

ALGORITHMS FOR RIGOROUS ENTROPY BOUNDS AND SYMBOLIC DYNAMICS

SARAH DAY*, RAFAEL FRONGILLO[†], AND RODRIGO TREVIÑO[‡]

Abstract. The aim of this paper is to introduce a method for computing rigorous lower bounds for topological entropy. The topological entropy of a dynamical system measures the number of trajectories that separate in finite time and quantifies the complexity of the system. Our method relies on extending existing computational Conley index techniques for constructing semi-conjugate symbolic dynamical systems. Besides offering a description of the dynamics, the constructed symbol system allows for the computation of a lower bound for the topological entropy of the original system. Our overall goal is to construct symbolic dynamics that yield a high lower bound for entropy. The method described in this paper is algorithmic and, although it is computational, yields mathematically rigorous results. For illustration, we apply the method to the Hénon map, where we compute a rigorous lower bound for topological entropy of 0.4320.

Key words. topological entropy, symbolic dynamics, Conley index, Hénon map, computer-assisted proof

AMS subject classifications. 37B10, 37B40, 37B30, 37C25, 37M99

1. Introduction. There has been a significant increase in computer-assisted proofs in dynamical systems in the past ten years. Many of these studies use topological techniques and carry at heart ideas introduced by Conley [Con78] as well as extensions derived from them. Conley's ideas, which were generalizations of Morse's theory for gradient-like flows, have spawned two computational approaches for studying complicated dynamics in discrete dynamical systems. The first is the method of *correctly aligned windows* (also known as the method of *covering relations*), which traces its roots to work on *windows* introduced by Easton in [Eas75]. In this paper, we exploit the algorithmic nature of a second approach that relies on the more general tools of Conley index theory. While many of the algorithms for this approach were introduced in earlier works (see e.g. [Szy95], [DJM04], [Day03] and references therein), it was necessary to develop additional techniques and algorithms for this project. In particular, we describe extended techniques for locating a region of the domain to be used for computations in Section 3.1 and present a newly developed automated procedure for taking a computed Conley index and producing an appropriate representative symbolic dynamical system in Section 3.2.

We use the computational techniques based on Conley index theory to build a semi-conjugacy from a map $f : S \rightarrow S$, $S \subset \mathbb{R}^n$, to a symbolic dynamical system and obtain a corresponding lower bound on the topological entropy (one measure of complexity) for the system. Since the symbols we use to construct the symbolic dynamics correspond to disjoint regions of the phase space \mathbb{R}^n , one benefit of this approach is that the symbolic dynamics offers a description of the dynamics on S , including information about the location of points along trajectories in S . Furthermore, the symbolic dynamics act as a lower bound (via the semi-conjugacy) for the dynamics of f on S ; for any trajectory in the symbolic system there is at least one corresponding trajectory of f in S . It follows, as stated in Theorem 2.7, that the topological entropy

*The College of William and Mary, Department of Mathematics, P. O. Box 8795, Williamsburg, VA 23187-8795 (sday@math.wm.edu)

[†]Cornell University (rmf25@cornell.edu)

[‡]Department of Mathematics, University of Maryland, College Park, MD 20742-4015 (rodrigo@math.umd.edu)

of the symbolic system is a lower bound for the topological entropy of f . Since our goal is to compute a high lower bound, our approach relies on trying to maximize the complexity of the constructed symbolic system. We discuss our main approach for maximizing the complexity of the constructed system in Section 3.

Topological entropy is a measurement that many have studied (see e.g. [NBGM08], [AAC90], [ACE+87], [Col02], [Gal02]) using a variety of techniques. We see the automation of our techniques and their independence from the typical constraint that stable and unstable manifolds are one-dimensional and restricted to the plane as the two main strengths in our approach. Indeed, results in [DJM04] lead to entropy bounds for the infinite dimensional Kot-Schaffer map in a similar way to the work described here, and in future work we plan to apply the automated techniques introduced in Section 3 to this map to improve the bounds. In this paper, we apply our approach to the well studied Hénon map in order to obtain results to compare with previous work in this area. We use our automated computational approach based on the ideas outlined above to construct a semi-conjugacy between the dynamics on an appropriate subset of the Hénon attractor and a constructed symbolic dynamical system. Based on this construction, we give a rigorous lower bound of 0.4320 on the topological entropy of the Hénon map in Theorem 4.2. Section 4 also contains a comparison of this sample result with other work in this area.

This paper is organized in the following way: in Section 2, we review the necessary background from dynamical systems and computational Conley index theory. Section 3 contains a detailed description of our extensions of this work to produce automated procedures for constructing complicated semi-conjugate symbolic dynamics. Finally, in Section 4 we apply these procedures to give sample results for the Hénon system.

2. Background. In this section we review some basic definitions and ideas from dynamical systems theory and computational Conley index theory. We will state definitions and theorems which are relevant to our work, and refer the reader to [Rob95], [Con78], [MM02] and references therein for further development and details.

2.1. Symbolic Dynamics and Topological Entropy. Let $f : X \rightarrow X \subset \mathbb{R}^n$ be a continuous map. We will focus on maps that exhibit complicated dynamics on a compact subset $S \subset X$. Because the study of such maps and sets can be very complicated, they are often studied via a representation on a symbol space giving rise to *symbolic dynamics*.

We focus on symbolic dynamics in the form of *subshifts of finite type*. Fix an integer $m \geq 2$ and let T be an $m \times m$ matrix with entries $t_{ij} \in \{0, 1\}$. The corresponding symbol space is

$$\Sigma_T := \{\mathbf{s} = (s_0 s_1 \dots) \mid t_{s_k s_{k+1}} = 1 \text{ for all } k\}$$

Although the matrix T is often referred to as the *adjacency matrix* in graph theory literature, we will refer to T as the *symbol transition matrix* since it captures the allowed or admissible “transitions” between symbols. Finally, we define the *shift map* $\sigma_T : \Sigma_T \rightarrow \Sigma_T$ by

$$\sigma_T(\mathbf{s}) := \mathbf{s}' \quad , \text{ where } s'_i = s_{i+1}.$$

In this framework, (Σ_T, σ_T) is called a *subshift of finite type*, denoting both that we are working with only a finite list of (m) symbols and that only a subset of the set of all sequences on these m symbols are allowed by the symbol transition matrix T .

It is important to note that for an appropriate choice of metric on Σ_T , σ_T is a continuous map and $\sigma_T : \Sigma_T \rightarrow \Sigma_T$ is a dynamical system (see e.g. [Rob95]). Subshifts of finite type are particularly nice in that dynamical objects of interest are often readily identifiable. For example, if one is looking for a period n orbit, then one checks that there is a symbol sequence $\mathbf{s}^* = (s_0, s_1, \dots) \in \Sigma_T$ such that $s_{i+n} = s_i$ for all $i = 0, 1, \dots$. If we view T as an adjacency matrix defining a directed graph, then \mathbf{s}^* corresponds to an n -cycle, or cycle of length n , in the graph. For clarity, we include the following definition of the terms *cycle* and *simple cycle*.

DEFINITION 2.1. *A path of length n in the directed graph G is a sequence of vertices v_0, v_1, \dots, v_n such that each pair (v_i, v_{i+1}) is an edge in G . If in addition, $v_0 = v_n$, then v_0, v_1, \dots, v_n is a cycle of length n . Finally, a cycle v_0, v_1, \dots, v_n is a simple cycle provided that it contains no shorter cycles, namely $v_i = v_j$ with $i \neq j$ if and only if $i, j \in \{0, n\}$.*

While subshifts of finite type and symbolic dynamical systems in general are nice to work with from a mathematical point of view, many interesting dynamical systems do not come in this form. Instead, as mentioned above, we may seek to represent a more general discrete dynamical system by symbolic dynamics. This representation often comes in the form of a *topological conjugacy* or *topological semi-conjugacy*.

DEFINITION 2.2. *A continuous map $\rho : X \rightarrow Y$ is a topological semi-conjugacy between $f : X \rightarrow X$ and $g : Y \rightarrow Y$ if $\rho \circ f = g \circ \rho$ and ρ is surjective (onto). If, in addition, ρ is injective (one-to-one), then ρ is a topological conjugacy.*

Topological conjugacies preserve many properties of dynamical systems. One such example is the following theorem. (For more details, see [Dev89].)

THEOREM 2.3. *Let ρ be a topological conjugacy between $f : X \rightarrow X$ and $g : Y \rightarrow Y$. Then $y \in Y$ is a periodic point of period n under g (i.e. $g^n(y) = y$) if and only if $\rho^{-1}(y)$ is a periodic point of period n under f .*

If f is topologically conjugate to a subshift of finite type, then we have a convenient list of trajectories of f given by the subshift. Indeed, in this case, the topological conjugacy acts as a coordinate transformation of the original system onto a decipherable (symbolic) system. In practice, such a complete description may be beyond our reach and we instead construct topological semi-conjugacies to appropriate subshifts of finite type. As illustrated by Theorem 2.7, these semi-conjugate subshift systems offer lower bounds for the complexity of the dynamics of the original system.

One way to quantify how complicated a given dynamical system is, is to compute its *topological entropy*. The following is based on Bowen's definition of topological entropy in [Bow71].

DEFINITION 2.4. *Let $f : X \rightarrow X$ be a continuous map. A set $W \subset X$ is called (n, ϵ, f) -separated if for any two different points $x, y \in W$ there is an integer j with $0 \leq j < n$ so that the distance between $f^j(x)$ and $f^j(y)$ is greater than ϵ . Let $s(n, \epsilon, f)$ be the maximum cardinality of any (n, ϵ, f) -separated set. The topological entropy of f is the number*

$$h_{top}(f) = \lim_{\epsilon \rightarrow 0} \limsup_{n \rightarrow \infty} \frac{\log(s(n, \epsilon, f))}{n}. \quad (2.1)$$

As a measurement of chaos, we say that a map f for which $h_{top}(f) > 0$ is chaotic, and, if $h_{top}(f) > h_{top}(g)$, then f is 'more chaotic' than g .

Once again, we can turn to symbolic dynamics in order to perform concrete computations.

THEOREM 2.5 (Robinson, [Rob95]). *Let T be a symbol transition matrix and let $\sigma_T : \Sigma_T \rightarrow \Sigma_T$ be the associated subshift of finite type. Then*

$$h_{top}(\sigma_T) = \log(sp(T))$$

where $sp(T)$ is the spectral radius of T .

In essence, (n, ϵ, σ_T) -separation is encoded in the representation of the system and may be computed directly from the symbol transition matrix T .

Computing the topological entropy of a system not given as a subshift proves to be more challenging. In this setting and from a computational perspective, (2.1) may appear daunting. For one thing, sensitive dependence on initial conditions, a property commonly associated to chaotic systems, makes careful, precise measurements of (n, ϵ, f) -separation for large n and small ϵ difficult if not impossible. One technique for dealing with this problem is to focus on computing periodic points up to some cut-off period N rather than length N segments of general trajectories. The problem of finding periodic points may be reduced to finding fixed points for a sufficiently high iterate of the map and two different periodic orbits of period n are necessarily (n, ϵ, f) -separated for sufficiently small ϵ . One then checks that

$$\left\{ \frac{\log(\#\{\text{periodic points of period } n\})}{n} \right\}_{n \leq N}$$

appears to be converging. Galias employed this approach in his study of the Hénon map in [Gal01]. The question now becomes, “is N sufficiently large to yield a good approximation for topological entropy?” This leads us to a second fundamental obstacle to a mathematically rigorous computational approach – the need to obtain asymptotic measurements in both n and ϵ . In Theorem 2.7 we use a special construction of a semi-conjugacy to a subshift system to overcome these difficulties and to compute a rigorous lower bound.

This construction relies on tools from *Conley index theory* discussed in Section 2.2. We use these tools to build the subshift system with the *itinerary function* serving as the semi-conjugacy linking the systems.

DEFINITION 2.6. *Suppose $N \subset X$ may be decomposed into $m < \infty$ disjoint, closed subsets ($N = \cup_{i=1, \dots, m} N_i$). Let S be the maximal invariant set in N (i.e. S is the largest set such that $S \subset N$ and $f(S) = S$). Then $f^j(S) \subset N$ for all $j = 0, 1, \dots$. Finally, let T be the $m \times m$ symbol transition matrix given by*

$$t_{ij} = \begin{cases} 1 & \text{if } f(S \cap N_j) \cap N_i \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

The itinerary function $\rho : S \rightarrow \Sigma_T$ is given by $\rho(x) = s_0 s_1 \dots$, where $s_j = i$ for $f^j(x) \in N_i$. This function is continuous under the appropriate choice of metrics. (See [Dev89], [Rob95] for more details.)

Finally, the following theorem allows us to use this semi-conjugacy to obtain a lower bound for the topological entropy of the system under study.

THEOREM 2.7. *Suppose that the itinerary function ρ is a semi-conjugacy from $f : S \rightarrow S$ to $\sigma_T : \Sigma_T \rightarrow \Sigma_T$ for some $S \subset X$ and subshift of finite type (σ_T, Σ_T) with symbol transition matrix T . Then*

$$h_{top}(f) \geq \log(sp(T)) = h_{top}(\sigma_T).$$

where $sp(T)$ is the spectral radius of T .

Proof. Let $d(N_i, N_j) := \min_{x \in N_i, y \in N_j} d(x, y) > 0$ be the minimal distance between the two disjoint, closed sets N_i and N_j . Since there are only a finite number of these sets, $\epsilon_* := \min_{1 \leq i \neq j \leq m} d(N_i, N_j) > 0$.

For $\mathbf{s} = (s_0, s_1, \dots) \in \Sigma_T$, call the sequence of n symbols, $B_n := (s_0, \dots, s_{n-1})$, an *admissible n -block under T* . For each admissible n -block $B_n = (s_0, \dots, s_{n-1})$, choose $x_{B_n} \in S$ such that $\rho(x_{B_n}) = (s_0, s_1, \dots, s_{n-1}, s_n, \dots) \in \Sigma_T$. Such a point exists in S since ρ maps S onto Σ_T . Furthermore, for $\epsilon < \epsilon_*$, the points chosen in S corresponding to two different admissible n -blocks must be (n, ϵ, f) -separated since within n iterates, their itineraries carry them to two disjoint subsets of $S \cap N$, separated by a distance of at least ϵ_* .

We now have that for $\epsilon < \epsilon_*$, $s(n, \epsilon, f) \geq \#\{\text{admissible } n\text{-blocks under } T\}$. The asymptotic size of the set of admissible n -blocks may be computed as follows (see Theorem 1.9(b) in [Rob95]), to obtain the desired result.

$$\begin{aligned} h_{top}(f) &= \lim_{\epsilon \rightarrow 0} \limsup_{n \rightarrow \infty} \frac{\log(s(n, \epsilon, f))}{n} \\ &\geq \limsup_{n \rightarrow \infty} \frac{\log(\#\{\text{admissible } n\text{-blocks under } T\})}{n} \\ &= \log(sp(T)) \\ &= h_{top}(\sigma_T). \end{aligned} \tag{2.2}$$

□

Thus, we may bound the topological entropy of a map f from below by finding a semi-conjugacy from f to an appropriate subshift of finite type. The higher the spectral radius of the symbol transition matrix T , the better the lower bound we achieve for the topological entropy of the original system.

2.2. Conley Index Theory. We now present some of the topological tools used to build the subshift of finite type required for Theorem 2.7. These tools are based on Conley index theory for which we now give definitions, facts, and theorems which are relevant to our work. A discussion of the implementation of these ideas in a computational framework follows in Section 2.4.

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a continuous map. A *trajectory through $x \in \mathbb{R}^n$* is a sequence

$$\gamma_x := (\dots, x_{-1}, x_0, x_1, \dots) \tag{2.3}$$

such that $x_0 = x$ and $x_{n+1} = f(x_n)$ for all $n \in \mathbb{Z}$. We now define the *invariant set relative to $N \subset \mathbb{R}^n$* as

$$Inv(N, f) := \{x \in N \mid \text{there exists a trajectory } \gamma_x \text{ with } \gamma_x \subset N\} \tag{2.4}$$

One example of a relative invariant set is the domain $S = Inv(N, f)$ on which we defined the itinerary function ρ in Definition 2.6.

We are now ready to present some of the basic structures in Conley index theory.

DEFINITION 2.8. A *compact set $N \subset \mathbb{R}^n$ is an isolating neighborhood if*

$$Inv(N, f) \subset \text{int}(N) \tag{2.5}$$

where $\text{int}(N)$ denotes the interior of N . S is an isolated invariant set if $S = Inv(N, f)$ for some isolating neighborhood N .

We use the next two definitions to encode the dynamics on an isolating neighborhood.

DEFINITION 2.9. Let $P = (P_1, P_0)$ be a pair of compact sets with $P_0 \subset P_1 \subset X$. The map induced on the pointed quotient space $(P_1/P_0, [P_0])$ is

$$f_P(x) := \begin{cases} f(x) & \text{if } x, f(x) \in P_1 \setminus P_0 \\ [P_0] & \text{otherwise} \end{cases} \quad (2.6)$$

DEFINITION 2.10. ([RS88]) The pair of compact sets $P = (P_1, P_0)$ with $P_0 \subset P_1 \subset X$ is an index pair for f provided that

1. the induced map, f_P , is continuous,
2. $\overline{P_1 \setminus P_0}$, the closure of $P_1 \setminus P_0$, is an isolating neighborhood.

In this case, we say that P is an index pair for the isolated invariant set $S = \text{Inv}(\overline{P_1 \setminus P_0}, f)$.

The following definition is required for the definition of the Conley index.

DEFINITION 2.11. Two group homomorphisms, $\phi : G \rightarrow G$ and $\psi : G' \rightarrow G'$ on abelian groups G and G' are shift equivalent if there exist group homomorphisms $r : G \rightarrow G'$ and $s : G' \rightarrow G$ and a constant $m \in \mathbb{N}$ (referred to as the ‘lag’) such that

$$r \circ \phi = \psi \circ r, \quad s \circ \psi = \phi \circ s, \quad r \circ s = \psi^m, \quad \text{and} \quad s \circ r = \phi^m.$$

The shift equivalence class of ϕ , denoted $[\phi]_s$, is the set of all homomorphisms ψ such that ψ is shift equivalent to ϕ .

DEFINITION 2.12. Let $P = (P_1, P_0)$ be an index pair for the isolated invariant set $S = \text{Inv}(\overline{P_1 \setminus P_0}, f)$ and let $f_{P_*} : H_*(P_1, P_0) \rightarrow H_*(P_1, P_0)$ be the map induced on the relative homology groups $H_*(P_1, P_0)$ from the map f_P . The Conley index of S is the shift equivalence class of f_{P_*}

$$\text{Con}(S, f) := [f_{P_*}]_s. \quad (2.7)$$

The Conley index for the isolated invariant set S given in Definition 2.12 is well-defined, namely, every isolated invariant set has an index pair, and the corresponding shift equivalence class remains invariant under different choices for this index pair (see e.g. [MM02]).

So far we have passed from continuous maps to induced maps on relative homology. Our overall goal, however, is to describe the dynamics of our original map. Here we present measurements based on the map on homology that may give us information about the original map. The first theorem is Ważewski’s Principle in the context of Conley index theory.

THEOREM 2.13. If $\text{Con}(S, f) \neq [0]_s$, then $S \neq \emptyset$.

By requiring additional structure in the isolating neighborhood N of S , we can use a modification of Theorem 2.13 to study finer structure in S .

COROLLARY 2.14. Let $N \subset X$ be the union of disjoint, compact sets N_1, \dots, N_m and let $S := \text{Inv}(N, f)$ be the isolated invariant set relative to N . Let

$$S' = \text{Inv}(N_1, f_{N_1} \circ \dots \circ f_{N_1}) \subset S$$

where f_{N_i} denotes the restriction of the map f to the region N_i . If

$$\text{Con}(S', f_{N_1} \circ \dots \circ f_{N_1}) \neq [0]_s, \quad (2.8)$$

then $S' \neq \emptyset$. More specifically, there exists a point in S whose trajectory under f travels through the regions N_1, \dots, N_n in the prescribed order.

We here note that given the hypotheses of Corollary 2.14, there is a nice technique for obtaining the index of S' given the computed index map f_{P^*} , where $P = (P_1, P_0)$ is an index pair with $N = \overline{P_1} \setminus P_0$. Using an approach developed by Szymczak in [Szy95], we set

$$f_P^{ij}(x) := \begin{cases} f(x) & \text{if } x \in N_i \text{ and } f(x) \in N_j \\ [P_0] & \text{otherwise} \end{cases} \quad (2.9)$$

Then $f_{P^*}^{ij} : H_*(P_1, P_0 \cup (\cup_{l \neq i} N_l)) \rightarrow H_*(P_1, P_0 \cup (\cup_{l \neq j} N_l))$. Given f_{P^k} in matrix form representing the linear map on $H_k(P_1, P_0)$, we may label the columns/rows by location of the associated relative homology generators in the subgroups $H_k(P_1, P_0 \cup (\cup_{l \neq 1} N_l)), \dots, H_k(P_1, P_0 \cup (\cup_{l \neq n} N_l))$. To simplify notation, we say that generator g is in region N_i if $g \in H_k(P_1, P_0 \cup (\cup_{l \neq i} N_l))$. Then $f_{P^k}^{ij}$ is the $n_j \times n_i$ submatrix with n_i columns corresponding to the n_i generators in N_i and n_j rows corresponding to the n_j generators in N_j . Furthermore, $(P_1, P_0 \cup (\cup_{l \neq 1} N_l))$ is an index pair for the isolated invariant set $S' = \text{Inv}(N_1, f_{N_n} \circ \dots \circ f_{N_1})$ with index map $f_{P^*}^{n1} \circ \dots \circ f_{P^*}^{12} : H_*(P_1, P_0 \cup (\cup_{l \neq 1} N_l)) \rightarrow H_*(P_1, P_0 \cup (\cup_{l \neq 1} N_l))$. Therefore,

$$\text{Con}(S', f_{N_n} \circ \dots \circ f_{N_1}) = [f_{P^*}^{n1} \circ \dots \circ f_{P^*}^{12}]_s \quad (2.10)$$

Since the more general problem of determining whether the linear map $f_{P^k} : H_k(P_1, P_0) \rightarrow H_k(P_1, P_0)$ is not shift equivalent to 0 may be difficult, we here focus on a computable, sufficient condition. Trace is preserved by shift equivalence, and we adopt the notation

$$\text{tr}_k(\text{Con}(S, f)) := \text{tr}(f_{P^k})$$

where $\text{tr}(f_{P^k})$ denotes the trace of the linear map $f_{P^k} : H_k(P_1, P_0) \rightarrow H_k(P_1, P_0)$. Then if $\text{tr}_k(\text{Con}(S, f)) \neq 0$ for some k , $\text{Con}(S, f) \neq [0]_s$.

COROLLARY 2.15. *If $\text{tr}_k(\text{Con}(S', f_{N_{i_n}} \circ \dots \circ f_{N_{i_1}})) \neq 0$ for some k then there exists $x \in S'$ with $\rho(x) = i_1 i_2 \dots i_n i_1 i_2 \dots i_n \dots$*

Taking this approach we are close to showing something stronger, namely that there is a periodic orbit under f with the corresponding cyclic symbol sequence. This stronger statement relies on computing the Lefschetz number.

DEFINITION 2.16. *Let S be an isolated invariant set. The Lefschetz number of S is defined as*

$$L(S, f) := \sum_k (-1)^k \text{tr}(f_{P^k}) \quad (2.11)$$

where $P = (P_1, P_0)$ is an index pair for S .

The Lefschetz number is essential to the following theorem and its corollary.

THEOREM 2.17. *Let S be an isolated invariant set. If*

$$L(S, f) \neq 0, \quad (2.12)$$

then S contains a fixed point.

For a proof, see [Szy96]. As before, a refinement of the approach allows us to study symbolic dynamics.

COROLLARY 2.18. *Let $N \subset X$ be the finite union of disjoint, compact sets N_1, \dots, N_m , and let $S := \text{Inv}(N, f)$. Let $S' = \text{Inv}(N_1, f_{N_1} \circ \dots \circ f_{N_m}) \subset S$ where f_{N_i} denotes the map f restricted to the region N_i . If*

$$L(S', f_{N_m} \circ \dots \circ f_{N_1}) \neq 0, \quad (2.13)$$

then $f_{N_m} \circ \dots \circ f_{N_1}$ contains a fixed point in S' that corresponds to a periodic point of period n in S that under f travels through the regions N_1, \dots, N_m in order.

In what follows, we will develop algorithms based on Corollary 2.15 to construct and verify symbolic dynamics. However, in the special case where the index map f_{P^*} is nontrivial on exactly one level (as occurs with the Hénon map), we may use Corollary 2.18 to show that the constructed semi-conjugate symbolic systems has the added stronger property that every periodic orbit in the symbolic system corresponds to a periodic orbit in the original system of the same period.

2.3. Multivalued and Combinatorial Maps. Now that we have the relevant tools from Conley index theory, we can begin applying them algorithmically to extract information about the dynamical system $f : X \rightarrow X$. In this section, we describe the construction of a combinatorial representation of f . The first step is to define a *multivalued map* F that will be used to incorporate bounded errors in the representation.

DEFINITION 2.19. *The multivalued map $F : X \rightrightarrows X$ is a map from X to its power set, i.e. for all $x \in X$, $F(x) \subset X$. If for a continuous, single-valued map $f : X \rightarrow X$, $f(x) \in F(x)$ and $F(x)$ is acyclic (i.e. has the homology of a point) for all $x \in X$, then f is a continuous selector of F and F is an enclosure of f .*

In what follows, we discuss how to construct an enclosure of the map under study. The purpose of the enclosure is to incorporate round-off and other errors that occur in computations. This construction requires rigorous, small error bounds in order to create an enclosure whose images are not so large as to obscure all relevant dynamics. Given an appropriate enclosure, the topological tools from Section 2.2 may be used to uncover dynamics of the underlying map. Furthermore, there are algorithms for both the construction of the enclosure and the computation of the Conley index. These algorithms require a further step – discretizing the domain in order to store it in the computer as a finite list of objects.

We begin by using the subdivision procedure implemented in the software package *GAIO* [DFJ01] to create a grid \mathcal{G} on a compact (rectangular) region in X . In practice, the region chosen for representation is usually determined either experimentally through non-rigorous numerical simulations or analytically given a special structure or symmetry for the system (e.g. a compact attracting region). We partition a specified rectangular set $W = \prod_{k=1}^n [x_k^-, x_k^+] \subset \mathbb{R}^n$ into a *cubical grid*

$$\mathcal{G}^{(d)} := \left\{ \prod_{k=1}^n \left[x_k^- + \frac{i_k r_k}{2^d}, x_k^- + \frac{(i_k + 1)r_k}{2^d} \right] \mid i_k \in \{0, \dots, 2^d - 1\} \right\}$$

where $r_k = x_k^+ - x_k^-$ is the radius of W in the k th coordinate and the depth d is a non-negative integer. We call an element of the grid, $B = \prod_{k=1}^n \left[x_k^- + \frac{i_k r_k}{2^d}, x_k^- + \frac{(i_k + 1)r_k}{2^d} \right]$, a *box*. For a collection of boxes, $G \subset \mathcal{G} = \mathcal{G}^{(d)}$, define the *topological realization* of G as $|G| := \cup_{B \in G} B \subset \mathbb{R}^n$.

Constructing a useful *combinatorial enclosure* involves bounding all round-off and other errors. In our study of the Hénon map in Section 4, we construct a combinatorial

enclosure by computing images of $f(|G|)$ using interval arithmetic software. This produces a bounding box, $\tilde{f}(|G|)$, for the image $f(|G|)$, which is then intersected with the grid \mathcal{G} to produce the combinatorial enclosure image

$$\mathcal{F}(G) := \{G' \in \mathcal{G} : |G'| \cap \tilde{f}(|G|) \neq \emptyset\}.$$

This combinatorial enclosure, $\mathcal{F} : \mathcal{G} \rightrightarrows \mathcal{G}$, yields an enclosure $F = |\mathcal{F}|$ of f in the following way: define $|\mathcal{F}| : W \rightrightarrows W$, where $W = \cup_{G \in \mathcal{G}} |G|$,

$$|\mathcal{F}|(x) := \bigcup_{G \in \mathcal{G} \text{ with } x \in |G|} |\mathcal{F}(G)|. \quad (2.14)$$

More importantly, efficient algorithms exist for computing isolating neighborhoods, index pairs, and Conley indices for f from an appropriate combinatorial enclosure \mathcal{F} of f .

2.4. Computational Conley Index Theory. Now we give algorithms for computing the isolating neighborhoods, index pairs, and Conley indices first introduced in Section 2.2 in the setting of combinatorial enclosures.

DEFINITION 2.20. A combinatorial trajectory of a combinatorial enclosure \mathcal{F} through $G \in \mathcal{G}$ is a bi-infinite sequence $\gamma_G = (\dots, G_{-1}, G_0, G_1, \dots)$ with $G_0 = G$, $G_n \in \mathcal{G}$, and $G_{n+1} \in \mathcal{F}(G_n)$ for all $n \in \mathbb{Z}$.

DEFINITION 2.21. The combinatorial invariant set in $\mathcal{N} \subset \mathcal{G}$ for a combinatorial enclosure \mathcal{F} is

$$\text{Inv}(\mathcal{N}, \mathcal{F}) := \{G \in \mathcal{G} : \text{there exists a trajectory } \gamma_G \subset \mathcal{N}\}.$$

DEFINITION 2.22. The combinatorial neighborhood of $\mathcal{B} \subset \mathcal{G}$ is

$$o(\mathcal{B}) := \{G \in \mathcal{G} : |G| \cap |\mathcal{B}| \neq \emptyset\}.$$

This set, $|o(\mathcal{B})|$, sometimes referred to as a *one box neighborhood* of \mathcal{B} in \mathcal{G} , is the smallest representable neighborhood of $|\mathcal{B}|$ in the grid \mathcal{G} .

While there are different characterizations of isolation in the setting of combinatorial enclosures, we chose the following for this work.

DEFINITION 2.23. If

$$o(\text{Inv}(\mathcal{N}, \mathcal{F})) \subset \mathcal{N}$$

then $\mathcal{N} \subset \mathcal{G}$ is a combinatorial isolating neighborhood under \mathcal{F} .

Note that by construction, the topological realization $|\mathcal{N}|$ of a combinatorial isolating neighborhood \mathcal{N} under \mathcal{F} is an isolating neighborhood under any continuous selector $f \in |\mathcal{F}|$. This definition is stronger than what is actually required to guarantee isolation on the topological level. It is, however, the definition that we will use in this work and is computable using the following approach.

Let $\mathcal{S} \subset \mathcal{G}$. Set $\mathcal{N} = \mathcal{S}$ and let $o(\mathcal{N})$ be the combinatorial neighborhood of \mathcal{N} in \mathcal{G} . If $\text{Inv}(o(\mathcal{N}), \mathcal{F}) = \mathcal{N}$, then \mathcal{N} is isolated under \mathcal{F} . If not, set $\mathcal{N} := \text{Inv}(o(\mathcal{N}), \mathcal{F})$ and repeat the above procedure. In this way, we grow the set \mathcal{N} until either the isolation condition is met, or the set grows to intersect the boundary of \mathcal{G} in which case the algorithm fails to locate an isolating neighborhood in \mathcal{G} . This procedure is outlined in more detail in the following algorithm from [DJM04].

ALGORITHM 1 (**Grow Isolating Neighborhood**).

INPUT: grid \mathcal{G} , combinatorial enclosure \mathcal{F} on \mathcal{G} , set $S \subset \mathcal{G}$
OUTPUT: a combinatorial isolating neighborhood \mathcal{N} containing S
or $\mathcal{N} = \emptyset$ if the isolation condition is not met

```

 $\mathcal{N} = \text{grow\_isolating\_neighborhood}(\mathcal{G}, \mathcal{F}, S)$ 
 $\mathcal{G\_boundary} := \{G \in \mathcal{G} : |G| \cap \partial|G| \neq \emptyset\};$ 
 $\mathcal{N} := S;$ 
while  $\text{Inv}(o(\mathcal{N}), \mathcal{F}) \not\subset \mathcal{N}$  and  $\mathcal{N} \cap \mathcal{G\_boundary} = \emptyset,$ 
     $\mathcal{N} := \text{Inv}(o(\mathcal{N}), \mathcal{F});$ 
end
if  $\mathcal{N} \cap \mathcal{G\_boundary} = \emptyset,$  return  $\mathcal{N};$ 
else return  $\emptyset;$ 
end

```

Once we have an isolating neighborhood for f , our next goal is to compute a corresponding index pair. The following definition of a *combinatorial index pair* again emphasizes our goal of using the combinatorial enclosure to compute structures for f .

DEFINITION 2.24. A pair $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_0)$ of cubical sets is a combinatorial index pair for a combinatorial enclosure \mathcal{F} if the corresponding topological realization $P = (P_1, P_0)$, where $P_i := |\mathcal{P}_i|$, is an index pair for any continuous selector $f \in |\mathcal{F}|$. Namely, $P_1 \setminus P_0 = |\mathcal{P}_1 \setminus \mathcal{P}_0|$ is an isolating neighborhood under f and the map f_P , as defined in Definition 2.9, is continuous.

The following algorithm produces a combinatorial index pair associated to a combinatorial isolating neighborhood produced via Algorithm 1. While there are other algorithms for producing combinatorial index pairs, this algorithm works well with later index computations. For more details, see the description of *modified combinatorial index pairs* in [Day03].

ALGORITHM 2 (Build Index Pair).

INPUT: grid \mathcal{G} , combinatorial enclosure \mathcal{F} on \mathcal{G} ,
combinatorial isolating neighborhood \mathcal{N} produced by Algorithm 1
OUTPUT: combinatorial index pair $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_0)$ with $\mathcal{P}_1 \setminus \mathcal{P}_0 = \mathcal{N}$

```

 $[\mathcal{P}_1, \mathcal{P}_0] = \text{build\_index\_pair}(\mathcal{G}, \mathcal{F}, \mathcal{N});$ 
 $\mathcal{P}_0 := \emptyset;$ 
 $\text{New} := \mathcal{F}(\mathcal{N}) \cap o(\mathcal{N}) \setminus \mathcal{N};$ 
while  $\text{New} \neq \emptyset$ 
     $\mathcal{P}_0 := \mathcal{P}_0 \cup \text{New};$ 
     $\text{New} := (\mathcal{F}(\mathcal{P}_0) \cap o(\mathcal{N})) \setminus \mathcal{P}_0;$ 
end
 $\mathcal{P}_1 := \mathcal{N} \cup \mathcal{P}_0;$ 
return  $[\mathcal{P}_1, \mathcal{P}_0];$ 

```

We now have an isolating neighborhood $|\mathcal{N}|$ and corresponding index pair $P := (|P_1|, |P_0|)$ for f . What remains in computing the Conley index for the associated isolated invariant set, $S := \text{Inv}(|\mathcal{N}|, f)$, is to compute the map $f_{P^*} : H_*(|P_1|, |P_0|) \rightarrow H_*(|P_1|, |P_0|)$. Once again, the combinatorial enclosure offers the appropriate computational framework and we use the software program *homcubes* in [Pil98] to compute f_{P^*} . This step is outlined in Algorithm 3.

ALGORITHM 3 (Compute Index Map).

INPUT: grid \mathcal{G} , combinatorial enclosure \mathcal{F} on \mathcal{G} ,
 combinatorial index pair $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_0)$ produced by Algorithm 2
 OUTPUT: relative homology groups $H_*(|\mathcal{P}_1|, |\mathcal{P}_0|)$,
 the induced index map $f_{P^*} : H_*(|\mathcal{P}_1|, |\mathcal{P}_0|) \rightarrow H_*(|\mathcal{P}_1|, |\mathcal{P}_0|)$,
 and the induced submaps $\{f_{P_k}^{ij}\}$ on connected components

```
[f_{P^*} H_*(|\mathcal{P}_1|, |\mathcal{P}_0|) {f_{P_k}^{ij}}] = compute_index_map(\mathcal{G}, \mathcal{F}, \mathcal{P}_1, \mathcal{P}_0)
Q_1 = \mathcal{F}(\mathcal{P}_1);
Q_0 = \mathcal{F}(\mathcal{P}_0);
[f_{P^*} H_*(|\mathcal{P}_1|, |\mathcal{P}_0|) {f_{P_k}^{ij}}] := homcubes(\mathcal{P}_1, \mathcal{P}_0, Q_1, Q_0, \mathcal{F});
return [f_{P^*} H_*(|\mathcal{P}_1|, |\mathcal{P}_0|) {f_{P_k}^{ij}}];
```

Algorithm 3 produces a sequence of matrices for the maps $f_{P_0}, f_{P_1}, \dots, f_{P_n}$ where n is the dimension of the phase space X . For $k > n$, $f_{P_k} = 0$. The associated Conley index is $Con_*(S) = [f_{P^*}]_s$, for $S := Inv(|\mathcal{P}_1 \setminus \mathcal{P}_0|, f)$. The submaps $f_{P_k}^{ij} : H_k(|\mathcal{P}_1|, |\mathcal{P}_0| \cup_{l \neq i} |\mathcal{N}_l|) \rightarrow H_k(|\mathcal{P}_1|, |\mathcal{P}_0| \cup_{l \neq j} |\mathcal{N}_l|)$, where $|\mathcal{N}_1|, \dots, |\mathcal{N}_n|$ are the connected components of $|\mathcal{P}_1 \setminus \mathcal{P}_0|$, are given as submatrices of f_{P_k} . These are the maps required for Corollaries 2.14, 2.15, and 2.18. In the following section, we describe an algorithmic procedure for using this index information to construct the appropriate subshift of finite type.

3. Constructing and Verifying Complicated Symbolic Dynamics. Given $f : X \rightarrow X$, the general method we adopt for computing a lower bound on topological entropy consists of the following steps:

- constructing a fixed cubical grid \mathcal{G} on a subset of X and a combinatorial enclosure \mathcal{F} of f on \mathcal{G} (Section 2.3),
- locating a region of interest \mathcal{S} in \mathcal{G} (Section 3.1),
- computing the associated Conley index (Section 2.4),
- constructing semi-conjugate symbolic dynamics (Section 3.2),
- using the constructed symbolic dynamical system to compute a lower bound on the topological entropy of f (Theorem 2.7).

While many steps of this general procedure have been carried out in previous work (e.g. [DJM04] for the first four steps, and [Gal01] for the last step), we here seek to uncover far more complicated symbolic dynamics. This requires a more automated approach based on setting verifiable conditions for uncovering and proving the existence of cyclic symbolic dynamics and ignoring or giving up on the verification of dynamics that does not satisfy these conditions. Along these lines, we now give algorithms for locating a region of interest (Section 3.1) and processing the resulting index information (Section 3.2) that allow us to uncover more complicated dynamics than previously found using related techniques. This improved procedure produces the entropy bounds presented in Section 4.

3.1. Locating a region of interest. We now turn to the second task in this list – that of locating the region of the grid where we will attempt to compute interesting symbolic dynamics. More specifically, the set that we are calling the *region of interest* will serve as the input, \mathcal{S} , for Algorithm 1. We show three different methods for locating regions of interest for the Hénon map in Sections 4.1, 4.2, and 4.3. In this section, we focus on the method that, of these three, is both general (i.e. is not restricted to studies of the Hénon map) and yields high entropy bounds. This is the method followed in Section 4.2. The first step in this approach is similar in spirit to the work of Cvitanović and others in using periodic orbits of low periods to approximate chaotic attractors. We begin by finding short cycles in the combinatorial enclosure (directed graph) \mathcal{F} . These short cycles correspond to possible periodic orbits of low period for f . We then add a level of complexity by searching for paths in the directed graph between these short cycles. From a dynamics point of view, these paths represent possible mixing between the periodic regions.

We construct a list of short cycles in \mathcal{G} by setting a computational parameter `Max.Cycle.Length` $\in \mathbb{Z}^+$, and locating the cycles in \mathcal{F} of length k with $1 \leq k \leq \text{Max.Cycle.Length}$. These cycles are nonzero entries on the diagonal of \mathcal{F} (when viewed as a transition matrix) raised to the k th power. The corresponding computed vertices in \mathcal{F} are the regions in \mathcal{G} that may contain period k points of f . Starting with $\mathcal{S} = \emptyset$, we begin adding the short cycles to \mathcal{S} one by one, starting with the shortest. Just before adding a cycle to \mathcal{S} , we grow its isolating neighborhood using Algorithm 1 and then check that this neighborhood does not intersect the isolating neighborhood of the current collection. This corresponds to a possible increase in the number of symbols and/or the number of admissible transitions between symbols in the resulting constructed symbolic system and may eventually lead to a higher entropy bound. If this condition is not met, we do not add the cycle and move to the next cycle in the list, continuing until the list is exhausted. We next use breadth first search (BFS) to find shortest path, pairwise connections between the short cycles in \mathcal{S} and add these connecting paths to \mathcal{S} . This procedure is outlined in Algorithm 4.

ALGORITHM 4 (Locating Region of Interest/Joining Low Cycles).

INPUT: grid \mathcal{G} , combinatorial enclosure \mathcal{F} on \mathcal{G} ,
 computational parameter `Max.Cycle.Length`
 OUTPUT: region of interest $\mathcal{S} \subset \mathcal{G}$

```

 $\mathcal{S} = \text{find\_and\_connect\_low\_cycles}(\mathcal{G}, \mathcal{F}, \text{Max.Cycle.Length})$ 
 $\mathcal{S} = \emptyset;$ 
 $\mathcal{N} = \emptyset;$ 
for  $n = 1 : \text{Max.Cycle.Length}$ ,
  for each length  $n$  cycle  $c$  in  $\mathcal{F}$ ,
     $\mathcal{N}_c = \text{grow\_isolating\_neighborhood}(\mathcal{G}, \mathcal{F}, c);$ 
    if  $\mathcal{N}_c \cap \mathcal{N} = \emptyset$ ,
       $\mathcal{S} = \mathcal{S} \cup c;$ 
       $\mathcal{N} = \text{grow\_isolating\_neighborhood}(\mathcal{G}, \mathcal{F}, \mathcal{N} \cup c);$ 
    end
  end
end
 $\mathcal{S}_c := \mathcal{S};$ 
for each vertex  $v_i \in \mathcal{S}_c$ ,
```

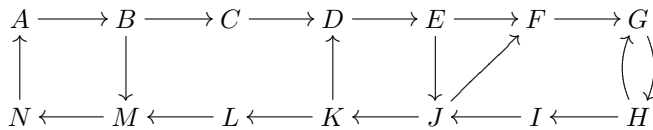


FIG. 3.1. Symbol transition graph constructed from a 2-cycle, two 4-cycles, and pairwise connections.

```

for each vertex  $v_j \in \mathcal{S}_c$ ,
   $\gamma = \text{shortest\_path in } \mathcal{F} \text{ from } v_i \text{ to } v_j \text{ in } \mathcal{G} \setminus \text{o}(\text{o}(\mathcal{S}))$ ;
   $\mathcal{S} := \mathcal{S} \cup \gamma$ ;
end
end
return  $\mathcal{S}$ ;

```

Here, we explicitly compute cycles with lengths up to `MaxCycleLength`, which in practice is small. However, we obtain many new cycles by adding pairwise connections between the computed cycles. This allows us to uncover complicated dynamics without having to explicitly search for the long cycles that correspond to periodic orbits of high period. As illustration, Figure 3.1 depicts a subshift of finite type constructed from a region of interest consisting of a length 2 cycle, two length 4 cycles and pairwise shortest connecting paths between these three objects. Note that the resulting subshift system contains infinitely many cycles (of lengths 5, 8, 10 and higher) and positive topological entropy.

While effective in computation of entropy bounds for the Hénon map (see Section 4.2), this approach for the construction of the region of interest, \mathcal{S} , could be improved. Given a fixed combinatorial enclosure \mathcal{F} on a grid \mathcal{G} , one goal would be to optimize the construction of \mathcal{S} in order to produce a subshift of finite type with the highest possible entropy. As a first step along these lines, the relationship between the entropy bound and the maximal cycle length used in Algorithm 4 in a study of the Hénon map is depicted in Figure 4.4. In addition, there is a clear trade-off between refining the grid in order to find, isolate, and connect more low period cycles to produce a higher bound and the associated increase in computational cost. (The effect of refining the grid on increasing the bound is illustrated for the Hénon map in Figure 4.5.) Another improvement to these techniques related to this balance would involve making the computation of \mathcal{G} , and therefore \mathcal{S} , adaptive. The goal here would be to refine the grid in areas where new low period periodic orbits and connections may be uncovered without having to recompute structures in the remainder of the space.

3.2. Processing Index Information. Recall that our goal is to compute complicated symbolic dynamics. If we are successful in locating an appropriate region of interest in the domain (one approach is described in Section 3.1), the corresponding Conley index computed by the algorithms described in Section 2.4 is given as a large matrix representing the map induced on an index pair consisting of many disjoint components.

From this index map, we wish to find a symbol transition matrix T such that f is semi-conjugate to the subshift on Σ_T . We first use some properties of shift equivalence to simplify the computed index map. We then construct T from a collection of cycles, called *verified cycles*, that satisfy the hypotheses of Corollary 2.15.

3.2.1. Removing Transient Generators. We begin our processing of the index map $f_{P^*} : H_*(P_1, P_0) \rightarrow H_*(P_1, P_0)$ by removing generators from $H_*(P_1, P_0)$ that do not correspond to asymptotic invariant behavior. More specifically, we utilize the fact that the Conley index, $\text{Con}_*(S, f)$, is the shift equivalence class of f_{P^*} to construct a new representative of the class obtained by removing generators $\alpha \in H_k(P_1, P_0)$ such that $f_{P^k}^l(\alpha) = 0$ or $\alpha \notin f_{P^k}^l(H_k(|\mathcal{P}_1|, |\mathcal{P}_0|))$ for some $l \in \mathbb{Z}$.

Note that since we are considering continuous maps f on \mathbb{R}^n , $f_{P^k} : H_k(P_1, P_0) \rightarrow H_k(P_1, P_0)$ are linear maps on (finite) vector spaces. We therefore choose to think of f_{P^k} as a square matrix. Suppose that f_{P^k} is similar to a matrix A , i.e. $f_{P^k} = B^{-1}AB$ for some invertible matrix B . Then, by setting $r = B$, $s = B^{-1}$, and $m = 0$ in Definition 2.11, we see that $[f_{P^k}]_s = [A]_s$. In what follows, B will be an appropriate reordering of the basis so that A takes on the block lower triangular form required for the following theorem.

THEOREM 3.1. *Let*

$$A = \begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

be a 3×3 block lower-triangular matrix, with square matrices A_{ii} on the diagonal. If $A_{11}^l = 0$ and $A_{33}^l = 0$ for some l , then A is shift equivalent to A_{22} .

Proof. For $i = 1, 2, 3$, let $n_i \times n_i$ be the size of the square matrix A_{ii} , and define projection and inclusion maps respectively as follows:

$$\pi = \begin{bmatrix} 0_{n_{22} \times n_{11}} & I_{n_{22}} & 0_{n_{22} \times n_{33}} \end{bmatrix}$$

and

$$\iota = \pi^\top.$$

One can check that the maps $R := \pi A^l$ and $S := A^l \iota$ satisfy the conditions stated in Definition 2.11 to give the desired shift equivalence between A and A_{22} with lag constant $m = 2l$. \square

The motivation for the previous theorem was to find a simpler representative for the shift equivalence class of f_{P^k} . This relies on finding a reordering of the basis for f_{P^k} that yields a similar matrix A satisfying the hypotheses of Theorem 3.1. In order to use existing efficient algorithms, we now turn to a graph interpretation of the $l \times l$ matrix f_{P^k} . More specifically, we consider the directed graph $G = (V, E)$ with vertices $1, \dots, l$ and edges $(j, i) \in E$ if and only if $f_{P^k}(i, j) \neq 0$. Let

$$V_3 := \{v \in V \mid \text{any path starting at } v \text{ has length less than } l\} \quad (3.1)$$

$$V_1 := \{v \in V \setminus V_3 \mid \text{any path ending at } v \text{ has length less than } l\}. \quad (3.2)$$

and

$$V_2 := V \setminus (V_1 \cup V_3). \quad (3.3)$$

Note that since there are l vertices, V_1 is the set of vertices that are not connected to cycles in backward time and V_3 is the set of all vertices that are not connected to cycles in forward time. The following two lemmas show that the partition $\{V_1, V_2, V_3\}$ of the vertex set V is useful for finding zeros in the matrix f_{P^k} .

LEMMA 3.2. *The submatrix $f_{P_k}(V_1, V_2 \cup V_3)$ of f_{P_k} corresponding to the rows indexed by V_1 and columns indexed by $V_2 \cup V_3$ is the zero matrix of the appropriate size.*

Proof. Suppose that $f_{P_k}(w, v) \neq 0$ for $w \in V_1$ and $v \in V_2 \cup V_3$. Then (v, w) is an edge in the associated directed graph G . Since v is not in V_1 , there exists a path v_1, \dots, v_l, v in G . Then v_1, \dots, v_l, v, w is a length $l + 1$ path in G , contradicting our assumption that $w \in V_1$. \square

LEMMA 3.3. *The submatrix $f_{P_k}(V_2, V_3)$ of f_{P_k} corresponding to the rows indexed by V_2 and columns indexed by V_3 is the zero matrix of the appropriate size.*

Proof. Suppose that $f_{P_k}(w, v) \neq 0$ for $w \in V_2$ and $v \in V_3$. Then (v, w) is an edge in the associated directed graph G . Since w is not in V_3 , there exists a path w, v_1, \dots, v_l in G . Then v, w, v_1, \dots, v_l must also be a path in G , contradicting our assumption that $v \in V_3$. \square

We have now shown that if we reorder the basis by listing the basis elements in V_1 , followed by those in V_2 , followed by those in V_3 , we obtain the following block form (with rows and columns labeled by location in the specified sets):

$$f_{P_k} \sim A = \begin{matrix} & V_1 & V_2 & V_3 \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \end{matrix} & \begin{pmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \end{matrix}$$

What remains to show in order to use Theorem 3.1 is the following lemma.

LEMMA 3.4. *The two matrices, A_{11}^l and A_{33}^l , are zero matrices of the appropriate sizes.*

Proof. We obtained the block lower triangular matrix A by a reordering of the basis for the matrix f_{P_k} . Therefore, the associated directed graph G_A for A is the directed graph G with relabeled vertices. With a slight abuse of notation, we consider again the subsets V_1, V_2, V_3 in G_A to be the sets satisfying (3.2), (3.3), and (3.1) respectively. Interpreting nonzero entries of A to be weights on the corresponding edges, we may use powers of A to study paths in G_A . More specifically, $A^l(i, j) \neq 0$ implies that there exists a length l path from vertex j to vertex i in G_A (see, e.g. [Die05]). Now suppose that $A^l(i, j) = A_{11}^l(i, j) \neq 0$ for some $i, j \in V_1$. Then by the above argument, there exists a length l path in G_A that ends at a vertex in V_1 . This contradicts (3.2). Therefore, $A_{11}^l = 0$. A similar argument shows that $A_{33}^l(i, j) = 0$ for all $i, j \in V_3$. \square

We now have that f_{P_k} is similar (and hence shift equivalent) to A which is shift equivalent to $\tilde{f}_{P_k} := A_{22}$ by Lemma 3.4 and Theorem 3.1. Therefore, we may take \tilde{f}_{P_k} to be the new, possibly smaller representative of the Conley index

$$\text{Con}(S, f) = [f_{P_k}]_s = [\tilde{f}_{P_k}]_s.$$

This procedure is outlined in Algorithm 5. Here, algorithms based on depth or breadth first search may be used to efficiently compute the required sets V_1, V_2 , and V_3 . As we will show in Section 4 this technique may give a drastic decrease in the size of the representative index map.

ALGORITHM 5 (**Remove Transient Generators**).

INPUT: square matrix f_{P_k}
 OUTPUT: shift equivalent (square) matrix \tilde{f}_{P_k}

```

 $\tilde{f}_{P_k} = \text{remove\_transient\_generators}(f_{P_k})$ 
 $G = (V, E)$  is the directed graph associated to  $f_{P_k}$ ;
 $V_3 = \{v \in V \mid \text{any path starting at } v \text{ is finite}\}$ ;
 $V_1 = \{v \in V \mid \text{any path ending at } v \text{ is finite}\}$ ;
 $V_2 = V \setminus (V_1 \cup V_3)$ ;
 $\tilde{f}_{P_k} = f_{P_k}(V_2, V_2)$ ;
return  $\tilde{f}_{P_k}$ ;

```

3.2.2. Cycle verification. We now automate a procedure for using Conley index computations to construct a semi-conjugate subshift of finite type. As described in Theorem 3.6 below, we construct the subshift system from a collection of cycles that are *verified* using Corollary 2.15. As will be seen in Section 4, the automation of this procedure becomes necessary as we build increasingly complicated subshifts of finite type. In particular, building a subshift system containing infinitely many periodic orbits may, in principle, lead to an infinite list of computations to verify that the hypotheses of Corollary 2.15 hold for each cycle. In the following approach, we present an algorithm which uses a finite list of computations to verify a possibly infinite set of cycles.

Given an index pair $P = (P_1, P_0)$, we begin by labeling each of the (m) disjoint regions of the isolating neighborhood $N := \overline{P_1} \setminus P_0$. Let $N = \cup_{i=1}^m N_i$ with N_i closed and nonempty and $N_i \cap N_j = \emptyset$ for all $i \neq j$. By construction, each N_i has a corresponding cubical representation $\mathcal{N}_i \subset \mathcal{N}$. Recall that the associated itinerary function ρ is defined by $\rho(x) = (s_0 s_1 \dots)$ with $s_j = i$ if $f^j(x) \in N_i$. Let \tilde{T} be the matrix of admissible transitions between the regions N_i allowed by \mathcal{F} . More specifically, \tilde{T} is the $m \times m$ matrix with entries

$$t_{ij} = \begin{cases} 1 & \text{if } \mathcal{F}(\mathcal{N}_j) \cap \mathcal{N}_i \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \quad (3.4)$$

Then $\rho : \tilde{S} \rightarrow \Sigma_{\tilde{T}}$ where $\tilde{S} := \text{Inv}(N, f)$ and $\Sigma_{\tilde{T}}$ and $\sigma_{\tilde{T}} : \Sigma_{\tilde{T}} \rightarrow \Sigma_{\tilde{T}}$ are the subshift of finite type defined in Section 2. As previously discussed, $\rho : \tilde{S} \rightarrow \Sigma_{\tilde{T}}$ may not be surjective and, hence, $\sigma_{\tilde{T}} : \Sigma_{\tilde{T}} \rightarrow \Sigma_{\tilde{T}}$ may not be semi-conjugate to $f : \tilde{S} \rightarrow \tilde{S}$. We will now construct a subshift system, $\sigma_T : \Sigma_T \rightarrow \Sigma_T$, with $\Sigma_T \subset \Sigma_{\tilde{T}}$, that we prove is semi-conjugate via ρ to $f : S \rightarrow S$ with $S \subset \tilde{S}$.

Let $G = (V, E)$ be the directed graph associated to the symbol transition graph \tilde{T} (viewed as an adjacency matrix). More specifically, the vertices are named for the regions, N_i with $V = \{1, 2, \dots, m\}$ and the edge set $E := \{(j, i) \in V \times V \mid t_{ij} = 1\}$ represents the admissible transitions between regions. In our approach, we begin by removing all paths in G that are not contained in cycles. These paths correspond to dynamics that we will not check using index theory. A practical way to perform this step is to remove edges and vertices not contained in the strongly connected components (SCC) of G . We will now study Conley indices for periodic symbol sequences in $\Sigma_{\tilde{T}}$ represented by cycles in G .

As discussed in Section 2.2, we consider restricted index maps

$$f_{P_k}^{i,j} : H_k(P_1, P_0 \cup (\cup_{l \neq i} N_l)) \rightarrow H_k(P_1, P_0 \cup (\cup_{l \neq j} N_l)).$$

To do this, we first group the generators of $H_k(P_1, P_0)$ remaining after running Algorithm 5 by region N_i . Again, thinking of f_{P_k} as a matrix with rows and columns

corresponding to the generators of $H_k(P_1, P_0)$, we have that

$$f_{P_k}^{ij} := f_{P_k}(g_{N_j}, g_{N_i}) \quad (3.5)$$

where g_{N_i} are the (row/column) indices of generators in $H_k(P_1, P_0 \cup \bigcup_{l \neq i} N_l) \subset H_k(P_1, P_0)$. Here, $f_{P_k}^{ij}$ is as an $n_j \times n_i$ matrix where n_i and n_j are the number of generators in regions N_i and N_j respectively. To simplify notation, for a path $p = (s_1, s_2, \dots, s_n)$, let

$$f_{P_k}^p := f_{P_k}^{s_{n-1}s_n} \circ \dots \circ f_{P_k}^{s_1 s_2}. \quad (3.6)$$

DEFINITION 3.5. *We say that a cycle, $c = (s_1, s_2, \dots, s_n, s_1)$ in G is verified if for some k ,*

$$\text{tr}(f_{P_k}^c) = \text{tr}_k \text{Con}(S', f_{N_{s_n}} \circ \dots \circ f_{N_{s_1}}) \neq 0$$

where $S' := \text{Inv}(N_{s_1}, f|_{N_{s_n} \rightarrow N_{s_1}} \circ \dots \circ f|_{N_{s_1} \rightarrow N_{s_2}})$. Note that by Corollary 2.15, $\rho^{-1}(\mathbf{s}) \neq \emptyset$, where $\mathbf{s} = (s_1 s_2 \dots s_n s_1 \dots s_n \dots)$ is the periodic symbol sequence corresponding to the verified cycle.

Before discussing our automated approach for verifying cycles, we give the following theorem to serve as motivation for this work.

THEOREM 3.6. *Let Σ_T be the space of symbol sequences with symbol transition matrix T and let $\text{Per}(\Sigma_T)$ be the set of periodic symbol sequences in Σ_T under σ_T . Suppose that $\Sigma_T = \overline{\text{Per}(\Sigma_T)}$ and for each $\mathbf{s} = (s_1 \dots s_n s_1 \dots s_n \dots) \in \text{Per}(\Sigma_T)$, the corresponding cycle $c = (s_1, s_2, \dots, s_n, s_1)$ in G has been verified according to Definition 3.5. Then the itinerary function ρ is a semi-conjugacy between $f : S \rightarrow S$ and $\sigma_T : \Sigma_T \rightarrow \Sigma_T$, where $S := \rho^{-1}(\Sigma_T) \subset \tilde{S}$.*

Proof. The itinerary function, $\rho : \tilde{S} \rightarrow \Sigma_T$ is continuous and $\rho \circ f = \sigma_T \circ \rho$ (see Section 2 and references therein). Furthermore, since each cycle in G corresponding to a periodic symbol sequence in Σ_T has been verified according to Definition 3.5, ρ maps onto $\text{Per}(\Sigma_T)$. Since ρ is continuous, $\tilde{S} := \text{Inv}(N, f)$ is compact, and Σ_T is Hausdorff, ρ must map onto the closure of the set of periodic symbol sequences, $\overline{\text{Per}(\Sigma_T)} = \Sigma_T$. Therefore, $\rho : S \rightarrow \Sigma_T$ is a semi-conjugacy. \square

The list of cycles that may be verified according to Definition 3.5 relies implicitly on the form of f_P and, more specifically, on $f_{P_k}^{ij}$ for $k = 0, 1, 2, \dots$ and $i, j \in \{1, \dots, m\}$. For the examples studied in Section 4, the homology maps f_{P_k} are trivial for all $k \neq 1$. Therefore, for these examples we fix $k = 1$, as any other choice will necessarily lead to a zero trace and failure to verify all cycles. For different systems, there may be more flexibility in the choice of k . Given a fixed k , the question of how the list of verified cycles relies on choices of i and j is far more subtle. We begin this discussion by fixing k and considering the case where each region contains exactly one homology generator ($n_i = 1$ for all $i = 1, \dots, m$). We will then discuss the more difficult case where some regions have multiple homology generators.

Note that if there is only one generator per region, then $f_{P_k}^{ij}$ is a scalar for all admissible transitions $\tilde{t}_{ji} = 1$ in \tilde{T} . In this case, if $f_{P_k}^{ij} \neq 0$ for all admissible transitions, then for any admissible periodic symbol sequence $\mathbf{s} = (s_1 s_2 \dots s_n s_1 s_2 \dots s_n \dots)$ with corresponding cycle $c = (s_1 s_2 \dots s_n s_1)$, $\text{tr}(f_{P_k}^c) \neq 0$, and, therefore, all cycles in G are verified. If, on the other hand, $f_{P_k}^{ij} = 0$ for some admissible transition, then any cycle c with edge (i, j) will have $\text{tr}(f_{P_k}^c) = 0$ and cannot be verified using this approach. In this case, we remove this transition from the set of admissible transitions by removing the edge (i, j) from G and, correspondingly, by setting $t_{ji} = 0$ in

T . In essence, cycle verification computations in the setting where there is exactly one homology generator per component boil down to a (finite) check that entries in f_{P_k} corresponding to admissible transitions in T are nonzero.

If there are regions that contain more than one generator of homology, then these computations become more complicated. In what follows, we will systematically process the cycles in G . In the first phase of the procedure, we process paths and cycles in \mathcal{G} in an attempt verify cycles. Alternatively, one can think about identifying all cycles that may not be verified by our approach. Along these lines, we will label certain cycles as *unverifiable* and certain paths as *unconcatenable*. From these, we will identify a collection of edges that need to be removed from the graph so that all remaining cycles are verified cycles. Note that in what follows, labeling a path *unconcatenable* does not mean that cycles containing this path may not be verified according to Definition 3.5, only that we may not verify some such cycles using our prescribed list of finite computations. Let `Max_Iter` be a nonnegative integer that will serve as a computational parameter.

DEFINITION 3.7. Define the edge set for a path $p = (v_0, \dots, v_n)$ to be

$$E(p) := \{(v_i, v_{i+1}) \in E(G) \mid i = 0, \dots, n-1\}$$

and the length of p to be $|p| = n$. Consider a cycle $c = (s, v_2, v_3, \dots, v_{n-1}, s)$ starting and ending at vertex s . If $\text{tr}(f_{P_k}^c) = 0$, then c is *unverifiable*. (See also Definition 3.5.)

A path $p = (s, v_2, v_3, \dots, v_{n-1}, t)$ from s to t of length $|p| \leq \text{Max_Iter}$, is *unconcatenable* if $f_{P_k}^p = \mathbf{0}$.

For a path $p = (s, v_2, v_3, \dots, v_{n-1}, t)$ from s to t of length $|p| = \text{Max_Iter}$, p is *concatenable* if there exists a path p' from s to t with $|p'| < \text{Max_Iter}$, $E(p') \subseteq E(p)$, and $f_{P_k}^p = \alpha f_{P_k}^{p'} \neq \mathbf{0}$ for some $\alpha \neq 0$. If no such path p' exists, then p is *unconcatenable*.

Finally, an edge set E is *prohibited* (at computational parameter `Max_Iter`) if at least one of the following holds:

1. there exists an *unverifiable* cycle c with $|c| \leq \text{Max_Iter}$ and $E(c) \subseteq E$
2. there exists an *unconcatenable* path p with $E(p) \subseteq E$

LEMMA 3.8. If c is a cycle whose edge set $E(c)$ is not prohibited then c is a *verified* cycle.

Proof. Suppose that $|c| \leq \text{Max_Iter}$. Since $E(c)$ is not prohibited, c must be a *verified* cycle.

Next, notice that in the natural partial ordering on edge sets, if E' is prohibited then so is E for any E containing E' . Therefore, $E(c)$ must not contain any prohibited subsets. If $|c| > \text{Max_Iter}$, then c is the concatenation of 2 paths, p_1 and p_2 , where $|p_2| = \text{Max_Iter}$. We will use the notation $p_1 p_2$ to denote the concatenation of paths p_1 and p_2 . Label the start/end vertices s_1, t_1 and s_2, t_2 of p_1 and p_2 respectively. Note that $s_1 = t_2$ and $t_1 = s_2$ by construction. Since $E(p_2) \subseteq E(c)$ is not prohibited, there exists a path p'_2 from s_2 to t_2 with $E(p'_2) \subseteq E(p_2)$, $|p'_2| < \text{Max_Iter}$, and $f_{P_k}^{p_2} = \alpha f_{P_k}^{p'_2}$ for some $\alpha \neq 0$. Therefore,

$$\begin{aligned} f_{P_k}^c &= f_{P_k}^{p_2} f_{P_k}^{p_1} \\ &= \alpha f_{P_k}^{p'_2} f_{P_k}^{p_1} \\ &= \alpha f_{P_k}^{c'} \end{aligned}$$

where $c' = p_1 p_2'$ is a cycle with $E(c') = E(p_1) \cup E(p_2') \subseteq E(c)$ and length $|c'| \leq |c| - 1$. Continuing this process, we obtain a cycle \tilde{c} with $|\tilde{c}| \leq \text{Max_Iter}$, $E(\tilde{c}) \subseteq E(c)$, and $f_{P_k}^c = \tilde{\alpha} f_{P_k}^{\tilde{c}}$ for some $\tilde{\alpha} \neq 0$. Since $E(\tilde{c})$ cannot be prohibited, \tilde{c} must be verifiable and

$$\text{tr}(f_{P_k}^c) = \text{tr}(\tilde{\alpha} f_{P_k}^{\tilde{c}}) = \tilde{\alpha} \text{tr}(f_{P_k}^{\tilde{c}}) \neq 0.$$

Therefore, c is a verified cycle. \square

Lemma 3.8 and Theorem 3.6 provide an outline of our approach for constructing the semi-conjugate system. By Lemma 3.8, we know that all cycles that do not have prohibited edge sets are verified cycles and may be used to construct the semi-conjugate system according to Theorem 3.6. In practice, we use the prohibited edge sets to identify a collection of edges to be removed from G , resulting in the desired semi-conjugate system.

We now give an outline of our procedure for locating prohibited edge sets by collecting and testing appropriate matrix products along paths in G . The algorithm outputs a collection of minimal prohibited edge sets \mathcal{B} , that is for any prohibited edge set E , there is a prohibited edge set $E' \in \mathcal{B}$ with $E' \subseteq E$.

ALGORITHM 6 (Find Prohibited Edge Sets).

INPUT: graph G , index map f_{P_k} , computational parameter Max_Iter ;

OUTPUT: list of minimal prohibited edge sets \mathcal{B}

$\mathcal{B} = \text{find_prohibited_edge_sets}(G, \{f_{P_k}\}, \text{Max_Iter});$

$\mathcal{B} = \emptyset;$

for all $s, t \in V(G), E \subseteq E(G)$, set all $\mathcal{M}(s, t, E, k) = \emptyset;$

for $(s, t) \in E(G)$,

if $s == t$ and $\text{tr}(f_{P_k}^{st}) == 0$, $\mathcal{B} = \mathcal{B} \cup \{(s, t)\};$

else $\mathcal{M}(s, t, \{(s, t)\}, 1) = \{f_{P_k}^{st}\};$

end

end

for $k = 1.. \text{Max_Iter}$,

for $s, t \in V(G), E \subseteq E(G), M \in \mathcal{M}(s, t, E, k)$,

for $(t, u) \in E(G)$,

$E' = E \cup (t, u);$

$M' = f_{P_k}^{tu} M;$

if $(M' == \mathbf{0})$ or $(s == u$ and $\text{tr}(M') == 0)$,

$\mathcal{B} = \mathcal{B} \cup \{E'\};$

$\mathcal{M}(s', t', E'', \ell) = \emptyset$ for all $s', t' \in V(G), E' \subseteq E'' \subseteq E(G), \ell \leq k;$

else if $\exists M'' \in \bigcup_{E'' \subseteq E'} \mathcal{M}(s, t, E'', \ell)$ and $\alpha \neq 0$ with $M'' = \alpha M'$,

$\mathcal{M}(s, t, E', k + 1) = \mathcal{M}(s, t, E', k + 1) \cup \{M'\};$

end

end

end

end

$\mathcal{B} = \{E \subseteq E(G) \mid \mathcal{M}(s, t, E, \text{Max_Iter}) \neq \emptyset \text{ and } E \text{ minimal}\}$

return $\mathcal{B};$

In practice, it is more efficient to apply Algorithm 6 only to a subgraph of G that captures the behavior of the system in the regions with multiple homology generators. More specifically, we first study G restricted to the vertices for multiple generator regions and the neighboring single generator regions. This allows us to take advantage of the fact that $f_{P_k}^p$ is a scalar for all paths p starting and ending at vertices for single generator regions. By removing enough edges so that there are no remaining prohibited edge sets in the subgraph, we can reduce the check that cycles remaining in G are verified to a check that the maps $f_{P_k}^{ij}$ between single generator regions are nonzero. This is the approach we adopt for the results described in Section 4.

For all cycles c in G that do not contain any prohibited edge sets (listed in \mathcal{B}), c is a verified cycle by Lemma 3.8. What remains for the construction of a subgraph G' of verified cycles, is to remove enough edges so that we no longer have any cycles with prohibited edge sets. Since our goal is to obtain a high lower bound for entropy, we will select one edge from each prohibited edge set so that the removal of these edges results in the semi-conjugate symbolic system with highest entropy. Again, since the list of prohibited edge sets is finite (and each prohibited edge set is finite), the computation of optimal edges to remove is finite. Removing the edges yields a graph in which all cycles may be verified using Corollary 2.15. By Theorem 3.6, the corresponding adjacency matrix, T , defines a semi-conjugate symbolic system.

The following is an outline of the procedure for breaking prohibited edge sets.

ALGORITHM 7 (**Break Prohibited Edge Sets**).

INPUT: graph G , a list of prohibited edge sets \mathcal{B} ,

OUTPUT: graph G' in which all cycles may be verified via Corollary 2.15

```

 $G' = \text{break\_prohibited\_edge\_sets}(G, \mathcal{B})$ 
  if  $\mathcal{B} = \emptyset$ , return  $G' = G$ ;
   $h_{\max} = -1$ ;
   $E_c = \emptyset$ ;
  for each set  $\{e_1, e_2, \dots, e_N\}$ , where  $e_i$  is an edge on the  $i$ th cycle in  $\mathcal{B}$ ,
    let  $G'$  be the subgraph of  $G$  obtained by removing edges  $e_1, e_2, \dots, e_N$ ;
    let  $T(G')$  be the adjacency matrix for  $G'$ ;
     $h = \log(\text{sp}(T(G')))$ ;
    if  $h > h_{\max}$ ,
       $h_{\max} = h$ ;
       $E_c = \{e_1, e_2, \dots, e_N\}$ ;
    end
  end
  let  $G'$  be the subgraph of  $G$  obtained by removing the edges in  $E_c$ ;
  return  $G'$ ;

```

Combining Algorithms 6 and 7, Theorem 3.6 guarantees that the following algorithm produces a symbol transition matrix T with $\sigma : \Sigma_T \rightarrow \Sigma_T$ semi-conjugate to $f : S \rightarrow S$. Noting that $f_{P_k}^{ij} = \mathbf{0}$ will cause the verification procedure to fail for any cycle containing edge (i, j) , we will start with a graph G on the same vertex set with the edge set $E = \{(i, j) | f_{P_k}^{ij} \neq \mathbf{0}\}$.

ALGORITHM 8 (**Build Subshift**).

INPUT: index map $f_{P_k} : H_k(|\mathcal{P}_1|, |\mathcal{P}_0|) \rightarrow H_k(|\mathcal{P}_1|, |\mathcal{P}_0|)$,
computational parameter `Max_Iter`
OUTPUT: symbol transition matrix T for semi-conjugate subshift of finite type

```

T = build_subshift(f_{P_k}, H_k(|\mathcal{P}_1|, |\mathcal{P}_0|), Max_Iter)
f_{P_k} = remove_transient_generators(f_{P_k});
set m to be the number of disjoint components of H_k(|\mathcal{P}_1|, |\mathcal{P}_0|);
V = {1, ..., m};
E = {(i, j) \in V \times V | f_{P_k}^{ij} \neq \mathbf{0}};
G = G(V, E);
G = SCC(G); (removes all edges not contained in cycles)
B = find_prohibited_edge_sets(G, f_{P_k}, Max_Iter);
G' = break_prohibited_edge_sets(G, B);
T is the adjacency matrix for graph G';
return T;

```

4. An example: the Hénon map. As illustration, we now apply our techniques to the Hénon map

$$h(x, y) = (1 + y - ax^2, bx) \quad (4.1)$$

at the classical parameters $a=1.4$, $b=0.3$. Since its first appearance in [Hén76], there has been extensive research on the Hénon map. The first result concerning a real description of the chaotic dynamics of the Hénon map is [MS80], where the existence of a transverse homoclinic point, and hence the existence of horseshoe dynamics, is proved. In [Szy97], Szymczak used Conley index theory to give a computer-assisted proof of the existence of periodic orbits of all periods except three and five. In [Gal02], Galias employed the method of covering relations (related to Easton’s windows) to give a computer-assisted proof of the existence of an infinite number of homoclinic and heteroclinic trajectories. [Gal02] also contains a result which gives a rigorous lower bound for the topological entropy of the map $h_{top}(h) \geq 0.4300$. In [NBGM08], Newhouse et al. use the planar structure of the Hénon map to compute $h_{top}(h) \geq 0.46469$, the highest lower bound currently reported for the entropy of Hénon at the classical parameter values.

For this work, we use the *GAIO* software package to construct grids, $\mathcal{G}^{(d)}$, at discretization depths $0 \leq d \leq 12$, on the initial box $[-1.425, 1.425] \times [-0.425, 0.425]$ (see Section 2.3). We then use the interval arithmetic package INTLAB [Cse99] to compute a combinatorial enclosure, \mathcal{H} , on $\mathcal{G}^{(d)}$ as

$$\mathcal{H}(I_1 \times I_2) = \{G \in \mathcal{G}^{(d)} | \tilde{h}(I_1, I_2) \cap G \neq \emptyset\}$$

where $I_1 \times I_2$ is an element in $\mathcal{G}^{(d)}$ in interval product notation and $\tilde{h}(I_1, I_2)$ denotes the rectangular image of $h(I_1, I_2)$ computed using (outward rounding) interval arithmetic. Finally, we use Matlab scripts encoding the algorithms outlined throughout the paper to find and compute the required Conley index structures and subshifts of finite type. In the following sample results, we describe three different techniques for producing the region of interest $\mathcal{S} \subset \mathcal{G}^d$. Given \mathcal{S} , the main approach is the following:

ALGORITHM 9 (Main).

INPUT: grid \mathcal{G}^d , combinatorial enclosure \mathcal{H} on \mathcal{G} ,

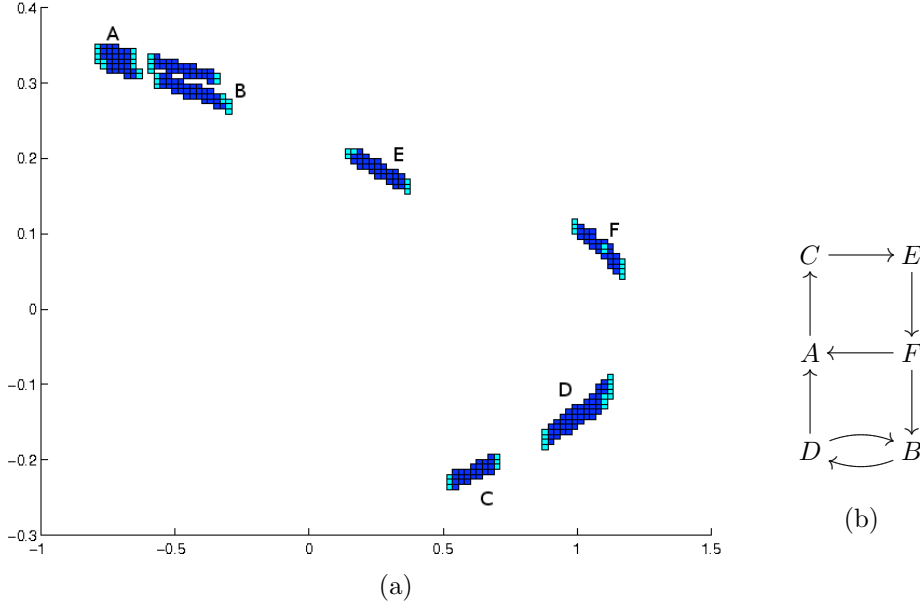


FIG. 4.1. (a) A combinatorial index pair, $(\mathcal{P}_1, \mathcal{P}_0)$, computed using Algorithms 1 and 2 for the Hénon map at depth $d = 7$. (\mathcal{P}_0 is the collection of boxes shown in cyan.) (b) The corresponding symbol transition graph produced by Algorithm 8.

```

region of interest  $\mathcal{S}$ , computational parameter Max.Iter
OUTPUT: lower bound on the topological entropy of  $h$  ENTROPY

ENTROPY = compute_entropy_lower_bound( $\mathcal{G}^d$ ,  $\mathcal{H}$ ,  $\mathcal{S}$ , Max.Iter)
ENTROPY = 0;
 $\mathcal{N}$  = grow_isolating_neighborhood( $\mathcal{S}$ ); (Algorithm 1)
 $[\mathcal{P}_1, \mathcal{P}_0]$  = build_index_pair( $\mathcal{N}$ ); (Algorithm 2)
 $f_{P^*}$  = compute_index_map( $\mathcal{P}_1$ ,  $\mathcal{P}_0$ ,  $\mathcal{H}$ ,  $\mathcal{G}^d$ ); (Algorithm 3)
for  $k = 1..\dim(\mathcal{G}^d)$ , with  $f_{P_k} \neq 0$ ,
     $T$  = build_subshift( $f_{P_k}$ ,  $H_k(|\mathcal{P}_1|, |\mathcal{P}_0|)$ , Max.Iter); (Algorithm 8)
    ENTROPY := max{ENTROPY,  $\log(sp(T))$ };
end
return ENTROPY;

```

4.1. Joining two short cycles. For purposes of illustration, we begin with a relatively simple example on the grid at depth $d = 7$. Although the resulting entropy lower bound, 0.2406, is small, this example provides us with matrices of reasonable sizes for depicting the results of various stages of the procedure. For this example, we locate a region of interest, \mathcal{S} , by searching the computed enclosure \mathcal{H} on $\mathcal{G}^{(7)}$ for a cycle of length 2, a cycle of length 4 and shortest path connections from the 2-cycle to the 4-cycle and from the 4-cycle to the 2-cycle. \mathcal{S} is the union of these four objects. Applying Algorithms 1 and 2 to \mathcal{S} result in the index pair given in Figure 4.1.

THEOREM 4.1. *The topological entropy of the Hénon map (4.1) is bounded from below by 0.2406.*

Proof. The computed index map for the index pair depicted in Figure 4.1(a) is

$$h_{P,1} = \begin{matrix} & A & B & B & B & B & C & D & E & F & F \\ \begin{matrix} A \\ B \\ B \\ B \\ B \\ C \\ D \\ E \\ F \\ F \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

The rows and columns are labeled by location of the corresponding homology generator in the labeled regions of the isolating neighborhood (see Figure 4.1(a)). Applying Algorithm 5 for removing transient generators to $h_{P,1}$ produces the shift equivalent matrix

$$A = \begin{matrix} & A & B & B & C & D & E & F & F \\ \begin{matrix} A \\ B \\ B \\ C \\ D \\ E \\ F \\ F \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

This is the matrix labeled A_{22} in Theorem 3.1 and is obtained by an appropriate reordering of the basis. Note that this algorithm removed two of the homology generators in region B and, therefore, reduced the size of the representative of the shift equivalence class/Conley index.

As an example computation, using Corollary 2.15 to verify the cycle (B, D, B) , we check that

$$\begin{aligned} \text{tr}_1(\text{Con}(S', f|_{D \rightarrow B} \circ f|_{B \rightarrow D})) &= \text{tr}(h_{P,1}^{DB} h_{P,1}^{BD}) \\ &= \text{tr}_1 \left(\begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \\ &= \text{tr}_1 \left(\begin{bmatrix} -1 & -1 \\ 0 & 0 \end{bmatrix} \right) \\ &\neq 0. \end{aligned}$$

Running Algorithm 8 on A to verify a collection of cycles results in the construction of a semi-conjugate subshift system with symbol transition matrix

$$T = \begin{matrix} & A & B & C & D & E & F \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}.$$

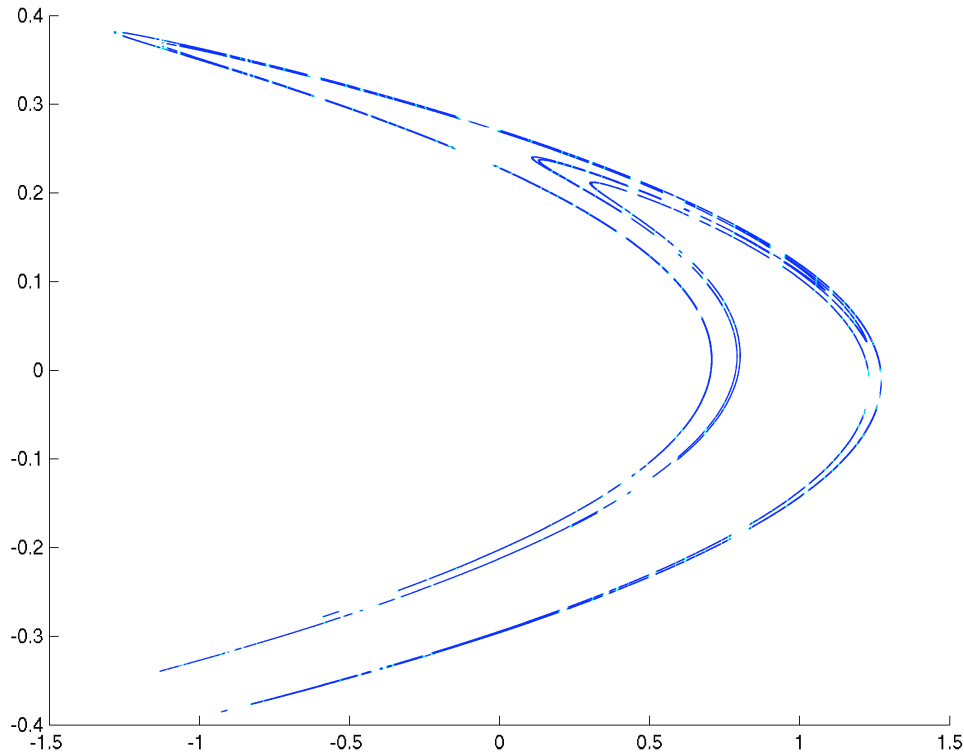


FIG. 4.2. The combinatorial index pair, $(\mathcal{P}_1, \mathcal{P}_0)$, constructed starting with Algorithm 4 for Theorem 4.2 at depth 12. (\mathcal{P}_0 is the collection of boxes shown in cyan.)

The corresponding symbol transition graph is given in Figure 4.1(b). Since the log of the spectral radius of T is greater than 0.2406, the result follows from Theorem 2.7. \square

4.2. Joining low cycles (Algorithm 4). We now focus on improving the bound by refining the grid and using Algorithm 4 to compute a more complicated region of interest.

This approach results in the following theorem.

THEOREM 4.2. *The topological entropy of the Hénon map (4.1) is bounded from below by 0.4320.*

Outline of Proof. Given the enclosure \mathcal{H} on $\mathcal{G}^{(12)}$, we use Algorithm 4 with `Max_Cycle_Length=7` to produce the region of interest \mathcal{S} . We then follow Algorithm 9. The index pair for \mathcal{S} appears in Figure 4.2. Algorithm 3 returns an index map on 1519 relative homology generators. Algorithm 5 reduces this map to a shift equivalent map on 302 generators. Finally, Algorithm 8 produces a semi-conjugate subshift of finite type with 199 symbols. The symbol transition matrix for the constructed subshift is depicted in Figure 4.3. The log of the spectral radius of T is bounded from below by 0.4320. The result then follows from Theorem 2.7. \square

For the above result computed on the grid $G^{(12)}$, we choose the maximal cycle length for Algorithm 4 to be `Max_Cycle_Length= 7`. This choice is made because choosing `Max_Cycle_Length < 7` yields a lower bound than that given in Theorem 4.2, and choosing `Max_Cycle_Length > 7` yields an entropy lower bound of 0. This behavior is depicted in Figure 4.4. The reason that choosing a large maximal cycle length leads

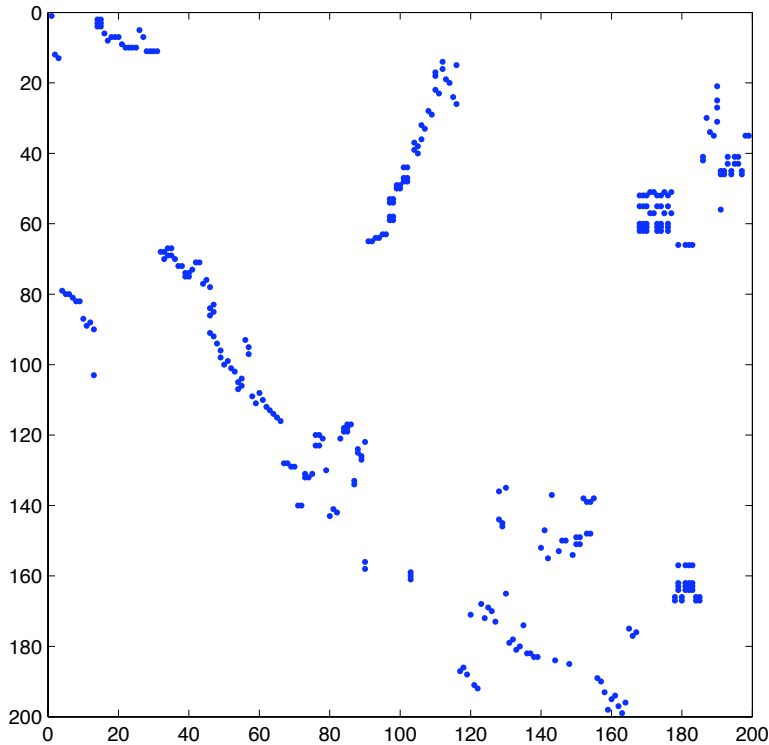


FIG. 4.3. A depiction of the nonzero entries of the 199×199 symbol transition matrix for the subshift of finite type constructed for Theorem 4.2.

to a 0 lower bound is that the corresponding isolating neighborhood produced by Algorithm 1 is a covering of the entire attractor, with corresponding trivial symbolic dynamics.

In principle, improving the bound requires only extra computational cost. Figure 4.5 shows the change in the computed entropy bound with increase in resolution of the grid (and corresponding increase in computational expense) for the Hénon map. The dip in the graph at depth 11 is of interest because, in general, we expect a monotonic increase in the computed entropy bound with increase in resolution of the grid. This non-monotonic behavior indicates that our choice of region of interest, \mathcal{S} , in Algorithm 4 is indeed sub-optimal. In fact, choosing \mathcal{S} to be the boxes in $\mathcal{G}^{(11)}$ contained in the isolating neighborhood \mathcal{N} returned by Algorithm 4 and Algorithm 1 on $\mathcal{G}^{(10)}$ would yield the same entropy as that computed at depth 10 and so it is possible to compute a higher entropy bound at this resolution.

4.3. Fold preimage removal. A priori knowledge of the Hénon map suggests another approach for constructing the region of interest \mathcal{S} . We notice that indices for cycles traveling too close to the “fold” of the attractor (at approximately $(1.2717, -0.0207)$) are necessarily trivial. Here, the Hénon map loses hyperbolicity, and the resulting induced map on homology maps the corresponding generator to zero. Out of curiosity, we now take the opposite approach of removing boxes from the covering of the attractor in an attempt to find an isolating neighborhood with interesting associated symbolic dynamics. Here we start with a box covering of the maximal invariant set (in this case, Hénon’s strange attractor) and remove a small box neighborhood of the fold. We then remove a fixed number of preimages of this

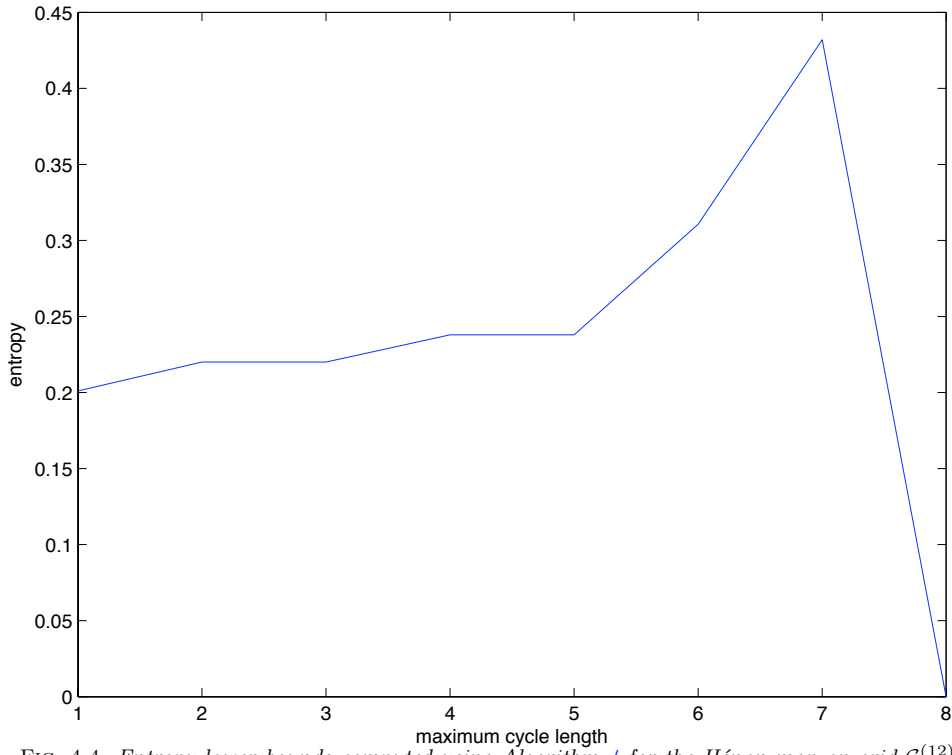


FIG. 4.4. Entropy lower bounds computed using Algorithm 4 for the Hénon map on grid $\mathcal{G}^{(12)}$ at varying maximal cycle lengths N .

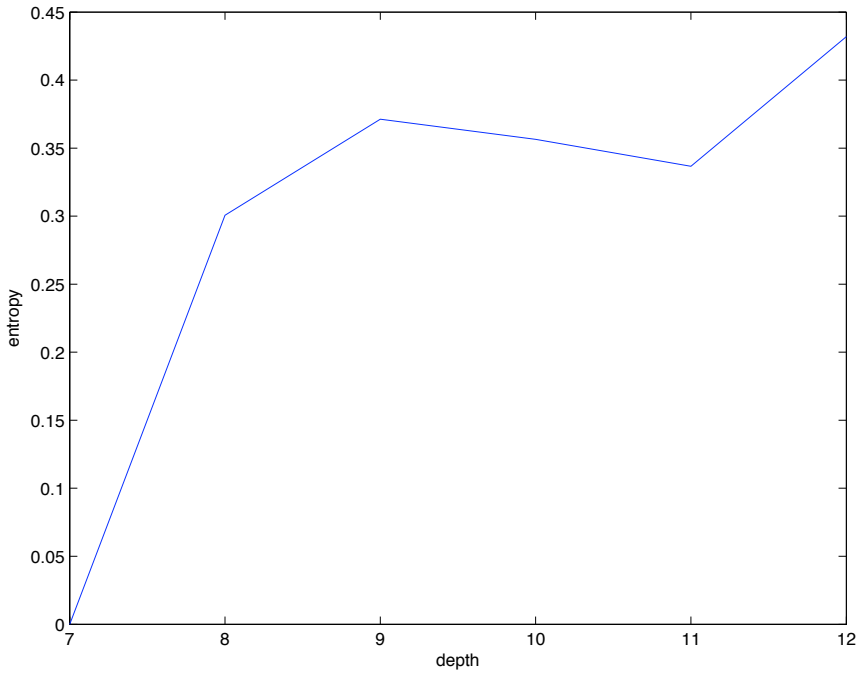


FIG. 4.5. Entropy lower bounds for the Hénon map computed on regions given by Algorithm 4 on grids $\mathcal{G}^{(d)}$ of varying depth d .

collection of boxes from the covering of the maximal invariant set. This procedure is outlined in Algorithm 10. From the resulting region of interest, we grow an isolating neighborhood and construct and verify symbolic dynamics as outlined in Algorithm 9.

ALGORITHM 10 (**Fold Preimage Removal for Constructing \mathcal{S}**).

INPUT: grid \mathcal{G}^d , combinatorial enclosure \mathcal{H} on \mathcal{G}^d ,
region $\mathcal{N}_f^0 \subset \mathcal{G}^d$ containing the fold point,
computational parameter `Max_Preimage_Iter`
OUTPUT: region of interest \mathcal{S}

```

 $\mathcal{S} = \text{fold\_preimage\_removal}(\mathcal{G}^d, \mathcal{H}, \text{Max\_Preimage\_Iter})$ 
 $\mathcal{N}_f = \mathcal{N}_f^0;$ 
 $\mathcal{S} = \mathcal{G}^d \setminus \mathcal{N}_f;$ 
for  $i = 1..\text{Max\_Preimage\_Iter}$ ,
     $\text{Fold\_Iter} = \mathcal{H}^{-1}(\mathcal{N}_f);$ 
     $\mathcal{S} = \mathcal{S} \setminus \mathcal{N}_f;$ 
end
return  $\mathcal{S};$ 

```

Figure 4.6 depicts entropy bounds resulting from computations made starting with Algorithm 10 and various values of `Max_Preimage_Iter`. Removing too few preimages of the fold boxes (`Max_Preimage_Iter` small) does not yield interesting symbolic dynamics since we are unable to isolate this set at the given resolution. Removing too many preimages (`Max_Preimage_Iter` large) results in a subshift system consisting of disjoint cycles with 0 entropy. At depth 12, `Max_Preimage_Iter` = 11 provides the highest entropy bound and this optimal constant increases at greater depths.

We obtain the following theorem by applying this third approach to the Hénon map.

THEOREM 4.3. (*Fold and preimage removal*) *The topological entropy of the Hénon map (4.1) is bounded from below by 0.4225.*

Outline of Proof. Starting with a covering of the Hénon attractor by elements in $\mathcal{G}^{(12)}$, we use Algorithm 10 to remove `Max_Preimage_Iter` = 11 preimages (under \mathcal{H}) of $(1.2717, -0.0207) + [-0.04, 0.04] \times [-0.002, 0.002]$, a neighborhood of the “fold”. We then use the resulting region of interest \mathcal{S} together with $\mathcal{G}^{(12)}$, and \mathcal{H} as the input for Algorithm 9. The computed index pair is shown in Figure 4.7, with the “fold” neighborhood in red. The homology map computed using Algorithm 3 is a map on 1281 generators of the first relative homology group. Algorithm 5 reduces the number of required generators to 191 by computing an appropriate shift equivalent index map. Finally, Algorithm 8 produces a topologically conjugate subshift on 129 symbols with topological entropy bounded from below by 0.4225. The result follows from Theorem 2.7. \square

5. Concluding Remarks. We have described an automated, algorithmic method for studying the dynamics of a discrete dynamical system $f : X \rightarrow X$. The method not only constructs a semi-conjugate subshift of finite type, but also uses this information to compute a rigorous lower bound on the topological entropy for the system. The essential ingredient to this approach is a computable “coarse” level of hyperbolicity in the map which is required to obtain a nontrivial Conley index. As the procedure

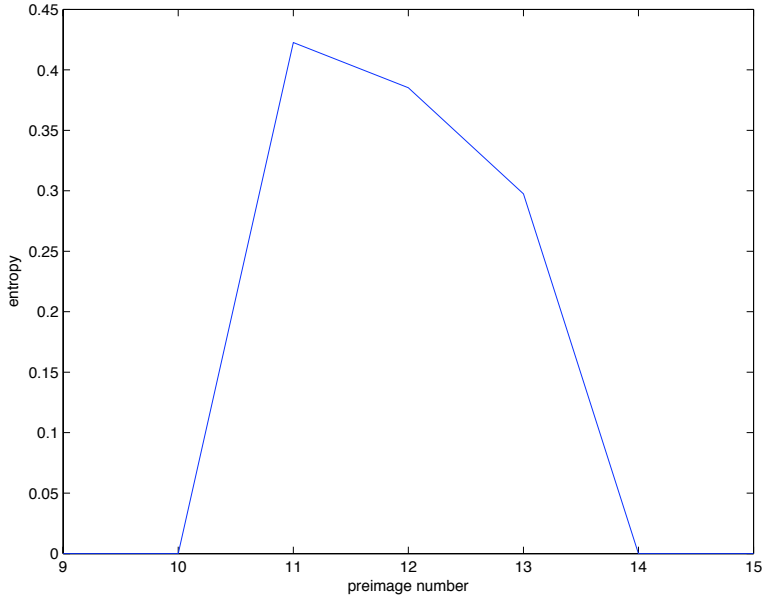


FIG. 4.6. Entropy lower bounds computed using the fold preimage removal technique for the Hénon map on grid $\mathcal{G}^{(12)}$. The horizontal axis gives the number, `Max.Preimage_Iter`, of preimages of the fold removed before growing the isolating neighborhood.

stands, greater computational effort may be employed to improve the bounds. However, further analysis and optimization of the procedure described in Section 3.1 for locating a region of interest should lead to even stronger results. A referee suggestion to consider more general sofic shifts rather than subshifts of finite type may also lead to the construction of semi-conjugate symbolic dynamical systems with higher entropy.

The index processing techniques introduced in Section 3.2 will enable further studies along these lines. As mentioned in the Introduction, even infinite dimensional systems may be studied in this manner. For such systems, it is necessary to incorporate both a dimension reduction for obtaining a computable system and analysis to overcome this reduction. These ideas are described in more detail in [DJM04] and would not, in principle, hinder entropy measurements of the type presented here.

6. Acknowledgements. The authors would like to thank Jim Wiseman and a referee for very helpful comments on the content and the structure of this paper. This work was made possible through the support of the Cornell University Summer 2006 REU program.

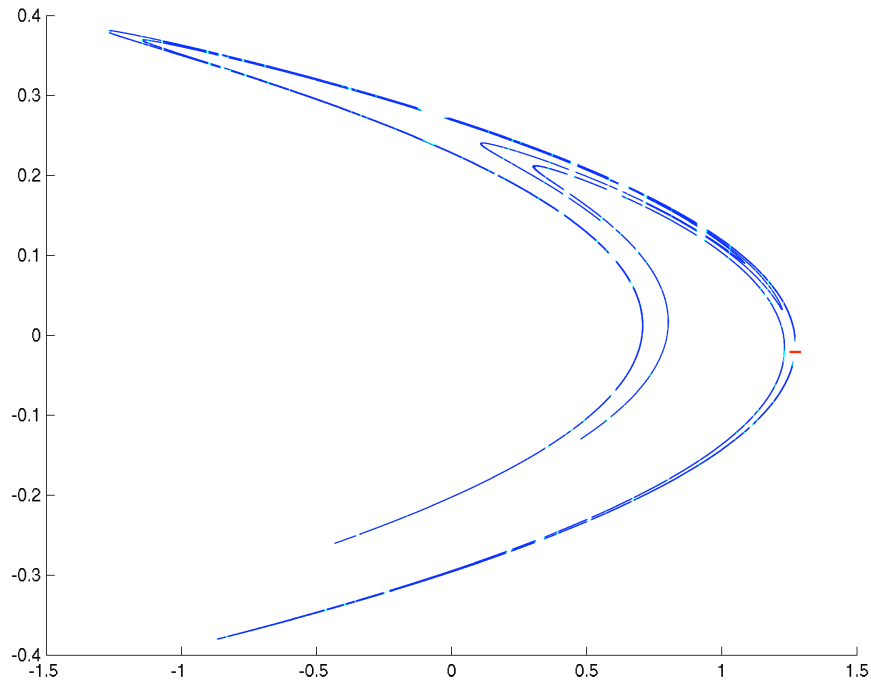


FIG. 4.7. The combinatorial index pair, $(\mathcal{P}_1, \mathcal{P}_0)$, constructed starting with Algorithm 10 at depth 12. (\mathcal{P}_0 is the collection of boxes shown in cyan.) The red rectangle shows the neighborhood of the fold point whose preimages were removed to construct the region of interest \mathcal{S} .

REFERENCES

- [AAC90] Roberto Artuso, Erik Aurell, and Predrag Cvitanović, *Recycling of strange sets. I. Cycle expansions*, *Nonlinearity* **3** (1990), no. 2, 325–359. MR MR1054579 (92c:58104)
- [ACE⁺87] Ditza Auerbach, Predrag Cvitanović, Jean-Pierre Eckmann, Gemunu Gunaratne, and Itamar Procaccia, *Exploring chaotic motion through periodic orbits*, *Phys. Rev. Lett.* **58** (1987), no. 23, 2387–2389. MR MR890474 (88g:58111)
- [Bow71] Rufus Bowen, *Periodic points and measures for Axiom A diffeomorphisms*, *Trans. Amer. Math. Soc.* **154** (1971), 377–397. MR MR0282372 (43 #8084)
- [Col02] Pieter Collins, *Symbolic dynamics from homoclinic tangles*, *Internat. J. Bifur. Chaos Appl. Sci. Engrg.* **12** (2002), no. 3, 605–617. MR MR1894884 (2003d:37053)
- [Con78] Charles Conley, *Isolated invariant sets and the Morse index*, CBMS Regional Conference Series in Mathematics, vol. 38, American Mathematical Society, Providence, R.I., 1978. MR MR511133 (80c:58009)
- [Cse99] Tibor Csendes (ed.), *INTLAB - INTERVAL LABORATORY*, Dordrecht, Springer, 1999, Proceedings of the International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics. Selected papers from the symposium (SCAN-98) held in Budapest, September 22–25, 1998, Reprinted in *Developments in reliable computing* [213–357, Kluwer Acad. Publ., Dordrecht, 1999], *Reliab. Comput.* **5** (1999), no. 3. MR MR1743193 (2001b:65007b)
- [Day03] S. Day, *A rigorous numerical method in infinite dimensions*, PhD. Thesis, Georgia Institute of Technology, 2003.
- [Dev89] Robert L. Devaney, *An introduction to chaotic dynamical systems*, second ed., Addison-Wesley Studies in Nonlinearity, Addison-Wesley Publishing Company Advanced Book Program, Redwood City, CA, 1989. MR MR1046376 (91a:58114)
- [DFJ01] Michael Dellnitz, Gary Froyland, and Oliver Junge, *The algorithms behind GAIO-set oriented numerical methods for dynamical systems*, *Ergodic theory, analysis, and efficient simulation of dynamical systems*, Springer, Berlin, 2001, pp. 145–174,

- 805–807. MR 2002k:65217
- [Die05] Reinhard Diestel, *Graph theory*, third ed., Graduate Texts in Mathematics, vol. 173, Springer-Verlag, Berlin, 2005. MR MR2159259 (2006e:05001)
- [DJM04] S. Day, O. Junge, and K. Mischaikow, *A rigorous numerical method for the global analysis of infinite-dimensional discrete dynamical systems*, SIAM J. Appl. Dyn. Syst. **3** (2004), no. 2, 117–160 (electronic). MR MR2067140
- [Eas75] Robert W. Easton, *Isolating blocks and symbolic dynamics*, J. Differential Equations **17** (1975), 96–118. MR MR0370663 (51 #6889)
- [Gal01] Zbigniew Galias, *Interval methods for rigorous investigations of periodic orbits*, Internat. J. Bifur. Chaos Appl. Sci. Engrg. **11** (2001), no. 9, 2427–2450. MR MR1862915 (2002i:65149)
- [Gal02] Z. Galias, *Obtaining rigorous bounds for topological entropy for discrete time dynamical systems*, Proc. Int. Symposium on Nonlinear Theory and its Applications, NOLTA'02 (Xi'an, PRC), 2002, pp. 619–622.
- [Hén76] M. Hénon, *A two-dimensional mapping with a strange attractor*, Comm. Math. Phys. **50** (1976), no. 1, 69–77. MR MR0422932 (54 #10917)
- [MM02] Konstantin Mischaikow and Marian Mrozek, *Conley index*, Handbook of dynamical systems, Vol. 2, North-Holland, Amsterdam, 2002, pp. 393–460. MR MR1901060 (2003g:37022)
- [MS80] Michał Misiurewicz and Bolesław Szewc, *Existence of a homoclinic point for the Hénon map*, Comm. Math. Phys. **75** (1980), no. 3, 285–291. MR MR581950 (82d:58046)
- [NBGM08] S. Newhouse, M. Berz, J. Grote, and K. Makino, *On the estimation of topological entropy on surfaces*, Contemporary Mathematics **469** (2008), 243–270.
- [Pil98] P. Pilarczyk, *Homology computation-software and examples*, Jagiellonian University, 1998, (<http://www.im.uj.edu.pl/~pilarczy/homology.htm>).
- [Rob95] Clark Robinson, *Dynamical systems*, Studies in Advanced Mathematics, CRC Press, Boca Raton, FL, 1995, Stability, symbolic dynamics, and chaos. MR MR1396532 (97e:58064)
- [RS88] Joel W. Robbin and Dietmar Salamon, *Dynamical systems, shape theory and the Conley index*, Ergodic Theory Dynam. Systems **8*** (1988), no. Charles Conley Memorial Issue, 375–393. MR MR967645 (89h:58094)
- [Szy95] Andrzej Szymczak, *The Conley index for decompositions of isolated invariant sets*, Fund. Math. **148** (1995), no. 1, 71–90. MR MR1354939 (96m:58154)
- [Szy96] ———, *The Conley index and symbolic dynamics*, Topology **35** (1996), no. 2, 287–299. MR MR1380498 (97b:58054)
- [Szy97] ———, *A combinatorial procedure for finding isolating neighbourhoods and index pairs*, Proc. Roy. Soc. Edinburgh Sect. A **127** (1997), no. 5, 1075–1088. MR MR1475647 (98i:58151)