More on pseudozeros for univariate polynomials

Stef Graillat, Philippe Langlois

Laboratoire MANO, Université de Perpignan
52, avenue Paul Alduy
F-66860 Perpignan, France

Abstract

When polynomials have limited accuracy coefficients or are computed in finite precision, classical algebraic problems such that GCD, primality, divisibility have to be redefined. Such approximate algebraic problems are still challenging open questions in the symbolic computation community. In this paper, we focus on a numerical and graphical tool: the pseudozero set. We show how pseudozeros may provide solutions to some approximate algebraic problems like polynomial stability and primality.

Key words: pseudozeros, polynomial, stability, approximate GCD, approximate primality, finite precision.
1991 MSC: 65F35, 68W30

1 Introduction

Most of real life polynomials have coefficients known to a limited accuracy. Such uncertainty may come from measured or observed data or previous computations performed in finite precision, *i.e.* in floating point arithmetic. Polynomials we consider in this paper suffer from such an uncertainty. Algebraic computation with uncertain polynomials occur in robotic, CAGD [23], molecular biology, etc. Classical polynomial problems like GCD, divisor or primality have to be redefined to take into account the limited accuracy of the polynomial coefficients.

Email addresses: graillat@univ-perp.fr (Stef Graillat), langlois@univ-perp.fr (Philippe Langlois).

Define reliable finite precision computations is an open challenge in symbolic computation and numerous results exist in this domain, *e.g.* [2,3,22]. On the other hand, the scientific computing community is used to manage the effects of finite precision computation to the stability of numerical algorithm and the accuracy of computed results, particularly in numerical linear algebra. Very less results are proposed for problems that involve polynomial computation. This gap may be justified since polynomial problems can be transformed in linear algebra problems (thanks to *ad hoc* matrices such that companion matrices, Sylvester matrix, ...) *in real arithmetic*.

The reliability of this transformation when problems are solved with finite precision arithmetic is not clear and motivates the kind of approach we describe herein. Two well known papers illustrate this difficulty for polynomial zerofinding : Toh and Trefethen report in [25, p.404] that "finding zeros via eigenvalues of companion matrices, the method used by the MATLAB `roots` command, is a stable algorithm" whereas Edelman and Murakami "construct examples for which a small componentwise relative backward error is neither predicted nor obtained in practice" [5, p.763]. This paper aims to illustrate that specific tools for polynomials exist and help to understand and solve some polynomial problems. We focus here the set of pseudozeros and some application of this tool introduced by Mosier [18] and, in our point of view, that have not been exploited enough. Main published results discuss the equivalence between the pseudozero set and the pseudospectra of the companion matrix [25,5]. We propose to revisit some aspects of polynomial pseudozeros proposing new applications, for example to test the primality of two univariate polynomials having coefficients known to a limited accuracy with these pseudozeros or stability criteria in system control.

The paper is organized as follows. Approximate polynomial problems are presented in Section 2. Next Section 3 is devoted to polynomial pseudozeros. This section is a survey on main results existing in the literature about the pseudozeros. Indeed, this notion is quite little know, that is why we recall some important results about it from Mosier [18], Trefethen and Tho [25], Edelman and Murakami [5]. First applications of pseudozeros are presented in Section 4. The test of the polynomial primality is proposed in Section 5. In Section 6, we deal with the computation of the pseudozero set in finite precision.

2 Polynomial GCD and primality in finite precision

2.1 Appropriate definitions are necessary in finite precision

The classic definition of the polynomial GCD does not fit the finite precision field.

For example, let p and q be two unitary polynomials such that $\deg p > 1$ and p divides q. It yields that $\gcd(p, q) = p$. Nevertheless, for any real number $\varepsilon > 0$, we have $\gcd(p, q+\varepsilon) = 1$: any small perturbation of the polynomial q critically affects the GCD. Since polynomial GCD does not depend continuously of the perturbation of its coefficients, computing a polynomial GCD is an ill-posed problem in the sense of Hadamard.

We have the same difficulty with an easiest problem: the *primality* of two polynomials. The following example from [3] is significant. As soon as the coefficients of the following polynomials

$$p(z) = (z - \frac{1}{3})(z - \frac{5}{3}) = z^2 - 2z + \frac{5}{9}, \quad \text{and} \quad q(z) = z - \frac{1}{3},$$

are represented with binary floating point numbers, p et q become *coprime* whereas they have a common root in real arithmetic. On the contrary, polynomials

$$p(z) = 50z - 7, \quad \text{and} \quad q(z) = z - \frac{1}{7},$$

are coprime in exact arithmetic whereas they share a common root if we seek it with two decimal digit numbers since $1/7 = 0.14285714$ and $7/50 = 0.14$.

To introduce an appropriate definition of polynomial primality in finite precision, we first briefly consider the more general notion of ε-GCD.

2.2 Polynomial ε-GCD

The following standard definition introduces ε-divisors and an ε-GCD (see [19] for example). We assume $\|\cdot\|$ is a norm on a polynomial field.

Definition 1 *Given two polynomials p and q of degree respectively n and m, and ε a positive real, an* ε-divisor *(or* approximate divisor*) of p and q is every divisor of perturbed polynomials $\widehat{p}$ and $\widehat{q}$ satisfying $\|p - \widehat{p}\| \leqslant \varepsilon$, $\|q - \widehat{q}\| \leqslant \varepsilon$ and $\deg(p - \widehat{p}) \leqslant n$, $\deg(q - \widehat{q}) \leqslant m$.*
An ε-GCD *of p and q is an ε-divisor of highest degree.*

One can verify that an ε-GCD is not unique in general.

Such ε-GCD fit well the classic case when coefficients of p and q polynomials suffer from uncertainty. Two types of algorithms have been proposed to compute polynomial ε-GCD.

(1) The classic Euclidean algorithm is modified by changing the tests and the stopping criterion [7,12]. This set of algorithms is natural since it is similar to the classic Euclidean algorithm but does not provide correct result in finite precision: it only yields an ε-divisor and no, in general, an ε-GCD.
(2) The second approach formulates the ε-GCD problem as an optimization problem [15]. Karmarkar and Lakshman compute an ε-GCD together with the perturbed polynomials. A part of the algorithm can be performed in finite precision but its complexity is exponential in the degree of the GCD.

In 1985, Schönhage was the first to tackle this problem introducing a nearby notion he called approximate GCD [22]. The proposed algorithm to compute this quantity is not always appropriated since it needs coefficients known up to an arbitrary precision. The very special case where polynomials are defined by roots suffering from uncertainty is considered by Pan in [19]. Reference [21] proposes entries and synthesis about ε-GCD algorithms.

2.3 Polynomial ε-primality

The corresponding *ε-primality* problem consists in proving whether ε-GCD(p, q) equals 1 or not.

Definition 2 *Let two polynomials p and q of degree respectively n and m and ε a positive real. Polynomials p and q are ε-coprime if ε-GCD$(p, q) = 1$.*

Of course, computing an ε-GCD and comparing it to 1 suffers from a too expensive complexity. A first challenging problem is to decide whether two given polynomials suffering from coefficient uncertainties bounded by a given quantity ε are coprime. Computing the minimum uncertainty that transforms coprime polynomials into non-coprime ones is a natural generalization of the previous question.

Beckermann and Labahn propose an algorithm to deal with primality without computing an ε-GCD [3]. Using the norm $\|p\| = \sum_j |p_j|$ defined on $\mathbb{C}[z]$ (p_j being p coefficients) and

$$\|(p, q)\| = \max\{\|p\|, \|q\|\} = \max\{\sum |p_i|, \sum |q_j|\},$$

they define $\epsilon(p,q)$ to be the minimum distance between two given polynomials and not coprime ones ; that is for $p, q \in \mathbb{C}[z]$,

$$\epsilon(p,q) = \inf\{\|(p-\widehat{p}, q-\widehat{q})\| : (\widehat{p},\widehat{q}) \text{ have a common root and } \deg\widehat{p} \leqslant n, \deg\widehat{q} \leqslant m\}.$$

Beckermann and Labahn compute a lower bound for $\epsilon(p,q)$ and so guarantee a primality neighborhood around p and q. This algorithm is integrated in the Maple distribution (since Release 8 [16]) as the SNAP package [14]. This algorithm costs $\mathcal{O}((n+m)^2)$ operations but does not always yield sharp bound for $\epsilon(p,q)$ as we exhibit it now.

Let us choose for example $p = z^2$ and $q = (z-1)^2$.
The function `DistanceToCommonDivisors` of the SNAP package yields 0.125 as a lower bound for $\epsilon(p,q)$. Now, we use the function `AreCoprime` asking for

```
> AreCoprime(p,q,z,0.2);
```

It returns `FAIL`, that means the software is not able to decide the 0.2-primality of p and q (other results are `TRUE` and `FALSE`).

Up to our knowledge, it exists no other algorithm to decide of polynomial ε-primality without computing an ε-GCD. In this paper, we solve this polynomial ε-primality problem thanks to the set of pseudozeros. We will explain later why separate pseudozeros of Figure 1 prove that p and q are 0.2-coprime.

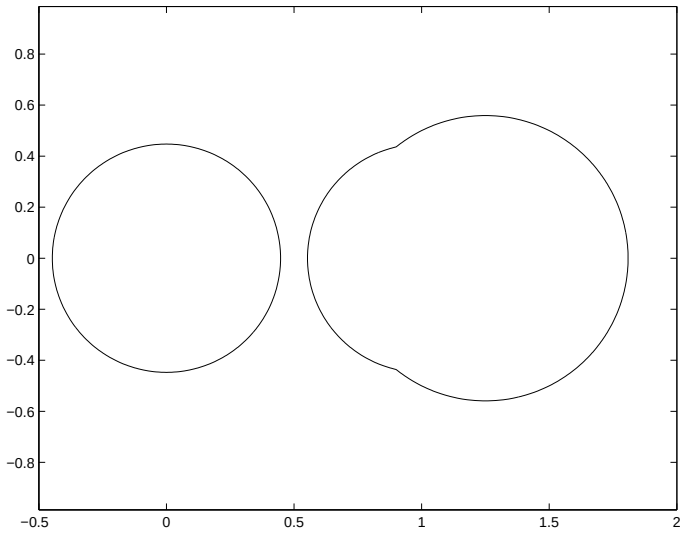


Fig. 1. ε-pseudozero set of the polynomials $p = z^2$ and $q(z) = (z-1)^2$ with $\varepsilon = 0.2$

3 Definition and computation of the ε-pseudozero set

In this section, we review definition and properties of pseudozeros from Mosier [18], Trefethen and Toh [25], Chatelin and Frayssé [4] and Stetter [24].

3.1 Definition of the ε-pseudozero set

The set $\mathbb{P}_n$ denotes the set of polynomials with complex coefficients and degree at most n, and $\mathbb{M}_n$ the monic polynomials of degree n with complex coefficients. Let $p \in \mathbb{P}_n$ given by

$$p(z) = p_0 + p_1 z + \cdots + p_n z^n. \tag{1}$$

Representing polynomial p by the vector of its coefficients, we choose the norm $\|\cdot\|$ on $\mathbb{P}_n$ being the norm on $\mathbb{C}^{n+1}$ of the polynomial coefficient vector.
For this norm, we define an ε-neighborhood of p to be the set of all polynomials of degree at most n, closed enough to p, that is,

$$N_\varepsilon(p) = \{\widehat{p} \in \mathbb{P}_n : \|p - \widehat{p}\| \leqslant \varepsilon\}. \tag{2}$$

Then the ε-pseudozero set of p is defined to include all the zeros of the ε-neighborhood of p. A non constructive definition of this set is

$$Z_\varepsilon(p) = \{z \in \mathbb{C} : \widehat{p}(z) = 0 \text{ for } \widehat{p} \in N_\varepsilon(p)\}. \tag{3}$$

3.2 A computable form of the ε-pseudozero set

The following theorem proves that the ε-pseudozero set can be obtain as a level contour of an easily computable function.

Theorem 3 *The ε-pseudozero set verifies*

$$Z_\varepsilon(p) = \left\{ z \in \mathbb{C} : |g(z)| = \frac{|p(z)|}{\|\underline{z}\|_*} \leqslant \varepsilon \right\}, \tag{4}$$

where $\underline{z} = (1, z, \ldots, z^n)$ and $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

Proof. We remind that the dual norm $\|\cdot\|_*$ on $\mathbb{C}^{n+1}$ is defined by

$$\|x\|_* = \max_{z \neq 0} \frac{|z^t x|}{\|z\|} = \max_{\|z\|=1} |z^t x|.$$

If $z \in Z_\varepsilon(p)$ then it exists $\widehat{p} \in \mathbb{P}_n$ such that $\widehat{p}(z) = 0$ et $\|p - \widehat{p}\| \leqslant \varepsilon$. From Hölder's inequality $|x^t y| \leqslant \|x\|\|y\|_*$, we get

$$|p(z)| = |p(z) - \widehat{p}(z)| = |\sum_{i=0}^{n}(p_i - \widehat{p}_i)z^i| \leqslant \|p - \widehat{p}\|\|\underline{z}\|_*.$$

It follows $|p(z)| \leqslant \varepsilon\|\underline{z}\|_*$.

To prove the reciprocal, let $u \in \mathbb{C}$ be such that $|p(u)| \leqslant \varepsilon\|\underline{u}\|_*$. The dual vector d of $\underline{u}$ verifies $d^*\underline{u} = \|\underline{u}\|_*$ and $\|d\| = 1$ [11, p. 278]. Let us introduce the polynomials r and p_u defined by

$$r(z) = \sum_{k=0}^{n} r_k z^k \text{ with } r_k = d_k, \tag{5}$$

$$p_u(z) = p(z) - \frac{p(u)}{r(u)}r(z). \tag{6}$$

This polynomial p_u is (with respect to the norm $\|\cdot\|$) the nearest polynomial of p with u as a root. It is clear that $r(u) = d^t\underline{u} = \|\underline{u}\|_*$. So we have

$$\|p - p_u\| = \frac{|p(u)|}{|r(u)|}\|r\| \leqslant \varepsilon\|d\|.$$

As $\|d\| = 1$, we get

$$\|p - p_u\| \leqslant \varepsilon.$$

And since $p_u(u) = 0$, u belongs to $Z_\varepsilon(p)$. □

3.3 Computing the ε-pseudozero set

Theorem 3 provides a computable expression for the ε-pseudozero set. It consists in evaluating a normalized form of polynomial p on a grid of the complex plane and comparing its value to the ε parameter.

MATLAB software, for example, provides primitives that allow us to plot pseudozeros with the following very simple Algorithm 1. Such an implementation is similar to existing pseudospectra software [6].

Without loss of generality, we assume a part of the complex plane has been chosen and discretized with a squared grid. Let L be the length of this square and h the step of discretization. The evaluation of $g(u)$ needs the evaluation of a polynomial, that costs $\mathcal{O}(n)$ operations, plus the computation of the norm of a vector whose complexity depends on the norm. For example, the computation of the $\|\cdot\|_1$ requires $n - 1$ operations and $\|\cdot\|_2$ requires $2n$

Algorithm 1 Computation of ε-pseudozero set (MATLAB version)

Require: polynomial p and precision ε
Ensure: pseudozero set layout in the complex plane
1: We grid a square containing all the roots of p with the MATLAB command `meshgrid`.
2: We compute $g(z)$ at the grid nodes z.
3: We draw the level line $|g(z)| = \varepsilon$ with the MATLAB command `contour`.

operations. Let us denote $\mathcal{O}(\|\cdot\|_*)$ this complexity. The complexity of the whole algorithm is in $\mathcal{O}(\lceil L/h \rceil^2 (n + \|\cdot\|_*))$.
Computing the pseudozero set has a lower complexity than computing the pseudospectra of the companion matrix. Indeed the pseudozero set requires the evaluation of a polynomial that costs $\mathcal{O}(n)$ whereas $\mathcal{O}(n^3)$ operations are necessary for the pseudospectra that uses SVD computations, and so for every node of the grid.

Previous results for pseudozeros are independent of the polynomial norm $\|\cdot\|$ that measures the coefficient perturbations. In practice, it is interesting to distinguish normwise and componentwise perturbations. The first ones describe every coefficient perturbations whereas the latter globally apply to the vector of coefficients. We expand the computable form of Theorem 3 for these two types of perturbations.

3.4 Pseudozeros for normwise perturbations

Let $p(z) = p_0 + p_1 z + \cdots + p_n z^n$ and $\widehat{p}$ a perturbed polynomial of p. We define the *normwise* norm by

$$\|p - \widehat{p}\|^{\mathcal{N}} = \frac{\|p - \widehat{p}\|}{\beta},$$

where $\|\cdot\|$ is a norm on the polynomials and β is a real. We usually choose $\beta = \|p\|$ to have a relative norm.

For such normwise perturbations, Theorem 3 gives the following result.

Corollary 4 *The ε-pseudozero set with* normwise *perturbations satisfies*

$$Z_\varepsilon^{\mathcal{N}}(p) = \left\{ z \in \mathbb{C} : \frac{|p(z)|}{\|\underline{z}\|_* \beta} \leqslant \varepsilon \right\}, \tag{7}$$

where $\underline{z} = (1, z, \ldots, z^n)$ and $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

3.5 Pseudozeros for componentwise perturbations

We define the *componentwise* norm by

$$\|p - \widehat{p}\|^{\mathcal{C}} = \max_i \frac{|p_i - \widehat{p}_i|}{f_i},$$

where $(f_i)_{i=0,\ldots,n}$ are non-negative real numbers. Usually, we take $f_i = |p_i|$ in order to have a relative norm. This perturbation provides a detailed description of the finite precision effect when the polynomial coefficient are represented with floating point numbers.

Theorem 3 now gives the following result.

Corollary 5 *The ε-pseudozero set with* componentwise *perturbations satisfies*

$$Z_\varepsilon^{\mathcal{C}}(p) = \left\{ z \in \mathbb{C} : \frac{|p(z)|}{\sum_{i=0}^n |f_i||z|^i} \leqslant \varepsilon \right\}. \tag{8}$$

4 First answers to approximate algebraic problems with pseudozeros

In this section, we present five first applications of pseudozeros. The two first ones focus on root computation in finite precision and are similar to results already presented in [25] for example. Two new applications are proposed to solve stability problems of polynomials that are very classic in system control. The last application illustrates a well-known result about the attainable accuracy when computing multiple polynomial roots.

4.1 A famous perturbed polynomial

The effect of finite precision representation of polynomial coefficients is simply illustrated with pseudozeros. Figure 2 is the well-known ε-pseudozero set of the Wilkinson polynomial

$$W_{20}(z) = (z-1)(z-2)\cdots(z-20),$$

where every coefficient is relatively perturbed with $\varepsilon = 2^{-24}$ (componentwise perturbations). This choice modelizes the behavior of W_{20} when it appears in computation processed with IEEE-754 single precision arithmetic [13].

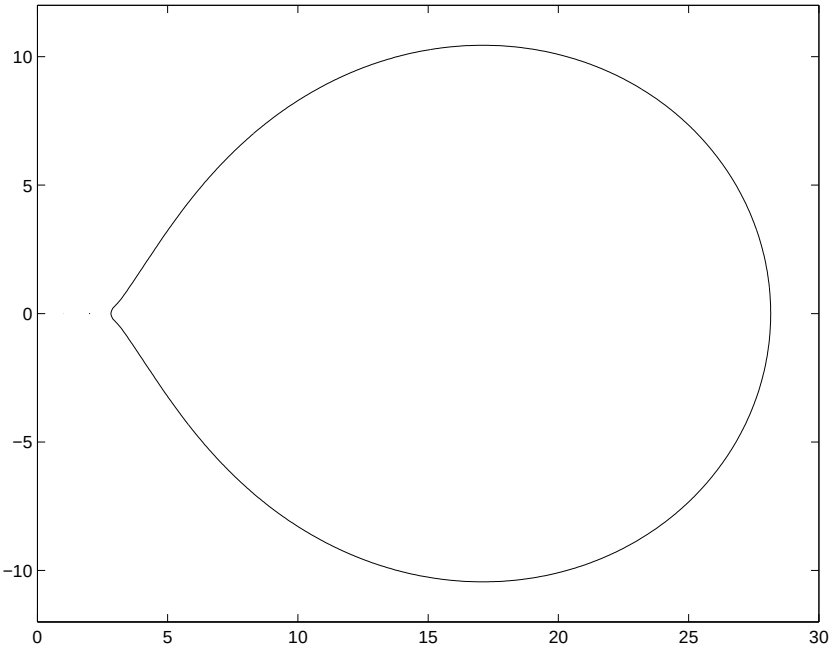


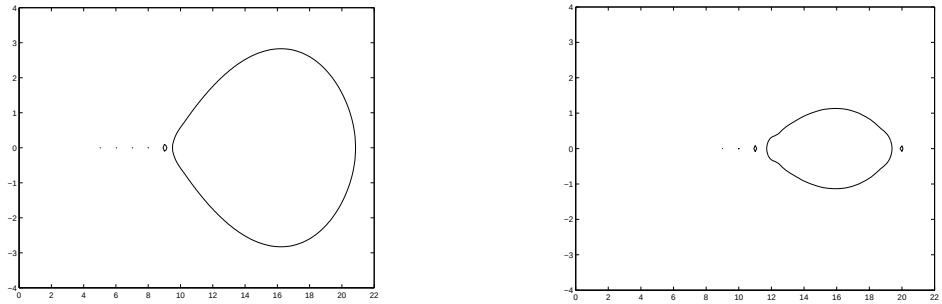
Fig. 2. ε-pseudozero set of Wilkinson polynomial for relative componentwise perturbation $\varepsilon = 2^{-24}$.

This plotting proves that a backward stable algorithm performed in IEEE-754 single precision may not be sufficient to isolate the roots $4, 5, \ldots, 20$ of W_{20}.

4.2 How much accuracy to compute the roots of this famous polynomial?

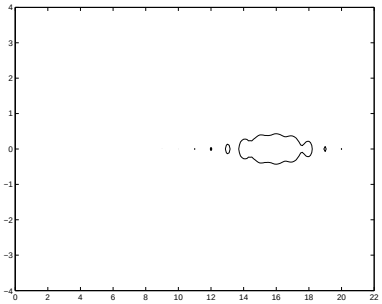
Figure 3 shows the evolution of the ε-pseudozero set of W_{20} when refining the precision ε. In these cases, we come back to the more classic localized perturbation of the z^{19} coefficient in a componentwise way. It is well known that extremal roots between 10 and 20 are less sensitive to coefficient perturbations than interior ones [26]. Pseudozeros illustrate very well this property.

These plots also give us the minimal precision necessary to represent the coefficients of a polynomial implementing W_{20} at precision ε such that its roots are still isolated. Here a binary mantissa of at least 33 bits is necessary to guarantee this property. From an algorithm point of view, this means one have to choose an algorithm that provides a backward error bounded by 2^{-33}. Hence, a backward stable algorithm running (at least) at this precision will compute approximate roots of W_{20} that are guaranteed to belong to distinct connected components of the pseudozero, *i.e.*, every connected component of the pseudozero includes only one approximate root of W_{20}.

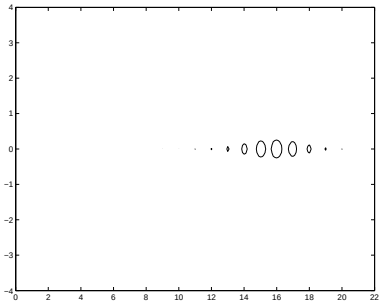


(a) $\varepsilon = 2^{-23}$

(b) $\varepsilon = 2^{-29}$



(c) $\varepsilon = 2^{-32}$



(d) $\varepsilon = 2^{-33}$

Fig. 3. Pseudozero set of the polynomial W_{20} for different values of ε.

4.3 Deciding of polynomial stability : a first criterion

The ε-pseudozero set can be used to decide the stability of system described by polynomials. A classic stability criterion in control theory is to compare the modulus roots to 1 (Schur stability). When the polynomial coefficients are known with a tolerance ε, it is difficult to compute every root of all polynomials in its ε-neighborhood. Of course, sensitivity analysis that uses the condition number of the polynomial with respect to its coefficients can be performed and yields a first order criterion. Pseudozeros provide an alternative answer to this question without neglecting higher order effects of coefficient uncertainties. It suffices to draw the ε-pseudozero set and verify if it is included in the unit circle.

Figure 4 shows the ε-pseudozero set of polynomial $p(z) = (z - 0.8)^2$ with two coefficient uncertainties $\varepsilon = 0.1$ and $\varepsilon = 0.01$.

From this figure, we cannot decide the stability for $\varepsilon = 0.1$ because some 0.1-pseudozeros have modulus larger than 1. On the other hand, the figure proves that all the 0.01-pseudozeros have modulus less than 1 and so we conclude for stability while $\varepsilon \leq 0.1$.

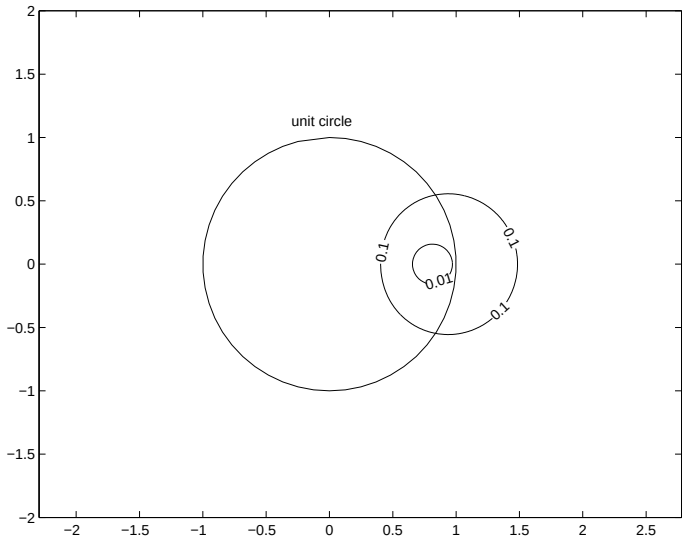


Fig. 4. ε-pseudozero set of the polynomial $p(z) = (z-0.8)^2$ with $\varepsilon = 0.1$ and $\varepsilon = 0.01$

4.4 Deciding of polynomial stability : a second criterion

Another kind of stability is defined when the real part of all the roots of a polynomial are negative (Hurwitz stability). For testing this other stability criterion, it suffices to draw the ε-pseudozero set and verify if it is included in the left half-plane. This is shown in Figure 5. We exhibit for example that polynomial $p(z) = (z+1)^2$ is stable if $\varepsilon = 0.4$.

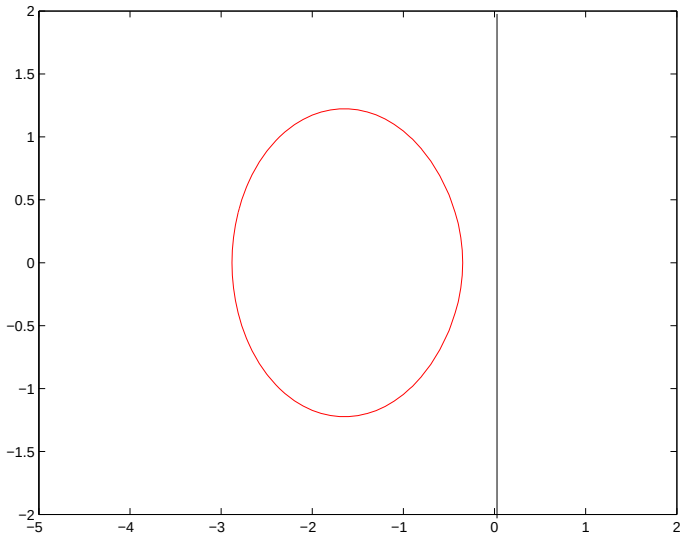


Fig. 5. ε-pseudozero set of the polynomial $p(z) = (z+1)^2$ with $\varepsilon = 0.4$

We may be interested in the stability of this polynomial but also in the minimum distance from a stable polynomial to the nearest unstable polynomial. This quantity, called the stability radius, is defined as follows. Let $a(p) = \max\{\mathrm{Re}(z) : p(z) = 0\}$ be the abscissa mapping for a polynomial

$p \in \mathbb{M}_n$. Hence, a stable polynomial satisfies $a(p) < 0$. The stability radius is defined to be

$$\beta(p) = \min\{\|p - q\| : q \in \mathbb{M}_n \text{ and } a(q) = 0\}.$$

Considering the 2-norm, we verify that

$$Z_\varepsilon(p) = \{(x, y) \in \mathbb{R}^2 : h_{\varepsilon,p}(x, y) \leqslant 0\}.$$

where $h_{\varepsilon,p}$ is the function from $\mathbb{R}^2$ to $\mathbb{R}$ defined by

$$h_{\varepsilon,p}(x, y) = |p(x + iy)|^2 - \varepsilon^2 \sum_{j=0}^{n-1} (x^2 + y^2)^j. \tag{9}$$

For a given x_0, function $h_{\varepsilon,p}(x_0, y)$ is a polynomial of degree $2n$, and so is $h_{\varepsilon,p}(x, y_0)$ for a given y_0. In [8], we derive a bisection algorithm to compute the stability radius $\beta(p)$ using this expression of the pseudozero set.

The first natural example is $p(z) = z + 1$. Of course, the nearest unstable polynomial of p is $q(z) = z$ and then $\beta(p) = 1$. If we apply the bisection algorithm, we find $\beta = 0.999996$ with a absolute tolerance equals to 0.00001. We plot the 0.999996-pseudozero set on Figure 6 to verify it.

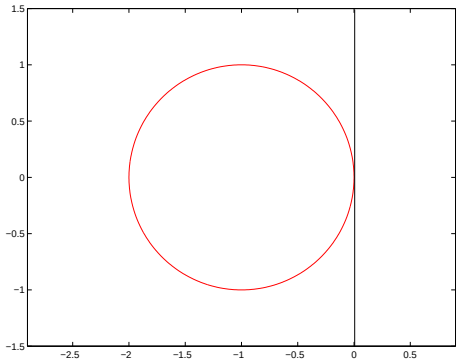


Fig. 6. ε-pseudozero set for $p(z) = z + 1$ with $\varepsilon = 0.999996 \approx \beta(p)$

Let us now compute the stability radius of $p(z) = (z - 1)(z - 1/2)$. Our algorithm yields $\beta = 0.485868$ with a tolerance 0.00001. Figure 7 presents the corresponding 0.485868-pseudozero set. It confirms that this perturbation limits the (Hurwitz) stability domain of considered polynomial.

4.5 Attainable accuracy of multiple root

A last application of pseudozeros illustrates the well-known "rule of thumb" that describes the attainable accuracy of a polynomial multiple root computed in precision ε: this accuracy is of the order of $\varepsilon^{1/m}$ where m is the multiplicity

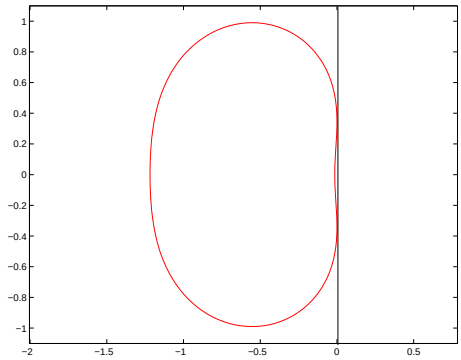
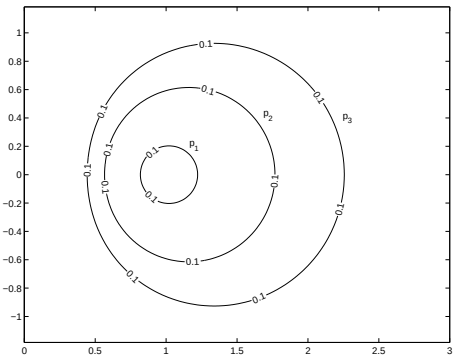


Fig. 7. ε-pseudozero set for $p(z) = z^2 + z + 1/2$ with $\varepsilon = 0.485868 \approx \beta(p)$

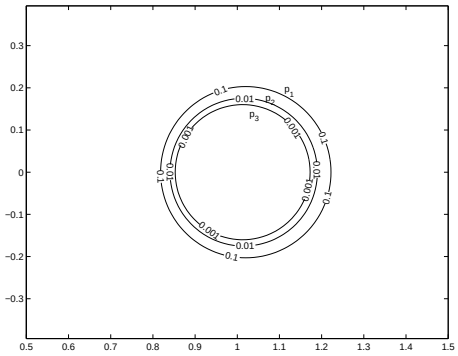
of the root ($\varepsilon < 1$). Another interpretation of this "rule of thumb" is that a backward stable algorithm needs a precision of at least ε^m to compute a root of multiplicity m satisfying an accuracy of the order of ε. This property is still valid for every polynomial that admits a polynomial with a m multiple root in its ε-neighborhood. We verify this property computing for example the ε-pseudozeros of polynomials

$$
\begin{aligned}
p_1(z) &= z - 1, \\
p_2(z) &= (z-1)^2, \\
p_3(z) &= (z-1)^3,
\end{aligned}
$$

with, respectively, $\varepsilon_1 = \varepsilon$, $\varepsilon_2 = \varepsilon^2$, $\varepsilon_3 = \varepsilon^3$ and $\varepsilon = 10^{-1}$. Figure 8 exhibits that the three sets $Z_{\varepsilon_1}(p_1)$, $Z_{\varepsilon_2}(p_2)$, and $Z_{\varepsilon_3}(p_3)$ are very similar (right side) compared to the ε-pseudozeros Z_ε of polynomials p_1, p_2 and p_3 (left side).



(a) Z_ε of p_1, p_2, p_3 and $\varepsilon = 10^{-1}$



(b) Pseudozero sets $Z_\varepsilon(p_1)$, $Z_{\varepsilon^2}(p_2)$, $Z_{\varepsilon^3}(p_3)$ for $\varepsilon = 10^{-1}$

Fig. 8. Comparison of different pseudozeros with respect to the root multiplicity

Now, we present how to solve the polynomial primality problem thanks to the pseudozero set.

5 Polynomial primality and ε-pseudozero set

Let p and q belonging respectively to $\mathbb{P}_n$ and $\mathbb{P}_m$. It follows from Definition 2 that p and q are ε-coprime if and only if for all $\widehat{p} \in N_\varepsilon(p)$, $\widehat{q} \in N_\varepsilon(p)$, the polynomials $\widehat{p}$ and $\widehat{q}$ are coprime.

The following proposition gives an answer to decide whether p and q are ε-coprime.

Proposition 6 *We have the two following assertions.*

(1) if the intersection of the ε-pseudozero sets of p and q is empty then the two polynomials are ε-coprime,
(2) if the intersection is not empty then they are not ε-coprime.

Proof. Let p and q be two polynomials with complex coefficients. If $Z_\varepsilon(p) \cap Z_\varepsilon(q) = \emptyset$ then from the ε-pseudozero set definition, we cannot find $\widehat{p} \in N_\varepsilon(p)$ and $\widehat{q} \in N_\varepsilon(q)$ having a common root. It means that p and q are ε-coprime. If now $Z_\varepsilon(p) \cap Z_\varepsilon(q) \neq \emptyset$, then let us take $a \in Z_\varepsilon(p) \cap Z_\varepsilon(q)$. It means that it exists $\widehat{p} \in N_\varepsilon(p)$ and $\widehat{q} \in N_\varepsilon(q)$ such that $\widehat{p}(a) = 0$, and $\widehat{q}(a) = 0$. Hence the polynomial $(z - a)$ divide $\widehat{p}$ and $\widehat{q}$. Therefore p et q are not ε-coprime. □

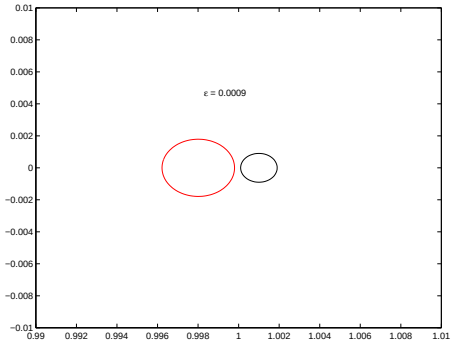
We apply this property considering, for example, p and q, where

$$p(z) = z^2 - 3.999z + 3.001, \quad \text{and} \quad q(z) = z^2 - 3.001z + 1.999.$$

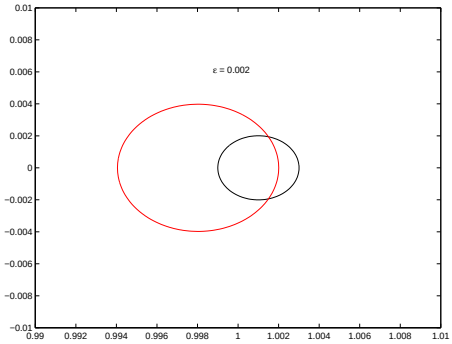
Figure 9 presents the ε-pseudozero sets of these two polynomials for two values of ε (0.0009 and 0.002). On the left hand side plotting, the intersection is empty so the two polynomial p and q are 0.0009-coprime. On the contrary, the intersection is not empty on the right hand plotting, so p and q are not 0.002-coprime. Here the perturbations due to coefficient reprsentation are neglected compared to presented values of ε.

Another example of the practical interest of this application of pseudozeros has been presented at the beginning of this paper to exhibit the weakness of the MATLAB function `AreCoprime`.

A more difficult problem is to compute the minimum perturbation that transforms two co-prime polynomials into non co-prime ones, *i.e.*, the minimum distance to the singularity for the polynomial primality problem. Using pseudozeros with a bisection algorithm is a natural idea to yield this minimum distance. Alas this approach needs to compute the intersection of two level contours, that is, for the 2-norm, to solve the system $f_1(x, y) = h_1$ and $f_2(x, y) = h_2$,



(a) p and q are 0.0009-coprime



(b) p and q have a common 0.002-divisor

Fig. 9. ε-pseudozero set for different values of ε of the polynomials p and q.

where f_i is a bivariate polynomial of degree $2n$. Up to our knowledge, this computation is also a difficult problem.

6 Computing ε-pseudozero set in finite precision

In this last section, we discuss two aspects of the finite precision computation of ε-pseudozero set.

6.1 How can we a priori choose the grid?

The initial grid must satisfy the two following conditions:

Condition a. Zeros and pseudozeros are included in its range;
Condition b. Roots are isolated by the grid discretization.

We discuss how to fulfill these conditions.

Condition a. Pseudozero set of monic polynomials are bounded (that is not guaranteed for arbitrary polynomials). So we restrict our pseudozero computation to monic polynomials p in $\mathbb{M}_n$. So now, for a polynomial p in $\mathbb{M}_n$, we denote $N_\varepsilon(p) = \{\widehat{p} \in \mathbb{M}_n : \|p - \widehat{p}\| \leqslant \varepsilon\}$ and $Z_\varepsilon(p) = \{z \in \mathbb{C} : \widehat{p}(z) = 0 \text{ for } \widehat{p} \in N_\varepsilon(p)\}$. In this case, the centered square $[-R, R] \times [-R, R]$ where

$$R = \max\left\{1, \sum_{k=0}^{n-1} |p_k| + n\varepsilon\right\},$$

ensures Condition a is satisfied. Let us prove this result.

Let p in M_n and $\{z_i\}$ the set of its n roots. The maximum root modulus verifies

$$r = \max_{i=1,\dots,n} |z_i| \leqslant \max\left\{1, \sum_{k=0}^{n-1} |p_k|\right\}$$

(see for example [17, p.154]). Let z be in $Z_\varepsilon(p)$, it exists $\widehat{p} \in N_\varepsilon(p)$ such that $\widehat{p}(z) = 0$. The modulus of this root z is bounded by the maximum root modulus of $\widehat{p}$. So we have the inequality

$$|z| \leqslant \max\left\{1, \sum_{k=0}^{n-1} |\widehat{p}_k|\right\}.$$

Assuming that the perturbation norm is an Hölder h-norm $\|\cdot\|_h$, we know that $\|p - \widehat{p}\|_h \leqslant \varepsilon$. Since $\|\cdot\|_\infty \leqslant \|\cdot\|_h$, $\|p - \widehat{p}\|_\infty \leqslant \varepsilon$. Then $|p_k - \widehat{p}_k| \leqslant \varepsilon$ and we have $|\widehat{p}| \leqslant |p_k| + \varepsilon$ for all $k = 0, \dots, n-1$. Hence,

$$|z| \leqslant \max\left\{1, \sum_{k=0}^{n-1} |p_k| + n\varepsilon\right\} = R.$$

We conclude that $Z_\varepsilon(p) \subset B(0, R)$, where $B(0, R)$ is the closed ball of center 0 and of radius R.

The drawbacks of this method is that if the polynomial coefficients are large then the grid can be very large whereas the roots are small. Other bounds for the roots may provide better results in this case (see [17] for instance).

Condition b. We need a grid that ensures the roots of p are isolated. The discretization step of the grid must be chosen consequently. The following two lower bounds on the distance between two distinct zeros of a polynomial are proposed by Rump [27] and Mignotte [17]. If $\{z_i\}$ is the set of its n roots, we have

$$\min |z_i - z_j| > \sqrt{\frac{8}{n^{n+2}}} \frac{1}{1 + \|p\|_\infty^n} =: \gamma_1 \quad \text{(Rump)},$$
$$\min |z_i - z_j| > \sqrt{\frac{3}{n^{n+2}}} \frac{1}{\|p\|_2^{n-1}} =: \gamma_2 \quad \text{(Mignotte)}.$$

For the drawing of the pseudozero sets, we choose the grid $[-R, R] \times [-R, R]$ described before with the step γ_1.

6.2 Accuracy limitation due to the finite precision evaluation of $p(z)$

The computation of the pseudozero set consists in the evaluation of the function $g(z) = p(z)/f(z)$ (where p is a polynomial and f a norm) performed

at every node of the chosen grid. For usual norms, we have $f(z) \geqslant 1$ and the associated computing error is negligible. The error in the evaluation of polynomial p has to be considered.

Let y be the evaluation of $p(z)$ using the Hörner's scheme. The following result of Kahan [20],

$$|y - p(z)| \leqslant 4\mathbf{u} \sum_{i=0}^{n} |s_i z^i| =: \eta \quad \text{with} \quad s_i = \sum_{j=i}^{n} p_j z^{j-i},$$

defines a precise *a priori* bound η of the error between p and its evaluation in the finite precision $\mathbf{u}$. Then no reliable interpretation of ε-pseudozeros can be proposed when $\varepsilon < \eta$.

From a practical point of view, it is then necessary to increase the computing precision $\mathbf{u}$ to guarantee a reliable evaluation of ε-pseudozeros. The running error bound of the rounding error in the polynomial evaluation yields both a more realistic bound of the error and, more importantly, an effective dynamic control of this error. The classic algorithm to compute this *a posteriori* error estimation for the Hörner scheme is given, for example, in [10, p.95]. Then it suffices to compare this computed bound to ε, and when is is necessary, to decide to increase the precision for example by using multiprecision library like [1] or MPFR [9] for example.

We limit this discussion on finite precision effects on pseudozeros to these two aspects. Of course, others limitations exist like for example the approximation error introduced by the MATLAB command `contour`. Future work will complete this analysis.

7 Conclusion

We have shown that plotting pseudozeros can give qualitative and sometimes quantitative interesting informations about the behavior of polynomials used in a finite precision environment. They offer a powerful tool to test the ε-primality of two polynomials and for example the stability and robustness of polynomials. They can be easily plotted used popular software as MATLAB.

We hope that pseudozeros will be used as much as pseudospectra since it seems to us that it could be useful for application fields as CGAD, control and network theory for example.

References

[1] David H. Bailey. *Multiprecision Software Directory.* http://crd.lbl.gov/~dhbailey/mpdist/index.html.

[2] Bernhard Beckermann and George Labahn. A fast and numerically stable Euclidean-like algorithm for detecting relatively prime numerical polynomials. *J. Symbolic Comput.*, 26(6):691–714, 1998.

[3] Bernhard Beckermann and George Labahn. When are two numerical polynomials relatively prime? *J. Symbolic Comput.*, 26(6):677–689, 1998.

[4] Françoise Chaitin-Chatelin and Valérie Frayssé. *Lectures on finite precision computations.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

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

[6] Mark Embree and Nick Trefethen. *Pseudospectra Gateway.* http://web.comlab.ox.ac.uk/projects/pseudospectra/.

[7] Ioannis Z. Emiris, André Galligo, and Henri Lombardi. Certified approximate univariate GCDs. *J. Pure Appl. Algebra*, 117/118:229–251, 1997.

[8] Stef Graillat and Philippe Langlois. Computation of stability radius for polynomials. Preprint no 31, Université de Perpigan, December 2003.

[9] Guillaume Hanrot, Vincent Lefèvre, Fabrice Rouillier, and Paul Zimmermann. The MPFR library, November 2003. Version 2.0.2. URL = `http://www.mpfr.org/`.

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

[11] Roger A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, 1990.

[12] V. Hribernig and H. J. Stetter. Detection and validation of clusters of polynomial zeros. *J. Symbolic Comput.*, 24(6):667–681, 1997.

[13] IEEE Computer Society, New York. *IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-1985*, 1985. Reprinted in SIGPLAN Notices, 22(2):9–25, 1987.

[14] C.-P. Jeannerod and G. Labahn. The SNAP package for arithmetic with numeric polynomials. In *Proceedings of the 2002 International Congress of Mathematical Software (ICMS'02)*, Beijing, China, August 2002.

[15] N. K. Karmarkar and Y. N. Lakshman. On approximate GCDs of univariate polynomials. *J. Symbolic Comput.*, 26(6):653–666, 1998. Symbolic numeric algebra for polynomials.

[16] *Maple 8*. http://www.maplesoft.com/.

[17] Maurice Mignotte. *Mathematiques pour le calcul formel*. Presses Universitaires de France, Paris, 1989.

[18] Ronald G. Mosier. Root neighborhoods of a polynomial. *Math. Comp.*, 47(175):265–273, 1986.

[19] Victor Y. Pan. Computation of approximate polynomial GCDs and an extension. *Inform. and Comput.*, 167:71–85, 2001.

[20] Douglas M. Priest. *On Properties of Floating Point Arithmetics: Numerical Stability and the Cost of Accurate Computations*. PhD thesis, Mathematics Department, University of California, Berkeley, CA, USA, November 1992. URL = `ftp://ftp.icsi.berkeley.edu/pub/theory/priest-thesis.ps.Z`.

[21] David Rupprecht. *Elément de Géométrie Approchée : Etude du pgcd et de la factorisation*. PhD thesis, Université de Nice-Sophia Antipolis, Janvier 2000.

[22] Arnold Schönhage. Quasi-gcd computations. *J. Complexity*, 1(1):118–137, 1985.

[23] T. W. Sederberg and G.-Z. Chang. Best linear common divisors for approximate degree reduction. *Computer-aided design*, 25(3):163–168, march 1993.

[24] Hans J. Stetter. Polynomials with coefficients of limited accuracy. In *Computer algebra in scientific computing—CASC'99 (Munich)*, pages 409–430. Springer, Berlin, 1999.

[25] Kim-Chuan Toh and Lloyd N. Trefethen. Pseudozeros of polynomials and pseudospectra of companion matrices. *Numer. Math.*, 68(3):403–425, 1994.

[26] James H. Wilkinson. The perfidious polynomial. In G. H. Golub, editor, *Studies in Numerical Analysis*, volume 24 of *Studies in Mathematics*, pages 1–28. Mathematical Association of America, Washington, D.C., 1984.

[27] Richard Zippel. *Effective polynomial computation.* The Kluwer International Series in Engineering and Computer Science. 241. Dordrecht: Kluwer Academic Publ., 1993.