Root certification using disk arithmetic

Nicolae Mihalache, François Vigneron

Root certification using disk arithmetic

Nicolae Mihalache[1] and François Vigneron[2]

June 27, 2025

Abstract

Splitting algorithms provide a list of all the roots of a polynomial, however, most numerical implementations lack the proper framework to provide certification, *i.e.* a computer assisted proof that the list is correct. This article adresses this issue, regardless of the underlying splitting algorithm, with minimal computational overhead. To ensure a strict control of the computational errors that is robust enough to withstand intensive iterations, we revisit and extend the theory of disk arithmetic. We also present two results, one regarding the localization of roots and one regarding the convergence of numerical refinements using Newton's method. The assumptions of both results can be certifiably checked numerically, using disk arithmetic and IEEE 754 compliant roundings. We release a compagnon libray `[Mandel]` in `C` code that implements those results and has been used in [MV25a] to provide a computer assisted certification of the first ever tera-scale splitting [`Mand.DB`].

1 Introduction

1.1 About root finding algorithms

A *splitting algorithm* is an algorithm that provides the list of all the roots of a polynomial through convergent numerical approximations. Splitting algorithms can be classified in three broad families: methods based on Newton's iterations, root isolation methods and eigenvalue methods.

The core of many iterative splitting algorithms is Newton's iteration:

$$z_{n+1} = N_f(z_n) \qquad \text{where} \qquad N_f(z) = z - \frac{f(z)}{f'(z)}. \tag{1}$$

The fixed points of the dynamics of N_f are the roots of f (a given polynomial or holomorphic function). Transforming Newton's method into a splitting algorithm requires choosing a set of starting points for which the dynamics will visit every root. The naive approach of findind one root at a time and factoring it out is both unpractical and numerically unstable [Wil84] and choosing starting points at random may miss some roots entirely. The global structure of the bassins of attraction of Newton's method is quite intricate. It has been extensively studied in [Sut89], [HSS01], [SS17], where a remarkable universal splitting algorithm is explained (see Theorem 1 below).

Aberth-Ehrlich method [Abe73] is a generalisation of Newton's method. It attempts to find all the roots of $P \in \mathbb{C}[z]$ simultaneously, with $d = \deg P$, by computing a fixed point for $z_n = (z_{n,j}) \in \mathbb{C}^d$:

$$z_{n+1,j} = z_{n,j} - \frac{1}{\frac{P'(z_{n,j})}{P(z_{n,j})} - \sum\limits_{k \neq j} \frac{1}{z_{n,j} - z_{n,k}}}.$$

The idea is that each root repels the others with an electrostatic potential, which prevents the Newton's dynamics to converge multiple times on a single root while missing another one completely. The Aberth method is at the heart of the implementation of `MPSolve` [BF00, BR14], which is the reference software for splitting polynomials using arbitrary precision arithmetic. A massively parallel implementation [GSKC16] of this algorithm on multiple GPUs can run up to a few millions of roots. In practice, the Aberth method has good global convergence properties. However, a theoretical foundation for this global convergence remains an open problem.

It was recently proven [RSS20], [KI04] that the Weierstrass variant (also known as Durand–Kerner method) fails generically, *i.e.* on an open set of polynomials for an open set of intialization points. This is a sharp reminder that one must pay close attention to the bassin of attraction of each numerical method.

Bracketing, or root isolation, consists in identifying a subset where the number of roots of P is prescribed. For real polynomials, textbook application of the intermediary value theorem (bisection method, ITP method) provides intervals along the real line. In the complex plane, the residue theorem implies that, for all $n \in \mathbb{N}$

$$\sum_{\substack{z \in \Omega \\ P(z)=0}} m_P(z) z^n = \frac{1}{2i\pi} \int_{\partial\Omega} \frac{P'(z)}{P(z)} z^n dz \tag{2}$$

for any smooth domain Ω whose boundary avoids the zeros of $P \in \mathbb{C}[z]$ and where $m_P(z)$ denotes the multiplicity of z as a root of P (splitting circle method, Lehmer–Schur algorithm).

Eigenvalue methods rely on the properties of so called compagnon matrices:

$$A_P = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ c_0 & c_1 & \cdots & \cdots & c_{n-1} \end{pmatrix} \qquad \text{if} \qquad P(x) = x^n - \sum_{k=0}^{n-1} c_k x^k. \tag{3}$$

The vector $(1, x, x^2, \dots, x^{n-1})$ is an eigenvector of A_P if and only if $P(x) = 0$. For example, in the QR algorithm, The sequence starts with $A_0 = A$, the matrix whose spectrum is being computed. The principle of each step consists in an orthogonal similarity transform $A_{k+1} = Q_k^T A_k Q_k = R_k Q_k$ where

$$A_k = Q_k R_k \quad \text{with} \quad \begin{cases} Q_k \text{ orthogonal} \quad (Q_k^T Q_k = I) \\ R_k \text{ upper triangular.} \end{cases}$$

Ideally, the sequence A_k converges to an upper triangular matrix, or at least, to a matrix whose spectrum is easier to compute.

In [MV25a], we have proposed a new splitting algorithm that combines Newton's method with the general principle of root isolation. The idea is to compute a discrete version of the level line $P(z) = \lambda$ where $\lambda > \max\{|P(z)|\,;\, P'(z) = 0\}$ exceeds just slightly the largest critical value, then use each of those points as the seed for Newton's descend (1). A natural parametrization of the level line has the specific property to adapt its density to the local density of the attraction channels for Newton's flow, the continuous version of (1).

Combining those ideas leads, at the theoretical level to a new proof of the fundamental theorem of algebra [AMV23] and, in practice, to an algorithm [MV25a] that behaves, at least on the polynomials we have applied it to, almost as well as an oracle (linear complexity with small constant). We have used this algorithm for the first ever splitting of a non-trivial polynomial of degree $\sim 10^{12}$ [`Mand.DB`].

1.2 Known results for root checking

As illustrated in [RSS17], there are various techniques to convince oneself that one has accurately found all the roots of a given polynomial.

One can use Viete and Newton's identities to check that a few sums of the powers of the roots are in concordance with the corresponding coefficients of the polynomial:

$$P(z) = z_d + \sum_{k=0}^{d-1} a_k z^k = \prod_{j=1}^{d} (z - z_j), \qquad a_{d-k} = (-1)^k \sum_{j_1 < \ldots < j_k} z_{j_1} \ldots z_{j_k} \tag{4}$$

and for all $k \in \mathbb{N}^*$

$$k a_{d-k} + \sum_{\ell=1}^{k} a_{d-k+\ell} \left(\sum_j z_j^\ell \right) = 0 \qquad (\text{with } a_d = 1,\ \ a_{-n} = 0). \tag{5}$$

Coupled with the inversion $\tilde{P}(z) = z^d P(1/z)$, one can also investigate negative powers of the roots. However, for large degrees, the cost of this kind of checkup is prohibitive for results that could be no more than a few lucky coincidences. Occasionally, this technique, which is essentially a discrete version of (2), can also help localize a few missing roots.

Rouché's theorem and standard estimates on the radius of quadratic convergence of Newton's method (see Proposition 1 below) are not new, yet their application remain, up to now, mostly limited to case-by-case studies.

A remarkable result that raises the confidence one can have in numerical methods is the following one, based on a precise study of the shape of the bassin of attraction of Newton's method.

Theorem 1 ([HSS01]). *Given a polynomial $P(z)$ of degree d whose roots are contained in $\mathrm{D}(0,r)$ and integers $\alpha \geq 4.16d\log d$, $\beta \geq 0.27\log d$, the mesh*

$$\mathbb{M}_r(d) = \bigcup_{1\leq\nu\leq\beta} r(1+\sqrt{2})\left(1-\frac{1}{d}\right)^{\frac{2\nu-1}{4\beta}} \mathbb{U}_\alpha$$

contains a point in each bassin of attraction of the roots of P, for Newton's method. Here, $\mathbb{U}_\alpha = \left\{e^{2ik\pi/\alpha}\,;\, 0\leq k<\alpha\right\}$ denotes the roots of unity of order α.

A similar guarantee for the level-line algorithm [MV25a] exists at the continuous level.

Theorem 2 ([AMV23]). *Given a polynomial $P(z)$ of degree d and $z_0\in\mathbb{C}$ with*

$$|P(z_0)| > \max\{|P(z)|\,;\, P'(z)=0\}, \qquad \arg P(z_0) \notin \{\arg P(z)\,;\, P'(z)=0\}$$

then the following Cauchy problems

$$\lambda'(t) = i\frac{P(\lambda(t))}{P'(\lambda(t))}, \qquad \lambda(0)=z_0 \tag{6}$$

and

$$z_k'(t) = \frac{P(z_k(t))}{P'(z_k(t))}, \qquad z_k(0)=\lambda(2k\pi), \qquad 0\leq k\leq d-1 \tag{7}$$

are all globally well posed and satisfy $P^{-1}(0) = \left\{\lim_{t\to\infty} z_k(t)\,;\, 0\leq k\leq d-1\right\}$.

Numerical stability is ensured by the fact that the measure of the set of $t\in[0,2\pi d)$ such that Newton's flow (7) initated on $\lambda(t)$ converges to a given root is exactly $2\pi\mu$, where μ is its multiplicity.

All those results can increase our confidence in a list of roots produced numerically, yet none of them can claim that they provide a numerical proof of that list. Leaving aside the question of how many iterates or refinements are necessary to ensure convergence, each computation is carried out in finite precision *i.e.* discrete arithmetic and thus carries rounding errors. Those errors could, for example, induce root swaps when the separation between roots is close to the limit of the machine precision, thus breaking the logic chain. Even an integer valued formula like (2) for $n=0$ may not guaranty a proper root count unless one can prove precise error bounds for the numerical scheme used to compute the integral.

1.3 Main results and structure of this article

In this article, we address this problem and claim that it is possible to perform numerical proofs, using finite precision arithmetic, while keeping the computational cost to a minimum.

We expose two results, Theorem 3 on the localization of roots and Theorem 4 on the convergence of Newton's method, whose assumptions can be verified in practice. To that effect, we present a theory of disk arithmetic, whose main statement is Theorem 5.

As a practical example, we show in Section 4 how we have combined those results to certify the roots of the first-ever splitting of a tera-scale polynomial [MV25a], *i.e.* a polynomial of degree $\sim 10^{12}$ along with other polynomials of lesser degree, that play a central role in the study of the Mandelbrot set. This article is paired with a code library `[Mandel]` that implements the ideas presented here and in [MV25a] and a certified database `[Mand.DB]` containing the roots of the aforementioned polynomials.

2 Results for root certification

We are interested in two types of certifications: either proving numerically that a root belongs to a certain disk that bounds our uncertainty, or proving that some numerical procedure (namely Newton's iterations) is guaranteed to refine the numerical value of that particular root to an arbitrarily high precision. The criteria presented in Theorem 3 and Theorem 4 are suitable for numerical verification, using the tools of Section 3. In this section, $U \subset \mathbb{C}$ is a domain of holomorphy and $\mathrm{Hol}(U)$ denotes the set of holomorphic functions on U.

2.1 How to prove the localization of a root

Let us consider $f \in \mathrm{Hol}(U)$ and assume that one has found a point $z_0 \in U$ such that $f(z_0) \in \mathrm{D}(0,\varepsilon)$ for some $\varepsilon > 0$ and $f'(z_0) \in \mathrm{D}(\lambda,\eta)$ with $|\lambda| > \eta > 0$. If $\varepsilon/(|\lambda| - \eta)$ is small enough, one can expect the existence of a root of f in the immediate vicinity of z_0. This idea is at the heart of Newton's method. We can combine it with the spirit of Rouché's theorem to produce a quantifiable statement.

Theorem 3. *Given a holomorphic function $f \in \mathrm{Hol}(U)$, a disk $B = \mathrm{D}(z_0, R)$ such that $\bar{B} \subset U$ and B' a second disk such that $f'(B) \subset B'$, we assume that*

$$R \operatorname{dist}(0, B') > |f(z_0)|. \tag{8}$$

Then there exists a unique point $z_* \in B$ such that $f(z_*) = 0$.

Proof. Let us introduce the arc $\gamma(t) = z_0 + Re^{2i\pi t}$ for $t \in [0,1]$. The fundamental theorem of calculus and the convexity of B' imply:

$$f(\gamma(t)) = f(z_0) + Re^{2i\pi t} \int_0^1 f'(z_0 + sRe^{2i\pi t})ds \in f(z_0) + Re^{2i\pi t} \cdot B'.$$

Assumption (8) thus implies that $f \circ \gamma$ is valued in an annulus centered at $f(z_0)$ that encircles zero. Its winding number with respect to zero is thus the same than that with respect to $f(z_0)$. Moreover, $f \circ \gamma$ is homotopic, within that annulus, to the path $t \mapsto f(z_0) + R\lambda e^{2i\pi t}$ where λ is the center of B' so the winding number is 1. Because of (8), $0 \notin f'(B)$ so all possible roots within B are simple and their number is

$$\#\left(f^{-1}(0) \cap B\right) = \frac{1}{2i\pi} \int_\gamma \frac{f'(z)}{f(z)} dz = \frac{1}{2i\pi} \int_{f\circ\gamma} \frac{d\zeta}{\zeta} = \frac{1}{2i\pi} \int_{f\circ\gamma} \frac{d\zeta}{\zeta - f(z_0)} = 1\,,$$

i.e. f admits exactly one root in B. □

2.2 How to prove the convergence of Newton refinements

In the vicinity of each simple root, Newton's method (1) converges bi-exponentially.

Proposition 1 (Folklore). *If $f \in \mathrm{Hol}(U)$, $z_* \in U$ with $f(z_*) = 0$ and $f'(z_*) \neq 0$, the bassin of attraction*

$$\mathcal{A}(z_*) = \{z \in U\,;\, \lim_{n\to\infty} N_f^n(z) = z_*\}$$

contains the open disk $\mathrm{D}(z_*, \rho)$ where

$$\rho = \sup\left\{ r \geq 0\,;\, \overline{\mathrm{D}}(z_*, r) \subset U \quad \text{and} \quad \frac{r}{2} \sup_{z\in\overline{\mathrm{D}}(z_*,r)} |f''(z)| < \inf_{z\in\overline{\mathrm{D}}(z_*,r)} |f'(z)| \right\} > 0. \tag{9}$$

Proof. Taylor's expansions for $f(z_n)$ and $f'(z_n)$ in the vicinity of z_* read

$$f(z_n) = (z_n - z_*)f'(z_*) + (z_n - z_*)^2 \int_0^1 (1-t)f''(tz_n + (1-t)z_*)dt,$$
$$f'(z_n) = f'(z_*) + (z_n - z_*) \int_0^1 f''(tz_n + (1-t)z_*)dt.$$

Those formulae can be combined into an exact formula for each Newton step:

$$z_{n+1} - z_* = \frac{(z_n - z_*)^2}{f'(z_n)} \int_0^1 t f''(tz_n + (1-t)z_*)dt. \tag{10}$$

For $r \geq 0$, one defines $C_r = \frac{1}{2}\sup|f''(z)|/\inf|f'(z)|$, each extremum being computed over all $z \in \overline{\mathrm{D}}(z_*, r)$. Observe that C_r is continuous and $C_0 = \frac{1}{2}\left|\frac{f''(z_*)}{f'(z_*)}\right| \in \mathbb{R}_+$, so $rC_r \to 0$ as $r \to 0$ and thus $\rho > 0$ in definition (9). If $z_{n_0} \in \mathrm{D}(z_*, r)$ with $r < \rho$, then $rC_r < 1$ and (10) gives

$$\forall n \geq n_0, \quad |z_{n+1} - z_*| \leq C_r |z_n - z_*|^2 \tag{11}$$

thus $|z_n - z_*|$ is decreasing for $n \geq n_0$ and $|z_n - z_*| \leq C_r^{-1}(rC_r)^{2^{n-n_0}}$. □

Remark 2. *When the root z_* is of multiplicity $\mu \geq 2$, then $N_f'(z) = \frac{f(z)f''(z)}{f'(z)^2}$ converges to $1 - \frac{1}{\mu} \in \left[\frac{1}{2}, 1\right)$, instead of 0, as $z \to z_*$. The convergence of Newton's algorithm is then only asymptotically exponential instead of bi-exponential, with (11) replaced by*

$$\lim_{n\to\infty} \frac{z_{n+1} - z_*}{z_n - z_*} = 1 - \frac{1}{\mu} \tag{12}$$

for any z_0 in the bassin of attraction $\mathcal{A}(z_*)$.

The practical limitation of Proposition 1 is that the radius ρ defined by (9) is not, usually, an explicit one. The following result adresses this issue, at least for simple roots. For further results, see [Hen74, Chap. 6].

Theorem 4. *Let us consider a simple root z_* of $f \in \mathrm{Hol}(U)$ and $z_0 \in \mathrm{D}(z_*, r/3)$ for some $0 < r \leq \mathrm{dist}(z_0, \partial U)$ such that there exists a disk B' satisfying*

$$f'(\mathrm{D}(z_0, r)) \subset B' \qquad \text{with} \qquad \mathrm{dist}(B', 0) > 2\,\mathrm{diam}(B'), \tag{13}$$

then $\mathrm{D}(z_0, r)$ is contained in the attraction bassin $\mathcal{A}(z_*)$ of z_* for Newton's method.

Proof. Let us consider $B = \mathrm{D}(z_0, r)$ and $f'(B) \subset B'$ where $B' = \mathrm{D}(d, r')$ is a disk such that (13) holds, *i.e.* $|d| > 5r'$. In particular B' does not contain zero. Given $z_0 + h \in \mathrm{D}(z_0, r)$, one has:

$$\begin{aligned} N_f(z_0+h) &= z_0 + h - \frac{f(z_0+h) - f(z_0) + f(z_0) - f(z_*)}{f'(z_0+h)} \\ &= z_0 + h\left(\int_0^1 1 - \frac{f'(z_0+th)}{f'(z_0+h)}dt\right) - (z_0 - z^*)\int_0^1 \frac{f'((1-t)z_* + tz_0)}{f'(z_0+h)}dt \end{aligned}$$

The ratio of two values $d + \vartheta_1$ and $d + \vartheta_2$ in B' satisfies:

$$\left|1 - \frac{d+\vartheta_1}{d+\vartheta_2}\right| = \frac{|\vartheta_2 - \vartheta_1|}{|d+\vartheta_2|} \leq \frac{2r'}{|d| - r'} < \frac{1}{2} \quad \text{thus} \quad \left|\frac{d+\vartheta_1}{d+\vartheta_2}\right| < \frac{3}{2}. \tag{14}$$

One thus has $|N_f(z_0+h) - z_0| < \frac{1}{2}|h| + \frac{3}{2}|z_0 - z_*| < \frac{r}{2} + \frac{3}{2}\cdot\frac{r}{3} = r$. As $\mathrm{D}(z_0, r)$ is forward invariant under N_f, this disk is contained in the Fatou set of N_f. The iterates of N_f on $\mathrm{D}(z_0, r)$ thus converge to z_*. □

Remark 3. *The assumption $|z_0 - z_*| < r/3$ of Theorem 4 can be checked using Theorem 3, namely that for some $r'' < r'$:*

$$f'(\mathrm{D}(z_0, r/3)) \subset \mathrm{D}(d, r'') \qquad \textit{and} \qquad r(|d| - r'') > 3|f(z_0)|.$$

In that case, one can claim $N_f(\mathrm{D}(z_0, r)) \subset \mathrm{D}(z_0, \alpha r)$ with

$$\alpha = \frac{1}{2} + \frac{1}{3}\left[1 + \frac{1}{4} + \frac{r''}{4r'}\right] < 1$$

because $(1-t)z_* + tz_0 \in \mathrm{D}(z_0, r/3)$, which improves (14) slightly. For example, if $r'' \simeq r'/3$, then $\alpha \simeq 0.944$.

In practice (see §4), Theorem 3 is applied with a much smaller radius than Theorem 4 because it is desirable to localize the root as precisely as possible, yet have the largest possible domain of convergence for Newton's method.

3 Strict control of errors in floating point arithmetic

We need a theoretical background for handling all the errors that occur within the numerical computations that are necessary to prove that the assumptions of Theorems 3 and 4 are indeed satisfied. It is often (wrongly) believed that integer arithmetic is the only one apt for formal verification. We intent to show here that proofs can also be carried in floating point (FP) arithmetic, even with complex numbers.

The certification of a computation is only possible if the implementation choices respect a universal norm, as for example the one defined by the IEEE 754 standards [IEE]. We use the MPFR library [MPFR] because it offers a reliable implementation of arbitrary precision that has been extensively tested. One of its main feature is the guarantee of proper handling

of roundings, which is essential for the certification process described below. However, any other compliant implementation could equally be used as a foundation.

Working with complex numbers requires special care because, contrary to the real case, multiplications in $\mathbb{C}$ are not elementary operations. In what follows, we will restrict our attention to the fields operations: $+$, $\times$.

3.1 Finite precision arithmetic

An ideal model of arithmetic with a finite precision $N \in \mathbb{N}^*$ consists in considering the following discrete subset of real numbers

$$\hat{\mathcal{R}}_N = \{0\} \cup \bigcup_{e \in \mathbb{Z}} 2^e \mathcal{Z}_N, \tag{15}$$

where

$$\mathcal{Z}_N = \left\{ \pm\Big(2^{-1} + \sum_{j=2}^{N} b_j 2^{-j}\Big) \text{ with } b_2, \dots, b_N \in \{0,1\} \right\} = \pm 2^{-N} [\![2^{N-1}, 2^N - 1]\!]. \tag{16}$$

Here and below, the notation $[\![m, n]\!]$ for $m \leq n \in \mathbb{N}$ denotes the integer interval $[m, n] \cap \mathbb{N}$. When $x \in 2^e \mathcal{Z}_N$, one defines $e = e(x)$ as the *exponent* of x. The $(b_j)_{1,\dots,N}$ are called the *bits* of x, with the convention that $b_1 = 1$. Let us point out that if $N' \geq N$ is a larger precision, then $\hat{\mathcal{R}}_N \subset \hat{\mathcal{R}}_{N'}$.

Practical implementations of finite precision arithmetic are restricted by physical contingencies; one then considers instead the finite set:

$$\mathcal{R}_N = \{\pm 0\} \cup \bigcup_{e=e_{\min}}^{e_{\max}} 2^e \mathcal{Z}_N \cup \{\pm\infty, \mathrm{NaN}\}, \tag{17}$$

where $-e_{\min}$ and $e_{\max}$ are (configurable but fixed) large positive integers. The additional symbols allow one to handle numerical exceptions without producing errors (NaN stands for *not a number*). For efficient processing by the hardware, bits are packed in groups of 8 called a *byte*, and packs of bytes (typically 4 or 8) are called *limbs*.

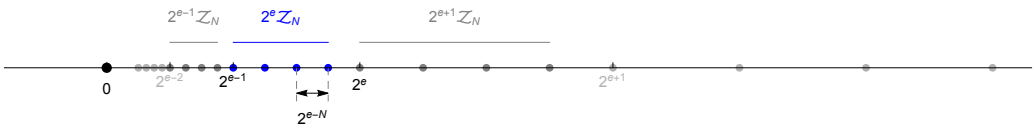


Figure 1: Representation of $2^e \mathcal{Z}_4 \cap \mathbb{R}_+$ for five consecutive values of e. Note how the gap between adjacent vertices varies with the exponent e, and the special role of zero as an accumulation point of $\hat{\mathcal{R}}_N$.

The arithmetic operations $+_N$ and $\times_N$ are naturally defined on $\hat{\mathcal{R}}_N$ by

$$\forall x, y \in \hat{\mathcal{R}}_N, \qquad x +_N y = \hat{\mathrm{R}}_N(x + y) \quad \text{and} \quad x \times_N y = \hat{\mathrm{R}}_N(x \times y),$$

where $\hat{\mathrm{R}}_N : \mathbb{R} \to \hat{\mathcal{R}}_N$ is the rounding operator that rounds to the nearest lattice point. Let us also define the operators $\hat{\mathrm{R}}_N^{\pm}$ that round respectively always up and always down, and the practical ones R_N, $\mathrm{R}_N^{\pm}$, that are valued in $\mathcal{R}_N$. By a convention, called *flush to zero*, which is not an IEEE 754 standard but is the choice made in the MPFR library, $\mathrm{R}_N(x) = 0$ if $|x| < 2^{e_{\min}}$ even though 0 is not the nearest lattice point if $|x| > 2^{e_{\min}-1}$. Similarly, $|\mathrm{R}_N(x)| = \infty$ if $|x| > 2^{e_{\max}}(1 - 2^{-N})$.

For a given number $x \in \hat{\mathcal{R}}_N \backslash \{0\}$, the gap that separates it from its farthest immediate neighbors is

$$\widehat{\mathrm{ulp}}(x) = 2^{e(x)-N}. \tag{18}$$

By convention, $\widehat{\mathrm{ulp}}(0) = 0$. The name of this operator is *unit on last position* because it reflects the metric effect of a change of one unit on the least significant bit b_N. If $x \in \mathcal{R}_N$ is a regular number *i.e.* if $e_{\min} \leq e(x) \leq e_{\max}$, one sets $\mathrm{ulp}(x) = \widehat{\mathrm{ulp}}(x)$. By convention, $\mathrm{ulp}(\pm 0) = 2^{1+e_{min}}$ and $\mathrm{ulp}(\pm\infty) = \infty$, which ensures the following statement. Note that the usual implementation choice is $e(0) = e_{\min} - 1$.

Proposition 4. *For $x \in \mathbb{R}$, one has*

$$\left|\hat{\mathrm{R}}_N(x) - x\right| \leq \tfrac{1}{2}\widehat{\mathrm{ulp}}(\hat{\mathrm{R}}_N(x)) \qquad \textit{and} \qquad |\mathrm{R}_N(x) - x| \leq \tfrac{1}{2}\,\mathrm{ulp}(\mathrm{R}_N(x)).$$

Remark 5. *IEEE 754 standards require gradual underflow instead of flush to zero. This means that $\mathcal{R}_N$ should be complemented with a set of denormalized numbers*

$$\{k2^{e_{min}-N}\,;\, k \in \mathbb{Z},\ \ |k| < 2^{N-1}\},$$

called subnormals, which ensures that the lattice $\mathcal{R}_N$ is regular near zero. In that case, $\mathrm{ulp}(x) = 2^{e_{min}-N}$ for all those additional points, including $x = 0$. With this alternate convention, Proposition 4 still holds. The MPFR library offers the possibility of emulating the norm, but it is not the default behavior.

3.2 Interval arithmetic

Interval arithmetic is a standard topic in numerical analysis [vdH09], [Rok01] whenever dependable results are critical. Reliable and fast libraries exist, like MPFI [RR05] or `[ARB]`.

Given a continuous numeric function f, the goal of interval arithmetic is to bound, as accurately as possible, the set to which $f(x)$ belongs when the prior knowledge on x is limited to a set of inequalities, *e.g.* $a \leq x \leq b$. The main challenge is to take dependency into account: for example, $f(x) = x^2 + x$ maps $[-1, 1]$ to $[-\frac{1}{4}, 2]$ while $g(x, y) = xy + x$ maps $[-1, 1]^2$ to $[-2, 2]$.

The following statement is an immediate but essential consequence of Proposition 4.

Proposition 6. *For $x_a, x_b \in \mathcal{R}_N$ and $r_a, r_b \in \mathcal{R}_M$, one has:*

$$\mathrm{I}(x_a, r_a) + \mathrm{I}(x_b, r_b) \subset \mathrm{I}\left(x_a +_N x_b\,,\, \mathrm{R}_M^{+}\left(r_a + r_b + \tfrac{1}{2}\,\mathrm{ulp}(x_a +_N x_b)\right)\right),$$

$$\mathrm{I}(x_a, r_a) \times \mathrm{I}(x_b, r_b) \subset \mathrm{I}\left(x_a \times_N x_b\,,\, \mathrm{R}_M^{+}\left(r_a r_b + r_a|x_b| + r_b|x_a| + \tfrac{1}{2}\,\mathrm{ulp}(x_a \times_N x_b)\right)\right)$$

where $\mathrm{I}(x, r) = [x - r, x + r]$ denotes closed intervals along the real line.

In practice, the new radius is computed using the R_M^+ operator for each intermediary computation to ensure that one gets a certifiable upper bound.

3.3 Naive rectangle arithmetic

For our purpose in complex dynamics, a naive use of a tensorized interval arithmetic faces the following shortcoming (see Fig. 2): if a is in $z+[-r,r]+i[-r,+r]$ with $|z|=1$, then a^2 belongs to $z^2+(2r|z|_1+r^2)\,[-1,1]+i(2r|z|_1+2r^2)\,[-1,1]$, with $|z|_1=|\operatorname{Re} z|+|\operatorname{Im} z|$, which means that, when $\operatorname{Arg} z \simeq \pi/4$, the size of the uncertainty box is roughly multiplied by $2\sqrt{2}$ instead of the factor 2 imposed by the derivative. If this happens at many iterations of the square function, the resulting precision loss is catastrophic. The average value of $\log|e^{i\theta}|_1$ on the unit circle is $\beta = 0.2365$. By the Birkhoff ergodic theorem, for a typical starting point a with $|a|=1$, computing n iterates of the map $z \mapsto z^2$ starting at a induces a cumulative multiplicative loss of precision of about $e^{\beta n}$. For example, for $n=200$, a loss of precision of order of 10^{20} should be expected.

3.4 Disk arithmetic

The idea of disk arithmetic consists in studying what optimal outcome can be deduced from the prior knowledge that $a \in \mathrm{D}(z,r)$. While the idea is not new [PP98], its usage has remained, up to now, fairly uncommon.

The practical gain of substituting disks to squares is illustrated on Figure 2. If $a \in \mathrm{D}(z,r)$, then a^2 belongs to $\mathrm{D}(z^2, 2r|z|+r^2)$. If r is small enough to ensure that $r^2 \ll 2r|z|$ but without being necessarily tiny, the first observation is that each iteration of the square function multiplies the radius of the uncertainty disk by roughly a factor $2|z|$ instead of $2|z|_1$; one thus gains a casual $\sqrt{2}$ factor over the naive use of interval arithmetic.

The second observation is more profound and pertains to small values of r: for any conformal map f, the image of a small disk is almost a disk. Naturally, a nearly circular shape can be enclosed within a disk of a barely larger diameter. In practice, r is infinitesimally small and the derivative of the map on the disk of radius r is almost constant. In comparison to the unavoidable action of the differential $f'(z)dz$ at the center point, the additive loss on the bounding radius of $f(\mathrm{D}(z,r))$ is then of order r^2 and the multiplicative loss is thus about $1+r/|f'(z)|$. For example, if we perform $k=10^5$ iterations with bounded derivatives $|f'(z)| \geq 1$ along the orbit, starting from a radius $r=10^{-20}$, then the cumulative multiplicative error of the disk arithmetic method is about $(1+r)^k \simeq 1+kr = 1+10^{-15} \simeq 1$.

Using disk arithmetic is the gateway to getting bounds that are both proven and almost optimal, even after a high number of iterates. It is a first and essential step towards computational proofs in complex dynamics.

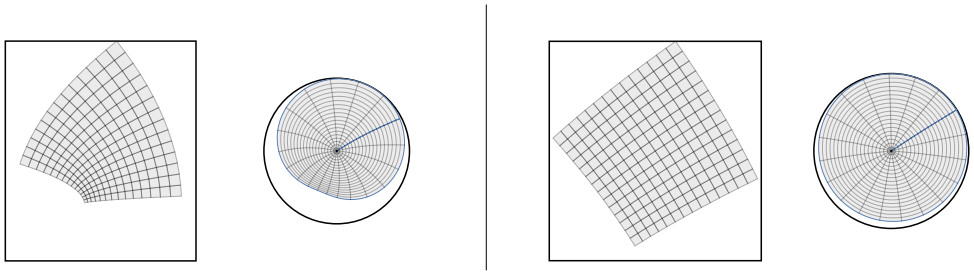


Figure 2: Comparison between the image by $f(z) = z^2$ of a square centered at $z = e^{3i\pi/16}$ of side $2r$, and that of a disks of center z and diameter $2r$ for $r = 0.5$ (left) and $r = 0.1$ (right). The images of the square and of the disk are drawn at the same scale, and compared respectively to the smallest possible enclosing box or disk centered around z^2.

For $z_a = x_a + iy_a$, $z_b = x_b + iy_b \in \hat{\mathcal{R}}_N + i\hat{\mathcal{R}}_N$, the sum $z_a +_N z_b$ is naturally defined in components

$$z_a +_N z_b = (x_a +_N x_b) + i(y_a +_N y_b).$$

The radical difference between interval and disk arithmetic is that the computation of the product of complex numbers requires four exact multiplications on the real line, followed by two additions. Therefore, the product requires an intermediary precision $N' \geq N$:

$$z_a \times_{N,N'} z_b = (x_a \times_{N'} x_b) -_N (y_a \times_{N'} y_b) + i\left[(x_a \times_{N'} y_b) +_N (y_a \times_{N'} x_b)\right].$$

Intermediary products are guaranteed exact only if $N' \geq 2N$. Let us extended the definition of ulp to complex finite precision numbers by

$$\operatorname{ulp}(x + iy) = \sqrt{\operatorname{ulp}(x)^2 + \operatorname{ulp}(y)^2}.$$

The following statement generalises Proposition 6 and is at the heart of the implementation of our library `[Mandel]`.

Theorem 5. *For centers $z_a, z_b \in \mathcal{R}_N + i\mathcal{R}_N$, radii $r_a, r_b \in \mathcal{R}_M$, one has:*

$$\mathrm{D}(z_a, r_a) + \mathrm{D}(z_b, r_b) \subset \mathrm{D}\left(z_a +_N z_b\,,\, \mathrm{R}^+_M\left(r_a + r_b + \tfrac{1}{2}\operatorname{ulp}(z_a +_N z_b)\right)\right).$$

For any intermediary precision $N' \geq N$, the product of exact centers satisfies:

$$z_a z_b \in \mathrm{D}\left(z_a \times_{N,N'} z_b, R_*\right),$$

where

$$R_* = \mathrm{R}^+_M\left(\tfrac{1}{2}\operatorname{ulp}(z_a \times_{N,N'} z_b) + \tfrac{1}{2}\sum_{\substack{u=x_a,y_a\\ v=x_b,y_p}} \operatorname{ulp}(u \times_{N'} v)\right).$$

The product of disks satisfies:

$$\mathrm{D}(z_a, r_a) \times \mathrm{D}(z_b, r_b) \subset \mathrm{D}\left(z_a \times_{N,N'} z_b\,,\, \mathrm{R}^+_M\left(r_a r_b + r_a|z_b| + r_b|z_a| + R_*\right)\right).$$

Finally, if $x \in \mathcal{R}_N$ and $r \in \mathcal{R}_M$, the scaling transform of the disk satisfies:

$$\mathrm{I}(x, r) \times \mathrm{D}(z_a, r_a) \subset \mathrm{D}\left(x \times_N z_a\,,\, \mathrm{R}^+_M\left((|x| + r)r_a + r|z_a| + \tfrac{1}{2}\operatorname{ulp}(x \times_N z_a)\right)\right).$$

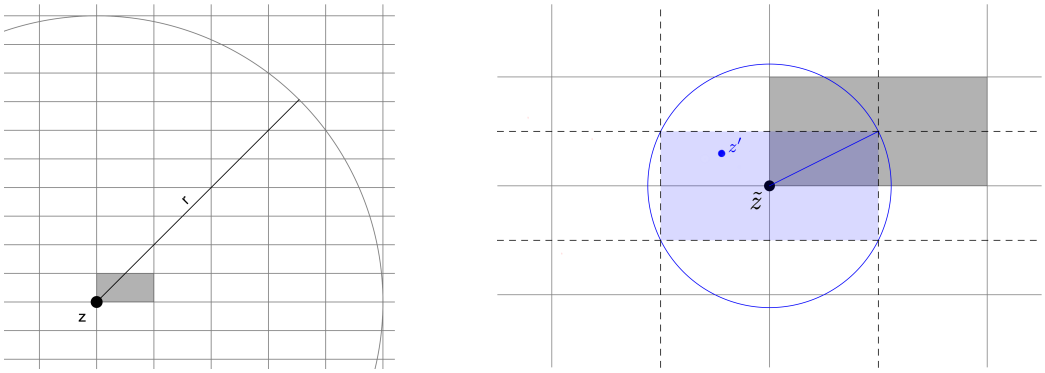


Figure 3: The key points to the proof of Theorem 5 : $\mathcal{R}_N \times \mathcal{R}_N$ grid (left) and the rounding strategy of $z' \in \mathbb{C}$ to $\tilde{z} \in \mathcal{R}_N \times \mathcal{R}_N$ within $\frac{1}{2}\operatorname{ulp}\tilde{z}$ (right).

The proof of the theorem is straightforward and the complex extension of the definition of ulp is illustrated on Figure 3. In our library, the role of R_* is implemented as an on-the-fly modification of ulp.

To ease the tedious task of checking Theorem 5, let us recall a few ground rules. Having Figure 3 in mind may help convey the key points. The center of a disk $\mathrm{D}_{z,r}$ is always considered exact; it belongs to $\mathcal{R}_N \times \mathcal{R}_N$. The corresponding ulp is thus a rectangle of dimensions $\operatorname{ulp} x \times \operatorname{ulp} y$ where $z = x + iy$. When computing the result of an operation, the new exact center $z' \in \mathbb{C}$ does not belong, in general, to the grid $\mathcal{R}_N \times \mathcal{R}_N$. The real and imaginary parts of z' are rounded to their closest value, which gives the new center $\tilde{z} \in \mathcal{R}_N \times \mathcal{R}_N$. To compensate, one needs to increase the radius of the disk by, at most, half the diagonal of the ulp rectangle. The claim that the center can be assumed exact is thus restored and one is ready for the next operation.

Remark 7. *The product of complex numbers is a multi-step operation over* $\mathbb{R}$*. As such, additional* ulp *have to be added to bound the successive rounding errors in each intermediary step.*

Remark 8. *There is a slightly tighter upper bound for the radius* R_**, namely*

$$R^* = \mathrm{R}_M^+ \Big\{ \tfrac{1}{2} \operatorname{ulp} \Big(z_a \times_{N,N'} z_b + \operatorname{ulp}(x_a \times_{N'} x_b) + \operatorname{ulp}(y_a \times_{N'} y_b) + i \left[\operatorname{ulp}(y_a \times_{N'} x_b) + \operatorname{ulp}(x_a \times_{N'} y_b) \right] \Big) \Big\}.$$

However, the code complexity and computational cost of using R^* *are unreasonably high compared to that of using* R_* *and are not justified for the expected gain.*

4 Practical application: root certification at the tera-scale

Of special interest for the study of the Mandelbrot set are the families of polynomials

$$p_0(z) = 0, \qquad p_{n+1}(z) = p_n(z)^2 + z \tag{19}$$

and

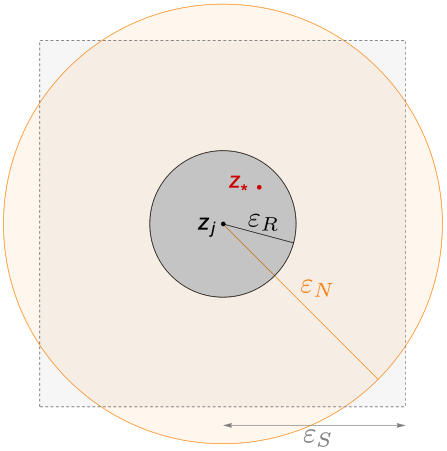
$$q_{\ell,k}(z) = p_{\ell+k}(z) - p_\ell(z). \tag{20}$$

For $n = \ell + k \geq 1$, one has $\deg p_n = \deg q_{\ell,k} = 2^{n-1}$.

We are releasing an exhaustive and proven list [Mand.DB] of all the roots of p_n (called hyperbolic centers) for n up to period 41 and all the roots of $q_{\ell,k}$ (called pre-periodic or Misiurewicz-Thurston parameters) of order $\ell + k \leq 35$. By proven, we mean that each point z_j in our database comes with a mathematical statement and a numerical proof, based on disk arithmetic, that guaranties two things: an exact root $\tilde{z}_j$ lies within a certain maximal tolerance from our numerical value, and a standard refinement technique (Newton's method) applied to z_j converges to $\tilde{z}_j$.

Let us underline that the following theorem thus contains about 3×10^{12} individual statements, whose proof are computer assisted.

Theorem 6. *There are constants ε_R, ε_N and ε_S given in Figure 4 for which the following holds. In the database for the roots of p_n or $q_{\ell,n}$, each published value $z_j \in 2^{-126}(\mathbb{Z} + i\mathbb{N})$ can be paired with a unique actual root $z_* \in \mathrm{D}(z_j, \varepsilon_R)$. This pairing is bijective in the upper half plane $\operatorname{Im} z \geq 0$. Each pair of published values z_j, z_k satisfies $|z_j - z_k| \geq \varepsilon_S$ and either $z_j \in \mathbb{R}$ or $|z_j - \bar{z}_j| \geq \varepsilon_S$. Lastly, the disk $\mathrm{D}(z_j, \varepsilon_N)$ is entirely contained in the attraction bassin of z_* for the Newton method of the associated polynomial.*



	$p_{n\leq 40}$	p_{41}	$q_{\ell,n}$
ε_R	10^{-30}		10^{-35}
ε_N	10^{-24}	10^{-25}	10^{-32}
ε_S	3.23×10^{-27}		8.08×10^{-28}

Figure 4: For each root z_* with $\operatorname{Im} z_* \geq 0$, the published value $z_j \in 2^{-126}(\mathbb{Z} + i\mathbb{N})$ comes with different radii. The radius of separation ε_S guaranties proper counting, the radius of certification ε_R is such that $z_* \in \mathrm{D}(z_j, \varepsilon_R)$ and the radius of convergence ε_N ensures that the $\mathrm{D}(z_j, \varepsilon_N)$ is contained in the bassin of attraction of z_* for the Newton method.

Proof. Using the controlled rounding features of the MPFR library [MPFR], we have implemented disk arithmetic *i.e.* Theorem 5. This allows us to provide, for each root candidate z_j, a numerical proof that the assumptions of Theorem 3 on the localisation of each root can indeed be applied. Thus, we can certify that the published coordinates of each

root differ from an actual root by no more than 10^{-30} for p_n or 10^{-35} for $q_{\ell,n}$. Similarly, using disk arithmetic with ε_N, we can check the assumptions of Theorem 4 which ensures the claim on the Newton bassin. Next, each pair of two published values or any pair with conjugate values satisfies the separation assumption

$$\max\{|\operatorname{Re}(z_1 - z_2)|, |\operatorname{Im}(z_1 - z_2)|\} \geq \varepsilon_S. \tag{21}$$

because it is build into our storage structure (see `nset` in [MV25a]). This separation guaranties that it is possible to count unambiguously all the real roots, and all the non real roots in the upper half-plane. Finally, comparing our root count with the theoretical root count of p_n and $q_{\ell,n}$ (whose multiplicities are known, see *e.g.* [MV25b]) ensures our bijection claim. □

Remark 9. *We are confident that each component of the published values is exact up to*

$$\pm\tfrac{1}{2}\operatorname{ulp}(\operatorname{Re}\texttt{u128}) = \pm 2^{-127} \simeq 5.9 \times 10^{-39},$$

which is better than the ε_R radius claimed in Theorem 6. Indeed, once we have a z_j value that passes the certifications for both disk arithmetic and Newton bassin, we can refine its value using Newton's method until reaching a fixed point at the desired precision, up to a final rounding error.

Remark 10. *The complexity of checking that $P(z) = 0$ using our method is $O(V_d)$ where V_d is the arithmetic cost of evaluating P. In the case at hand, the polynomials are defined recursively so $V_d = O(\log_2 d)$. In general, Hörner's method ensures only $V_d = O(d)$. For a recent improvement, on average and with finite precision, see [AMV22].*

At the terabyte scale, the possibility of bit corruption becomes a significant issue. Besides the previous statement, practical precautions must therefore be taken to ensure the integrity of the data along the whole production chain, from the initial computation that finds a root to its final storage in a file. This protection must also extends to any subsequent use of the stored values.

Our library `[Mandel]` incorporates a strict data certification procedure. Our `nset` file format implements a header that contains the MD5 checksum of the data stored in the rest of the file. When writing a file, a checksum of the original data is first computed in memory, then the file is written onto the disk and the checksum is written in the file header. To detect subsequent data corruption (due to an error in a file transfert or a random bit flip), each time a file is loaded, the checksum of the data stored in the file is computed again and checked against its original value stored in the header. This protocol ensures the data integrity once the original MD5 stamp has been generated.

Lastly, one needs to check that a random memory corruption has not affected the initial creation process, after the value was computed but before the original MD5 checksum was generated. The only sensible way to guard against this problem is to perform a new independent certification (count and proofs) of Theorem 6 of all the data written on the disk, which we did using the applications `hypCount`, `hypProve`, `misCount` and

`misProve` from our library [Mandel]. To encourage independent verifications, the code of those applications has been kept as minimalistic as possible.

The only uncertainty left is a data corruption that could happen after this second certification but would remain undetected by the MD5 checksums. As MD5 is a 128 bit cryptographic hash function (granted that our database is not subject to a malicious attack that would actively seek this vulnerability), the probability of such an event is of order $2^{-128} \simeq 3 \times 10^{-39}$.

References *

[Abe73] O. Aberth. Iteration methods for finding all zeros of a polynomial simultaneously,. *Math. Comput.*, 27(122):339–344, 1973.

[AMV22] R. Anton, N. Mihalache, and F. Vigneron. Fast evaluation of real and complex polynomials. *Numerische Mathematik*, 157:355-408, 2025.

[AMV23] R. Anton, N. Mihalache, and F. Vigneron. A short ODE proof of the fundamental theorem of algebra. *The Mathematical Intelligencer*, 2023.

[BF00] D. A. Bini and G. Fiorentino. Design, analysis and implementation of a multiprecision polynomial rootfinder. *Numer. Algorithms*, 23:127–173, 2000.

[BR14] D. A. Bini and L. Robol. Solving secular and polynomial equations: a multiprecision algorithm. *J. Comput. Appl. Math.*, 272:276–292, 2014.

[GSKC16] K. Ghidouche, A. Sider, L.Z. Khodja, and R. Couturier. Two parallel implementations of Ehrlich-Aberth algorithm for root-finding of polynomials on multiple GPUs with OpenMP and MPI. In *Intl Conference on Computational Science and Engineering*, 2016.

[Hen74] P. Henrici. *Applied and computational complex analysis, Volume 1: Power series, integration, conformal mapping, location of zeros.* Wiley, 1974.

[HSS01] J.H. Hubbard, D. Schleicher, and S. Sutherland. How to find all roots of complex polynomials by Newton's method. *Invent. math.*, 146:1–33, 2001.

[IEE] IEEE 754. https://en.wikipedia.org/wiki/IEEE_754.

[KI04] N. Kyurkchiev and A. Iliev. Failure of convergence of the newton-weierstrass iterative method for simultaneous rootfinding of generalized polynomials. *Computer and Mathematics with Applications*, 47:441–446, 2004.

[MV25a] N. Mihalache, F. Vigneron. How to split a tera-polynomial. *Preprint ArXiv 2402.06083.*

[MV25b] N. Mihalache, F. Vigneron. Factorization of the quadratic Misiurewicz-Thurston polynomials. *Preprint ArXiv 2506.17662.*

[PP98] M.S. Petković and L.D. Petković. *Complex interval arithmetic and its applications*, volume 105. Wiley-VCH, 1998.

[Rok01] J.G. Rokne. *Interval Arithmetic and Interval Analysis: An Introduction.* in book Granular Computing: An Emerging Paradigm, 2001.

[RR05] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the MPFI library. *Reliable Computing*, 11:275–290, 2005.

* Numerical libraries are typeset `[lib]` and are listed at the end of the bibliography.

[RSS17] M. Randig, D. Schleicher, and R. Stoll. Newton's method in practice II: The iterated refinement Newton method and near-optimal complexity for finding all roots of some polynomials of very large degrees. arXiv:1703.05847, 2017.

[RSS20] B. Reinke, D. Schleicher, and M. Stoll. The weierstrass root finder is not generally convergent. ArXiv:2004.04777, 2020.

[SS17] D. Schleicher and R. Stoll. Newton's method in practice: finding all roots of polynomials of degree one million efficiently. *Theor. Comput. Sci.*, 681:146–166, 2017.

[Sut89] S. Sutherland. *Finding root of complex polynomials with Newton's method.* PhD thesis, Boston University, 1989.

[vdH09] Joris van der Hoeven. Ball arithmetic. *Technical report: HAL-00432152.*, 2009.

[Wil84] J. H. Wilkinson. *The perfidious polynomial*, pages 1–28. Studies in Numerical Analysis. G. H. Golub, 1984.

[ARB] F. Johansson. Arb library. https://github.com/fredrik-johansson/arb, 2012.

[Mandel] N. Mihalache and F. Vigneron. Mandel library: a Numerical Microscope onto the Mandelbrot set. https://github.com/fvigneron/Mandelbrot, 2024.

[Mand.DB] N. Mihalache and F. Vigneron. Complete certified list of hyperbolic centers of period ≤ 41 and of all Misiurewicz-Thurston points whose type (pre-period and period sum) is ≤ 35, available upon request to arrange transfer of $\sim$50TB of data.
Databases extract in CSV format: https://doi.org/10.5281/zenodo.15527027 (up to period 32, inclusive) and https://doi.org/10.5281/zenodo.15527762 (up to type 28, inclusive).

[MPFR] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier and P. Zimmermann. MPFR: a Multiple-Precision binary Floating-point library with correct Rounding. *ACM Trans. Math. Software*, 33(2):13–28. https://www.mpfr.org, 2007.

[1] **Nicolae Mihalache.** Univ Paris-Est Creteil, CNRS UMR 8050, LAMA, F-94010 Creteil, France and Univ Gustave Eiffel, LAMA, F-77447 Marne-la-Vallée, France

nicolae.mihalache@u-pec.fr

[2] **François Vigneron.** Université de Reims Champagne-Ardenne, Laboratoire de Mathématiques de Reims, UMR 9008 CNRS, Moulin de la Housse, BP 1039, F-51687 Reims

francois.vigneron@univ-reims.fr