FINITE RESOLUTION DYNAMICS

STEFANO LUZZATTO AND PAWEŁ PILARCZYK

Abstract. We develop a new mathematical model for describing a dynamical system at limited resolution (or finite scale), and we give precise meaning to the notion of a dynamical system having some property at all resolutions coarser than a given number. Open covers are used to approximate the topology of the phase space in a finite way, and the dynamical system is represented by means of a combinatorial multivalued map. We formulate notions of transitivity and mixing in the finite resolution setting in a computable and consistent way. Moreover, we formulate equivalent conditions for these properties in terms of graphs, and provide effective algorithms for their verification. As an application we show that the Hénon attractor is mixing at all resolutions coarser than 10^{-5}.

1. Introduction and statement of results

The theory of dynamical systems has experienced tremendous growth and development throughout the recent decades. Nevertheless, in spite of the undeniable importance of the results describing complicated dynamical properties many of these results are abstract existence or genericity results. This can be problematic for example in applications, where generally more concrete and quantitative results are desirable, especially if one is studying the properties of some very specific objects for which analytic methods fail. It is therefore natural to try to use numerical and computational techniques, and indeed there is an enormous amount of literature in this direction. Unfortunately, this approach is limited to *finite resolution* and to *finite time* and is therefore unsuitable for direct rigorous verification of infinite time or infinitesimal scale properties without additional theoretical interpretation of the results of computations.

This combination of numerical methods and theoretical interpretation can indeed be remarkably successful in certain situations. In many cases non-trivial mathematics can reduce the problem to the verification of certain "inclusion conditions" or the existence of certain geometric structures which can be rigorously verified numerically; then this information can be used to develop symbolic dynamics or prove existence of certain trajectories, e.g., heteroclinic connections. This approach includes the application of some topological methods, like the fixed point index or the Conley index. We mention for example [1, 2, 12, 13, 16, 20–22, 28, 31, 37, 39, 40, 42, 46, 50] and

2000 *Mathematics Subject Classification.* Primary: 37M99, 65P20 Secondary: 65G20.

Key words and phrases. dynamical system, finite resolution, open cover, combinatorial dynamics, rigorous numerics, directed graph, transitivity, mixing, algorithm.

Communicated by Peter Kloeden.

Acknowledgements: We express our gratitude to several colleagues who provided us with constructive feedback on our work. We also thank the anonymous referees for their very careful reading of the paper and for several suggestions which allowed us to significantly improve the presentation and even the overall point of view of the results. This research was partly funded by the Royal Society (UK). All the computations for this paper were conducted on a computer funded by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (No. 1806039), Ministry of Education, Science, Technology, Culture and Sports, Japan.

The final version of this preprint has been published in *Foundations of Computational Mathematics*, Vol. 11, No. 2, 2011, pp. 211–239, DOI: 10.1007/s10208-010-9083-z, and is available at www.springerlink.com.

refer the reader to references therein for further information. However, the bottom line is that by the intrinsic limitations of numerical computation, only "robust" phenomena, i.e., phenomena which persist for small perturbations of the system, can be proved in this way. Indeed, the requirement that all estimates be rigorously bounded, necessarily implies that there is always a little margin of perturbation in which they continue to hold true; see [30] for an interesting discussion of the properties which can be rigorously proved numerically, referred to as "inheritable properties". This is of course in many ways also one of the strengths of the methods rather than just a limitation, but it does mean that "unstable" phenomena, i.e., dynamical features which exist for one system but perhaps not for nearby systems, are essentially *undecidable*, both in a theoretical as well as in a practical sense: see [4] for a formal discussion of this problem. The best we can hope for in this situation is to be able to prove that such phenomena occur for "some", or perhaps even "many", systems close to the one of interest, but even this strategy is sometimes extremely hard to implement.

One specific example of this situation is the famous and very well studied Hénon family of two-dimensional diffeomorphisms given by

$$H_{a,b}(x,y) = (1 + y - ax^2, bx).$$

This family was introduced by Hénon in [15] where he focused particularly on the parameter values $a = 1.4$, $b = 0.3$, now commonly referred to as the "classical" parameter values. For these parameter values, non-rigorous computer simulations

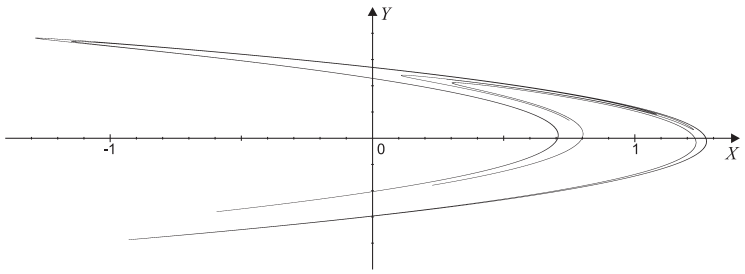

FIGURE 1. A numerical approximation of the attractor of the Hénon map for the classical parameter values.

suggest that most orbits converge to a transitive attractor with a complicated fractal geometric structure (see Figure 1), and exhibit chaotic dynamics. While the existence of an attractor follows by relatively elementary arguments, it turns out that it is extremely difficult, and indeed arguably *impossible*, to make any rigorous assertions about the infinitesimal structure of the attractor. Indeed, the geometry of the Hénon map suggests the occurrence of tangencies between stable and unstable manifolds, and in fact in [3] it is proved that for b sufficiently close to 0.3 there exists some parameter $a \in [1.392419807915, 1.392419807931]$ for which a homoclinic tangency occurs (this is proved using some clever application of Conley index theory combined with some numerical "inclusion" estimates as discussed above). Such tangencies are a well known source of instability and bifurcations and associated to a variety of dynamical phenomena such as infinitely many sinks [34,36], "small" stochastic attractors [10,33] and others [14]. In particular the dynamics is extremely unstable and there seems to be no hope to establish the actual dynamics of any single given parameter. For small values of $|b|$ it is nevertheless possible to show that there is a *positive Lebesgue measure* of parameters a for which the dynamics is stochastic in a very well defined sense, thus showing that stochastic dynamics has in some sense *positive probability*. This was first proved for $b = 0$ in [18], with

some generalization to other one-dimensional maps in [26, 27, 35, 44, 45], and for $b \neq 0$ in [5, 6], with some generalizations in [33, 47]. A quantitative approach was developed in [25] with some explicit measure bounds. Unfortunately, further extensions of these results to larger values of $|b|$, to include for example the classical parameter values, seem for the moment still out of reach. We also emphasize that, as mentioned above, these results still only provide a probabilistic description of the parameter space, and do not in general yield information about any particular given parameter.

Our goal in this paper is to approach the problem from a radically different point of view, which provides a different kind of information and is hopefully much more flexible and versatile, and more widely applicable. The underlying philosophy is that only finite precision is usually of practical meaning for applications, and all phenomena which take place below certain resolution are of no real importance, either because of the limited accuracy with which a mathematical model describes a physical system of interest (due to noise or truncations, for example), or because of the finite precision of measuring devices which provide observable properties of the system. Our goal is therefore to sketch the beginnings of a theory of *finite resolution dynamics* and to give an example of a non-trivial implementation of the concepts developed in such a theory. The starting point is defined by two basic "axioms" or "principles" which finite resolution dynamics should verify:

(A) ***Computability*** and **(B)** ***Consistency***

These principles can in practice be implemented in a variety of ways, but they embody the fundamental ideas that: (A) dynamical properties should be formulated in such a way that they are rigorously verifiable using computational methods, in particular they should be related to finite time and finite resolution properties of the system; (B) computations at a finer scale should yield more valuable results in the sense that if a dynamical property holds at some finite scale then it should also hold for any coarser representation of the same dynamics.

As an example of the application of these ideas we define a notion of *combinatorial mixing* which, in the framework of finite resolution dynamics, is analogous to the standard notions of topological or measure-theoretical mixing for topological or measurable systems, respectively. We shall prove the following

Theorem 1. *The Hénon attractor is mixing at all resolutions* $> 10^{-5}$.

We emphasize that, once the appropriate definitions have been put in place, this is a non-trivial rigorous statement about the dynamics of the Hénon map for the classical parameter values, obtained by computer-assisted methods.

The structure of the paper is as follows. In Section 2 we introduce the definitions which are at the foundations of our theory of finite resolution dynamics, and we state and prove our main abstract result. More specifically, we give a precise formulation of Axioms (A) and (B), define what we mean by the statement that a certain abstract property is verified at all resolutions coarser than some ε, and demonstrate the key fact that this can be rigorously proved by computer-assisted arguments. In Section 3 we formulate natural notions of transitivity and mixing in the finite resolution setting and prove that these finite-resolution definitions satisfy the required consistency conditions (B). In Section 4 we provide explicit algorithms for the verification of transitivity and mixing at finite resolution, and thus in particular provide a constructive proof that they satisfy the computability conditions (A). In Section 5 we describe the particular application of our theory to the Hénon map and we prove that this map is mixing at some finite but very fine scale.

This paper is accompanied by efficient and flexible software programmed in the C++ language and made freely available at [38] together with raw data for the applications discussed in Section 5.

2. Abstract theory

One possible approach to represent a dynamical system in a finite way is based on *partitioning*, that is, subdividing the phase space into a finite family of compact sets with nonempty and disjoint interiors. This approach goes back several decades and has proved extremely successful for describing certain classes of systems such as uniformly hyperbolic diffeomorphisms and flows [7, 9, 41] and has been recently further developed and used in various applications (see [2, 28, 32, 37, 42] for some examples). However, working with this kind of discretization in more general systems may not well reflect the actual topology of the phase space. Therefore, in this paper we introduce the idea of using a structure more closely related to the topology of the phase space: an *open cover*. As we shall see below, the non-trivial overlap between elements of an open cover provides a crucial ingredient for the development of a formal theory of finite resolution dynamics.

2.1. **Open covers.** Here and for the rest of the paper, X will always denote a metric space; without loss of generality, we shall assume that the metric is bounded. This generality is convenient, because in applications, like those considered in Section 5, it may be necessary to work with the space X being a bounded, not necessarily open, subset of $\mathbb{R}^n$ with the induced metric. An open ball centered at a point $x \in X$ and of radius $r > 0$ will be denoted by $B(x, r)$. For a subset $U \subset X$, its *diameter* $\operatorname{diam} U$ is the supremum of the distances between any two of its elements. Since X is bounded, the diameter of any subset of X is finite. For a family $\mathcal{A}$ of subsets of X we denote their union $\bigcup_{A\in\mathcal{A}} A$ by $|\mathcal{A}|$.

Definition 2.1. A finite family $\mathcal{U}$ of open subsets of X such that $X = |\mathcal{U}|$ is called a *cover* of X.

We are going to use the elements of $\mathcal{U}$ as a finite approximation of the topology on X. In general, the cover $\mathcal{U}$ provides a better approximation if it consists of smaller elements.

Definition 2.2. The *outer resolution*

$$\mathcal{R}^+(\mathcal{U}) := \max\{\operatorname{diam}(U) : U \in \mathcal{U}\}$$

of a cover $\mathcal{U}$ is the maximal diameter of its elements.

Intuitively, if two points $x, y \in X$ are at a distance larger than the outer resolution of a cover then they are well distinguished from each other by the cover since they must belong to distinct elements of the cover.

Remark 2.3. For a totally bounded metric space (e.g., a bounded subset of the Euclidean space with the inherited metric), there exist covers with arbitrarily small outer resolution.

Definition 2.4. We say that a cover $\mathcal{U}_1$ of X is *finer* than another cover $\mathcal{U}_2$ of X (or, equivalently, $\mathcal{U}_2$ is *coarser* than $\mathcal{U}_1$) if every element of $\mathcal{U}_1$ is contained in some element of $\mathcal{U}_2$, and every element of $\mathcal{U}_2$ contains some element of $\mathcal{U}_1$.

Note that in this definition we require a mutual relation between the elements of $\mathcal{U}_1$ and $\mathcal{U}_2$ in order to ensure that $\mathcal{U}_2$ is indeed a coarser description of the topology of X and does not recognize any finer topological structures of the space than $\mathcal{U}_1$ does. This relation between (essential) covers defines a partial order on the space of (essential) covers of X, so we can write $\mathcal{U}_1 \prec \mathcal{U}_2$ if $\mathcal{U}_1$ is finer than $\mathcal{U}_2$.

In this paper we shall focus on *essential* covers, which are in some sense minimal, that is, no element can be removed from them without leaving a considerable part of X outside the cover.

Definition 2.5. A cover $\mathcal{U}$ of X is *essential* if there exists $\varepsilon > 0$ such that every $U \in \mathcal{U}$ contains some point $x \in X$ such that $B(x, \varepsilon) \subset U$ and $B(x, \varepsilon) \cap W = \emptyset$ for all $W \in \mathcal{U} \setminus \{U\}$.

Notice that if a cover is not essential, i.e., there is an element U of the cover that does not satisfy the condition above for any $\varepsilon > 0$, then U must be contained in the union of closures of the other elements of the cover, and thus is in some sense superfluous; therefore, covers that are not essential may introduce a false feeling of the topology of the space, and indeed, as will be made clear in the sequel, some features of our theory do not go through without this assumption. Because of this reason, **from now on we shall always work with essential covers**.

Remark 2.6. From any cover it is possible to create an essential one (without increasing $\mathcal{R}^+$, but possibly decreasing the number of elements); see Appendix A for details.

Remark 2.7. For essential covers, the definition of finer and coarser covers given in Definition 2.4 reduces to the following: A cover $\mathcal{U}_1$ of X is *finer* than another cover $\mathcal{U}_2$ of X (or, equivalently, $\mathcal{U}_2$ is *coarser* than $\mathcal{U}_1$) if every element of $\mathcal{U}_1$ is contained in some element of $\mathcal{U}_2$. Indeed, this automatically implies the second part: every element of $\mathcal{U}_2$ contains some elements of $\mathcal{U}_1$. To see this, just suppose by contradiction that there exists some $U_2 \in \mathcal{U}_2$ which does not fully contain any element of $\mathcal{U}_1$. Then, since every element of $\mathcal{U}_1$ is contained in some element of $\mathcal{U}_2$, it follows that $\mathcal{U}_2 \setminus \{U_2\}$ is still a cover, contradicting the fact that it is essential. We thank Francesca Aicardi for this observation.

2.2. **Combinatorial maps.** We shall use the symbol $\multimap$ to denote a possibly multivalued map between two sets. Let X be a bounded metric space and $\mathcal{U}$ an open cover of X.

Definition 2.8. A *combinatorial map* is a multivalued map $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$.

Since $\mathcal{U}$ is finite, $\mathcal{F}$ is a finite object that can be represented in a purely combinatorial way. If $f \colon X \to X$ is a map, we can use combinatorial maps to approximate the map f.

Definition 2.9. We say that a combinatorial map $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$ is a *representation* of a map $f \colon X \to X$ if for every $U \in \mathcal{U}$ we have

$$\mathcal{F}(U) \supseteq \{W \in \mathcal{U} : W \cap f(U) \neq \emptyset\}.$$

The ideal situation would be to have a representation where we have an equality $\mathcal{F}(U) = \{W \in \mathcal{U} : W \cap f(U) \neq \emptyset\}$ but computing such a representation is generally not possible in practice due to numerical approximation and computer round-off errors in the calculation of a guaranteed outer bound for $f(U)$: see Remark 2.14 below. At the other extreme we have the trivial combinatorial map which maps each U to all elements of the cover. This is a combinatorial representation of f but gives no information whatsoever about f. To control in some way this issue, and keeping in mind also that generally $\mathcal{F}$ gives a better approximation if it is defined on a finer cover, we introduce the following

Definition 2.10. The *outer resolution*

$$\mathcal{R}^+(\mathcal{F}) := \max\{\operatorname{diam} U, \operatorname{diam} |\mathcal{F}(U)| : U \in \mathcal{U}\}$$

of a combinatorial map $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$ is the larger of the maximum diameter of the elements of the cover $\mathcal{U}$ and of the maximum diameter of their images.

If f is a continuous map with some Lipschitz constant $L \geq 0$ and the elements of $\mathcal{U}$ are of similar size (e.g., balls of the same radius), then in principle one should expect $\mathcal{R}^+(\mathcal{F}) \approx \max\{1, L\}\mathcal{R}^+(\mathcal{U})$ for a reasonable representation $\mathcal{F}$ of f.

Definition 2.11. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ be two covers of X. We say that $\mathcal{F}_1 \colon \mathcal{U}_1 \multimap \mathcal{U}_1$ is *finer* than $\mathcal{F}_2 \colon \mathcal{U}_2 \multimap \mathcal{U}_2$ (or, equivalently, $\mathcal{F}_2$ is *coarser* than $\mathcal{F}_1$) if $\mathcal{U}_1$ is finer than $\mathcal{U}_2$ and for every $U_1 \in \mathcal{U}_1$ and every $U_2 \in \mathcal{U}_2$ such that $U_1 \subset U_2$, every element of $\mathcal{F}_1(U_1)$ is contained in some element of $\mathcal{F}_2(U_2)$.

Intuitively, $\mathcal{F}_1$ is finer than $\mathcal{F}_2$ if the images of $\mathcal{F}_1$ are smaller than those of $\mathcal{F}_2$. Indeed, it follows immediately from the definition that if $\mathcal{F}_1$ is finer than $\mathcal{F}_2$ then $|\mathcal{F}_1(U_1)| \subset |\mathcal{F}_2(U_2)|$ for all $U_1 \in \mathcal{U}_1$ and $U_2 \in \mathcal{U}_2$ such that $U_1 \subset U_2$. A key point of our approach is to develop some dynamical notions which are actually computable. In particular, to take advantage of the notions of combinatorial maps we need to be able to compute such maps.

Definition 2.12. f is *ε-computable* if there exists a cover $\mathcal{U}$ of X and a method for computing a combinatorial representation $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$ of f with $\mathcal{R}^+(\mathcal{F}) \leq \varepsilon$.

This essentially depends on how explicitly we know the map f and is generally not a serious issue: see Remark 2.17. We conclude this section with a series of remarks concerning the definitions above.

Remark 2.13. Unlike in the case of covers, the relation of being finer does not define a partial order between combinatorial maps. Consider the following example: $X = \{a, b, c\}$ with the discrete topology, $\mathcal{U}_1 = \big\{\{a\}, \{b\}, \{c\}\big\}$, $\mathcal{U}_2 = \big\{\{a, b\}, \{c\}\big\}$, $\mathcal{U}_3 = \big\{\{a, b\}, \{b, c\}\big\}$, $\mathcal{F}_1 \colon \{a\} \mapsto \big\{\{a\}\big\}, \{b\} \mapsto \big\{\{a\}\big\}, \{c\} \mapsto \big\{\{c\}\big\}$, $\mathcal{F}_2 \colon \{a, b\} \mapsto \big\{\{a, b\}\big\}, \{c\} \mapsto \big\{\{c\}\big\}$, $\mathcal{F}_3 \colon \{a, b\} \mapsto \big\{\{a, b\}\big\}, \{b, c\} \mapsto \big\{\{b, c\}\big\}$. Obviously, $\mathcal{U}_1 \prec \mathcal{U}_2 \prec \mathcal{U}_3$. Moreover, $\mathcal{F}_1$ is finer than $\mathcal{F}_2$ and $\mathcal{F}_2$ is finer than $\mathcal{F}_3$. However, $\mathcal{F}_1$ is not finer than $\mathcal{F}_3$, because for $U_1 := \{b\} \subset U_2 := \{b, c\}$ the element $\{a\} \in \mathcal{F}_1(U_1)$ is not contained in any element of $\mathcal{F}_2(U_2) = \big\{\{b, c\}\big\}$.

Remark 2.14. Every map $f \colon X \to X$ has a unique *minimal representation* relative to a given cover $\mathcal{U}$ defined by

$$\mathcal{F}_{f,\mathcal{U}}(U) := \{W \in \mathcal{U} : W \cap f(U) \neq \emptyset\}.$$

This is the obvious "abstract" definition of a representation of f which would be natural if it did not have to be explicitly and rigorously computed. It is minimal in the sense that the image of any $U \in \mathcal{U}$ by any other representation $\mathcal{F}$ of f contains $\mathcal{F}_{f,\mathcal{U}}(U)$. We emphasize, however, that although *some* representation $\mathcal{F}$ of f can usually be computed in a relatively straightforward manner, the *minimal* representation for the same map need not be computable at all. This is because, in general, when using rigorous numerical methods, the best one can typically compute is some outer approximation B_U^f of the image $f(U)$ of each $U \in \mathcal{U}$. Therefore, it is usually possible to compute some $\mathcal{W}_U^f$ containing all the elements of $\mathcal{U}$ which intersect B_U^f, good for constructing a representation of f. However, in general it may be impossible to determine which of the elements of $\mathcal{W}_U^f$ actually do intersect $f(U)$ and which do not. This justifies our definition of a representation of f in which we do not require that it is actually the *minimal* representation, and we allow that $\mathcal{F}(U)$ contains some superfluous elements of $\mathcal{U}$.

Remark 2.15. The composition of combinatorial maps can be easily defined. Formally, let $2^{\mathcal{U}}$ denote the set of all possible subsets of $\mathcal{U}$ and define the map $\widehat{\mathcal{F}} \colon 2^{\mathcal{U}} \to 2^{\mathcal{U}}$ as $\widehat{\mathcal{F}}(A) = \bigcup_{U \in A} \mathcal{F}(U)$ for any $A \in 2^{\mathcal{U}}$. Then for all $n \geq 1$ we can write $\mathcal{F}^n = \widehat{\mathcal{F}}^n \circ i$, where $i \colon \mathcal{U} \to 2^{\mathcal{U}}$ is the "embedding" $i(U) := \{U\}$. In the same way we can define the composition of two different combinatorial maps $\mathcal{F}$ and $\mathcal{G}$ as long

as they are both defined on the same cover. It is easy to see that if $\mathcal{F}$ and $\mathcal{G}$ are representations of $f, g\colon X \to X$, respectively, then $\mathcal{G}\circ\mathcal{F}$ is a representation of $g\circ f$. In particular, if $\mathcal{F}$ is a representation of f then $\mathcal{F}^n$ is a representation of f^n for every $n > 0$.

Remark 2.16. Note that if $\mathcal{F}_1, \mathcal{G}_1 \colon \mathcal{U}_1 \multimap \mathcal{U}_1$ are finer than $\mathcal{F}_2, \mathcal{G}_2 \colon \mathcal{U}_2 \multimap \mathcal{U}_2$, respectively, then $\mathcal{G}_1 \circ \mathcal{F}_1$ is finer than $\mathcal{G}_2 \circ \mathcal{F}_2$. In particular, if $\mathcal{F}_1$ is finer than $\mathcal{F}_2$ then also the same relation holds for iterations of these maps, that is, $\mathcal{F}_1^n$ is finer than $\mathcal{F}_2^n$. Indeed, take any $U_1 \in \mathcal{U}_1$ and any $U_2 \in \mathcal{U}_2$ such that $U_1 \subset U_2$. Take any $W_1 \in \mathcal{G}_1(\mathcal{F}_1(U_1))$. There exists $V_1 \in \mathcal{F}_1(U_1)$ such that $W_1 \in \mathcal{G}_1(V_1)$. Since $\mathcal{F}_1$ is finer than $\mathcal{F}_2$, and $U_1 \subset U_2$, for this $V_1 \in \mathcal{F}_1(U_1)$ there exists some $V_2 \in \mathcal{F}_2(U_2)$ such that $V_1 \subset V_2$. Since $\mathcal{G}_1$ is finer than $\mathcal{G}_2$, and $V_1 \subset V_2$, for this $W_1 \in \mathcal{G}_1(V_1)$ there exists some $W_2 \in \mathcal{G}_2(V_2)$ such that $W_1 \subset W_2$. As a consequence, $W_1 \subset W_2$, where $W_2 \in \mathcal{G}_2(\mathcal{F}_2(U_2))$. Note that this proof does not go through if we replace "and every $U_2 \in \mathcal{U}_2$" with "there exists $U_2 \in \mathcal{U}_2$" in Definition 2.11.

Remark 2.17. We make some brief remarks on the computability of combinatorial maps. If $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is a continuous map defined by means of an explicit formula involving only elementary operations (addition, multiplication, etc.) and simple arithmetic functions (like the trigonometric functions, $\sqrt{x}$, or e^x), and an open bounded set $X \subset \mathbb{R}^n$ can be found such that $f(X) \subset X$, then one can compute a finite representation of $f\colon X \to X$ using the concept of interval analysis [29]. For a cover $\mathcal{U}$ of X that consists of products of open intervals, a set B_U^f containing $f(U)$ can be computed for each $U \in \mathcal{U}$, where B_U^f is also a product of open intervals. Then we define $\mathcal{W}_U^f$ to be the union of those elements of $\mathcal{U}$ which intersect B_U^f (such a set can be computed easily). The multivalued map $\mathcal{F}\colon \mathcal{U} \ni U \mapsto \mathcal{W}_U^f \subset \mathcal{U}$ is then a combinatorial representation of f on X. Obviously, the smaller the elements of the cover $\mathcal{U}$ are taken, the smaller the outer resolution of the constructed representation should be expected. This argument proves that a finite representation of dynamics for a wide class of maps is computable at virtually any resolution. We shall use the idea described above in Section 5 for the analysis of the Hénon map.

2.3. **The axioms of finite resolution dynamics.** We are now ready to formalize the axioms of computability and consistency introduced in Section 1, based on the notion of combinatorial maps. Let $\mathcal{P} := \mathcal{P}(\mathcal{F})$ be a predicate concerning a combinatorial map $\mathcal{F}$.

Definition 2.18. We say that $\mathcal{P}$ is a *finite resolution property* if it satisfies the following two conditions:

- **(A) Computability:** $\mathcal{P}$ is *computable*, that is, there exists an algorithm which for any combinatorial map $\mathcal{F}$ can establish in finite time whether $\mathcal{F}$ satisfies $\mathcal{P}$ or not.
- **(B) Consistency:** $\mathcal{P}$ is *consistent*, that is, for any pair of combinatorial maps $\mathcal{F}_1$ and $\mathcal{F}_2$ such that $\mathcal{F}_1$ is finer than $\mathcal{F}_2$, we have $\mathcal{P}(\mathcal{F}_1) \implies \mathcal{P}(\mathcal{F}_2)$.

We remark that the definition is given purely in terms of combinatorial maps with no reference to the underlying space or any map whose representation $\mathcal{F}$ might be. Thus we have an *a priori* notion of what it means for a property $\mathcal{P}$ to be an "acceptable" property for the investigation of finite resolution dynamics. In general we will apply this definition to combinatorial maps which arise as representations of some particular map $f\colon X \to X$ on a metric space X and thus obtain some coherent statements about the "finite resolution" dynamics of the map f.

We make here a simple observation which is crucial to motivate the theory. Part (B) in the definition above is not in itself sufficient to define what it means for the property $\mathcal{P}$ to hold at "all resolutions $> \varepsilon$". Indeed, assuming that $\mathcal{P}$ holds for some

combinatorial representation $\mathcal{F}$ with respect to some cover $\mathcal{U}$ with outer resolution $\mathcal{R}^+(\mathcal{F}) \leq \varepsilon$, it is easy to construct another combinatorial representation $\tilde{\mathcal{F}}$ with $\mathcal{R}^+(\tilde{\mathcal{F}}) > \varepsilon$ but which is not coarser than $\mathcal{F}$ and therefore is not automatically guaranteed to satisfy $\mathcal{P}$. In other words, choosing a cover with a larger resolution does not necessarily guarantee that the cover is coarser, and therefore the consistency condition does not provide any information. We need some notion which guarantees that $\mathcal{P}$ holds for **all** combinatorial maps at sufficiently coarse scales, based on the fact that $\mathcal{P}$ has been verified at some scale for a **single** particular choice of combinatorial map. To formulate this notion we need to introduce an additional definition.

2.4. **Inner Resolution.** The following definition and its application constitute in some sense the key idea of this paper.

Definition 2.19. The *inner resolution* of a cover $\mathcal{U}$ is

$$\mathcal{R}^-(\mathcal{U}) := \sup\{d \geq 0 : \forall x \in X \; \exists U \in \mathcal{U} : B(x,d) \subset U\}.$$

The *inner resolution* of a combinatorial map $\mathcal{F}\colon \mathcal{U} \multimap \mathcal{U}$ is

$$\mathcal{R}^-(\mathcal{F}) := \mathcal{R}^-(\mathcal{U}).$$

Positive inner resolution is a crucial feature that distinguishes our approach from one based on partitions, where the inner resolution is 0.

We have used the convention here that $B(x,0) = \{x\}$ so that the inner resolution is the supremum of the numbers $d > 0$ such that every ball $B(x,d) \subset X$ is contained in some $U \in \mathcal{U}$, or 0 if such $d > 0$ does not exist. In a connected space, the quantity $\mathcal{R}^-(\mathcal{U})$ can be interpreted as the minimal width of overlapping between adjacent elements of $\mathcal{U}$. Intuitively, if the distance between two points $x, y \in X$ is smaller than the inner resolution of a cover then these points can be identified by means of belonging to a common element of the cover.

Remark 2.20. If X is compact then we can apply Lebesgue's number lemma to any open cover $\mathcal{U}$ of X in order to know that there exists a number $\delta = \delta(\mathcal{U}) > 0$ such that every subset of X whose diameter does not exceed δ is contained in some element of the cover. Then obviously $\mathcal{R}^-(\mathcal{U}) \geq \delta/2 > 0$.

Remark 2.21. In certain cases, e.g., if X is a bounded subset of a Euclidean space, for any $\varepsilon \geq \delta > 0$ it is possible to construct a cover $\mathcal{U}$ of X such that $\mathcal{R}^+(\mathcal{U}) \leq 2\varepsilon$ and $\mathcal{R}^-(\mathcal{U}) \geq \delta$. In particular, this is true if there exists a finite number of points $\{x_1, \ldots, x_n\} \subset X$ such that $\mathcal{U}_0 := \{B(x_i, \varepsilon - \delta) : i = 1, \ldots, n\}$ is a cover of X. Indeed, let us consider $\mathcal{U} := \{B(x_i, \varepsilon) : i = 1, \ldots, n\}$. Obviously, $\mathcal{U}$ is a cover of X and $\mathcal{R}^+(\mathcal{U}) \leq 2\varepsilon$. Moreover, every $x \in X$ is within the distance of $r < \varepsilon - \delta$ from x_k for some $k \in \{1, \ldots, n\}$, and thus $B(x, \delta) \subset B(x_k, \varepsilon)$ by the triangle inequality, so $\mathcal{R}^-(\mathcal{U}) \geq \delta$.

Remark 2.22. The inner resolution also provides a lower bound for how "thick" the cover elements are in terms of containing a ball of a big enough radius. Define $\mathcal{T}(\mathcal{U}) := \sup\{d > 0 : \forall U \in \mathcal{U} \; \exists x \in X : B(x,d) \subset U\}$. (See Figure 2 for an elementary example.) It is easy to see that for an essential cover $\mathcal{U}$ of X we have $\mathcal{R}^-(\mathcal{U}) \leq \mathcal{T}(\mathcal{U})$. Indeed, take any positive number $d < \mathcal{R}^-(\mathcal{U})$. Take any $U \in \mathcal{U}$. By the assumption that $\mathcal{U}$ is essential, there exists $x \in U$ such that for some $\varepsilon > 0$ the ball $B(x, \varepsilon)$ is contained in U and disjoint from all the other elements of the cover $\mathcal{U}$. Since $d < \mathcal{R}^-(\mathcal{U})$, for this particular x there exists some $U' \in \mathcal{U}$ such that $B(x,d) \subset U'$. However, U is the only element of $\mathcal{U}$ that contains x, so clearly $U' = U$, and thus $B(x,d) \subset U$. This shows that for every $U \in \mathcal{U}$ there exists an

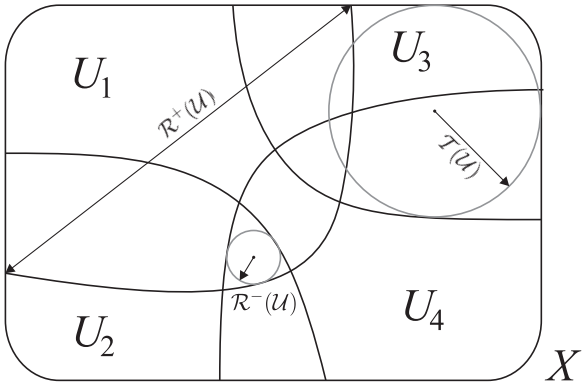

FIGURE 2. An illustration of the quantities $\mathcal{R}^+(\mathcal{U})$, $\mathcal{R}^-(\mathcal{U})$ in Definitions 2.2 and 2.19, and $\mathcal{T}(\mathcal{U})$ in Remark 2.22, for a sample cover $\mathcal{U} = \{U_1, U_2, U_3, U_4\}$.

$x \in \mathcal{U}$ such that $B(x, d) \subset U$. Since $d < \mathcal{R}^-(\mathcal{U})$ can be arbitrarily close to $\mathcal{R}^-(\mathcal{U})$, the supremum of these numbers is greater than or equal to $\mathcal{R}^-(\mathcal{U})$.

2.5. **Finite resolution properties for all resolutions $> \varepsilon$.** We are now ready to state precisely what we mean when we say that a property holds at all resolutions coarser than some ε, and to state and prove our main abstract theorem. Let $\mathcal{P}$ be a finite resolution property and let $f\colon X \to X$ be a map for which there exists a representation whose outer resolution is $\leq \varepsilon$.

Definition 2.23. *f satisfies $\mathcal{P}$ at all resolutions $> \varepsilon$* if $\mathcal{P}$ is satisfied for every representation $\mathcal{F}\colon \mathcal{U} \multimap \mathcal{U}$ of f with $\mathcal{R}^-(\mathcal{F}) > \varepsilon$.

We are using here the notion of inner resolution to quantify the resolution of a cover, but recall from Remark 2.21 that in many cases the inner and outer resolution can be chosen arbitrarily close (up to the factor of 2). Notice also that the definition requires $\mathcal{P}$ to hold for *any* cover $\mathcal{U}$ and *any* combinatorial representation $\mathcal{F}\colon \mathcal{U} \multimap \mathcal{U}$ as long as $\mathcal{R}^-(\mathcal{F}) > \varepsilon$. It is therefore *a priori* an unverifiable condition. However, we have the following

Theorem 2. *Suppose $\mathcal{P}$ is a finite resolution property. If $\mathcal{P}$ holds for a representation $\mathcal{F}_0$ of f with $\mathcal{R}^+(\mathcal{F}_0) \leq \varepsilon$ then $\mathcal{P}$ holds for all representations $\mathcal{F}$ of f with $\mathcal{R}^-(\mathcal{F}) > \varepsilon$. Thus f satisfies $\mathcal{P}$ at all resolutions $> \varepsilon$.*

Proof. We present the proof in the form of two lemmas.

Lemma 2.24. *Let $\mathcal{U}_1$ and $\mathcal{U}_2$ be two covers of X. If $\mathcal{R}^+(\mathcal{U}_1) < \mathcal{R}^-(\mathcal{U}_2)$ then $\mathcal{U}_1 \prec \mathcal{U}_2$.*

Proof. Take any number r such that $\mathcal{R}^+(\mathcal{U}_1) < r < \mathcal{R}^-(\mathcal{U}_2)$. Let $U_1 \in \mathcal{U}_1$. Consider a ball $B(x, r)$ for some $x \in U_1$. From the fact that $r > \mathcal{R}^+(\mathcal{U}_1)$ it follows that $U_1 \subset B(x, r)$. Since $r < \mathcal{R}^-(\mathcal{U}_2)$, there exists $U_2 \in \mathcal{U}_2$ such that $B(x, r) \subset U_2$, and thus $U_1 \subset U_2$. Now consider any $U_2 \in \mathcal{U}_2$. Then (by Remark 2.22) there exists a ball $B(x, r) \subset U_2$. Since $\mathcal{U}_1$ is a cover of X, there exists $U_1 \in \mathcal{U}_1$ such that $x \in U_1$. Since $\operatorname{diam} U_1 < r$, we have $U_1 \subset B(x, r)$, and thus $U_1 \subset U_2$. □

Lemma 2.25. *If $\mathcal{U}_1$ and $\mathcal{U}_2$ are two covers of X, $\mathcal{F}_1\colon \mathcal{U}_1 \multimap \mathcal{U}_1$ is a representation of some map $f\colon X \to X$, and $\mathcal{R}^-(\mathcal{U}_2) > \mathcal{R}^+(\mathcal{F}_1)$, then any representation $\mathcal{F}_2\colon \mathcal{U}_2 \multimap \mathcal{U}_2$ of f is coarser than $\mathcal{F}_1$.*

Proof. By Lemma 2.24, we have $\mathcal{U}_1 \prec \mathcal{U}_2$. Take any $U_1 \in \mathcal{U}_1$ and any $U_2 \in \mathcal{U}_2$ such that $U_1 \subset U_2$. Take $x \in U_1$. Take a number r such that $\mathcal{R}^+(\mathcal{F}_1) < r <$

$\mathcal{R}^-(\mathcal{U}_2)$. Then there exists $W_2 \in \mathcal{U}_2$ such that $B(f(x), r) \subset \mathcal{W}_2$. Since $\mathcal{F}_2$ is a representation of f and $x \in U_2$, $\mathcal{F}_2(U_2)$ contains all the elements of $\mathcal{U}_2$ which contain $f(x)$, including W_2. On the other hand, since $\operatorname{diam} |\mathcal{F}_1(U_1)| < r$, every $W_1 \in \mathcal{F}_1(U_1)$ is contained in $B(f(x), r) \subset W_2$. This proves that indeed $\mathcal{F}_2$ is coarser than $\mathcal{F}_1$. □

The statement in the Theorem now follows immediately from Lemma 2.25 and the definitions. □

Remark 2.26. Lemmas 2.24 and 2.25 show a tremendous advantage of using open covers as opposed to partitions, because there is no easy condition defined in terms of the features of a partition itself which would guarantee that this partition and any map on it are coarser than some other partition and some other map, respectively, both possibly given *a priori* (e.g., as a result of some computation, as we do in Section 5).

Remark 2.27. If one prefers to do computations for partitions instead of using open covers, then $\mathcal{F}_0$ in the assumptions of Theorem 2 can in fact be a multivalued map on a partition. Definition 2.10 of the outer resolution in this case can be applied directly, Definition 2.9 of a representation can be taken the same and is equivalent to requesting that $f(U) \subset \operatorname{int} |\mathcal{F}(U)|$, and the predicate $\mathcal{P}$ can also be the same as for a combinatorial map on a cover. Note, however, that the conclusion of Theorem 2 is not valid for partitions, because their inner resolution is 0.

3. Transitivity and mixing

Let $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$ be a combinatorial map, and let $\mathcal{F}^{-1}$ denote its inverse defined by the following condition: $V \in \mathcal{F}^{-1}(U)$ if and only if $U \in \mathcal{F}(V)$. We introduce the notions of transitivity and mixing for combinatorial maps, further also called combinatorial transitivity and combinatorial mixing, respectively.

Definition 3.1. $\mathcal{F}$ is *transitive* if for every $U, V \in \mathcal{U}$ there exists $n > 0$ such that $V \in \mathcal{F}^{-n}(U)$. $\mathcal{F}$ is *mixing* if for every $U, V \in \mathcal{U}$ there exists $N > 0$ such that $V \in \mathcal{F}^{-n}(U)$ for all $n > N$.

We shall prove the following

Proposition 3.2. *Combinatorial transitivity and combinatorial mixing are finite resolution properties.*

To prove the proposition we need to prove the computability condition (A) and the consistency condition (B) in the definition of finite resolution property. Condition (A) says that the properties of (combinatorial) transitivity and mixing are algorithmically verifiable in finite time. This is almost immediate from the definitions and Lemma 3.5 below, since combinatorial transitivity and mixing only require checking a finite number of conditions (because $\mathcal{U}$ is finite); however, in Section 4 we provide explicit algorithms which can be used to verify the conditions in an efficient way in practice. In this section we prove (B).

Notice first that mixing implies transitivity but the converse is not true in general. Moreover, since $V \in \mathcal{F}^{-n}(U)$ is equivalent to $U \in \mathcal{F}^n(V)$, one can equivalently formulate the combinatorial conditions for the n-th iterate of $\mathcal{F}$ instead of the n-th preimage of $\mathcal{F}$, thanks to the symmetry between U and V in these conditions. We shall use this easy observation transparently in the sequel. We shall consider transitivity and mixing separately.

Lemma 3.3. *Combinatorial transitivity is a consistent property.*

Proof. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ be two covers of X, with $\mathcal{U}_1$ finer than $\mathcal{U}_2$. Let $\mathcal{F}_1 \colon \mathcal{U}_1 \multimap \mathcal{U}_1$ and $\mathcal{F}_2 \colon \mathcal{U}_2 \multimap \mathcal{U}_2$ be combinatorial maps. Assume that $\mathcal{F}_1$ is finer than $\mathcal{F}_2$ and that $\mathcal{F}_1$ is transitive. We shall show that under these assumptions also $\mathcal{F}_2$ is transitive. Let $U_2, V_2 \in \mathcal{U}_2$. Let B be a ball contained in U_2 and disjoint from all the other elements of $\mathcal{U}_2$; the existence of such a ball follows from the fact that $\mathcal{U}_2$ is essential. Let $U_1, V_1 \in \mathcal{U}_1$ be such that $U_1 \cap B \neq \emptyset$ and $V_1 \subset V_2$. Obviously, in this case $U_1 \subset U_2$. By the assumption on transitivity of $\mathcal{F}_1$, there exists $n > 0$ such that $U_1 \in \mathcal{F}_1^n(V_1)$. Since $|\mathcal{F}_1^n(V_1)| \subset |\mathcal{F}_2^n(V_2)|$, we know that $U_1 \subset |\mathcal{F}_2^n(V_2)|$. In particular, $U_1 \cap B \subset |\mathcal{F}_2^n(V_2)|$, and therefore $U_2 \in \mathcal{F}_2^n(V_2)$, because otherwise no part of B could have been covered by $\mathcal{F}_2^n(V_2)$. Since the choice of $U_2, V_2 \in \mathcal{U}_2$ was arbitrary, this proves that $\mathcal{F}_2$ is transitive. □

Lemma 3.4. *Combinatorial mixing is a consistent property.*

Before proving Lemma 3.4, we first prove two simple lemmas regarding an equivalent condition for mixing of combinatorial maps.

Lemma 3.5. *Let $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$ be a combinatorial map, and let $k > 0$. If $\mathcal{F}^k(U) = \mathcal{U}$ for all $U \in \mathcal{U}$ then also $\mathcal{F}^n(U) = \mathcal{U}$ for all $U \in \mathcal{U}$ and for all $n > k$.*

Proof. This follows by induction from the fact that $\mathcal{F}^{k+1}(U) = \bigcup_{V \in \mathcal{F}(U)} \mathcal{F}^k(V)$ and that $\mathcal{F}(U) \neq \emptyset$ and $\mathcal{F}^k(V) = \mathcal{U}$. □

Lemma 3.6. *Let $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$ be a combinatorial map. Then $\mathcal{F}$ is mixing if and only if there exists $k > 0$ such that $\mathcal{F}^k(U) = \mathcal{U}$ for all $U \in \mathcal{U}$.*

Proof. Let us first assume that $\mathcal{F}$ is mixing. Then for every $U, V \in \mathcal{U}$ there exists $N_{U,V}$ such that $V \in \mathcal{F}^n(U)$ for all $n > N_{U,V}$. Since $\mathcal{U}$ is finite, the number $k := \max\{N_{U,V} : U, V \in \mathcal{U}\}$ is well defined and finite. Obviously, for this k, the set $\mathcal{F}^k(U)$ contains every $V \in \mathcal{U}$, for all $U \in \mathcal{U}$.

Now suppose that $\mathcal{F}^k(U) = \mathcal{U}$ for all $U \in \mathcal{U}$ and some $k > 0$. Lemma 3.5 implies that $\mathcal{F}^n(U) = \mathcal{U}$ for all $U \in \mathcal{U}$ and all $n \geq k$, which immediately implies that $\mathcal{F}$ is mixing. □

Proof of Lemma 3.4. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ be two covers of X, with $\mathcal{U}_1$ finer than $\mathcal{U}_2$. Let $\mathcal{F}_1 \colon \mathcal{U}_1 \multimap \mathcal{U}_1$ and $\mathcal{F}_2 \colon \mathcal{U}_2 \multimap \mathcal{U}_2$ be combinatorial maps. Assume that $\mathcal{F}_1$ is finer than $\mathcal{F}_2$ and that $\mathcal{F}_1$ is mixing. We shall show that under these assumptions also $\mathcal{F}_2$ is mixing. Take $k > 0$ such that $\mathcal{F}_1^k(U) = \mathcal{U}_1$ for all $U \in \mathcal{U}_1$, given by Lemma 3.6. Take any $W \in \mathcal{U}_2$ and consider $\mathcal{F}_2^k(W)$. Since $\mathcal{U}_1$ is finer than $\mathcal{U}_2$, there exists some $U \in \mathcal{U}_1$ such that $U \subset W$. Since $\mathcal{F}_1$ is finer than $\mathcal{F}_2$, it follows that $|\mathcal{F}_1^k(U)| \subset |\mathcal{F}_2^k(W)|$. The assumption that $\mathcal{U}_2$ is essential implies that $\mathcal{F}_2^k(W) = \mathcal{U}_2$. Since the choice of $W \in \mathcal{U}_2$ was arbitrary, it follows from Lemma 3.6 that $\mathcal{F}_2$ is mixing. □

4. Graph algorithms

In this section we provide explicit algorithms for the verification of combinatorial transitivity and mixing. In particular, we give a constructive proof of (A). The first step is to translate the properties defined in Section 3 into the language of graphs associated with combinatorial maps.

We recall that a *finite directed graph*, further called *graph* for short, is a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set whose elements are called *vertices*, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of selected (ordered) pairs of vertices. The elements of $\mathcal{E}$ are called *edges*. Combinatorial maps are very naturally encoded as graphs.

Definition 4.1. We say that $G = (\mathcal{V}, \mathcal{E})$ is the graph *associated* with a combinatorial map $\mathcal{F}$ on a cover $\mathcal{U}$ of X if $\mathcal{V} = \mathcal{U}$ and $\mathcal{E} = \{(U, V) \in \mathcal{U} \times \mathcal{U} : V \in \mathcal{F}(U)\}$.

In this section we explain how the verification of the properties of combinatorial transitivity and mixing can be reduced to the verification of certain properties of the associated graphs. We then describe in some detail specific algorithms that verify these properties, and we upper provide bounds for their effectiveness. Before we show such algorithms, let us explain how their effectiveness is measured. A running time estimate of an algorithm is typically given by means of the order of the number of required primitive operations as a function of the size of the input, here given by $|\mathcal{V}|$, the size of the vertex set, and $|\mathcal{E}|$, the size of the edge set. We use the notation $O\big(\Psi(|\mathcal{V}|,|\mathcal{E}|)\big)$ to indicate that there exist constants $c, n_0 > 0$ such that for any graph $G = (\mathcal{V}, \mathcal{E})$ for which $|\mathcal{V}|, |\mathcal{E}| \geq n_0$, the number of operations $\Phi(G)$ of the algorithm applied to the graph G satisfies the inequality $\Phi(G) \leq c\Psi(|\mathcal{V}|, |\mathcal{E}|)$. 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 [11, §3.1]. In what follows, we say that an algorithm runs in *linear* time if its worst case running time is $O\big(|\mathcal{V}| + |\mathcal{E}|\big)$.

Proposition 4.2. *There exists an algorithm which verifies in linear time whether a combinatorial map is transitive or not. There exists an algorithm which verifies in linear time whether a combinatorial map is mixing or not.*

Notice that the size $|\mathcal{V}|$ of the vertex set is exactly the number of elements of the cover $\mathcal{U}$. Moreover, in many situations, such as if the map f is Lipschitz and the elements of $\mathcal{U}$ are of similar size, the expected size $\mathcal{F}(U)$ of the image of each element U, is uniformly bounded, independently of the resolution of the cover. Therefore, in practice, the size $|\mathcal{E}|$ of the edge set is bounded by a constant multiple of the size $|\mathcal{V}|$ of the vertex set, which is exactly the number of elements in the cover. Thus the algorithms we describe, in many situations, actually run in linear time in the number of the elements of the cover $\mathcal{U}$. We consider this linear time property as an important characteristic of our method since it implies that as computing memory and power increase, it will be realistically possible to obtain significant improvements to the scale at which mixing is verified in systems of interest, such as the Hénon map which we consider in this paper (see, however, additional remarks on this point in Section 5.2.3).

The remaining part of this section is devoted to the proof of Proposition 4.2. In particular we shall obtain completely explicit forms of the required algorithms. In Section 4.1 we show that the problem can be formulated in terms of the verification of certain properties of directed graphs, namely strong connectedness and aperiodicity. The algorithms for verifying these properties are not new. An algorithm that computes strongly connected components of a directed graph in linear time belongs to the canons of graph algorithms (see [11, §22.5]), and we use it below to determine whether a given graph is strongly connected. An algorithm for proving aperiodicity is given in [48] but without a formal proof of its correctness, which we shall give here. For completeness, and for the benefit of readers who are not familiar with graph algorithms or algorithms in general, we shall give full descriptions of these algorithms below. They actually require some non-trivial constructions which we believe are of independent, albeit relatively technical, interest. In particular, we provide a fully self-contained proof of Proposition 4.2.

In Section 4.2 we discuss the representation of graphs in computer's memory. In Section 4.3 we describe a basic way of "exploring" graphs algorithmically and of describing certain structures of graphs. This approach, called "depth first search", underpins both the method for computing strongly connected components described in Section 4.4, and the method for verifying aperiodicity described in Section 4.5.

4.1. **Strongly connected and aperiodic graphs.** We now show how the notions of combinatorial transitivity and mixing can be reduced to certain properties of the associated graphs. A *path* in a graph $G = (\mathcal{V}, \mathcal{E})$ is a sequence $(U_i)_{i=0}^n$ such that $(U_i, U_{i+1}) \in \mathcal{E}$ for all $i = 0, \dots, n-1$.

Definition 4.3. We say that a graph $G = (\mathcal{V}, \mathcal{E})$ is *strongly connected* if for every $U, V \in \mathcal{V}$ there exists a path $(v_0, \dots, v_k)$ in G such that $v_0 = U$ and $v_k = V$.

The *length* of a path $(U_i)_{i=0}^n$ is n. A *cycle* in G is a path in which $U_0 = U_n$. The greatest common divisor of the lengths of all the cycles in a graph $G = (\mathcal{V}, \mathcal{E})$ is called the *period* of G.

Definition 4.4. We say that a graph $G = (\mathcal{V}, \mathcal{E})$ is *aperiodic* if its period equals 1.

Lemma 4.5. *A combinatorial map is transitive if and only if the associated graph is strongly connected. A combinatorial map is mixing if and only if the associated graph is strongly connected and aperiodic.*

Proof. Since the existence of a path of length n from U to V in the graph associated with a combinatorial map $\mathcal{F}$ is equivalent to $V \in \mathcal{F}^n(U)$, the equivalence in the case of transitivity follows immediately from the definitions.

The statement on mixing can be derived, with certain effort, from [17], but for the sake of completeness we provide a detailed proof. Assume that a combinatorial map $\mathcal{F} \colon \mathcal{U} \multimap \mathcal{U}$ is mixing. Then Lemma 3.6 implies that there exists $k > 0$ such that $\mathcal{F}^k(U) = \mathcal{U}$ for all $U \in \mathcal{U}$. It is thus obvious that G is strongly connected. Lemma 3.5 implies that also $\mathcal{F}^{k+1}(U) = \mathcal{U}$ for all $U \in \mathcal{U}$. In particular, $U \in \mathcal{F}^k(U)$ and $U \in \mathcal{F}^{k+1}(U)$. In terms of the graph G, this implies that there exist cycles of the co-prime lengths k and $k+1$ in G through U, and thus G is aperiodic.

Let us now focus on the opposite implication. Let $T = (U_{j_0}, \dots, U_{j_t})$ for some $t > 0$ be a cycle in G that runs through all the vertices of G (its existence follows from the strong connectedness of G). Since the GCD of the lengths of all the cycles in G is 1, there exist cycles $C_1, \dots, C_r$ and $D_1, \dots, D_s$ of lengths $p_1, \dots, p_r$ and $q_1, \dots, q_s$, respectively, such that $p_1 + \dots + p_r - q_1 - \dots - q_s = 1$. Consider the cycles T_{kl} in G composed of T, k copies of $C_1, \dots, C_r$, and l copies of $D_1, \dots, D_s$. The length of each such cycle is $t_{kl} = t + kp + lq$, where $p := p_1 + \dots + p_r$ and $q := q_1 + \dots + q_s$. Since $p = q + 1$, we have $t_{kl} = t + (k+l)q + k$. If we fix any vertex U of G at the cycle T and consider $k = 0, \dots, t-1$ with $l = t - k$, then we obtain paths from U to U whose lengths are $t + tq + k = tp + k$. We can complement these cycles by following T to paths of length $tp + t = t(p+1)$ that end at each subsequent vertex of T. Since T runs through all the vertices of G, it follows that $\mathcal{F}^{t(p+1)}(U) = \mathcal{U}$. Since $U \in \mathcal{U}$ was chosen arbitrarily, it follows from Lemma 3.6 that $\mathcal{F}$ is mixing. □

4.2. **Representation of graphs.** The first step in developing algorithms for studying properties of a graph is to represent such a graph in computer's memory. One standard way to do this is as a collection of *adjacency lists*. We say that a vertex v is adjacent to another vertex u if the edge (u, v) is in the graph. The adjacency-list representation of a graph $G = (\mathcal{V}, \mathcal{E})$ consists of an array of $|\mathcal{V}|$ lists, one for each vertex in $\mathcal{V}$. For each $u \in \mathcal{V}$, the adjacency list of u contains all the vertices $v \in \mathcal{V}$ such that $(u, v) \in \mathcal{E}$ in certain order. Searching through this list or taking one adjacent vertex after another can be done in the time proportional to the number of vertices in the list.

Other representations, such as an "adjacency matrix" representation, can also be used and may be more or less convenient depending on the structure of the graph under investigation. As a general rule, the adjacency-list representation is preferable for *sparse* graphs, that is graphs for which $|\mathcal{E}|$ is of a lower order than

its maximum possible value of $|\mathcal{V}|^2$, e.g., if it is proportional to $|\mathcal{V}|$. This is usually the case for combinatorial maps and thus we shall use this representation of graphs here.

4.3. **The Depth First Search algorithm.** Depth First Search (DFS) is an algorithm for scanning a graph $G = (\mathcal{V}, \mathcal{E})$. It starts at an arbitrarily chosen vertex, and follows edges and paths to visit other vertices in the graph. If no more vertices can be reached in this way then an arbitrary unvisited vertex is chosen and the algorithm continues running from that vertex. Each vertex $u \in \mathcal{V}$ is assigned a unique *discovery time* $t_u \in \mathbb{N}$ which reflects the order in which the vertices are visited, or discovered. The general strategy of the algorithm while visiting a vertex is to take the first edge leading to an unvisited vertex and to follow it. If no more vertices can be discovered from a given vertex, then the algorithm traces back and checks the other edges leading from the previous vertex. The name "depth first search" comes from the fact that the search goes as deep into the graph as possible using the first edge leading to an unvisited vertex, and the other edges emanating from each vertex are only checked afterwards. This is different from another well known strategy, called "breadth first search", in which all the edges leading from each vertex are checked before the search continues from the discovered vertices. The details of the DFS procedure are summarized in the following

Algorithm 4.6.
function `DFS`
input: $G = (\mathcal{V}, \mathcal{E})$ — directed graph
code: `time` $:= 0$
for each $u \in \mathcal{V}$ **do**
if not defined t_u **then**
`DFS-Visit` (u);

function `DFS-Visit` (u)
`time` $:=$ `time` $+ 1$;
$t_u :=$ `time`;
for each $v \in \mathcal{V}$ such that $(u, v) \in \mathcal{E}$ **do**
if not defined t_v **then**
`DFS-Visit` (v);

This algorithm runs in linear time $O\big(|\mathcal{V}| + |\mathcal{E}|\big)$, because the function `DFS-Visit` is called exactly once for each vertex in the graph, and all the edges emanating from each vertex are also checked exactly once. Note that in order to achieve the linear time in practice, it is necessary to represent the graph in such a way that scanning through the set of all the edges emanating from a given vertex can be done in the time proportional to the number of these edges, like in the case of the adjacency-list representation of a graph. The reader is referred to [11, §22.3] for more details on the DFS algorithm.

4.4. **Strongly connected components.** One of many applications of the depth first search algorithm is a method for finding the strongly connected components of a graph.

The most standard approach (see [11, §22.5]) is to run the DFS algorithm on the graph G, and then to compute depth first trees of G^T (the transpose of G, obtained from G by inverting the direction of all the edges), which turn out to form the strongly connected components of G. Although this algorithm runs in linear time and memory, it is inconvenient if applied to huge graphs because of the need to compute G^T, which takes up as much memory as G itself, and thus effectively doubles memory usage.

Because of this reason, we use Tarjan's algorithm instead for computing strongly connected components of a graph (see [43,49]). This algorithm is based directly upon DFS. In addition to computing the discovery time t_u for each visited vertex, it also computes the lowest discovery time l_u of all the vertices reachable from that vertex. This number is computed during the DFS itself, so that it is always known when needed. Moreover, during the DFS run, the visited vertices are put on a stack S, and each strongly connected component C_k is taken from the stack whenever a visited vertex is determined to "close" such a component. For the sake of completeness, we provide a pseudocode of this algorithm below, but we refer to [43] for the proof of its correctness and for more explanations.

Algorithm 4.7.
function `SCC`
input: $G = (\mathcal{V}, \mathcal{E})$ — directed graph;
code: `time` $:= 0$; $k := 0$; $S :=$ empty stack;
for each $u \in \mathcal{V}$ **do**
if not defined t_u **then**
`tarjan` (u);
return $(C_1, \ldots, C_k)$

function `tarjan` (u)
`time` $:=$ `time` $+ 1$; $t_u := l_u :=$ `time`;
put u on the top of the stack S;
for each $v \in \mathcal{V}$ **such that** $(u, v) \in \mathcal{E}$ **do**
if not defined t_v **then**
`tarjan` (v);
$l_u := \min(l_u, l_v)$;
else if v is in S **then**
$l_u := \min(l_u, t_v)$;
if $t_u = l_u$ **then**
$k := k + 1$;
remove the elements from the top of S until u has been
removed, too, and put all of the removed elements into C_k;

With this algorithm, we now determine whether a given graph is strongly connected in the most obvious way possible. Namely, we compute the strongly connected components and check if there is only one such component, as made precise below.

Algorithm 4.8.
function `StronglyConnected`
input: $G = (\mathcal{V}, \mathcal{E})$ — directed graph;
code: $(C_1, \ldots, C_k) :=$ `SCC` (G);
if $k = 1$ **and** $\operatorname{card} C_1 = \operatorname{card} \mathcal{V}$ **then**
return `true`;
else
return `false`.

The following features of the above algorithm are obvious:

Lemma 4.9. *Algorithm 4.8 applied to a directed graph $G = (\mathcal{V}, \mathcal{E})$ returns* `true` *if and only if G is strongly connected. This algorithm runs in linear time.*

4.5. **Aperiodic graphs.** Efficient computation of the greatest common denominator of cycles in a strongly connected directed graph is less standard, and the solution suggested in [17] yields cubic time. However, there exists an algorithm which runs

in linear time, see [48]. For the sake of completeness, we describe this algorithm below and prove its correctness (note that the latter is not provided in [48]). This algorithm is based upon the standard DFS (Depth First Search) algorithm that scans the entire graph starting at an arbitrarily chosen vertex (recall the assumption that the graph is strongly connected), and computes the greatest common divisor of certain numbers related to the edges of the graph.

Algorithm 4.10.
function `GraphPeriod`
input: $G = (\mathcal{V}, \mathcal{E})$ — a strongly connected directed graph;
u — an element of $\mathcal{V}$;
$d_u \in \mathbb{Z}$ — the depth of u in the DFS tree being constructed;
$p \in \mathbb{Z}$ — the GCD of cycle periods found so far;
code: **for each** $v \in \mathcal{V}$ **such that** $(u, v) \in \mathcal{E}$ **do**
if defined d_v **then**
$p :=$ GCD $(p,\, d_v - d_u - 1)$;
else
`GraphPeriod` $(G,\, v,\, d_v := d_u + 1,\, p)$;
return p.

Lemma 4.11. *Algorithm 4.10 applied to a strongly connected directed graph $G = (\mathcal{V}, \mathcal{E})$, any element $r \in \mathcal{V}$, and the numbers $d_r := 0$ and $p := 0$, returns the greatest common divisor of the lengths of all the cycles in G. This algorithm runs in linear time.*

Proof. Given a strongly connected graph $G = (\mathcal{V}, \mathcal{E})$, let $T = (\mathcal{V}, \mathcal{E}')$ denote the tree obtained by running Algorithm 4.10 on G, starting with the vertex $r \in \mathcal{V}$, which becomes the root of T. The set $\mathcal{E}'$ consists of all the edges which incur recursive calls of the function `GraphPeriod`. The computed depth of each vertex $u \in \mathcal{V}$ in T is denoted by d_u, with $d_r = 0$. For each edge $e = (u, v) \in \mathcal{E} \setminus \mathcal{E}'$, define $d_e := d_v - d_u - 1$. Let c be the GCD (greatest common divisor) of all these numbers d_e. This is the quantity returned by this algorithm. We shall prove that c equals the GCD of the lengths of all the cycles in G, further denoted by c'.

Let us first prove that $c'|c$. Consider $e = (u, v) \in \mathcal{E} \setminus \mathcal{E}'$. Since G is strongly connected, there exists a path $p_{v,r}$ in G from v to r; denote its length by d. Note that the path $p_{r,u}$ in T from r to u is of length d_u, and the path $p_{r,v}$ in T from r to v is of length d_v. Since $p_{r,u}$ combined with e and $p_{v,r}$ is a cycle in G of length $d_u + 1 + d$, and $p_{r,v}$ combined with $p_{v,r}$ is a cycle in G of length $d_v + d$, the common divisor c' of the lengths of all the cycles in G must divide the difference between these lengths, which is $d_v - d_u - 1$. Therefore, c is a GCD of numbers divisible by c', and thus $c'|c$.

Let us now prove that $c|c'$. Let $(v_0, \ldots, v_n)$ be any cycle in G. We shall prove that $c|n$. Consider the depths of each v_i in T. The difference in the depth traversed by each edge $e_i = (v_i, v_{i+1})$ is $\delta_i := d_{v_{i+1}} - d_{v_i}$, which equals 1 if $e_i \in \mathcal{E}'$. Since $v_0 = v_n$, obviously $d_{v_0} = d_{v_n}$, and thus $\sum_{i=0}^{n-1} \delta_i = 0$. In particular, $-n = \sum_{i=0}^{n-1} (\delta_i - 1) = \sum_{i=0}^{n-1} (d_{v_{i+1}} - d_{v_i} - 1)$. If $e_i \in \mathcal{E}'$ then the corresponding item in the sum is zero, otherwise it is divisible by c. Therefore, $c|n$.

The observation that each edge in the graph G is processed exactly once shows that the algorithm runs in linear time $O(|\mathcal{E}|)$, which completes the proof. □

5. Numerical computations

We are finally ready to apply all the ideas above to a specific example. First of all, we state a more precise version of the theorem given in the introduction.

Theorem 3. *There exists an open set $X \subset \mathbb{R}^2$ such that $H_{a,b}(X) \subset X$ and $H_{a,b}|_X$ is (combinatorially) mixing at all resolutions $> 10^{-5}$ for all (a, b) in an open set $P \subset \mathbb{R}^2$ containing $(1.4, 0.3)$.*

We choose the Hénon map for the so-called "classical" parameter values simply because it is a very well known and well studied example for which nothing substantial is known. We have purposefully formulated all our definitions and results so far in full generality so that it is straightforward how to apply them to many other examples. What is left is to construct a representation $\mathcal{F}$ of the Hénon map such that $\mathcal{R}^+(\mathcal{F}) \leq \varepsilon$ and then apply the algorithms described above to show that $\mathcal{F}$ is mixing. We describe the construction of $\mathcal{F}$ in detail so that minor modifications should allow essentially the same construction to work in many other cases.

Notice that the result applies to an open set of parameter values. This is an intrinsic feature of the method deriving from the use of interval arithmetic. Since both numbers 1.4 and 0.3 are not representable in the binary floating-point arithmetic, small intervals containing these numbers are taken for the actual computations. Therefore, the statement is automatically proved for an open set of parameters containing these numbers. This costs nothing from the computational point of view and in fact corresponds to the natural fact that numerical methods necessarily only yield results which are stable under sufficiently small perturbations.

5.1. **General strategy.** We describe in detail the strategy for a general continuous map $f\colon \mathbb{R}^n \to \mathbb{R}^n$.

5.1.1. *Open interval arithmetic.* For the purpose of the numerical computations, we use a slight modification of interval analysis introduced in [29].

In the typical approach, closed intervals are used instead of numbers, and the result of each arithmetic operation on such intervals is defined as the smallest possible interval containing the results of the operation on any numbers taken from each of the intervals, e.g., $[a_1, a_2] - [b_1, b_2] = [a_1 - b_2, a_2 - b_1]$. Since we are interested in doing these computations using a fixed-size floating point representation of real numbers, the set of actual numbers that can be represented is finite, and thus we must round the endpoints of the resulting intervals to the nearest representable number in the downwards direction for the left endpoint, and in the upwards direction for the right one. As a consequence, the result of calculations carried out on intervals is an interval that contains every possible exact result of those operations on the numbers belonging to these intervals.

Since we work with open covers, in our case it is necessary to work with open intervals, instead of closed ones. This implies slight differences in the case of some arithmetic operations on intervals. Although addition, subtraction, multiplication and division are the same, those operations in which closed intervals would arise must be slightly changed, like rising to even powers (e.g., computing x^2) or some trigonometric functions (mainly sin and cos). Namely, the result of an operation on open intervals with representable endpoints is defined as the smallest *open* interval with representable endpoints which contains the results of the operation on single elements taken from these intervals, e.g., $(-1, 1)^2 = (-\varepsilon, 1)$, where $-\varepsilon < 0$ is the largest representable negative number (the set of representable numbers is finite, so this number is well defined).

5.1.2. *Open cover parameters.* We start by selecting a bounded rectangular area $B := (a_1, a_1 + w_1) \times \cdots \times (a_n, a_n + w_n) \subset \mathbb{R}^n$ which is assumed to contain the dynamics of our interest. We cover this area with a finite family of overlapping open boxes (n-dimensional hypercubes). We set some number $p_1 \in \mathbb{N}$ of parts into which we intend to subdivide the first interval $(a_1, a_1 + w_1)$ in the definition of

B, and then we compute the numbers $p_2, \dots, p_n$ so that each p_i is approximately proportional to w_i for $i = 1, \dots, n$. Then we choose some small $\kappa > 0$ to be the size of the overlapping margin. As a general rule we can choose κ to be the smallest positive representable number in the chosen computer precision (recall that the set of representable numbers in $(0, 1)$ is finite). We then consider the family

$$\mathcal{B} :\approx \left\{ \Pi_{i=1}^{n} \big(a_i + k_i w_i / p_i, a_i + (k_i + 1) w_i / p_i + \kappa \big) : k_i = 0, \dots, p_i - 1 \right\},$$

where the $:\approx$ symbol is used to indicate the fact that the actual endpoints of the intervals are computed in the floating point arithmetic with rounding the result of each operation to the nearest representable number either in the downwards direction (for the left endpoints) or in the upwards direction (for the right endpoints).

5.1.3. *Construction of X.* The final cover $\mathcal{U}$, and thus the actual space X, is chosen by an iterative procedure from the elements of $\mathcal{B}$, so that $\mathcal{U} \subset \mathcal{B}$. In particular, this cover is necessarily essential.

The iterative procedure is carried out as follows. We start with a point $x_0 \in \mathbb{R}^n$ as close to the attractor as possible. This point may be obtained for example by some non-rigorous numerical simulation of the dynamics, but the construction is in general not particularly sensitive to this choice. We then define an initial approximation of the cover by letting

$$\mathcal{U}_0 := \{ U \in \mathcal{B} : x_0 \in U \}.$$

For each $U \in \mathcal{U}_0$ we compute an open set $\widehat{f(U)}$ as the image of U under f using interval arithmetic. This is a rigorous upper bound for $f(U)$ in the sense that

$$f(U) \subset \widehat{f(U)}.$$

We then define the multivalued map $\mathcal{F}_0 \colon \mathcal{U}_0 \multimap \mathcal{B}$ by

$$\mathcal{F}_0(U) := \{ B \in \mathcal{B} : B \cap \widehat{f(U)} \neq \emptyset \}.$$

At this point we compare $\mathcal{U}_0$ with $\mathcal{F}_0(\mathcal{U}_0) = \bigcup_{U \in \mathcal{U}_0} \mathcal{F}_0(U)$. If $\mathcal{F}_0(\mathcal{U}_0) \subset \mathcal{U}_0$ then we let $\mathcal{U} := \mathcal{U}_0$ and define $\mathcal{F} := \mathcal{F}_0|_{\mathcal{U}} \colon \mathcal{U} \multimap \mathcal{U}$. This is our combinatorial representation of the map $f \colon X \to X$ on the set $X := |\mathcal{U}|$. If $\mathcal{F}_0(\mathcal{U}_0) \not\subset \mathcal{U}_0$ then we define

$$\mathcal{U}_1 := \mathcal{U}_0 \cup \mathcal{F}_0(\mathcal{U}_0)$$

and we compute the map $\mathcal{F}_1 \colon \mathcal{U}_1 \multimap \mathcal{B}$ as an extension of $\mathcal{F}_0$ using interval arithmetic. Note that only the images of the sets in $\mathcal{F}_0(\mathcal{U}_0) \backslash \mathcal{U}_0$ have to be computed. Proceeding again in the same way, we repeat this procedure and we obtain $\mathcal{U}_0, \dots, \mathcal{U}_k$, as well as $\mathcal{F}_0, \dots, \mathcal{F}_k$, until we eventually get $\mathcal{F}(\mathcal{U}_k) \subset \mathcal{U}_k$, at which point we define

$$\mathcal{U} := \mathcal{U}_k \quad \text{and} \quad \mathcal{F} := \mathcal{F}_k|_{\mathcal{U}} \colon \mathcal{U} \multimap \mathcal{U}.$$

Although this process is guaranteed to terminate, because of the finiteness of $\mathcal{B}$, the obtained result might be faulty if $\widehat{f(U)} \not\subset B$ for some $U \in \mathcal{U}$, as then $\mathcal{F}$ is not a combinatorial representation of $f|_X$. If this happens then we say that the construction fails, and one has to choose different parameters for the open cover (Section 5.1.2) and try again. In particular, if the system has an attractor with a sufficiently large basin of attraction and strong enough convergence, then the computations should result in a valid map $\mathcal{F}$, provided the set B is taken large enough, the number p_1 is high enough, and the accuracy of computations is good enough.

Note that if this construction succeeds then the open set $X := |\mathcal{U}|$ is contained in the attraction basin of the attractor, $f(X) \subset |\mathcal{F}(X)| \subset X$, and $\mathcal{F}$ is a combinatorial representation of $f|_X \colon X \to X$.

5.1.4. *Verification of the properties of $\mathcal{F}$.* The outer resolution $\mathcal{R}^+(\mathcal{F})$ of the map $\mathcal{F}$ can be easily calculated using interval arithmetic during the computation of $\mathcal{F}_i$, based on the diameter of a cover of $\widehat{f(U)}$ whenever $\mathcal{F}_i(U)$ is constructed.

After the combinatorial map $\mathcal{F}$ has been constructed, one can verify whether it is transitive and mixing by a straightforward application of Algorithms 4.8 and 4.10.

5.2. **Application to the Hénon map.** We now give the specific data involved in the calculations and discuss in more details the numerical and computational issues involved.

5.2.1. *Constructing a combinatorial representation.* We closely follow the general strategy described in Section 5.1. We choose the rectangular region B in such a way that an approximation of the attractor found in numerical simulations is contained in B with some safety margin (about 0.1 at each side). More precisely, we set

$$\begin{aligned} a_1 &:\approx -1.4, & w_1 &:\approx 2.8, \\ a_2 &:= -0.5, & w_2 &:= 1.0, \end{aligned}$$

where the symbol ":≈" indicates that we take a representable number close to the given decimal number (it is most likely the nearest representable number, depending on a particular compiler, but it does not really matter). Note that the numbers which are powers of 2 are representable and thus the actual values of a_2 and w_2 used in the calculations are exact. We choose κ to be the smallest positive (normal) number representable in the standard double precision floating point arithmetic, which is approximately $2 \cdot 10^{-308}$.

The number p_1 is a parameter of the program that can be changed at each run (it is set up from the command line each time the program is launched), and we figured out by trial and error that for $p_1 := 2{,}132{,}419$ (for which the corresponding p_2 was taken as 761,578) the algorithm constructs a map $\mathcal{F}$ with $\mathcal{R}^+(\mathcal{F}) < 10^{-5}$.

As an initial point supposedly very close to the attractor we take

$$x_0 :\approx (0.61989426930989, 0.17586130934794).$$

This point was found in numerical simulations by iterating the origin a little over 100 million times.

The parameters of the Hénon map for which we want to carry out the computations, $a = 1.4$, $b = 0.3$, are not representable in the binary floating point arithmetic. Therefore, we take open intervals containing them, computed in the program as the quotients 14/10 and 3/10, respectively, with the left endpoints of the intervals rounded down to the nearest representable numbers, and the right endpoints rounded up.

For each $U \in \mathcal{B}$, which is a product of two open intervals, we calculate $\widehat{f(U)}$ by means of open interval arithmetic, as described in Section 5.1.1, using the formula for the Hénon map and the intervals containing the parameters a and b, as explained above.

Instead of constructing subsequent maps $\mathcal{F}_i$, the algorithm is set up in such a way that it constructs a list $\mathcal{U}$ containing the elements of the initial cover $\mathcal{U}_0$ (which covers the initial point x_0), and then for each element of this list it computes $\mathcal{F}(U)$ and immediately appends the elements of $\mathcal{F}(U)$ which are not yet in the list $\mathcal{U}$ to the end of this list. The algorithm terminates if the end of the list has been reached, which means that all the elements of $\mathcal{F}(U)$ for every U in the list are already in the list. The algorithm quits with a failure result if $\widehat{f(U)} \not\subset B$. A sample result of the constructed set $\mathcal{U}$ is illustrated in Figure 5.2.1.

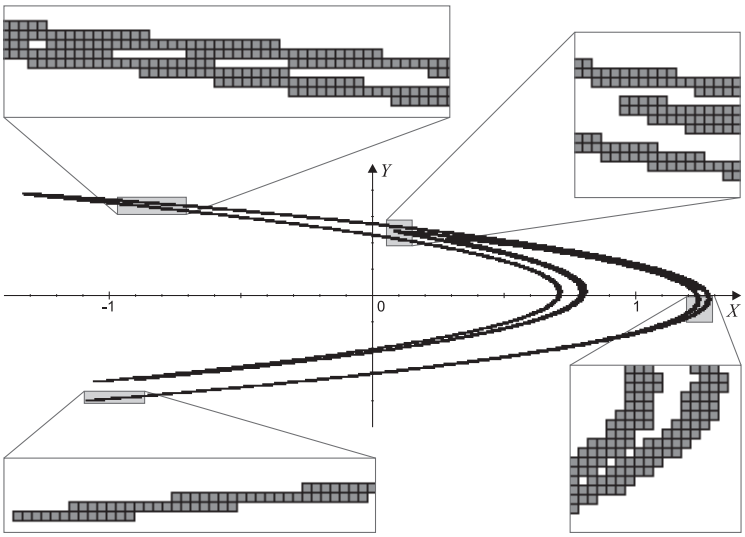

FIGURE 3. A sample low-resolution cover of the Hénon attractor with close-ups of selected details.

5.2.2. *Applying the graph algorithms.* The constructed cover $\mathcal{U}$ and combinatorial map $\mathcal{F}$ are represented in two data structures: an array of pairs of open intervals which represent the elements of $\mathcal{U}$ (that are products of these pairs of intervals), and a directed graph whose vertices are the indices to this array. After the map $\mathcal{F}$ has been constructed, the memory used for the array representing $\mathcal{U}$ is released, and the graph algorithms are run on the graph representation of $\mathcal{F}$.

In the actual implementation of Algorithms 4.7 and 4.10, instead of using the recursive call to the subroutines `DFS-Visit` and `tarjan`, respectively, we use a stack version of DFS because of the limitations of some systems which allow for a limited recursion depth only. In this version, instead of calling the subroutine, the current parameters are put on the stack and another round of the main loop is started with new values, with a return from the recursive call corresponding to taking the previously stored values of the variables from the stack. This is a standard technique, and one can check the details of its implementation in our software available at [38].

5.2.3. *The cost of the computations.* As it should be expected, the time and memory usage of the computations for the Hénon map heavily depend on the number p_1. For small values of p_1, up to some 20,000, the time of computation on a computer with a contemporary modern processor (Intel® Xeon® 5030 2.66 GHz was used in our computations) should not exceed 10 seconds and use a negligible amount of memory (up to about 25 MB). The smallest number p_1 for which the computations were successful was $p_1^{\min} = 446$ (with the corresponding $p_2^{\min} = 159$, and the computed $\mathcal{R}^+(\mathcal{F}^{\min})$ was below 0.05. For smaller values of p_1 (and also for a few larger ones) the algorithm fails because of encountering the situation in which $\widehat{f(U)} \not\subset B$ for some U in the cover being constructed.

For higher values of p_1, we observed that if p_1 is increased 10 times then the outer resolution of the computed map $\mathcal{F}$ decreases about 10 times, while the number of elements in the constructed cover $\mathcal{U}$ grows about 20 times, and so do both the computation time and memory usage (see the actual results available at [38]). In particular, for the highest tested $p_1^{\max} := 2{,}132{,}419$ (with the corresponding $p_2^{\max} = 761{,}578$), a cover $\mathcal{U}^{\max}$ consisting of 161,448,094 boxes was constructed,

and $\mathcal{R}^+(\mathcal{F}^{\max})$ was slightly below 10^{-5}. The computation time was about 1 hour and 17 minutes, out of which the majority was used for the construction of the cover $\mathcal{U}$ and the graph associated to the map $\mathcal{F}$, while running the graph algorithms took less than 6 minutes.

In these computations we reached the limits of the computer equipment available to us at the time of writing this paper, but with the development of technology one can speculate on the possibilities of doing more extensive computations in the future. Based on our rough estimates, obtaining the resolution 10^{-6} would require some 26 hours of processor time and about 360 GB of memory (twice more than $20 \cdot 9 = 180$ GB because of the need to switch from 32-bit to 64-bit integers in the graph representation, due to the high number of vertices and edges). Getting down to 10^{-7} would take over 500 hours (3 weeks) and use some 7 TB of memory. Without switching to distributed computations and probably also changing the approach (e.g., re-computing the graph on-the-fly instead of storing it in the memory), these and higher resolutions seem to be still out of reach for many years to come. However, one might argue that such high precision of the results is not necessary for real applications, and 10^{-5} is more than one might require, so investing in better results does not make sense at this point.

Finally, we would like to point out that the method we have developed is dimension-independent, and so is the related software available at [38]. However, the number of elements in a typical cover of a complicated attractor should be expected to grow considerably faster in higher dimensions with the increase of the target resolution. This might be a major limitation of the applicability of our method for obtaining results at very fine resolutions for more complicated systems. In practice, however, even complicated dynamics in high-dimensional spaces often concentrates around low-dimensional attractors (like the topological horseshoes), in which case the resolutions interesting from the point of view of applications might be still within the reach of contemporary computers, and we sincerely hope to see such applications in the future.

6. Final remarks: Finite resolution versus topological and measurable dynamics

It is tempting to try to see finite resolution properties as approximations of the "real", for example topological, properties of the underlying map $f\colon X \to X$. In terms of the transitivity and mixing properties considered here, this is partly justified in the sense that if f is topologically transitive or mixing then it is also combinatorially transitive or mixing, respectively, at all resolutions (for completeness, we give a formal proof in Appendix B). However our combinatorial notions of transitivity and mixing are not uniquely defined by this requirement, and to fully justify the definitions and the idea that we are approximating the topological properties we would need to show the converse result that if f is combinatorially transitive or mixing at all resolutions then it is topologically transitive or mixing. It is possible to obtain such a double implication for certain kinds of properties, see for instance the comprehensive discussion on precisely this topic in [30]. However, as mentioned in the introduction, this can be achieved only for "robust" properties which are persistent under small perturbations of the system. It seems therefore that in general the finite resolution dynamical properties of a system cannot be thought of as an approximation of the topological properties, at least not in a naive sense.

For a time we struggled with this limitation of the theory and perceived it as a weakness. However, on further reflection we realized that it was in fact arising from an incorrect understanding of what this approach is actually about. The mistake

was to assume that the topological point of view is in fact the "real" dynamics and thus the ultimate goal. In fact, on a fundamental level, the real dynamics, if we really want to have such a notion, has to be just the dynamics of f thought of as a function on the set X. Any more sophisticated description necessarily relies on additional structure and the description then necessarily has to be in a form which is compatible with this structure. There are at least two major frameworks or structures through which, in many cases, one and the same dynamical system can be studied: the *topological* and the *measurable*, which provide in some sense two alternative points of view on the same dynamics. Each of these points of view comes with its own definitions, notions, and results which are intrinsically motivated within that particular framework. A good example, very relevant to the present paper, is provided by the notions of topological mixing and measure-theoretic mixing. Both of these notions formalise some intuitive notion of "mixing" within the corresponding framework, but there is no direct relationship or formal implication between them in general (some systems may be topologically mixing but not measure-theoretically mixing and vice versa). Analogously, the notion of mixing which we give in the paper is a combinatorial notion which is very natural in the finite resolution setting. The relationship of this notion with those of topological and measure-theoretical mixing is interesting and quite complex, and certainly deserving of further thought, though beyond the scope of this paper.

In conclusion, we believe that *the framework of finite resolution dynamics provides one of several possible structures through which to study the dynamics. Accordingly, the definitions and dynamical features of interest should be intrinsically motivated within the finite resolution framework.* Thus finite resolution dynamics should be seen as an alternative structure, alongside the topological and the measurable, which can contribute to an effective study of a dynamical system from a different point of view.

Appendix A. Essential covers

Proposition A.1. *From any cover it is possible to create an essential one (without increasing $\mathcal{R}^+$, but possibly decreasing the number of elements).*

Proof. Let $\mathcal{U}^0 = (U_1^0, \dots, U_n^0)$ be an ordered cover that is not necessarily essential. We shall construct a series of gradually "corrected" ordered covers $\mathcal{U}^1, \dots, \mathcal{U}^n$ such that each of the covers has exactly n elements, $\mathcal{U}^k = (\mathcal{U}_1^k, \dots, \mathcal{U}_n^k)$, and each of the first k elements of $\mathcal{U}^k$ either is empty, or satisfies the exclusive ball condition that appears in Definition 2.5 of an essential cover. Then $\mathcal{U}^n$ will give rise to an essential cover after having removed the empty sets. We proceed by induction. Note that $\mathcal{U}^0$ satisfies the inductive assumption. Suppose that $\mathcal{U}^k$ has been constructed for some $k < n$. Then each U_i^k for $i \le k$ is either empty or satisfies the exclusive ball condition, and then we set $U_i^{k+1} := U_i^k$ for $i \le k$. Consider U_{k+1}^k. If it is empty or satisfies the exclusive ball condition then take $U_i^{k+1} := U_i^k$ also for all $i = k+1, \dots, n$. If $U_{k+1}^k \subset \bigcup_{i \neq k+1} U_i^k$ then set $U_{k+1}^{k+1} := \emptyset$ and $U_i^{k+1} := U_i^k$ for all $i = k+2, \dots, n$. Otherwise, take any $x \in U_{k+1}^k$. Since U_{k+1}^k is open, there exists $r > 0$ such that $B(x, r) \subset U_{k+1}^k$. Take $U_{k+1}^{k+1} := U_{k+1}^k$ and $U_i^{k+1} := U_i^k \setminus \operatorname{cl} B(x, r/2)$ for $i = k+2, \dots, n$. Note that all U_i^{k+1}, $i = 1, \dots, n$, are open sets (although some might be empty), and $\bigcup_{i=1}^n U_i^k = \bigcup_{i=1}^n U_i^{k+1}$, so $\mathcal{U}^{k+1}$ is an ordered cover of the same space. Since $U_i^{k+1} = U_i^k$ for $i \le k$ and $U_i^{k+1} \subset U_i^k$ for $i > k$, the exclusive ball condition for each nonempty U_i^{k+1} with $i \le k$ follows directly from the one for the corresponding U_i^k. Moreover, by the construction, U_{k+1}^{k+1} is either empty, or satisfies the exclusive ball condition with $B(x, r/2)$. □

Appendix B. Topological mixing implies finite resolution mixing

Proposition B.1. *Suppose X is a metric space and $f\colon X \to X$ a continuous map which is topologically transitive or topologically mixing. Then any combinatorial representation $\mathcal{F}$ of f is transitive or mixing, respectively.*

Proof. Recall that f f is *transitive* if for every two open sets $U, V \subset X$ there exists $n > 0$ such that $f^{-n}(U) \cap V \neq \emptyset$. Let $\mathcal{F}\colon \mathcal{U} \multimap \mathcal{U}$ be a combinatorial representation of a transitive map $f\colon X \to X$. We shall show that $\mathcal{F}$ is transitive. Take any $U, V \in \mathcal{U}$. Since f is transitive, there exists $n > 0$ such that $f^{-n}(U) \cap V \neq \emptyset$. Therefore, $f^n(f^{-n}(U)) \cap f^n(V) \neq \emptyset$. Since $f^n(f^{-n}(U)) \subset U$, we have $U \cap f^n(V) \neq \emptyset$. Note that $\mathcal{F}^n$ is a representation of f^n, and thus $U \in \mathcal{F}^n(V)$. Since the choice of U and V was arbitrary, this implies that $\mathcal{F}$ is transitive.

Similarly, f is *mixing* if for every two open sets $U, V \subset X$ there exists $N > 0$ such that $f^{-n}(U) \cap V \neq \emptyset$ for all $n > N$. Let $\mathcal{F}\colon \mathcal{U} \multimap \mathcal{U}$ be a combinatorial representation of a mixing map $f\colon X \to X$. We shall show that $\mathcal{F}$ is mixing. Let $U, V \in \mathcal{U}$. Since f is mixing, there exists $N > 0$ such that $f^{-n}(U) \cap V \neq \emptyset$ for all $n > N$. Fix this n. Note that $f^n(f^{-n}(U)) \cap f^n(V) \neq \emptyset$. Since $f^n(f^{-n}(U)) \subset U$, we have $U \cap f^n(V) \neq \emptyset$. Recall that $\mathcal{F}^n$ is a representation of f^n, and thus $U \in \mathcal{F}^n(V)$. Since the choice of U and V was arbitrary, this proves that $\mathcal{F}$ is mixing. □

References

[1] Z. Arai, On hyperbolic plateaus of the Hénon map, *Experiment. Math.*, 16 (2007), 181–188.

[2] Z. Arai, W. Kalies, H. Kokubu, K. Mischaikow, H. Oka and P. Pilarczyk, A database schema for the analysis of global dynamics of multiparameter systems, *SIAM J. Appl. Dyn. Syst.* 8 (2009), 757–789.

[3] Z. Arai and K. Mischaikow, Rigorous computations of homoclinic tangencies. *SIAM Journal on Applied Dynamical Systems*, 5 (2006) 280–292.

[4] A. Arbieto and C. Matheus, Decidability of Chaos for some families of dynamical systems, *Found. Comput. Math.* (2004), 269–275.

[5] M. Benedicks and L. Carleson, The dynamics of the Hénon map, *Ann. of Math. (2)* 133 (1991), 73–169.

[6] M. Benedicks and L.S. Young, Sinai-Bowen-Ruelle measures for certain Hénon maps, *Invent. Math.*, 112 (1993), 541–576.

[7] R. Bowen, Markov partitions for Axiom A diffeomorphisms, *Amer. J. Math.* 92 (1970), 725–747.

[8] R. Bowen, *Equilibrium states and the ergodic theory of Anosov diffeomorphisms*, Lecture Notes in Mathematics, Vol. 470. Springer-Verlag, Berlin, 1975.

[9] R. Bowen and D. Ruelle, The ergodic theory of Axiom A flows, *Invent. Math.*, 29 (1975), 181–202.

[10] E. Colli, Infinitely many coexisting strange attractors, *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 15(5) (1998), 539–579.

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

[12] J.-P. Eckmann, H. Koch and P. Wittwer, A computer-assisted proof of universality for area-preserving maps, *Mem. Amer. Math. Soc.*, 47(289):vi+122, 1984.

[13] Z. Galias, Rigorous investigation of the Ikeda map by means of interval arithmetic, *Nonlinearity*, 15 (2002), 1759–1779.

[14] S. V. Gonchenko, D. V. Turaev and L. P. Shilprimenikov, On the dynamic properties of diffeomorphisms with homoclinic tangencies, *Sovrem. Mat. Prilozh.* 7 (2003), 91–117.

[15] M. Hénon, A two-dimensional mapping with a strange attractor, *Comm. Math. Phys.*, 50 (1976), 69–77.

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

[17] D. Jabłoński and M. Kulczycki, Topological transitivity, mixing and nonwandering set of subshifts of finite type – a numerical approach, *Intern. J. Computer Math.* 80 (2003), 671–677.

[18] M. V. Jakobson, Absolutely continuous invariant measures for one-parameter families of one-dimensional maps, *Comm. Math. Phys.*, 81(1) (1981), 39–88.

[19] W. D. Kalies, K. Mischaikow and R. C. A. M. VanderVorst, An algorithmic approach to chain recurrence, *Found. Comput. Math.* 5 (2005), 409–449.
[20] T. Kapela and C. Simó, Computer assisted proofs for nonsymmetric planar choreographies and for stability of the eight, *Nonlinearity*, 20 (2007), 1241–1255.
[21] T. Kapela and P. Zgliczyński, The existence of simple choreographies for the N-body problem–a computer-assisted proof, *Nonlinearity*, 16 (2003), 1899–1918.
[22] O. Lanford, A computer-assisted proof of the Feigenbaum conjectures, *Bull. Amer. Math. Soc. (N.S.)* 6 (1982), 427–434.
[23] E. N. Lorenz, Deterministic non-periodic flows, *J. Atmos. Sci.* 20 (1963), 130–141.
[24] S. Luzzatto, I. Melbourne and F. Paccaut, The Lorenz attractor is mixing, *Comm. Math. Phys.* 260 (2005), 393–401.
[25] S. Luzzatto and H. Takahasi, Computable conditions for the occurrence of non-uniform hyperbolicity in families of one-dimensional maps, *Nonlinearity* 19 (2006), 1657–1695.
[26] S. Luzzatto and W. Tucker, Non-uniformly expanding dynamics in maps with singularities and criticalities, *Inst. Hautes Etudes Sci. Publ. Math.* 89 (1999), 179–226.
[27] S. Luzzatto and M. Viana, Positive Lyapunov exponents for Lorenz-like families with criticalities, *Astérisque*, (261):xiii, 201–237, 2000.
[28] K. Mischaikow and M. Mrozek, Chaos in Lorenz equations: A computer assisted proof, *Bull. Amer. Math. Soc. (N.S.)* 33 (1995), 66–72.
[29] R. E. Moore, *Interval analysis*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966.
[30] M. Mrozek, Topological invariants, multivalued maps and computer assisted proofs in dynamics, *Computers Math. Applic* 32 No. 4 (1996), 83-104.
[31] M. Mrozek, An algorithm approach to the Conley index theory, *J. Dynam. Differential Equations* 11 (1999), 711–734.
[32] M. Mrozek and P. Pilarczyk, The Conley index and rigorous numerics for attracting periodic orbits, *Proceedings of the Conference on Variational and Topological Methods in the Study of Nonlinear Phenomena (Pisa, 2000)*, 65–74, Progr. Nonlinear Differential Equations Appl., 49, Birkhäuser Boston, Boston, MA, 2002.
[33] L. Mora and M. Viana, Abundance of strange attractors, *Acta Math.* 171 (1993), 1–71.
[34] S.E. Newhouse, Diffeomorphisms with infinitely many sinks, *Topology*, 13 (1974), 9–18.
[35] M.J. Pacifico, A. Rovella and M. Viana, Infinite-modal maps with global chaotic behavior, *Ann. of Math. (2)*, 148 (1998), 441–484.
[36] J. Palis and F.Takens, Hyperbolicity and sensitive chaotic dynamics at homoclinic bifurcations, Cambridge University Press, Cambridge, 1993.
[37] P. Pilarczyk, Computer assisted method for proving existence of periodic orbits, *TMNA* 13 (1999), 365–377.
[38] P. Pilarczyk, Finite resolution dynamics. Software and examples, `http://www.pawelpilarczyk.com/finresdyn/`.
[39] P. Pilarczyk, Topological-numerical approach to the existence of periodic trajectories in ODEs, *Discrete and Continuous Dynamical Systems 2003*, A Supplement Volume: Dynamical Systems and Differential Equations, 701–708.
[40] P. Pilarczyk and K. Stolot, Excision-preserving cubical approach to the algorithmic computation of the discrete Conley index, *Topology and its Applications* 155 (2008), 1149–1162.
[41] J. G. Sinaĭ, Markov partitions and U-diffeomorphisms, *Funkcional. Anal. i Priložen* 2 (1968), 64–89.
[42] A. Szymczak, A combinatorial procedure for finding isolating neighbourhoods and index pairs, *Proc. Royal Soc. Edinburgh Sect. A* 127 (1997), 1075–1088.
[43] R. Tarjan, Depth-first search and linear graph algorithms, *SIAM J. Comput.* 1 (1972), 146–160.
[44] H. Thunberg, Positive exponent in families with flat critical point, *Ergodic Theory Dynam. Systems*, 19 (1999) 767–807.
[45] M. Tsujii, Positive Lyapunov exponents in families of one-dimensional dynamical systems, *Invent. Math.*, 111 91993) 113–137.
[46] W. Tucker, The Lorenz attractor exists, *C.R. Acad. Sci. Paris*, Série I 328 (1999), 1197–1202.
[47] Q. Wang and L.-S. Young, Strange attractors with one direction of instability, *Comm. Math. Phys.* 218 (2001), 1–97.
[48] Wikipedia contributors, Aperiodic graph, *Wikipedia, The Free Encyclopedia* (2008) `http://en.wikipedia.org/w/index.php?title=Aperiodic_graph&oldid=224030707`.
[49] Wikipedia contributors, Tarjan's strongly connected components algorithm, *Wikipedia, The Free Encyclopedia* (2009) `http://en.wikipedia.org/w/index.php?title=Tarjan%27s_strongly_connected_components_algorithm&oldid=295022950`.

[50] P. Zgliczyński, Rigorous numerics for dissipative partial differential equations. II. Periodic orbit for the Kuramoto-Sivashinsky PDE–a computer-assisted proof, *Found. Comput. Math.*, 4 (2004), 157–185.

Stefano Luzzatto, Mathematics Department, Imperial College, 180 Queen's Gate, London SW7 2AZ, United Kingdom

Current address: Abdus Salam, International Centre for Theoretical Physics, Strada Costiera 11, 34151 Trieste, Italy

E-mail address: luzzatto [at] ictp.it

URL: http://www.ictp.it/~luzzatto/

Paweł Pilarczyk, Centro de Matemática, Universidade do Minho, Campus de Gualtar, 4710-057 Braga, Portugal

E-mail address: pawel.pilarczyk [at] math.uminho.pt

URL: http://www.pawelpilarczyk.com/