Certified evaluations of Hölder continuous functions at roots of polynomials

Parker B. Edwards[1][0000−0001−6875−5328],
Jonathan D. Hauenstein[1][0000−0002−9252−8210], and
Clifford D. Smyth[2][0000−0003−1486−7900]

[1] Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 {parker.edwards,hauenstein}@nd.edu
https://sites.nd.edu/parker-edwards, https://www.nd.edu/~jhauenst
[2] Department of Mathematics and Statistics, University of North Carolina at Greensboro, Greensboro, NC 27402 cdsmyth@uncg.edu
https://www.uncg.edu/~cdsmyth/

Abstract. Various methods can obtain certified estimates for roots of polynomials. Many applications in science and engineering additionally utilize the value of functions evaluated at roots. For example, critical values are obtained by evaluating an objective function at critical points. For analytic evaluation functions, Newton's method naturally applies to yield certified estimates. These estimates no longer apply, however, for Hölder continuous functions, which are a generalization of Lipschitz continuous functions where continuous derivatives need not exist. This work develops and analyzes an alternative approach for certified estimates of evaluating locally Hölder continuous functions at roots of polynomials. An implementation of the method in `Maple` demonstrates efficacy and efficiency.

Keywords: Roots of polynomials · Hölder continuous functions · Certified evaluations.

1 Introduction

For a univariate polynomial $p(x)$, the Abel-Ruffini theorem posits that the roots cannot be expressed in terms of radicals for general polynomials of degree at least 5. A simple illustration of this is that the solutions of the quintic equation

$$p(x) = x^5 - x - 1 = 0 \tag{1}$$

cannot be expressed in radicals. Thus, a common technique is to compute numerical approximations with certified bounds for the roots of a polynomial. Some approaches based on Newton's method are the Kantorovich theorem [7] and Smale's α-theory [19]. Kantorovich's approach is based on bounds for a twice-differentiable function in an open set while Smale's approach only uses local estimates at one point coupled with the analyticity of the function. Certified approximations of roots of polynomials can also be obtained using interval methods such as [4,8,13,18] along with the Krawczyk operator [10,12].

Although computing certified estimates for roots of polynomials is important, many applications in science and engineering utilize the roots in further computations. As an illustrative example, consider the optimization problem

$$\min\{21x^8 - 42x^4 - 56x^3 + 3 \ : \ x \in \mathbb{R}\}. \tag{2}$$

For this problem, the global minimum is the minimum of the critical values which are obtained by evaluating the objective function $g(x) = 21x^8 - 42x^4 - 56x^3 + 3$ at its critical points, i.e. at the real roots of $g'(x) = 168x^2(x^5 - x - 1)$. Since the quintic in (1) is a factor of $g'(x)$, only approximations of the roots of g' can be computed. One must translate these approximate roots to certified evaluations of the objective function $g(x)$ evaluated at the roots of the polynomial $g'(x)$ to obtain certified bounds on the global minimum of (2).

One approach for computing a certified evaluation of $f(x)$ at roots of a polynomial $p(x)$ is via certified estimates of solutions to the multivariate system $p(x) = y - f(x) = 0$. For sufficiently smooth f, approaches based on Newton's method generate certified estimates. When f is not differentiable, one can alternatively follow a two-stage procedure: first, certifiably estimate a root of $p(x)$ to error at most ϵ and then use interval evaluation methods, e.g. see [13, Chap. 5], to compute a certified estimate of $f(x)$ evaluated at the root. Such an approach provides direct control on the approximation error of a root of $p(x)$ but not on the output evaluation error of $f(x)$ which will typically be larger than ϵ.

The approach in this paper considers certified evaluations of locally Hölder continuous functions at roots of polynomials and links the desired output of the certified evaluation with the error in the approximation of the root. Hölder continuous functions are a generalization of Lipschitz functions which are indeed continuous, but need not be differentiable anywhere, e.g., see Section 5.3. Moreover, satisfying the local Hölder continuity condition does not guarantee that a function can be evaluated exactly for, say, rational input. Therefore, our approach also incorporates numerical evaluation error into the certified bounds.

The rest of the paper is organized as follows. Section 2 describes the necessary analysis of locally Hölder continuous functions, with a particular focus on polynomials and rational functions. Section 3 summarizes the approach used for developing certified bounds on roots of polynomials. Section 4 combines the certification of roots and evaluation bounds on Hölder continuous functions yielding our approach for computing certified evaluations. Section 5 presents information regarding the implementation in `Maple` along with several examples demonstrating its efficacy and efficiency. Section 6 applies the techniques developed for certified evaluations to prove non-negativity of coefficients arising in a series expansion of a rational function. The paper concludes in Section 7.

2 Hölder continuous functions

The following describes the collection of functions under consideration.

Definition 1. *A function* $f : \mathbb{C} \to \mathbb{C}$ *is* locally Hölder continuous *at a point* $x^* \in \mathbb{C}$ *if there exist positive real constants* ϵ, C, α *such that*

$$|f(x^*) - f(y)| \leq C \cdot |x^* - y|^\alpha \leq C \cdot \epsilon^\alpha \tag{3}$$

for all $y \in B(x^*, \epsilon)$ *where* $B(x^*, \epsilon) = \{z \in \mathbb{C} \ : \ |z - x^*| \leq \epsilon\}$. *In this case,* $f(x)$ *is said to have* Hölder constant C *and* Hölder exponent α *at* x^*. *Moreover, if* $\alpha = 1$, *then* $f(x)$ *is said to be* Lipschitz continuous *at* x^* *with* Lipschitz constant C.

Functions which are locally Hölder continuous at a point are clearly continuous at that point and the error bound provided in (3) will be exploited in Section 4 to provide certified evaluations. Every function $f(x)$ which is continuously differentiable in a neighborhood of x^* is locally Hölder continuous with $\alpha = 1$, i.e., locally Lipschitz continuous. For $n \geq 1$, $f(x) = \sqrt[n]{|x|}$ is continuous but not differentiable at $x^* = 0$. It is locally Hölder continuous at $x^* = 0$ with Hölder constant $C = 1$ and Hölder exponent $\alpha = 1/n$.

A computational challenge is to determine a Hölder constant C and Hölder exponent α for $f(x)$ on $B(x^*, \epsilon)$ given $f(x)$, x^*, and $\epsilon > 0$. Sections 2.1 and 2.2 describe a strategy for polynomials and rational functions, respectively.

2.1 Polynomials

Since every polynomial $f(x)$ is continuously differentiable, we can take the Hölder exponent to be $\alpha = 1$ at any point x^*. However, the Hölder constant C depends upon x^* and ϵ. The Fundamental Theorem of Calculus shows that one just needs

$$C \geq \max_{y \in B(x^*, \epsilon)} |f'(y)|. \tag{4}$$

Although one may attempt to compute this maximum directly, the Taylor series expansion of $f'(x)$ at x^* provides an easy to compute upper bound. If $d = \deg f$, $f'(x) = \sum_{i=1}^{d} \frac{f^{(i)}(x^*)}{(i-1)!}(x - x^*)^{i-1}$ so that the triangle inequality yields the bound

$$C := \sum_{i=1}^{d} \frac{|f^{(i)}(x^*)|}{(i-1)!}\epsilon^{i-1} \geq \max_{y \in B(x^*, \epsilon)} |f'(y)|. \tag{5}$$

2.2 Rational functions

The added challenge with a rational function $f(x)$ is to ensure that it is defined on $B(x^*, \epsilon)$. One may attempt to compute the poles of $f(x)$ and ensure that none are in $B(x^*, \epsilon)$. However, the implementation in Section 5 is based on a local approach that also enables computing local upper bounds on $|f'(x)|$. For $f(x) = a(x)/b(x)$, one can prove $b(y) \neq 0$ for all $y \in B(x^*, \epsilon)$ by showing that $|b(x^*)| > |b(y) - b(x^*)|$ for all $y \in B(x^*, \epsilon)$. If $d_b = \deg b$, then

$$|b(y) - b(x^*)| = \left| \sum_{i=1}^{d_b} \frac{b^{(i)}(x^*)}{i!}(y - x^*)^i \right| \leq \sum_{i=1}^{d_b} \frac{|b^{(i)}(x^*)|}{i!}\epsilon^i.$$

Therefore, a certificate that $f(x)$ is continuously differentiable on $B(x^*, \epsilon)$ is

$$|b(x^*)| > \sum_{i=1}^{d_b} \frac{|b^{(i)}(x^*)|}{i!} \epsilon^i$$

yielding

$$\min_{y \in B(x^*,\epsilon)} |b(y)| \geq |b(x^*)| - \sum_{i=1}^{d_b} \frac{|b^{(i)}(x^*)|}{i!} \epsilon^i > 0. \tag{6}$$

When $b(x^*) \neq 0$, it is clear that one can always take ϵ small enough to satisfy (6).

When $f(x)$ is continuously differentiable on $B(x^*, \epsilon)$, then one can take the Hölder exponent $\alpha = 1$ and the Hölder constant C as in (4). Hence,

$$\max_{y \in B(x^*,\epsilon)} |f'(x)| \leq \frac{\max\limits_{y \in B(x^*,\epsilon)} |a'(y)|}{\min\limits_{y \in B(x^*,\epsilon)} |b(y)|} + \frac{\max\limits_{y \in B(x^*,\epsilon)} |a(y)| \cdot \max\limits_{y \in B(x^*,\epsilon)} |b'(y)|}{\min\limits_{y \in B(x^*,\epsilon)} |b(y)|^2}$$

where the maxima can be upper bounded similar to (5) and the minimum can be lower bounded using (6).

3 Certification of roots

The initial task of determining certified evaluation bounds at roots of a given polynomial is to compute certified bounds of the roots. From a theoretical perspective, we assume that we know the polynomial $p(x)$ exactly. From a computational perspective, we assume that $p(x)$ has rational coefficients, i.e., $p(x) \in \mathbb{Q}[x]$. The certification of roots of $p(x)$ can thus be performed using `RealRootIsolate` based on [3,15,20,21,22] in `Maple` as follows.

Since $p(x)$ is known exactly, we can first reduce down to the irreducible case with multiplicity 1 roots by computing an irreducible factorization of $p(x)$, say

$$p(x) = p_1(x)^{r_1} \cdots p_s(x)^{r_s}$$

where $p_1, \ldots, p_s$ are irreducible with corresponding multiplicities $r_1, \ldots, r_s \in \mathbb{N}$. For $p(x) \in \mathbb{Q}[x]$, `factor` in `Maple` computes the irreducible factors in $\mathbb{Q}[x]$, i.e., each $p_i(x) \in \mathbb{Q}[x]$. If $z \in \mathbb{C}$ is a root of $p_j(x)$, then z has multiplicity 1 with respect to $p_j(x)$, i.e., z is a simple root of $p_j(x)$ with $p_j(z) = 0$ and $p_j'(z) \neq 0$. In contrast, z has multiplicity r_j with respect to $p(x)$. Note that one could alternatively use a squarefree factorization with appropriate modifications.

For each irreducible factor $q := p_j$, one computes certified approximations of each root. Although methods over $\mathbb{C}$ are more efficient [2,14], we utilize the `RealRootIsolate` function in `Maple` by transforming the domain $\mathbb{C}$ into $\mathbb{R}^2$ via

$$q(x + iy) = q_r(x, y) + i \cdot q_i(x, y) \qquad \text{where } i = \sqrt{-1}. \tag{7}$$

Therefore, solving $q = 0$ on $\mathbb{C}$ corresponds with solving $q_r = q_i = 0$ on $\mathbb{R}^2$. Applying `RealRootIsolate` with an optional absolute error bound `abserr` that will

be utilized later guarantees as output isolating boxes for every real solution to $q_r = q_i = 0$ on $\mathbb{R}^2$. Therefore, looping over the irreducible factors of p, one obtains certified bounds for every root z of $p(x)$ in $\mathbb{C}$ of the form $a_1 \leq \text{real}(z) \leq a_2$ and $b_1 \leq \text{imag}(z) \leq b_2$ where $a_1, a_2, b_1, b_2 \in \mathbb{Q}$.

4 Certified evaluations

Combining information on Hölder continuous functions from Section 2 and certification of roots of polynomials from Section 3 yields the following approach to develop certified evaluations. With input a polynomial $p(x)$, a Hölder continuous function $f(x)$ which is defined at each root of $p(x)$, and an error bound $\epsilon > 0$, the goal is to develop an approach that computes an approximation of $f(z)$ within ϵ for each root z of $p(x)$. Since one may not be able to evaluate $f(x)$ exactly, we incorporate an evaluation error of $\delta \in (0, \epsilon)$. Typically, δ can be decreased by utilizing higher precision computations. For rational input, rounding to produce finite decimal representations constitutes the only source of representation error in `Maple`. Our implementation utilizes enough digits to have $\delta = \epsilon/10$.

The first step is to utilize Section 3 to determine initial certified bounds for each root of $p(x)$. For an initial error bound on the roots, we start with $\gamma = \epsilon/2$ and approximate each root z by x^* with $|z - x^*| < \gamma$. Certified evaluations are then obtained root by root since the Hölder constants are dependent upon local information near each root. In particular, the next step is to compute a Hölder exponent α and Hölder constant C that is valid on the ball $B(x^*, 2\gamma)$. If this is not possible, e.g., if $f(x)$ cannot be certified to be defined on $B(x^*, 2\gamma)$, one can simply reduce γ, e.g., by replacing γ by $\gamma/2$ and repeating the process using a newly computed certified approximation of z. Since $f(x)$ is defined at each root of $p(x)$, such a loop must terminate.

The final step is to utilize local information to compute a new approximation of root z that will produce a certified evaluation within ϵ. Consider μ such that

$$0 < \mu \leq \min\left\{\gamma, \sqrt[\alpha]{\frac{\epsilon - \delta}{C}}\right\}$$

and z^* an approximation of z such that $z \in B(z^*, \mu)$. Since $|x^* - z^*| \leq 2\gamma$, we have $B(z^*, \mu) \subset B(x^*, 2\gamma)$ so that all of the Hölder constants are valid on $B(z^*, \mu)$. Hence, all that remains is to compute a certified approximation of $f(z^*)$, say f^*, within the evaluation error of δ since

$$|f^* - f(z)| \leq |f^* - f(z^*)| + |f(z^*) - f(z)| \leq \delta + C \cdot |z^* - z|^\alpha \leq \delta + C \cdot \mu^\alpha \leq \epsilon.$$

Remark 1. Software packages that deal with numerical approximations express numerical estimates either in terms of absolute or relative error. The choice is typically made based on the intended application. For example, relative error is more easily associated with memory requirements for storing approximations regardless of the magnitude of a value being approximated so that control over relative error in approximations is often easier than controlling absolute error.

We have chosen to bound the absolute error here since the application of our Hölder function based approach presented in Section 6 is simpler to state in terms of absolute error.

Remark 2. When f is polynomial, one could use the built-in `Maple` function `RootFinding[Isolate]` which implements various root isolation strategies based on [1,9,16,17,20]. The `Maple` function call is:

```
RootFinding[Isolate]([p_r,p_i],[x,y],constraints=[f],
                     digits=ceil(-log[10](eps)))
```

where p_r and p_i are the real and imaginary parts of $p(x+iy)$, respectively. The outputs estimate both the roots of p and evaluations of f at those roots. One principal difference between our approach and the built-in functionality is that `RootFinding[Isolate]` is only implemented for polynomial evaluation functions. See Section 5.1 for a polynomial example using `RootFinding[Isolate]`.

Example 1. As an illustration, consider evaluating the Cantor ternary function

$$f\left(\sum_{j=1}^{\infty}\frac{a_j}{3^j}\right)=\frac{1}{2^N}+\frac{1}{2}\sum_{j=1}^{N-1}\frac{a_j}{2^j}\quad\text{where}\quad\begin{cases}a_j\in\{0,1,2\}\text{ for }j=1,2,\ldots,\\ N=\min\{j\mid a_j\text{ is odd}\}\in\mathbb{Z}_{>0}\cup\{\infty\}.\end{cases}$$

at the unique root $z\in[0,1]$ of the polynomial

$$p(x)=\left(\frac{3}{2}\right)^{101}x^5+17\left(\frac{3}{2}\right)^{101}x-1$$

with error $\epsilon=10^{-16}$. Figure 1 plots the Cantor ternary function on the domain $[0,1]$ along with the point $(z,f(z))$. Clearly, the Cantor ternary function $f(x)$ is not polynomial so that `RootFinding[Isolate]` can not be utilized. Since $f(x)$ can be evaluated exactly at points with a finite ternary expansion, we can take $\delta=0$. Moreover, $f(x)$ is Hölder continuous on $[0,1]$ with Hölder exponent $\alpha=\log 2/\log 3$ and Hölder constant $C=2$ so that we can simply take

$$\mu=10^{-26}<\sqrt[\alpha]{\frac{\epsilon-\delta}{C}}.$$

Hence,

$$z^*=\frac{2}{3^{40}}+\frac{2}{3^{41}}+\frac{1}{3^{42}}+\frac{1}{3^{43}}+\frac{2}{3^{44}}+\frac{2}{3^{45}}+\frac{1}{3^{48}}+\frac{1}{3^{49}}+\frac{1}{3^{50}}+\frac{1}{3^{51}}+\frac{2}{3^{52}}+\frac{1}{3^{53}}+\frac{1}{3^{54}}+\frac{2}{3^{55}}+\frac{1}{3^{56}}$$

satisfies $|z-z^*|<\mu$ so that

$$f(z^*)=\frac{1}{2^{42}}+\frac{1}{2}\left(\frac{2}{2^{40}}+\frac{2}{2^{41}}\right)=\frac{7}{2^{42}}$$

is certifiably within ϵ of $f(z)$.

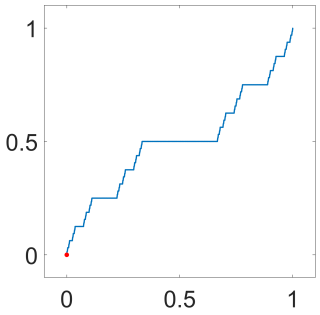


Fig. 1. Plot of the Cantor ternary function f with evaluation at a root of p.

5 Implementation and examples

The certified evaluation procedure has been implemented as a `Maple` package entitled `EvalCertification` available at https://github.com/P-Edwards/EvalCertification along with `Maple` notebooks for the examples. The main export is the procedure `EstimateRootsAndCertifyEvaluations` which has the following high level signature:

Input:

- Univariate polynomial $p \in \mathbb{Q}[x]$.
- List of locally Hölder continuous functions $f_1, \ldots, f_m$ with which to certifiably estimate evaluations at the roots of $p(x)$.
- List of procedures specifying how to compute local Hölder constants and exponents for $f_1, \ldots, f_m$. (See Section 5.3 for example of the syntax).
- Desired accuracy $\epsilon \in \mathbb{Q}_{>0}$.

Main output:

- Complex rational root approximations $z_1^*, \ldots, z_s^*$, one for each of the distinct roots $z_1, \ldots, z_s$ of $p(x)$, such that $|z_j - z_j^*| \leq \epsilon$.
- For each f_i and x_j, a complex decimal number f_{ij}^* with $|f_i(x_j) - f_{ij}^*| \leq \epsilon$.

The `EvalCertification` package is formatted in a .mpl file which can be read into a notebook with:

```
read("EvalCertification.mpl")
with(EvalCertification)
```

This lists the package's following four exports: the main function and three built in procedures for determining local Hölder constants and exponents for common classes of Hölder functions.

```
EstimateRootsAndCertifyEvaluations, HolderInformationForExponential,
HolderInformationForPolynomial,  HolderInformationForRationalPolynomial
```

The following highlight specific `Maple` types of inputs and outputs as well as other interface details.

5.1 Critical values

As a first example, consider (2) by certifiably evaluating

$$f(x) = 21x^8 - 42x^4 - 56x^3 + 3$$

at the roots of $p(x) = f'(x) = 168x^2(x^5 - x - 1)$ with error $\epsilon = 10^{-14}$.

```
f_polynomial := 21x^8 - 42x^4 - 56x^3 + 3;
f_derivative := diff(f_polynomial, x);
EstimationPrecision := 1/10^14;
```

The main call to `EstimateRootsAndCertifyEvaluations` is subsequently:

```
solutions_information :=
EstimateRootsAndCertifyEvaluations(f_derivative,
                        [f_polynomial, f_derivative],
                        HolderInformationForPolynomial,
                        EstimationPrecision);
```

The first argument provides the polynomial to solve and the second is a list of polynomials to evaluate. For illustration, we include evaluating the polynomial to solve in the evaluation list. The third argument is a procedure for computing Hölder constants which, in this case, uses the procedure that implements the estimates in Section 2.1 for polynomials. Notice that we need only provide the procedure once since all functions for evaluation fall into the same class of Hölder functions, namely polynomials. The last argument is the final error bound.

The output `solutions_information` is formatted as a Record. Certifiably estimated roots are stored in a list as illustrated.

```
solutions_information:-root_values =
[0,
2691619717901426047/2305843009213693952,
26745188167908553113/147573952589676412928 -
19995423894655642147*I/18446744073709551616, ...]
```

Evaluations are also stored in lists, one list for each function to evaluate with one entry for each root of p. Estimates are ordered so that the estimate at index i in its list corresponds to the root at index i in the roots list.

```
solutions_information:-evaluations_functions_1 =
[3., -91.6600084778015707, ...];
solutions_information:-evaluations_functions_2 =
[0, -6.692143197043304*10^(-16), ...];
```

Therefore, the solution to (2) is -91.6600084778015707 which is certifiably correct within an error of 10^{-14}.

Since $f(x)$ is polynomial, we can compare with `RootFinding[Isolate]` as discussed in Remark 2. Since evaluations at only the real critical points are of interest, one can simply utilize

```
Isolate(f_derivative, constraints = [f_polynomial], digits = 14);
```

which yields

```
[x = 0., x = 1.1673039782614],
[[21*x^8 - 42*x^4 - 56*x^3 + 3 = 3.],
[21*x^8 - 42*x^4 - 56*x^3 + 3 = -91.660008477802]]
```

The 14 digits of 1.1673039782614 are indeed correct, but the result has an absolute error of approximately $1.87 \cdot 10^{-14}$ while the evaluation -91.660008477802 has an absolute error of approximately $4.29 \cdot 10^{-13}$. Nonetheless, the relative error for both the root and evaluation estimates is less than the requested 10^{-14}, both of which are controlled by the `digits` argument. This fulfills the documented specifications for `Isolate`, though the `digits` argument's control over the evaluation's relative error is undocumented [11]. An additional function call with `digits` set to 16 would be necessary to decrease the absolute error below 10^{-14} for both the root and evaluation.

5.2 Comparison with ball arithmetic

Interval and ball arithmetic methods can provide similar certification functionality as `EvalCertification` by first isolating each root and then evaluating a interval extension of the function. As mentioned in the Introduction, this two-step procedure does not provide direct control on the size of the evaluation error and thus one may need to perform several loops to refine the isolation of each root to have sufficiently small evaluation error. In contrast, for evaluating functions without poles, `EvalCertification` always performs root estimation exactly twice: once to obtain local Hölder information and then a second time to guarantee small evaluation error.

For illustration, consider evaluating the function $f(x) = 50^x$ at the roots of the polynomial $p(x) = (x^7 + x - 1)(x - 1000)$ with an error at most $\epsilon = 2^{-1000}$. The library `Arb` [6] required a root estimation error of at most $0.91 \cdot 2^{-8176}$ to provide the requisite evaluation error. In `Arb`, relative error is input as a number of bits of precision available to computations. Thus, by supplying additional bits of precision, one lowers the relative and absolute error. For our computation, the precision in `Arb` was initialized at $2^{10} = 1024$ bits of precision which is enough to accurately store the desired error of 2^{-1000} exactly. We then utilized the two-stage procedure which loops back to refine the root if the output evaluation error is unacceptably large. If we simply double the number of bits of precision used in each loop, then three iterations are required to yield 8192 bits of precision which is sufficient to perform root estimation accurately enough for the function evaluation to yield the desired evaluation error.

As mentioned in Section 3, transforming complex root isolation into bivariate real root isolation is a costly maneuver. However, such an approach was used in `EvalCertification` to take advantage of the already existing `RealRootIsolate` in `Maple`. Since `Arb` implements a faster univariate solver that only allows relative error bounds on the estimates as input, this accounts for the drastic difference in computing time on this problem using `Arb` (0.52s) and using `RealRootIsolate` in `Maple` via `EvalCertification` (153s).

5.3 Extending with custom Hölder information procedures

Polynomial and rational functions can utilize the built-in procedures for computing local Hölder constants. One more feature of `EvalCertification` is the ability to extend the certification procedures to new classes of functions by specifying how to compute local Hölder constants. To illustrate, consider the Weierstrass function $f : \mathbb{R} \to \mathbb{R}$ given by

$$f(x) = \sum_{n=0}^{\infty} 7^{-\frac{n}{3}} \cos(7^n \pi x)$$

which we aim to evaluate at the unique real root z of $p(x) = x^7 + x - 1$. Figure 2 plots $f(x)$ and $p(x)$ along with the point $(z, f(z))$. The Weierstrass function $f(x)$ is nowhere differentiable but is globally Hölder continuous [5] with exponent $\alpha = 1/3$ and constant $C \leq 4.73$. The following is the format for defining a new procedure to supply the Hölder information:

```
WeierInfo := proc(f, point, radius, domain_estimate := false)
return Record('exponent' = 1/3,
              'constant' = 4.73,
              'avoid_roots' = false); end proc;
```

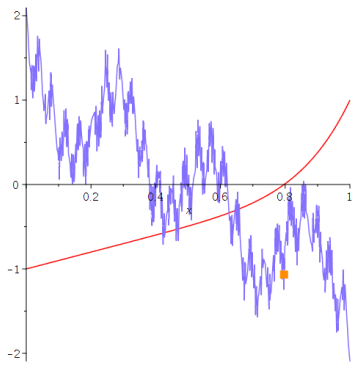


Fig. 2. Plot of the Weierstrass function $f(x)$, the polynomial $p(x)$, and point $(z, f(z))$ where z is the unique real root of $p(x)$.

All custom Hölder information procedures must follow the same signature as this example. The `exponent` α and `constant` C in the output `Record` should satisfy the Hölder conditions for `InputFunction` on the ball $B(\texttt{point}, \texttt{radius})$. For this example, the Hölder information is independent of the `point` and `radius` since the Weierstrass function is globally Hölder continuous. The entry `avoid_roots` lists estimates within `radius` of points missing from the input function's domain or false if defined everywhere.

Since $f(x)$ is an infinite series, we must evaluate a finite truncation of it, say

$$f_N(x) = \sum_{n=0}^{N} 7^{-\frac{n}{3}} \cos(7^n \pi x) \quad \text{with} \quad |f(x) - f_N(x)| \leq \sum_{n=N+1}^{\infty} 7^{-\frac{n}{3}} = \frac{7^{-\frac{N+1}{3}}}{1 - 7^{-\frac{1}{3}}} =: E_N.$$

Therefore, to approximate $f(z)$, one has three sources of error: approximation error in z, finite truncation error E_N, and numerical error when evaluating f_N. After selecting N such that $E_N < \epsilon$, one can simply replace ϵ by $\epsilon - E_N$ with the other two errors already accounted for in our approach. The following commands produce certified evaluations of f to precision 10^{-14} at z utilizing $N = 51$ so that $E_{51} < 4.71 \cdot 10^{-15}$:

```
p := x^7 + x - 1;
MaxErr := 1/10^14-E_N;
solutions_information :=
EstimateRootsAndCertifyEvaluations(p,[F_N],WeierInfo,MaxErr);
```

This yields $z^* = 0.79654435412846$ and $f^* = -1.06659590869988$.

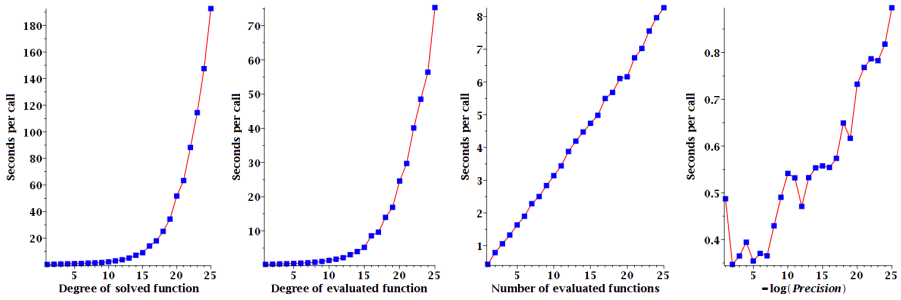
5.4 Benchmarking

As mentioned in Section 5.2, the dominant computational cost is in estimating roots with the next largest cost associated with computing local Hölder constants. Suppose that $R(p, \epsilon)$ is the complexity of approximating roots of p within ϵ, $H(f_1, \ldots, f_n, p, \epsilon)$ is the minimum complexity of computing Hölder constants at one root, and $A(p, f_1, \ldots, f_n, \epsilon)$ is the number of repetitions required to find an accuracy $\gamma \leq \epsilon$ where local Hölder constants can be calculated. Then,

$$A(p, f_1, \ldots, f_n, \epsilon)(R(p, \epsilon) + n \deg(p) H(f_1, \ldots, f_n, p, \epsilon)) + R(p, \epsilon)$$

is a lower bound on the complexity. The number of repetitions A is 1 for functions without poles and otherwise depends on the input in a complicated way which we do not attempt to characterize here.

We benchmarked `EvalCertification` using random polynomials generated by the command `randpoly` in `Maple`. All tests computed roots of a random polynomial $p(x)$ with integer coefficients between -10^{10} and 10^{10} and evaluated rational functions where the numerator and denominator were polynomials of degree D. The average was taken over 50 random selections. Figure 3 shows the results of the benchmarking tests, which were performed on Ubuntu 18.04 running `Maple` 2020 with an Intel Core i7-8565U processor. They were based on the degree d of $p(x)$, the value of D, the number of functions n to evaluate, and the size of the output error ϵ.



(a) (b) (c) (d)

Fig. 3. Results of tests with (a) $d \in \{1, \ldots, 25\}$, $D = 5$, $n = 1$, and $\epsilon = 10^{-14}$; (b) $d = 5$, $D \in \{1, \ldots, 25\}$, $n = 1$, and $\epsilon = 10^{-14}$; (c) $d = 5$, $D = 5$, $n \in \{1, \ldots, 25\}$, and $\epsilon = 10^{-14}$; (d) $d = 5$, $D = 5$, $n = 1$, and $\epsilon \in \{1, 10^{-1}, \ldots, 10^{-25}\}$.

6 Application to prove non-negativity

One application of our approach for computing certified evaluations is to certifiably decide whether or not all coefficients of the Taylor series expansion centered at the origin are non-negative for a given real rational function $r(x)$. We focus on non-negativity since non-positivity is equivalent to non-negativity for $-r(x)$ and alternating in sign is equivalent to non-negativity for $r(-x)$. The following method uses certified evaluations to obtain information about the coefficients in the tail of the Taylor series expansion reducing the problem to only needing to inspect finitely many coefficients. This approach assumes that the function does not have a pole at the origin, its denominator has only simple roots, and its denominator has a real positive root that is strictly smallest in modulus amongst all its roots. This approach can be extended to more general settings, but will not considered here due to space considerations.

We will make use of the following standard theorem.

Theorem 1. *Let $p(x), q(x) \in \mathbb{R}[x]$ such that $p(x)$ and $q(x)$ have no common root, $q(0) \neq 0$ and $\deg(p(x)) < \deg(q(x)) = d$. If $q(x)$ has only simple roots say $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$, then $r(x) = p(x)/q(x)$ has a Taylor series expansion of the form $r(x) = \sum_{n=0}^{\infty} r_n x^n$ converging for all $x \in \mathbb{C}$ with $|x| < \min\{|\alpha_1|, \ldots, |\alpha_d|\}$. Furthermore, for all $n \geq 0$,*

$$r_n = -\sum_{i=1}^{d} \frac{p(\alpha_i)}{\alpha_i q'(\alpha_i)} \alpha_i^{-n}. \tag{8}$$

Theorem 1 follows from partial fraction decomposition of rational functions or using linear recurrences. For completeness, we provide a proof in the Appendix. Using Theorem 1, we obtain the following result on the eventual behavior of the coefficients of the Taylor series of certain rational functions.

Theorem 2. *With the setup from Theorem 1, define* $C_i = -p(\alpha_i)/(\alpha_i q'(\alpha_i))$ *for* $i = 1, \ldots, d$. *If* $\alpha_1 \in \mathbb{R}$ *is such that* $|\alpha_1| < \min\{|\alpha_2|, \ldots, |\alpha_d|\}$, *then there exists* N *after which exactly one of the following conditions on* r_n *holds:*

1. *If* $\alpha_1 > 0$ *and* $C_1 > 0$, *then* $r_n > 0$ *for all* $n > N$.
2. *If* $\alpha_1 > 0$ *and* $C_1 < 0$, *then* $r_n < 0$ *for all* $n > N$.
3. *If* $\alpha_1 < 0$, *then* r_n *is alternating in sign for all* $n > N$, *i.e.,* $(-1)^n \cdot r_n > 0$ *for all* $n > N$ *or* $(-1)^n \cdot r_n < 0$ *for all* $n > N$.

Moreover, one may take $N = \log(K)/\log(M/m)$ *where* $K = \sum_{i=2}^{d} |C_i|/|C_1|$, $m = |\alpha_1|$, *and* $M = \min\{|\alpha_2|, \ldots, |\alpha_d|\}$.

A proof of Theorem 2 is provided in the Appendix. Theorems 1 and 2 yield the following.

Corollary 1. *Suppose that* $f(x), q(x) \in \mathbb{R}[x]$ *have no common root,* $q(0) \neq 0$, *and* $q(x)$ *has only simple roots, namely* $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$, *such that* $\alpha_1 \in \mathbb{R}$ *and* $|\alpha_1| < \min\{|\alpha_2|, \ldots, |\alpha_d|\}$. *Let* $g(x), p(x) \in \mathbb{R}[x]$ *be the unique polynomials such that* $f(x) = q(x) \cdot g(x) + p(x)$ *with* $\deg(p(x)) < \deg(q(x))$. *Define* $C_i = -p(\alpha_i)/(\alpha_i q'(\alpha_i))$ *for* $i = 1, \ldots, d$. *Then,* $f(x)/q(x)$ *has a Taylor series expansion* $f(x)/q(x) = \sum_{n=0}^{\infty} R_n x^n$ *converging for all* $x \in \mathbb{C}$ *with* $|x| < \min\{|\alpha_1|, \ldots, |\alpha_d|\}$ *and there is a threshold* N_0 *so that exactly one of the following conditions on* R_n *holds:*

1. *If* $\alpha_1 > 0$ *and* $C_1 > 0$, *then* $R_n > 0$ *for all* $n > N_0$.
2. *If* $\alpha_1 > 0$ *and* $C_1 < 0$, *then* $R_n < 0$ *for all* $n > N_0$.
3. *If* $\alpha_1 < 0$, *then* R_n *is alternating in sign for all* $n > N_0$, *i.e.* $(-1)^n \cdot R_n > 0$ *for all* $n > N_0$ *or* $(-1)^n \cdot R_n < 0$ *for all* $n > N_0$.

One can take $N_0 = \max\{\deg(f(x)) - \deg(q(x)) + 1, \log(K)/\log(M/m)\}$ *where* $K = \sum_{i=2}^{d} |C_i|/|C_1|$, $m = |\alpha_1|$, *and* $M = \min\{|\alpha_2|, \ldots, |\alpha_d|\}$.

Proof. Since $f(x)/q(x) = g(x) + p(x)/q(x)$, applying Theorem 1 yields the first part. Since the Taylor series coefficients of $f(x)/q(x)$ and $p(x)/q(x)$ are same for $n > \deg(f(x)) - \deg(g(x))$, the second part immediately follows from Theorem 2.

One key to utilizing Theorem 2 and Corollary 1 is to certify that $q(x)$ satisfies the requisite assumptions. Validating that $q(x)$ has only simple roots follows from computing an irreducible factorization as in Section 3 and checking if every factor has multiplicity 1. Section 6.1 describes a certified approach to verify the remaining conditions on $q(x)$. Section 6.2 yields a complete algorithm for certifiably deciding non-negativity of all Taylor series coefficients which is demonstrated on two examples.

6.1 Classification of roots

Given a polynomial $q(x) \in \mathbb{R}[x]$ with only simple roots and $q(0) \neq 0$, the following describes a method to certifiably determine if $q(x)$ has a positive root

that is strictly smallest in modulus amongst all its roots. This method uses the ability to certifiably approximate all real points in zero-dimensional semi-algebraic sets. Computationally, this can be accomplished using the command `RealRootIsolate` in `Maple`. Note that some of these computations could also be accomplished using `RootFinding[Isolate]` following Remark 2.

The first step is to certifiably determine if $q(x)$ has a positive root via

$$\mathcal{P} = \{p \in \mathbb{R} \ : \ q(p) = 0, p > 0\}.$$

If $\mathcal{P} = \emptyset$, then one returns that $q(x)$ does not have a positive root. Otherwise, one proceeds to test the modulus condition for $\alpha_1 = \min \mathcal{P}$.

The modulus condition needs to be tested against negative roots and non-real roots. For negative roots, consider

$$\mathcal{N} = \{n \in \mathbb{R} \ : \ q(-n) = 0, n > 0\} \ \text{ and } \ \mathcal{B} = \{b \in \mathbb{R} \ : \ q(b) = q(-b) = 0, b > 0\}.$$

By using certified approximations of α_1 and points in $\mathcal{N}$ and $\mathcal{B}$ of decreasing error, one can certifiably determine which of the following holds: $\alpha_1 < \min \mathcal{N}$, $\alpha_1 > \min \mathcal{N}$, or $\alpha_1 \in \mathcal{B} \subset \mathcal{P}$. If $\alpha_1 > \min \mathcal{N}$ or $\alpha_1 \in \mathcal{B}$, then one returns that $q(x)$ does has not a positive root that is strictly smallest in modulus amongst all its roots. Otherwise, one proceeds to the non-real roots by considering

$$\mathcal{L} = \{(r, a, b) \in \mathbb{R}^3 \ : \ q(r) = 0, r > 0, q(a + ib) = 0, b > 0, a^2 + b^2 < r^2\} \text{ and}$$
$$\mathcal{E} = \{(r, a, b) \in \mathbb{R}^3 \ : \ q(r) = 0, r > 0, q(a + ib) = 0, b > 0, a^2 + b^2 = r^2\}.$$

Note that $q(a + ib) = 0$ provides two real polynomial conditions on $(a, b) \in \mathbb{R}^2$ via the real and imaginary parts as in (7) so that $\mathcal{L}$ and $\mathcal{E}$ are clearly zero-dimensional semi-algebraic sets. Moreover, for the projection map $\pi_1(r, a, b) = r$, $\pi_1(\mathcal{L} \cup \mathcal{E}) \subset \mathcal{P}$. By using certified approximations of α_1 and points in $\mathcal{L}$ and $\mathcal{E}$ of decreasing error, one can certifiably determine if $\alpha_1 \in \pi_1(\mathcal{L} \cup \mathcal{E})$ or $\alpha_1 \notin \pi_1(\mathcal{L} \cup \mathcal{E})$. If the former holds, then one returns that $q(x)$ does has not a positive root that is strictly smallest in modulus amongst all its roots. If the later holds, then one returns that $q(x)$ does indeed have a positive root that is strictly smallest in modulus amongst all its roots.

6.2 Certification of non-negativity

Suppose that $f(x), q(x) \in \mathbb{R}[x]$ which satisfy the assumptions in Corollary 1. The following describes a method to certifiably determine if all of the coefficients R_n of the Taylor series expansion for $f(x)/q(x)$ centered at the origin are non-negative or provides an integer n_0 such that $R_{n_0} < 0$.

First, the Euclidean algorithm is utilized to determine $g(x), p(x) \in \mathbb{R}[x]$ with $\deg(p(x)) < \deg(q(x))$ such that $f(x) = q(x) \cdot g(x) + p(x)$. Define $h(x) = x \cdot q'(x)$ and $C(x) = -p(x)/h(x)$. Hence, $d = \deg(q(x)) = \deg(h(x))$ such that $q(x)$ and $h(x)$ have no common roots. As in Corollary 1, let $\alpha_1, \ldots, \alpha_d$ be the roots of $q(x)$ with $\alpha_1 \in \mathbb{R}_{>0}$ such that $\alpha_1 < \min\{\alpha_2, \ldots, \alpha_d\}$. Let $\beta_1, \ldots, \beta_d \in \mathbb{C}$ (not necessarily all distinct) be the roots of $h(x)$.

Certified evaluations at the roots of $q(x)$ and $h(x)$ with error bound $\epsilon_k = 2^{-k}$ for $k = 1, 2, \ldots$ can be used until the following termination conditions are met:

1. α_i^* and β_j^* are such that $\alpha_1^* \in \mathbb{R}$, $|\alpha_i^* - \alpha_i| < \epsilon_k$, and $|\beta_j^* - \beta_j| < \epsilon_k$
2. the set $\{0, \alpha_1^*, \ldots, \alpha_d^*\}$ is $2 \cdot \epsilon_k$ separated, i.e., $|s-t|^2 \geq (2\epsilon_k)^2$ for all distinct s, t in this set,
3. $\gamma^* \leq \min\{|\alpha_i^* - \beta_j^*| \ : \ 1 \leq i, j \leq d\}$ such that $\gamma^* > 2 \cdot \epsilon_k + \epsilon_k^{1/(4d)}$,
4. for $m^* = \alpha_1^* + \epsilon_k$ and $M^* \leq \min\{|\alpha_2^*|, \ldots, |\alpha_d^*|\} - \epsilon_k$, one has $m^* < M^*$,
5. L_i^* such that $L_i^* \geq |c_d|^{-2} \sum_{\ell=0}^{2d-1} |u^{(\ell)}(\alpha_i^*)| \epsilon_k^\ell / \ell!$ where c_d is the leading coefficient of $q(x)$ and $u(x) = -p'(x)h(x) + p(x)h'(x)$, and
6. either (a) $C_1^* + L_1^* \sqrt{\epsilon_k} < 0$ or (b) $C_1^* - L_1^* \sqrt{\epsilon_k} > 0$.

Note that starred quantities in the termination conditions above, or in the further discussion below, are either exact rational approximations to real constants or complex numbers with rational real and imaginary parts that approximate roots.

Before proving that such a termination condition can be met, we describe the last steps which are justified by Corollary 1. Let $\epsilon = \epsilon_k$ be the value where the termination conditions are met and calculate $0 < b^* \leq \min\{|C_1^* \pm L_1^* \sqrt{\epsilon}|\}$,

$$K^* \geq (1/b^*) \sum_{i=2}^{d} (|C_i^*| + L_i^* \sqrt{\epsilon}), \ \ A^* \geq \log(K^*)/\log(M^*/m^*),$$
$$N_0^* = \max\{\deg(f(x)) - \deg(q(x)), \lceil A^* \rceil\}.$$

The inequalities in Items 3, 4, and 5 and these values above are meant to signify the rounding direction in machine precision used to compute the values. With this, if $C_1^* + L_1^* \sqrt{\epsilon_k} < 0$, then return $n_0 = N_0^* + 1$ for which $R_{n_0} < 0$. Otherwise, the non-negativity of all R_n is equivalent to the non-negativity of $R_0, \ldots, R_{N_0^*}$ which can be computed by explicit computation. If there exists $n_0 \in \{0, \ldots, N_0^*\}$ such that $R_{n_0} < 0$, return n_0. Otherwise, return that all R_n are non-negative.

First, note that Item 1 is obtained by real root certification. We have

$$\delta = \min\{|\alpha_i - \alpha_j|, |\alpha_i| \ : \ 1 \leq i < j \leq d\} > 0.$$

By Item 1, $|\alpha_i^* - \alpha_j^*| \geq |\alpha_i - \alpha_j| - 2\epsilon_k$ and $|\alpha_i^*| \geq |\alpha_i| - \epsilon_k$. Thus,

$$\delta^* = \min\{|\alpha_i^* - \alpha_j^*|, |\alpha_i^*| \ : \ 1 \leq i < j \leq d\} \geq \delta - 2\epsilon_k - \eta$$

where $\eta > 0$ is the machine precision on the lower bounds on the quantities in δ^*. Since $\epsilon_k \to 0$ and η can be made arbitrarily small, eventually $\delta^* - 2\epsilon_k - \eta > 2\epsilon_k$ and Item 2 will be met.

Since $q(x)$ and $h(x)$ have no roots in common, consider

$$\gamma = \min\{|\alpha_i - \beta_j| \ : \ 1 \leq i, j \leq d\} > 0.$$

We have $\gamma^* \geq \gamma - 2\epsilon_k - \eta$ where $\eta > 0$ is the machine precision on the lower bounds of the quantities in γ^*. Eventually, $\gamma^* \geq \gamma - 2\epsilon_k - \eta > 2\epsilon_k + \nu + \epsilon_k^{1/(4d)}$ where $\nu > 0$ is the machine precision on the upper bound for $\epsilon_k^{1/(4d)}$ and Item 3 will be met.

Since $\Delta = M - m > 0$, we have $M^* > M - \epsilon_k - \eta$ where $\eta > 0$ is the machine precision on the lower estimates in M^* and also $m^* = \alpha_1^* + \epsilon_k < \alpha_1 + 2\epsilon_k$. Thus, $M^* - m^* \geq \Delta - 3\epsilon_k - \eta$ so that eventually Item 4 will be met.

Since $C(x) = p(x)/h(x)$, we have $C'(x) = u(x)/h^2(x)$. Assuming the previous items have all been met, we have

$$|\beta_j - \alpha_i^*| \geq |\beta_j^* - \alpha_i^*| - |\beta_j - \beta_J|^* > 2\epsilon_k + \epsilon_k^{1/(4d)} - \epsilon_k > \epsilon_k.$$

Hence, $h(x)$ has no roots in $B(\alpha_i^*, \epsilon_k)$ and $C'(x)$ is continuous on $B(\alpha_i^*, \epsilon_k)$. Fix $z \in B(\alpha_i^*, \epsilon_k)$ and let ζ be the straight line segment contour from α_i^* to z in $B(\alpha_i^*, \epsilon_k)$. Since $C'(x)$ exists on $B(\alpha_i^*, \epsilon_k)$, $C(z) - C(\alpha_i^*) = \int_\zeta C'(x)dx$. Thus, $|C(z) - C(\alpha_i^*)| \leq P \cdot |z - \alpha_i^*|$ where $P = \max\{|C'(x)| \ : \ x \in B(\alpha_i^*, \epsilon_k)\}$. Therefore, we have $P \leq P_1/P_2^2$ where $P_1 = \max\{|u(x)| \ : \ x \in B(\alpha_i^*, \epsilon_k)\}$ and $P_2 = \min\{|h(x)| \ : \ x \in B(\alpha_i^*, \epsilon_k)\}$. Since $u(z) = \sum_{\ell=0}^{2d-1} u^{(\ell)}(\alpha_i^*)(z - \alpha_i^*)^\ell/\ell!$, we have $P_1 \leq \sum_{\ell=0}^{2d-1} |u^{(\ell)}(\alpha_i^*)|\epsilon_k^\ell/\ell!(1+\eta)$ where $\eta > 0$ is the machine precision that results from the upper bound on the quantities $|u^{(\ell)}(\alpha_i^*)|$. Since

$$h(z) = c_d \prod_{j=1}^{d}(z - \beta_j) \ \text{ and } \ |z - \beta_j| \geq |\alpha_i^* - \beta_j^*| - |z - \alpha_i^*| - |\beta_j^* - \beta_j| > \epsilon_k^{1/(4d)},$$

we have $|h(z)| \geq |c_d|\epsilon^{1/4}$. Thus, $P_2^2 \geq |c_d|^2\epsilon^{1/2}$ and $|C(z) - C(\alpha_i^*)| \leq L_i^*\sqrt{\epsilon}$. Thus, $|C_i^* - C_i| \leq L_i^*\sqrt{\epsilon}$ where $C_i = C(\alpha_i)$ and $C_i^* = C(\alpha_i^*)$. By the inclusions $B(\alpha_i^*, \epsilon_k) \subset B(\alpha_i, 2\epsilon_k) \subset B(\alpha_i, 1)$, all estimates $|u^{(\ell)}(\alpha_i^*)|$ will be bounded above by the corresponding maximum values of $|u^{(\ell)}(z)|$ for $z \in B(\alpha_1, 1)$. Thus, even though P_1 varies in each step k, P_1 and L_i^* will be uniformly bounded above for all k. Since L_1^* is uniformly bounded above and $C_1 \neq 0$ by the proof of Theorem 2, Item 6(a) will eventually be met if $C_1 < 0$ while Item 6(b) will eventually be met if $C_1 > 0$.

Having proved termination, we note that in order to get a value of N_0^* which is reasonably close to the value of N_0 in Corollary 1, one may continue to decrease ϵ_k past the point where all termination conditions are first met. The reason for this is to separate m^* and M^* as far all possible, i.e., to match the actual gap between m and M as closely as possible. This could be wise especially when m^* is very close to M^* in which case $\log(M^*/m^*)$ will be very close to 0 so that N_0^* will be very large.

Example 2. To demonstrate the approach, consider the rational functions

$$(1 - x^3 - x^7 + x^{18})^{-1} \qquad \text{and} \qquad (1 - x^3 - x^7 + x^{21})^{-1}.$$

The implementation of this approach in `Maple` certifies that both rational functions have Taylor series expansions centered at the origin where all of the coefficients are non-negative. The value of N_0 from Corollary 1 which could be certified by the method described above was $N_0^* = 204$ and $N_0^* = 55$, respectively. Thus, it was easy to utilize `series` in `Maple` to check the non-negativity of the Taylor series coefficients up to N_0^* combined with Corollary 1 for the tail.

7 Conclusion

This manuscript developed techniques for certified evaluations of locally Hölder continuous functions at roots of polynomials along with an implementation in `Maple`. These techniques were demonstrated on several problems including certified bounds on critical values and proving non-negativity of coefficients in Taylor series expansions. Although this paper focused on roots of univariate polynomials, it is natural to extend to multivariate polynomial systems in the future.

Acknowledgments

JDH was supported in part by NSF CCF 1812746. CDS was supported in part by Simons Foundation grant 360486.

References

1. Aubry, P., Lazard, D., Moreno Maza, M.: On the theories of triangular sets. J. Symbolic Comput. **28**(1-2), 105–124 (1999)
2. Becker, R., Sagraloff, M., Sharma, V., Yap, C.: A near-optimal subdivision algorithm for complex root isolation based on the Pellet test and Newton iteration. J. Symbolic Comput. **86**, 51–96 (2018)
3. Boulier, F., Chen, C., Lemaire, F., Maza, M.M.: Real root isolation of regular chains. In: The Joint Conference of ASCM 2009 and MACIS 2009, COE Lect. Note, vol. 22, pp. 15–29. Kyushu Univ. Fac. Math., Fukuoka (2009)
4. Gargantini, I., Henrici, P.: Circular arithmetic and the determination of polynomial zeros. Numer. Math. **18**, 305–320 (1971/72)
5. Hardy, G.H.: Weierstrass's non-differentiable function. Trans. Amer. Math. Soc. **17**(3), 301–325 (1916)
6. Johansson, F.: Arb: efficient arbitrary-precision midpoint-radius interval arithmetic. IEEE Transactions on Computers **66**, 1281–1292 (2017)
7. Kantorovich, L.V.: On Newton's method for functional equations. Doklady Akad. Nauk SSSR (N.S.) **59**, 1237–1240 (1948)
8. Kearfott, R.B.: Rigorous global search: continuous problems, Nonconvex Optimization and its Applications, vol. 13. Kluwer Academic Publishers, Dordrecht (1996)
9. Kobel, A., Rouillier, F., Sagraloff, M.: Computing real roots of real polynomials ... and now for real! In: Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation. pp. 303–310. ACM, New York (2016)
10. Krawczyk, R.: Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. Computing (Arch. Elektron. Rechnen) **4**, 187–201 (1969)
11. Maple 2020 Program Committee Chairs: Private Communication
12. Moore, R.E.: A test for existence of solutions to nonlinear systems. SIAM J. Numer. Anal. **14**(4), 611–615 (1977)
13. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to interval analysis. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2009)
14. Pan, V.Y.: Old and new nearly optimal polynomial root-finders. In: Computer algebra in scientific computing, Lecture Notes in Comput. Sci., vol. 11661, pp. 393–411. Springer, Cham (2019)

15. Rioboo, R.: Real algebraic closure of an ordered field: Implementation in axiom. In: Papers from the International Symposium on Symbolic and Algebraic Computation. p. 206–215. ISSAC '92, Association for Computing Machinery, New York, NY, USA (1992)
16. Rouillier, F.: Solving zero-dimensional systems through the rational univariate representation. Appl. Algebra Engrg. Comm. Comput. **9**(5), 433–461 (1999)
17. Rouillier, F., Zimmermann, P.: Efficient isolation of polynomial's real roots. J. Comput. Appl. Math. **162**(1), 33–50 (2003)
18. Rump, S.M.: Verification methods: rigorous results using floating-point arithmetic. Acta Numer. **19**, 287–449 (2010)
19. Smale, S.: Newton's method estimates from data at one point. In: The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics (Laramie, Wyo., 1985), pp. 185–196. Springer, New York (1986)
20. Xia, B., Yang, L.: An algorithm for isolating the real solutions of semi-algebraic systems. J. Symbolic Comput. **34**(5), 461–477 (2002)
21. Xia, B., Zhang, T.: Real solution isolation using interval arithmetic. Comput. Math. Appl. **52**(6-7), 853–860 (2006)
22. Yang, L., Hou, X., Xia, B.: A complete algorithm for automated discovering of a class of inequality-type theorems. Sci. China Ser. F **44**(1), 33–49 (2001)

Appendix

Proof of Theorem 1. Suppose that $C \neq 0$ such that $q(x) = C \cdot \prod_{i=1}^{d}(x-\alpha_i)$. Thus, we know $q'(x) = C \cdot \sum_{i=1}^{d} \prod_{j\neq i}(x-\alpha_j)$ and $q'(\alpha_i) = C \cdot \prod_{j\neq i}(\alpha_i - \alpha_j) \neq 0$ for all i. Let $p_i(x) = q(x)/(x-\alpha_i) = C \cdot \prod_{j\neq i}(x-\alpha_j)$. Hence, $p_i(\alpha_i) = q'(\alpha_i)$ and $p_i(\alpha_j) = 0$ if $j \neq i$. The polynomials $p_1, \ldots, p_d$ are linearly independent since, if $\sum_{i=1}^{d} a_i p_i(x) = 0$, then evaluating at $x = \alpha_j$ yields $a_j \cdot q'(\alpha_j) = 0$ which implies $a_j = 0$. Thus, they must form a basis for the d-dimensional vector space of polynomials of degree at most $d-1$.

Since $p(x)$ has degree at most $d-1$, there are unique constants a_i so that $\sum_{i=1}^{d} a_i p_i(x) = p(x)$. Evaluating at $x = \alpha_j$ yields $a_j q'(\alpha_j) = p(\alpha_j)$ so that $a_j = p(\alpha_j)/q'(\alpha_j)$. Therefore, for all $x \in \mathbb{C} \setminus \{\alpha_1, \ldots, \alpha_d\}$,

$$\frac{p(x)}{q(x)} = \sum_{i=1}^{d} \frac{p(\alpha_i)}{q'(\alpha_i)} \frac{1}{x-\alpha_i} = \sum_{i=1}^{d} -\frac{p(\alpha_i)}{\alpha_i q'(\alpha_i)} \frac{1}{1-x/\alpha_i}. \tag{9}$$

The terms in (9) have a Taylor series expansion centered at the origin that converge for all x with $|x| < \min\{|\alpha_1|, \ldots, |\alpha_d|\}$ such that, as (8) claims,

$$\frac{p(x)}{q(x)} = \sum_{i=1}^{d} -\frac{p(\alpha_i)}{\alpha_i q'(\alpha_i)} \sum_{n=0}^{\infty} \alpha_i^{-n} x^n = \sum_{n=0}^{\infty} \left(-\sum_{i=1}^{d} \frac{p(\alpha_i)}{\alpha_i q'(\alpha_i)} \alpha_i^{-n} \right) x^n.$$

Proof of Theorem 2. Clearly, one has $r_n = \frac{d^n}{dz^n} \left. \frac{p(z)}{q(z)} \right|_{z=0}$. Since $p(x)$ and $q(x)$ have real coefficients, r_n is real for all $n \geq 0$. For $i \in \{1, \ldots, d\}$, let $t_n^i = C_i \alpha_i^{-n}$ so

that (8) reduces to $r_n = \sum_{i=1}^{d} t_n^i$. Moreover, $\alpha_1 \in \mathbb{R} \setminus \{0\}$ implies $C_1 \in \mathbb{R} \setminus \{0\}$. Clearly, if $\alpha_1 < 0$, then t_n^1 is alternating in sign.

Consider the case when $\alpha_1 > 0$. First, note that t_n^1 and C_1 always have the same sign. The following derives a threshold N such that $|r_n - t_n^1| < |t_n^1|$ for all $n > N$. Given such an N, r_n will have the same sign as t_n^1 and C_1 for $n > N$ and the theorem will be proved. To that end, since $(r_n - t_n^1)/t_n^1 = \sum_{i=2}^{d} t_n^i/t_n^1$,

$$\frac{|r_n - t_n^1|}{|t_n^1|} \leq \sum_{i=2}^{d} \frac{|C_i|}{|C_1|} \frac{|\alpha_1|^n}{|\alpha_i|^n} \leq K \left(\frac{m}{M}\right)^n$$

for all n. Since, by assumption, $m/M < 1$, there is a threshold N so that $K(m/M)^n < 1$ and $|r_n - t_n^1| < |t_n^1|$ for all $n > N$. We may take N so that $K(m/M)^N = 1$ or $N = \log(K)/\log(M/m)$ as claimed.