Computational Complexity of Iterated Maps on the Interval

Christoph Spandl

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

Abstract

The correct computation of orbits of discrete dynamical systems on the interval is considered. Therefore, an arbitrary-precision floating-point approach based on automatic error analysis is chosen and a general algorithm is presented. The correctness of the algorithm is shown and the computational complexity is analyzed. There are two main results. First, the computational complexity measure considered here is related to the Lyapunov exponent of the dynamical system under consideration. Second, the presented algorithm is optimal with regard to that complexity measure.

Keywords: Discrete dynamical systems, Lyapunov exponent,
arbitrary-precision floating-point arithmetic
2000 MSC: 37M05, 65P20

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 [14, 9] are used. However, if the dynamical

---

system shows chaotic behavior, a problem arises. The finite and constant length of the significand of a `double` variable causes rounding errors which are magnified after each iteration step. Already after a few iterations, the error is so big that the computed values are actually useless. For example in [17, 21] this phenomenon is examined for the dynamical system (D, f) with $D = [0, 1]$, $f(x) = 3.75 \cdot x \cdot (1 - x)$ and the initial value $x_0 = 0.5$. To put things right, a rigorous method for computations with real numbers has to be used. There already exist some rigorous numerical methods in the field of dynamical systems and chaos [11, 10, 30, 23, 25, 20].

In the next section, a rigorous method based on arbitrary-precision floating point arithmetic is presented and used to investigate the iteration of a generalization of the above mentioned function. Correctness of the results are obtained by using a method called running error analysis. The method and the numerics are compared to interval arithmetic. In Section 3, the algorithm is generalized to arbitrary functions f. The aim of the present paper is to give bounds on some kind of space complexity of the algorithm. To be more precise, the behavior of the the length of the significand in arbitrary-precision arithmetic is analyzed in the task of iterates of discrete dynamical systems. The minimal length of the significand needed for floating-point numbers such that any computed point of an initial segment of the orbit has a specified and guaranteed accuracy is examined. This minimal length will be related to the length of the initial segment of the orbit. To cope with this task, a precise mathematical framework for floating-point computations is applied. This framework should be suited to computability concepts over the reals. Finally, a complexity measure for describing the computational effort on arbitrary-precision floating-point numbers is introduced. Roughly speaking, it is the ratio of the length of the significand to the number of iterations in the limit of number of iterations to infinity. The first main result shows that this complexity measure is related to the Lyapunov exponent. The second main result proves that the presented algorithm to compute the orbit up to any given accuracy is optimal with respect to that complexity measure. As a consequence, these results give some advice for economically designing reliable algorithms simulating one-dimensional discrete dynamical systems.

2

2. Dynamic behavior of the logistic equation and rounding error

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* [5]. When implementing the logistic equation on a real computer and demanding to obtain true values for the orbit $(x_n)_n$, some rigorous method is needed. Since for some values of μ the dynamics is highly chaotic, inaccuracies are magnified exponentially in time [6, 13]. Therefore, it is clear that using floating-point numbers with a predefined, fixed precision makes sense only if the maximum iteration time N also is a predefined, fixed number. If the algorithm should work for any N, a more elaborate approach is needed. First one can work with arbitrarily high precision floating-point numbers, the precision dynamically set and the error control implemented in the algorithm separately. A software package for doing this task is for example MPFR [8]. Second, there are methods with automatic error control, for example interval arithmetic [19, 1], the Feasible Real RAM model [4] or significance arithmetic [18]. Implementations are for example MPFI [28], the iRRAM [21] and Mathematica [31] respectively.

All these methods have the same theoretical background. They are all practical instances of the model of Computable Analysis [34, 24, 16] used in computer science. While the Feasible Real RAM model directly implements the theory of Computable Analysis, the other mentioned methods all have their background in scientific computing. Looking closer at the various validated methods in use, they all have in common implementing some kind of intervals for representing real numbers numerically. Therefore, the starting point here is looking at interval arithmetic for computing orbits $(x_n)_n$. For any time step n, let the phase point x_n together with its computational error be represented by two floating-point numbers x_n^l and x_n^u ($x_n^u \geq x_n^l$) with given length m_n of the significand, called the precision, forming an interval $[x_n^l, x_n^u]$. The interval is an enclosure of the real value x_n, that is $x_n \in [x_n^l, x_n^u]$ for all n. The interval length $d_n := x_n^u - x_n^l$ gives a measure of the uncertainty about x_n and is therefore a kind of error. Interval arithmetic often models quantities which are not known exactly. But here, the true orbit $(x_n)_n$ can be, in principle, calculated to any given accuracy. Thus, in the present setting, the true object of interest is not an interval, but an approximation $\hat{x}_n$ of x_n together with an absolute error e_n. The interval is only used for mathematical convenience. To transform the interval to a floating point value $\hat{x}_n$

3

of precision m_n, just do

$$\hat{x}_n := rd\left(\frac{x_n^l + x_n^u}{2}, m_n\right) \tag{1}$$

where $rd(x, m)$ performs a rounding to some floating-point number of precision m nearest to x. Note that rounding to nearest is not unique if x is equidistant from two floating-point numbers. The absolute error $e_n := |\hat{x}_n - x_n|$ of $\hat{x}_n$ can be estimated via the interval length d_n by

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

where r_n is an error caused by the rounding operation $rd(.,.)$ in Equation (1). 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.

The aim now is to calculate, for given initial value $x = x_0$, $N \in \mathbb{N}$ and $p \in \mathbb{Z}$ the orbit up to time N with relative error at most 10^{-p}. That is, for $(\hat{x}_n)_{0 \leq n \leq N}$ it should hold

$$e_n = |\hat{x}_n - x_n| \leq 10^{-p}x_n \leq 10^{-p}. \tag{3}$$

Why using here and in the following the relative error and not the absolute error is discussed in some detail at the end of Subsection 3.2. The minimal m, fulfilling the precision requirement (3) on the relative error of x_n, which depends on x, N and p, is denoted by $m_{min}(x, N, p)$. Now, a central quantity of this work is introduced, which is some complexity measure. Consider the growth rate of $m_{min}(x, N, p)$,

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

The *loss of significance rate* $\sigma(x)$, which may depend on the initial value x is given by

$$\sigma(x) = \lim_{p \to \infty} \sigma(x, p).$$

This quantity describes the limiting amount of significant precision being lost at each iteration step in the limit of infinite output precision. Significant means here the part of the digits being correct. A general treatment of this complexity measure is given in the next section. Roughly speaking, $\lceil \sigma(x_0, p)N + p \cdot ld(10) \rceil$ is the precision for any floating-point number needed in an algorithm doing the iteration starting with x_0 and calculating to x_N, if the output should be precise to at least p decimal places. Here, $ld(.)$ is the logarithm to base 2.

4

2.1. Dynamic behavior of the logistic equation

Before analyzing the different numerical behavior, it is worth having an analytical look at the dynamical behavior of the system. Despite the fact that these results are well known [13, 7], they are reviewed here for the sake of self containment.

First have a look at the fixed points of the logistic equation and their stability. In the range $D = [0, 1]$, the equation possesses exactly one fixed point $x^o = 0$ if $\mu \in (0, 1]$ and exactly two fixed points $x^o = 0$ and $x^{(\mu)} = 1 - \frac{1}{\mu}$ if $\mu \in (1, 4]$. Looking at the derivatives $f'_\mu(x^o) = \mu$ and $f'_\mu(x^{(\mu)}) = 2 - \mu$ gives the stability of the fixed points. Since $|f'_\mu(x^o)| < 1$ for $\mu \in (0, 1)$ and $|f'_\mu(x^o)| > 1$ for $\mu \in (1, 4]$, x^o is a stable fixed point, an attractor for $\mu \in (0, 1)$ and an unstable fixed point, a repeller for $\mu \in (1, 4]$. If $\mu = 1$, the only fixed point x^o is hyperbolic, that is $|f'_1(x^o)| = 1$. At $\mu = 1$, a bifurcation occurs. If $\mu \in (1, 3)$, x^o becomes unstable and the newly occurring fixed point $x^{(\mu)}$ is stable. At $\mu = 3$ a second bifurcation occurs and for $\mu > 3$ both fixed points are unstable.

Second, examine the basin of attraction of the stable fixed point. If $\mu \in (0, 1)$, the contraction mapping principle directly gives $\lim_{n\to\infty} f^n_\mu(x) = x^o$ for all $x \in [0, 1]$. 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, also $\lim_{n\to\infty} f^n_1(x) = x^o$ holds for all $x \in [0, 1]$. Finally, in the case $\mu \in (1, 3)$, $\lim_{n\to\infty} f^n_\mu(x) = x^{(\mu)}$ holds for all $x \in [0, 1]$. For a proof, the interested reader is referred to the literature:[7], Proposition 5.3 in Section 1.5.

Finally, for $\mu > 3$ the system goes into a region showing 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 tends to the stable fixed point for any initial value $x \in [0, 1]$. 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. Next have a look at the computational effort in the various control parameter ranges.

2.2. Numerical analysis of the computational complexity

The logistic equation is implemented in various forms using an arbitrary-precision interval library. For that purpose, the already mentioned interval

library MPFI based on the arbitrary-precision floating-point number library MPFR, both written in C, is used. For each control parameter μ ranging from 0.005 to 4 and a step size of 0.005, the orbit for initial value $x = 0.22$ is calculated up to $N = 2000$. For each μ, the minimum precision m_{min} needed to guarantee $e_n \leq 10^{-6} x_n$ for $n = 0, \ldots, N$ is searched. Then, $\sigma_{est} := m_{min}/N$ is calculated. First, f_μ is implemented using a natural interval extension based on the expressions $\mu x(1-x)$, $\mu(x-x^2)$ and $\frac{\mu}{4} - \mu(x - \frac{1}{2})^2$. The natural interval extension is obtained by replacing any occurrence of the variable x in the expression by an interval $\boldsymbol{x}$ [26]. The results are shown in Figures 1, 2 and 3 respectively. Second, the logistic equation is implemented using a centered form, actually the mean value form [26, 17]: $T_{1,\mu}(\boldsymbol{x}) = f_\mu(\mathrm{mid}(\boldsymbol{x})) + \boldsymbol{f}'_\mu(\boldsymbol{x})(\boldsymbol{x} - \mathrm{mid}(\boldsymbol{x}))$ where $\boldsymbol{x}$ is an interval and $\mathrm{mid}(\boldsymbol{x})$ is the midpoint of $\boldsymbol{x}$. The result is shown in Figure 4. In the following, these 4 calculations are referred to as 1 to 4 respectively.

The interval computation is in agreement with the dynamical picture only in Calculation 4. While for $\mu \in (0, 1)$, the results shown in Calculations 1, 2 and 4 are in agreement with the dynamical analysis, 3 is not since it would suggest an exponential divergence of initially nearby orbits which is not true in reality. A similar situation occurs for $\mu \in (1, 3)$. Here, the Calculations 3 and 4 are in agreement with the dynamical picture, 1 and 2 on the other hand not. The picture does not change if $\mu \geq 3$ and hence it can be said that in the range $\mu \in (1, 4]$, the Calculations 3 and 4 are in agreement with the dynamic picture, while 1 and 2 are not. How can this be explained?

2.3. Investigating Calculation 1

This subsection deals with the explanation of the curve obtained by Calculation 1. For doing an error analysis of the logistic equation analytically, some idealizing assumptions have to be made. Generally, executing the iteration in interval arithmetic, two types of error are present. First, error propagation solely due to the iteration and second the newly added rounding error caused by the calculation of f_μ. In the following, only the error propagation is regarded. This means that there is only one primary made error caused by rounding the initial value $x = x_0$ to some floating-point number of some specified precision m. The next idealization is that the value of μ is assumed to be given with such a high precision that no interval representation is needed. Finally, the value of r_n in Equation (2) is neglected. The

recursion relation then reads

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

with the interval length d_n given by the recursion relation

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

with the obvious solution $d_n = \mu^n d_0$. The absolute error e_n of $\hat{x}_n$ according to Equation (1) can be bounded from above by

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

The ideal assumptions require the somewhat unreal setting that the precision 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. To get a relation connecting m and the output precision p in (3), some upper bound on d_0 is needed. The value of d_0 is given as the rounding error by representing x_0 as a floating-point number of precision m. For that, the well known estimate

$$d_0 \leq 2^{-m+1} x_0 \leq 2^{-m+1} \tag{5}$$

exists. Combining (3), (4) and (5) gives as a sufficient condition

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

for $n = 0, \ldots, N$. So, the sufficient condition gives an upper bound on $m_{min}(x, N, p)$ by

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

This finally leads to an upper bound for the loss of significance rate,

$$\sigma(x, p) \leq \sigma(x) \leq \max(0, \mathrm{ld}(\mu)).$$

The curve in Figure 1 shows that σ_{est} exceeds the estimated bound $\max(0, \mathrm{ld}(\mu))$ only slightly. So, the above made ideal assumptions seem to be valid. In [21], the logistic equation was also investigated for $\mu = 3.75$

7

using the exact real arithmetic package iRRAM. In the paper, the maximum bounding precision needed to guarantee the correctness of the first 6 decimal places are reported up to $N = 100000$. Relating this quantity to m_{min} shows full agreement with the simulation results performed here. So, for $\mu > 1$, the interval length d_n increases exponentially in time n which is in contrast to the dynamic behavior for $\mu \in (1,3)$. The reason is that the natural interval approach implicitly, due to the dependency problem, 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.
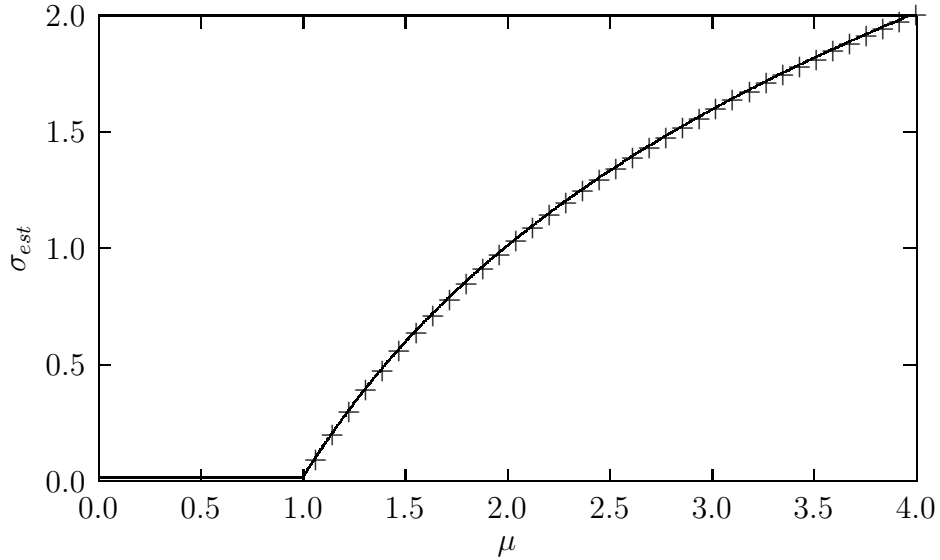

Figure 1: Estimated loss of significance rate for the logistic equation, formula $\mu x(1 - x)$. The crosses indicate the theoretical curve from error analysis.

2.4. Investigating Calculation 2

Calculation 2 is similar to Calculation 1. An analogous analytic approach as in Calculation 1 gives the recursion relation

$$x_{n+1}^l = \mu(x_n^l - (x_n^u)^2)$$
$$x_{n+1}^u = \mu(x_n^u - (x_n^l)^2)$$

and hence

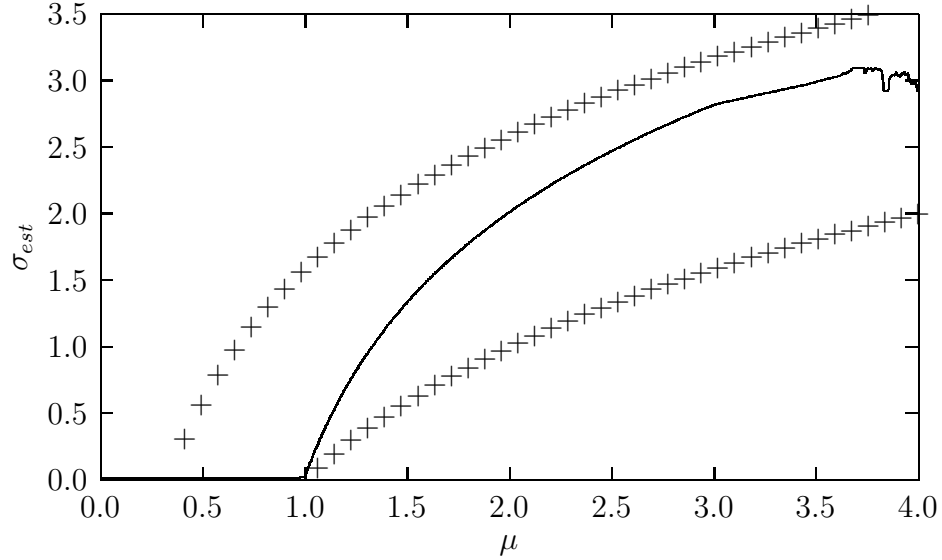$$d_{n+1} = \mu d_n + \mu((x_n^u)^2 - (x_n^l)^2) = \mu d_n(1 + x_n^u + x_n^l).$$

Figure 2: Estimated loss of significance rate for the logistic equation, formula $\mu(x - x^2)$. The crosses indicate the theoretical curves from error analysis.

The recursion relation fulfills therefore $d_{n+1} \geq \mu d_n$ and $d_{n+1} \leq 3\mu d_n$. As a consequence the bounds $\mu^n d_0 \leq d_n \leq (3\mu)^n d_0$ are obtained. In analogy to Calculation 1, an upper bound for $m_{min}(x, N, p)$,

$$m_{min}(x, N, p) \leq \lceil p \cdot \mathrm{ld}(10) + N \cdot \max(0, \mathrm{ld}(3\mu)) \rceil$$

9

and hence
$$\sigma(x, p) \leq \sigma(x) \leq \max(0, \mathrm{ld}(3\mu))$$
is calculated. A brief look at Figure 2 shows that this upper bound is too rough. On the other hand, $d_n \geq \mu d_0$ suggests that the bound derived in Calculation 1 is a lower bound, hence $\max(0, \mathrm{ld}(\mu)) \leq \sigma(x) \leq \max(0, \mathrm{ld}(3\mu))$. This is actually verified by numerical evidence.

2.5. Investigating Calculation 3

Calculation 3 is explained here in the parameter range $\mu \in (0, 1)$, where it is not in agreement with the dynamic picture. Nevertheless it should be mentioned that the natural interval extension used here seems to be in full agreement with the dynamic picture in the parameter range $\mu \in [1, 4]$ as is suggested by Figure 3. The curve seems to be identical to Figure 4 in the range $\mu \in [1, 4]$.
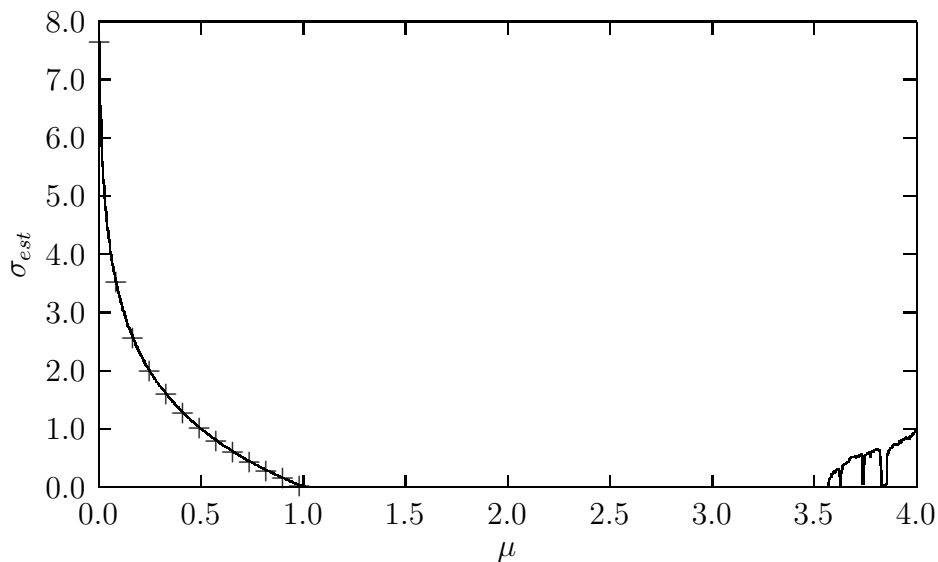

Figure 3: Estimated loss of significance rate for the logistic equation, formula $\frac{\mu}{4} - \mu(x - \frac{1}{2})^2$. The crosses indicate the theoretical curve from error analysis.

To explain the observed behavior, first note that for $\mu < 1$, $f_\mu(x) \leq \mu x$ follows for all $x \in [0, 1]$. Hence, $x_n \leq \mu^n x_0$ holds for all $n \in \mathbb{N}$ and the

10

orbit $(x_n)_n$ tends exponentially fast to zero. A brief look at the expression of $f_\mu(x)$, $f_\mu(x_n) = \frac{\mu}{4} - \mu(x_n - \frac{1}{2})^2$, shows that the value of the second term in the difference tends exponentially fast in n to the value of the first term. Hence, the big values of the loss of significance rate for small values of μ can be explained by cancellation. So, the behavior may be explained solely by a typical phenomenon of floating-point arithmetic and not an effect due to the dependency problem in interval arithmetic. To give an analytical description of the problem, it is easier now to chance from interval notation to classical error analysis notation.

First note that even $\hat{x}_0$ may differ from the initial value x since the conversion to a floating-point number may cause the very first rounding error. Next, already mentioned, in calculating the orbit $(\hat{x}_n)_n$, two types of error are present. First, error propagation due to the iteration scheme and second the rounding 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 $\hat{x}_{n+1} - x_{n+1} = \hat{f}_\mu(\hat{x}_n) - f_\mu(x_n)$. Inserting a constructive zero gives a sum

$$\hat{x}_{n+1} - x_{n+1} = (f_\mu(\hat{x}_n) - f_\mu(x_n)) + (\hat{f}_\mu(\hat{x}_n) - f_\mu(\hat{x}_n)) \tag{6}$$

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.

Let us fix some $n \in \mathbb{N}$ and consider the absolute error of $\hat{x}_{n+1}$. To get the formulas more compact, set $g_\mu(x) := \mu(x - \frac{1}{2})^2$. Then,

$$\begin{aligned}
|\hat{x}_{n+1} - x_{n+1}| &= |(\widehat{(\frac{\mu}{4})} - \widehat{\hat{g}_\mu(\hat{x}_n)}) - (\frac{\mu}{4} - g_\mu(x_n))| \\
&\leq |(\widehat{(\frac{\mu}{4})} - \widehat{\hat{g}_\mu(\hat{x}_n)}) - ((\widehat{\frac{\mu}{4}}) - \hat{g}_\mu(\hat{x}_n))| + |(\widehat{\frac{\mu}{4}}) - \frac{\mu}{4}| \\
&\quad + |\hat{g}_\mu(\hat{x}_n) - g_\mu(x_n)| \\
&\leq |(\widehat{(\frac{\mu}{4})} - \widehat{\hat{g}_\mu(\hat{x}_n)}) - ((\widehat{\frac{\mu}{4}}) - \hat{g}_\mu(\hat{x}_n))| + |(\widehat{\frac{\mu}{4}}) - \frac{\mu}{4}| \\
&\quad + |\hat{g}_\mu(\hat{x}_n) - g_\mu(\hat{x}_n)| + |g_\mu(\hat{x}_n) - g_\mu(x_n)|
\end{aligned}$$

follows. Let m be assumed to be the actual precision under calculation at time n. The last term in the previous inequality can be estimated the following way. As discussed in [35], the rounding error produced in calculating

11

g_μ can be estimated by

$$|\hat{g}_\mu(\hat{x}_n) - g_\mu(\hat{x}_n)| \leq 1.06 K 2^{-m} |g_\mu(\hat{x}_n)| \tag{7}$$

where K is the number of rounding operations performed in computing $\hat{g}_\mu$. In the case considered here, $K = 4$ follows. 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 precision must not be chosen too small. Furthermore, with $|g_\mu(\hat{x}_n) - g_\mu(x_n)| \leq \mu |\hat{x}_n - x_n|$ it follows

$$
\begin{aligned}
|\hat{x}_{n+1} - x_{n+1}| &\leq 2^{-m} |\widehat{(\tfrac{\mu}{4})} - \hat{g}_\mu(\hat{x}_n)| + 2^{-m} \cdot \tfrac{\mu}{4} + 1.06 K 2^{-m} |g_\mu(\hat{x}_n)| \\
&\quad + \mu |\hat{x}_n - x_n| \\
&\leq 2^{-m}(|\widehat{(\tfrac{\mu}{4})}| + |\hat{g}_\mu(\hat{x}_n)| + \tfrac{\mu}{4} + 1.06 K |g_\mu(\hat{x}_n)|) + \mu |\hat{x}_n - x_n| \\
&\leq 2^{-m}((1 + 2^{-m})\tfrac{\mu}{4} + (1 + 1.06 K 2^{-m}) |g_\mu(\hat{x}_n)| + \tfrac{\mu}{4} \\
&\quad + 1.06 K |g_\mu(\hat{x}_n)|) + \mu |\hat{x}_n - x_n| \\
&\leq 2^{-m} \tfrac{\mu}{4}(1 + 2^{-m} + 1 + 1.06 K 2^{-m} + 1 + 1.06 K) \\
&\quad + \mu |\hat{x}_n - x_n| \\
&\leq C\mu 2^{-m} + \mu |\hat{x}_n - x_n|
\end{aligned}
$$

where $C > 0$ holds. In other words, one obtains the recursion relation $e_{n+1} \leq \mu e_n + C\mu 2^{-m}$. Iterating the recursion gives $e_{n+1} \leq C\mu 2^{-m} \sum_{k=0}^n \mu^k + \mu^{n+1} e_0 \leq C\frac{\mu}{1-\mu} 2^{-m} + \mu^{n+1} 2^{-m} x_0$.

As already mentioned, x_n is bounded from above by $x_n \leq \mu^n x_0$. To come to a sufficient condition for the precision, also a lower bound is needed. First observe that $f_\mu(x) \geq \mu x(1 - a)$ holds for all $x \leq a$, $a, x \in [0, 1]$. Hence, for $\mu < 1$, $x_{n+k} \geq \mu^n x_k (1 - x_k)^n$ follows. This gives the sufficient condition

$$C\frac{\mu}{1-\mu} 2^{-m} + \mu^{n+1} 2^{-m} x_0 \leq 10^{-p} \mu^{n+1-k} x_k (1 - x_k)^{n+1-k} \leq 10^{-p} x_{n+1}$$

on the precision. Note that $n + 1 \geq k \geq 0$. Then, an upper bound on $m_{min}(x_0, N, p)$ is given by

$$m_{min}(x, N, p) \leq \lceil p \cdot \mathrm{ld}(10) + (N - k)(\mathrm{ld}(\tfrac{1}{\mu}) - \mathrm{ld}(1 - x_k)) + C' \rceil$$

with $C' = \mathrm{ld}(C\frac{\mu}{1-\mu} + \mu^N x_0) - \mathrm{ld}(x_k)$. This leads to an upper bound on the loss of significance rate given by $\sigma(x, p) \leq \mathrm{ld}(\tfrac{1}{\mu}) - \mathrm{ld}(1 - x_k)$ for all $k \in \mathbb{N}$. Since $x_k \to 0$ follows for $k \to \infty$, the final result on the loss of significance rate is

$$\sigma(x, p) \leq \sigma(x) \leq \mathrm{ld}(\tfrac{1}{\mu}).$$

The curve in Figure 2 shows that this upper bound is in full agreement with the numeric result.

12

2.6. Investigating Calculation 4

The observation at the end of the subsection describing Calculation 1 directly leads to the already introduced mean value form. The calculation is shown in Figure 4. This calculation is the optimum of both, Calculation 1 and 3. The curve reflects in the parameter range $\mu \in (0, 3)$ well the dynamic behavior.
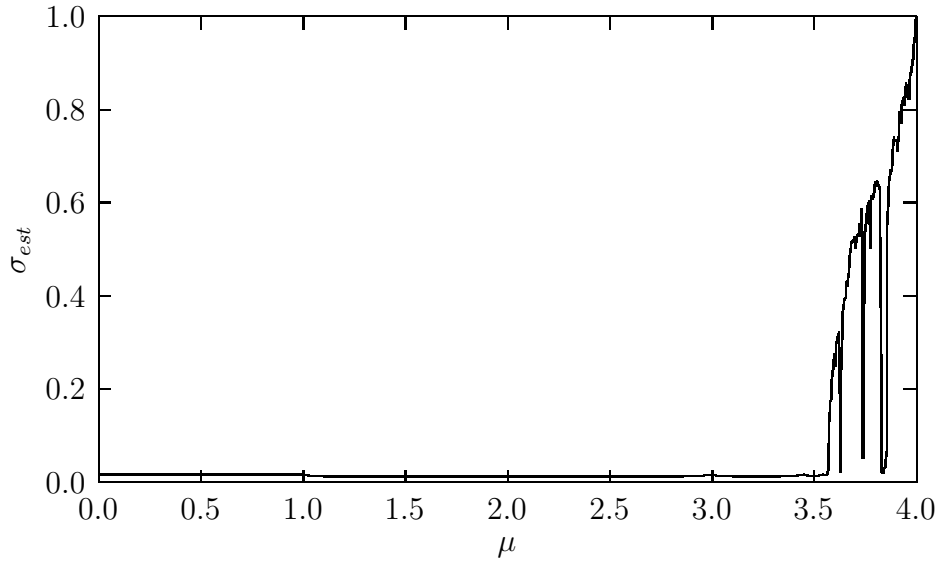

Figure 4: Estimated loss of significance rate for the logistic equation, meanvalue form.

Furthermore, in the range $\mu \in [3, 4]$, the curve suggests a relation between the loss of significance rate and the Lyapunov exponent $\lambda(x)$ for the logistic map:

$$\sigma(x) = \frac{1}{\ln(2)} \max(0, \lambda(x))$$

for all $\mu \in (0, 4]$. For a curve of the Lyapunov exponent of the logistic map see [5]. This relation will be shown in the next section for general dynamical systems on the interval. Furthermore, it will be shown that the algorithm based on Calculation 4 is optimal in some sense.

But before, some crucial reflections governing the analysis in the next section. The mean value form representation, on which the calculation is based, can also be seen from a different viewpoint. Have again a look at Equation (6). The true error is the sum of the error propagation (first term) according to the iteration and the rounding error due to the computation of f_μ (second term). The first term of Equation (6) 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 in a similar way as was done in (7) by

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

where $K = 4$ because there are 3 arithmetic operations and the rounding of μ. Using the fact that $f_\mu(x) \leq \frac{\mu}{4}$ holds and $f_\mu(x) < x$ if $\mu \leq 1$, the unknown value $|f_\mu(\hat{x}_n)|$ can be estimated from above. This calculation shows that there exists a recursive equation on an upper bound $\overline{e}_n$ on e_n for all n:

$$\overline{e}_{n+1} = L(\hat{x}_n, \overline{e}_n)\overline{e}_n + 1.06K2^{-m}E_\mu(\hat{x}_n), \quad \overline{e}_0 = 2^{-m} \tag{8}$$

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

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

This description, which is in line with the analysis of Calculation 3, is equivalent to the interval description using the mean value form. Instead of using intervals, pairs of the form value $\hat{x}_n$ and corresponding guaranteed error bound $\overline{e}_n$ is used. This approach is an automated error analysis called *running error analysis* [12]. From a technical point of view, the representation as value and error has the advantage that the rounded values $\hat{x}_n$ are calculated as usual in floating-point arithmetic except that arbitrary-precision floats are used. The guaranteed error bounds may be calculated using interval arithmetic according to (8), to really guarantee a validated bound. Only a fixed precision is needed for calculating the error bounds. Similar results as in Figure 4 are reported in [2] by using a method analog to the one presented here [3]. However, the connection to the Lyapunov exponent is not made in [2].

Before continuing, three remarks. First, interval libraries are primarily divided int two types concerning their representation of an interval [29]: There exist libraries using the infimum-supremum representation of intervals, like MPFI, and there exist libraries using the midpoint-radius representation of intervals. If arbitrary precision is needed, the inf-sup libraries have the disadvantage that two floating-point variables with high precision are needed to represent an interval. Contrary to that, like the value and error description, in mid-rad libraries only the midpoint of the interval needs a high precision floating-point variable. The radius can be stored in a floating-point variable which need not have a high precision. Clearly, the mid-rad concept has an computational advantage in the case considered here over the inf-sup concept. But the dependency problem of interval arithmetic persists. Second, also the iRRAM package implements mid-rad intervals and has therefore to cope with the dependency problem. However, it also permits an optimized way for computing the iteration based on a similar algorithm as described above [22]. Third it should be mentioned that, executing the first three presented calculations in Mathemathica using significance arithmetic, exactly the same results are obtained. This shows that also significance arithmetic suffers from the dependency problem as interval arithmetic does. This is already noted in [32].

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, two times continuously differentiable on D and f'' is bounded. Furthermore, f and f' are assumed to be computable in the sense of Computable Analysis. The definition of a computable real function is given below.

In this section, a general algorithm for computing the iteration

$$x_{n+1} = f(x_n), \quad x_0 \in D \tag{9}$$

is presented. To be more precise, for given $x \in \mathbb{Q}$, $N \in \mathbb{N}$ and $p \in \mathbb{Z}$, this algorithm computes a finite part $(\hat{x}_n)_{0 \leq n \leq N}$ of length N of the true orbit $(x_n)_{n \in \mathbb{N}}$ with initial value $x_0 = x$. Each computed value $\hat{x}_n$ of this finite trajectory has a relative error of at most 10^{-p}: $|\hat{x}_n - x_n| \leq 10^{-p}|x_n|$ for all $n = 0, 1, \ldots, N$. The correctness of the algorithm and its relation to Computable Analysis is shown. Finally, its complexity is examined.

3.1. Computability issues and specifying the algorithm

The set of all computationally accessible real numbers are the floating-point numbers of arbitrary precision and arbitrary exponent range denoted by $\hat{\mathbb{R}}$. A floating-point number is a real number of the form $\hat{x} = s \cdot 2^{e-t}$ where $t \in \mathbb{N}$ is the *precision*, $e \in \mathbb{Z}$ the *scale* and $s \in \mathbb{Z}$ where $|s| \in \{0, 1, \ldots, 2^t - 1\}$ is called the *significand*. To get a unique representation of $\hat{x}$ for given t, $|s| \geq 2^{t-1}$ is assumed if $\hat{x} \neq 0$ and $e = 0$ if $\hat{x} = 0$. Since actually no bound is assumed on the precision and the scale, the set $\hat{\mathbb{R}} \subseteq \mathbb{R}$ is the set of the dyadic real numbers and therefore countable infinite. Thus, $\hat{\mathbb{R}}$ forms a natural basis for computability considerations over finite objects. Consider some floating-point number $\hat{x} \in \hat{\mathbb{R}}$, then the scale and the precision are two properties of different type. While the scale is a direct function of the value of $\hat{x}$, the precision is clearly not. Reversely, let $x \in \mathbb{R}$ be some real number and $\hat{x} \in \hat{\mathbb{R}}$ a floating-point number representing x. Then the scale of $\hat{x}$ is generally determined by x while the precision can be chosen arbitrary. Regarding $\hat{x}$ as a data structure, then $\hat{x}$ has as its essential property the precision. In object oriented notation, the precision of $\hat{x}$ can be written as $\hat{x}.t$.

Any real number x is represented in an algorithm concerning numerical computation by a pair $\boldsymbol{x}$ consisting of a floating point number $\boldsymbol{x}.fl \in \hat{\mathbb{R}}$ of arbitrary precision $\boldsymbol{x}.fl.t$ approximating x and a floating-point number $\boldsymbol{x}.err \in \hat{\mathbb{R}}$ of fixed precision giving an upper bound on the absolute error, $|\boldsymbol{x}.fl - x| \leq \boldsymbol{x}.err$. Reversely, any such pair $\boldsymbol{x} \in \hat{\mathbb{R}}^2$ can be seen as the real interval $[\boldsymbol{x}.fl - \boldsymbol{x}.err, \boldsymbol{x}.fl + \boldsymbol{x}.err]$. If $x \in [\boldsymbol{x}.fl - \boldsymbol{x}.err, \boldsymbol{x}.fl + \boldsymbol{x}.err]$ holds for some $x \in \mathbb{R}$, then $\boldsymbol{x}$ is called an *approximation* of x. To represent a single real number, a sequence $(\boldsymbol{x}_n)_{n \in \mathbb{N}}$ of such pairs $\boldsymbol{x}$ are needed. A sequence $(\boldsymbol{x}_n)_{n \in \mathbb{N}}$ is called a *floating-point name* of a real number x, if any $\boldsymbol{x}_n$ approximates x, $\lim_{n \to \infty} \boldsymbol{x}_n.fl = x$, $\lim_{n \to \infty} \boldsymbol{x}_n.fl.t = \infty$ and $\lim_{n \to \infty} \boldsymbol{x}_n.err = 0$ holds. Clearly any real number has a floating-point name.

As already indicated, it is a straightforward task to define what a computable function $\hat{f} : \hat{\mathbb{R}} \to \hat{\mathbb{R}}$ is by using classical computability theory over finite objects. Additionally, computability over integers, computability of functions with mixed arguments and computable predicates are defined in the same manner [33]. Consider a function $f : D \to D$, $D \subseteq \mathbb{R}$ and a pair $\boldsymbol{f}$ of two functions $\boldsymbol{f}.fl : \hat{\mathbb{R}} \to \hat{\mathbb{R}}$ and $\boldsymbol{f}.erf : \hat{\mathbb{R}}^2 \to \hat{\mathbb{R}}$ having the following property. For any approximation $\boldsymbol{x}$ of some real number $x \in D$, the pair $\boldsymbol{f}(\boldsymbol{x}) = (\boldsymbol{f}.fl(\boldsymbol{x}.fl), \boldsymbol{f}.erf(\boldsymbol{x}))$ is an approximation of $f(x)$. Thus, $\boldsymbol{f}.erf$ gives an upper bound on the absolute error of $\boldsymbol{f}.fl(\boldsymbol{x}.fl)$, $|\boldsymbol{f}.fl(\boldsymbol{x}.fl) - f(x)| \leq \boldsymbol{f}.erf(\boldsymbol{x})$. Considering $\boldsymbol{f}$ as an interval function, the

above property is just the fundamental property of interval arithmetic, [27] Property 2.12. Then, $\boldsymbol{f}$ is called an *approximation function* for f. Now consider an approximation function $\boldsymbol{f}$ for f such that for all $x \in D$ and any floating point name $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ of x, $(\boldsymbol{f}(\boldsymbol{x}_n))_{n\in\mathbb{N}}$ is a floating-point name of $f(x)$. Such an approximation function is called *approximation-continuous*. Additionally, if the two functions $\boldsymbol{f}.fl$ and $\boldsymbol{f}.erf$ of an approximation function $\boldsymbol{f}$ are computable, then $\boldsymbol{f}$ is called a *computable approximation function*. Finally, $f : D \to D$ is called *computable*, if there exists a computable approximation function $\boldsymbol{f}$ for f which is approximation-continuous.

The algorithm with the specification described at the beginning of this section reads

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

where $\boldsymbol{f}$ is an approximation-continuous approximation function for f specified below. To initialize $\boldsymbol{x}$, a rounding function $rd : \mathbb{Q} \times \mathbb{N} \to \hat{\mathbb{R}}^2$ is needed where $rd(x,m).fl$ is a floating-point number of precision m being the exactly rounded value of x for some rounding convention, in the following nearest. Clearly, the value $rd(x,m).err$ is an upper bound on the absolute rounding error, $rd(x,m).err = \frac{1}{2}ulp(x)$ if the rounding mode is nearest. The predicate $prec : \hat{\mathbb{R}}^2 \times \mathbb{Z} \to \{\textbf{true}, \textbf{false}\}$ is a test whether the relative error of $\boldsymbol{x}.fl$ is bounded by 10^{-p}. The semantics reads:

$$\text{If } \boldsymbol{x} \in \hat{\mathbb{R}}^2 \text{ approximates } x \in \mathbb{R} \text{ and } prec(\boldsymbol{x}, p) = \textbf{true} \text{ holds,} \qquad (10)$$
$$\text{then } |\boldsymbol{x}.fl - x| \le 10^{-p}|x| \text{ follows.}$$

While the object oriented notation is convenient for a compact and instructive description of the algorithm, in the following analytical analysis an

abbreviation for this notation is sometimes more handsome. As in the line of the preceding section, floating-point numbers and functions are indicated by a hat: $\hat{x} := \boldsymbol{x}.fl$ and $\hat{f} := \boldsymbol{f}.fl$. An over-bar indicates an error bound: $\overline{e} := \boldsymbol{x}.err$ and $\overline{erf} := \boldsymbol{f}.erf$. Hence, $\boldsymbol{x}$ is equivalent to $(\hat{x}, \overline{e})$ and $\boldsymbol{f}$ is equivalent to $(\hat{f}, \overline{erf})$.

Finally a remark on optimization. The algorithm is not optimized in performance. Including performance issues, in Line 11 something like $m \leftarrow a \cdot m + b$ can be used where $a > 1$ and $c \in \mathbb{N}$ are constants. Here, the aim is to find the minimal m to guarantee some given upper bound on the relative error of x_n.

3.2. Computability and correctness

It is clear that the rounding function rd is computable. So let us begin with the predicate $prec$.

Proposition 3.1. *The predicate*

$$prec(\boldsymbol{x}, p) := \begin{cases} \textbf{true} & \text{if } \boldsymbol{x}.err \leq \frac{10^{-p}}{1+10^{-p}}|\boldsymbol{x}.fl| \\ \textbf{false} & \text{otherwise} \end{cases} \tag{11}$$

is computable and satisfies (10).

Proof. Let $\boldsymbol{x}$ be an approximation of x. If $\boldsymbol{x}.err \leq \frac{10^{-p}}{1+10^{-p}}|\boldsymbol{x}.fl|$ holds, then $\boldsymbol{x}.err \leq 10^{-p}(|\boldsymbol{x}.fl| - \boldsymbol{x}.err)$ follows. Using $|\boldsymbol{x}.fl| \leq |\boldsymbol{x}.fl - x| + |x| \leq \boldsymbol{x}.err + |x|$, $|\boldsymbol{x}.fl - x| \leq \boldsymbol{x}.err \leq 10^{-p}(|\boldsymbol{x}.fl| - \boldsymbol{x}.err) \leq 10^{-p}|x|$ follows.

The predicate (11) only uses the approximation $\boldsymbol{x}$, basic arithmetic and finite tests. Hence, this formula is computable. $\square$

Note that the definition of the predicate also gives **true** in the singular case where $\boldsymbol{x}.fl = 0$ and $\boldsymbol{x}.err = 0$ and hence $x = 0$.

An algorithm for computing $\boldsymbol{f}.fl$ is possible by assumption. To derive an algorithm for computing $\boldsymbol{f}.erf$ on the absolute error, return to Equations (6) and (8).

Proposition 3.2. *Let $x \in D$ be given and $\boldsymbol{x}$ an approximation of x with $\boldsymbol{x}.fl \in D$. Assume that $\boldsymbol{f}.fl(\boldsymbol{x}.fl)$ computes the value $f(\boldsymbol{x}.fl)$ up to a cor-*

rectly rounded last bit in the significand.[1] *Furthermore assume $\boldsymbol{f}.fl(\boldsymbol{x}.fl).t = \boldsymbol{x}.fl.t$. Then the absolute error of $\boldsymbol{f}(\boldsymbol{x})$ is bounded from above by*

$$L(\boldsymbol{x}) \cdot \boldsymbol{x}.err + 2^{-\boldsymbol{x}.fl.t} \cdot |\boldsymbol{f}.fl(\boldsymbol{x}.fl)|. \tag{12}$$

Here, $L(\boldsymbol{x}) = \sup(|f'([\boldsymbol{x}.fl - \boldsymbol{x}.err, \boldsymbol{x}.fl + \boldsymbol{x}.err] \cap D)|)$.

Proof. Equation (6) gives $|\hat{f}(\hat{x}) - f(x)| \leq |f(\hat{x}) - f(x)| + |\hat{f}(\hat{x}) - f(\hat{x})|$. Using the mean value theorem, $|f(\hat{x}) - f(x)| \leq \sup(|f'([\hat{x} - \overline{e}, \hat{x} + \overline{e}])|)\overline{e}$ follows. According to the assumption on $\hat{f}$ and Theorem 2.3 of [12], $|\hat{f}(\hat{x}) - f(\hat{x})| \leq 2^{-m}|\hat{f}(\hat{x})|$ holds where m is the precision of $\hat{x}$. □

Corollary 3.1. *Let f be as specified in the beginning of this section, $\boldsymbol{f}.fl$ and $L(\boldsymbol{x})$ specified as in Proposition 3.2. Then there exists a function $\overline{L}(\boldsymbol{x})$ with $L(\boldsymbol{x}) \leq \overline{L}(\boldsymbol{x}) \leq \overline{L}_{max}$ for some $\overline{L}_{max} \geq 0$ such that $\boldsymbol{f}$ with $\boldsymbol{f}.erf(\boldsymbol{x}) = \overline{L}(\boldsymbol{x}) \cdot \boldsymbol{x}.err + 2^{-\boldsymbol{x}.fl.t} \cdot |\boldsymbol{f}.fl(\boldsymbol{x}.fl)|$ is an approximation-continuous, computable approximation function of f.*

Proof. Let $\overline{L}(\boldsymbol{x})$ be some computable upper bound of $L(\boldsymbol{x})$. $\overline{L}(\boldsymbol{x})$ can be computed by global optimization, for example by using interval arithmetic. Since f' is continuous and D compact, $L(\boldsymbol{x})$ is bounded. So, $\overline{L}(\boldsymbol{x}) \leq \overline{L}_{max}$ for some $\overline{L}_{max} \geq 0$. Also, $\boldsymbol{f}$ is computable. Using Proposition 3.2, it follows that $\boldsymbol{f}$ is also an approximation function of f. Remains to show that $\boldsymbol{f}$ is approximation-continuous. Let $(\boldsymbol{x}_n)_n$ be some floating-point name of $x \in D$. Clearly $\lim_{n\to\infty} \boldsymbol{f}.fl(\boldsymbol{x}_n.fl).t = \infty$ holds. Since $\lim_{n\to\infty} \boldsymbol{x}_n.err = 0$ and the sequences $(\overline{L}(\boldsymbol{x}_n))_n$ and $(|\boldsymbol{f}.fl(\boldsymbol{x}_n.fl)|)_n$ are bounded, $\lim_{n\to\infty} \boldsymbol{f}.erf(\boldsymbol{x}_n) = 0$ follows. Furthermore, by this result and the statement of Proposition 3.2, also $\lim_{n\to\infty} \boldsymbol{f}.fl(\boldsymbol{x}_n) = x$ holds. □

To summarize, the iteration (9) is performed in the algorithm by iterating a value $\hat{x}_n$ approximating x_n with an upper bound on its absolute error $\overline{e}_n$

---

[1]This assumption is pragmatic. The already mentioned software package MPFR implements this specification. The problem of achieving this task for transcendental functions may be of unknown cost and is known as The Table Maker's Dilemma, see `http://perso.ens-lyon.fr/jean-michel.muller/Intro-to-TMD.htm`. Additionally note that this assumption can be weakened without abandoning the main statements of this work.

according to

$$\hat{x}_{n+1} = \hat{f}(\hat{x}_n) \qquad\qquad \hat{x}_0 = rd(x, m) \qquad (13)$$
$$\overline{e}_{n+1} = \overline{L}(\hat{x}_n, \overline{e}_n)\overline{e}_n + 2^{-m}|\hat{x}_{n+1}| \qquad\qquad \overline{e}_0 = 2^{-m}|\hat{x}_0| \qquad (14)$$

where $\overline{L}(\hat{x}_n, \overline{e}_n)$ is a computable upper bound on $L(\hat{x}_n, \overline{e}_n)$ as described in the preceding corollary and m the precision of any floating-point number involved at that stage. This is Line 9 in the inner for-loop of the algorithm which is executed with successively increasing precision m, controlled by the outer do-while-loop. Finally, it has to be shown that this outer loop eventually terminates.

Proposition 3.3. *Let $x \in D$ with $x \neq 0$ be given and $(\boldsymbol{x}_m)_{m\geq 1}$ a floating-point name of x obeying $\boldsymbol{x}_m.fl.t = m$. Then $\lim_{m\to\infty} prec(\boldsymbol{x}_m, p) = \textbf{true}$ follows for all $p \in \mathbb{Z}$.*

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

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 the recursion (9) and $((\boldsymbol{x}_n)_m)_{m\geq 1}$ a sequence given according to the recursion equations (13) and (14) with increasing precision $(\boldsymbol{x}_n)_m.fl.t = m$. Then $((\boldsymbol{x}_n)_m)_{m\geq 1}$ is a floating-point name of x_n.*

Proof. Let $\overline{L}_{max}$ according to Corollary 3.1 and $\overline{M} \geq \sup\{|x| : x \in D\}$ such that $|\hat{x}_n| \leq \overline{M}$ holds for all n. Then Equation (14) leads to $\overline{e}_{n+1} \leq \overline{L}_{max}\overline{e}_n + 2^{-m}\overline{M}$. Iteration gives $\overline{e}_n \leq \overline{L}_{max}^n \overline{e}_0 + 2^{-m}\overline{M} \sum_{k=0}^{n-1} \overline{L}_{max}^k \leq 2^{-m}\overline{M} \sum_{k=0}^{n} \overline{L}_{max}^k$. Hence, for n fixed, $\lim_{m\to\infty}(\boldsymbol{x}_n)_m.err = 0$ follows and consequently also $\lim_{m\to\infty}(\boldsymbol{x}_n)_m.fl = x_n$. $\qquad\square$

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 $p \in \mathbb{Z}$.

The drawback of the algorithm is, that in the case $x_n = 0$ for some $n \leq N$, the computation does not terminate. This is only due to the fact that the relative error controls the outer do-while-loop. If the absolute error would be used instead, this drawback is eliminated. However, controlling the

relative error is more general. Consider for example a dynamics with positive phase space, the concentration of a substance for example. If the value varies in time over a wide range in scale, it is fortunate to illustrate the orbit in a logarithmic plot. If the relative error is controlled, the error bars in the plot are constant, in contrast to large varying error bars in the case where absolute errors are used.

Absolute errors are in the line with Computable Analysis. Replacing the test $prec(\boldsymbol{x}, p)$ by the test on $\boldsymbol{x}.err \leq 10^{-p}$ in the algorithm would give a segment $(x_n)_{0 \leq n \leq N}$ of the orbit with accuracy $|\hat{x}_n - x_n| \leq 10^{-p}$. It is now straightforward to see that the function $g : D \times \mathbb{N} \to D$ with $g(x, n) := f^n(x)$ is computable. Here, a function $g : D \times \mathbb{N} \to D$ is computable if there exists a computable approximation function $\boldsymbol{g} : \hat{\mathbb{R}}^2 \times \mathbb{N} \to \hat{\mathbb{R}}^2$ for g which is approximation-continuous with respect to the first argument.

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 previous section. Here is the formal definition.

Definition 3.1. The minimal precision, for which the described algorithm eventually halts is denoted by $m_{min}(x, N, p)$, where x, N and p are the corresponding input parameters. The growth rate of m_{min} is given by

$$\sigma(x, p) := \limsup_{N \to \infty} \frac{m_{min}(x, N, p)}{N}. \tag{15}$$

Then, the *loss of significance rate* $\sigma : \mathbb{Q} \cap D \to \mathbb{R}$ is defined by

$$\sigma(x) := \lim_{p \to \infty} \sigma(x, p). \tag{16}$$

To achieve bounds on the loss of significance rate, the drawback of the preceding subsection also makes problems here. If $x_n = 0$ for some $n \in \mathbb{N}$, the loss of significance rate may be unbounded. Therefore, one more assumption in addition to the ones on the dynamical system stated 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 furthermore, for any orbit $(x_n)_n$ under consideration, $x_n \neq 0$ holds for any $n \in \mathbb{N}$

as well as
$$\lim_{N \to \infty} \frac{\mathrm{ld}(\min\{|x_n| : n = 0, 1, \ldots, N\})}{N} = 0.$$

If only a finite range in scale is relevant, the additional assumption is no loss of generality. An example is the logistic equation where $0 \in D$ but 0 has no distinguished role. Instead of considering (D, f), consider the following dynamical system $(\tilde{D}, \tilde{f})$. 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 the additional assumption. 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 $m_{min}(x, N, p)$ as in Definition 3.1. Then, for given $p \in \mathbb{Z}$, there exists a constant $C \geq 0$, depending on f, such that $m_{min}(x, N, p) \leq C \cdot N + o(N)$ holds for all $N \in \mathbb{N}$, $x \in \mathbb{Q} \cap D$.*

Proof. According to the requirements made on (D, f), there are some constants $L > 1$ and $M > 0$ such that $\overline{e}_{n+1} \leq L\overline{e}_n + 2^{-m}M$ holds for all $n \in \mathbb{N}$ and all precisions m. Analogous to the treatment in the proof of Proposition 3.4, iteration gives $\overline{e}_n \leq 2^{-m}M \sum_{k=0}^{n} L^k = 2^{-m}M\frac{L^{n+1}-1}{L-1}$. Let $B(N) := \min\{|\hat{x}_n| : n = 0, 1, \ldots, N\} > 0$. Then, for all $n \leq N$, $\overline{e}_n/|\hat{x}_n| \leq \overline{e}_n/B(N) \leq \frac{M}{(L-1)g(N)}2^{-m}L^{N+1}$ follows. If now $\frac{M}{(L-1)B(N)}2^{-m}L^{N+1} \leq \frac{10^{-p}}{1+10^{-p}}$ holds, $prec((\hat{x}_n, \overline{e}_n), p) = \textbf{true}$ for all $n = 0, \ldots, N$. This leads to the bound $m_{min}(x, N, p) \leq \mathrm{ld}(L) \cdot N + \max(1, \mathrm{ld}(\frac{LM}{L-1}) - \mathrm{ld}(B(N)) + p \cdot \mathrm{ld}(10) + \mathrm{ld}(1 + 10^{-p}))$. $\square$

Corollary 3.2. *Let (D, f) be as in Assumption 3.1, $\sigma(x, p)$ as in (15) and $\sigma(x)$ the loss of significance rate. Then, for given $p \in \mathbb{Z}$, there exists some constant $C \geq 0$ such that $\sigma(x, p) \leq \sigma(x) \leq C$ holds for all $x \in \mathbb{Q} \cap D$.*

In the following, the main statements of this paper are be formulated: A lower and an upper bound for the loss of significance rate is given. Furthermore, the relation of these bounds to the Lyapunov exponent $\lambda(x)$ is shown. Before the theorem is stated, for sake of completeness, the definition of the Lyapunov exponent and its basic properties are presented.

Definition 3.2. Let (D, f) be a dynamical system, $D \subseteq \mathbb{R}$ compact and $f : D \to D$ continuously differentiable on D. Then the *Lyapunov exponent* at $x \in D$ is defined by

$$\lambda(x) := \lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} \ln(|f'(f^k(x))|) \tag{17}$$

if the limit exists.

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

(a) If (D, f) has an *invariant measure* ρ, then the limit in Equation (17) exists ρ-almost everywhere.

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

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

These properties are a direct consequence of the Birkhoff ergodic theorem, see [15], Theorem 4.1.2 and Corollary 4.1.9. Now the first theorem.

Theorem 3.1. *Let (D, f) be as in Assumption 3.1, $\sigma(x, p)$ as in (15) and $\lambda(x)$ the Lyapunov exponent of (D, f). Then $\sigma(x, p) \geq \max(0, \lambda(x))/\ln(2)$ holds for all $x \in \mathbb{Q} \cap D$, $p \in \mathbb{Z}$ if $\lambda(x)$ exists.*

Proof. Let $N \in \mathbb{N}$ be given and $M > 0$ a constant with $|\hat{x}_n| \leq M$ for all $n \in \mathbb{N}$. According to Equation (14) and Proposition 3.2, $\bar{e}_{n+1} \geq |f'(x_n)|\bar{e}_n$ holds. Iteration gives $\bar{e}_N \geq |\hat{x}_0|2^{-m} \prod_{n=0}^{N-1} |f'(x_n)|$. So, $\frac{\bar{e}_N}{|\hat{x}_N|} \geq \frac{|\hat{x}_0|2^{-m}}{M} \prod_{n=0}^{N-1} |f'(x_n)|$ follows. A necessary condition for the algorithm to terminate is therefore $\frac{|\hat{x}_0|}{M}2^{-m} \prod_{n=0}^{N-1} |f'(x_n)| \leq \frac{10^{-p}}{1+10^{-p}}$. This gives the bound on $m_{min}(x, N, p) \geq \sum_{n=0}^{N-1} \mathrm{ld}(|f'(x_k)|) + \mathrm{ld}(\frac{|\hat{x}_0|}{M}) + p \cdot \mathrm{ld}(10) + \mathrm{ld}(1+10^{-p})$. Following the definitions of $\sigma(x, p)$ and the Lyapunov exponent, $\sigma(x, p) \geq \lambda(x)/\ln(2)$ follows. $\square$

Before a realistic upper bound on $\sigma(x, p)$ can be presented, one more definition is needed.

Definition 3.3. Let $\alpha > 0$ then define a function $\eta_\alpha : (0, \infty) \to \mathbb{R}$ by

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

23

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 Lyapunov exponent $\lambda(x)$ exists, $\lambda(x) \leq \overline{\lambda}_\alpha(x)$ holds.*

Proof. According to the requirements made on (D, f), f is Lipschitz with a Lipschitz constant $L > 0$. Furthermore, let $\alpha > 0$ be given. 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 on $\limsup_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} \eta_\alpha(|f'(f^k(x))|) \leq \ln(\max(\alpha, L))$. The second assertion follows from the fact that $\ln(x) \leq \eta_\alpha(x)$ holds for all $x > 0$, $\alpha > 0$. $\qquad\square$

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

$$\lim_{\substack{\alpha \to 0 \\ \alpha > 0}} \overline{\lambda}_\alpha(x) =: \overline{\lambda}(x) \tag{18}$$

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. Letting $\alpha \to 0$, $\alpha > 0$, the assertion follows. $\qquad\square$

Theorem 3.2. *Let (D, f) be as in Assumption 3.1, $\sigma(x, p)$ as in (15) and $\overline{\lambda}(x)$ as in (18). Let $x \in \mathbb{Q} \cap D$ be given, then for any $\varepsilon > 0$ there is some $p_0 \in \mathbb{Z}$ such that for all $p \geq p_0$,*

$$\sigma(x, p) \leq \frac{1}{\ln(2)} \max(0, \overline{\lambda}(x)) + \varepsilon$$

holds if $\lambda(x)$ exists.

So there is the following bound on the loss of significance rate.

Corollary 3.3. *Let (D, f) be as in Assumption 3.1, $\sigma(x, p)$ as in (15), $\sigma(x)$ the loss of significance rate, $\overline{\lambda}(x)$ as in (18) and $\lambda(x)$ the Lyapunov exponent. Then,*

$$\frac{1}{\ln(2)} \max(0, \lambda(x)) \leq \sigma(x) \leq \frac{1}{\ln(2)} \max(0, \overline{\lambda}(x))$$

holds for all $x \in \mathbb{Q} \cap D$ if $\lambda(x)$ exists.

24

Before the proof of the theorem 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 $\varepsilon > 0$. Two cases are considered.

1st 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.

2nd 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 prove 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. $\qquad\square$

Now everything is prepared to prove Theorem 3.2.

Proof of Theorem 3.2. Let $N \in \mathbb{N}$, $B(N) := \min\{|\hat{x}_n| : n = 0, 1, \ldots, N\} > 0$ and $M > 0$ a constant with $|\hat{x}_n| \leq M$ for all $n \in \mathbb{N}$. Starting with Equation (14) and iterating gives

$$\overline{e}_n = \overline{e}_0 \prod_{l=0}^{n-1} \overline{L}(\hat{x}_l, \overline{e}_l) + 2^{-m} \sum_{k=1}^{n} |\hat{x}_k| \prod_{l=k}^{n-1} \overline{L}(\hat{x}_l, \overline{e}_l)$$

$$= 2^{-m} \sum_{k=0}^{n} |\hat{x}_k| \prod_{l=k}^{n-1} \overline{L}(\hat{x}_l, \overline{e}_l) \leq M 2^{-m} \sum_{k=0}^{n} \prod_{l=k}^{n-1} \overline{L}(\hat{x}_l, \overline{e}_l).$$

Define

$$sp_n := \sum_{k=0}^{n} \prod_{l=k}^{n-1} \overline{L}(\hat{x}_l, \overline{e}_l)$$

and

$$SP(N) := \max\{sp_n : n = 0, 1, \ldots, N\}.$$

Then, $\overline{e}_n \leq M 2^{-m} SP(N)$ follows for all $n \leq N$. A sufficient condition for the algorithm to terminate is given by $\frac{M}{B(N)} 2^{-m} SP(N) \leq \frac{10^{-p}}{1+10^{-p}}$. Hence,

$$m_{min}(x, N, p) \leq \max(1, C - \mathrm{ld}(B(N)) + \mathrm{ld}(SP(N)))$$

25

follows with $C = \mathrm{ld}(M) + p \cdot \mathrm{ld}(10) + \mathrm{ld}(1 + 10^{-p})$. Using the Assumption 3.1 leads to

$$\sigma(x, p) \leq \frac{1}{\ln(2)} \max\left(0, \limsup_{N \to \infty} \frac{1}{N} \ln(SP(N))\right).$$

By definition, $\limsup_{N \to \infty} \frac{\ln(SP(N))}{N} \leq \limsup_{n \to \infty} \frac{\ln(sp_n)}{n}$ follows and hence

$$\sigma(x, p) \leq \frac{1}{\ln(2)} \max\left(0, \limsup_{n \to \infty} \frac{\ln(sp_n)}{n}\right).$$

Next let

$$p_{k,n} := \prod_{l=k}^{n-1} \overline{L}(\hat{x}_l, \overline{e}_l)$$

for $n \in \mathbb{N}$ and $k \leq n$, and furthermore

$$P_n := \max\{p_{k,n} : k = 0, 1, \ldots, n\},$$

then $sp_n \leq (n+1)P_n$ follows for all $n \in \mathbb{N}$. This gives

$$\sigma(x, p) \leq \frac{1}{\ln(2)} \max\left(0, \limsup_{n \to \infty} \frac{\ln(P_n)}{n}\right).$$

Let $K(n) \in \{0, \ldots, n\}$ be the smallest number such that $\prod_{l=K(n)}^{n-1} \overline{L}(\hat{x}_l, \overline{e}_l) = P_n$. Then consider $\frac{\ln(P_n)}{n} = \frac{1}{n} \sum_{l=K(n)}^{n-1} \ln(\overline{L}(\hat{x}_l, \overline{e}_l))$. Let L' be a Lipschitz constant of f', then $L(\hat{x}_n, \overline{e}_n) \leq |f'(x_n)| + L'2\overline{e}_n$ holds for all $n \in \mathbb{N}$. Consequently, there exists some $\overline{L}' \geq L'$ such that $\overline{L}(\hat{x}_n, \overline{e}_n) \leq |f'(x_n)| + 2\overline{L}'\overline{e}_n$ holds for all $n \in \mathbb{N}$. This inequality leads to $\overline{L}(\hat{x}_n, \overline{e}_n) \leq |f'(x_n)| + 2\overline{L}'M\frac{10^{-p}}{1+10^{-p}} \leq |f'(x_n)| + 2\overline{L}'M \cdot 10^{-p}$. Inserting gives

$$\frac{\ln(P_n)}{n} \leq \frac{1}{n} \sum_{l=K(n)}^{n-1} \ln(|f'(x_l)| + 2\overline{L}'M \cdot 10^{-p}).$$

Now let $\varepsilon > 0$ and $0 < \alpha < 1$ be given. Then choose $p_0 \in \mathbb{N}$ such that $\sqrt{2\overline{L}'M \cdot 10^{-p_0}} < \min(\alpha, \ln(2)\frac{\varepsilon}{2})$ holds. Then for all $p \geq p_0$, the above lemma gives

$$\frac{\ln(P_n)}{n} \leq \frac{1}{n} \sum_{l=K(n)}^{n-1} \left(\eta_\alpha(|f'(x_l)|) + \ln(2)\frac{\varepsilon}{2}\right) \tag{19}$$

$$\leq \ln(2)\frac{\varepsilon}{2} + \frac{1}{n} \sum_{l=K(n)}^{n-1} \eta_\alpha(|f'(x_l)|). \tag{20}$$

Consider the sequence $(K(n))_{n\in\mathbb{N}}$. Observe that, first the sequence $(K(n))_{n\in\mathbb{N}}$ is increasing and second if $K(n+1) > K(n)$ for some $n \in \mathbb{N}$, then $K(n+1) = n$ or $K(n+1) = n+1$. There are two cases.

1st case: $(K(n))_{n\in\mathbb{N}}$ is bounded. Then, there exists some constant $N_0 \in \mathbb{N}$ such that $K(n) = K(N_0)$ holds for all $n \geq N_0$. Choose now α small enough such that $\overline{\lambda}_\alpha(x) \leq \overline{\lambda}(x) + \ln(2)\frac{\varepsilon}{2}$ holds. Then, compute the upper limit to $\limsup_{n\to\infty} \frac{1}{n}\sum_{l=K(n)}^{n-1}\eta_\alpha(|f'(x_l)|) = \limsup_{n\to\infty}\frac{1}{n}\sum_{l=K(N_0)}^{n-1}\eta_\alpha(|f'(x_l)|) = \limsup_{n\to\infty}\frac{1}{n}\sum_{l=0}^{n-1}\eta_\alpha(|f'(x_l)|) = \overline{\lambda}_\alpha(x)$. By taking the upper limit of (20), $\limsup_{n\to\infty}\frac{\ln(P_n)}{n} \leq \ln(2)\frac{\varepsilon}{2} + \overline{\lambda}_\alpha(x) \leq \ln(2)\varepsilon + \overline{\lambda}(x)$ follows.

2nd case: $(K(n))_{n\in\mathbb{N}}$ is not bounded. Then, for any $\delta > 0$ and any $N_0 \in \mathbb{N}$ there is some $n \geq N_0$ with $\frac{\ln(P_n)}{n} < \delta$. Since, by definition, $\sum_{l=0}^{n-1}\ln(\overline{L}(\hat{x}_l,\overline{e}_l)) \leq \ln(P_n)$ holds as well as $|f'(x_l)| \leq \overline{L}(\hat{x}_l,\overline{e}_l)$, the inequality $\frac{1}{n}\sum_{l=0}^{n-1}\ln(|f'(x_l)|) \leq \frac{\ln(P_n)}{n}$ follows. This shows $\lambda(x) \leq 0$.

Next it is stated that for all $\varepsilon > 0$ and $p \geq p_0$, $\limsup_{n\to\infty}\frac{\ln(P_n)}{n} \leq \ln(2)\varepsilon$ holds. This shows $\sigma(x,p) \leq \varepsilon = \frac{1}{\ln(2)}\max(0,\lambda(x))+\varepsilon \leq \frac{1}{\ln(2)}\max(0,\overline{\lambda}(x))+\varepsilon$.

Assume otherwise. Then, for some $\varepsilon > 0$ and $N \in \mathbb{N}$, first $\frac{\ln(P_N)}{N} > \ln(2)\varepsilon$ holds and second $\lambda(x)-\ln(2)\frac{\varepsilon}{4} < \frac{1}{n}\sum_{l=0}^{n-1}\ln(|f'(x_l)|) < \lambda(x)+\ln(2)\frac{\varepsilon}{4}$ holds for all $n \geq K(N)$. Using (20), the first expression gets $\frac{1}{N}\sum_{l=K(N)}^{N-1}\eta_\alpha(|f'(x_l)|) > \ln(2)\frac{\varepsilon}{2}$. Choose α small enough such that $\eta_\alpha(|f'(x_l)|) = \ln(|f'(x_l)|)$ holds for all $l \leq N$. Then, for sufficiently high p, $\frac{1}{N}\sum_{l=K(N)}^{N-1}\ln(|f'(x_l)|) > \ln(2)\frac{\varepsilon}{2}$ follows. In the second statement, the sum can be split the following way: $\frac{1}{N}\sum_{l=0}^{K(N)-1}\ln(|f'(x_l)|) + \frac{1}{N}\sum_{l=K(N)}^{N-1}\ln(|f'(x_l)|) < \lambda(x) + \ln(2)\frac{\varepsilon}{4}$. The first addend on the left side is bounded form below by $\frac{K(N)}{N}(\lambda(x) - \ln(2)\frac{\varepsilon}{4}) \geq \lambda(x) - \ln(2)\frac{\varepsilon}{4}$, the second addend is bounded from below by $\ln(2)\frac{\varepsilon}{2}$. Hence, $\lambda(x) + \ln(2)\frac{\varepsilon}{4} < \frac{1}{N}\sum_{l=0}^{N-1}\ln(|f'(x_l)|) < \lambda(x) + \ln(2)\frac{\varepsilon}{4}$ follows, but this is a contradiction. $\square$

In the end, it is shown that, if $\lambda(x) = \overline{\lambda}(x)$ holds, the algorithm presented here is optimal with respect to the loss of significance rate. This means that no algorithm with the specification presented at the beginning of this section has a lower loss of significance rate than the algorithm presented in this section.

Proposition 3.8. *Let $(x_n)_n$ be an orbit of the dynamical system (D, f) and $\lambda(x_0) > 0$. Then, for any $\varepsilon > 0$ there exists an $N_0 \in \mathbb{N}$ such that for any $N \geq N_0$ there is some $\delta > 0$ such that the following holds. Let an initial*

value $y_0 \in [x_0 - \delta, x_0 + \delta] \cap D$ be given and consider the corresponding orbit $(y_n)_n$. Then,

$$|y_N - x_N| \geq e^{N(\lambda(x_0) - \varepsilon)}|y_0 - x_0|$$

holds.

Proof. Let $\varepsilon' > 0$ be given. Then there exists some $N_0 \in \mathbb{N}$ such that $\frac{1}{N}\sum_{n=0}^{N-1} \ln(|f'(x_n)|) \geq \lambda(x_0) - \varepsilon'$ holds for all $N \geq N_0$. For given $N \geq N_0$ there is some $\delta' > 0$ with $\min\{|f'(x_n)| : 0 \leq n \leq N\} > \delta'$, otherwise $\lambda(x_0)$ would not exist. Consider now an orbit $(y_n)_n$ with $|y_n - x_n| \leq 2\sigma\frac{\delta'}{L'}$ for $n = 0, \ldots, N$ where L' is a Lipschitz constant of f' and $0 < \sigma < 1$ arbitrary. Then, for $n = 1, \ldots, N$, the following estimation holds.

$$\begin{aligned}
|y_n - x_n| &= |f(y_{n-1}) - f(x_{n-1})| \\
&= |f'(x_{n-1})(y_{n-1} - x_{n-1}) + \frac{1}{2}f''(\xi_{n-1})(y_{n-1} - x_{n-1})^2| \\
&\geq (|f'(x_{n-1})| - \frac{1}{2}|f''(\xi_{n-1})| \cdot |y_{n-1} - x_{n-1}|)|y_{n-1} - x_{n-1}| \\
&\geq (|f'(x_{n-1})| - \frac{1}{2}L'|y_{n-1} - x_{n-1}|)|y_{n-1} - x_{n-1}| \\
&\geq (|f'(x_{n-1})| - \sigma\delta')|y_{n-1} - x_{n-1}|
\end{aligned}$$

where $\xi_{n-1} \in [x_{n-1}, y_{n-1}] \cup [y_{n-1}, x_{n-1}]$. Iterating finally gives $|y_N - x_N| \geq \prod_{n=0}^{N-1}(|f'(x_n)| - \sigma\delta')|y_0 - x_0|$. Now determine some constant $C > 0$ such that $\ln(|f'(x_n)| - \sigma\delta') \geq \ln(|f'(x_n)|) - C$ holds for all $n = 0, \ldots, N$ the following way. A short calculation shows that this is equivalent to $C \geq \ln(1+\frac{\sigma\delta'}{|f'(x_n)|-\sigma\delta'})$. Using $\ln(1+\frac{\sigma\delta'}{|f'(x_n)|-\sigma\delta'}) \leq \ln(1+\frac{\sigma\delta'}{\delta'-\sigma\delta'}) = \ln(1+\frac{\sigma}{1-\sigma}) \leq \frac{\sigma}{1-\sigma}$ finally gives $C \geq \frac{\sigma}{1-\sigma}$ as a sufficient condition. Set $C = \frac{\sigma}{1-\sigma}$. Let $\varepsilon > 0$ be given. Set $C = \varepsilon' = \varepsilon/2$, then

$$\begin{aligned}
\sum_{n=0}^{N-1} \ln(|f'(x_n)| - \frac{C\delta'}{1+C}) &\geq \sum_{n=0}^{N-1} \ln(|f'(x_n)|) - NC \\
&\geq N(\lambda(x_0) - \varepsilon' - C)
\end{aligned}$$

and hence

$$|y_N - x_N| \geq e^{N(\lambda(x_0) - \varepsilon' - C)}|y_0 - x_0|$$

follows. $\square$

Proposition 3.9. *Let (D, f) be as in Assumption 3.1 and $x \in \mathbb{Q} \cap D$ given such that $\lambda(x)$ exists and $\lambda(x) > 0$. Consider an algorithm computing an initial segment $(x_n)_{0 \leq n \leq N}$ of the orbit of x with relative error $\leq 10^{-p}$ for some $N \in \mathbb{N}$, $p \in \mathbb{Z}$. Then the algorithm has a loss of significance rate $\sigma(x) \geq \frac{1}{\ln(2)} \lambda(x)$.*

Proof. Let $0 < \varepsilon < \lambda(x)$ be given and N big enough such that the previous proposition holds for some $\delta > 0$. Choose some $y \in [x - \delta, x + \delta] \cap D$, $y \neq x$. Let p_0 be big enough such that

$$2M10^{-p_0} < e^{N(\lambda(x) - \varepsilon)} |y - x| \tag{21}$$

holds, where $M > 0$ such that $|x| \leq M$ for all $x \in D$. Consider y as the initial value of another orbit $(y_n)_n$. Then, with the above proposition,

$$|y_N - x_N| > 10^{-p_0}(|x_N| + |y_N|) \tag{22}$$

follows. Condition (21) can also be written as $\delta > 2M10^{-p_0} e^{-N(\lambda(x) - \varepsilon)}$.

Consider now some precision m, the algorithm actually is working with. Assume for simplicity further that for the initial value $x = \hat{x}_0$ holds and assume without loss of generality $\delta < \frac{1}{2} ulp(\hat{x}_0)$. Then, first, $\hat{y}_0 = \hat{x}_0$ holds. Second, the above condition gives $2^{-(m+1)} > 10^{-p_0} e^{-N(\lambda(x) - \varepsilon)}$ since $ulp(\hat{x}_0) \leq 2^{-m+1}|\hat{x}_0| \leq 2^{-m+1}M$ holds. In other words, the above condition gives an upper bound $m < p_0 \cdot \mathrm{ld}(10) + N(\lambda(x) - \varepsilon)/\ln(2) - 1$ on the needed precision m. Assume furthermore that m is big enough such that $\hat{x}_N$ and $\hat{y}_N$ is computed with the demanded precision, that is $|\hat{x}_N - x_N| \leq 10^{-p_0}|x_N|$ and $|\hat{y}_N - y_N| \leq 10^{-p_0}|y_N|$ holds. Using (22) gives $\hat{x}_N \neq \hat{y}_N$. But this is a contradiction since $\hat{y}_0 = \hat{x}_0$. So the upper bound on m calculated above is still too small. Hence, $m \geq p_0 \cdot \mathrm{ld}(10) + N(\lambda(x) - \varepsilon)/\ln(2) - 1$ must hold. Since Condition (21) also holds for any $N' > N$ and the same p_0 as well as for any $p \geq p_0$, $\sigma(x, p) \geq (\lambda(x) - \varepsilon)/\ln(2)$ follows for all $p \geq p_0$. Computing $\sigma(x)$ finally gives the assertion. $\square$

Furthermore, if $\overline{\lambda}(x) = \lambda(x)$ holds, then Corollary 3.3 gives $\sigma(x) = \frac{1}{\ln(2)} \max(0, \lambda(x))$ for the algorithm presented at the beginning of this section. Using the above proposition then leads to the following theorem.

Theorem 3.3. *Let (D, f) be as in Assumption 3.1 and $x \in \mathbb{Q} \cap D$ given such that $\lambda(x)$ exists and $\overline{\lambda}(x) = \lambda(x)$ holds. Consider an algorithm computing an initial segment $(x_n)_{0 \leq n \leq N}$ of the orbit of x with relative error $\leq 10^{-p}$. Then this algorithm has a loss of significance rate greater or equal to that of the algorithm specified by the recursion (13) and (14).*

29

4. Conclusions

In this paper, two main issues are addressed. First it is shown that a mathematically rigorous treatment of the computability aspects of the iteration of a real function in terms of arbitrary-precision floating-point arithmetic including automated error analysis is straightforward. 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 approach does not only allow answers concerning the existence of an algorithm which meets the requirements of computability theory, but it also allows a treatment of its space complexity in form of the loss of significance rate (which is actually the lookahead in Computable Analysis) and optimality as discussed in the preceding section. As a consequence, the approach here enables a motivated reader the real implementation and supports a practical performance analysis.

Second, the results show that the Lyapunov 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 Lyapunov exponent describes the rate of divergence in the course of time of initially infinitesimal nearby states. For two states having a small but finite initial separation, the Lyapunov exponent has only relevance for short time scales [6]. The reason is that due to the boundedness of the phase space, any two different orbits cannot separate arbitrarily far away. However, the loss of significance rate shows that the Lyapunov exponent has on long time scales not only an asymptotic significance but also a concrete practical one.

References

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

[2] J. Blanck, Efficient exact computation of iterated maps, The Journal of Logic and Algebraic Programming 64 (2005) 41–59.

[3] J. Blanck, Exact real arithmetic using centred intervals and bounded error terms, The Journal of Logic and Algebraic Programming 66 (2006) 50–67.

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

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

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

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

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

[9] D. Goldberg, What every computer scientist should know about floating-point arithmetic, ACM Computing Surveys 23 (1991) 5–48.

[10] C. Grebogi, S.M. Hammel, J.A. Yorke, T. Sauer, Shadowing of physical trajectories in chaotic dynamics: Containment and refinement, Physical Review Letters 65 (1990) 1527–1530.

[11] S.M. Hammel, J.A. Yorke, C. Grebogi, Do numerical orbits of chaotic dynamical processes represent true orbits?, Journal of Complexity 3 (1987) 136–145.

[12] N.J. Higham, Accuracy and Stability of Numerical Algorithms, SIAM, Philadelphia, 2. edition, 2002.

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

[14] IEEE 2008, IEEE Standard for Floating-Point Arithmetic (ANSI/IEEE Std 754-2008), IEEE, New York, 2008.

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

[16] K.I. Ko, Complexity Theory of Real Functions, Birkhäuser, Boston Basel Berlin, 1991.

[17] R.J. Lohner, Interval arithmetic in staggered correction format, in: E. Adams, U. Kulisch (Eds.), Scientific Computing with Automatic Result Verification, volume 189 of *Mathematics in Science and Engineering*, Academic Press, San Diego, 1993.

[18] N. Metropolis, G.C. Rota, S.M. Tanny, Significance arithmetic: The carrying algorithm, Journal of Combinatorial Theory A 14 (1973) 386–421.

[19] R.E. Moore, Interval Analysis, Prentice-Hall Inc., Englewood-Cliffs N.J., 1966.

[20] M. Mrozek, Rigorous numerics of chaotic dynamical systems, in: P. Garbaczewski, M. Wolf, W. Aleksander (Eds.), Chaos - The Interplay Between Stochastic and Deterministic Behaviour, volume 457 of *Lecture Notes in Physics*, Springer-Verlag, Berlin Heidelberg New York, 1995, pp. 283–296.

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

[22] N.T. Müller, Efficient implementation of exact real numbers, 2005. Tutorial. An electronic copy of it is available at http://www.cc.kyoto-su.ac.jp/~yasugi/page/Kakenhi/mueller.pdf.

[23] A. Neumaier, T. Rage, Rigorous chaos verification in discrete dynamical systems, Physica D 67 (1993) 327–346.

[24] M.B. Pour-El, J.I. Richards, Computability in Analysis and Physics, Springer-Verlag, Berlin Heidelberg New York, 1989.

[25] T. Rage, A. Neumaier, C. Schlier, Rigorous verification of chaos in a molecular model, Physical Review E 50 (1994) 2682–2688.

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

[27] H. Ratschek, J. Rokne, New Computer Methods for Global Optimization, Ellis Horwood Limited, Chichester, 1988.

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

[29] S.M. Rump, Fast and parallel interval arithmetic, BIT 39 (1999) 534–554.

[30] T. Sauer, J.A. Yorke, Rigorous verification of trajectories for the computer simulation of dynamical systems, Nonlinearity 4 (1991) 961–979.

[31] M. Sofroniou, G. Spaletta, Precise numerical computation, The Journal of Logic and Algebraic Programming 64 (2005) 113–134.

[32] P.H. Sterbenz, Floating Point Computation, Prentice Hall, Englewood Cliffs, NJ, 1974.

[33] K. Weihrauch, Computability, volume 9 of *EATCS Monographs on Theoretical Computer Science*, Springer, Berlin, 1987.

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

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