of $0 \cdot z^{d+1} + p(z)$. It has an eigenvalue at infinity and its finite eigenvalues are the roots of $p(z)$. We then apply a two-sided diagonal scaling to the pencil $C(z)$,

$$\widehat{C}(z) = D_l C(z) D_r = \widehat{A} - z\widehat{B}$$

with diagonal matrices D_l, D_r constructed such that $\widehat{A}$ is balanced in the sense that its nonzero entries are in modulus bounded by 1 and the diagonal of $\widehat{B}$ is graded. This is done as follows. We define

$$\widetilde{\tau}_1, \widetilde{\tau}_2, \ldots, \widetilde{\tau}_d := \underbrace{\tau_1, \ldots, \tau_1}_{m_1 \text{ times}}, \underbrace{\tau_2, \ldots, \tau_2}_{m_2 \text{ times}}, \ldots, \underbrace{\tau_t, \ldots, \tau_t}_{m_t \text{ times}} \tag{4.2}$$

with τ_ℓ as in (3.2) and construct

$$D_l = \operatorname{diag}\left(\frac{1}{|p_d|}, 1, \prod_{j=d}^{d} \widetilde{\tau}_j, \prod_{j=d-1}^{d} \widetilde{\tau}_j, \ldots, \prod_{j=2}^{d} \widetilde{\tau}_j \right) \tag{4.3}$$

and

$$D_r = \operatorname{diag}\left(1, 1/\prod_{j=d}^{d} \widetilde{\tau}_j, 1/\prod_{j=d-1}^{d} \widetilde{\tau}_j, \ldots, 1/\prod_{j=1}^{d} \widetilde{\tau}_j \right). \tag{4.4}$$

The matrices of the scaled pencil $\widehat{C}(z) = \widehat{A} - z\widehat{B}$ have the form

$$\widehat{A} = \begin{bmatrix} \widehat{a}_d & \widehat{a}_{d-1} & \cdots & \widehat{a}_1 & \widehat{a}_0 \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{bmatrix}, \qquad \widehat{B} = \begin{bmatrix} 0 & & & & \\ & \widehat{b}_1 & & & \\ & & \widehat{b}_2 & & \\ & & & \ddots & \\ & & & & \widehat{b}_d \end{bmatrix} \tag{4.5}$$

with

$$|\widehat{a}_i| = \begin{cases} \gamma_i^{-1} & \text{if } p_i \neq 0, \\ 0 & \text{if } p_i = 0, \end{cases} \qquad |\widehat{b}_i| = \widetilde{\tau}_{d-i+1}^{-1}, \tag{4.6}$$

and γ_i as in Theorem 1. Now the parameters γ_{k_ℓ}, $\ell = 1, 2, \ldots, t$, corresponding to the indices of the Newton polytope are equal to one. The other γ_i values are larger than one, i.e., γ_i^{-1} is smaller than one for these other values. Hence, all elements of the top row of the transformed matrix $\widehat{A}$ are less than or equal to one in modulus. The nonzero diagonal part of $\widehat{B}$ contains the inverses of the tropical roots and, since $\widetilde{\tau}_j \le \widetilde{\tau}_{j+1}$, we have that $\widehat{b}_j \le \widehat{b}_{j+1}$. Note that once the tropical roots are computed, the factors γ_i naturally appear as inverses of the modulus of the corresponding elements $\widehat{a}_i$ when $p_i \neq 0$.

The trivial eigenvalue at infinity is easily deflated: let G be a 2×2 Givens rotation such that $G\left[\begin{smallmatrix}\widehat{a}_d\\ 1\end{smallmatrix}\right] = \left[\begin{smallmatrix}\breve{a}_d\\ 0\end{smallmatrix}\right]$ and embed it in I_{d+1} as the 2×2 leading block to form $\widetilde{G}$. Then

$$\widetilde{A} = \widetilde{G}\widehat{A} = \begin{bmatrix} \breve{a}_d & \breve{a}_{d-1} & \cdots & \breve{a}_1 & \breve{a}_0 \\ 0 & \widetilde{a}_{d-1} & \cdots & \widetilde{a}_1 & \widetilde{a}_0 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{bmatrix}, \qquad \widetilde{B} = \widetilde{G}\widehat{B} = \begin{bmatrix} 0 & \breve{b}_1 & & & \\ & \widetilde{b}_1 & & & \\ & & \widehat{b}_2 & & \\ & & & \ddots & \\ & & & & \widehat{b}_d \end{bmatrix}. \tag{4.7}$$

The roots of $p(z)$ are the d (finite) eigenvalues of the trailing $d \times d$ submatrix of $\widetilde{A} - z\widetilde{B}$. The $d \times d$ trailing submatrix of $\widetilde{A}$ remains well-balanced with entries in the first row that are of modulus less or equal to $\sqrt{2}/2$. The diagonal entries of $\widetilde{B}$ remained ordered with increasing moduli.

Large differences in the magnitude of the entries of $\widetilde{B}$ generically lead to a large difference in the magnitude of the eigenvalues of $\widetilde{A} - z\widetilde{B}$. As already observed in [14, Section 3], when the difference in the entries of $\widetilde{B}$ is larger than $\epsilon_{\text{mach}}^{-1}$, ϵ_{mach} being the machine precision, the LAPACK implementation of the QZ algorithm may decide too quickly to deflate an eigenvalue and declare it to be at infinity. So we slightly modify the LAPACK routines `xHGEQZ` such that besides the trivial eigenvalues at infinity only finite eigenvalues are generated. Note that the latter can be very large when they correspond to exact infinite eigenvalues. To be more specific, we replace the value of `BTOL` by the smallest positive nonzero floating point number in strategic places in the fortran code as to avoid that a specific entry of $\widetilde{B}$ is explicitly set to zero, thereby leading to a computed infinite eigenvalue. At the same time, we also increase the maximum number of iterations MAXIT. The steps of our approach are summarized in Algorithm 1.

Algorithm 1 Computes all zeros of a polynomial $p(z) = \sum_{i=0}^{d} p_i z^i$

```
Input: the coefficients $p_i$, $i = 0, 1, \ldots, d$ of $p(z)$.
Output: the $d$ zeros of $p(z)$.

Construct the companion pencil $C(z) = A - zB$ of $p(z)$ as in (4.1).
Compute the tropical roots $\widetilde{\tau}_j$, $j = 1, 2, \ldots, d$ as in (4.2).
Scale/balance the companion pencil: $\widehat{C}(z) = D_l C(z) D_r$ with $D_l, D_r$ as in (4.3)–(4.4).
Deflate the eigenvalue at infinity with a Givens rotation as in (4.7).
Compute the eigenvalues of the deflated pencil using the QZ algorithm implemented
with a strict deflation criterion for the detection of eigenvalues at infinity.
Return these eigenvalues as roots of $p(z)$.
```

In Section 5, we show that, under certain assumptions on the graded character of the backward error for the generalized Schur form obtained after applying a QZ algorithm with strict deflation criterion for eigenvalues at infinity, Algorithm 1 is min-max elementwise backward stable according to Definition 1. In Section 6 several numerical experiments will be given illustrating the backward stable behaviour of the newly designed algorithm.

5. Min-max backward error for Algorithm 1

In [14, Section 6] we gave numerical evidence for the following assumption that is required for our backward error analysis of Algorithm 1.

Assumption 1. *The QZ algorithm with a strict deflation criterion for eigenvalues at infinity applied to $\widetilde{G}(\widehat{A} - z\widehat{B})$ in* (4.7) *computes the exact generalized Schur form of the matrix pencil*

$$(\widehat{A} + \Delta\widehat{A}) - z(\widehat{B} + \Delta\widehat{B}),$$

where all entries in $\Delta\widehat{A}$ have modulus of size $\mathcal{O}(\epsilon_{mach})$, ϵ_{mach} being the machine precision, and the entries in column j of $\Delta\widehat{B}$ have modulus of size $\mathcal{O}(\widetilde{\tau}_{d-j+2}^{-1}\epsilon_{mach})$, with $\widetilde{\tau}_j$ as in (4.2)*, except for the first column, which is equal to zero.*

It follows then that, under Assumption 1, the backward error on the matrix $\widehat{B}$ has a graded structure if $\widehat{B}$ is graded, since $\widetilde{\tau}_1 \leq \widetilde{\tau}_2 \leq \cdots \leq \widetilde{\tau}_d$.

Theorem 2. *Algorithm 1 applied to $p(z) = \sum_{i=0}^{d} p_i z^i$ is min-max elementwise backward stable under Assumption 1, that is, it computes roots $\widehat{\mathbf{z}} = [\widehat{z}_1, \ldots, \widehat{z}_d]^T$ of $p(z)$ with min-max elementwise backward error $\eta_{\widetilde{\boldsymbol{\gamma}}}^{\mathrm{elem}}(\widehat{\mathbf{z}}) = \mathcal{O}(\epsilon_{mach})$.*

Proof. To prove this theorem, we transform the matrix pencil $(\widehat{A} + \Delta\widehat{A}) - z(\widehat{B} + \Delta\widehat{B})$ into

$$P\left[(\widehat{A} + \Delta\widehat{A}) - z(\widehat{B} + \Delta\widehat{B})\right]Q = (\widehat{A} + \Delta\widehat{A}') - z\widehat{B} \tag{5.1}$$

with P and Q nonsingular such that the resulting error is fully concentrated on the first row of $\widehat{A}$, i.e., $\Delta\widehat{A}'$ is zero except possibly in its first row. We then show that the absolute error on each of the elements $\widehat{a}_i$ in the first row is of order $\mathcal{O}(\epsilon_{\mathrm{mach}})$. Finally, we show that performing the inverse of the original scaling/balancing operation, i.e., $D_l^{-1}\Delta\widehat{A}'D_r^{-1}$ with D_l, D_r as in (4.3)–(4.4) leads to a min-max backward error of size $\mathcal{O}(\epsilon_{\mathrm{mach}})$.

To show how to restore $(\widehat{A} + \Delta\widehat{A}) - z(\widehat{B} + \Delta\widehat{B})$ into $\widehat{A} + \Delta\widehat{A}' - z\widehat{B}$ with $\widehat{A}$ and $\widehat{B}$ as in (4.6), and $\Delta\widehat{A}' = e_1\Delta\widehat{a}^T$ with $\Delta\widehat{a} \in \mathbb{C}^{d+1}$, we use the 4×4 pencil

$$\begin{bmatrix} \widehat{a}_3 + \Delta\widehat{a}_3' & \widehat{a}_2 + \Delta\widehat{a}_2' & \widehat{a}_1 + \Delta\widehat{a}_1' & \widehat{a}_0 + \Delta\widehat{a}_0' \\ 1^{(8)} & 0^{(6)} & 0^{(4)} & 0^{(2)} \\ 0^{(9)} & 1^{(6)} & 0^{(4)} & 0^{(2)} \\ 0^{(9)} & 0^{(11)} & 1^{(4)} & 0^{(2)} \end{bmatrix} - z \begin{bmatrix} 0 & 0^{(1)} & 0^{(1)} & 0^{(1)} \\ 0 & \widehat{b}_1^{(7)} & 0^{(5)} & 0^{(3)} \\ 0 & 0^{(10)} & \widehat{b}_2^{(5)} & 0^{(3)} \\ 0 & 0^{(10)} & 0^{(12)} & \widehat{b}_3^{(3)} \end{bmatrix}$$

and the same notation as in [15, Section 4] to indicate the order in which the elements are restored. The construction together with the error analysis we present below extend to arbitrary dimension in the obvious way. We refer to the constant matrix coefficient in the above pencil as the "first matrix" and to the other matrix as the "second matrix". Recall that by construction, $|\widehat{a}_j| \leq 1$, $j = 0, \ldots, d$, and $|\widehat{b}_i| = \widetilde{\tau}_{d-i+1}^{-1}$, $i = 1, \ldots, d$ (here $d = 3$), and that $\widetilde{\tau}_1 \leq \widetilde{\tau}_2 \leq \cdots \leq \widetilde{\tau}_d$.

- Step 1 introduces zeros in the second matrix in positions $(1, j)$, $j = 2, \dots, 4$ using a Gauss transform applied to the left as follows. Partition the first and second matrices as
$$\begin{bmatrix} a^T + \Delta a^T & \widehat{a}_0 + \Delta\widehat{a}_0 \\ I + \Delta I & \Delta o \end{bmatrix}, \qquad \begin{bmatrix} 0 & b^T \\ 0 & D + \Delta C \end{bmatrix}, \tag{5.2}$$
where by Assumption 1 $\Delta a^T, \Delta\widehat{a}_0, \Delta I, \Delta o$ have entries of order $\mathcal{O}(\epsilon_{\text{mach}})$, $D = \operatorname{diag}(\widehat{b}_1, \widehat{b}_2, \widehat{b}_3)$ is nonsingular, and the entries in column j of ΔC have modulus of size $\mathcal{O}(\widetilde{\tau}_{4-j}^{-1}\epsilon_{\text{mach}})$ (so $D + \Delta C$ is nonsingular). Then the Gauss transform $I - e_1 v^T$ with $v^T = \begin{bmatrix} 0 & \tilde{v}^T \end{bmatrix}$ and $\tilde{v}^T = b^T(D + \Delta C)^{-1}$ introduces the desired zeros in the first row of the second matrix. It also introduces perturbations in the first row of the first matrix,
$$(I - e_1 v^T) \begin{bmatrix} a^T + \Delta a^T & \widehat{a}_0 + \Delta\widehat{a}_0 \\ I + \Delta I & \Delta o \end{bmatrix} = \begin{bmatrix} a^T + \Delta a^T - \tilde{v}^T - \tilde{v}^T \Delta I & \widehat{a}_0 + \Delta\widehat{a}_0 - \tilde{v}^T \Delta o \\ I + \Delta I & \Delta o \end{bmatrix}.$$
The modulus of each entry of $\tilde{v} = b^T(D + \Delta C)^{-1} = b^T D^{-1}(I + \Delta C D^{-1})^{-1}$ has size $\mathcal{O}(\epsilon_{\text{mach}})$. This is because the modulus of the entries of $b^T D^{-1}$ and $\Delta C D^{-1}$ are of size $\mathcal{O}(\epsilon_{\text{mach}})$, and since $\|\Delta C D^{-1}\|_1 < 1$ (as long as $d\epsilon_{\text{mach}} < 1$), $(I + \Delta C D^{-1})^{-1} = I - \Delta C D^{-1} + (\Delta C D^{-1})^2 + \cdots$. So step 1 does not increase the size of the perturbations on $\widehat{A}$ and $\widehat{B}$.

- Step 2 introduces zeros in positions $(i, 4)$, $i = 2, 3, 4$ of the first matrix, which still has the form (5.2) with $\Delta a, \Delta\widehat{a}_0, \Delta I$ and Δo having entries of magnitude $\mathcal{O}(\epsilon_{\text{mach}})$ while maintaining the zeros already introduced. This is done by applying the Gauss transform $I - v e_4^T$ to the right of the pencil with $v = \left[\begin{smallmatrix} \tilde{v} \\ 0 \end{smallmatrix}\right]$, $\tilde{v} = (I + \Delta I)^{-1} \Delta o$ leads to
$$\begin{bmatrix} a^T + \Delta a^T & \widehat{a}_0 + \Delta\widehat{a}_0 \\ I + \Delta I & \Delta o \end{bmatrix} \left(I - \begin{bmatrix} 0 & \tilde{v} \\ 0 & 0 \end{bmatrix} \right) = \begin{bmatrix} a^T + \Delta a^T & \widehat{a}_0 + \Delta\widehat{a}_0 - (a^T + \Delta a^T)\tilde{v} \\ I + \Delta I & 0 \end{bmatrix}.$$
But the vector a has entries of modulus bounded by one and the entries of $\tilde{v}$ all have modulus of size $\mathcal{O}(\epsilon_{\text{mach}})$, so step 2 does not increase the size of the perturbation on the first matrix.

  For the second matrix, we have
$$\begin{bmatrix} 0 & 0 \\ 0 & D + \Delta C \end{bmatrix} \left(I - \begin{bmatrix} 0 & \tilde{v} \\ 0 & 0 \end{bmatrix} \right) = \begin{bmatrix} 0 & 0 \\ 0 & D + \Delta C + u e_3^T \end{bmatrix},$$
where the entries in the last column of ΔC have modulus of size $\mathcal{O}(\widetilde{\tau}_1^{-1}\epsilon_{\text{mach}})$ and u has entries of modulus no larger than $(\widetilde{\tau}_2^{-1} + \widetilde{\tau}_3^{-1})\epsilon_{\text{mach}}$ so the perturbation $u e_3^T$ does not change the graded nature of the second matrix.

- Step 3 uses elementary row operations to introduce zeros in positions $(i, 4)$, $i = 2, 3$, of the second matrix. This is done by replacing row i with row i plus a scalar multiple of the last row. The scalar multiples are of size $\mathcal{O}(\epsilon_{\text{mach}})$ and with arguments similar to those in steps 1 & 2, these elementary row operations add perturbations of order at most $\mathcal{O}(\epsilon_{\text{mach}})$ to rows 2 and 3 of the first matrix and preserve the graded structure of the second matrix. Finally, the latter has $(4, 4)$

entry equal to $\widehat{b}_3 + \Delta\widehat{b}_3$ with $|\Delta\widehat{b}_3| = \mathcal{O}(\widetilde{\tau}_1^{-1}\epsilon_{\mathrm{mach}})$ that is scaled to $\widehat{b}_3$ with a diagonal scaling of the form $I + \alpha e_4 e_4^T$, $|\alpha| = \mathcal{O}(\epsilon_{\mathrm{mach}})$, applied to the right of the pencil. This results in the $(1, 4)$ entry of the first matrix to be multiplied by $1 + \alpha$ but since $|\widehat{a}_0| \leq 1$, this entry remains of the form $\widehat{a}_0 + \Delta\widehat{a}_0$ with $|\Delta\widehat{a}_0| = \mathcal{O}(\epsilon_{\mathrm{mach}})$.

- Step 4 introduces zeros in positions $(i, 3)$, $i = 2, 3$, of the first matrix using a Gauss transform of the form $I - ve_3^T$ applied to the right of the pencil and with vector v having entries of size at most $\mathcal{O}(\epsilon_{\mathrm{mach}})$. With arguments similar to those used in step 2, we find that this transformation does not increase the size of the perturbations present in columns $1, 2, 3$ of the first matrix and does not affect the graded structure of the second matrix. A diagonal scaling applied to the right of the pencil is then used to set the $(4, 3)$ entry of the first matrix, which is of the form $1 + \delta$, $|\delta| = \mathcal{O}(\epsilon_{\mathrm{mach}})$ to 1. This scaling does not affect the size of the perturbations present in the third column of both matrices since $(1 + \delta)^{-1} = 1 + \tilde{\delta}$ with $|\tilde{\delta}| = \mathcal{O}(\epsilon_{\mathrm{mach}})$.

This procedure is continued in a similar way in step 5 to step 8 creating the desired structure in the upper part of each column without destroying the zero-structure that was already introduced nor increasing the size of the perturbation of the entries in the first row of the first matrix or destroying the graded structure of the second matrix. Step 9 to step 12 generate the desired structure in the lower part of each column. In step 9 elements $(3, 1)$ and $(4, 1)$ of the first matrix are made equal to zero by subtracting from the third and fourth row a suitable multiple of the second row. In step 10 elements $(3, 2)$ and then $(4, 2)$ of the second matrix are made equal to zero by subtracting from the second column a suitable multiple of the third column for $(3, 2)$ and then the fourth column for $(4, 2)$. In step 11 element $(4, 2)$ of the first matrix is made equal to zero by subtracting from the fourth row a suitable multiple of the third row. In step 12 element $(4, 3)$ of the second matrix is made equal to zero by subtracting from the third column a suitable multiple of the fourth column resulting in the desired structure with $|\Delta\widehat{a}_i'| = \mathcal{O}(\epsilon_{\mathrm{mach}})$, $i = 0, \ldots, d$.

Reversing the scaling/balancing operation leads to an absolute error Δp_i on the ith initial coefficient p_i of $p(z)$ of size

$$|\Delta p_i| = |p_d||\Delta\widehat{a}_i'| \prod_{j=i+1}^{d} \widetilde{\tau}_j \tag{5.3}$$

$$\overset{(4.2)}{=} |p_d||\Delta\widehat{a}_i'| \underbrace{\tau_\ell, \ldots, \tau_\ell}_{(k_\ell - i)\text{ times}} , \underbrace{\tau_{\ell+1}, \ldots, \tau_{\ell+1}}_{m_{\ell+1}\text{ times}}, \ldots, \underbrace{\tau_t, \ldots, \tau_t}_{m_t\text{ times}} \tag{5.4}$$

$$\overset{(3.2)}{=} |p_d||\Delta\widehat{a}_i'|\tau_\ell^{k_\ell - i} \frac{|p_{k_\ell}|}{|p_{k_{\ell+1}}|} \frac{|p_{k_{\ell+1}}|}{|p_{k_{\ell+2}}|} \cdots \frac{|p_{k_{t-1}}|}{|p_{k_t}|} \tag{5.5}$$

$$= |p_d||\Delta\widehat{a}_i'|\tau_\ell^{k_\ell - i} \frac{|p_{k_\ell}|}{|p_d|} = |\Delta\widehat{a}_i'|\tau_\ell^{k_\ell - i}|p_{k_\ell}| \tag{5.6}$$

with $k_{\ell-1} \leq i \leq k_\ell$. On using (2.12), (2.14), Theorem 1, and (5.3) we have that

$$\eta_{\widetilde{\gamma}}^{\mathrm{elem}}(\mathbf{z}) \leq \max_i \frac{|\Delta p_i|}{\widetilde{\gamma}_i} = \max_i |\Delta\widehat{a}_i'| = \mathcal{O}(\epsilon_{\mathrm{mach}}). \qquad \square$$

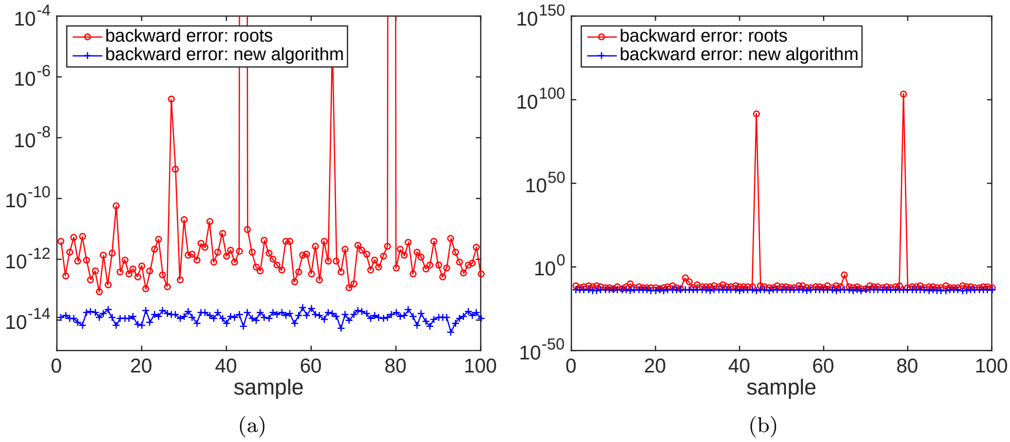


Figure 6.1: Min-max backward errors for the zeros computed by the MATLAB `roots` function and the new algorithm for Experiment 1.

6. Numerical experiments

In section 6.1, the MATLAB `roots` function is compared to our new algorithm (i.e., Algorithm 1). In section 6.2, we generalize our approach to polynomial eigenvalue problems (PEVPs) and compare the resulting algorithm to other polynomial eigensolvers. In all our numerical experiments, we observed that Assumption 1 holds.

6.1. Scalar polynomials

The backward error measured in all the experiments of this section is the upper bound $\eta_{\widetilde{\gamma}}^{\text{elem}}$ in (3.3) on the min-max elementwise backward error.

Experiment 1. We take 100 samples of a polynomial of degree 50. Each polynomial has random complex zeros generated as follows: the multiplicity is 1, the modulus is 10^e with e uniformly random between -20 and $+20$, and the argument is uniformly random between 0 and 2π. Figure 6.1(a) shows the backward errors (less than 10^{-4}) for the zeros returned by `roots` and by the new algorithm, while Figure 6.1(b) also shows the larger backward errors.

For sample number 44, we compare in Figure 6.2 the modulus of the coefficients of $p(z)$ to the modulus of the coefficients of the polynomial $\widetilde{p}(z) = p_d \prod_{k=1}^{50}(z - \widehat{z}_k)$ constructed from the zeros $\widehat{z}_k$, $k = 1, \ldots, 50$ returned by `roots` and by the new algorithm. The zeros returned by `roots` do not reproduce the first coefficients of the polynomial with a high relative accuracy, leading to a large backward error, as shown in Figure 6.1(b).

Experiment 2. We generate 100 polynomials of degree 30. Each polynomial has random complex zeros computed as follows: the multiplicity is uniformly random between 1 and 30, the modulus is 10^e with e uniformly random between -10 and $+10$, and the argument is uniformly random between 0 and 2π. Figure 6.3 compares the min-max backward error between `roots` and our new algorithm.

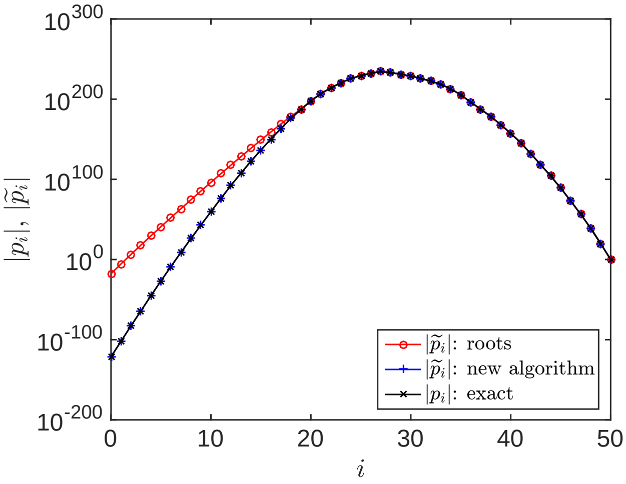


Figure 6.2: Plot of $|p_i|$ and $|\widetilde{p}_i|$, where $\widetilde{p}(z) = p_d \prod_{k=1}^{50}(z - \widehat{z}_k)$ is constructed from the zeros $\widehat{z}_1, \ldots, \widehat{z}_{50}$ computed by `roots` and by the new algorithm for sample 44 of Experiment 1.

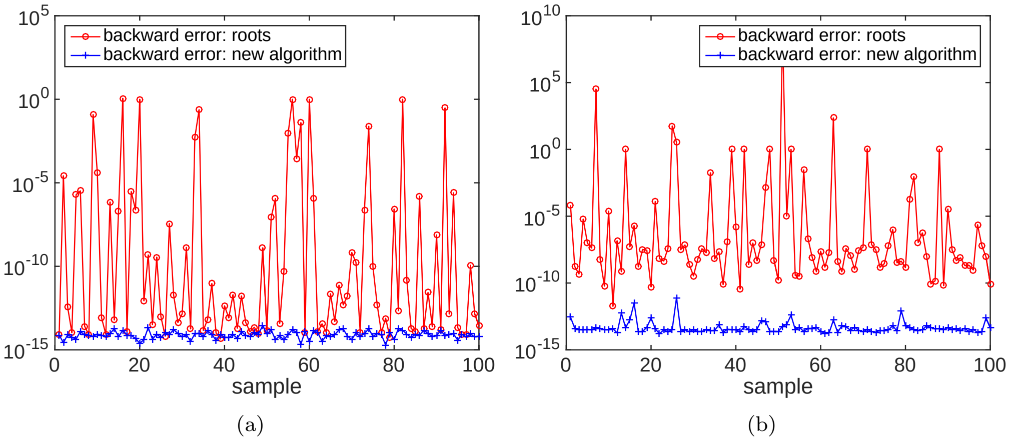
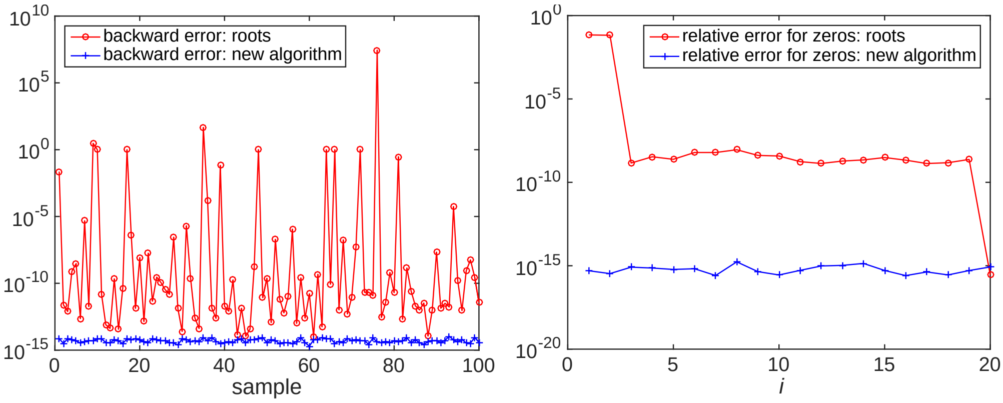


Figure 6.3: Min-max backward error for the zeros computed by MATLAB's `roots` function and the new algorithm for Experiment 2 in (a) and for Experiment 3 in (b).



(a)

(b)

Figure 6.4: Plot (a): min-max backward error for the zeros computed by MATLAB's `roots` function and the new algorithm for Experiment 4. Plot (b): relative error on the computed zeros by `roots` and the new algorithm for sample 39 of Experiment 4.

Experiment 3. We take 100 samples of a polynomial of degree 100. Each polynomial has random complex coefficients as follows. The modulus is 10^e with e uniformly random between -20 and $+20$ and the argument is uniformly random between 0 and 2π. Figure 6.3(b) shows the backward error.

Experiment 4. The parameters are the same as in Experiment 3 but now the degree is 20 instead of 100. Figure 6.4(a) shows the backward error. Figure 6.4(b) compares the relative errors on the computed zeros by `roots` and the new algorithm for sample 39. For this sample, Figure 6.5 with plot (a) for the new algorithm and plot (b) for `roots`, shows the magnitude of the coefficients of $p(z)$ and $\widetilde{p}(z)$ with $\widetilde{p}_d = p_d$, the absolute errors $|p_i - \widetilde{p}_i|$ and compare them to the convex hull of the set of points $(i, \log|p_i|)$, $i = 0, 1, \dots, d$ as well as the points on this upper boundary multiplied by the machine precision ϵ_{mach}. For our new algorithm the absolute error is not much larger than ϵ_{mach} times the convex hull, indicating that the corresponding backward error $\eta_{\widetilde{\gamma}}^{\text{elem}}$ is of the size of the machine precision ϵ_{mach}. This is not the case for `roots`, in particular, the absolute error on the coefficient p_1 is almost as large as the coefficient itself. This indicates that the backward error $\eta_{\widetilde{\gamma}}^{\text{elem}}$ is of order 1.

6.2. *Generalization to polynomial eigenvalue problems*

Given a matrix polynomial $P(z) = \sum_{i=0}^{d} P_i z^i \in \mathbb{C}[z]^{s\times s}$, the polynomial eigenvalue problem (PEVP) consists of finding scalars λ (eigenvalues) and corresponding nonzero vectors v (eigenvectors) such that

$$P(\lambda)v = 0.$$

Algorithm 1 extends easily from scalar polynomials to matrix polynomials. The entries in the companion form (4.1) are replaced by matrices (i.e., p_i is replaced by P_i, 1 by I_s and 0 by the $s \times s$ identity matrix) to obtain a $ds \times ds$ block companion linearization $C(\lambda)$ for the grade $d+1$ matrix polynomial $0z^{d+1} + P(z)$. For the two-sided diagonal scaling, we use $D_l \otimes I_s$ and $D_r \otimes I_s$, with D_l and D_r as in (4.3)–(4.4), and $\widetilde{\tau}_i$ as in (4.2).

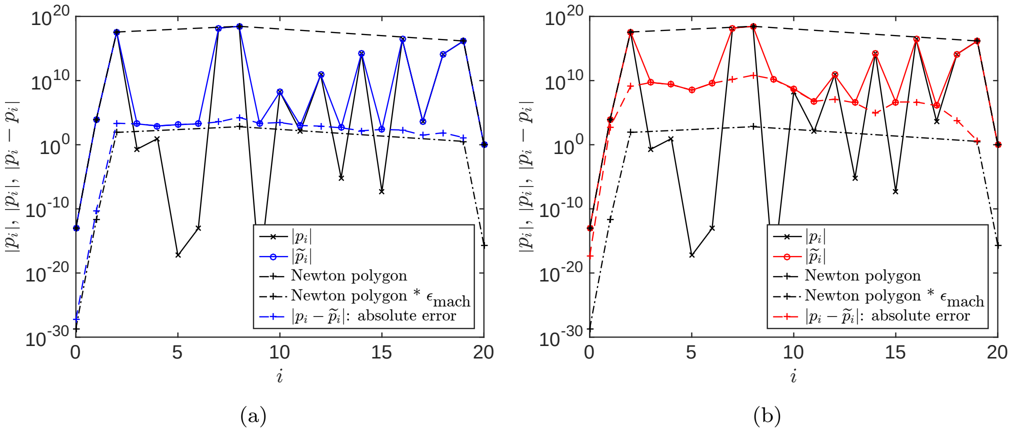


Figure 6.5: Sample 39 of Experiment 4. Magnitude of the coefficients p_i of $p(z)$ and the coefficients $\widetilde{p}_i$ of $\widetilde{p}(z)$ as well as the absolute error $|p_i - \widetilde{p}_i|$. Plot (a) corresponds to coefficients $\widetilde{p}_i$ obtained from zeros computed by the new algorithm, whereas plot (b) corresponds to coefficients $\widetilde{p}_i$ obtained from zeros computed by `roots`. The Newton polygon of the points $(i, \log|p_i|)$, $i = 0, 1, \ldots, d$ is shown as well as the points on this polygon shifted down by a factor ϵ_{mach}.

The positive scalars τ_i, $i = 1, \ldots, t$ with τ_i of multiplicity m_i, are the tropical roots of $\mathtt{t}p(x) = \max_i \|P_i\| x^i$. The resulting block pencil $\widehat{C}(\lambda) = (D_l \otimes I_s) C(\lambda) (D_r \otimes I_s) = \widehat{A} - \lambda \widehat{B}$ is such that $\widehat{A}$ is well-balanced in the sense that the $s \times s$ matrices in the first block row of $\widehat{A}$ have norms less or equal to 1, and $\widehat{B}$ is graded. The deflation of the s extra eigenvalues at infinity is performed by constructing a QR factorization of the first block column of $\widehat{C}(\lambda)$ and by forming $Q^* \widehat{C}(\lambda)$. We can deflate the first s rows and columns of the resulting pencils and call the QZ algorithm together with the strong deflation criterion for eigenvalues at infinity we discussed in Section 4.

We consider the following polynomial eigensolvers:

1. the MATLAB `polyeig` function;
2. `quadeig` from [7] when the degree $d = 2$;
3. Gaubert and Sharify's Algorithm [6, Alg. 1] (see also [10, Alg. 4.1]). We use the same MATLAB implementation as in [10], which we refer to as the `G&S` eigensolver;
4. the polynomial eigensolver based on a tropically scaled Lagrange linearization using well-separated tropical roots described in [14], which we refer to as the `Lagrange` eigensolver;
5. the eigensolver based on scaled block companion pencil as described at the start of Section 6.2, which we refer to as the `new` eigensolver.

The normwise backward error for an approximate eigenvalue $\tilde{\lambda}$ of P can be computed as [13, Lem.3]

$$\eta_P(\tilde{\lambda}) = \frac{\|P(\tilde{\lambda})^{-1}\|_2^{-1}}{\sum_{i=0}^{d} |\tilde{\lambda}|^i \|P_i\|_2} = \frac{\sigma_{\min}(P(\tilde{\lambda}))}{\sum_{i=0}^{d} |\tilde{\lambda}|^i \|P_i\|_2}. \tag{6.1}$$

This backward error is the smallest ϵ such that $\tilde{\lambda}$ is an eigenvalue of $P(\lambda) + \Delta P(\lambda)$, with $\Delta P(z) = \sum_{i=0}^{d} \Delta P_i z^i$, such that $\|\Delta P_i\| \le \epsilon \|P_i\|$, $i = 0, \ldots, d$. This measure of

Table 6.1: Largest backward errors $\eta_P^{\max}$ for eigenvalues computed by the eigensolvers `polyeig`, `quadeig` (for quadratics only), `G&S`, `Lagrange` and `new` on test problems from the NLEVP collection as described in Experiment 5.

Problem	d	s	`polyeig`	`quadeig`	`G&S`	`Lagrange`	`new`
`cd_player`	2	60	**3.1e-10**	2.5e-16	**7.5e-07**	4.1e-16	1.4e-15
`damped_beam`	2	200	**2.6e-11**	2.9e-16	1.2e-16	4.8e-16	6.9e-17
`hospital`	2	24	**2.9e-13**	1.3e-15	1.6e-15	3.9e-15	2.7e-15
`metal_strip`	2	9	**4.1e-14**	6.8e-16	2.7e-16	3.0e-16	3.5e-16
`mirror`	4	9	**2.1e-14**	—	3.7e-17	9.8e-16	5.4e-17
`orr_sommerfeld`	4	64	**9.1e-08**	—	7.1e-15	1.5e-15	1.4e-15
`pdde_stability`	2	225	**1.6e-13**	4.0e-14	1.4e-14	8.8e-14	9.1e-14
`planar_waveguide`	4	129	**4.7e-12**	—	3.2e-14	2.7e-15	1.8e-14
`plasma_drift`	3	128	**2.2e-13**	—	1.3e-14	1.6e-14	**1.0e-13**
`power_plant`	2	8	**5.3e-12**	4.2e-18	3.3e-18	1.3e-16	3.1e-18
`relative_pose_5pt`	3	10	9.6e-18	—	1.1e-16	**2.1e-14**	8.5e-17
`speaker_box`	2	107	**1.7e-13**	6.0e-17	4.1e-17	6.8e-16	8.2e-18
`wiresaw1`	2	10	**1.3e-14**	1.4e-15	1.0e-15	1.0e-15	1.8e-15
`wiresaw2`	2	10	**2.0e-14**	1.9e-15	1.5e-15	9.7e-16	8.3e-16

the backward error for a single eigenvalue can be regarded as an extension to matrix polynomials of the elementwise backward error for an approximate root of a scalar polynomial in (2.1). They have in common the fact that the perturbations are measured coefficientwise. Note that for a min-max normwise backward error, it suffices to replace $\sum_i |\tilde{\lambda}|^i \|P_i\|_2$ in (6.1) with $\max_i |\tilde{\lambda}|^i \|P_i\|_2$. As was shown in (2.9) for a single approximate zero of a scalar polynomial, there is not much difference between these two measures of the backward error. Note that we are not looking at a global measure of the backward error here but, instead, report

$$\eta_P^{\max} = \max\{\eta_P(\tilde{\lambda}) : \tilde{\lambda} \text{ is an eigenvalue of } P\}$$

which is a lower bound of the global backward error for all the computed eigenpairs of P. We consider that all the eigenvalues have been computed with a small backward error if $\eta_P^{\max} \le ds\epsilon_{\text{mach}}$, where for our numerical experiments $\epsilon_{\text{mach}} \approx 2.2 \times 10^{-16}$.

Experiment 5. We consider all square problems from the NLEVP collection [3] with size $s \le 300$ and with $\eta_P^{\max} \le s\epsilon_{\text{mach}}$. The value of $\eta_P^{\max}$ is displayed in Table 6.1 for each polynomial eigensolver under consideration. A backward error $\eta_P^{\max}$ larger than $ds\epsilon_{\text{mach}}$ is highlighted in red and bold. The eigensolvers `G&S`, `Lagrange`, and `new` return eigenvalues with small backward errors for almost all the problems as opposed to `polyeig`. For the `cd_player` problem, the `G&S` eigensolver returns eigenvalues with a large backward error, whereas $\eta_P^{\max} \approx ds\epsilon_{\text{mach}}$ for the `relative_pose_5pt` problem when solved by `Lagrange` and for the `plasma_drift` problem when solved by `new`.

Experiment 6. In [14], we considered several PEVPs with large variations in the magnitude of their eigenvalues (and, hence, also in norm of their matrix coefficients). The backward errors for these problems are provided in Table 6.2, and the backward errors for which $\eta_P^{\max} \ge ds\epsilon_{\text{mach}}$ are highlighted in red and bold. The `Lagrange` and `new` eigen-

Table 6.2: Largest backward errors $\eta_P^{\max}$ for eigenvalues computed by the eigensolvers polyeig, quadeig (for quadratics), G&S, Lagrange and new on test problems used in [14].

Problem	d	s	polyeig	quadeig	G&S	Lagrange	new
Problem 1	7	4	**3.0e-02**	—	**1.7e-08**	1.4e-15	6.3e-16
Problem 2	7	4	**2.7e-01**	—	**6.8e-13**	2.1e-15	5.7e-16
Problem 3	2	4	2.3e-16	2.2e-16	1.1e-16	**3.2e-15**	1.4e-16
Problem 4	2	5	2.2e-16	**1.3e-11**	1.9e-16	3.0e-16	2.3e-16
Problem 5	2	5	4.7e-16	**2.8e-13**	1.0e-16	4.1e-16	3.0e-16
Problem 6	2	2	3.2e-17	3.4e-17	4.0e-18	2.8e-16	4.5e-18
Problem 7	2	10	2.1e-16	**1.3e-02**	1.4e-16	3.2e-16	2.1e-16
Problem 8	2	10	4.9e-15	**3.4e-12**	3.1e-16	3.0e-16	7.8e-16
Problem 9	2	40	**6.8e-07**	2.1e-15	4.0e-16	4.0e-16	2.5e-16
Problem 10	5	20	**3.1e-12**	—	1.4e-15	1.4e-15	1.1e-15
Problem 11	10	8	**2.9e-09**	—	**1.6e-13**	2.3e-15	1.9e-15
Problem 12	4	30	**2.2e-11**	—	**5.5e-14**	9.0e-16	6.5e-15
Problem 13	4	9	**4.6e-12**	—	1.8e-15	**1.1e-14**	**1.2e-14**
Problem 14	4	64	**9.1e-08**	—	7.1e-15	1.5e-15	1.4e-15
Problem 17	10	2	**3.2e-01**	—	**2.8e-01**	**8.1e-13**	8.8e-16
Problem 18	4	4	**1.5e-11**	—	**2.3e-14**	6.9e-16	3.7e-16
Problem 19	4	4	**7.8e-13**	—	**4.4e-14**	7.9e-16	8.7e-17
Problem 20	5	4	**8.7e-03**	—	**1.3e-06**	1.1e-15	1.3e-15
Problem 21	5	4	**3.1e-07**	—	**4.7e-07**	8.9e-16	3.4e-16
Problem 22	4	4	**7.2e-08**	—	**5.5e-12**	8.4e-16	2.2e-16

solvers return eigenvalues with a small backward error for almost all the problems, the backward errors highlighted in red for these two eigensolvers being just slightly larger $ds\epsilon_{\text{mach}}$ (an exception being Lagrange with Problem 17).

7. Conclusions

We introduced a new measure of the backward error for roots of scalar polynomials that is less strict than the elementwise relative backward error but is still meaningful. This new measure allows larger perturbations on the coefficients that do not participate much in the backward error. For this we used an associated max-times polynomial and its tropical roots to determine how much each coefficient can be perturbed. We showed with examples of scalar polynomials with well conditioned zeros that our new backward error can provide an upper bound on the forward error that is sharper than the elementwise relative backward error.

We designed a new algorithm for computing the zeros of scalar polynomials as well as the eigenvalues of matrix polynomials. Our algorithm is based on a companion linearization $A - zB$ of the (matrix) polynomial to which we artificially added a zero leading (matrix) coefficient. In doing so, we found that we could construct a two-sided diagonal scaling that balances A and transforms B into a graded matrix. We observed in [14] that if we use an implementation of the QZ algorithm with a strict deflation criterion for the eigenvalues at infinity on such scaled pencils, then the backward error on the scaled pencil has certain properties: it is of the order of the machine precision for the scaled matrix A and graded for the scaled matrix B. So under the assumption that this observation holds, we proved that our new polynomial root finder is backward stable with respect to the newly defined backward error. Several numerical experiments show the stability of this approach for approximating the zeros of scalar polynomials as well as the eigenvalues of matrix polynomials.

References

[1] M. Akian, R. Bapat, and S. Gaubert. Max-plus algebra. In L. Hogben, editor, *Handbook of Linear Algebra*, pages 35.1–35.14. Chapman and Hall/CRC, Boca Raton, FL, USA, 2014.

[2] J. L. Aurentz, T. Mach, R. Vandebril, and D. S. Watkins. Fast and backward stable computation of roots of polynomials. *SIAM J. Matrix Anal. Appl.*, 36(3):942–973, 2015.

[3] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans. Math. Software*, 39(2):7:1–7:28, 2013.

[4] D. A. Bini, V. Noferini, and M. Sharify. Locating the eigenvalues of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 34(4):1708–1727, 2013.

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

[6] S. Gaubert and M. Sharify. Tropical scaling of polynomial matrices. In *Positive systems*, volume 389 of *Lecture Notes in Control and Information Sciences*, pages 291–303. Springer-Verlag, Berlin, 2009.

[7] S. Hammarling, C. J. Munro, and F. Tisseur. An algorithm for the complete solution of quadratic eigenvalue problems. *ACM Trans. Math. Software*, 39(3):18:1–18:19, Apr. 2013.

[8] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2002.

[9] N. Mastronardi and P. Van Dooren. Revisiting the stability of computing the roots of a quadratic polynomial. *Electron. Trans. Numer. Anal.*, 44:73–82, 2015.

[10] V. Noferini, M. Sharify, and F. Tisseur. Tropical roots as approximations to eigenvalues of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 36(1):138–157, 2015.

[11] W. Oettli and W. Prager. Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. *Numer. Math.*, 6:405–409, 1964.

[12] M. Sharify. *Scaling Algorithms and Tropical Methods in Numerical Matrix Analysis: Application to the Optimal Assignment Problem and to the Accurate Computation of Eigenvalues.* PhD thesis, Ecole Polytechnique, Palaiseau, France, Sept. 2011.

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

[14] M. Van Barel and F. Tisseur. Polynomial eigenvalue solver based on tropically scaled Lagrange linearization. *Linear Algebra Appl.*, 542:186–208, 2018.

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