

COMPUTABILITY IN DYNAMICAL SYSTEMS

MICHAEL BURR AND CHRISTIAN WOLF

ABSTRACT. In this paper we present an introduction to the area of computability in dynamical systems. This is a fairly new field which has received quite some attention in recent years. One of the central questions in this area is if relevant dynamical objects can be algorithmically presented by a Turing machine. After providing an overview of the relevant objects we discuss recent results concerning the computability of the entropy for symbolic systems and the computability of Julia sets as well as their Brolin-Lyubich measures.

1. INTRODUCTION

In [16], Milnor poses the question of whether an algorithm exists to approximate the entropy of a dynamical system up to a specified error. Such an algorithm does not exist in the most general setting, but this nonexistence leads to the following question:

For which classes of dynamical systems and invariants can these invariants be algorithmically approximated to any specified precision?

In dynamics and the sciences, computational experiments are often used to gain insight into the behavior of dynamical systems. For invariants which cannot be algorithmically approximated, the results of such computational experiments may be meaningless. We approach the generalization of Milnor's question from the point of view of computability theory.

We begin with the computational model for computability theory. From a theoretical standpoint, we may study a dynamical system $f : X \rightarrow X$ defined by an infinite amount of data. From an algorithmic standpoint, however, only a finite amount of data may be provided as input. Therefore, by necessity, we replace the dynamical system $f : X \rightarrow X$ by a convergent sequence of approximations $f_i : X_i \rightarrow X_i$, each of which may be described by a finite amount of data. Milnor's question then leads to the question

Do computations with a sequence of approximations $f_i : X_i \rightarrow X_i$ provide insight into the properties of the original dynamical system $f : X \rightarrow X$?

We apply computability theory to three case studies from dynamical systems which yield both positive and negative results.

1.1. Dynamical Systems. A dynamical system consists of a state space X (the set of attainable states) and a time evolution law

$$(\Phi_t)_{t \in T} : X \rightarrow X,$$

Burr was partially supported by grants from the National Science Foundation (CCF-1527193 and DMS-1913119).

Wolf was partially supported by grants from the Simons Foundation (#637594 to Christian Wolf) and PSC-CUNY (TRADB-51-63715 to Christian Wolf).

where T denotes the set of times when the state of the system can be observed. In other words, if the system is in state x_0 at time $t = 0$, then it is in state $\Phi_t(x_0)$ at time t . We focus on discrete dynamical systems where $T = \mathbb{N}_0$. We assume that $\Phi_0 = \text{id}_X$ and that $\Phi_{n+m} = \Phi_n \circ \Phi_m$ for all $n, m \in \mathbb{N}$. Hence

$$\Phi_n(x_0) = \underbrace{\Phi_1 \circ \dots \circ \Phi_1}_{n\text{-times}}(x_0) =: \Phi_1^n(x_0).$$

Consequently, such a dynamical system is given by the iteration of a map $f := \Phi_1 : X \rightarrow X$.

We consider discrete dynamical systems given by continuous maps $f : X \rightarrow X$ where (X, d) is a compact metric space. Given $x \in X$ we call $\mathcal{O}(x) := \mathcal{O}^+(x) := \{x, f(x), f^2(x), \dots\}$ the (*forward*) *orbit* of x . If f is invertible, we analogously define the *backward orbit* $\mathcal{O}^-(x)$ by iterating f^{-1} . When $f^n(x) = x$ for some finite positive n , we say that x is *periodic* and define the *period* of x to be the cardinality of the orbit, i.e., a periodic point of period n has orbit $\mathcal{O}(x) = \{x, f(x), \dots, f^{n-1}(x)\}$, where each $f^i(x)$ is distinct for $0 \leq i < n$ and $x = f^n(x)$. We denote the set of *periodic points of f with period n* by $\text{Per}_n(f)$ and the set of *periodic points of f* by $\text{Per}(f) = \bigcup_n \text{Per}_n(f)$.

In dynamics, it is a central goal to understand the asymptotic behavior of the orbits of $f : X \rightarrow X$ for a large class of points in X . For instance, in ergodic theory, large often means a set of full measure for an f -invariant probability measure.

To motivate some of the challenges in studying families of dynamical systems computationally, the following example illustrates one of the pitfalls in the computational study of a single dynamical system.

Example 1.1. Consider the $2x\text{-mod-}1$ dynamical system where $X = [0, 1)$ and $f : X \rightarrow X$ takes x to the fractional part of $2x$. It is straightforward to show that some iterate of x is periodic if and only if x is rational. Moreover, every dyadic rational is eventually mapped onto the fixed point 0. Conversely, the closure of the orbit of an irrational point is either a Cantor set or X itself.

One challenge in studying this dynamical system computationally is that the orbits of rational points are not representative of a typical orbit, even though the rationals are dense in X . Also, the most common number types on a computer are dyadic rationals, which exhibit even more atypical behavior. This example illustrates that care must be taken with both the selection of representative points and how conclusions are drawn from computational experiments.

1.2. Computability theory. The main technique in computability theory is to replace a single mathematical object by a convergent sequence, and to use the convergent sequence to study the original object, see [4, 20].

Definition 1.2. An *oracle approximation* of a real number α is a function ψ such that for any $n \in \mathbb{N}$, $\psi(n)$ is a rational number such that $|\alpha - \psi(n)| < 2^{-n}$. A real number α is *computable* if there exists a Turing machine which is an oracle approximation for α .

For most purposes, it is sufficient to think of a Turing machine as an algorithm. Since every Turing machine is represented by a finite string of characters from a finite alphabet, we conclude that the computable numbers form a countable dense subset of the real numbers. This conclusion immediately implies that noncomputable real numbers exist.

The error estimates in the definition of a computable real number are quite strong. In some cases, there is a Turing machine that approximates α without an estimate on the convergence rate.

Definition 1.3. A real number α is *computable from above* (respectively, *from below*) if there exists a Turing machine ψ such that on input n , $\psi(n)$ is a rational number such that the sequence $(\psi(n))_{n \in \mathbb{N}}$ is a decreasing (an increasing) sequence converging to α .

We observe that computability is equivalent to simultaneous computability from above and below. The ideas from computability theory can be extended to functions as well as to many other settings.

Definition 1.4. A function $f : \mathbb{R} \rightarrow \mathbb{R}$ is a *computable function* if there exists a Turing machine ϕ with the following property: For any $\alpha \in \mathbb{R}$ and oracle approximation ψ for α , on input (ψ, n) , $\phi(\psi, n)$ is a rational number such that $|f(\alpha) - \phi(\psi, n)| < 2^{-n}$.

The Turing machine ϕ must be able to accept every oracle approximation for any α (not merely Turing machines) and produce meaningful results. Computable functions are necessarily continuous, and there are analogous notions of computability from above and below for computable functions.

We introduce the concept of a computable metric space and refer the reader to [13] for more details.

Definition 1.5. Let (X, d_X) be a separable metric space and let $\mathcal{S}_X = (s_i)_{i \in \mathbb{N}}$ be a dense sequence of points in X . We say that (X, d_X, \mathcal{S}_X) is a *computable metric space* if there is a Turing machine $\chi : \mathbb{N}^2 \times \mathbb{N} \rightarrow \mathbb{Q}$ such that $|\chi(i, j, n) - d_X(s_i, s_j)| < 2^{-n}$.

The definition for a computable metric space is equivalent to the computability of the function $(i, j) \mapsto d_X(s_i, s_j)$. Importantly the approximation is uniform, that is, the same Turing machine χ is used for every i and j in \mathbb{N} . The definitions of computable numbers and functions generalize to computable metric spaces by replacing the output of Turing machines by an index $i \in \mathbb{N}$ of a point in \mathcal{S}_X .

2. COMPUTABILITY OF ENTROPY

Arguably, the most important invariant of a dynamical system f is its topological entropy $H_{\text{top}}(f)$. The topological entropy measures the complexity of the dynamical system f , and it is invariant under topological conjugation. Roughly, $H_{\text{top}}(f)$ measures the exponential growth rate of the number of orbit segments that can be distinguished up to an error of ε . Specifically, for $n \in \mathbb{N}$ and $\varepsilon > 0$, a finite set $F \subset X$ is (n, ε) -separated if $\max_{k=0, \dots, n-1} d(f^k(x), f^k(y)) \geq \varepsilon$ for distinct points $x, y \in F$. Let $F_n(\varepsilon)$ denote the largest cardinality of an (n, ε) -separated set. The *topological entropy* of f is

$$(1) \quad H_{\text{top}}(f) := \lim_{\varepsilon \rightarrow 0} \limsup_{n \rightarrow \infty} \frac{1}{n} \log F_n(\varepsilon).$$

From a computability standpoint, the limits in the formula for the topological entropy make its calculation difficult. Even if $F_n(\varepsilon)$ can be approximated for n and ε , a computation cannot decide without additional theory whether ε is small enough and n is large enough for $F_n(\varepsilon)$ to approximate the limit.

We say that f is *expansive* with *expansivity constant* $2\varepsilon_0$ if for distinct $x, y \in X$, there is some n such that $d(f^n(x), f^n(y)) > 2\varepsilon_0$. In this case, we can omit the limit on ε and compute the \limsup for ε_0 . Then sequence $(h_n)_n$ defined by

$$(2) \quad h_n := \min_{k \leq n} \frac{1}{k} \log F_k(\varepsilon_0)$$

is non-increasing and converges to $H_{\text{top}}(f)$.

Corollary 2.1. Let $f : X \rightarrow X$ be an expansive dynamical system with expansivity constant $2\varepsilon_0$ that is given by an oracle χ . If there exists a Turing machine that takes χ and $n \in \mathbb{N}$ as input and computes $F_n(\varepsilon_0)$, then $H_{\text{top}}(f)$ is upper semi-computable.

2.1. Symbolic dynamical systems. The class of symbolic systems form a rich family of discrete dynamical systems. These systems provide good test cases for many theories as they can be studied in a hands-on way, but have enough variability to provide challenging examples. We review the relevant material from symbolic dynamics, see [15].

Let $\mathcal{A} = \{0, \dots, d-1\}$ be a finite set called the *alphabet*. Let \mathcal{A}^* denote the set of all finite words in \mathcal{A} . The *full shift* Σ with alphabet \mathcal{A} is the set of bi-infinite sequences $(x_k)_{k \in \mathbb{Z}}$ of elements of \mathcal{A} . With the *Tychonov product topology*, which coincides with the topology generated by the metric

$$d(x, y) := 2^{-\min\{|k| : x_k \neq y_k\}},$$

Σ is a compact metrizable topological space.

The *(left) shift map* $f : \Sigma \rightarrow \Sigma$ is defined by $f(x)_k = x_{k+1}$, and we note that f is a homeomorphism. We call a closed shift-invariant set $X \subset \Sigma$ a *shift space*, and we say that $f : X \rightarrow X$ is a *subshift*.

For a fixed subshift X , let $\mathcal{L}(X, n)$ be the set of length- n words which occur in some element of X . Moreover, we call $\mathcal{L} = \bigcup_{n \in \mathbb{N}} \mathcal{L}(X, n)$ the *language* of X . Languages and shift spaces are equivalent objects.

For a shift map $f : X \rightarrow X$, the entropy formula in Definition (1) reduces to

$$(3) \quad H_{\text{top}}(f|_X) := \lim_{n \rightarrow \infty} \frac{1}{n} \log \# \mathcal{L}(X, n).$$

Moreover, this limit converges from above.

An oracle χ of the language $\mathcal{L}(X)$ is a function such that $\chi(n)$ is a list of the words of the language of length n . In light of the simplicity of the entropy of a subshift X and Corollary 2.1, one might suspect that if the language of shift space is given by an oracle, then the entropy $H_{\text{top}}(f|_X)$ is computable. This statement, however, is not true.

The following theorem uses the notion of *computability at a point*, which is a computable version of continuity at a point, see [8] for more details.

Theorem 2.2 ([6]). The generalized entropy map $X \mapsto H(f|_X)$ is computable at a nonempty shift space X_0 if and only if X_0 has zero entropy. Moreover, there is one Turing machine that uniformly computes the topological entropy at all shift spaces with zero entropy.

This fact is proved by noting that the entropy is nonnegative, and, for any value of $\mathcal{L}(X, n)$, there is a zero-entropy shift space X' which has the same language as X up to length n . For example, X' can be a union of orbits of points whose forward and backward tails are periodic. Thus, given only the language, the only possible lower bound on the entropy is zero.

2.2. SFTs of finite type and Sofic shifts. Subshifts of finite type and Sofic shifts are among the shifts for which the computability of their dynamics is best understood. Both of these shifts can be described in terms of finite data.

Let $\mathcal{F} \subset \mathcal{A}^*$ be called a *forbidden set*. Then $X_{\mathcal{F}}$ is defined to be the largest shift space such that no point $x \in X_{\mathcal{F}}$ contains any of the words in \mathcal{F} .

Definition 2.3. A shift space X is a *subshift of finite type (SFT)* if $X = X_{\mathcal{F}}$ for some finite set of words \mathcal{F} .

The set \mathcal{F} in this definition is not unique. A set \mathcal{F} is a *minimal forbidden set* for an SFT X if for every word in \mathcal{F} , all of its proper subwords appear in the language of X . It follows that there is only one minimal forbidden set for an SFT. One less than the length of the largest word in the minimal forbidden set is called the *step size* of the SFT. Every SFT $X = X_{\mathcal{F}}$ is topologically conjugate to a one-step SFT $X' = X'_{\mathcal{F}'}$ on a possibly larger alphabet \mathcal{A}' . Moreover, given \mathcal{F} , the forbidden set \mathcal{F}' can be computed by a Turing machine. Thus, it suffices to study one-step SFTs. One-step SFTs can also be described via a *transition matrix*, as follows: Let $A = (a_{i,j})$ be an $\mathcal{A} \times \mathcal{A}$ zero-one matrix. Then

$$(4) \quad X_A := \{(x_k)_k : a_{x_i, x_{i+1}} = 1\}$$

is a one-step SFT. Moreover, every one-step SFT can be written this way. Additionally, the entropy $H(f|_{X_A})$ of the SFT X_A is the logarithm of the spectral radius of the matrix A . Since there are algorithms that compute the spectral radius of A at any given precision, we obtain the following result.

Proposition 2.4. There is a Turing machine that computes the topological entropy of any SFT given by either a finite forbidden set or a transition matrix.

Definition 2.5. Let $\mathcal{T} = (G, E, L)$ be a finite labeled directed graph, where G is a graph with directed edge set E and *labeling function* $L : E \rightarrow \mathcal{A}$, which assigns a *label* $L(e)$ from the finite alphabet \mathcal{A} to each edge $e \in E$. Let $\xi = \cdots e_{-1}e_0e_1\cdots$ be a bi-infinite path on G . The *label* of the path ξ is

$$L(\xi) := \cdots L(e_{-1})L(e_0)L(e_1)\cdots \in \mathcal{A}^{\mathbb{Z}}.$$

The set of all bi-infinite labels of paths is denoted by $X_{\mathcal{T}} = \{L(\xi) : \xi \in G\}$. A subset X of the full shift is a *Sofic shift* if $X = X_{\mathcal{T}}$ for a finite labeled graph \mathcal{T} .

A shift space X is Sofic if and only if X is a (finite-to-one) factor of some SFT Y . In particular, every SFT is Sofic, but the converse is not true. Moreover, there is a Turing machine that takes, as input, a finite labeled directed graph \mathcal{T} and outputs a transition matrix A such that the Sofic shift $X_{\mathcal{T}}$ is a factor of the SFT X_A . Since finite-to-one factors preserve entropy, we obtain the following result.

Proposition 2.6. There exists a Turing machine that computes the topological entropy of any Sofic shift that is given by a finite labeled directed graph.

2.3. Invariant measures. There is a natural connection between topological and measure-theoretic entropies, and we are interested in the computability properties of the measure-theoretic entropy. A probability measure μ is said to be *f-invariant* if $\mu(A) = \mu(f^{-1}(A))$ for all Borel sets A . For example, if $x \in \text{Per}_n(f)$ then $\mu_x = \frac{1}{n} \sum_{k=0}^{n-1} \delta_{f^k(x)}$ is the unique invariant measure supported on $\mathcal{O}(x)$, where $\delta_{f^k(x)}$ denotes the Dirac measure on $f^k(x)$. Let \mathcal{M}_f denote the space of all f -invariant Borel probability measures on X endowed with the weak* topology. With

this topology, \mathcal{M}_f is a compact convex metrizable topological space. The extreme points in this convex space are denoted by $\mathcal{M}_{f,E}$. These measures are precisely the ergodic measures, where an f -invariant measure is *ergodic* if every f -invariant set has either measure zero or one.

Example 2.7. Let $X = [0, 1)$ and $\alpha \in [0, \infty)$. Let $f : X \rightarrow X$ be a translation by α , i.e., $f(x)$ is the fractional part of $x + \alpha$. If $\alpha \in \mathbb{Q}$, then every point in X is periodic and $\mathcal{M}_{f,E} = \{\mu_x : x \in X\} \cup \{\lambda\}$, where λ denotes the Lebesgue measure. On the other hand, if $\alpha \in \mathbb{R}^+ \setminus \mathbb{Q}$ then $\mathcal{M}_f = \mathcal{M}_{f,E} = \{\lambda\}$.

Let $H_\mu(f)$ denote the measure-theoretic entropy of f with respect to μ . Roughly, $h_\mu(f)$ is similar to the topological entropy, but ignores sets of measure zero, see Section 4 for details. The relationship between the topological and measure-theoretic entropies is the variational principle for the entropy, see Equation (5).

2.4. General and coded shifts. The computability properties of more general classes of shifts are more complicated. One reason why these spaces are more challenging to study is that their descriptions may be given in terms of infinite amounts of data. By Corollary 2.1, if the language is given by an oracle, then the topological entropy is upper semi-computable. Thus, we are interested in identifying when the entropy is also lower semi-computable.

We approach this problem by using a sequence $\{X_m\}_{m \in \mathbb{N}}$ of Sofic shifts which “exhausts” X in the sense that the entropy of these X_m ’s approaches that of X from below. The challenge with this approach is that the generalized entropy function $X \mapsto H_{\text{top}}(f|_X)$ might not be continuous. Whenever such an approximation exists, one calls such a shift an *almost Sofic shift*, see [19]. In most cases, the language itself is not enough to compute such an approximation, we, therefore, focus on the case where X is generated from concatenations of a countable set of finite words, that is X is a coded shift.

To emphasize the generality of the class of coded shifts, we note that every shift space X can be expressed as the closure of the set of bi-infinite paths on a countable directed labeled graph, called a *representation* of X , see [15]. Coded shifts are precisely those shift spaces which have a representation by an irreducible countable directed labeled graph. Many classes of shifts spaces are coded shifts, including Sofic shifts, S-gap, and Beta-shifts.

Definition 2.8. A two-sided shift space X over a finite alphabet $\mathcal{A} = \{0, \dots, d-1\}$ is *coded* if there exists a *generating set* $\mathcal{G} = \{g_i\}_{i \in \mathbb{N}} \subset \mathcal{A}^*$, such that $X = X(\mathcal{G})$ is the smallest shift space that contains all bi-infinite concatenations of generators, i.e.,

$$X_{\text{seq}} := \{\dots g_{i-2}g_{i-1}g_{i_0}g_{i_1}g_{i_2}\dots : i_j \in \mathbb{N}, j \in \mathbb{Z}\}.$$

In other words, X is the topological closure of X_{seq} , for more details, see [3, 15].

Let $X_m := X(\{g_1, \dots, g_m\})$ be the coded shift generated by the first m generators in \mathcal{G} . Since each X_m is a Sofic shift, its entropy is computable. We observe that $\{X_m\}_{m \in \mathbb{N}}$ is an increasing sequence of shift spaces in X_{seq} . In order to establish computability results for the entropy of coded shifts, we study when the entropy of $\{X_m\}_{m \in \mathbb{N}}$ converges to the entropy of X . For this convergence to hold, we show that the entropy on X cannot be “concentrated” on $X \setminus \bigcup_m X_m$.

To make these notions more precise, we introduce the following two definitions:

Definition 2.9. [[6]] We say that X has *full sequential entropy* if

$$H_{\text{top}}(f|_X) = \sup_{\mu \in \mathcal{M}} \{H_{\mu}(f) : \mu(X_{\text{seq}}) = 1\}.$$

We say that X has *strict full sequential entropy* if

$$\sup_{\mu \in \mathcal{M}} \{H_{\mu}(f) : \mu(X_{\text{seq}}) = 1\} > \sup_{\mu \in \mathcal{M}} \{H_{\mu}(f) : \mu(X \setminus X_{\text{seq}}) = 1\}.$$

We observe that strict full sequential entropy implies full sequential entropy.

Definition 2.10 (cf [18]). A *unique representation* of a coded shift X is a generating set \mathcal{G} of X such that each $x \in X_{\text{seq}}(\mathcal{G})$ can be uniquely written as an infinite concatenation of elements of \mathcal{G} .

The manuscript [1] includes a proof that all coded shifts have a unique representation.

Theorem 2.11 ([6]). Let X be a coded shift with unique representation \mathcal{G} .

- (i) If X has strict full sequential entropy then $H_{\text{top}}(f|_X) = \lim_{m \rightarrow \infty} H_{\text{top}}(f|_{X_m})$.
- (ii) Suppose X has strict full sequential entropy. There exists a Turing machine, independent of X , that computes $H_{\text{top}}(f|_X)$ from oracles for the language and generating set \mathcal{G} of X . In particular, $H_{\text{top}}(f|_X)$ is computable; and
- (iii) If X does not have full sequential entropy, then the entropy is not continuous at X and hence is not computable at X .

Since the entropy is not continuous as a function of the shift space, the main challenge in the proof of Statement (ii) of this theorem is to show that the sequence of entropies for X_m converges to that of X . In the proof of Statement (iii), it is shown that the entropy for subshifts X_m does not converge to the entropy of f on X . We refer to [6, 19] for examples of coded shifts without full sequential entropy. In particular, there are coded shifts for which the topological entropy is not computable.

2.5. Classification of entropies. An alternative to computing the entropy of a dynamical system is to classify the computability properties of entropies of dynamical systems. For instance, by Proposition 2.4, the entropies of all SFTs are not only computable, but also are rational logarithms of Perron numbers, where a *Perron number* is a real zero of an integer polynomial such that the zero is greater than one and is the zero of largest magnitude. The converse of this statement also holds, i.e., every rational logarithm of a Perron number is the entropy of some mixing SFT, see [14].

A natural generalization of one-dimensional SFTs are multidimensional SFTs which are finite type \mathbb{Z}^d -actions. In this case, the set of possible entropies coincides with the set of positive upper computable numbers [12]. Since there are upper computable numbers which are not computable, there exist symbolic systems whose entropy is not a computable number. On the other hand, the entropy of strongly irreducible higher dimensional SFTs is computable.

In more generality, for computable dynamical systems, see the definition in [10], the entropy is always a supremum of a computable sequence of upper-computable numbers [10]. Moreover, for every positive such number, there is a dynamical system over $\{0, 1\}^{\mathbb{N}}$ which has this entropy and for which the time evolution function is computable.

3. COMPUTABILITY OF JULIA SETS

Computer generated pictures of the Mandelbrot set and Julia sets can be quite striking, but computability theory raises the question of whether the underlying sample used to generate these sets accurately represents their true shape. Julia sets are prototypes of fractals consisting of the points with unpredictable dynamics. Since the dynamical behavior of nearby points can differ, computations with sample points may not correctly capture the dynamical behavior.

Let $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ be the Riemann Sphere endowed with the spherical metric $d_{\overline{\mathbb{C}}}$. Let

$$f_c(z) = z^d + c_{d-2}z^{d-2} + \cdots + c_0$$

be a polynomial of degree d , where $c = (c_{d-2}, \dots, c_0) \in \mathbb{C}^{d-1}$. We denote the space of polynomials of this form by $\text{Pol}(d)$.

Fatou and Julia introduced a partition of $\overline{\mathbb{C}}$ into two completely invariant sets, F_f and J_f , called the *Fatou* and *Julia* sets, respectively. The Fatou set F_f is the set of points $z \in \overline{\mathbb{C}}$ where the family of iterates of f is equicontinuous in a neighborhood of z . The Julia set J_f is the complement of F_f in the Riemann sphere, and thus, the Fatou set is open, while the Julia set is compact. In particular, $z \in F_f$ if and only if for all $\varepsilon > 0$ there exists $\delta > 0$ such that $d_{\overline{\mathbb{C}}}(z, w) < \delta$ implies $d_{\overline{\mathbb{C}}}(f^n(z), f^n(w)) < \varepsilon$ for all $n \in \mathbb{N}$. In other words, the Fatou set is the set of points whose orbits are Lyapunov stable. The Julia set is the set of points whose orbits have sensitive dependence on the initial conditions, i.e., *chaotic dynamics*.

When $d = 2$, the polynomials are of the form $f_c(z) = z^2 + c$. The Julia set of a quadratic polynomial is either connected or totally disconnected. The connectivity-locus

$$\mathcal{M} = \{c \in \mathbb{C} : J_c \text{ is connected}\}$$

is the famous *Mandelbrot set*. The *filled-in Julia set* of f is defined by

$$K_f = \{z \in \mathbb{C} : \{f^n(z) : n \in \mathbb{N}_0\} \text{ is bounded}\}.$$

The iterates of the points in $U_f = \overline{\mathbb{C}} \setminus K_f$ converge to ∞ . Moreover,

$$J_f = \partial K_f = \partial U_f \text{ and } F_f = U_f \cup \text{int } K_f.$$

The sensitivity to the initial conditions makes computing the Julia set particularly challenging.

3.1. Computability of sets.

Definition 3.1. Let $S \subset \mathbb{R}^m$ be an open set. S is a *recursively open set* (also called a semi-decidable set or a lower-computable set) if there exists a Turing machine ψ that produces a (possibly infinite) sequence of pairs (z_i, n_i) so that $z_i \in \mathbb{Q}^m$ is a rational vector and $n_i \in \mathbb{Z}$ so that $S = \cup_i B(z_i, 2^{-n_i})$

There is a Turing machine, whose input is ψ , that terminates if $s \in S$, but may fail to terminate if $s \notin S$.

Definition 3.2. Let $S, T \subset \mathbb{R}^m$ be compact sets. The *Hausdorff distance* between S and T is

$$d_H(S, T) := \max\{\max_{s \in S} \min_{t \in T} d(s, t), \max_{t \in T} \min_{s \in S} d(s, t)\}.$$

Definition 3.3. Let $S \subset \mathbb{R}^m$ be compact. S is *computable* if there exists a Turing machine ψ such that on input n , $\psi(n)$ is a finite collection of pairs (q_i, l_i) where $q_i \in \mathbb{Q}^m$ and $l_i \in \mathbb{Z}$ such that $d_H(\cup_i B(q_i, 2^{-l_i}), S) < 2^{-n}$

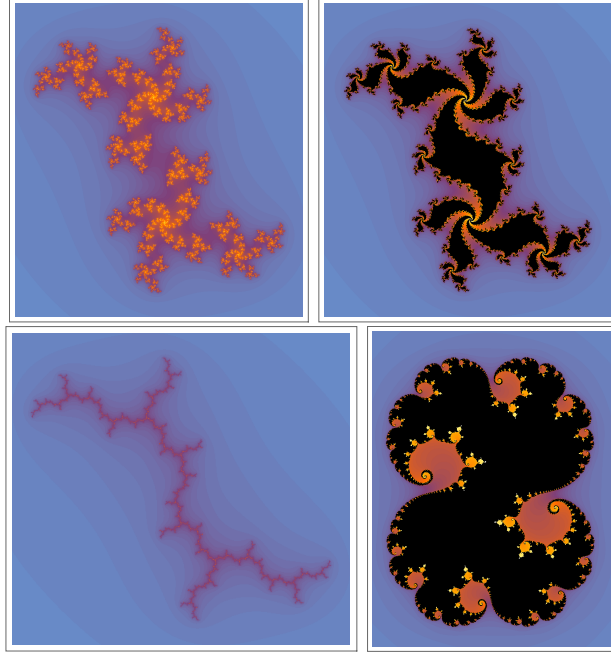


FIGURE 1. Examples of (filled-in) Julia sets for $z^2 + c$: (a) $c \notin \mathcal{M}$, totally disconnected Julia set (b) $c \in \text{int}\mathcal{M}$, f is hyperbolic (c) $c \in \partial\mathcal{M}$, Julia set is a dendrite (d) $c \in \partial\mathcal{M}$, 0 is parabolic fixed point. In all these examples the Julia set is computable. These images were created with Mathematica.

There is a Turing machine, whose input is ψ , that terminates if $s \notin S$, but may fail to terminate if $s \in S$.

3.2. Properties of periodic points. By a theorem of Fatou and Julia, the Julia set is the closure of the set of repelling periodic orbits. A periodic point $\alpha \in \text{Per}_k(f)$ is *repelling* (respectively, *attracting*) if $|(f^k)'(\alpha)| > 1$ (if $|(f^k)'(\alpha)| < 1$). A periodic point $\alpha \in \text{Per}_k(f)$ is *neutral* if $|(f^k)'(\alpha)| = 1$.

For a neutral periodic point, we replace f by its k -th iterate so that α is a fixed point. We write $f'(\alpha) = e^{2\pi i\theta}$. We say that α is *parabolic* if $f'(\alpha)$ is a root of unity, i.e., $\theta \in \mathbb{Q}$. In this case, $\alpha \in J$ and the dynamics near α is given by the Leau-Fatou Flower theorem, i.e., alternating attracting and repelling petals [17].

We say that α is a *Siegel point* if the dynamics near α is conjugate to the linear map $L_\alpha(z) = f'(\alpha)z$. Then $\alpha \in F_f$ and the connected component $U \subset F_f$ containing α is a *Siegel disk*, i.e., there exists a conformal map φ from U to the unit disk D such that $f(z) = \varphi^{-1} \circ L_\alpha \circ \varphi(z)$ for all $z \in U$.

By Bryuno's Theorem, α is Siegel if θ is a *Bryuno number*, that is,

$$\sum_n \frac{\log(q_{n+1})}{q_n} < \infty,$$

where q_n is the denominator of the n -th continued fraction expansion of θ . Moreover, the Bryuno numbers have full Lebesgue measure. We denote the set of degree d polynomials with a Siegel disk by $\text{Pol}_{\text{Sie}}(d)$.

3.3. Noncomputability of Julia sets. The map $c \mapsto J_c$ is not continuous at c if f_c has either a parabolic or a Siegel point, see [5]. In particular, the map $c \mapsto J_c$ is not computable on $\text{Pol}(d)$. Furthermore, for each non-computable real number c , the Julia set J_c is not computable without oracle access to c .

Braverman and Yampolsky show that $c \mapsto J_c$ is computable on $\text{Pol}(d) \setminus \text{Pol}_{\text{Sie}}(d)$ by one Turing machine once some non-uniform information about each parabolic point α of f_c is provided. This shows that the computability of the Julia set is problematic only for polynomials with Siegel disks. For example, consider the quadratic family

$$f_\theta(z) = z^2 + e^{2\pi i\theta}z, \quad \theta \in [0, 1),$$

which has a neutral fixed point at 0. If θ is Bryuno, then f_θ has a Siegel disk centered at 0. In particular, f_θ has a Siegel disk at the origin for θ -values in a set of full Lebesgue measure. It is natural to conjecture that the non-computability of Siegel Julia sets J_θ stems from the non-computability of θ – surprisingly, this is not the case.

Braverman and Yampolsky also construct an explicit algorithm that computes θ -values with a Siegel disk at the origin. In particular, there exist computable θ values with a noncomputable Julia set J_θ . The idea of this proof is to show that the Julia set J_θ is computable if and only if the conformal radius $r(\theta)$ of its Siegel disk is computable. In addition, they show that there are computable θ 's such that $r(\theta)$ is not computable. This shows that it is impossible to compute the Julia set from θ alone since then $r(\theta)$ could be computed from θ , which is not possible.

3.4. Computability of filled Julia sets. For the filled-in Julia set the situation is better.

Theorem 3.4 ([5]). For any c , there is a Turing machine that produces a 2^{-n} -approximation of the filled-in Julia set K_c from an oracle for c and $n \in \mathbb{N}$.

The proof of Theorem 3.4 describes a pair of Turing machines, one which terminates on points further than 2^{-n} from the filled-in Julia set, and a second which terminates on points which are near the filled-in Julia set. The second Turing machine classifies the behavior of points, e.g., near the Julia set or an iterate in the interior of a Siegel disk.

This result does not imply that $c \mapsto K_c$ is computable on $\text{Pol}(d)$ since there would need to be *one* Turing machine that computes K_c for *all* $c \in \text{Pol}(d)$.

There are other recent results that discuss the computability of dynamically defined sets, e.g., the computability of Lorenz attractors [11] and the computability of generalized rotation sets [7].

4. COMPUTABILITY OF MEASURES OF MAXIMAL ENTROPY

Let $f : X \rightarrow X$ be a continuous map on a compact metric space (X, d) . As in Section 2.3, there is the notion of the measure-theoretic entropy $H_\mu(f)$ of an invariant measure μ , see Section 4.1 for the definition. A measure which maximizes $H_\mu(f)$ is called a *measure of maximal entropy*.

Roughly, μ is a measure of maximal entropy if the set of its typical points carries the full topological complexity in terms of entropy. A dynamical system might not have a measure of maximal entropy, or, when such a measure exists, it might not be unique.

For several classes of dynamical systems, there exists a unique measure of maximal entropy, e.g., transitive SFTs and sofic shifts. If $f : \overline{\mathbb{C}} \rightarrow \overline{\mathbb{C}}$ is a polynomial of degree $d \geq 2$, then f has a unique measure of maximal entropy, called the *Brolin-Lyubich (B-L) measure* λ_f . The measure of maximal entropy λ_f is fully supported on the Julia set J_f .

One might suspect that since there are non-computable Julia sets, the B-L measure is also not computable. Surprisingly, this is not the case, and the B-L measure is computable.

4.1. Measure-theoretic entropy. Using a result of Katok and Brin, we provide an intuitive description of $H_\mu(f)$ as a measure-theoretic notion of complexity.

For $n \in \mathbb{N}$, define the Bowen d_n -metric on X by

$$d_n(x, y) = \max_{k=0, \dots, n-1} d(f^k(x), f^k(y)).$$

We observe that d_n induces the same topology on X as d . Let $B_n(x, \varepsilon)$ denote the Bowen-ball of radius ε with center x in the d_n -metric.

Theorem 4.1 (Katok-Brin). Let $\mu \in \mathcal{M}_f$. Then

$$H_\mu(x) := \lim_{\varepsilon \rightarrow 0} \lim_{n \rightarrow \infty} -\frac{\log(\mu(B_n(x, \varepsilon)))}{n}$$

exists for μ for almost every $x \in X$ and $H_\mu(\cdot)$ is μ -measurable. Moreover, if μ is ergodic, then $H_\mu(\cdot)$ is constant almost everywhere.

The quantity $H_\mu(x)$ is called the *point-wise entropy* of μ at x , and we define

$$H_\mu(f) = \int_X H_\mu(x) d\mu(x).$$

For $H_\mu(x)$ to be positive, the measure of the Bowen balls must decay exponentially to zero in n . Since Bowen balls are sets of points that stay close together under iteration, one infers that $H_\mu(f) > 0$ implies that μ must be “concentrated” on sets whose points are pushed apart under iteration. Thus, μ is concentrated on sets where a small change in the initial condition results in different dynamical behavior.

Bowen also defines the topology entropy $H_{\text{top}}(Y)$ for arbitrary sets $Y \subset X$. If μ is ergodic, then $H_\mu(f) = H_{\text{top}}(\mathbb{G}_\mu)$, where \mathbb{G}_μ denotes the set of μ -generic points of μ , where a point x is μ -generic if its orbit distribution coincides with the measure μ , i.e.,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} \delta_{f^k(x)} = \mu,$$

in the weak* topology. As a consequence of Birkhoff’s Ergodic theorem, for every ergodic measure μ , the set of μ -generic points has full μ -measure. The topological and measure-theoretic entropies are related by the *variational principle for the entropy*:

$$(5) \quad H_{\text{top}}(f) = \sup_{\mu \in \mathcal{M}_f} H_\mu(f).$$

4.2. Properties of B-L measures. The B-L measure λ is characterized by the following equivalent properties, see e.g. [2]:

- (1) $H_\lambda(f) = \log d = H_{\text{top}}(f)$;
- (2) $\lambda(f(A)) = d \cdot \lambda(A)$ for all measurable sets A on which f is injective;
- (3) $\lim_{n \rightarrow \infty} \frac{1}{d^n} \sum_{w \in f^{-n}(z)} \delta_w = \lambda$ for all but finitely many $z \in \mathbb{C}$ in the weak* topology;
- (4) λ is the harmonic measure at infinity of the filled-in Julia set K_f .

4.3. Computability of measures. Let $\mathcal{M}(\overline{\mathbb{C}})$ denote that space of Borel probability measures on $\overline{\mathbb{C}}$ endowed with the weak* topology. We endow $\mathcal{M}(\overline{\mathbb{C}})$ with the Wasserstein-Kantorovich metric defined by

$$W_1(\mu, \nu) := \sup_{\phi \in 1\text{-Lip}(\overline{\mathbb{C}})} \left| \int \phi d\mu - \int \phi d\nu \right|,$$

where $1\text{-Lip}(\overline{\mathbb{C}})$ is the space of real-valued Lipschitz functions on $\overline{\mathbb{C}}$ with Lipschitz constant 1. Let \mathbb{D} denote the set of probability measures on $\overline{\mathbb{C}}$ that are finite rational convex combinations of Dirac measures supported on points in \mathbb{C} with rational coordinates. Then $(\mathcal{M}(\overline{\mathbb{C}}), W_1, \mathbb{D})$ is a computable metric space.

4.4. Computability of B-L measures. We address the computability of the map $\text{Pol}(d) \ni c \mapsto \lambda_c$ from polynomials to B-L measures.

Definition 4.2. The function $c \mapsto \lambda_c$ is computable if there exists a Turing machine ϕ with the following property: For any $c \in \text{Pol}(d)$ and any oracle approximation of ψ of c , on input ψ and n , $\phi(\psi, n)$ is a measure in \mathbb{D} with $W_1(\phi(\psi, n), \lambda_c) < 2^{-n}$.

Theorem 4.3 ([2]). Let $d \geq 2$. Then $c \mapsto \lambda_c$ is computable.

The proof of Theorem 4.3 follows from general statements about the computability of measures in $\mathcal{M}(\overline{\mathbb{C}})$ and from the characterization of the B-L measure in Property 2 of Section 4.2.

We note that Theorem 4.3 is surprising for parameters with non-computable Julia sets. To see this consider a (non-exceptional) point $z_0 \in \mathbb{C}$. For any given $n \in \mathbb{N}$, the pre-images $f_c^{-n}(z_0)$ can be calculated at any pre-described accuracy. Moreover,

$$(6) \quad f_c^{-n}(z_0) \longrightarrow J_c$$

in the Hausdorff metric. If c is a parameter for which J_c is not computable, then the accuracy of the convergence in Equation (6) can not be controlled by a Turing machine, but the convergence of

$$(7) \quad \frac{1}{d^n} \sum_{w \in f_c^{-n}(z_0)} \delta_w \longrightarrow \lambda_c$$

in the weak* topology can be computed at any prescribed accuracy. Since the support of the B-L measure is the entire Julia set, we interpret the convergence in Equation (7) as covering most of the Julia set at a prescribed rate which is sufficient to compute the B-L measure.

There are additional results in the literature concerning the computability and non-computability of other dynamically relevant measures including physical measures, e.g. [9].

REFERENCES

- [1] M.P. Béal, D. Perrin, and A. Restivo. Unambiguously coded systems. 2021. <https://arxiv.org/abs/2103.01012>.
- [2] I. Binder, M. Braverman, C. Rojas, and M. Yampolsky. Computability of Brolin-Lyubich measure. *Comm. Math. Phys.*, 308, 2011.
- [3] F. Blanchard and G. Hansel. Systèmes codés. *Theoret. Comput. Sci.*, 44:17–49, 1986.
- [4] V. Brattka, P. Hertling, and K. Weihrauch. A tutorial on computable analysis. In S.C. Barry, B. Löwe, and A. Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 425–491. Springer, New York, 2008.
- [5] M. Braverman and M. Yampolsky. Computability of Julia sets. *Mosc. Math. J.*, pages 185–231, 2008.
- [6] M. Burr, S. Das, C. Wolf, and Y. Yang. Computability of topological pressure on compact shift spaces beyond finite type. *Nonlinearity*, 35:4250–4282, 2022.
- [7] M. Burr, M. Schmoll, and C. Wolf. On the computability of rotation sets and their entropies. *Ergodic Theory Dynam. Systems*, 40:367–401, 2020.
- [8] M. Burr and C. Wolf. Computability at zero temperature. *Nonlinearity*, 33:6157–6175, 2020.
- [9] S. Galatolo, M. Hoyrup, and C. Rojas. Dynamics and abstract computability: computing invariant measures. *Discrete Contin. Dyn. Syst.*, 29:193–212, 2011.
- [10] S. Gangloff, A. Herrera, C. Rojas, and M. Sablik. Computability of topological entropy: from general systems to transformations on Cantor sets and the interval. *Discrete Contin. Dyn. Syst.*, 40:4259–4286, 2020.
- [11] D. S. Graça, C. Rojas, and N. Zhong. Computing geometric Lorenz attractors with arbitrary precision. *Trans. Amer. Math. Soc.*, 370:2955–2970, 2018.
- [12] M. Hochman and T. Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. *Annals of Mathematics*, 171:2011–2038, 2010.
- [13] Mathieu Hoyrup and Cristóbal Rojas. Computability of probability measures and Martin-Löf randomness over metric spaces. *Inform. and Comput.*, 207:830–847, 2009.
- [14] D. A. Lind. The entropies of topological Markov shifts and a related class of algebraic integers. *Ergodic Theory Dynam. Systems*, 4:283–300, 1984.
- [15] Douglas Lind and Brian Marcus. *An introduction to symbolic dynamics and coding*. Cambridge University Press, Cambridge, 1995.
- [16] John Milnor. Is entropy effectively computable? <http://www.math.stonybrook.edu/~jack/comp-ent.pdf>, 2002.
- [17] John Milnor. *Dynamics in one complex variable*, volume 160 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, third edition, 2006.
- [18] R. Pavlov. On entropy and intrinsic ergodicity of coded subshifts. *Proc. Amer. Math. Soc.*, 148:4717–4731, 2020.
- [19] Karl Petersen. Chains, entropy, coding. *Ergodic Theory and Dynamical Systems*, 6(3):415–448, 1986.
- [20] Cristóbal Rojas and Michael Yampolsky. Computable geometric complex analysis and complex dynamics. In *Handbook of computability and complexity in analysis*, Theory Appl. Comput., pages 143–172. Springer, Cham, 2021.

MICHAEL BURR, SCHOOL OF MATHEMATICAL AND STATISTICAL SCIENCES, CLEMSON UNIVERSITY
 Email address: burr2@clemson.edu

CHRISTIAN WOLF, DEPARTMENT OF MATHEMATICS, THE CUNY GRADUATE CENTER, NY, NY
 Email address: cwolf@gc.cuny.edu