Quantitative hyperbolicity estimates in one-dimensional dynamics

S Day[1]‡, H Kokubu[2]§ S Luzzatto[3], K Mischaikow[4]‖, H Oka[5]¶ and P Pilarczyk[6]+ *

[1]The College of William and Mary, Department of Mathematics, P.O.Box 8795, Williamsburg, VA 23187-8795, U.S.A., http://www.math.wm.edu/~sday/

[2]Kyoto University, Department of Mathematics, Kyoto 606-8502, Japan, http://www.math.kyoto-u.ac.jp/~kokubu/

[3]Mathematics Department, Imperial College, 180 Queen's Gate, London SW7 2AZ, United Kingdom, http://www.ma.ic.ac.uk/~luzzatto/

[4]Department of Mathematics, Hill Center-Busch Campus, Rutgers, The State University of New Jersey, 110 Frelinghusen Rd, Piscataway, NJ 08854-8019, U.S.A., http://www.math.rutgers.edu/~mischaik/

[5]Department of Applied Mathematics and Informatics, Faculty of Science and Technology, Ryukoku University, Seta, Otsu 520-2194, Japan, http://www.math.ryukoku.ac.jp/~oka/

[6]Kyoto University, Department of Mathematics, Kyoto 606-8502, Japan, http://www.pawelpilarczyk.com/

AMS classification scheme numbers: Primary: 37D20; Secondary: 37E05, 37M25

Abstract. We develop a rigorous computational method for estimating the Lyapunov exponents in uniformly expanding regions of the phase space for one-dimensional maps. Our method uses rigorous numerics and graph algorithms to provide results that are mathematically meaningful and can be achieved in an efficient way.

Submitted to: *Nonlinearity*

‡ S. Day was partially supported by NSF/DMS-9983660.

§ H. Kokubu was partially supported by Grant-in-Aid for Scientific Research (No. 17340045), Ministry of Education, Science, Technology, Culture and Sports, Japan.

‖ K. Mischaikow was partially supported by NSF grants DMS-0511115 and DMS-0107396, by DARPA, and by DOE grant DE-FG02-05ER25711.

¶ H. Oka was partially supported by Grant-in-Aid for Scientific Research (No. 17540206), Ministry of Education, Science, Technology, Culture and Sports, Japan.

+ P. Pilarczyk was partially supported by the JSPS Postdoctoral Fellowship No. P06039 and by Grant-in-Aid for Scientific Research (No. 1806039), Ministry of Education, Science, Technology, Culture and Sports, Japan.

* H. Kokubu, S. Luzzatto, H. Oka, and P. Pilarczyk were partially supported by The Royal Society (UK).

1. Introduction and statement of results

The theory of one-dimensional dynamical systems has seen huge growth over the last couple of decades and is undoubtedly one of the most subtle and sophisticated current areas of research. The last few years have seen the resolution of some of the main classical problems in the area, such as the hyperbolicity conjecture [11]. The fact remains, however, that the deepest results are also generally the most abstract ones, often providing existence or genericity results of little use when studying a specific system or when trying to obtain explicit quantitative information. The aim of this paper is to establish some computational methods to obtain *rigorous* and explicit numerical bounds for certain dynamical quantities of great relevance and importance in the theory of one-dimensional dynamics. The key idea of our approach is to convert the problem into the computation of some quantities for a weighted directed graph constructed from a given family of dynamical systems, for which various graph algorithms work very effectively.

1.1. Lyapunov exponents

A key property of interest in many arguments in the dynamical systems theory is that of *hyperbolicity*, or *non-zero Lyapunov exponents* [4]. In dimension one the Lyapunov exponent at a point x is simply defined as

$$\lambda(x) = \liminf_{n\to\infty} \log |Df^n(x)|^{1/n}.$$

There is an extensive literature on computational methods for estimating Lyapunov exponents, ranging from basic derivative calculations to relatively sophisticated arguments, but the asymptotic nature of the Lyapunov exponent makes it intrinsically difficult, if not in many cases impossible, to estimate it accurately by computational methods. This is the case, for example, if the orbit of x accumulates onto a critical point (or, in the higher-dimensional case, if the map f exhibits tangencies between stable and unstable manifolds) as then even the sign of the Lyapunov exponent is impossible to establish within a finite number of iterations.

It turns out, however, that as long as we exclude this case, we can obtain some rigorous numerical estimates. We consider the following setting. Let I be an interval and $\Delta \subset I$ a finite union of open subintervals having disjoint closures and containing all the critical points of f. We suppose that $f\colon I \setminus \Delta \to I$ is a C^1 map. We are interested in obtaining some uniform expansivity estimates for pieces of orbits outside Δ in the following form:

Statement 1 *There exist constants $C, \lambda > 0$ such that for every x and for every n such that $f^i(x) \notin \Delta$ for all $i = 0, \ldots, n-1$ we have*

$$|Df^n(x)| \geq Ce^{\lambda n}. \tag{1}$$

This statement is always true under the assumptions that the map f is at least C^2 with all periodic orbits hyperbolic repelling, and that Δ is a neighbourhood of the critical points [15, 16, 17]. This result has had huge applicability and scope in the "abstract" theory of one-dimensional dynamics; see e.g. [16] and references therein. It should be noted, however, that the assumption of all hyperbolic periodic points being repelling is non-trivial and in practice unverifiable for specific maps. In addition to this, the theorem itself does not give any indication of the actual values of the constants λ and C and their relationship to the size of the critical neighbourhood Δ (although, remarkably, in certain cases it can be shown that the exponent λ is independent of the size of Δ [19]).

In this paper we are interested in proving Statement 1 for some specific maps f and specific choices of critical neighbourhoods Δ, and in obtaining rigorous explicit numerical bounds for the constants C and λ. Our work is in part motivated by [12] where this kind of estimate appears, and one of our goals is to contribute to the development of a computational version of Jacobson-Benedicks-Carleson type theorem [5, 9] in the future. Therefore, we are also interested in the following statement closely related to Statement 1:

Statement 2 *There exists a constant $\lambda_0 > 0$ such that for every x and every n such that $f^i(x) \notin \Delta$ for all $i = 0, \dots, n-1$ and either $x \in f(\Delta)$ and/or $f^n(x) \in \Delta$ we have*

$$|Df^n(x)| \geq e^{\lambda_0 n}. \tag{2}$$

The difference between Statements 1 and 2 is that in Statement 2 we have some additional information about the piece of orbit under consideration in the form of the location of either its initial or its final point. The conclusion is a significantly stronger expansivity estimate without the (possibly small) multiplicative constant C.

Before discussing the details of our strategy and methodology, we mention some recent related work on the subject of the verification of hyperbolicity. In particular, computational methods have been used by Tucker [25] as part of his groundbreaking work on the verification of hyperbolicity for the return map of the Lorenz attractor, by Arai [1, 2] and by Hruska [8] for the real and complex Hénon map family. Interestingly, all these approaches are very different, with [1, 2] using ideas from the Conley index theory, while [8] and [25] in different ways more or less explicitly constructing a family of invariant conefields. It is not completely clear to us to what extent these methods actually give explicit bounds for the hyperbolicity constants and it would certainly be interesting to see if some combinations of the methods given there with ours might lead to some results in this direction.

1.2. Rigorous hyperbolicity computations

The main result of this paper is the development of a combination of procedures for the explicit verification of Statements 1 and 2 for arbitrary maps. More specifically, we develop a fully fledged computer-run algorithm with the following input and output.

Input: The map $f\colon I \to I$ and the region $\Delta \subset I$. In detail:

- the bounded interval $I \subset \mathbb{R}$,

- the region $\Delta \subset I$ defined as a finite union of open intervals,
- a method for computing, for each interval $J \subset I$, a (possibly small) interval which covers $f(J)$,
- a method for computing, for each pair of intervals $J \subset I \setminus \Delta$, $K \subset I$, a (possibly tight) lower estimate for $\log |f'(x)|$ valid for all $x \in J$ such that $f(x) \in K$.

There are also several other "auxiliary" parameters related to the level of precision of the computations which can be thought of as part of the input, too. The computations can therefore be run either in "simple" mode in which these additional parameters are set to some default values, or in "advanced" mode in which they can be modified to increase the efficiency of the calculations taking into account specific features of the map under consideration.

Output: The constants C, λ, λ_0 with which Statements 1 and 2 hold true, or "fail" if these statements cannot be verified. More precisely, the algorithm returns the largest possible constant $\lambda \in \mathbb{R}$ for which it is able to verify that for every $x \in I$ and for every $n > 0$ such that $f^i(x) \notin \Delta$ for all $i = 0, \ldots, n-1$ the inequality (1) holds true for some $C > 0$ which is also computed explicitly. If the constant λ returned by this algorithm is not positive then this situation is called "failure", because Statement 1 requires $\lambda > 0$. The same explanation applies to the computation of λ_0 for Statement 2.

We emphasize the simplicity of applying our computational procedure (the non-trivial mathematics and computer algorithms can all remain in the background and fully hidden from view for the user) as well as the wide scope of its applicability. Indeed, we have no particular assumptions on f except that it should be C^1 outside Δ. Obvious classes of maps to which we could therefore apply our computations are polynomial maps, including the well known quadratic family, but also maps with discontinuities and/or unbounded derivatives. The maps in Figure 1 are all related to return maps for smooth flows and have all been studied in the context of parameter exclusion arguments [13, 14, 20, 21]. Statements 1 and 2 are verified there analytically in certain quite specific parameter regions and play a crucial role in the arguments. The computational techniques given here could be useful in extending these arguments to other parameter

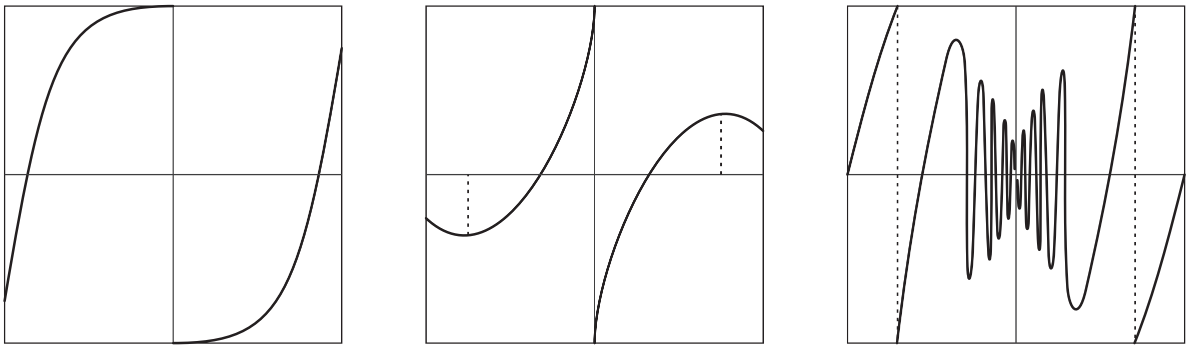

Figure 1. Interval maps with critical points, discontinuities and/or unbounded derivatives.

regions and in obtaining some more explicit estimates for the measure of the good parameter set in a similar way to [12].

In this paper we limit ourselves to some applications to the quadratic family since, as discussed above, applications to other maps are conceptually no more difficult. Thus we compute the constants C and λ for several thousand parameters of the well known quadratic family. We discuss the results at length in Section 5; to give a feeling for the kind of numbers we rigorously obtain, we give here just a small sample.

Theorem 1 *Let $f_a(x) = x^2 - a$ and let $\Delta = (-\delta, \delta)$ denote a neighbourhood of the critical point $c = 0$. Then Statements 1 and 2 hold true for the following parameters a with the following values of δ, C, λ, λ_0:*

a	δ	C	λ	λ_0
1.7	0.1	0.1358	0.3864	0.3864
2	0.1	0.1004	0.6889	0.6886
2	0.01	0.0113	0.5643	0.5643
2	0.001	0.0015	0.2358	0.2358
$[1.99, 2]$	0.1	0.1061	0.6331	0.6330

The values of a and δ in this theorem should be understood as computer representations of the decimal numbers listed in the table. The computed numbers C, λ and λ_0 were slightly higher than the ones listed in the table, and have been rounded down to make sure that the statements hold true for the actual decimal numbers provided.

We have included three different calculations for the parameter value $a = 2$ to highlight the effect of choosing increasingly smaller critical neighbourhoods on the computed values of λ and λ_0. Moreover, we remark that for $a = 2$ we can actually calculate the values of these constants analytically and it turns out that $\lambda = \lambda_0 = \ln 2 = 0.6931\ldots$ *independent of* δ. The computations give quite a good approximation to this value for $\delta = 0.1$ and increasingly bad as δ is chosen smaller. As it will be discussed in Section 5 below, this is just due to working with a fixed level of accuracy (more precisely, a fixed number of partition elements) and the achieved values can be made to approximate $\ln 2$ arbitrarily well by increasing this accuracy. In the last line of the table, we show constants which are valid for all $a \in [1.99, 2]$. It is a remarkable fact that this result was obtained in just one run of the algorithm, which was possible thanks to the use of *interval arithmetic* (see Section 4.1).

We emphasize that these results do not depend in any way on the classical existence proofs for the constants C, λ and thus, in particular, do not involve the verification of any assumptions used in those arguments, but follow instead by direct rigorous computation.

1.3. Overview of the paper and additional remarks

The computations we carry out are fundamentally based on a set of *graph optimization algorithms*. In Section 2 we explain how to encode certain features of the dynamics in a *weighted digraph* and how the problem of finding a lower bound for the expansion exponent can be reformulated in terms of a problem of calculating *mean weights* of certain *paths* in the graph. In Section 3 we discuss in detail the algorithms and techniques used to estimate these quantities. In Section 4 we discuss the more strictly computational part of the procedure and explain how the computer code is programmed in order to carry out the required calculations. Finally, in Section 5 we discuss the result of applying these computations to various parameters in the quadratic family.

The source code of the software introduced in Section 4, as well as detailed results of the computations referred to in Section 5 are available at the website `http://www.pawelpilarczyk.com/unifexp/`.

2. Derivatives and weighted digraphs

The basic idea of our approach is to reduce the problem to one of bounding the mean weights of paths in certain weighted digraphs (directed graphs) related to the map f under consideration. In this section we introduce some notation related to graphs, define what it means to represent a map by a graph, and reformulate the problem of expansion in the language of graphs.

2.1. Weighted digraphs

We start with some definitions and notation. Throughout this paper $G = (V, E, w)$ will denote a *weighted finite digraph*, where V denotes the finite set of *vertices*, $E \subset V \times V$ is the set of *edges*, and $w\colon E \to \mathbb{R}$ is the *weight function.*

The set of paths in G is denoted by $\mathcal{P}(G)$, where a *path* is a nonempty finite sequence of edges

$$\Gamma = (e_1, \dots, e_n) \quad \text{such that} \quad e_j = (v_j^0, v_j^1) \in E \text{ and } v_j^1 = v_{j+1}^0.$$

The vertex v_1^0 is called the *starting vertex* of Γ, and v_n^1 is called the *ending vertex* of Γ. The *length* of the path Γ is n, the number of edges in the sequence, and denoted by $|\Gamma|$. A path Γ' is called a *subpath* of Γ if Γ' is a subsequence of Γ which consists of consecutive edges, that is, $\Gamma' = (e_k, \dots, e_l)$ for some k, l such that $1 \le k \le l \le n$. We say that the subpath Γ' of Γ is *proper* if $|\Gamma'| < |\Gamma|$. The path Γ is called a *cycle* if $v_1^0 = v_{|\Gamma|}^1$. A path is called *simple* if it does not contain a subpath which is a cycle (in particular, no cycle is a simple path). A cycle is called *simple* if it does not contain any proper subpath which is a cycle. The set of cycles, simple paths, and simple cycles in G will be denoted by $\mathcal{C}(G)$, $\mathcal{S}(G)$, and $\mathcal{SC}(G)$, respectively.

It is an elementary fact that every path can be *decomposed* into a simple path and a finite number of simple cycles, as illustrated in Figure 2. More precisely, if $\Gamma = (e_1, \dots, e_n)$ is a path then there exists either a simple path $\Gamma_0 = (e_{j(0,1)}, \dots, e_{j(0,k_0)})$ (if Γ is not a cycle) or $\Gamma_0 = \emptyset$ (if Γ is a cycle) and a finite number of simple cycles $\Gamma_i = (e_{j(i,1)}, \dots, e_{j(i,k_i)})$, where $i = 1, \dots, k$ for some k (possibly $k = 0$), such that each Γ_i is a subsequence of Γ and each index $\{1, \dots, n\}$ appears exactly once in all the sequences of indices $(j(i,l))_{l=1,\dots,k_i}$ over $i = 0, \dots, k$.

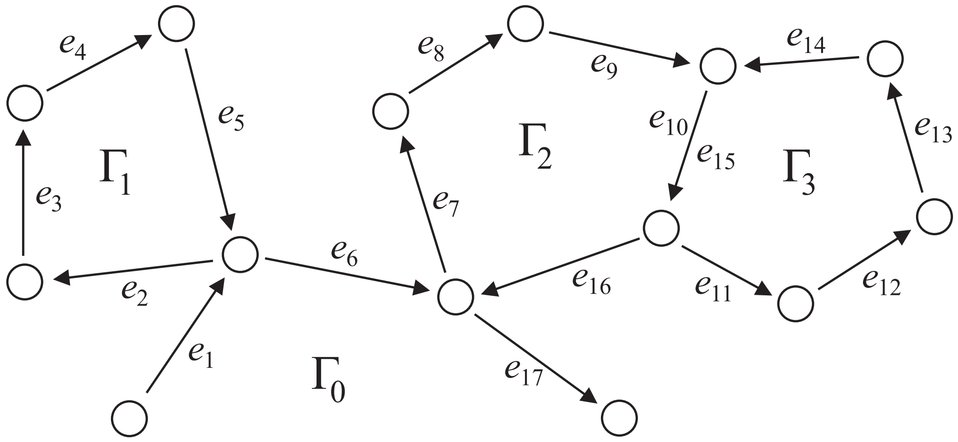

Figure 2. The decomposition of the path $\Gamma = (e_1, \ldots, e_{17})$ into the simple path $\Gamma_0 = (e_1, e_6, e_{17})$ and simple cycles $\Gamma_1 = (e_2, e_3, e_4, e_5)$, $\Gamma_2 = (e_7, e_8, e_9, e_{10}, e_{16})$, and $\Gamma_3 = (e_{11}, e_{12}, e_{13}, e_{14}, e_{15})$. Note that $e_{10} = e_{15}$.

The *weight* and *mean weight* of a path $\Gamma = (e_1, \ldots, e_n) \in \mathcal{P}(G)$ are defined by

$$W(\Gamma) = \sum_{j=1}^{n} w(e_j) \quad \text{and} \quad \overline{W}(\Gamma) = \frac{W(\Gamma)}{n},$$

respectively.

2.2. Graph representation of f

Recall that we are assuming that $f\colon I \setminus \Delta \to I$ is a C^1 map, where $I \subset \mathbb{R}$ is an interval and Δ is a finite union of open subintervals of I having disjoint closures and containing $\mathrm{Crit}(f)$, the set of all critical points of f.

A finite collection of closed intervals $\mathcal{I} = \{I_j \mid j = 1, \ldots, K\}$ is an *f-admissible cover of* $I \setminus \Delta$ if $\mathrm{int}(I_i \cap I_j) = \emptyset$ for $i \neq j$, $I_j \cap \mathrm{Crit}(f) = \emptyset$ for all j, and $I \setminus \Delta \subseteq \bigcup_{j=1}^{K} I_j$.

Given an f-admissible cover $\mathcal{I}$ of $I \setminus \Delta$, we call the weighted digraph $G = (V, E, w)$ a *representation of f on* $I \setminus \Delta$ provided that:

(a) $V = \mathcal{I} \cup \{\mathrm{cl}\Delta\}$;

(b) $\{e = (I_1, I_2) \in \mathcal{I} \times V \mid f(I_1) \cap I_2 \neq \emptyset\} \subset E$

(c) For each $e = (I_1, I_2) \in E$,
$$w(e) \leq \inf\{\log|Df(x)| : x \in I_1 \cap f^{-1}(I_2)\}$$

Representing the map f on $I \setminus \Delta$ by a weighted digraph G allows us to reduce the problem of estimating the minimum accumulated derivatives to computing the weights of certain paths in G. Observe the following straightforward relationship between the weight of a path and the derivative along points whose orbit is described by the path. Given a point $x \in I \setminus \Delta$ and a path $\Gamma = (e_1, \ldots, e_n) \in \mathcal{P}(G)$ such that $e_j = (I_{j-1}, I_j)$ and $f^j(x) \in I_j$ for all $j = 0, \ldots, n$, we have

$$\log|Df^n(x)| = \sum_{j=0}^{n-1} \log|Df(f^j(x))| \geq W(\Gamma). \tag{3}$$

Throughout this paper, we will make use of the following quantity. The *minimum mean weight over all cycles in $G = (V, E, w)$* is

$$\begin{aligned}\mu(G) &:= \begin{cases} \min\{\overline{W}(\Gamma) : \Gamma \in \mathcal{C}(G)\} & \text{if } \mathcal{C}(G) \neq \emptyset \\ +\infty & \text{if } \mathcal{C}(G) = \emptyset \end{cases} \\ &= \begin{cases} \min\{\overline{W}(\Gamma) : \Gamma \in \mathcal{SC}(G)\} & \text{if } \mathcal{C}(G) \neq \emptyset \\ +\infty & \text{if } \mathcal{C}(G) = \emptyset \end{cases}\end{aligned}$$

Note that since G is a finite graph, $\mathcal{SC}(G)$ is a finite set, and therefore, the minimum is attained if $\mathcal{C}(G) \neq \emptyset$. For our studies, $\mu(G) < \infty$ for any graph representation of the map f under consideration.

We say that a path $\Gamma = (e_1, \ldots, e_n)$, where $e_i = (I_i^0, I_i^1)$, *begins in* Δ or *ends in* Δ if $f(\Delta) \cap I_1^0 \neq \emptyset$ or $I_n^1 = \mathrm{cl}\Delta$, respectively. Let $\mathcal{P}_\Delta(G)$ denote the set of paths that begin or end in Δ. Define

$$\mu_\Delta(G) := \inf\{\overline{W}(\Gamma) : \Gamma \in \mathcal{P}_\Delta(G)\}.$$

2.3. Estimating derivatives using graphs

For given $\lambda \in \mathbb{R}$ and $G = (V, E, w)$, define

$$\kappa(\lambda, G) := \exp\Big(\min\{W(\Gamma) - |\Gamma|\lambda : \Gamma \in \mathcal{S}(G)\}\Big).$$

Proposition 1 *Let G be a weighted finite digraph that is a representation of f on $I \setminus \Delta$. If $\lambda \leq \mu(G)$ and $C \leq \kappa(\lambda, G)$, then*

$$|Df^n(x)| \geq Ce^{\lambda n}$$

for any $n > 0$ and $x \in I$ such that $f^i(x) \notin \Delta$ for all $i = 0, \ldots, n-1$.

Proof. The assumptions on C and λ can be restated as follows:

$$W(\Gamma_s) - |\Gamma_s|\lambda \geq \log(C) \quad \text{for all } \Gamma_s \in \mathcal{S}(G)$$

and

$$W(\Gamma_c) - |\Gamma_c|\lambda \geq 0 \quad \text{for all } \Gamma_c \in \mathcal{C}(G).$$

Given a path $\Gamma \in \mathcal{P}(G)$, consider the decomposition of Γ into $\Gamma_s \in \mathcal{S}(G) \cup \{\emptyset\}$ and $\Gamma_c^i \in \mathcal{SC}(G)$, $i = 1, \ldots, k$ for some $k \geq 0$, as explained in Section 2.1. Obviously, $|\Gamma| = |\Gamma_s| + \sum_{i=1}^k |\Gamma_c^i|$, where $|\emptyset| := 0$ and $W(\emptyset) := 0$. Therefore,

$$\begin{aligned} W(\Gamma) = W(\Gamma_s) + \textstyle\sum_{i=1}^k W(\Gamma_c^i) = (W(\Gamma_s) - |\Gamma_s|\lambda) + \\ + \textstyle\sum_{i=1}^k (W(\Gamma_c^i) - |\Gamma_c^i|\lambda) + |\Gamma|\lambda \geq \log(C) + |\Gamma|\lambda. \end{aligned} \tag{4}$$

Finally, consider a point $x \in I \setminus \Delta$ such that $f^i(x) \notin \Delta$ for all $i = 1, \ldots, n-1$. Then there exists a path $\Gamma = (e_1, \ldots, e_n) \in \mathcal{P}(G)$ where $e_j = (I_{j-1}, I_j)$ such that $f^j(x) \in I_j$. The result follows from (3) and (4). □

Proposition 2 *Let G be a weighted digraph that is a representation of f on $I \setminus \Delta$. If $\lambda_0 \le \mu_\Delta(G)$, then*

$$|Df^n(x)| \ge e^{\lambda_0 n}$$

for any $n > 0$ and $x \in I$ such that $f^i(x) \notin \Delta$ for all $i = 0, \dots, n-1$, and $x \in f(\Delta)$ or $f^n(x) \in \Delta$.

Proof. Given $x \in I \setminus \Delta$ such that $f^i(x) \notin \Delta$ for all $i = 0, \dots, n-1$, and $x \in f(\Delta)$ or $f^n(x) \in \Delta$, there exists a path $\Gamma = (e_1, \dots, e_n) \in \mathcal{P}_\Delta$ where $e_j = (I_{j-1}, I_j)$ such that $f^j(x) \in I_j$. By (3),

$$\log |Df^n(x)| \ge W(\Gamma) = \overline{W}(\Gamma)\,|\Gamma| \ge \mu_\Delta(G)\,|\Gamma| \ge \lambda_0 |\Gamma| = \lambda_0 n,$$

which proves the Proposition. □

2.4. Refining the representation of f

Let $G = (V, E, w)$ and $G' = (V', E', w')$ be representations of f on $I \setminus \Delta$. We say that G' is a *refinement of* G if there exists a function $\pi \colon V' \to V$ such that $J \subset \pi(J)$ for all $J \in V'$, and for every $e' = (I'_1, I'_2) \in E'$ we have $e := (\pi(I'_1), \pi(I'_2)) \in E$ and $w'(e') \ge w(e)$.

Proposition 3 *Let G and G' be representations of f on $I \setminus \Delta$. If G' is a refinement of G then $\mu(G') \ge \mu(G)$.*

Proof. For each cycle $\Gamma' = (e'_1, e'_2, \dots, e'_n)$ in G', the path $\pi(\Gamma') := (\pi(e'_1), \pi(e'_2), \dots, \pi(e'_n))$ is a cycle in G with $\overline{W}(\Gamma') \ge \overline{W}(\pi(\Gamma'))$. Therefore,

$$\begin{aligned} \mu(G') = \min\{\overline{W}(\Gamma') : \Gamma' \in \mathcal{C}(G')\} &\ge \min\{\overline{W}(\pi(\Gamma')) : \Gamma' \in \mathcal{C}(G')\} \ge \\ &\ge \min\{\overline{W}(\Gamma) : \Gamma \in \mathcal{C}(G)\} = \mu(G), \end{aligned}$$

which ends the proof. □

It is reasonable to expect that if one applies the procedure described in Section 4 for the construction of a representation G of f on $I \setminus \Delta$ with an f-admissible cover $\mathcal{I}$, and another representation G' using a refinement $\mathcal{I}'$ of $\mathcal{I}$, then G' should be a refinement of G, and Proposition 3 says that in that case the expansion exponent computed for G' will be better than the one computed for G.

3. Mean weight algorithms

The results of Section 2.3 reduce the problem of determining expansion estimates to the computation of mean weights of certain paths. More precisely, given a weighted digraph $G = (V, E, w)$ which is a representation of some f on $I \setminus \Delta$ (as defined in Section 2.2), we need to be able to compute $\mu(G)$, $\mu_\Delta(G)$, and $\kappa(\lambda, G)$. In this section, we describe the algorithms that we use to determine these quantities. We also provide a measure of the complexity of each algorithm by giving the associated *running time.*

The running time measures the order of the number of required primitive operations (such as addition and multiplication) as a function of the size of the input, here given by $|V|$, the size of the vertex set, and $|E|$, the size of the edge set. We use the notation $O(\Psi(|V|, |E|))$ to indicate that there exist constants $c, n_0 > 0$ such that for any graph $G = (V, E, w)$ for which $|V|, |E| \geq n_0$, the number of operations $\Phi(G)$ of the algorithm applied to the graph G satisfies the inequality $\Phi(G) \leq c\Psi(|V|, |E|)$. This gives an asymptotic upper bound for the worst case running time. For a more detailed explanation of this notation and of running time in general, the reader is referred to [6, §3.1]. For our computations, $|V|$ is fixed using a computational parameter K specifying a partition size, and $|E|$ depends on the size of V and on the expansion of the map f relative to the partition giving V. This will be described in more detail in the next section.

3.1. Computation of $\mu(G)$

The constant $\mu(G)$ is obtained by a straightforward application of *Karp's Algorithm* [10] which computes the minimum mean weight of any cycle in a weighted digraph. Its running time is $O(\max\{|V|\,|E|, |V|^2\})$.

3.2. Computation of $\kappa(\lambda, G)$

The *Floyd-Warshall Algorithm* [6, §25.2] and *Johnson's Algorithm* [6, §25.3] both find the minimum weights of paths between all pairs of vertices in a weighted digraph, and thus, applied to the graph $G' = (V, E, w')$ where $w'(e) = w(e) - \lambda$, provide a number that does not exceed $\kappa(\lambda, G)$. They have different running times, $O(|V|^3)$ for Floyd-Warshall and $O(|V|^2 \log |V| + |V|\,|E|)$ for Johnson and thus may be more or less efficient depending on the specific situation. Johnson's Algorithm is generally more efficient for sparse graphs, which is the case in our applications. Note that both these algorithms require that G does not have any cycle whose weight is negative.

3.3. Computation of $\mu_\Delta(G)$

The specific computation required for $\mu_\Delta(G)$ is less standard. Following the idea of Karp's Algorithm, we formulate an algorithm which computes the minimum mean weight $\mu_1(G, V_0)$ of paths of length up to $|V| - 1$ whose *starting* vertex belongs to the given set $V_0 \subset V$. To compute the minimum mean weight $\mu_2(G, V_0)$ of paths of length up to $|V| - 1$ whose *ending* vertex belongs to V_0, one can apply this algorithm to the transposed graph G^T which is obtained from G by reversing the direction of edges.

First of all, we compute the functions $F_k(v)$ for each $v \in V$ and $k = 1, \ldots, |V| - 1$, which are defined as the minimum weight of any path of length k whose starting vertex belongs to V_0 and ending vertex equals v, or we set $F_k(v) := \infty$ if no such path exists. These functions are computed using the recursive formula

$$F_k(v) = \min_{(u,v)\in E} (F_{k-1}(u) + w(u, v))$$

for $k = 1, \dots, |V|$, with the initial condition $F_0(v) = \infty$ for $v \in V \setminus V_0$, and $F_0(v) = 0$ for $v \in V_0$. More precisely, we have the following

Algorithm 1

```
for all $v \in V$, $k = 0, \dots, |V| - 1$ do $F_k(v) := \infty$;
for all $v \in V_0$ do $F_0(v) := 0$;
for $k := 1$ to $|V| - 1$ do
    for all $(u, v) \in E$ do
        $F_k(v) := \min(F_k(v),\ F_{k-1}(u) + w(u, v))$;
return $\min\{F_k(v)/k$ such that $v \in V$ and $k = 1, \dots, |V| - 1\}$.
```

As it can be clearly seen from counting the arguments of the "for" loops, as well as the number of elements in the set whose minimum is computed in the last line, the running time of Algorithm 1 is $O(\max\{|V|\,|E|, |V|^2\})$.

We use this algorithm to compute $\mu_1(G, V_0)$, where $V_0 \subset \mathcal{I}$ is some set of intervals whose union covers $f(\mathrm{cl}\Delta)$, and to compute $\mu_2(G, \{\mathrm{cl}\Delta\})$. Together with $\mu(G)$, these two numbers suffice to find a lower estimate for $\mu_\Delta(G)$, as it is claimed in the following

Proposition 4 *Let $V_0 \subset \mathcal{I}$ be as defined above. Then*

$$\min\{\mu(G), \mu_1(G, V_0), \mu_2(G, \{\mathrm{cl}\Delta\})\} \leq \mu_\Delta(G). \tag{5}$$

Proof. Let Γ be any path in $\mathcal{P}(G)$ which begins in Δ or ends in Δ. Since the union of V_0 covers $\mathrm{cl}\Delta$, the starting vertex of Γ belongs to V_0 (in the former case) or its ending vertex equals $\mathrm{cl}\Delta$ (in the latter case). Consider the decomposition of Γ into a simple path Γ_s (possibly $\Gamma_s = \emptyset$) and simple cycles Γ_c^i, $i = 1, \dots, k$ for some $k \geq 0$. If $\Gamma_s \neq \emptyset$ then the starting vertex of Γ_s belongs to V_0 or the ending vertex of Γ_s equals $\mathrm{cl}\Delta$. Moreover, $|\Gamma_s| \leq |V| - 1$, and thus

$$\overline{W}(\Gamma_s) \geq \min\{\mu_1(G, V_0), \mu_2(G, \{\mathrm{cl}\Delta\})\}.$$

where $\overline{W}(\Gamma_s) := +\infty$ if $\Gamma_s = \emptyset$. As for the simple cycles, $\overline{W}(\Gamma_c^i) \geq \mu(G)$. Obviously,

$$\overline{W}(\Gamma) \geq \min\{\overline{W}(\Gamma_s), \overline{W}(\Gamma_c^i), i = 1, \dots, k\}.$$

This proves (5) and thus Proposition 4. □

4. Computations

For the sake of clarity, in this section we provide an outline for the principle procedures and tools employed in our computations. We purposely provide an outline as opposed to specific details for the following reason: While we believe the strategy to be problem independent, the optimal numerical methods do depend on the particular family of nonlinear functions being studied. In particular, we cover the following topics:

(1) the use of interval arithmetic to ensure the mathematical validity of our computations;

(2) a procedure for defining an admissible cover $\mathcal{I}$ of $I \setminus \Delta$;

(3) a procedure for constructing a weighted directed graph;

(4) the computation of the constants λ, C, and λ_0, and a proof that Statements 1 and 2 are satisfied for these constants;

(5) an interface for running series of computations in a concurrent way on multiple machines.

The software developed for the computations described in this section has been programmed in the object-oriented C++ programming language for optimal performance and expandability. We outline its features and capabilities in the subsections below.

4.1. Interval arithmetic

The software does rigorous computations using *interval arithmetic* (see [18] for a comprehensive introduction). Since computers can only store limited precision rational numbers, the idea of interval arithmetic is to represent any real number by an interval with the computer-representable rational ends that contains the real number in question. We then replace arithmetic operations on real numbers with the corresponding operations on intervals whose result is an interval which contains all possible results of the operation on any arguments from the argument intervals, e.g.,

$$\{x + y : x \in [a, b], y \in [c, d]\} \subset [a, b] \langle + \rangle [c, d].$$

Interval arithmetic is used to compute the images of intervals by the map f, as well as in the graph algorithms to compute rigorous lower estimates for the mean weights of paths. Note that in the graph algorithms, usually only one end of the interval needs to be computed, which simplifies the computations to using the arithmetic operators with controlled direction of rounding (upwards or downwards). Also, the comparison of intervals with the "<" and ">" operators must be done by comparing their lower or upper ends only, depending on whether we compute the minimum or the maximum of weights.

4.2. Defining an admissible cover

Recall that $I \subset \mathbb{R}$ is an interval and Δ is defined as a finite union of open subintervals of I. Assume that the values of the endpoints of Δ lie in the set of representable numbers used in the above mentioned interval arithmetic. For simplicity, in the software which accompanies this paper, Δ is defined as $\bigcup_{i=1}^{k}(c_i - \delta, c_i + \delta)$, where $\mathrm{Crit}(f) = \{c_1, \dots, c_k\}$ and $\delta > 0$, but this form of Δ is not required by the theory. Moreover, the actual endpoints of the intervals which define Δ taken for the computations are rounded to some possibly close representable numbers.

In principle, an admissible cover $\mathcal{I}$ can be chosen to be any finite cover of $\mathcal{I} \setminus \Delta$ where the endpoints of the intervals are representable numbers. In practice, the simplest way to define $\mathcal{I}$ is to fix some number K of elements and partition $I \setminus \Delta$ into K

essentially equal subintervals. We use this partition for the computations described in Section 5. However, we note that for certain particular maps estimates can be significantly improved by having a non-uniform partition.

4.3. Constructing a weighted digraph

We now describe how to construct a graph representation $G = (V, E, w)$ for a parameterized family of maps, $f\colon I \times \Omega \to I$, where Ω is a closed interval and $f \in C^1((I \setminus \Delta) \times \Omega)$ with $\mathrm{Crit}(f_a) \subset \Delta$ for all $a \in \Omega$. Note that here we assume that Δ is independent of the parameter $a \in \Omega$. In practice, this is a reasonable assumption if we choose the parameter interval small enough so that the critical points of f do not go out of the fixed Δ. For a larger parameter interval, we may decompose it into small enough parameter subintervals which satisfy this assumption, and repeat the computations for each of these parameter subintervals.

4.3.1. The vertices We suppose an f-admissible cover $\mathcal{I} := \{I_1, \ldots, I_k\}$ of $I \setminus \Delta$ has been fixed. These intervals together with $\mathrm{cl}\Delta$ serve as the vertices in G, that is, we define

$$V := \mathcal{I} \cup \{\mathrm{cl}\Delta\}.$$

4.3.2. The edges For each $I_i \in \mathcal{I}$ we use a combination of analytic bounds and interval arithmetic to compute an interval denoted by $F(I_i)$ which is an outer estimate for $f_a(I_i)$ for all $a \in \Omega$, that is,

$$f(I_i, \Omega) \subset F(I_i). \tag{6}$$

Define

$$E := \{(I_i, J) \in \mathcal{I} \times V \mid F(I_i) \cap J \neq \emptyset\}.$$

4.3.3. The weights For $e = (I_i, J) \in E$, choose a closed interval $L(e) \subset I_i$ such that $\{x \in I_i \mid f(x, \Omega) \cap J \neq \emptyset\} \subset L(e)$ (note that $L(e) = I_i$ is an admissible choice). Using a combination of analytic bounds and interval arithmetic, we compute a representable number $b(L(e))$ such that

$$b(L(e)) \leq \min \{ \log |D_x f(x, a)| : x \in L(e), a \in \Omega \} \tag{7}$$

Then we define

$$w(e) := b(L(e)).$$

4.4. Computing the Constants

Having constructed $G = (V, E, w)$, we run Karp's Algorithm (Section 3.1) in interval arithmetic to obtain a lower bound $\mu_{\approx}$ for $\mu(G)$. Define

$$\lambda := \mu_{\approx}. \tag{8}$$

We now construct a new weighted digraph $G' = (V, E, w')$ where $w'(e)$ is computed as the largest representable number not exceeding $w(e)-\lambda$. Define $\log C$ to be the output of running either the Floyd-Warshall Algorithm or Johnson's Algorithm (Section 3.2) on G' in interval arithmetic.

Define λ_0 to be the minimum of the number λ and the numbers $\mu_1(G, V_0)$ and $\mu_2(G, \{\mathrm{cl}\Delta\})$ computed with Algorithm 1 applied to $G = (V, E, w)$ in interval arithmetic.

Theorem 2 *Let $a \in \Omega$, and define $f_a := f(\cdot, a)\colon I \setminus \Delta \to I$. Then Statements 1 and 2 are satisfied for f_a using the constants λ, C, and λ_0.*

Proof. As indicated in Section 4.2, we have constructed an admissible cover of $I \setminus \Delta$ for f_a. Following the procedures of Sections 4.3.1, 4.3.2, and 4.3.3, we have computed $G = (V, E, w)$ which is a representation of f on $I \setminus \Delta$; indeed, the inclusion (6) implies (b) in the definition of a representation, and the inequality (7) implies (c) in that definition.

Since $\lambda \leq \mu(G)$ and $C \leq \kappa(\lambda, G)$, Statement 1 follows from Proposition 1. Similarly, $\lambda_0 \leq \mu_\Delta(G)$, and hence Statement 2 follows from Proposition 2. □

Remark. In practice, the numbers $w'(e)$ computed for $e \in E$ by the computer as $w(e) - \mu_\approx$ with downwards rounding to the nearest representable number might be so small that negative-weight cycles might occur in the graph G' (especially since the weights of paths are also computed as sums of weights of edges with the rounding direction set to "downwards"), which causes both the Floyd-Warshall Algorithm and Johnson's Algorithm to fail. Therefore, in order to ensure successful computation of C, instead of taking the constant λ defined by (8), λ is defined as $\mu_\approx$ decreased by 0.000000001% of its absolute value, that is, $\lambda := \mu_\approx - 10^{-11}|\mu_\approx|$. This tiny decrease turns out to be large enough for our computations to alleviate the problem of spurious negative cycles, but it cannot be ruled out that for some other cases a slightly larger decrease of λ might be necessary.

4.5. Running the software on multiple machines

In order to accomodate the need for running the computations for many parameter intervals $\Omega \ni a$ for the family f_a, the software uses an elementary network communication interface based on the TCP/IP sockets in order to allow one to spread the computations over several machines (a local network or a computer cluster, for example). A centralized model of concurrent computations is used, with one process designated as a *coordinator* who iterates the interval of parameters, and all the other processes designated as *workers* who do the computations for each specific value or subinterval of the parameter a obtained from the coordinator. The task of the coordinator is also to store the results of computations obtained from workers.

5. Applications and Discussion

As an example application of the algorithms introduced in this paper, we consider the well-known family of unimodal maps $f_a \colon I \to I$ given by

$$f_a(x) = x^2 - a,$$

where $I = [-2, 2]$, $a \in [1.7, 2]$, and $\Delta = (-\delta, \delta)$ for some small $\delta > 0$. First we fix the parameter value $a = 2$ and carry out several computations and comparisons by varying δ, the size of Δ, and K, the number of subdivisions of $I \setminus \Delta$. Then we fix δ and K and carry out the computations for a few thousand equally spaced (up to the rounding precision) parameter intervals which fill the parameter range under consideration.

Note that for the map f_a, the interval $L(I_i, J)$ in (7) can be taken as $f_a^{-1}(J) \cap I_i$ easily computed in interval arithmetic, which gives a nearly optimal weight function w.

We would like to state at this point that applying our method to other maps as described in Section 1.2 is no more difficult than to the quadratic family. Therefore, we believe that discussing the well known family of maps would allow the reader to confront our results with the results established in the literature, and also to expose both the power and the weaknesses of our approach. The variety of dynamical behaviours observed for the interval of parameters which we consider (from a stable periodic orbit to a strange attractor) allows for a comprehensive test of our method.

5.1. The parameter value $a = 2$

The map f_a for the parameter value $a = 2$ is a particular example of a Chebyshev polynomial and as such is smoothly conjugate to a piecewise affine expanding map (see, for example, [12]). This property was exploited by Ulam and von Neumann [26] proving that f_2 exhibits stochastic-like behaviour; indeed, this was the first case of such dynamical behaviour being proven in the quadratic family. The special conjugacy can also be used to prove Statements 1 and 2 analytically with $\lambda = \lambda_0 = \ln 2 = 0.6931\ldots$ for any $\delta > 0$. We now check how close our methods get to this expected value.

5.1.1. Computation of λ for various sizes of Δ Figure 3 shows the computed values of λ for a few thousand different sizes of the critical neighbourhood $\Delta = (-\delta, \delta)$ and a uniform admissible cover $\mathcal{I}$ constructed by dividing $I \setminus \Delta$ into K essentially equal subintervals with $K = 5000$. The actual lengths of the subintervals may slightly differ because of rounding their endpoints to representable numbers.

The estimated value of λ decreases monotonically with decreasing δ (although we emphasize that the horizontal axis is a logarithmic scale) even though the theoretical value of λ does not. This is naturally to be expected as both the number and size of the partition elements giving rise to the graph remain constant. Increasing the "resolution" of the calculations by increasing the number of partition elements, and thus also descreasing their size, gives improved estimates as shown below.

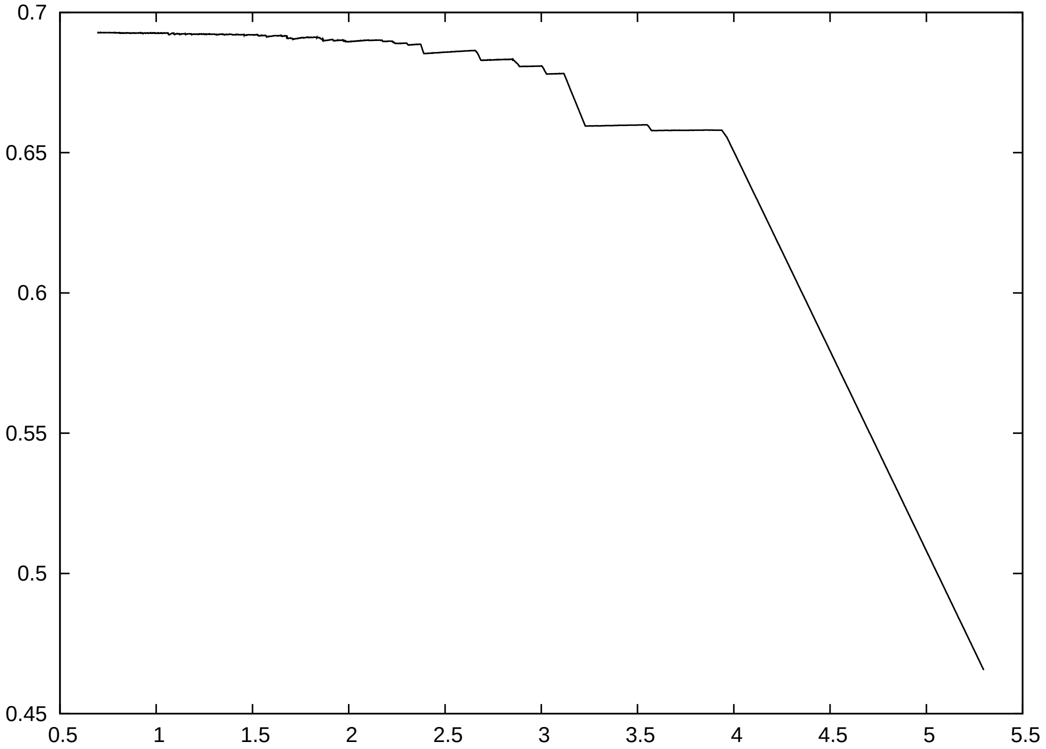

Figure 3. Computed values of λ for the map f_a with $a = 2$, $K = 5000$, and various values of $\delta > 0$. The horizontal scale represents $-\log\delta$.

5.1.2. Computation of λ for various values of K We now consider a fixed critical neighbourhood $\Delta = (-\delta, \delta)$ with $\delta = 0.01$, and compute λ with a uniform admissible cover $\mathcal{I}$ with K elements for different values of K (see Figure 4).

As it should be expected, the results improve with larger K. Values of K between 4,000 and 6,000 already give relatively good estimates, and choosing K near 8,000 provides values of λ already close to the true value of $\ln 2$.

For a fixed value of K, the computed values of λ can be improved significantly by using a non-uniform partition which has more and smaller elements of $\mathcal{I}$ in the *large derivative* regions, i.e., close to the extreme points of the domain I, and which has relatively larger and fewer elements near the critical region Δ. Figure 5 shows a comparison between the values of λ computed using these two partitions.

Although at first sight this may appear counter-intuitive, it is easy to see that it is actually natural to have smaller partition elements where the derivative is large and not where the derivative is small. Indeed, this approximates better the real dynamics and thus picks up more of the expansion which in the real map is essentially dominated by the time that orbits spend in the regions of large derivative.

Another strategy for defining a non-uniform partition might be to have smaller intervals close to a fixed number of initial iterates of the critical point. This strategy is justified by the expectation that the derivative should gain exponential growth along the critical orbit, and keeping intervals small prevents from losing it.

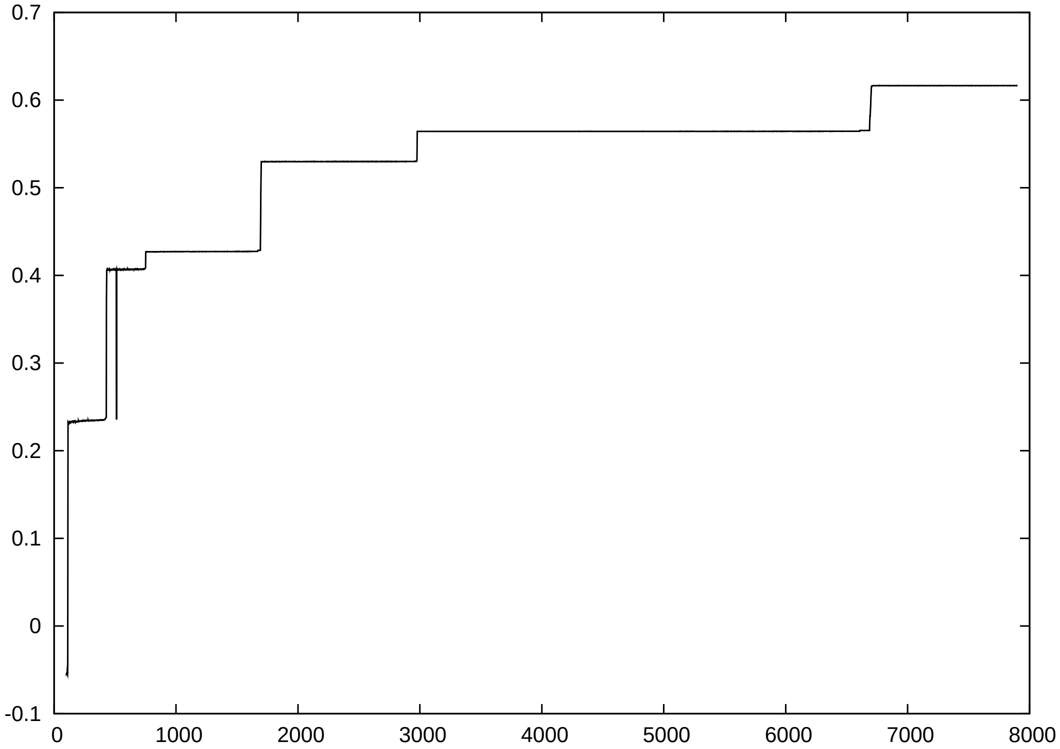

Figure 4. Computed values of λ for the map f_a with $a = 2.0$, $\delta = 0.01$, and several different values of K.

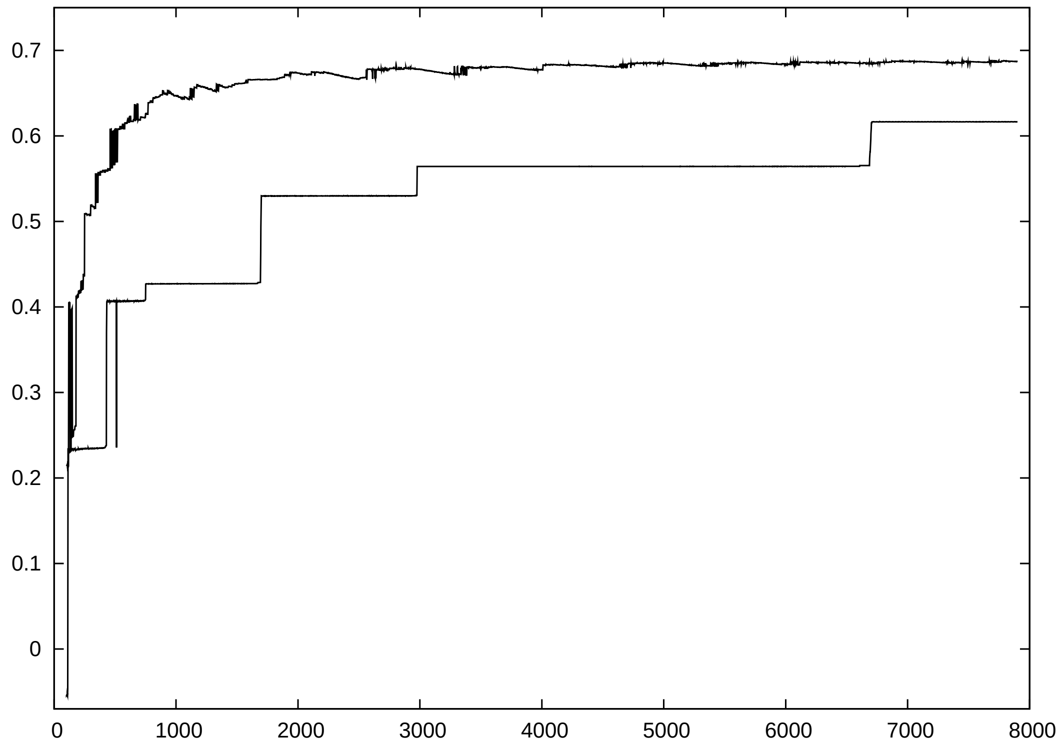

Figure 5. Computed values of λ with the uniform partition (lower curve) and a non-uniform partition (upper curve).

The question remains, however, how to determine the proportions between the sizes of intervals in the regions where they should be small and where they should be wide. Our experiments with the quadratic map at different values of a indicate that there is no simple answer to this question, and finding an optimal strategy appears to be problem-specific.

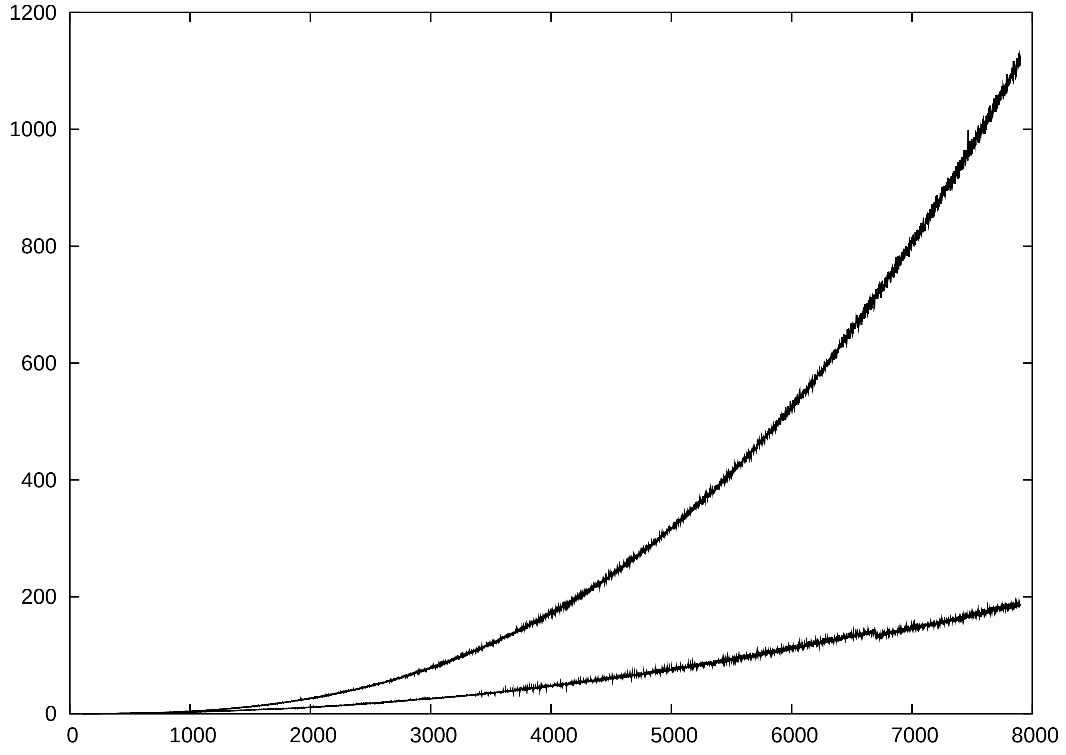

Figure 6. Time (in seconds) required to compute the constant C for different values of K using Johnson's Algorithm (lower curve) and the Floyd-Warshall Algorithm (upper curve).

5.1.3. The time of computation of the constant C Figure 6 shows a graph of the computer processor time required to compute the constant C using Johnson's Algorithm and Floyd-Warshall Algorithm. For large values of K, Johnson's Algorithm performs significantly better due to its better asymptotic time complexity. Although it cannot be seen from the picture, the opposite is true for small values of K, such as $K = 400$.

5.2. Global parameter estimates

We now want to take advantage of the power of multiple computers to compute lower bounds for λ and λ_0 for several thousand parameter intervals of essentially the same length (up to rounding their endpoints to representable numbers) which fill the parameter interval $[1.7, 2]$. As discussed above, the estimates improve with the number K of intervals in $\mathcal{I}$, and for the purpose of this calculation we fix the value $K = 5000$.

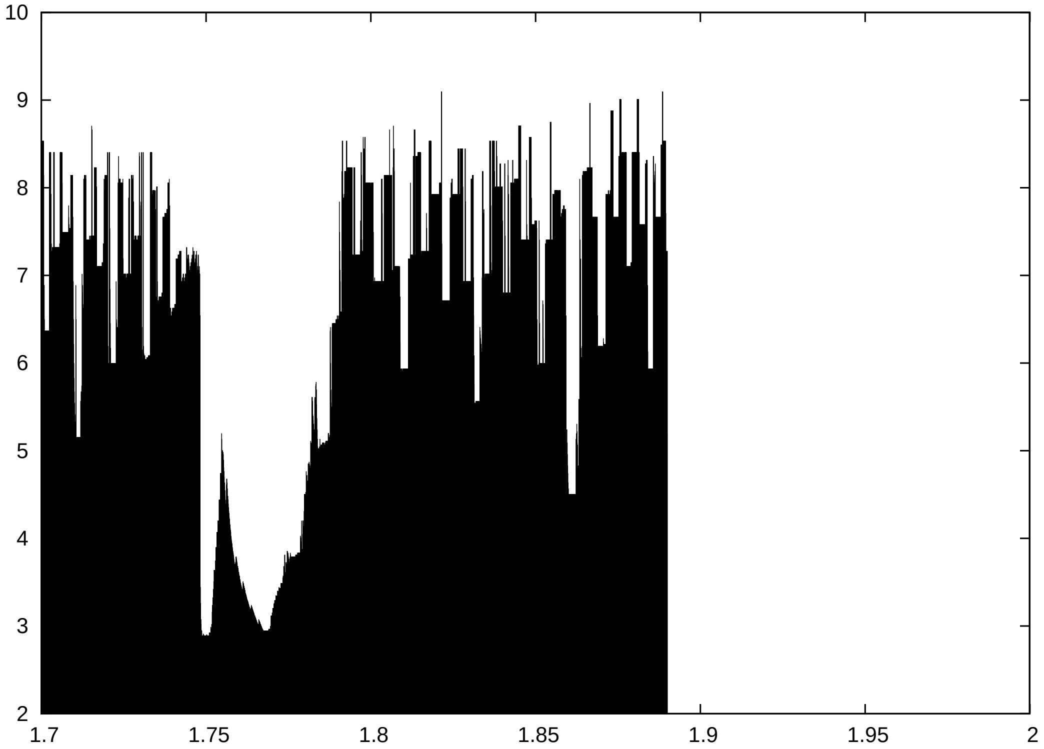

Figure 7. The values of $-\log\delta$ as a function of a, where $\delta > 0$ is the minimum radius of Δ for which the value λ computed for f_a on $I \setminus \Delta$ with $K = 5000$ is positive.

5.2.1. Choosing δ The first thing we need to do when dealing with such a large range of parameter values is to think about the appropriate choice of δ. For general parameter values other than $a = 2$, it is no longer true that the value of λ is independent of δ. Indeed, for an open and dense set of parameter values with periodic attractors, λ will be positive only if δ is sufficiently large. Therefore, we first use our algorithms to obtain some estimate $\delta(a)$ of the minimum value of δ which gives a positive value of λ. These values of $\delta = \delta(a)$ are thus the natural choices for the computation of the exponent $\lambda = \lambda(a)$. Figure 7 shows the resulting values of $-\log\delta(a)$ computed with the bisection method applied until the accuracy in $-\log\delta(a)$ of about 0.06 has been reached (the actual accuracy is determined heuristically in the algorithm, based on the initial guess for $\delta(a)$).

Remark. It is interesting to note that, although we have not checked this on a systematic basis, the variation of λ with respect to δ in the cases in which $\delta(a)$ is positive, i.e., for parameters which have periodic attractors, seems to be a function which has a jump-type discontinuity at $\delta(a)$. For many parameter values, the computations give $\lambda < 0$ for $\delta < \delta(a)$ and λ *strictly positive* (sometimes quite large) for $\delta \geq \delta(a)$. In fact, it does not appear that the estimate for λ improves significantly by taking δ moderately larger than $\delta(a)$. Intuitively, one can speculate that we have the following picture: For a given parameter a, the map f_a may have a periodic attractor. By general theory, the immediate basin of attraction of this periodic attractor will contain a neighbourhood of the critical point. However, any orbit which lies in the complement of this immediate basin is hyperbolic repelling with some multiplier which may be uniformly strictly bounded away from zero.

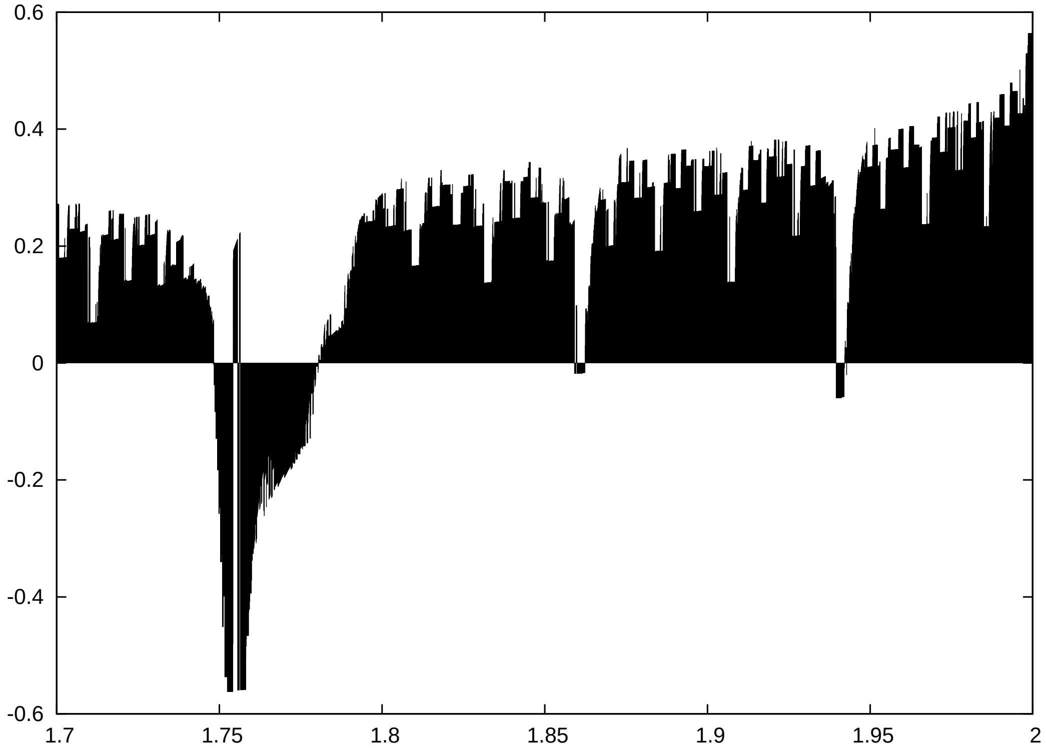

Figure 8. Computed values of λ for $a \in [1.7, 2.0]$, $K = 5000$ and $\delta = 0.01$.

5.2.2. Computing λ for different parameter values Finally, Figures 8 and 9 show values of λ computed for Statement 1 for several thousand parameter intervals which fill $[1.7, 2.0]$, and a fixed choice of $K = 5000$. The analogous graphs for the values of λ_0 computed for Statement 2 are very similar, so we do not plot them in separate pictures (they can be found at the website whose address was indicated in the Introduction).

The graph in Figure 8 corresponds to a fixed choice $\delta = 0.01$. In this case, as discussed above, λ can be negative and this indicates failure in proving the exponential expansion for the corresponding value of a. Figure 9 reflects choosing $\delta = \delta(a)$ as above. Notice that in general this gives significantly lower values for λ. This should be expected, as in most cases we have $\delta(a) < 0.01$ and therefore we are computing λ for a much smaller critical neighbourhood. However, by increasing the number of partition elements (for better results, combined with using a non-uniform partition, as discussed in Section 5.1.2), we might expect to be able to recover the (positive) values of λ from the graph depicted in Figure 8 even for the choices of δ as in Figure 9, although we have not done such an experiment.

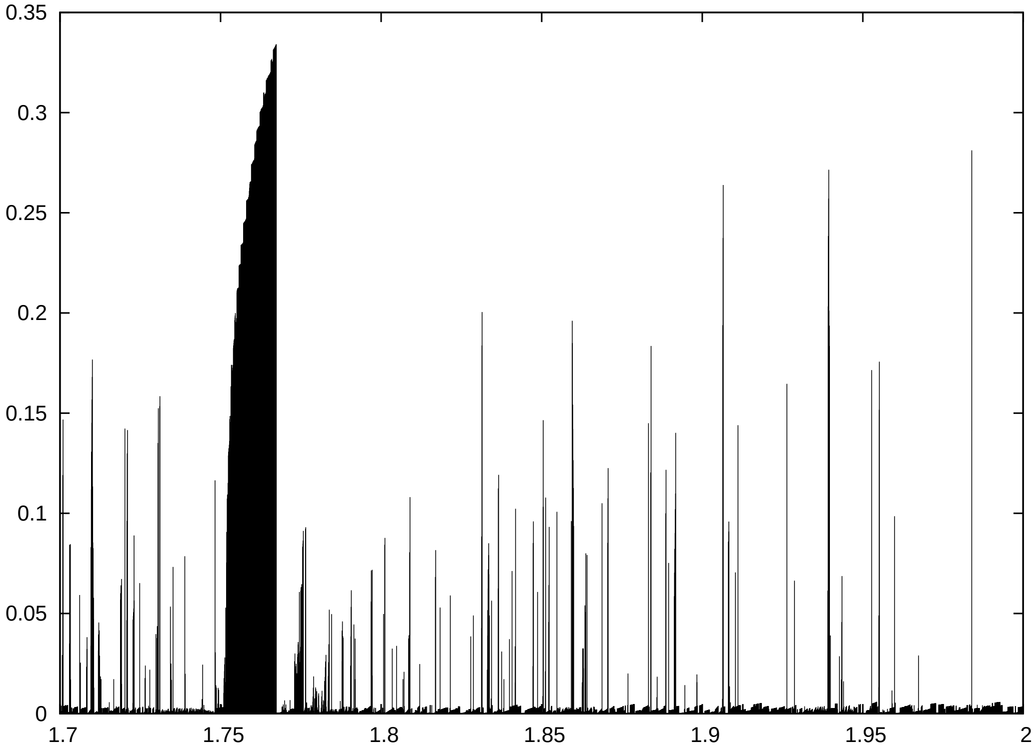

Figure 9. Computed values of λ for $a \in [1.7, 2.0]$, $K = 5000$ and $\delta = \delta(a)$.

5.2.3. Comparison with non-rigorous approach It is interesting to compare the results and methods described above with the more standard, albeit non-rigorous, Lyapunov exponent calculations. These generally consist of picking some more or less arbitrary initial condition x and directly computing the value of

$$\frac{1}{n}\log|Df^n(x)| = \frac{1}{n}\sum_{i=0}^{n-1}\log|Df(f^i(x))| \tag{9}$$

for some large value of n chosen in such a way that the final iterate is close to the initial point and that estimates obtained using some substantially different large values of n are in agreement, e.g., for $n_1 < n_2 < n_3$ with $n_1 \approx 0.5\,n_3$ and $n_2 \approx 0.75\,n_3$. The Lyapunov exponent $\lambda(x)$ at the point x is variously defined as the *liminf* or *limsup*, or *limit* if it exists, of (9) as $n \to \infty$. It should be noted straight away that by virtue of its definition as a limit, the value of $\lambda(x)$ cannot generally be rigorously estimated just by direct calculation of (9) no matter how large n is chosen, without additional theoretical considerations (however, see [7, 23, 24] for interesting discussions and approaches to this issue). Also, in principle the value of $\lambda(x)$ depends on x and thus specific choice of initial condition can play a role in the value that is obtained. Nevertheless, it turns out empirically that such calculations can be quite useful and give some reasonable idea of the actual values to be expected, at least in part because it turns out that in many situation the value of $\lambda(x)$ is actually constant Lebesgue-almost everywhere. Figure 10 gives the graph of the Lyapunov exponents calculated via (9) for various parameters in the quadratic family.

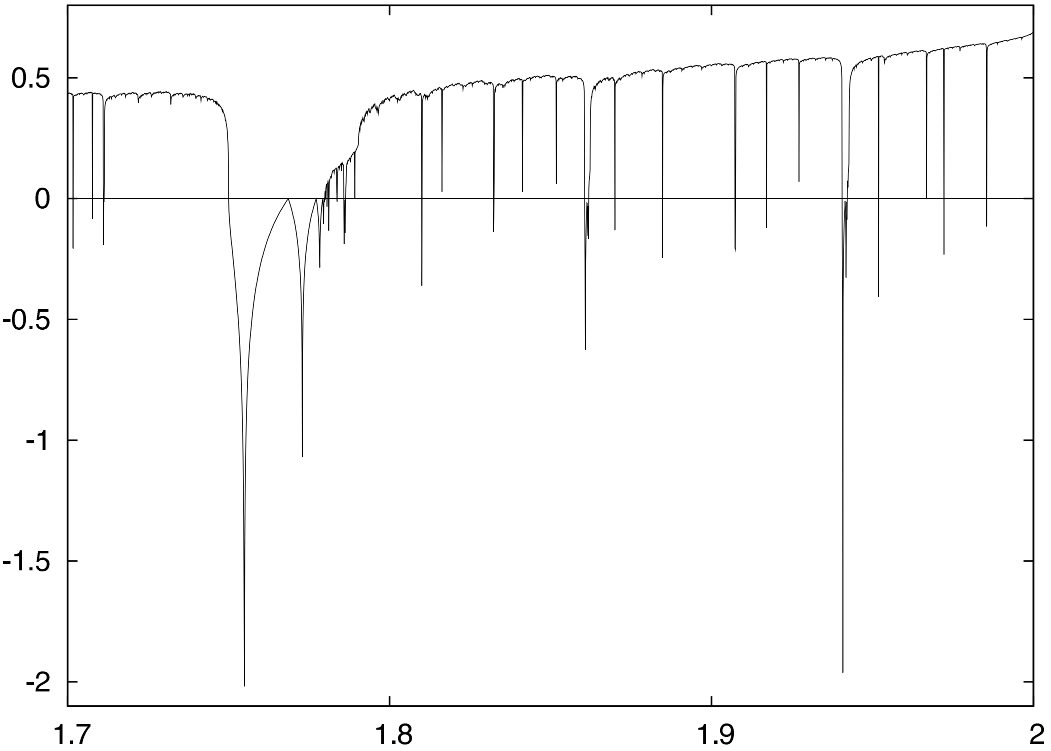

Figure 10. Non-rigorous calculations of the Lyapunov exponents in the quadratic family.

A comparison with the results of analogous computations illustrated in Figure 8 makes one notice that both graphs are very similar, except the direct computation of (9) yields consistently higher values of the Lyapunov exponent than the graph theoretic computations. This may appear unexpected in view of the fact that the computations shown in Figure 8 refer to orbits which never enter a certain neighbourhood, in this case $(-0.01, 0.01)$ of the critical point, whereas the computations in Figure 10 allow orbits in principle to come arbitrarily close to the critical point, thus in possibly picking up arbitrarily small derivatives. It seems necessary therefore to address this apparent contradiction. We explain it as follows.

First of all, we recall some non-trivial theoretical results from one-dimensional dynamics. The first is that by [3] for the quadratic family almost all parameters either have an attracting periodic orbit, and thus negative Lyapunov exponent, or satisfy the so-called Collet-Eckmann condition (positive Lyapunov exponent along the critical orbit). Moreover, by [22] the Collet-Eckmann condition implies that all Lyapunov exponents (of all invariant measures) are positive and uniformly bounded away from 0. In particular, this means that the values of λ and λ_0 as defined above in relation to the uniform expansion exponent outside some critical neighbourhood, do not depend on the size of the critical neighbourhood. This remarkable result can be understood intuitively by noticing that points very close to the critical point, shadow the orbit of the critical point for a relatively long time during which they exhibit an exponential derivative growth (from the Collet-Eckmann condition assumption). It turns out that this period of derivative growth is actually sufficient to compensate the small derivative near the critical point, giving an overall exponential growth at a uniform rate, i.e., points which are very close to the critical point shadow its orbit for a longer time so that in the end the average rate of growth is uniformly bounded from below.

The conclusion from the observations in the previous paragraph is that the computed value of the Lyapunov exponents should not depend very much on whether we allow or not entries to some critical neighbourhood and thus in theory the values computed in Figures 8 and 10 should be *comparable*, rather than one being smaller than the other, as might at first have been thought to be the case. The discrepancy is therefore down to the computational methods used in the graph-theoretic approach, and more precisely to the intrinsic limitation of using a fixed "finite-resolution" implicitly defined by our choice of partition. Indeed, in regions of high derivative, such as near an expanding fixed or periodic point, small partition elements are strongly expanded and thus mapped across other partition elements. In the graph-theoretic encoding we need to consider the worst case scenario which means that we need to assume that points in such a partition element are mapped outside this region to other regions with possibly small derivative, whereas in reality points may spend a very large number of iterates close to such an expanding fixed or period point, and thus picking up much more expansion. This explains why our approach generally yields lower values for the expansion than those computed using (9). The only exception from this rule can be observed for the thin "spikes" that go down in Figure 10, as in those few cases the orbits gain strong contraction while coming close to the critical point, which is ruled out in our case illustrated in Figure 8.

Fortunately, the limitations of the graph-theoretic approach are not a real obstruction to achieveing optimal estimates, as improved bounds can be obtained either by more cleverly constructed partitions or simply finer partitions, as discussed above. The most constructive point of view is then perhaps to see the direct non-rigorous calculations as providing a very useful benchmark for the expected values and as a guide to how fine a partition one should use in order to achieve rigorous bounds which are as close as possible to the expected ones.

Acknowledgments

We would like to acknowledge the referees for useful suggestions which helped us improve the paper, in particular one of the referees for pointing out the issue discussed in Section 5.2.3 and providing the graph shown in Figure 10.

The computations reported on in this paper were performed on a cluster of AMD Opteron 248 Processors in Mischaikow's lab at Rutgers University which is supported in part by NSF, DoE, and DARPA.

References

[1] Z. Arai. On hyperbolic plateaus of the Hénon map. *Experimental Mathematics*, 16(2): 181–188, 2007.

[2] Z. Arai. On loops in the hyperbolic locus of the complex Hénon map and their monodromies. Submitted.

[3] A. Avila and C. G. Moreira. Statistical properties of unimodal maps: the quadratic family. *Ann. of Math. (2)*, 161(2):831–881, 2005.

[4] L. Barreira and Y. Pesin. *Nonuniform hyperbolicity: Dynamics of systems with nonzero Lyapunov exponents.* Springer-Verlag, 2007.

[5] M. Benedicks and L. Carleson. The dynamics of the Hénon maps. *Ann. of Math.* 133:73–169, 1991.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to algorithms.* MIT Press, Cambridge, MA, second edition, 2001.

[7] G. Gottwald and I. Melbourne. Testing for chaos in deterministic systems with noise. *Physica D: Nonlinear Phenomena*, 212(1-2):100–110, 2005.

[8] S. L. Hruska. A numerical method for constructing the hyperbolic structure of complex Hénon mappings. *Found. Comput. Math.*, 6(4):427–455, 2006.

[9] M. Jacobson. Absolutely continuous invariant measures for one-parameter families of one-dimensional maps. *Comm. Math. Phys.* 81: 39–88, 1981.

[10] R. M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Math.*, 23(3):309–311, 1978.

[11] O. Kozlovski, W. Shen, and S. van Strien. Rigidity for real polynomials. *Annals of Mathematics*, 165(3):749–841, 2007.

[12] S. Luzzatto and H. Takahasi. Computable conditions for the occurrence of non-uniform hyperbolicity in families of one-dimensional maps. *Nonlinearity*, 19(7):1657–1695, 2006.

[13] S. Luzzatto and W. Tucker. Non-uniformly expanding dynamics in maps with singularities and criticalities. *Inst. Hautes Études Sci. Publ. Math.*, (89):179–226 (2000), 1999.

[14] S. Luzzatto and M. Viana. Positive Lyapunov exponents for Lorenz-like families with criticalities. *Astérisque*, (261):xiii, 201–237, 2000. Géométrie complexe et systèmes dynamiques (Orsay, 1995).

[15] R. Mañé. Hyperbolicity, sinks and measure in one-dimensional dynamics. *Comm. Math. Phys.*, 100(4):495–524, 1985.

[16] W. de Melo and S. van Strien. *One-dimensional dynamics*, volume 25 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)].* Springer-Verlag, Berlin, 1993.

[17] M. Misiurewicz. Absolutely continuous measures for certain maps of an interval. *Inst. Hautes Études Sci. Publ. Math.*, 53:17–51, 1981.

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

[19] T. Nowicki and D. Sands. Non-uniform hyperbolicity and universal bounds for S-unimodal maps. *Invent. Math.*, 132(3):633–680, 1998.

[20] M. J. Pacífico, A. Rovella and M. Viana. Infinite-modal maps with global chaotic behavior. *Ann. of Math. (2)*, 148(2):441–484, 1998.

[21] A. Rovella. The dynamics of perturbations of the contracting Lorenz attractor. *Bol. Soc. Brasil. Mat. (N.S.)*, 24(2):233–259, 1993.

[22] D. Sands and T. Nowicki. Quasisymmetric invariance of the Collet-Eckmann condition. *Ergodic Theory Dynam. Systems*, 18(3):703–715, 1998.

[23] C. Simó. Computer assisted studies in dynamical systems. In *Progress in nonlinear science, Vol. 1 (Nizhny Novgorod, 2001)*, pages 152–165. RAS, Inst. Appl. Phys., Nizhniĭ Novgorod, 2002.

[24] C. Simó and J.C. Tatjer. Windows of attraction of the logistic map. In *European Conference on Iteration Theory (Batschuns, 1989)*, pages 335–342. World Sci. Publ., River Edge, NJ, 1991.

[25] W. Tucker. The Lorenz attractor exists. *C. R. Acad. Sci. Paris Sér. I Math.*, 328(12):1197–1202, 1999.

[26] S. Ulam and J. von Neumann. On combination of stochastic and deterministic processes. *Bull AMS*, 53:1120, 1947.