Computational Complexity of Iterated Maps on the Interval

Christoph Spandl
Computer Science Department
Universität der Bundeswehr München
D-85577 Neubiberg, Germany

Abstract

The exact computation of orbits of discrete dynamical systems on the interval is considered. Therefore, a multiple-precision floating point approach based on error analysis is chosen and a general algorithm is presented. The correctness of the algorithm is shown and the computational complexity is analyzed. As a main result, the computational complexity measure considered here is related to the Ljapunow exponent of the dynamical system under consideration.

Keywords: Discrete dynamical systems, Ljapunow exponent, multiple-precision floating point arithmetic.

1 Introduction

Consider a discrete dynamical system (D, f) on some compact interval $D \subseteq \mathbb{R}$, called the phase space, given by a function $f : D \to D$, a recursion relation $x_{n+1} = f(x_n)$ and an initial value $x_0 \in D$. The sequence $(x_n)_n$ of iterates is called the orbit of the dynamical system in phase space corresponding to the initial value x_0. If such a dynamical system is implemented, that is a computer program is written for calculating a finite initial segment of the orbit for given x_0, care has to be taken in choosing the appropriate data structure for representing real numbers. Traditionally, IEEE 754 `double` floating point numbers [8] are used. However, if the dynamical system shows chaotic behavior, a problem arises. The finite and constant length of the mantissa of a `double` variable causes round off errors which are magnified after each iteration step. Only after a few iterations, the error is so big that the computed values are actually useless [10]. To put things right, multiple precision floating point libraries providing floating point numbers

with arbitrary high mantissa length have to be used. In the following, it is analyzed how the needed mantissa length behaves in multiple-precision computations of iterates of discrete dynamical systems. The mantissa length needed for floating point numbers such that any computed point of the orbit has a specified and guaranteed accuracy is examined. Therefore, a precise mathematical framework for floating point computations has to be established. The main result shows that the ratio of mantissa length to iteration length in the limit of iteration length to infinity is related to the Ljapunow exponent. This result also gives some advice for economically designing exact algorithms simulating one-dimensional discrete dynamical systems.

2 Roundoff Error, Error Propagation and Dynamic Behavior

In this section, the discrete dynamical system (D, f_μ) with $D = [0, 1]$ and $f_\mu : D \to D$, $f_\mu(x) := \mu x(1 - x)$ for some control parameter $\mu \in (0, 4]$ is investigated. In the literature, the recursion relation $x_{n+1} = f_\mu(x_n)$ is called the *logistic equation* [3]. When implementing the logistic equation on a real computer and demanding to obtain exact values for the orbit $(x_n)_n$, the analysis of roundoff errors and of error propagation requires some care. This is due to the fact that for some values of μ the dynamics is highly chaotic and therefore inaccuracies are magnified exponentially in time [4, 7].

In the following, for a given initial value x_0, the true orbit is denoted by $(x_n)_n$, whereas the really computed orbit, suffering from roundoff errors and error propagation, is denoted by $(\hat{x}_n)_n$. Note that even $\hat{x}_0$ may differ form x_0 since the conversion to a floating point number may cause the very first roundoff error. One goal of this section is to give a rigorous estimation of the total error in dependence of the iteration step n.

Calculating the orbit $(\hat{x}_n)_n$, two types of error are present. First, error propagation due to the iteration scheme and second the roundoff error caused by the calculation of f_μ. Now, let $\hat{x}_n$ for some $n \in \mathbb{N}$ be given. Then the true error after one iteration step is $\hat{x}_{n+1} - x_{n+1}$. Since in reality not $f_\mu(\hat{x}_n)$ is calculated but some erroneous approximation $\hat{f}_\mu(\hat{x}_n)$, the true error can be written as $\hat{x}_{n+1} - x_{n+1} = \hat{f}_\mu(\hat{x}_n) - f_\mu(x_n)$. Hence, the true error can be written as a sum

$$\hat{x}_{n+1} - x_{n+1} = (f_\mu(\hat{x}_n) - f_\mu(x_n)) + (\hat{f}_\mu(\hat{x}_n) - f_\mu(\hat{x}_n)) \tag{1}$$

of two terms. The first term describes solely the error propagation while the

second term gives exactly the newly produced error due to the approximate calculation of f_μ.

To handle the exact values of both errors computationally, interval arithmetic can be used [1]. Interval arithmetic can be seen in the setting here as a special case of the computational model of TTE [13], which gives a precise notion for describing computations over the real numbers. Another strongly related model, which in some sense reflects the situation here more adequate is the Feasible Real RAM model [2]. For the sake of simplicity however, an interval setting is used here. For any time step n, let the phase point x_n together with its error be represented by two floating point numbers x_n^l and x_n^u ($x_n^u \geq x_n^l$) with given mantissa length m_n forming an interval $[x_n^l, x_n^u]$. The interval is an enclosure of the real value x_n, that is $x_n \in [x_n^l, x_n^u]$. It is straightforward to transform the interval to a floating point value $\hat{x}_n$ of mantissa length m_n by setting

$$\hat{x}_n := gl\left(\frac{x_n^l + x_n^u}{2}\right) \tag{2}$$

where $gl(.)$ performs the rounding to nearest floating point number. The absolute error $e_n := |\hat{x}_n - x_n|$ of $\hat{x}_n$ can be estimated via the interval length $d_n := x_n^u - x_n^l$ by

$$e_n \leq \frac{1}{2} d_n + r_n \tag{3}$$

where r_n is an error introduced by the rounding operation $gl(.)$ in Equation 2. An upper bound on r_n will be discussed later, for now it suffices to say that in general it is small compared to d_n.

For doing an error analysis of the logistic equation analytically, some idealizing assumptions are made. First, the value of μ is assumed to be given with such a high precision that no interval representation is needed. Second, only the error propagation is considered caused by the initial error due to rounding x_0 to some floating point number of mantissa length m. Third, the value of r_n in Equation 3 is neglected. The recursion relation then reads in natural interval extension

$$\begin{aligned} x_{n+1}^l &= \mu x_n^l (1 - x_n^u) \\ x_{n+1}^u &= \mu x_n^u (1 - x_n^l) \end{aligned}$$

with the interval length d_n given by the recursion relation

$$\begin{aligned} d_{n+1} &= x_{n+1}^u - x_{n+1}^l = \mu(x_n^u - x_n^u x_n^l - x_n^l + x_n^u x_n^l) \\ &= \mu d_n \end{aligned}$$

with the obvious solution $d_n = \mu^n d_0$. Finally the absolute error e_n of $\hat{x}_n$ according to Equation 2 can be bounded from above by

$$e_n \leq \frac{1}{2} d_n = \frac{1}{2} \mu^n d_0. \tag{4}$$

The aim now is to calculate, for given $N \in \mathbb{N}$, $p \in \mathbb{Z}$ and mantissa length m, the orbit up to time N with relative error 10^{-p}. That is, for $(\hat{x}_n)_{0 \leq n \leq N}$ should hold

$$e_n = |\hat{x}_n - x_n| \leq 10^{-p} x_n \leq 10^{-p}. \tag{5}$$

The ideal assumptions require the somewhat unreal setting that the mantissa length has to be set to some finite, but big enough value m for representing x_0 and a virtually infinite value m_∞ for doing the iteration. Finally, some upper bound on d_0 is needed. The value of d_0 is given as the roundoff error by representing x_0 as a floating point number of mantissa length m. For that, the well known estimate

$$d_0 \leq 2^{-m+1} x_0 \leq 2^{-m+1} \tag{6}$$

exists. Combining (5), (4) and (6) gives as a sufficient condition

$$\mu^n \cdot 2^{-m} \leq 10^{-p} \tag{7}$$

for $n = 0, \ldots, N$.

The minimal m, fulfilling the precision requirement (5) on the relative error of x_n, which depends on x_0, N and p, is denoted by $m_{min}(x_0, N, p)$. So, the sufficient condition (7) gives an upper bound on $m_{min}(x_0, N, p)$ by

$$m_{min}(x_0, N, p) \leq \lceil p \cdot \mathrm{ld}(10) + N \cdot \max(0, \mathrm{ld}(\mu)) \rceil \tag{8}$$

where ld(.) is the logarithm to base 2. At that stage, a central quantity of this work is introduced which is a kind of complexity measure. The *loss of significance rate* $\sigma(x, p)$, which may depend on the initial value $x = x_0$ and the precision p is defined by

$$\sigma(x, p) := \limsup_{N \to \infty} \frac{m_{min}(x, N, p)}{N}.$$

This quantity describes the limiting amount of significant mantissa length being lost at each iteration step. Significant means here the part of the places being exact. A general treatment of this complexity measure is given in the next section. Roughly speaking, $\lceil \sigma(x_0, p) N + p \cdot \mathrm{ld}(10) \rceil$ is the mantissa length

for any floating point number needed in an algorithm doing the iteration starting with x_0 and calculating up to x_N, if the output should be precise up to p decimal places. Formula 8 gives an upper bound for the loss of significance rate by $\sigma(x, p) \leq \max(0, \mathrm{ld}(\mu))$.

It is interesting to see whether the upper bound calculated analytically, which needed strong idealizations, is in the region of the real value. So, the logistic equation was implemented using a multiple-precision interval library. For that purpose, the interval library MPFI [12] based on the multiple-precision floating point number library MPFR [6], both written in C, was used. For each control parameter μ ranging from 0.005 to 4 and a step size of 0.005, the orbit for initial condition 0.22 was calculated up to $N = 2000$. For each μ, the minimum mantissa length m_{min} needed to guarantee $e_n \leq 10^{-6} x_n$ for $n = 0, \ldots, N$ was searched. Then, $\sigma_{est} := m_{min}/N$ was calculated. The result is shown in Figure 1.

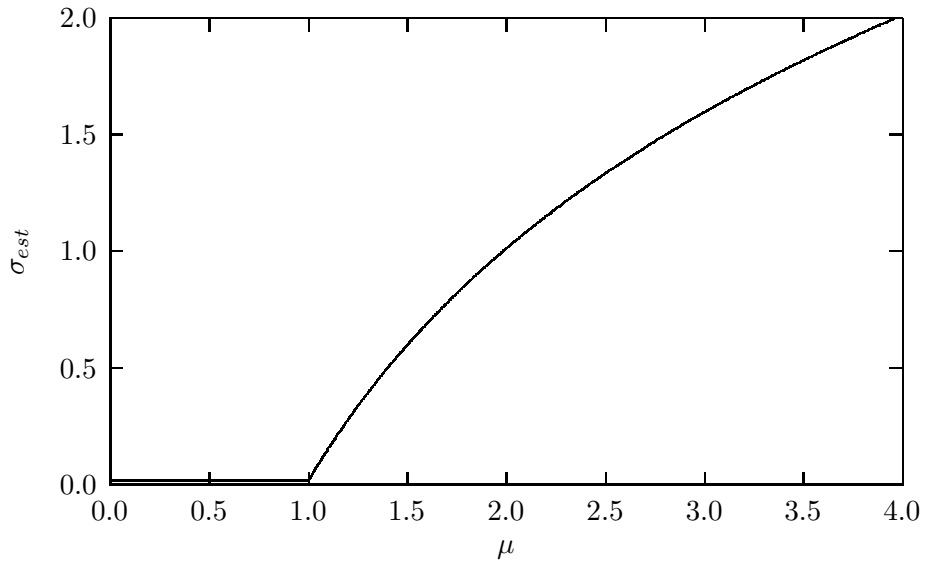

Figure 1: Estimated loss of significance rate for the logistic equation.

The curve shows that σ_{est} exceeds the analytical bound $\max(0, \mathrm{ld}(\mu))$ only slightly. So, the above made ideal assumptions seem to be valid. In [10], the logistic equation was also investigated for $\mu = 3.75$ using the exact

real arithmetic package iRRAM based on the Feasible Real RAM model [2]. In the paper, also the precision needed to guarantee the exactness of the first 6 decimal places are reported up to $N = 100000$. The values are in full agreement with the simulation results performed here.

Figure 1 show that for $\mu > 1$, the interval length d_n increases exponentially in time n. This result should be interpreted in terms of the dynamical behavior of the logistic equation. So, at this point is worth having an analytical look at the behavior of the dynamical system. Despite the fact that these results are well known [7, 5], they are reviewed here for the sake of self containment. First, the equation possesses in the range $D = [0, 1]$ exactly one fixed point $x^o = 0$ if $\mu \in (0, 1]$ and exactly two fixed pints $x^o = 0$ and $x^{(\mu)} = 1 - \frac{1}{\mu}$ if $\mu \in (1, 4]$. Since $f'_\mu(0) = \mu$ and $f'_\mu(x^{(\mu)}) = 2 - \mu$, x^o is a stable fixed point (an attractor, $|f'_\mu(x^o)| < 1$) for $\mu \in (0, 1)$ and an unstable fixed point (a repeller, $|f'_\mu(x^o)| > 1$) for $\mu \in (1, 4]$. If $\mu = 1$, the only fixed point x^o is hyperbolic ($|f'_1(x^o)| = 1$) and a bifurcation occurs at that value of the control parameter μ. If $\mu \in (1, 3)$, x^o becomes unstable and the newly occurring fixed point $x^{(\mu)}$ is stable. Finally, $\lim_{n\to\infty} f^n_\mu(x) = x^{(\mu)}$ for $\mu > 1$ and $\lim_{n\to\infty} f^n_\mu(x) = x^o$ if $\mu \leq 1$ holds for all $x \in (0, 1)$. If $\mu \in (0, 1)$, this is a direct consequence of the contraction mapping principle. If $\mu = 1$, observe that $f_1(x) < x$ holds for all $x \in (0, 1)$. Hence, any sequence $(f^n_1(x))_n$, $x \in (0, 1)$, is strictly decreasing and bounded from below. So it converges to the only fixed point x^o. For the case $\mu \in (1, 3)$, the interested reader is referred to the literature: [5], Proposition 5.3. At $\mu = 3$ a second bifurcation occurs and for $\mu > 3$ the system goes into a region of periodic behavior with period doubling bifurcations. Finally, for some $\mu < 4$, chaotic behavior is reached.

This analysis shows that in the parameter range $\mu \in (0, 3)$, the orbit converges to the stable fixed point for any initial value $x_0 \in (0, 1)$. Furthermore, there exists some closed interval $I \subseteq D$, which depends on μ, containing the stable fixed point such that $f_\mu(I) \subseteq I$ holds and f_μ is a contraction on I. The interval computation using a natural interval extension of the recursion function, on the other hand, is not very compatible with this picture. While for $\mu \in (0, 1)$, the results shown in Figure 1 are in agreement with the dynamical analysis, the calculations for $\mu \in (1, 3)$ are not handled very well by the interval arithmetic since the interval approach would suggest an exponential divergence of initially nearby orbits which is not true in reality. The reason is that the natural interval approach implicitly takes account only of the global behavior of f_μ in the form of a global Lipschitz constant $\max\{|f'_\mu(x)| : x \in D\} = \mu$. However, a local Lipschitz constant

$\max\{|f'_\mu(x)| : x \in [x_n^l, x_n^u]\}$ governs the real error propagation at time step n and also describes the dynamic behavior. This notion can be made precise and finally leads to a more efficient algorithm for computing orbits.

Let us return to Equation 1. The true error is the sum of the error propagation (first term) according to the iteration and the roundoff error due to the computation of f_μ (second term). The first term of Equation 1 can be handled using the mean value theorem, $|f_\mu(\hat{x}_n) - f_\mu(x_n)| = |f'_\mu(y_n)| \cdot |\hat{x}_n - x_n|$ with $y_n \in [\hat{x}_n - e_n, \hat{x}_n + e_n]$. This gives directly the bound

$$|f_\mu(\hat{x}_n) - f_\mu(x_n)| \leq \sup(|f'_\mu([\hat{x}_n - e_n, \hat{x}_n + e_n])|)e_n.$$

The second term can be estimated the following way. As discussed in [14], the roundoff error produced in calculating f_μ can be estimated by

$$|\hat{f}_\mu(\hat{x}) - f_\mu(\hat{x})| \leq 1.06K2^{-m}|f_\mu(\hat{x})|$$

where K is the number of rounding operations performed in computing $\hat{f}_\mu$ and m is the mantissa length of $\hat{x}$. In the case considered here, $K = 4$ follows since there are 3 arithmetic operations and the rounding of μ. It is further crucial to mention that the factor 1.06 is only valid if $K \leq 0.1 \cdot 2^m$ holds so that the mantissa length must not be chosen too small. Using the fact that $f_\mu(x) \leq \frac{\mu}{4}$ holds and $f_\mu(x) < x$ if $\mu \leq 1$, the unknown value $|f_\mu(\hat{x})|$ can be eliminated. This calculation shows that there exists a recursive equation on an upper bound $\overline{e}_n$ on e_n for all n:

$$\overline{e}_{n+1} = L(\hat{x}_n, \overline{e}_n)\overline{e}_n + 1.06K2^{-m}E_\mu(\hat{x}_n), \quad \overline{e}_0 = 2^{-m} \tag{9}$$

with $L(x, e) := \sup(|f'_\mu([x - e, x + e])|)$ and

$$E_\mu(x) := \begin{cases} x & \text{if } \mu \leq 1 \\ \frac{\mu}{4} & \text{if } \mu > 1 \end{cases}.$$

The approach now is not to calculate intervals, but pairs of values $\hat{x}_n$ and corresponding guaranteed error bounds $\overline{e}_n$. The difference to the interval concept is not to compute the errors *implicitly* but to compute them *explicitly* and independent of the values of interest. It should be mentioned that the approach described here is compatible with an interval approach using special centered forms, namely mean value forms [11]. However, the approach here explicitly devises values and errors, describes an automated error analysis, whereas an interval approach primarily does not disclose any error. The rounded values $\hat{x}_n$ are calculated as usual in floating point arithmetic except that multiple-precision floats are used. The guaranteed error

bounds are also calculated using floating point according to (9), where interval arithmetic is used for calculating L. Only standard precision is needed for calculating the error bounds. Implementing this improved algorithm using MPFR and MPFI, the setting as given in the interval case produces the result shown in Figure 2.

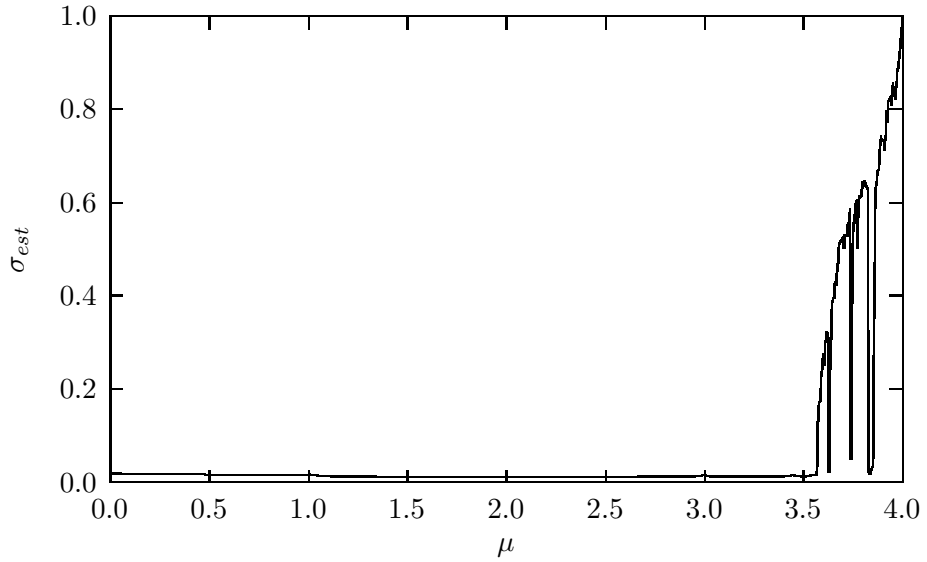

Figure 2: Estimated loss of significance rate for the logistic equation using the improved algorithm.

The curve reflects in the parameter range $\mu \in (0, 3)$ well the dynamic behavior. Furthermore, in the range $\mu \in [3, 4]$, the curve suggests a relation between the loss of significance rate and the Ljapunow exponent $\lambda(x)$ for the logistic map (for a curve of the Ljapunow exponent of the logistic map see [3]): $\sigma(x) = \max(0, \lambda(x))/\ln(2)$ for all $\mu \in (0, 4]$. To be complete, the definition of the Ljapunow exponent reads

Definition 2.1. Let (D, f) be a dynamical system, $D \subseteq \mathbb{R}$ compact and $f : D \to D$ continuously differentiable on the interior of D. Then the

Ljapunow exponent at x is defined by

$$\lambda(x) := \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \ln \left| f'(f^k(x)) \right| \tag{10}$$

if the limit exists.

The Ljapunow exponent may depend on x. However, the following properties hold:

(a) If (D, f) has an *invariant measure* ρ, then the limit in Equation 10 exists ρ-almost everywhere.

(b) Furthermore, if ρ is *ergodic* then $\lambda(x)$ is ρ-almost everywhere constant and equal to

$$\int_D \ln \left| f'(x) \right| \, \rho(dx).$$

These properties are a direct consequence of the Birkhoff ergodic theorem, see [9], Theorem 4.1.2 and Corollary 4.1.9.

3 The General Algorithm and its Complexity

Let D be a compact real interval and $f : D \to D$ a self mapping. In the following, f is assumed to be continuous on D, continuously differentiable on the interior of D and f' is bounded. Furthermore, f and f' are assumed to be computationally feasible. The precise definition of "computationally feasible" is given below.

In this section, a general algorithm for computing the iteration

$$x_{n+1} = f(x_n), \quad x_0 \in D \tag{11}$$

is presented. To be more precise, for given $N \in \mathbb{N}$ and $p \in \mathbb{Z}$, this algorithm computes a finite part of the orbit, $(x_n)_{0\le n\le N}$, exact in the sense that the relative error at each point x_n does not exceed 10^{-p}. The correctness of the algorithm and its computational feasibility is shown. Finally, its complexity is examined.

3.1 Syntax, Semantics and the Algorithm

The set of all computationally accessible real numbers are the floating point numbers of arbitrary mantissa length denoted by $\hat{\mathbb{R}}$. In the following, by a

floating point number any real number is meant which can be expressed by normalized scientific notation. Hence, the set $\hat{\mathbb{R}} \subseteq \mathbb{R}$ of all floating point numbers is countable infinite and therefore a natural basis for standard computability considerations. Let $\hat{x} \in \hat{\mathbb{R}}$ be some floating point number, then $\hat{x}$ has as an essential property, its *mantissa length* denoted by $\hat{x}.m$. Any real number x is represented in an algorithm concerning real computation by a pair $[x] \in \hat{\mathbb{R}}^2$ consisting of a floating point number $[x].fl$ approximating x and an upper bound on the relative error, $[x].err \geq 0$, also being a floating point number. Furthermore, the inequality $|[x].fl - x| \leq [x].err$ holds. The pair $[x]$ is called a *finite precision representation* of x. Although $[x].err$ has the property mantissa length, it is irrelevant in what follows. So, the mantissa length of $[x].err$ can be assumed to be some big enough constant value. Analogously, a function $f : D \to D$, $D \subseteq \mathbb{R}$, is called *computationally feasible* if a pair $[f]$ exists of a computable (partial) function $[f].fl : \hat{\mathbb{R}} \to \hat{\mathbb{R}}$ approximating f on D and a computable (partial) function $[f].erf : \hat{\mathbb{R}}^2 \to \hat{\mathbb{R}}$ giving an upper bound on the absolute error of $[f].fl$ in the sense $|[f].fl([x].fl) - f(x)| \leq [f].erf([x])$. Here, a partial function $\hat{f} : \hat{\mathbb{R}} \to \hat{\mathbb{R}}$ is called *computable* if $\hat{f}$ is computable as a string function over some finite alphabet where the floating point numbers are interpreted as finite strings. Finally, computability over integers, computability of functions with mixed arguments and computable predicates are defined in a standard way.

The algorithm with the above described specification reads

```
 1 Input parameter: x̂₀, N, p
 2 Initialize mantissa length m ← m₀
 3 do
 4    Initialize value and error [x] ← gl(x̂₀, m)
 5    for n = 0 to N do
 6        If prec([x], p) is true then
 7           If not printed print  n, [x].fl, [x].err
 8        else break
 9        [x] ← [f]([x])
10    [x].fl.m ← [x].fl.m + 1
11 while prec([x], p) is false
```

To initialize $[x]$, a rounding function $gl : \hat{\mathbb{R}} \times \mathbb{N} \to \hat{\mathbb{R}}^2$ is needed where $gl(\hat{x}_0, m_0).fl$ is a floating point number of mantissa length m_0 being the exactly rounded value of $\hat{x}_0$ for some rounding convention. Clearly, the value $gl(\hat{x}_0, m_0).err$ is an upper bound on the absolute rounding error, e.g. $gl(\hat{x}_0, m_0).err = \frac{1}{2} ulp(\hat{x}_0)$ if the rounding mode is to nearest. The predicate

$prec : \hat{\mathbb{R}}^2 \times \mathbb{Z} \to \{\mathbf{true}, \mathbf{false}\}$ is a test whether the relative error of $[x]$, $|[x].fl - x|/|x|$ if $x \neq 0$, is bounded by 10^{-p}. The semantics reads: If $[x] \in \hat{\mathbb{R}}^2$ is a finite precision representation of $x \in \mathbb{R}$ and $prec([x], p) = \mathbf{true}$ holds, the $|[x].fl - x| \leq 10^{-p}|x|$ follows.

In the following, some abbreviations are used occasionally. The floating point numbers and functions are indicated by a hat: $\hat{x} := [x].fl$ and $\hat{f} := [f].fl$. An over-bar indicates an error bound: $\overline{e} := [x].err$ and $\overline{erf} := [f].erf$. Hence, $[x]$ is equivalent to $(\hat{x}, \overline{e})$ and $[f]$ is equivalent to $(\hat{f}, \overline{erf})$.

Finally a remark on optimization. The algorithm is not optimized in the performance. Otherwise, in Line `10` something like $m \leftarrow 2m$ should be used. Here, the aim is to find the minimal m to guarantee some given upper bound on the relative error of x_n.

3.2 Feasibility and Correctness

It is clear, that the rounding function gl is computationally feasible. So lets begin with the predicate $prec$.

Proposition 3.1. *The computationally feasible formula*

$$prec((\hat{x}, \overline{e}), p) := \begin{cases} \mathbf{true} & \text{if } \overline{e} \leq \frac{10^{-p}}{1+10^{-p}}|\hat{x}| \\ \mathbf{false} & \text{else} \end{cases} \tag{12}$$

fulfills the above described semantics.

Proof. Let $(\hat{x}, \overline{e})$ be a finite precision representation of x. So, if $\overline{e} \leq 10^{-p}|x|$ holds, then also $|\hat{x}-x| \leq 10^{-p}|x|$ holds. If $(\hat{x}-\overline{e})(\hat{x}+\overline{e}) \geq 0$, then $|\hat{x}|-\overline{e} \leq |x|$ holds. Hence, if $(\hat{x} - \overline{e})(\hat{x} + \overline{e}) \geq 0$ and $\overline{e} \leq 10^{-p}(|\hat{x}| - \overline{e})$ holds, then also $|\hat{x}-x| \leq 10^{-p}|x|$. Finally, if $\overline{e} \leq \frac{10^{-p}}{1+10^{-p}}|\hat{x}|$ holds, then also $(\hat{x}-\overline{e})(\hat{x}+\overline{e}) \geq 0$.

Formula 12 only uses the accessible floating point values $\hat{x}$ and $\overline{e}$, basic arithmetics and finite tests. Hence, this formula is computationally feasible. □

Note that the definition of the predicate this way also gives **true** in the singular case where $\hat{x} = 0$ and $\overline{e} = 0$ and hence $x = 0$.

A computable formula for $\hat{f}$ is by assumption possible. To derive a computable formula for $\overline{erf}$ on the absolute error, return to Equations 1 and 9.

Proposition 3.2. *Assume that $\hat{f}(\hat{x})$ computes $f(\hat{x})$ up to a correctly rounded last bit in mantissa according to rounding convention. Then there exists a*

constant $K > 0$ such that the absolute error of $f(x)$ of the computation $[f]([x])$ is bounded from above by

$$L(\hat{x}, \overline{e}) \cdot \overline{e} + \frac{K2^{-m}}{1 - K2^{-m}} |\hat{f}(\hat{x})| \tag{13}$$

if $K2^{-m} < 1$. Here, $L(\hat{x}, \overline{e}) := \sup(|f'([\hat{x} - \overline{e}, \hat{x} + \overline{e}])|)$ and m is the mantissa length of $\hat{x}$: $\hat{x}.m$.

Furthermore, this bound is computable.

Proof. Using Equation 1 and following the calculations leading to Equation 9, $|\hat{f}(\hat{x}) - f(x)| \leq L(\hat{x}, \overline{e}) \cdot \overline{e} + |\hat{f}(\hat{x}) - f(\hat{x})|$ follows. According to the assumption on $\hat{f}$, $|\hat{f}(\hat{x}) - f(\hat{x})| \leq K2^{-m}|f(\hat{x})|$ holds, with a value $K \in \{1, 2\}$ depending on the rounding convention. However, $f(\hat{x})$ is unknown, only $\hat{f}(\hat{x})$ is accessible. To overcome this, set $\hat{f}(\hat{x}) - f(\hat{x}) = \delta f(\hat{x})$ with $|\delta| \leq K2^{-m}$. Since $|\delta| < 1$ holds, resolve to $f(\hat{x}) = \frac{1}{1+\delta}\hat{f}(\hat{x})$. Hence,

$$|\hat{f}(\hat{x}) - f(\hat{x})| = \left|\frac{\delta}{1+\delta}\right| \cdot |\hat{f}(\hat{x})| \leq \frac{K2^{-m}}{1 - K2^{-m}} |\hat{f}(\hat{x})|$$

follows. Since an upper bound on $L(\hat{x}, \overline{e})$ can be computed using global optimization techniques, e.g. with interval arithmetic, the above described bound is computable. □

To summarize, the mathematical iteration (11) is performed in the algorithm by iterating a value $\hat{x}_n$ approximating x_n with an upper bound on its absolute error $\overline{e}_n$ according to

$$\hat{x}_{n+1} = \hat{f}(\hat{x}_n) \qquad \hat{x}_0 = gl(x_0, m) \tag{14}$$

$$\overline{e}_{n+1} = \overline{L}(\hat{x}_n, \overline{e}_n)\overline{e}_n + \frac{K2^{-m}}{1 - K2^{-m}} |\hat{x}_{n+1}| \qquad \overline{e}_0 = \frac{K2^{-m}}{1 - K2^{-m}} |\hat{x}_0| \tag{15}$$

where $\overline{L}(\hat{x}_n, \overline{e}_n)$ is computable upper bound on $L(\hat{x}_n, \overline{e}_n)$ as described in the preceding proposition. This is Line `9` in the inner `for`-loop of the algorithm which is executed with successively increasing mantissa length m, controlled by the outer `do-while`-loop. Finally, it has to be shown that this outer loop eventually terminates. Therefore, two more propositions are needed.

Proposition 3.3. *Let x be a real number, $x \neq 0$ and $([x]_m)_{m \geq m_0}$ a sequence of finite precision representations of x with increasing mantissa lengths obeying $([x]_m).fl.m \geq m$ such that $\lim_{m\to\infty}([x]_m).err = 0$ holds and consequently $\lim_{m\to\infty}([x]_m).fl = x$. Then $\lim_{m\to\infty} prec([x]_m, p) =$ **true** follows for all $p \in \mathbb{Z}$.*

Proof. Since $x \neq 0$ and $\lim_{m\to\infty}([x]_m).err = 0$, there exists some $M \in \mathbb{N}$ such that for all $m \geq M$, $\frac{1}{2}|x| \leq |([x]_m).fl|$ and $([x]_m).err \leq \frac{10^{-p}}{2(1+10^{-p})}|x|$ holds for all $m \geq M$. Then, $prec([x]_m, p) = \textbf{true}$ holds for all $m \geq M$. □

The next proposition makes the link to Line 9 in the algorithm.

Proposition 3.4. *Let x_n be the n-th element of the orbit of Equation 11 and $([x_n]_m)_{m\geq m_0}$ a sequence given according to the recursion equations (14) and (15) with increasing mantissa length $([x_n]_m).fl.m \geq m$. Then $\lim_{m\to\infty}([x_n]_m).err = 0$ holds and consequently $\lim_{m\to\infty}([x_n]_m).fl = x_n$.*

Proof. Let $L := \sup(f'(D))$ and $\overline{L} \geq L$ be some computationally accessible value using some global optimization technique. Then Equation 15 leads to $\overline{e}_{n+1} \leq \overline{L}\overline{e}_n + \frac{K2^{-m}}{1-K2^{-m}}\overline{M}$ where $\overline{M} \geq \sup\{|x| : x \in D\}$ such that $|\hat{x}_n| \leq \overline{M}$ holds for all n. Iteration gives $\overline{e}_n \leq \overline{L}^n\overline{e}_0 + \frac{K2^{-m}}{1-K2^{-m}}\overline{M}\sum_{k=0}^{n-1}\overline{L}^k \leq \frac{K2^{-m}}{1-K2^{-m}}\overline{M}\sum_{k=0}^{n}\overline{L}^k$. Hence, for n fixed, $\lim_{m\to\infty}([x_n]_m).err = 0$ follows. □

These two propositions finish the correctness proof of the algorithm. They show that, if $x_n \neq 0$ for $n = 0, \ldots, N$, the outer loop eventually terminates for any $p \in \mathbb{Z}$.

3.3 Computational Complexity

After having presented the preliminary work, the main issue of the paper is addressed - the computational complexity of the presented algorithm. The complexity measure of interest here is the loss of significance rate already introduced informally in the last section. Here is the formal definition.

Definition 3.1. The minimal mantissa length, for which the described algorithm eventually halts is denoted by $m_{min}(x_0, N, p)$, where x_0, N and p are the corresponding input parameters. Then, the *loss of significance rate* $\sigma : \hat{\mathbb{R}} \cap D \times \mathbb{Z} \to \mathbb{R}$ is defined by

$$\sigma(x, p) := \limsup_{N\to\infty} \frac{m_{min}(x, N, p)}{N}. \tag{16}$$

However, to achieve bounds on the loss of significance rate, a technical difficulty has to be circumvented. Therefore, one more assumption on the dynamical system (D, f), additional to the ones already mentioned in the beginning of this section, has to be made.

Assumption 3.1. The dynamical system (D, f) is assumed to have the properties already mentioned in the beginning of this section and additionally $0 \notin D$.

It was already seen in the last subsection that $x_n = 0$ makes difficulties such that it cannot be proven that the algorithm eventually halts. However, the restriction $0 \notin D$ is no loss of generality. If all other conditions are fulfilled except that D contains zero, consider the following dynamical system $(\tilde{D}, \tilde{f})$ instead. Choose some $M > \min(D)$ and set $\tilde{D} := \{x + M \mid x \in D\}$ as well as $\tilde{f}(x) := f(x - M) + M$ for all $x \in \tilde{D}$. Then $(\tilde{D}, \tilde{f})$ fulfills all required properties. Furthermore $\tilde{f}'(x) = f'(x - M)$ holds and therefore there is no substantial difference in the complexity analysis of the algorithm between the original system and the modified system.

First, the boundedness of $\sigma(x)$ is shown.

Proposition 3.5. *Let (D, f) be as in Assumption 3.1 and $m_{min}(x_0, N, p)$ as in Definition 3.1. Then, for given $p \in \mathbb{Z}$, there exist some $C_1, C_2 \geq 0$, dependent of f, such that $m_{min}(x_0, N, p) \leq C_1 N + C_2$ holds for all $N \in \mathbb{N}$, $x \in \hat{\mathbb{R}} \cap D$.*

Proof. According to the requirements made on (D, f), there are some constants $L > 0$ and $M > 0$ such that $\overline{e}_{n+1} \leq L\overline{e}_n + \frac{K2^{-m}}{1-K2^{-m}}M$ holds for all $n \in \mathbb{N}$ and all mantissa lengths m. Without loss of generality assume $L \neq 1$, otherwise set $L > 1$. Analogous to the treatment in the proof of Proposition 3.4, iteration gives $\overline{e}_N \leq \frac{K2^{-m}}{1-K2^{-m}} M \sum_{n=0}^{N} L^n = \frac{K2^{-m}}{1-K2^{-m}} M \frac{L^{N+1}-1}{L-1}$. Since there exists some $B > 0$ with $B \leq |\hat{x}_n|$ for all n, $\overline{e}_N/|\hat{x}_N| \leq \overline{e}_N/B \leq C2^{-m}L^{N+1}$ follows with $C := MK/(B(1 - K2^{-m_0})(L - 1))$ where m_0 is the initial mantissa length, Line 2 in the algorithm. Then, if $C2^{-m}L^{N+1} \leq \frac{10^{-p}}{1+10^{-p}}$ holds, $prec((\hat{x}_n, \overline{e}_n), p) = \textbf{true}$ for all $n = 0, \ldots, N$. This leads to $m_{min}(x_0, N, p) \leq$
$max(0, ld(L))N + max(m_0, \mathrm{ld}(L) + \mathrm{ld}(C) + p \cdot \mathrm{ld}(10) + \mathrm{ld}(1 + 10^{-p}))$. □

Corollary 3.1. *Let (D, f) be as in Assumption 3.1 and $\sigma(x, p)$ the loss of significance rate. Then, for given $p \in \mathbb{Z}$, there exists some constant $C \geq 0$ such that $\sigma(x, p) \leq C$ holds for all $x \in \hat{\mathbb{R}} \cap D$.*

The treatment has now come to a stage that the main statements of this paper can be formulated. A lower and an upper bound for the loss of significance rate is given. Furthermore, these bounds are strongly related to the Ljapunow exponent $\lambda(x)$ defined in the previous section.

Theorem 3.1. *Let (D, f) be as in Assumption 3.1, $\sigma(x, p)$ the loss of significance rate and $\lambda(x)$ the Ljapunow exponent of (D, f). Then $\sigma(x, p) \geq \lambda(x)/ln(2)$ holds for all $x \in \hat{\mathbb{R}} \cap D$, $p \in \mathbb{Z}$ if $\lambda(x)$ exists.*

Proof. First there are two constants $B, M > 0$ such that $|\hat{x}_n| \geq B$ and $|\hat{x}_n| \leq M$ holds for all n. According to Equation 15 and Proposition 3.2, $\overline{e}_{n+1} \geq |f'(x_n)|\overline{e}_n$ holds. Iteration gives $\overline{e}_N \geq \frac{BK2^{-m}}{1-K2^{-m}} \prod_{n=0}^{N-1} |f'(x_n)|$. Hence, $\frac{\overline{e}_N}{|\hat{x}_N|} \geq \frac{BK2^{-m}}{M(1-K2^{-m})} \prod_{n=0}^{N-1} |f'(x_n)|$ follows. A necessary condition for the algorithm to terminate is therefore $\frac{BK}{M} 2^{-m} \prod_{n=0}^{N-1} |f'(x_n)| \leq \frac{10^{-p}}{1+10^{-p}}$ which gives $m_{min}(x_0, N, p) \geq \sum_{n=0}^{N-1} \mathrm{ld}(|f'(x_k)|) + p \cdot \mathrm{ld}(10) + \mathrm{ld}(\frac{BK}{M}) + \mathrm{ld}(1 + 10^{-p})$. Following the definitions of the loss of significance rate and the Ljapunow exponent, $\sigma(x_0, p) \geq \lambda(x_0)/\ln(2)$ follows. □

Before a realistic upper bound on the loss of significance rate can be presented, one more definition is needed.

Definition 3.2. Let $\alpha > 0$ then define a function $\eta_\alpha : (0, \infty) \to \mathbb{R}$ by

$$\eta_\alpha(x) := \begin{cases} \ln(x) & \text{if } x \geq \alpha \\ \ln(\alpha) & \text{if } x < \alpha \end{cases}.$$

Furthermore, for any $\alpha > 0$ define

$$\overline{\lambda}_\alpha(x) := \limsup_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} \eta_\alpha(|f'(f^k(x))|)$$

Proposition 3.6. *For all $\alpha > 0$ there exists some constant $C \geq 0$ such that $\overline{\lambda}_\alpha(x) \leq C$ holds for all $x \in D$. Furthermore, if the Ljapunow exponent $\lambda(x)$ exists, $\lambda(x) \leq \overline{\lambda}_\alpha(x)$ holds.*

Proof. Let L be a Lipschitz constant of f and $\alpha > 0$. Then for all $n \in \mathbb{N}$, $\frac{1}{n} \sum_{k=0}^{n-1} \eta_\alpha(|f'(f^k(x))|) \leq \ln(\max(\alpha, L))$ holds. Hence it follows the upper bound $\limsup_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \eta_\alpha(|f'(f^k(x))|) \leq \ln(\max(\alpha, L))$. The second assertion follows from the fact that $\ln(x) \leq \eta_\alpha(x)$ holds for all $x > 0$, $\alpha > 0$. □

Proposition 3.7. *Let $x \in D$ be given. If $\lambda(x)$ exists, then also the limit*

$$\lim_{\alpha \searrow 0} \overline{\lambda}_\alpha(x) =: \overline{\lambda}(x) \tag{17}$$

exists and $\overline{\lambda}(x) \geq \lambda(x)$.

Proof. Since $\ln(x) \leq \eta_\alpha(x) \leq \eta_\beta(x)$ holds for all $x > 0$, $0 < \alpha \leq \beta$, also $\lambda(x) \leq \overline{\lambda}_\alpha(x) \leq \overline{\lambda}_\beta(x)$ follows. So if α converges in a monotonic decreasing way to 0, $\alpha \searrow 0$, the assertion follows.

□

Theorem 3.2. *Let (D, f) be as in Assumption 3.1, $\sigma(x, p)$ the loss of significance rate and $\overline{\lambda}(x)$ as in (17). Let $x \in \hat{\mathbb{R}} \cap D$ be given, then for any $\varepsilon > 0$ there is some $p_0 \in \mathbb{Z}$ such that for all $p \geq p_0$, $\sigma(x, p) \leq \overline{\lambda}(x)/ln(2) + \varepsilon$ holds if $\overline{\lambda}(x)$ exists.*

Before the proof can be presented, the following lemma is needed.

Lemma 3.1. *Let $\varepsilon \geq 0$ and $\alpha > \sqrt{\varepsilon}$. Then for all $x > 0$,*

$$\ln(x + \varepsilon) \leq \eta_\alpha(x) + \sqrt{\varepsilon}$$

holds.

Proof. There is nothing to prove in the case $\varepsilon = 0$. So let $\varepsilon > 0$. The proof is split into two cases.

1. case: $x \geq \alpha$. Then the inequality reads $\ln(x + \varepsilon) \leq \ln(x) + \sqrt{\varepsilon}$ which is equivalent to $x \geq \frac{\varepsilon}{\exp(\sqrt{\varepsilon})-1}$. Since $\frac{\varepsilon}{\exp(\sqrt{\varepsilon})-1} \leq \frac{\varepsilon}{\sqrt{\varepsilon}} < \alpha \leq x$, the assertion follows.

2. case: $x < \alpha$. Then the inequality reads $\ln(x+\varepsilon) \leq \ln(\alpha)+\sqrt{\varepsilon}$ which is equivalent to $x \leq \alpha \exp(\sqrt{\varepsilon}) - \varepsilon$. A sufficient condition to proof the assertion is $\alpha \leq \alpha \exp(\sqrt{\varepsilon}) - \varepsilon$ which is equivalent to $\alpha \geq \frac{\varepsilon}{\exp(\sqrt{\varepsilon})-1}$. This was already proven in the first case. □

Now one is able to proof Theorem 3.2.

Proof of Theorem 3.2. Let $N \in \mathbb{N}$ be given and $0 < B \leq |\hat{x}_n| \leq M$ for all $n \in \mathbb{N}$. Starting with Equation 15 and iterating gives

$$\begin{aligned}
\overline{e}_N &= \overline{e}_0 \prod_{n=0}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n) + \frac{K2^{-m}}{1 - K2^{-m}} \sum_{k=1}^{N} \prod_{n=k}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n)|\hat{x}_k| \\
&= \frac{K2^{-m}}{1 - K2^{-m}} \sum_{k=0}^{N} \prod_{n=k}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n)|\hat{x}_k| \\
&\leq \frac{MK2^{-m}}{1 - K2^{-m_0}} \sum_{k=0}^{N} \prod_{n=k}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n) = C2^{-m} \sum_{k=0}^{N} \prod_{n=k}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n)
\end{aligned}$$

with $C := \frac{MK}{1-K2^{-m_0}}$. Define $K_N^{max} \in \{0,\ldots,N\}$ by $\prod_{n=K_N^{max}}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n) = \max\{\prod_{n=k}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n) : k = 0,\ldots,N\}$. Then,

$$\overline{e}_N \leq C2^{-m}(N+1) \prod_{n=K_N^{max}}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n)$$

follows. Consider the sequence $(K_N^{max})_{N\in\mathbb{N}}$. There are 2 cases.

1. case: $(K_N^{max})_{N\in\mathbb{N}}$ is not bounded. Observe that $(K_N^{max})_{N\in\mathbb{N}}$ is an increasing sequence and if $K_{N+1}^{max} > K_N^{max}$, then $K_{N+1}^{max} = N+1$. Therefore, there exists a constant $P > 0$ such that for all $N \in \mathbb{N}$ and $k = 0,\ldots,N$, $\prod_{n=k}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n) \leq P$ holds. Then, $\overline{e}_N \leq C2^{-m}(N+1)P$ follows. A sufficient condition for the algorithm to terminate is $\frac{CP}{B} \cdot 2^{-m}(N+1) \leq \frac{10^{-p}}{1+10^{-p}}$. This lead to the bound $m_{min}(x_0, N, p) \leq \mathrm{ld}(N+1) + p \cdot \mathrm{ld}(10) + \mathrm{ld}(1+10^{-p}) + \mathrm{ld}(CP/B)$ and hence $\sigma(x_0, p) = 0$ follows. On the other hand, $\sum_{n=0}^{N-1} \ln|f'(x_n)| \leq \sum_{n=0}^{N-1} \ln(\overline{L}(\hat{x}_n, \overline{e}_n)) \leq \ln(P)$ holds and therefore $\lambda(x_0) = 0$.

2. case: $(K_N^{max})_{N\in\mathbb{N}}$ is bounded. Then there exists some $N_0 \in \mathbb{N}$ such that $K_N^{max} = K_{N_0}^{max} =: K_0$ holds for all $N \geq N_0$. Then, for $N \geq N_0$, a sufficient condition for the algorithm to terminate is $\frac{C}{B}2^{-m}(N+1)\prod_{n=K_0}^{N-1} \overline{L}(\hat{x}_n, \overline{e}_n) \leq \frac{10^{-p}}{1+10^{-p}}$. Hence,

$$m_{min}(x_0, N, p) \leq C' + \mathrm{ld}(N+1) + \max(0, \sum_{n=K_0}^{N-1} \mathrm{ld}(\overline{L}(\hat{x}_n, \overline{e}_n)))$$

follows with $C' := p \cdot \mathrm{ld}(10) + \mathrm{ld}(1+10^{-p}) + \mathrm{ld}(C/B)$. Let L' be a Lipschitz constant of f', then there exists some $\overline{L}' \geq L'$ such that $\overline{L}(\hat{x}_n, \overline{e}_n) \leq |f'(x_n)| + \overline{L}' 2\overline{e}_n$ holds for all n. This inequality leads to $\overline{L}(\hat{x}_n, \overline{e}_n) \leq |f'(x_n)| + 2\overline{L}' M \frac{10^{-p}}{1+10^{-p}} \leq |f'(x_n)| + 2\overline{L}' M \cdot 10^{-p}$. Putting the terms together gives

$$\sigma(x_0, p) \leq \frac{1}{\ln(2)} \max(0, \limsup_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \ln(|f'(x_n)| + 2\overline{L}' M \cdot 10^{-p})).$$

Now let $\varepsilon > 0$ be given. Then choose α small enough such that $\overline{\lambda}_\alpha(x_0) \leq \overline{\lambda}(x_0) + \frac{\varepsilon}{2}$ holds. Next choose $p_0 \in \mathbb{N}$ with $\sqrt{2\overline{L}' M \cdot 10^{-p_0}} < \min(\alpha, \frac{\varepsilon}{2})$ holds. Then for all $p \geq p_0$, with the above lemma, $\sigma(x_0, p) \leq \frac{1}{\ln(2)} \max(0, \overline{\lambda}(x_0) + \frac{\varepsilon}{2} + \sqrt{2\overline{L}' M \cdot 10^{-p_0}}) \leq \frac{1}{\ln(2)} \max(0, \overline{\lambda}(x_0)) + \varepsilon$ follows. □

4 Conclusions

In this paper, two main issues are addressed. First it is shown that a mathematically precise treatment of multiple-precision floating point computability including automated error analysis is far away from being a difficult and confusing task. Also, this treatment is in a manner which is familiar to people working in the field of numerical analysis or scientific computing and also for theoretical computer scientists. Furthermore, the formalism does not only allow exact answers concerning the existence of a computationally feasible algorithm, but is also allows a treatment of its complexity. As a consequence, the described algorithm is formulated not only in an exact and guaranteed way, but also enables a motivated reader the real implementation and gives a practical performance analysis.

Second, the results show that the Ljapunow exponent, a central quantity in dynamical systems theory, also finds its way into complexity theory, a branch in theoretical computer science. In dynamical systems theory, the Ljapunow exponent describes the rate of divergence of initially infinitesimal nearby points. For two points having a small but finite initial separation, the Ljapunow exponent has only relevance for short time scales [4]. The reason is that due to the boundedness of D, any two different orbits cannot separate arbitrarily far away. However, the loss of significance rate shows that the Ljapunow exponent has on long time scales not only an asymptotic significance but also a concrete practical one.

Acknowledgments

The author wishes to express his gratitude to Peter Hertling for helpful discussions and comments.

References

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.

[2] V. Brattka and P. Hertling. Feasible real random access machines. *Journal of Complexity*, 14(4):490–526, 1998.

[3] P. Collet and J.-P. Eckmann. *Iterated Maps on the Interval as Dynamical Systems*. Progress in Physics. Birkhäuser, Boston, Massachusetts, 1980.

[4] P. Collet and J.-P. Eckmann. *Concepts and Results in Chaotic Dynamics*. Theoretical and Mathematical Physics. Springer-Verlag, Berlin, Heidelberg, 2006.

[5] R. L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, Redwood City, California, 2nd edition, 1989.

[6] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software*, 33(2):13:1–13:15, June 2007.

[7] M. W. Hirsch, S. Smale, and R. L. Devaney. *Differential Equations, Dynamical Systems and an Introduction to Chaos*. Elsevier Academic Press, Amsterdam, 2004.

[8] IEEE 1987. *IEEE Standard 754-1985 for Binary Floating-Point Arithmetic*. IEEE, New York, 1987.

[9] A. Katok and B. Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press, Cambridge New York Melbourne, 1995.

[10] N. T. Müller. The iRRAM: Exact arithmetic in C++. In J. Blanck, V. Brattka, and P. Hertling, editors, *Computability and Complexity in Analysis*, volume 2064 of *Lecture Notes in Computer Science*, pages 222–252, Berlin, 2001. Springer. 4th International Workshop, CCA 2000, Swansea, UK, September 2000.

[11] H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions*. Ellis Horwood Limited, Chichester, 1984.

[12] N. Revol and F. Rouillier. Motivations for an Arbitrary Precision Interval Arithmetic and the MPFI Library. *Reliable Computing*, 11(4):275–290, 2005.

[13] K. Weihrauch. *Computable Analysis*. Springer-Verlag, Berlin Heidelberg New York, 2000.

[14] J. H. Wilkinson. *Rounding Errors in Algebraic Processes*. Prentice-Hall, Englewood Cliffs, N.J., 1963.