Electronic Transactions on Numerical Analysis.
Volume 12, pp. 66-87, 2001.
Copyright © 2001, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

NUMERICAL CONDITION OF POLYNOMIALS IN DIFFERENT FORMS[*]

HONG ZHANG [†]

Abstract. The zeros of high-degree polynomials are notoriously sensitive to changes in the coefficients, causing problems for available zero-finding software. In this paper, we study how this sensitivity depends on the polynomial representation. We first extend the algebraic characterization of polynomial pseudozero sets from the power basis to general bases. We show that for a polynomial, the numerical conditions of its values and zeros are closely related and can be visualized simultaneously by its pseudozero sets. Comparing the pseudozero sets on a set of testing polynomials in the power, Taylor, Chebyshev, and Bernstein bases reveals that appropriate representation of polynomials gives rise to locally well-conditioned zeros, which then leads to an Iterative Refinement Algorithm that combines symbolic formulation with numeric processing to reduce computational errors of polynomial zeros located in the region of interest.

Key words. polynomial equation, polynomial basis, pseudozero set, iterative refinement algorithm.

AMS subject classifications. 65F35.

1. Introduction.

The problem of solving a polynomial equation

$$p(z) = c_0 + c_1 z + c_2 z^2 + \cdots + c_n z^n = 0 \tag{1.1}$$

has deeply influenced the development of mathematics since antiquity and continues to play a major role in many areas of science and engineering. Recent sources of interest include computer algebra, computational algebraic geometry, and computer aided geometric design. In these applications, one usually needs to solve (1.1) for large n, typically well above 100 and sometimes on the order of several thousands [7, 9, 11]. The zeros of high-degree polynomials are notoriously sensitive to changes in coefficients, causing problems for available software. However, the dependency of this sensitivity on the choice of polynomial basis has received little attention. Deeper quantitative understanding of this sensitivity dependency would be helpful for improving computational techniques and design of new algorithms.

Condition numbers are commonly used in the study of the sensitivity of polynomials. They are a kind of derivative, enabling us to estimate the magnitude of the changes of the values and zeros of a given polynomial that result from changes in the coefficients. See the classical work of Wilkinson [15]. Some authors have investigated the effect of polynomial bases on the condition numbers. For instance, Gautschi compared the condition numbers of polynomials in the power, orthogonal, and Lagrangian bases [6]. Farouki and Rajan derived the condition numbers for polynomials in the Bernstein basis [4, 5]. In recent years, a geometric approach has been used, which enables one to compute and view the regions in the complex plane in which the zeros may vary as the coefficients are perturbed [3, 10, 13]. With this approach, one can display detailed geometric information not only for simple polynomial zeros, but also for multiple zeros and clusters of zeros. The mathematical insight provided by this approach leads to well-conditioned representations of the problems as well as enhanced algorithms.

This paper takes the geometric approach. In section 2, we first extend the algebraic characterization of the polynomial pseudozero set represented in the power basis, introduced by Mosier [10], Toh and Trefethen [13], to general polynomial bases. We then show that for a polynomial, the numerical conditions of its values and zeros are closely related, and can be visualized simultaneously by its pseudozero sets. In section 3, we display and compare

[*] Received February 13, 2000. Accepted for publication January 26, 2001. Recommended by G. S. Ammar.

[†] Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, U.S.A. Email: hzhang@mcs.anl.gov

the pseudozero sets for a set of carefully selected polynomials in the power, Taylor, Chebyshev, and Bernstein bases. In section 4, the insight gained from the comparison leads to an Iterative Refinement Algorithm, which is shown to generate local optimal approximations to polynomial zeros under described practical situations. Finally, we present and discuss results of experiments with the algorithm, and point out issues for further investigation.

2. Condition of polynomial evaluation and polynomial zero. Let P_n denote the set of polynomials with complex coefficients and degree at most n, and let

$$B(z) = (p_0(z), p_1(z), ..., p_n(z))$$

be a basis of P_n. For each $p(z) \in P_n$, there are unique $c_k \in C$ such that

$$p(z) = \sum_{k=0}^{n} c_k p_k(z). \tag{2.1}$$

We will also use p to denote the vector of coefficients $(c_0, c_1, ..., c_n)^T$, when there is no danger of confusion. As usual, $\|p\|_\infty := \max_{0\leq k\leq n}\{c_k\}$. For any vector $d = (d_0, d_1, ..., d_n)^T$, $\|\cdot\|_d$ defines a pseudo-norm on P_n defined by

$$\|p\|_d = \max_{0\leq k\leq n} |d_k c_k|, \qquad p \in P_n.$$

Of course, if all $d_k \neq 0$, $\|\cdot\|_d$ is a d-weighted ∞-norm. In particular, $\|p - \hat{p}\|_d = \max_{0\leq k\leq n} |d_k(c_k - \hat{c}_k)|$ measures the perturbations in the coefficients of $p(z)$ relative to the weights given by d. We define the vector $d^{-1} = (d_0^{-1}, \cdots, d_n^{-1})^T$, where

$$d_k^{-1} = \begin{cases} 1/d_k & \text{if } d_k \neq 0 \\ 0 & \text{otherwise} \end{cases}, \qquad k = 0, \cdots, n,$$

and we let

$$\|B(z)\|_{d^{-1}} = \sum_{k=0}^{n} |d_k^{-1} p_k(z)|.$$

Below, p is generally fixed and the c_k's are relative to a basis B. We put $d := \|p\|_\infty p^{-1} = \|p\|_\infty (c_0^{-1}, \cdots, c_n^{-1})^T$, which corresponds to coefficient-wise perturbations to p. Under this d, we have

$$\|p\|_d = \|p\|_\infty = \max_{0\leq k\leq n} |c_k|, \quad \text{and} \quad \|B(z)\|_{d^{-1}} = \left(\sum_{k=0}^{n} |c_k p_k(z)|\right) / \|p\|_\infty.$$

We define the ϵ-neighborhood and the ϵ-pseudozero set of $p(z)$ as

$$N_\epsilon(p, B) = \{\hat{p} \in P_n : \|p - \hat{p}\|_d \leq \epsilon\},$$

and

$$Z_\epsilon(p, B) = \{z \in C : \hat{p}(z) = 0 \text{ for some } \hat{p} \in N_\epsilon(p, B)\}. \tag{2.2}$$

They are respectively, the set of all polynomials $\hat{p}$ and their zeros obtained by coefficient-wise perturbations of p of size $\leq \epsilon$.

LEMMA 2.1. *For any $\hat{z} \in C$, define the polynomials*

$$r(z) = \sum_{k=0}^{n} r_k p_k(z), \qquad \text{where } r_k = |d_k^{-1}| e^{-i \arg(p_k(\hat{z}))},$$

and

$$p_{\hat{z}}(z) = p(z) - \frac{p(\hat{z})}{r(\hat{z})} r(z).$$

Then, we have,

1. $p_{\hat{z}}(\hat{z}) = 0$,
2. $p_{\hat{z}}(z) \in N_\epsilon(p, B)$, *with* $\epsilon = \frac{|p(\hat{z})|}{\|B(\hat{z})\|_{d^{-1}}}$, *and*
3. $\|p_{\hat{z}} - p\|_d = \frac{|p(\hat{z})|}{\|B(\hat{z})\|_{d^{-1}}} = \min\{\|\hat{p} - p\|_d : \hat{p} \in P_n \text{ and } \hat{p}(\hat{z}) = 0\}$.

Proof. Using $r(\hat{z}) = \|B(\hat{z})\|_{d^{-1}}$ and $\|r\|_d = 1$, follow the proof given by Mosier (see [10, p. 266]). □

An algebraic characterization of the ϵ-pseudozero set $Z_\epsilon(p, B)$ under the power basis was first introduced by Mosier [10], then by Toh and Trefethen [13]. Here we extend their results to general bases $B(z)$.

PROPOSITION 2.2. *Let $p(z) \in P_n$. If $p(z)$ has the coefficients $p = (c_0, c_1, ..., c_n)^T$ with respect to the basis $B(z) = (p_0(z), p_1(z), ..., p_n(z))$, then*

$$Z_\epsilon(p, B) \quad = \quad \{z \in C : \frac{|p(z)|}{\|B(z)\|_{d^{-1}}} \leq \epsilon\}. \tag{2.3}$$

Proof. Straightforward application of Lemma 2.1. □

The quantity $\|B(z)\|_{d^{-1}}$, which appeared in the ϵ-pseudozero set of $p(z)$, also appears in the condition numbers for the polynomial evaluation at z.

LEMMA 2.3. *Let $p(z) = \sum_{k=0}^{n} c_k p_k(z)$. Then the absolute and the relative condition numbers for the evaluation of $p(z)$ at z_0 are respectively*

$$\hat{\kappa}(z_0, p, B) = \|B(z_0)\|_{d^{-1}} \qquad \text{and} \qquad \kappa(z_0, p, B) = \frac{\|p\|_d \|B(z_0)\|_{d^{-1}}}{|p(z_0)|}. \tag{2.4}$$

Proof. Define the map $M : C^{n+1} \to C$, which associates with each vector $p = (c_0, c_1, ..., c_n)^T \in C^{n+1}$ the value $p(z) = \sum_{k=0}^{n} c_k p_k(z)$. The absolute condition number of M at p is defined as (see [14, p. 90])

$$\hat{\kappa}(z_0, p, B) = \lim_{\delta \to 0} \sup_{\|\hat{p} - p\|_d \leq \delta} \frac{|\hat{p}(z_0) - p(z_0)|}{\|\hat{p} - p\|_d}, \qquad \text{where } \hat{p}(z_0) = \sum_{k=0}^{n} \hat{c}_k p_k(z_0).$$

Since

$$|\hat{p}(z_0) - p(z_0)| = \left| \sum_{k=0}^{n} d_k (\hat{c}_k - c_k) d_k^{-1} p_k(z_0) \right| \leq \|\hat{p} - p\|_d \|B(z_0)\|_{d^{-1}} \text{ for any } \hat{p} \in C^{n+1},$$

thus $\hat{\kappa}(z_0, p, B) \leq \|B(z_0)\|_{d^{-1}}$. The upper bound $\|B(z_0)\|_{d^{-1}}$ can be attained for any given $p = (c_0, c_1, ..., c_n)^T \in C^{n+1}$ and $z_0 \in C$ by choosing $\hat{p} = (\hat{c}_0, \hat{c}_1, ..., \hat{c}_n)^T$ with $\hat{c}_k = c_k + \delta |d_k^{-1}| e^{-i \arg(p_k(z_0))}$, because $\|\hat{p} - p\|_d = \delta$ and $|\hat{p}(z_0) - p(z_0)| = \delta \|B(z_0)\|_{d^{-1}}$.

The relative condition number is

$$\kappa(z_0, p, B) := \hat{\kappa}(z_0, p, B)\frac{\|p\|_d}{|p(z_0)|} = \frac{\|p\|_d \|B(z_0)\|_{d^{-1}}}{|p(z_0)|}.\Box$$

The pseudozero set $Z_\epsilon(p, B)$ quantifies the conditioning of the zeros of the polynomial p. Using a stable zero-finding algorithm, the computed zeros of p should lie in a region $Z_{Cu}(p, B)$, where $C = O(\|p\|_d)$ and u is the machine precision. ($O(f)$ denotes any function whose absolute value is at most $c|f|$ for a positive constant c, see [1, p. 7]). If ϵ is chosen of the same order as u, i.e., $\epsilon = O(u)$, then the computed zeros of p should fall into a region that almost coincides with $Z_{\epsilon\|p\|_d}(p, B)$. On the other hand, the condition number $\kappa(z, p, B)$ provides bounds for possible perturbations of the computed value of $p(z)$. When p is evaluated at z by a stable method with the machine precision ϵ, the relative computing error in evaluating $p(z)$ should be no more than $\epsilon\kappa(z, p, B)$. Although the polynomial evaluations and polynomial zeros are two distinct problems, Proposition 2.2 and Lemma 2.3 reveal an interesting relationship between their numerical conditions:

Corollary 2.4.

$$Z_{\epsilon\|p\|_d}(p, B) = \{z \in C : \frac{1}{\kappa(z, p, B)} \leq \epsilon\}. \tag{2.5}$$

That is, when the coefficients of a polynomial $p(z)$ are perturbed, κ and $1/\kappa$ provide information on how the values and the zeros, respectively, of p may change; or in other words, the numerical conditions of a polynomial, for both its values and zeros, can be revealed simultaneously by the magnitudes of either $\kappa(z, p, B)$ or $1/\kappa(z, p, B)$. Since $\kappa(z, p, B) = \infty$ at the zeros of p, it is appropriate to use the quantity $1/\kappa(z, p, B)$, whose ϵ-level set is exactly the boundary of $Z_{\epsilon\|p\|_d}(p, B)$, the $\epsilon\|p\|_d$-pseudozero set of $p(z)$.

3. Condition of polynomial basis.

The conditions of polynomials are basis-dependent, a fact that has been noticed and investigated by several researchers using the condition numbers. For instance, Gautschi introduced the condition numbers of polynomials in the power, orthogonal, and Lagrangian bases [6]. He compared the condition numbers of the zeros of the polynomials (i) and (vii) given below in the power and the Chebyshev bases, and concluded that “it is quite possible that equations that are ill-conditioned in power form become well-conditioned when expanded as orthogonal polynomials, and vice versa.” Farouki and Rajan [4] derived the condition numbers for polynomials in the Bernstein basis, and proved that the condition numbers for both simple and multiple polynomial zeros on a real interval (α, β) are always smaller in the Bernstein basis than in the Taylor basis about any $s \notin (\alpha, \beta)$.

Our discussions in the previous section yield a geometric, closed-form representation (eqs. (2.4) and (2.5)) for the conditions of polynomials with respect to any basis $B(z)$ of P_n. To be more specific, the influence of a basis on the conditioning of the polynomial (2.1) is determined by the quantity

$$\frac{1}{\kappa(z, p, B)} = \frac{|p(z)|}{\|p\|_d \|B(z)\|_{d^{-1}}} = \frac{|p(z)|}{\sum_{k=0}^{n} |c_k p_k(z)|}, \qquad \text{when } d = \|p\|_\infty p^{-1},$$

whose ϵ-level sets are boundaries of the pseudozero sets $Z_{\epsilon\|p\|_d}(p, B)$, which can be efficiently computed and displayed in the complex plane.

In this section, we examine four bases of P_n:

- the power basis: $B^{(0)}(z) = (1, z, z^2, \cdots, z^n)$,
- the Taylor basis about s: $B^{(s)}(z) = (1, z - s, (z - s)^2, \cdots, (z - s)^n)$ (note that the power basis $B^{(0)}(z)$ is a special case of the Taylor basis),

- the Chebyshev basis: $B^c(z) = (T_0(z), T_1(z), \cdots, T_n(z))$, where $T_k(z) = \cos k(\arccos z)$, $-1 \le Re(z) \le 1$, and
- the Bernstein basis: $B^b(z) = (b_0^n(z), b_1^n(z), \cdots, b_n^n(z))$, with $b_k^n(z) = \binom{n}{k}(\beta - z)^{n-k}(z-\alpha)^k/(\beta-\alpha)^n$, $\alpha \le Re(z) \le \beta$.

The power basis is by far the best understood and most commonly used basis for polynomials. The Taylor basis is simply a shift or a variable exchange of the power basis. The Chebyshev basis is an orthogonal basis and has been used in approximation theory, algebra, and number theory, as well as numerical analysis and computation. The Bernstein basis was originally constructed by probabilistic reasoning for use in approximating continuous functions. One of its applications is in parameterizing Bernstein-Bézier curves, which have enjoyed considerable popularity in computer aided design applications due to their elegant geometric properties and the simple recursive algorithms available for processing them.

For each of the bases listed above, we explore the following degree-20 polynomials, most of which are chosen or modified from [13]:

(i) scaled “Wilkinson polynomial”: $p(z) = \prod_{k=1}^{20}(z - k/20)$.
(ii) the Chebyshev polynomial of degree 20: $p(z) = \cos 20(\arccos z)$, $-1 \le Re(z) \le 1$.
(iii) the monic polynomial with zeros equally spaced on the curve $z = x + i\sin(\pi x)$, $-1 \le x \le 1$, namely, $2(k+0.5)/19 + i \sin 2\pi(k+0.5)/19$, $k = -10, -9, \cdots, 9$.
(iv) $p(z) = \sum_{k=0}^{20}(10z)^k/k!$.
(v) the monic polynomial with zeros: $10/11 - 2^{-k}$, $k = 1, 2, \cdots, 20$.
(vi) $p(z) = (z - 10/11)^{20}$.
(vii) the monic polynomial with zeros: 2^{-19}, 2^{-18}, $\cdots$, 1.
(viii) $p_1(z) = 1 + z + z^2 + \cdots + z^{19} + z^{20}$, $p_2(z) = (1 + z + z^2 + \cdots + z^{10})^2$, and $p_3(z) = (1 + z + z^2 + \cdots + z^5)^4$.

Figures 6.1 - 6.9 display the pseudozero sets $Z_{\epsilon\|p\|_d}(p, B)$ whose boundaries are labeled by $\log_{10}\epsilon$. The experiments were implemented using MATLAB and its Symbolic Math Toolbox. We first input a polynomial $p(z)$ in the power basis. We then converted the coefficients to the coefficients in three other bases symbolically or using 128 decimal precision. Next, we numerically evaluated the function $1/\kappa(z, p, B)$ at 100×100 grid points in a rectangular region that contains all the zeros of p. We finally drew the level sets of $-\log_{10}\kappa(z, p, B)$ using the MATLAB function *contour*.

All the figures confirm that the pseudozero sets of polynomials are basis-dependent, consequently the numerical conditioning of polynomial evaluation and polynomial zero are basis-dependent.

The first phenomenon observed in our experiments is that the components of the pseudozero sets in the power basis $B^{(0)}$ that contain polynomial zeros with small magnitudes are smaller than those containing zeros with larger magnitudes. In particular, the component that contains a root in a neighborhood of $z = 0$ is significantly smaller than the rest of the components of $Z_\epsilon(p, B^{(0)})$. This observation is verified by eqs. (2.4)-(2.5), since $\|B^{(0)}(z)\|_{d^{-1}}$ is a monotonic decreasing function of $|z|$:

$$\|B^{(0)}(z)\|_{d^{-1}} = \sum_{k=0}^{n} |d_k^{-1}||z^k| < \sum_{k=0}^{n} |d_k^{-1}||\tilde{z}^k| = \|B^{(0)}(\tilde{z})\|_{d^{-1}}, \quad \text{whenever } |z| < |\tilde{z}|.$$

Consequently, the component of $Z_\epsilon(p, B^{(s)})$ in the Taylor basis about s that contains the polynomial zeros in the neighborhood of s would be smaller than the other components of $Z_\epsilon(p, B^{(s)})$. The Bernstein basis $b_k^n(z) = \binom{n}{k}(\beta - z)^{n-k}(z-\alpha)^k/(\beta-\alpha)^n$, $k = 0, \cdots, n$, behaves analogously around $z = \alpha$ and $z = \beta$.

Now let us consider polynomial zeros that are far from the origin $z = 0$. Figures 6.1 - 6.4 show that at well-isolated simple zeros, either real or complex, the pseudozero sets in any of the three bases, power, Chebyshev, and Bernstein, have areas within an order of magnitude of one another. In general, the Bernstein basis has slightly smaller pseudozero sets than the other two. On the other hand, figures 6.5 - 6.7 show that the conditions of clustered or multiple polynomial zeros are generally worse than those of simple zeros for each of the four bases. Between these bases, the Bernstein basis has the smallest pseudozero sets, followed by the Chebyshev basis, and the power basis is the worst (except the case in which a cluster or a multiple zero is located around $z = 0$, such as the polynomial (vii) for which the power basis behaves better than the Chebyshev basis; also see [6]).

Figures 6.8 and 6.9 reveal that, as polynomial zeros become closer or multiplicities of polynomial zeros become larger, the conditioning of the polynomials in the power basis, and of course in the Taylor basis, deteriorates rapidly, yet the pseudozero sets of the polynomials in the Bernstein basis are relatively unchanged. This phenomenon is also observed in figures 6.5 and 6.6, in which the disparities between the conditioning of the power, Chebyshev, and Bernstein bases become more striking as the cluster of polynomial zeros $\xi_k = 10/11 - 2^{-k}$, $k = 1, \cdots, 20$, merges into a single 20-fold zero $\xi = 10/11$.

4. Accuracy refinement of polynomial zeros. The analysis and experiments given in the previous sections demonstrate that polynomial bases that have smaller $\|B(z)\|_{d^{-1}}$ in the region of interest give rise to locally better conditioned representations, which suggests that conditioning of polynomials can be improved through careful basis transformation. Since the zero-finding procedure used for the power basis can be applied directly to the Taylor basis representation by a variable substitution, an immediate application of this idea is to use the Taylor basis $B^{(s)}(z)$ with s chosen in the region of interest to improve local numerical condition of polynomials that are initially represented in the power basis. Another application is to refine approximated polynomial zeros to optimal accuracy using a sequence of Taylor bases $\{B^{(s_k)}(z), \quad k = 1, 2, \cdots\}$.

Let $p(z) \in P_n$ be represented in the power basis $B^{(0)}(z)$. With no loss of generality, we assume $p(0) \neq 0$. (Otherwise $p(z) = z^m g(z)$ with $g(0) \neq 0$, the same discussions are applied to $g(z)$.) For any two Taylor bases $B^{(s_1)}(z)$ and $B^{(s_2)}(z)$, assuming $O(|p(s_2)|) = O(|p(s_1)|)$ as $s_i \rightarrow \xi$ with $p(\xi) = 0$, if $|z - s_2| \ll |z - s_1|$, then

$$\begin{aligned}\|B^{(s_2)}(z)\|_{d^{-1}} &= \sum_{k=0}^{n} \left|\frac{p^{(k)}(s_2)}{k!}(z - s_2)^k\right| \\ &\ll \sum_{k=0}^{n} \left|\frac{p^{(k)}(s_1)}{k!}(z - s_1)^k\right| = \|B^{(s_1)}(z)\|_{d^{-1}},\end{aligned}$$

which implies that when $|\xi - s_2| \ll |\xi - s_1|$ with $p(\xi) = 0$, the conditioning of the zero ξ under the basis $B^{(s_2)}(z)$ is likely better than using the basis $B^{(s_1)}(z)$. Proposition 4.1 formalizes this argument.

Let $p(z) \in P_n$. Suppose $p(z)$ has n_0 well-separated zeros (here, a cluster of zeros is considered as a multiple zero). For $\epsilon > 0$ small enough, the set $Z_\epsilon(p, B)$ has n_0 disjoint connected components, each of which contains a zero of $p(z)$ (see [10]). We use $Z_\epsilon^{(\xi)}(p, B)$ to denote the connected component of $Z_\epsilon(p, B)$ that contains the zero ξ.

PROPOSITION 4.1. *Let ξ be an m-fold zero of $p(z)$, and $Z_1 = Z^{(\xi)}_{\epsilon\|p\|_d}(p, B^{(0)}) \cap \{z \in C : |z - \xi| \leq \delta_0\}$, where δ_0 is chosen such that Z_1 is a small neighborhood of ξ. Define*

$$Z_k = Z^{(\xi)}_{\epsilon\|p\|_d}(p, B^{(s_{k-1})}) \cap Z_{k-1}, \qquad k = 2, \cdots, m,$$

where $s_{k-1} \in Z_{k-1}$, and $\delta_k = max\{|z - \hat{z}| : \forall z(\hat{z} \in Z_k)\}$. Then,

$$|p(z)| \leq O(\epsilon^k) \textit{ for any } z \in Z_k, \textit{ and } \delta_k \leq O(\epsilon^{k/m}). \tag{4.1}$$

Proof. We prove the proposition by induction on k. Let T denote the function

$$T(z,s) = \|p\|_d \|B^{(s)}(z)\|_{d^{-1}} = \sum_{i=0}^{n} \left| \frac{p^{(i)}(s)}{i!}(z-s)^i \right|, \quad \forall (z,s) \in C^2,$$

(note that $d = \|p\|_\infty p^{-1}$ and $p(z) = \sum_{i=0}^{n} \frac{p^{(i)}(s)}{i!}(z-s)^i$). For $k=1$, $\forall z \in Z_1$, $|p(z)| \le \epsilon T(z,0) = O(\epsilon)$, because $T(z,0)$ is bounded on Z_1. Then, we have,

$$|z-\xi| = \left| \frac{p(z)}{p^{(m)}(\xi)/m! + O(z-\xi)} \right|^{1/m} \le O(|p(z)|^{1/m}) \le O(\epsilon^{1/m}),$$

leading to $|z-\hat{z}| \le |z-\xi| + |\hat{z}-\xi| \le O(\epsilon^{1/m})$ for any z and $\hat{z} \in Z_1$. Thus $\delta_1 \le O(\epsilon^{1/m})$.

Now, suppose $|p(z)| \le O(\epsilon^k)$, $\forall z \in Z_k$ and $\delta_k \le O(\epsilon^{k/m})$ for $2 \le k < m$. Then, for $s_k \in Z_k$,

$$|p(s_k)| \le O(\epsilon^k),$$

and

$$|p^{(i)}(s_k)| = \left| \frac{p^{(m)}(\xi)}{(m-i)!}(s_k-\xi)^{m-i} + O((s_k-\xi)^{m-i+1}) \right| \le O(\delta_k^{m-i}), \quad i = 1, 2, \cdots, m-1.$$

Thus, $\forall z \in Z_{k+1}$,

$$\begin{array}{rcl} T(z,s_k) & = & |p(s_k)| + \sum_{i=1}^{m-1} \left| \frac{p^{(i)}(s_k)}{i!}(z-s_k)^i \right| + \sum_{i=m}^{n} \left| \frac{p^{(i)}(s_k)}{i!}(z-s_k)^i \right| \quad (\text{ since } z, s_k \in Z_k) \\ & \le & O(\epsilon^k) + \sum_{i=1}^{m-1} O(\delta_k^{m-i})\delta_k^i + O(\delta_k^m) \\ & \le & O(\epsilon^k), \end{array}$$

which gives

$$|p(z)| \le \epsilon T(z,s_k) \le O(\epsilon^{k+1}).$$

Using the same discussion as in the case $k=1$, we obtain $|z-\hat{z}| \le O(\epsilon^{(k+1)/m})$, for any z and $\hat{z} \in Z_{k+1}$, i.e., $\delta_{k+1} \le O(\epsilon^{(k+1)/m})$. □

Proposition 4.1 yields a constructive way to refine the accuracy of approximations to polynomial zeros. Based on this, we designed the following *Iterative Refinement Algorithm*.

```
**Algorithm:** $\hat{\xi}_{new} = REFI(p, \hat{\xi}_{old}, tol)$
Input: $p$: polynomial with complex coefficients and degree $n$.
       $\hat{\xi}_{old}$: an approximation to $\xi$, where $\xi$ is an m-fold zero of $p$.
       $tol$: error tolerance.
Output: $\hat{\xi}_{new}$: $|\hat{\xi}_{new} - \xi| < |\hat{\xi}_{new}| \cdot tol$, or a new approximation to $\xi$.

Set $\hat{\xi}_1 = \hat{\xi}_{old}$.
For $k = 1, \cdots, m-1$
   (i) Form $p(z) = \sum_{i=0}^{n} a_i(z - \hat{\xi}_k)^i = f(y)$, where $y = z - \hat{\xi}_k$.
   (ii) Compute $\hat{\xi}_f$, the root of $f(y)$ with the smallest magnitude by a stable zero-finding numerical method.
   (iii) Set $\hat{\xi}_{k+1} = \hat{\xi}_k + \hat{\xi}_f$.
   (iv) If $|\hat{\xi}_f| < |\hat{\xi}_{k+1}| \cdot tol$, stop.
Set $\hat{\xi}_{new} = \hat{\xi}_{k+1}$.
```

The algorithm generates a set of approximations $\{\hat{\xi}_k\}$ to ξ, an m-fold zero of $p(z)$. Assuming that ϵ is of the order of the machine precision, and that the input $\hat{\xi}_{old} \in Z_1 = Z^{(\xi)}_{\epsilon\|p\|_d}(p, B^{(0)})$ is well separated from other connected components of $Z_{\epsilon\|p\|_d}(p, B^{(0)})$, then each $\hat{\xi}_k$ is expected to fall into $Z^{(\xi)}_{\epsilon\|p\|_d}(p, B^{(\hat{\xi}_{k-1})})$, since $\hat{\xi}_k$ is computed by a stable zero-finding numerical method for $p(z)$ represented in the basis $B^{(\hat{\xi}_{k-1})}(z)$. Although it is impossible to rigorously establish that each $\hat{\xi}_k$ also lies inside Z_{k-1} as required by Proposition 4.1, we can reasonably conjecture that when ϵ is small enough,

$$\delta_k = O(\epsilon^{k/m}) \ll O(\epsilon^{(k-1)/m}) = \delta_{k-1}, \qquad k = 2, 3, \cdots,$$

i.e., Z_k is much smaller than Z_{k-1} if they have similar geometric shapes. Since ξ is a common interior point of Z_{k-1} and Z_k, it is likely that $Z_k = Z^{(\xi)}_{\epsilon\|p\|_d}(p, B^{(\hat{\xi}_{k-1})}) \cap Z_{k-1} \approx Z^{(\xi)}_{\epsilon\|p\|_d}(p, B^{(\hat{\xi}_{k-1})})$, and consequently it is highly possible that $\hat{\xi}_k \in Z_k$. According to Proposition 4.1, $\{\hat{\xi}_k\}$ then converges to ξ linearly at the rate $\epsilon^{1/m}$, and it would take at most $m-1$ refinements to get an optimal approximation within the given machine precision.

The Taylor coefficients $p = (a_0, a_1, \cdots, a_n)^T$ in Step (i) can be computed by using from about $9n \log_2 n$ to $18n \log_2 n$ arithmetic operations [1]. Since basis conversions from the power basis are usually numerically unstable [5], the coefficients in the new bases should be computed either symbolically or numerically with high digit precision.

The algorithm can be used interactively. For example, after a set of desirable zeros of the polynomial $p(z)$ is computed by a zero-finding routine, the pseudozero set can be plotted around the computed zeros as shown in figures 6.1 - 6.9. If a suspect ill-conditioned zero is viewed, say $\hat{\xi}$, we may implement REFI with $\hat{\xi}_{old} = \hat{\xi}$ ($\hat{\xi}_{old}$ can also be input by clicking the mouse on the computer screen at the center of the suspect component of the pseudozero set). The zeros can be refined either one at a time, or simultaneously on several subsets. For the former, after each zero is refined, the polynomial is deflated and REFI is then applied to the deflated polynomial (care should be taken to ensure deflation stability, see [8, 12, 15] for details); while for the latter, since the refinement of one zero is independent of the others, REFI can be applied concurrently to several subsets of the computed zeros on parallel computers or on clusters of workstations.

5. Numerical experiments.

We implemented REFI for the polynomials given below on a SUN Ultra 170E workstation. Our programs were written in the MATLAB language, which uses double precision numeric arithmetic. We first computed numerical zeros $\{\xi_i^{(0)}\}$ in the power basis by the MATLAB *roots* function. We then implemented REFI with $\hat{\xi}_{old}$ chosen in the desired region. The Taylor coefficients were computed symbolically using the MATLAB Symbolic Math Toolbox. There was no noticeable time delay for using symbolic formulations in our implementations.

Example 5.1. The square of "Wilkinson polynomial": $p(z) = \prod_{k=1}^{20}(z-k)^2$.

We first computed all 40 numerical zeros in the power basis. We then plotted them (marked by green o) together with $Z_\epsilon(p, B^{(0)})$ (marked by green curves) in figure 6.10 (left), in which $\epsilon = eps\|p\|_d$, $eps = 2.2 \times 10^{-16}$ is the MATLAB machine precision. As shown, 34 numerical zeros fell into the same connected component of $Z_\epsilon(p, B^{(0)})$, indicating that they could be ill-conditioned. Therefore, we implemented one iteration of REFI with $\hat{\xi}_{old} = 14$, because 14 is located roughly in the center of this suspect component. The roots of $f(y)$ were also computed using the MATLAB *roots* function, which generated 40 new numerical zeros of $p(z)$ (marked by blue x). Next, we plotted $Z_\epsilon(p, B^{(14)})$ (marked by blue curves) in the same figure. Among a total of 80 numerical zeros, those that lie inside the set $Z_\epsilon(p, B^{(0)}) \cap Z_\epsilon(p, B^{(14)})$ are better approximations. All 40 x's and 9 o's fell into this set. Among them, we picked 9 o's (because they are closer to $s = 0$ than those x's to $s = 14$ in the same region) and 31 x's that are closer to $s = 14$ as new approximations to the zeros of $p(z)$. figure 6.10 (right) shows the approximation errors before REFI (marked by green o), after one iteration of REFI (marked by blue x), and the new approximations (marked by dot-dash line).

Example 5.2. The monic polynomial with the n-fold zero $\xi = 10/11$: $p(z) = (z - 10/11)^n$.

We first computed n zeros $\{\xi_i^{(0)}\}$ in the power basis $B^{(0)}(z)$. We then arbitrarily picked an $\xi_i^{(0)}$ as $\hat{\xi}_{old}$ and implemented REFI with $tol = 10^{-15}$. The approximations at the k-th iteration are denoted by $\{\xi_i^{(k)}\}$. Table 5.1 gives, when the multiplicity of the roots increases, the number of refinements (k), the accuracy improvement made by REFI, and the computed ϵ that appears in the rate of convergence. In the 5-th column, we list $\delta_0 = \epsilon_0^{1/m}$, where $\epsilon_0 = 2^{-1022}$ is the underflow threshold for IEEE double precision (see [2, p. 11]), m is the multiplicity of the zero $\xi = 10/11$ (here, $m = n$). It is interesting to notice that the approximations achieved by REFI (the 4-th column) almost coincide with the m-th root of the machine underflow threshold, δ_0. This observation not only confirms the inequalities (4.1) proved by Proposition 4.1, but also further indicates that any numerical approximation $\hat{\xi}$ to an m-fold zero of $p(z)$, ξ, is subject to the constraint

$$O(|\hat{\xi} - \xi|) \geq \delta_0 = (\text{ the machine underflow threshold })^{1/m},$$

i.e., $|\hat{\xi} - \xi|^m = O(|p(\hat{\xi})|) > 0$ has to be machine representable. The numerical results suggest that REFI is able to improve computed polynomial zeros to the optimal numerical approximations under the given machine precision.

	No. of Refi.	Error before REFI	Error after REFI		Computed
n	k	$\min_i \lvert\xi_i^{(0)} - \xi\rvert$	$\min_i \lvert\xi_i^{(k)} - \xi\rvert$	δ_0	ϵ
10	12	4.4507e-02	1.1111e-16	1.7169e-31	1.5859e-13
20	27	2.9050e-01	1.7870e-15	4.1435e-16	1.0656e-11
30	32	2.4666e-01	1.5771e-11	5.5579e-11	1.0890e-10
40	35	7.3744e-01	8.0332e-09	2.0356e-08	3.5192e-10
50	37	9.2639e-01	3.3792e-07	7.0299e-07	8.0302e-10

TABLE 5.1
Error reduction for the zeros of $p(z) = (z - 10/11)^n$

For $n = 40$, figure 6.11 (left) shows $\{\xi_i^{(0)}\}$ (green o's), $\{\xi_i^{(k)}\}$ (x's, each color represents an iteration of REFI) obtained by the first few refinements ($k = 1, 2, \cdots$); figure 6.11 (right) shows $\{\xi_i^{(k)}\}$ (x's), obtained from the last few refinements ($k = \cdots, 34, 35$) and the exact multiple zero ξ (red +). Figure 6.12 shows $\min_i |\xi_i^{(k)} - \xi|$, the error reduction of REFI for $n = 40$. It confirms the linear convergence of the algorithm with the rate $\epsilon^{1/m}$.

We attempted to refine all zeros computed in the power basis, $\xi_i^{(0)}$, $i = 1, \cdots, n$, for a set of test polynomials. For a polynomial $p(z)$ satisfying

$$Z_\epsilon^{(\xi_i)}(p, B^{(0)}) \cap Z_\epsilon^{(\xi_j)}(p, B^{(0)}) = \emptyset, \qquad \forall i \neq j, \tag{5.1}$$

where $\epsilon = eps\|p\|_d$, and $\{\xi_i\}$ are exact zeros of $p(z)$, REFI is guaranteed to produce the optimal approximations to each ξ_i (in the meaning given by Example 5.2) by choosing $\hat{\xi}_{old} = \xi_i^{(0)}$. Table 5.2 gives numerical results of REFI on

Example 5.3. The scaled "Wilkinson polynomial": $p(z) = \prod_{k=1}^{20}(z - k/20)$.

	Error before REFI	Error after REFI	No. of Refi.
ξ_i	$\lvert\xi_i^{(0)} - \xi_i\rvert$	$\lvert\xi_i^{(k)} - \xi_i\rvert$	k
1/20	1.6453e-13	0	1
2/10	2.2462e-12	0	1
3/20	2.7104e-10	0	1
4/20	1.5041e-09	0	1
5/20	1.3142e-07	0	2
6/20	1.9093e-06	0	2
7/20	1.5435e-05	0	2
8/20	8.5185e-05	0	2
9/20	3.5601e-04	0	2
10/20	1.1012e-03	0	2
11/20	2.9596e-03	0	2
12/20	5.2658e-03	0	2
13/20	9.2826e-03	0	2
14/20	1.0416e-02	0	2
15/20	8.8301e-03	0	2
16/20	6.5150e-03	0	2
17/20	2.7471e-03	0	1
18/20	9.8058e-04	0	1
19/20	1.9272e-04	1.1102e-16	1
20/20	1.8238e-05	0	1

TABLE 5.2
Error reduction for the zeros of $p(z) = \prod_{k=1}^{20}(z - k/20)$.

For polynomials that fail to meet condition (5.1), i.e., when there exists $Z_\epsilon^{(\xi_i)}(p, B^{(0)})$ that contains more than one distinct zero of $p(z)$, REFI is able to generate at least one optimal approximation to a zero $\xi_j \in Z_\epsilon^{(\xi_i)}(p, B^{(0)})$. Conceptually, the rest of the zeros in $Z_\epsilon^{(\xi_i)}(p, B^{(0)})$ can be approximated to their optimal approximations by removing the accepted approximation of ξ_j in the manner similar to the approaches of [8, 12, 15], and then applying REFI to the deflated polynomial. However, we encounted numerical difficulties in deflation stability for high-degree polynomials with multiple zeros, e.g., $p(z)$ in Example 5.1. Since polynomial deflation is a challenging research topic of its own, and is beyond the consideration of this paper, we would leave it for future investigation.

6. Concluding remarks.

- REFI is independent of the zero-finding algorithm by which the original zero approximations $\{\xi_i^{(0)}\}$ are computed, and can be adapted to any numerical scheme that is used for computing zeros of $f(y)$ at Step (ii) to reduce the approximation errors in $\{\xi_i^{(0)}\}$.
- REFI is a local algorithm mixing symbolic and numerical computations. For low-degree polynomials, computational speed is not a concern. The high-degree poly-

nomials encountered in practice are usually sparse. Moreover, only a small subset of their zero set is generally of interest (see [7]). In this situation, REFI could be implemented at reasonable cost by exploring sparse data structure at Step (i) and by adopting more efficient numerical local zero-finding algorithm at Step (ii). Readers may refer to [11] for available local zero-finding methods.

Acknowledgements. I would like to thank Prof. James Madden of Louisiana State University for interesting discussions and careful reading of the manuscript.

REFERENCES

[1] D. Bini and V. Pan, *Polynomial and Matrix Computations, Vol. 1: Fundamental Algorithms*, Birkhäuser, Boston, Cambridge, MA, 1994.

[2] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.

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

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

[5] ———, *Algorithms for polynomials in Bernstein form*, Comput. Aided Geom. Design, 5 (1988), pp. 1–26.

[6] W. Gautschi, *Questions of numerical condition related to polynomials*, in Studies in Numerical Analysis, G. H. Golub, ed., MAA Studies in Math., 24 (1984), pp. 140–177.

[7] L. Gonzalez-Vega, *The needs of industry for polynomial system solving*, FRISCO Consortium (LTR 21.024), 1997.

[8] M. A. Jenkins and J. Traub, *Algorithm 419: Zeros of a complex polynomial*, Comm. ACM, 15 (1972), pp. 97–99.

[9] J. McNamee, *A bibliography on roots of polynomials*, J. Comput. Appl. Math., 47 (1993), pp. 391–394.

[10] R. Mosier, *Root neighborhoods of a polynomial*, Math. Comp., 47 (1986), pp. 265–273.

[11] V. Y. Pan, *Solving a polynomial equation: some history and recent progress*, SIAM Rev., 39 (1997), pp. 187–220.

[12] B. Parlett, *Laguerre's method applied to the matrix eigenvalue problem*, Math. Comp., 18 (1964), pp. 464–485.

[13] K. C. Toh and L. N. Trefethen, *Pseudozeros of polynomials and pseudospectra of companion matrices*, Numer. Math., 68 (1994), pp. 403–425.

[14] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997.

[15] J. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs. NJ, 1963.

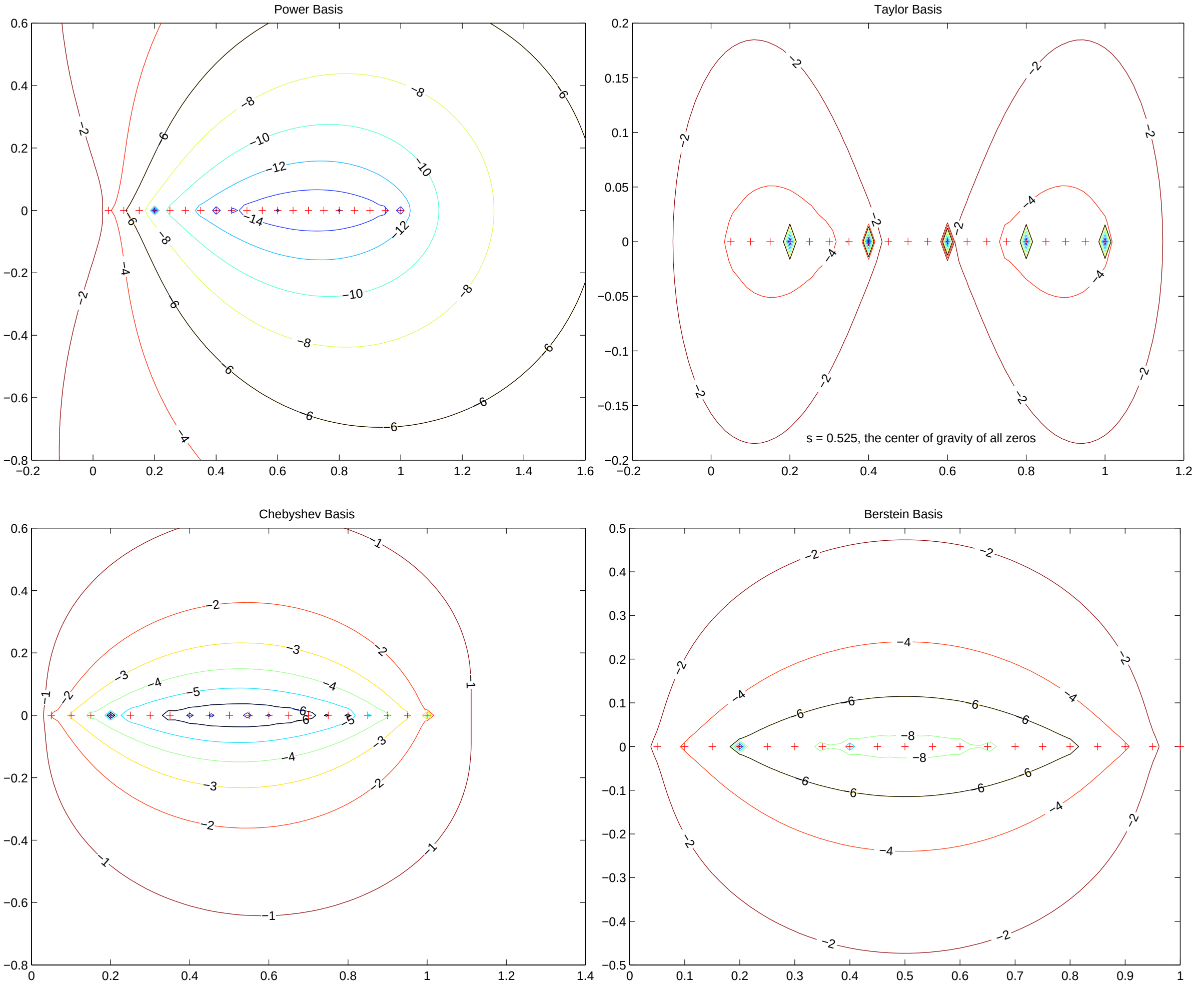


FIG. 6.1. *Pseudozero sets for the scaled "Wilkinson polynomial":* $p(z) = \prod_{k=1}^{20}(z - k/20)$.

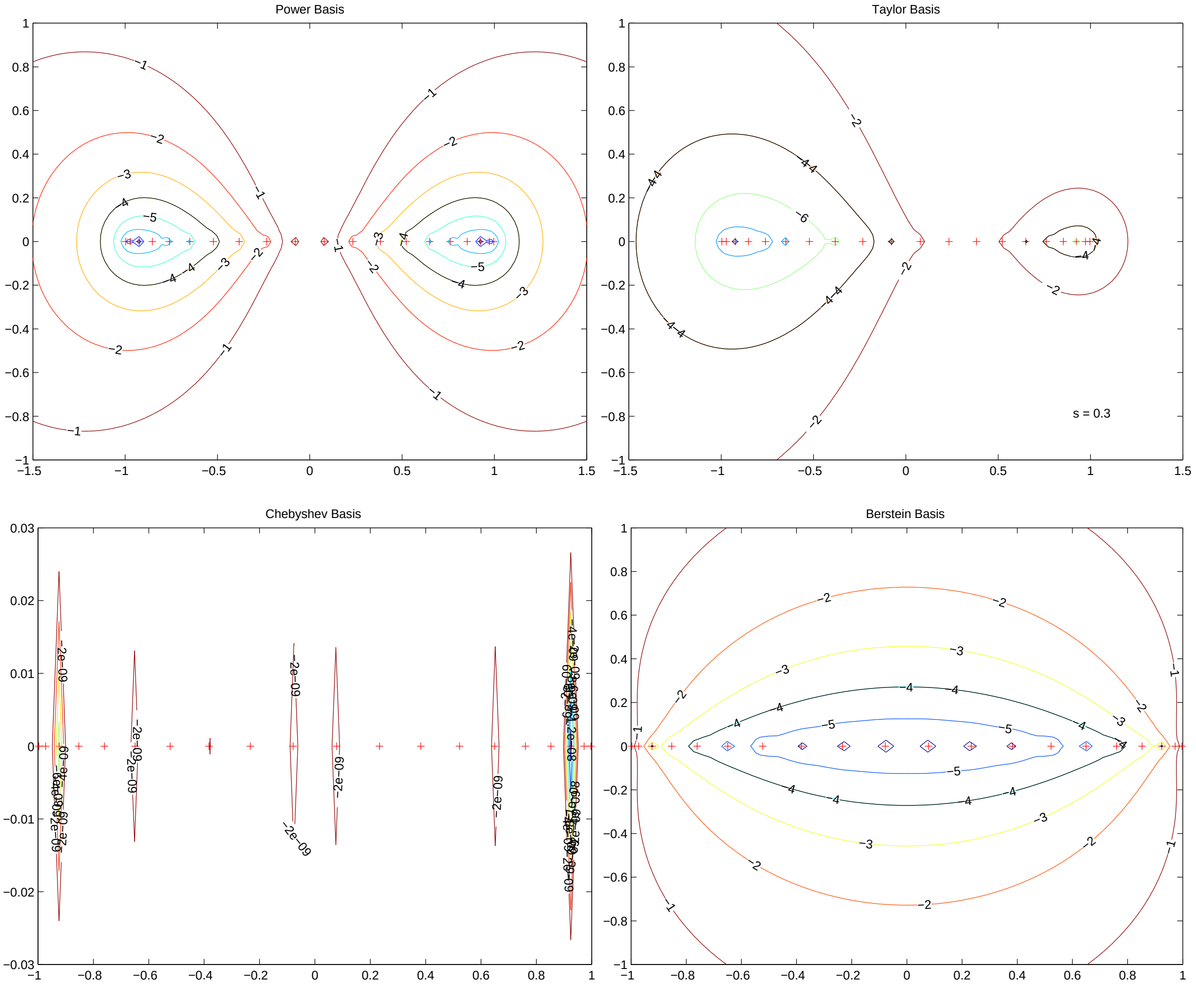


FIG. 6.2. *Pseudozero sets for the Chebyshev polynomial of degree 20.*

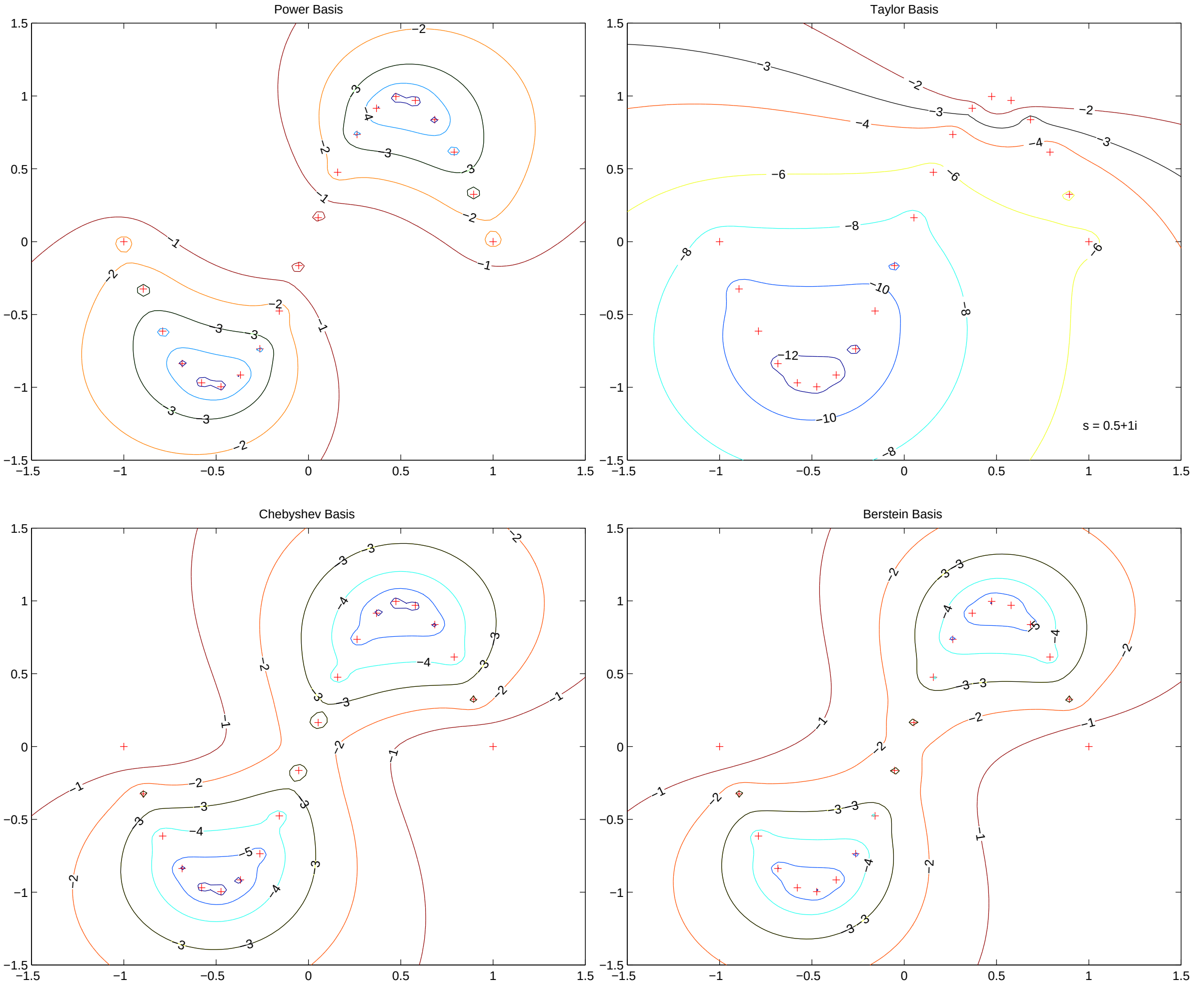


FIG. 6.3. *Pseudozero sets for the monic polynomial with zeros equally spaced on the curve* $z = x + i\sin(\pi x)$, $-1 \le x \le 1$.

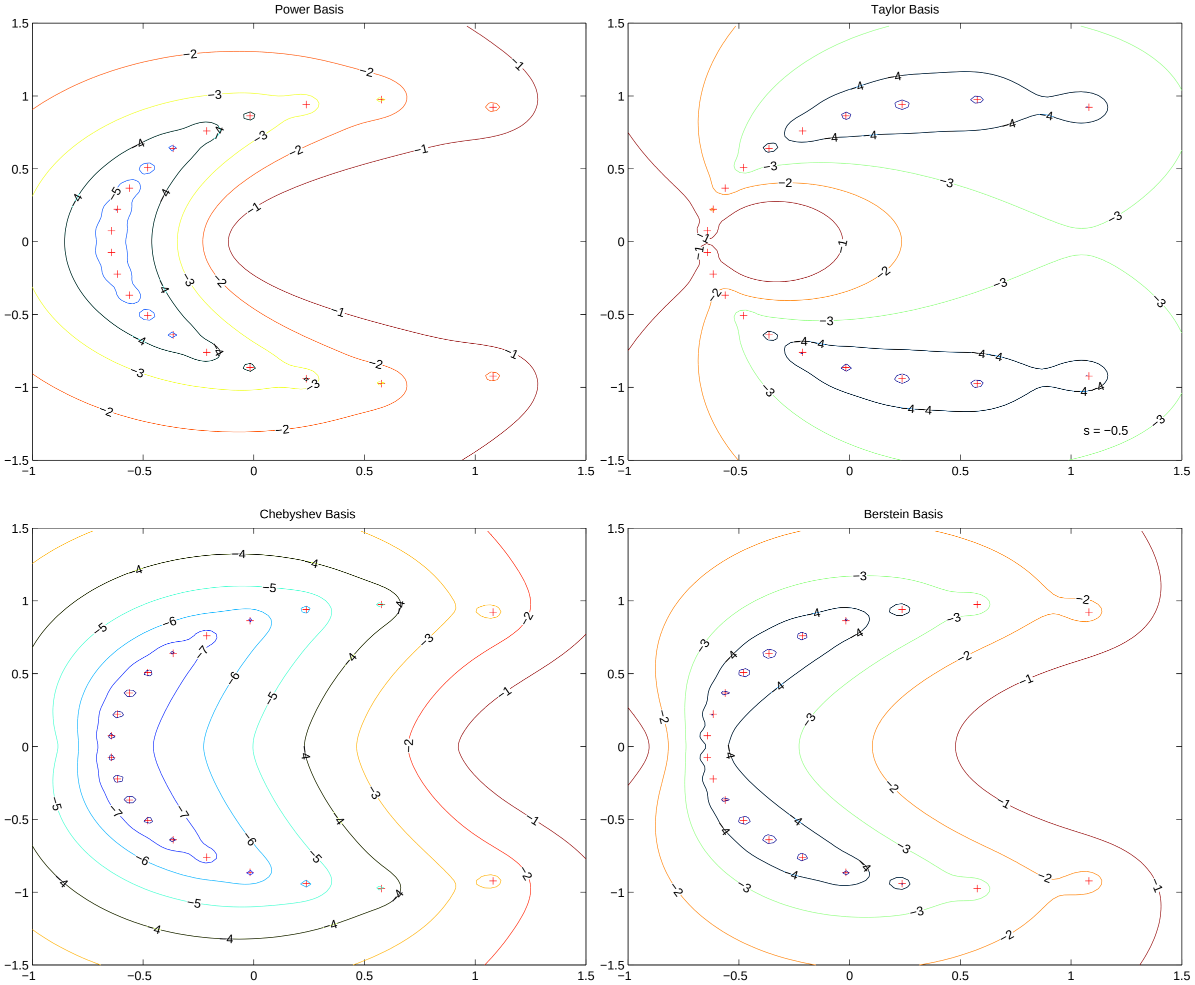


FIG. 6.4. *Pseudozero sets for* $p(z) = \sum_{k=0}^{20} (10z)^k / k!$.

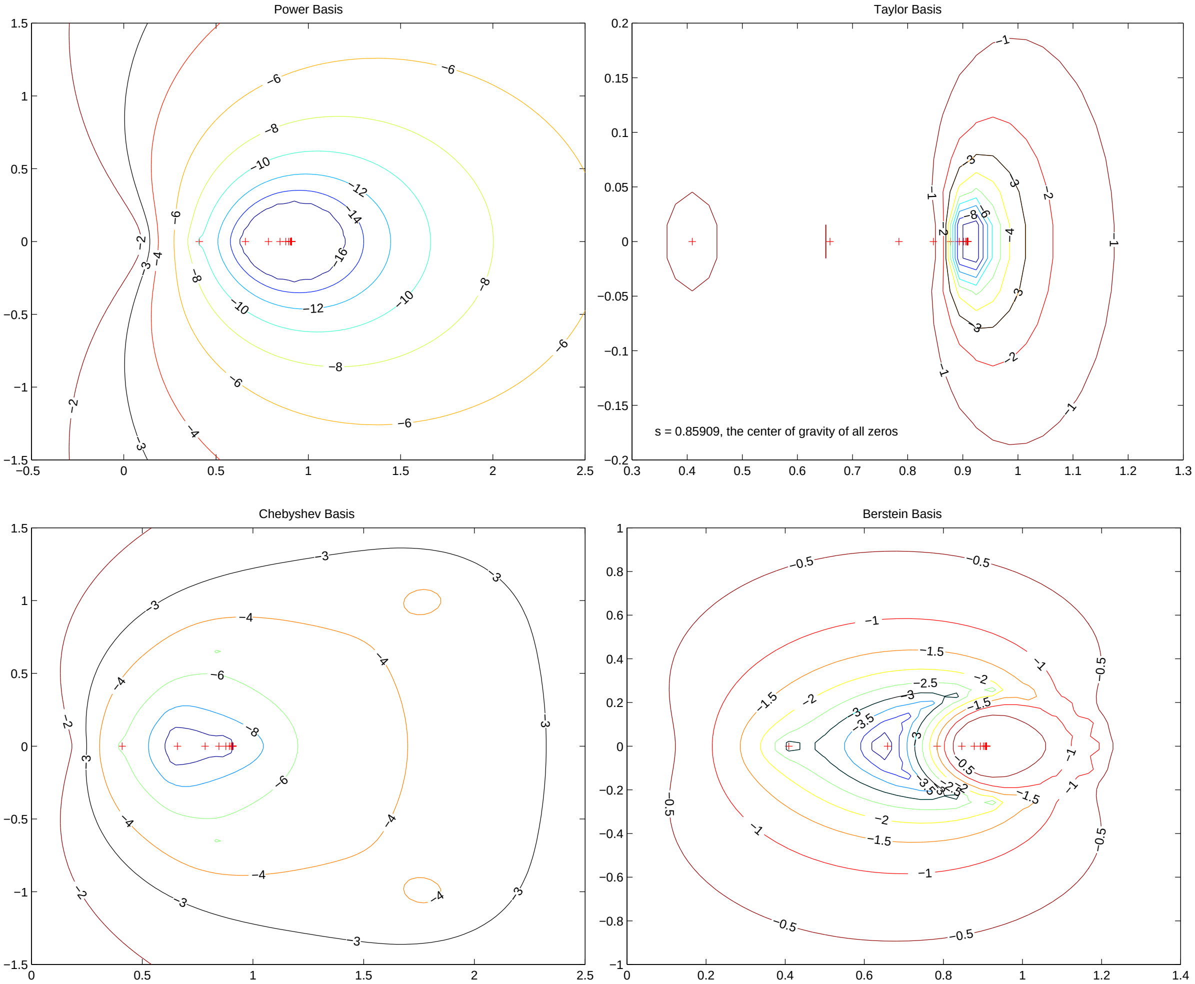


FIG. 6.5. *Pseudozero sets for the monic polynomial with zeros:* $10/11 - 2^{-k}$, $k = 1, 2, \cdots, 20$.

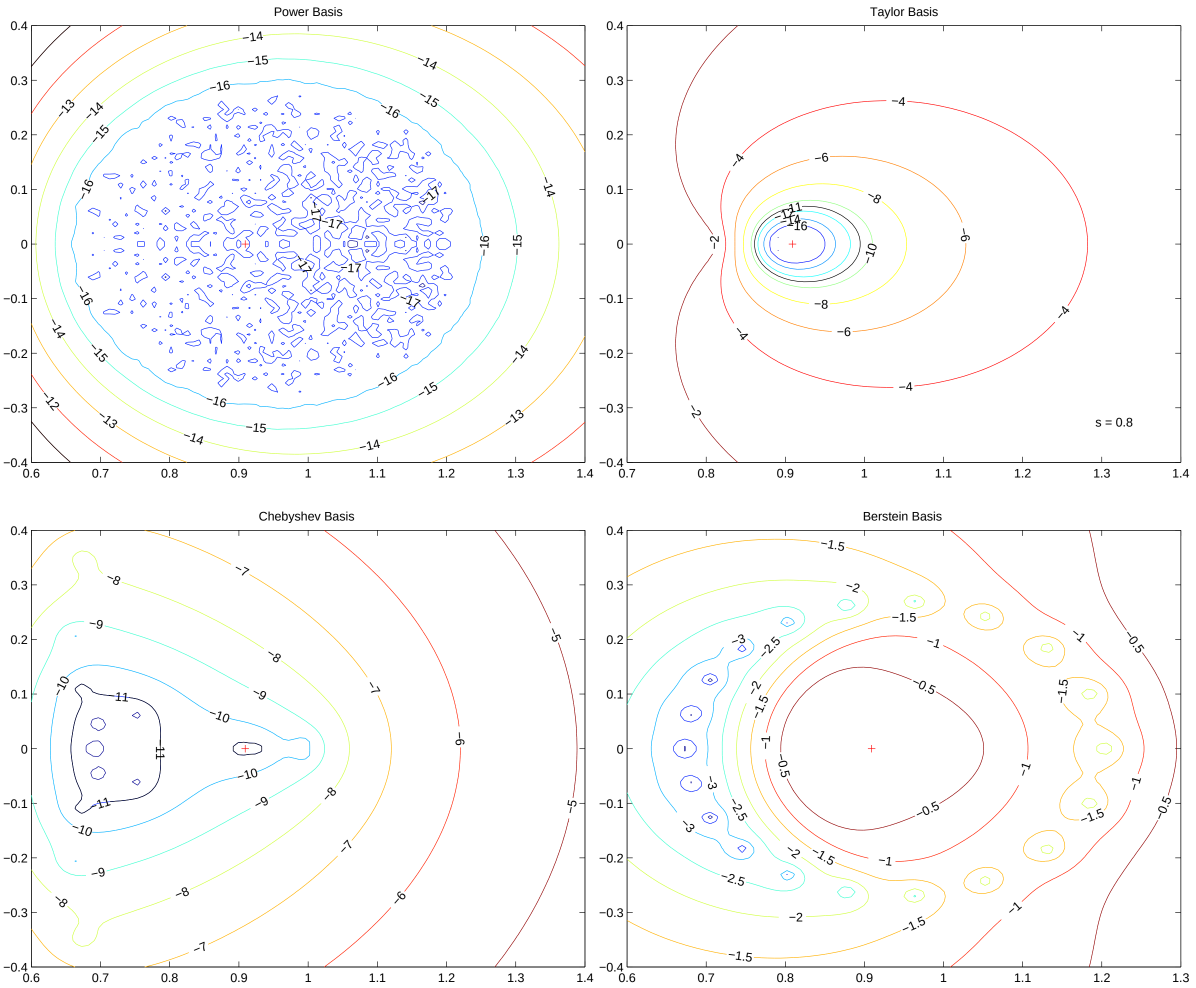
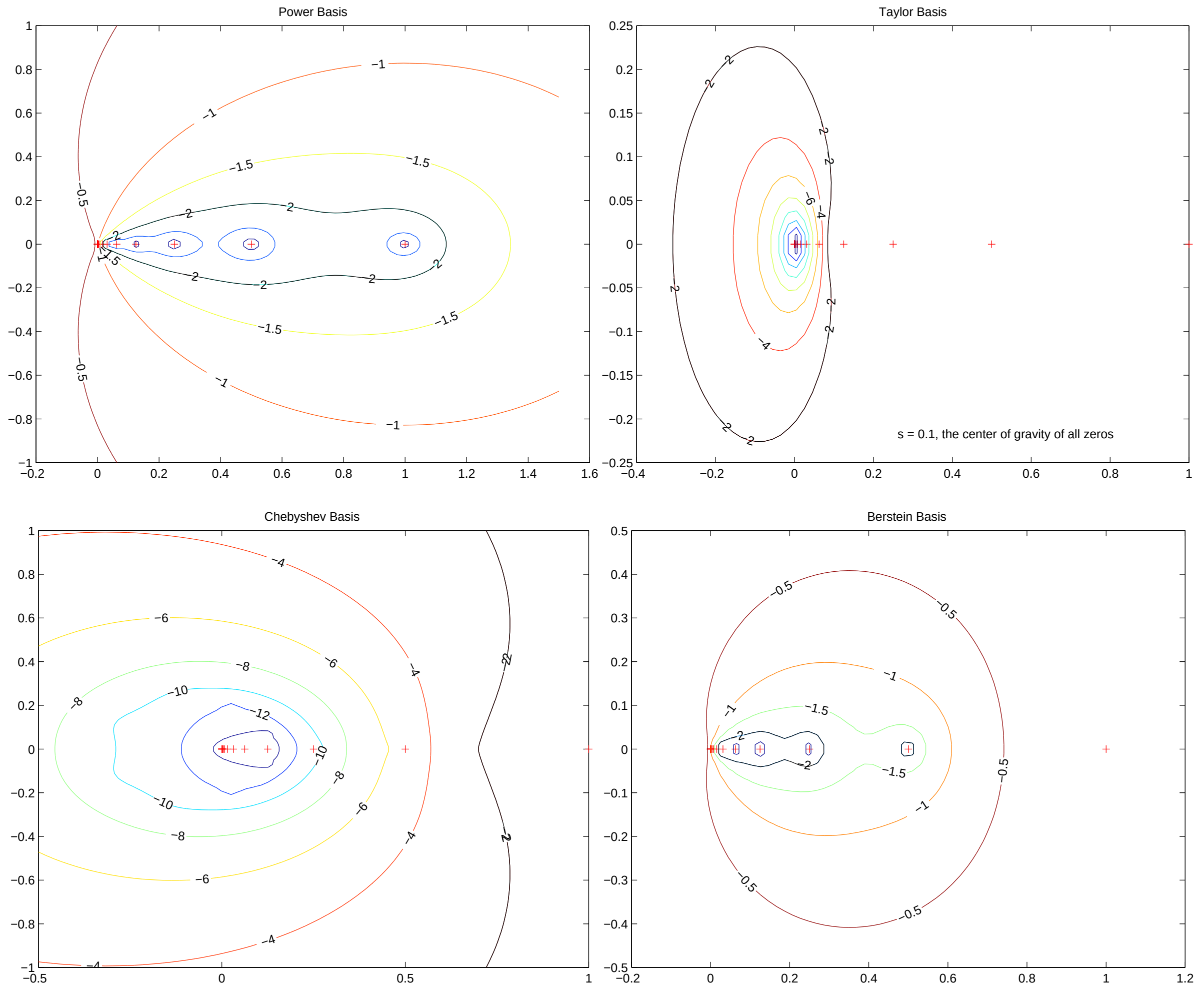


FIG. 6.6. *Pseudozero sets for* $p(z) = (z - 10/11)^{20}$.



Power Basis

Taylor Basis

Chebyshev Basis

Berstein Basis

FIG. 6.7. *Pseudozero sets for the monic polynomial with zeros:* $2^{-19}, 2^{-18}, \cdots, 1$.

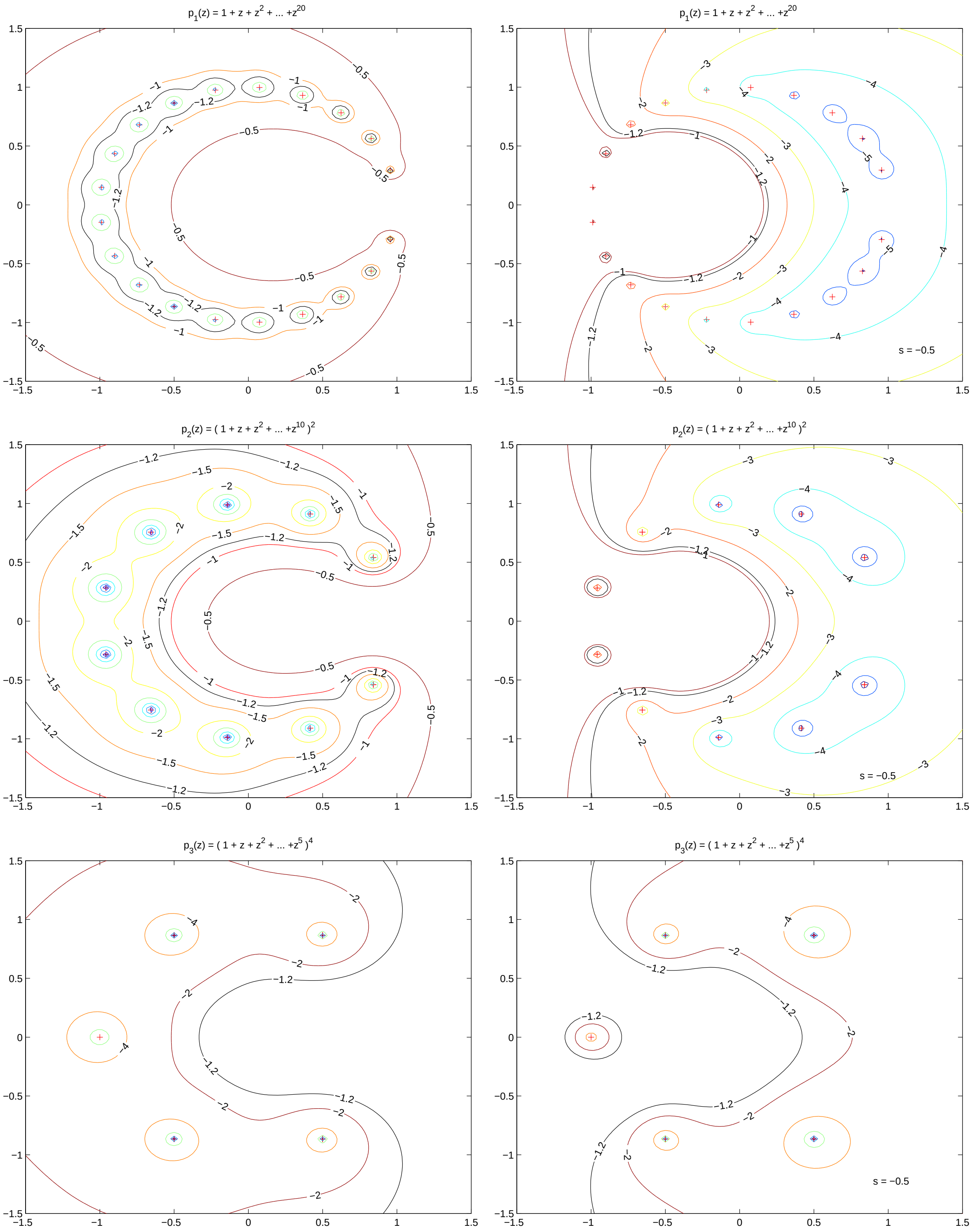
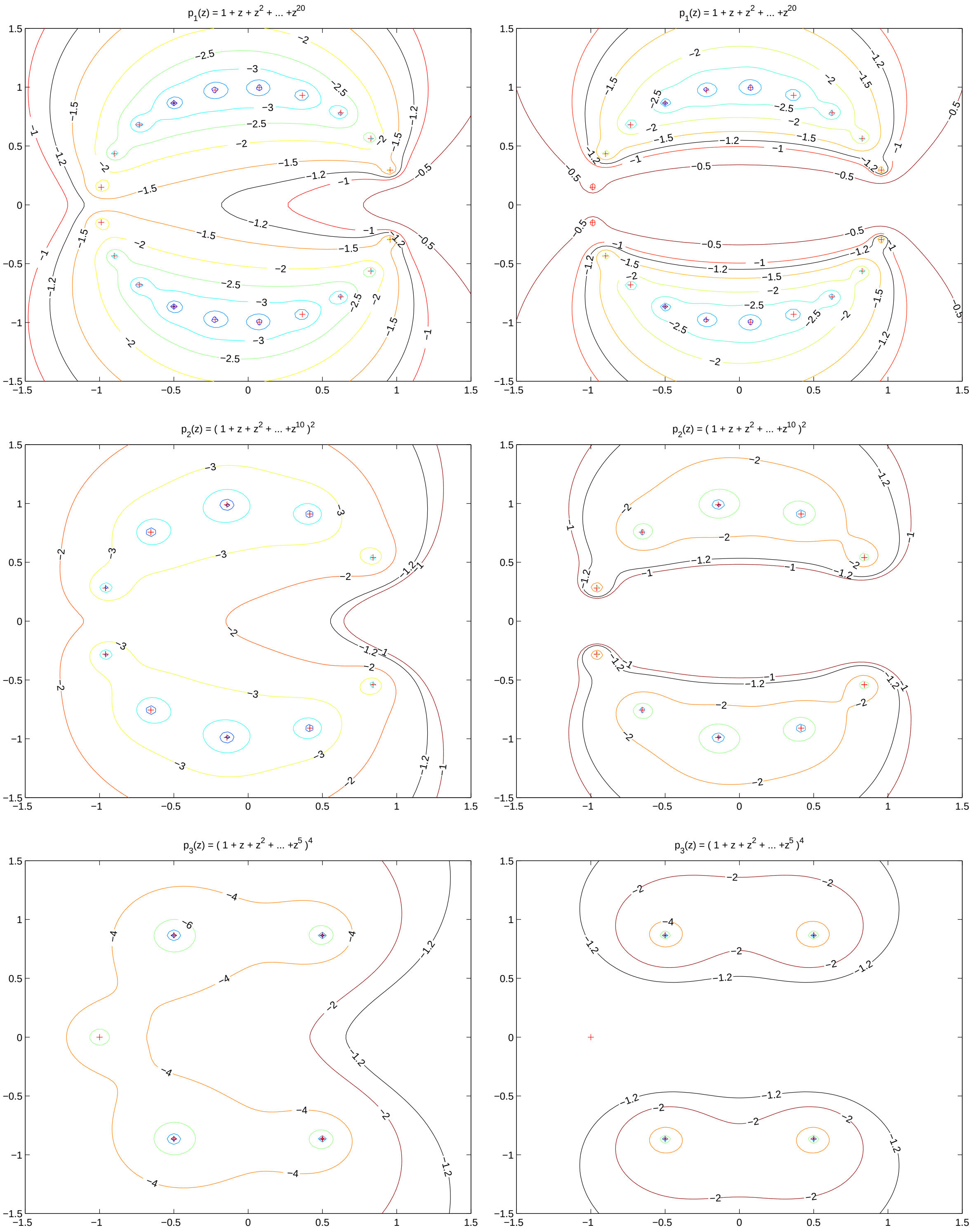


FIG. 6.8. *Pseudozero sets for the polynomials* $p_1(z) = 1 + z + z^2 + \cdots + z^{20}$, $p_2(z) = (1 + z + z^2 + \cdots + z^{10})^2$, *and* $p_3(z) = (1 + z + z^2 + \cdots + z^5)^4$. *Left-hand column: in the power basis; Right-hand column: in the Taylor basis.*



$p_1(z) = 1 + z + z^2 + \dots + z^{20}$

$p_1(z) = 1 + z + z^2 + \dots + z^{20}$

$p_2(z) = (1 + z + z^2 + \dots + z^{10})^2$

$p_2(z) = (1 + z + z^2 + \dots + z^{10})^2$

$p_3(z) = (1 + z + z^2 + \dots + z^5)^4$

$p_3(z) = (1 + z + z^2 + \dots + z^5)^4$

FIG. 6.9. *Pseudozero sets for the polynomials* $p_1(z) = 1 + z + z^2 + \cdots + z^{20}$, $p_2(z) = (1 + z + z^2 + \cdots + z^{10})^2$, *and* $p_3(z) = (1 + z + z^2 + \cdots + z^5)^4$. *Left-hand column: in the Chebyshev basis; Right-hand column: in the Bernstein basis.*

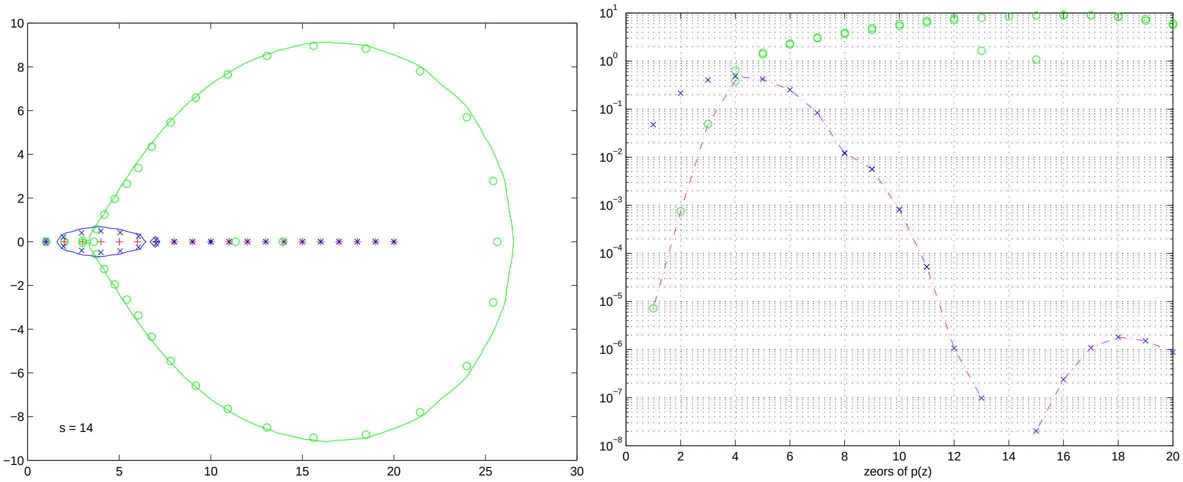


FIG. 6.10. $p(z) = \prod_{k=1}^{20}(z-k)^2$.
Left: exact zeros (red +); computed zeros in the basis $B^{(0)}$ (green o) and $Z_\epsilon(p, B^{(0)})$ (green curves); computed zeros in the basis $B^{(14)}$ (blue x) and $Z_\epsilon(p, B^{(14)})$ (blue curves).
Right: approximation errors in the basis $B^{(0)}$ (green o), in the basis $B^{(14)}$ (blue x), and the new approximation errors (dot-dash line).

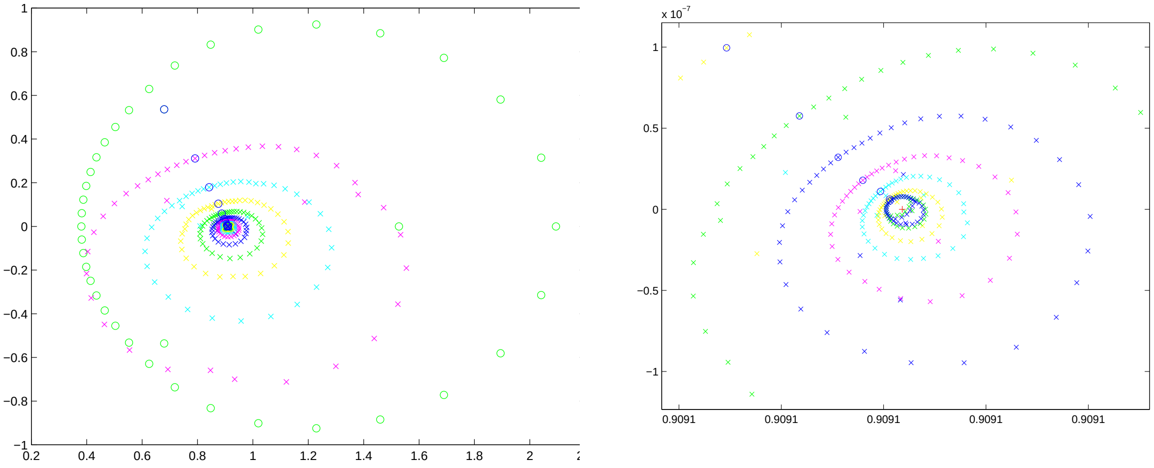


FIG. 6.11. $p(z) = (z - 10/11)^{40}$.
Left: exact zero $\xi = 10/11$ (red +); computed zeros in the basis $B^{(0)}$ (green o) and improved zeros obtained by the first few refinements, $k = 1, 2, \cdots$, (x's, each color represents an iteration of REFI).
Right: exact zero $\xi = 10/11$ (red +); improved zeros obtained from the last few refinements, $k = \cdots, 34, 35$, (x's, each color represents an iteration of REFI).

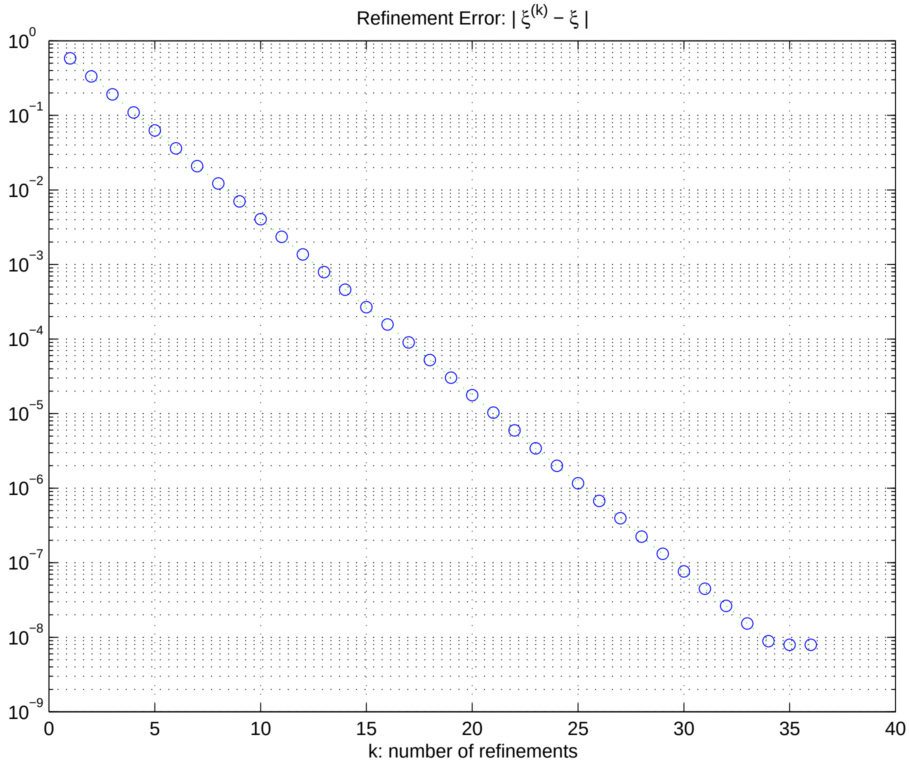


FIG. 6.12. *Error reduction of REFI for* $p(z) = (z - 10/11)^{40}$.