SOFIC SHIFTS VIA CONLEY INDEX THEORY: COMPUTING LOWER BOUNDS ON RECURRENT DYNAMICS FOR MAPS

SARAH DAY* AND RAFAEL FRONGILLO†

Abstract. We extend and demonstrate the applicability of computational Conley index techniques for computing symbolic dynamics and corresponding lower bounds on topological entropy for discrete-time systems governed by maps. In particular, we describe an algorithm that uses Conley index information to construct sofic shifts that are topologically semi-conjugate to the system under study. As illustration, we present results for the two-dimensional Hénon map, the three-dimensional LPA map, and the infinite-dimensional Kot-Schaffer map. This approach significantly builds on methods first presented in [DFT08] and is related to work in [Kwa00, Kwa04].

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

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

1. Introduction.

Computational techniques based on Conley index theory have been used to study a variety of dynamical systems. With roots in algebraic topology and Morse theory, Conley index theory allows for the rigorous detection of invariant dynamics even in the presence of bounded error. Researchers have used Conley index theory to prove the existence of fixed points, connecting orbits, and (chaotic) horseshoes in systems ranging from the two-dimensional Hénon map [DJM05] and the three-dimensional Lorenz system of ODEs [MM95, MM98, MMS01] to the infinite dimensional Kot-Schaffer map [DJM04, DK13]. Many of these results demonstrate the successful inclusion of topological tools into a computational framework and have led to a growing collection of software available for extensions and similar studies. Perhaps the most ambitious, systematic application of these techniques may be found in [AKK+09] where a database of Conley indices is constructed via a systematic study of parameter space. While these prior studies present significant progress in this field, they tend to include either fairly localized dynamics [DJM05, DJM04] or rather coarse descriptions of global dynamics [AKK+09]. The work presented here offers a needed extension to allow for the detection of very complicated dynamics on a larger, more global scale.

In this paper, we extend work started in [DFT08] toward automating the processing of Conley index information in detecting highly complicated, often high-entropy, dynamics. Our approach builds on the work of Szymczak [Szy95] for computing symbolic dynamics from the Conley index and Kwapiscz [Kwa00, Kwa04] establishing the theoretical framework for extracting complicated dynamics from information of this type. As the regions where we compute index information grow in an attempt to approximate and measure a global attractor or maximal invariant set, the index information becomes increasingly difficult to interpret by hand, making automated methods necessary for meaningful analysis. The new approach we describe here focuses on computing semi-conjugate sofic subshifts supported by computed Conley index information. Towards this goal, we define the *labeled Conley index*, which contains additional phase space information necessary to the construction and verification of symbolic dynamics, and places us in the theoretical framework of cocyclic subshifts developed by Kwapiscz [Kwa00, Kwa04]. We improve on the methods in [DFT08] by introducing an algorithm to process labeled Conley index information into symbolic systems of

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

†University of Colorado, Boulder, Department of Computer Science, Boulder, CO 80309-0430 (raf@colorado.edu)

greater complexity, yielding higher computed lower bounds on topological entropy. Furthermore, in many cases the constructed system represents the maximal system supported by the index.

A secondary goal of this work is to demonstrate the applicability of this more automated approach to studying discrete-time systems of any dimension. Toward this goal, we present sample results for the well-studied two-dimensional Hénon map, the three-dimensional LPA population model, and the infinite-dimensional Kot-Schaffer (integrodifference) map.

To put our extensions to the work in [DFT08] in perspective, we recall the outline of the general procedure:

1. Compute an outer approximation (a combinatorial representation of the map that incorporates bounded error).
2. Find a region on which to compute Conley index information.
3. Compute the index.
4. Process the index information.

Steps 1 and 3 are well-studied and we will give only a brief description of these steps below, referring the reader to other sources for more details. Step 4 will be our primary focus in this paper with an additional, extended discussion of an updated approach for Step 2 as a secondary focus.

In what follows, we present necessary definitions, background, and motivating examples in Section 2, construct and discuss two key algorithms for analyzing cocyclic shifts and Conley indices and producing symbolic dynamics in Section 3, and provide sample results for the Hénon, LPA, and Kot-Schaffer models in Section 4. Following Theorem 4.1 in Section 4.2, we also briefly compare this method using Conley index theory with an approach based on *trellises*, a construction developed by Pieter Collins.

2. Setting and Examples. In this section we review some basic definitions and ideas from dynamical systems and computational Conley index theory and combine a number of ideas to introduce what we refer to as a *labeled Conley index representative*. The references [LM95], [Rob95], [KMM04], [Con78], [MM02], and [DFT08] contain further development and details for background material. In order to illustrate some of the basic ideas, we will also present some sample computations for the benchmark Hénon system in Section 2.6.

2.1. Symbolic Dynamics and Topological Entropy. The primary goal of the methods described in this paper is the construction of a symbolic dynamical system that is topologically semi-conjugate to the original system. The topological semi-conjugacy links the two systems, establishing the constructed symbolic system as a "lower bound" on the dynamics of the original system. In this framework, the symbolic system serves as a catalogue of dynamics—fixed points, periodic orbits, connecting orbits, and topological entropy are all readily available from a directed graph representation of the symbolic system as self-loops and cycles, connecting paths, and the log of the spectral radius (of the associated adjacency matrix for the graph) respectively.

The symbolic systems we focus on here are *sofic shifts*, which are defined as invariant subsystems of full shifts as follows. Given a set of symbols $\mathcal{A}$, also referred to as an *alphabet*, we define the (one-sided) *full shift* to be the set of all infinite symbol sequences

$$\mathcal{A}^{\mathbb{N}} := \{a_1 a_2 a_3 \ldots | a_i \in \mathcal{A}\}$$

together with the shift map $\sigma : \mathcal{A}^{\mathbb{N}} \to \mathcal{A}^{\mathbb{N}}$

$$\sigma(a_1 a_2 a_3 \ldots) := a_2 a_3 a_4 \ldots .$$

A *subshift* is given by a set $\Sigma \subset \mathcal{A}^{\mathbb{N}}$ that is forward invariant under σ, that is, $\sigma(\Sigma) \subseteq \Sigma$ so that $\sigma : \Sigma \to \Sigma$ is a well-defined subsystem. There are various presentations for a subshift $\sigma : \Sigma \to \Sigma$. One method for defining a subshift is creating a list of prohibited *words*, or blocks of symbols, and allowing all infinite symbol sequences that do not contain a prohibited word. This list can be chosen to be finite in the case of *subshifts of finite type*; methods described in [DFT08] focused on the construction of subshifts of this form. Graphical presentations, including *vertex shifts* and *edge shifts*, are also useful. See [LM95] for more details. In what follows we focus primarily on vertex presentations.

Definition 2.1. *Let G be a directed graph with vertex set $V(G)$ and edge set $E(G)$. Given an alphabet set $\mathcal{A}$ and vertex labeling $\mathcal{L} : V(G) \to \mathcal{A}$, we define the corresponding collection of label sequences of infinite walks in G by*

$$\Sigma_G = \{\mathcal{L}(v_1)\mathcal{L}(v_2)\mathcal{L}(v_3)\ldots \mid (v_i, v_{i+1}) \in E(G)\} \subseteq \mathcal{A}^{\mathbb{N}}.$$

A graph G (with labeling $\mathcal{L}$) is a vertex presentation *of a subshift $\Sigma \subseteq \mathcal{A}^{\mathbb{N}}$ if $\Sigma_G = \Sigma$. If, furthermore, for each vertex $v \in V(G)$ and pair of edges $(v, u_1), (v, u_2) \in E(G)$ out of v, we have $\mathcal{L}(u_1) \neq \mathcal{L}(u_2)$, then we say that G is* right-resolving.

Similarly, an *edge presentation* is a labeling of the edges in a graph, and again it presents the subshift of labelings of infinite walks in the graph. Note that we can verify $\sigma(\Sigma_G) \subset \Sigma_G$ since each $\boldsymbol{a} = \mathcal{L}(v_1)\mathcal{L}(v_2)\mathcal{L}(v_3)\ldots \in \Sigma_G$ has a corresponding walk $v_1v_2v_3\ldots$ in G, and $\sigma(\boldsymbol{a})$ will be given as the label sequence for the walk $v_2v_3v_4\ldots$ in G obtained by removing the first vertex/edge.

A subshift is *sofic* if it is a factor of a subshift of finite type. We instead take as our definition a well-known equivalent statement, namely that a subshift has a finite presentation [LM95].

Definition 2.2. *A subshift $\Sigma \subseteq \mathcal{A}^{\mathbb{N}}$ is* sofic *if it has a right-resolving vertex presentation with finitely many vertices.*

Subshifts can be characterized by their *language*, or *admissible blocks*, which are all words which appear in some point of the subshift.

Definition 2.3. *Given a subshift $\Sigma \subseteq \mathcal{A}^{\mathbb{N}}$, the* language $\mathcal{B}(\Sigma)$ *of Σ, also called the set of* admissible blocks, *is the set of finite words appearing in points of Σ. Formally, given $\boldsymbol{a} \in \mathcal{A}^{\mathbb{N}}$, let $[\boldsymbol{a}]_n = a_1a_2\cdots a_n$, and $\mathcal{B}_n(\Sigma) = \{[\boldsymbol{a}]_n : \boldsymbol{a} \in \Sigma\}$; then $\mathcal{B}(\Sigma) = \bigcup_{n\in\mathbb{N}_+} \mathcal{B}_n(\Sigma) = \{[\boldsymbol{a}]_n : \boldsymbol{a} \in \Sigma, n \in \mathbb{N}_+\}$.*

In particular, we can now see that the shift space Σ' is a subshift of Σ if and only if $\mathcal{B}(\Sigma') \subseteq \mathcal{B}(\Sigma)$.

It is important to note that for an appropriate choice of metric on $\mathcal{A}^{\mathbb{N}}$ (and hence on Σ_G), σ is a continuous map and $\sigma : \Sigma_G \to \Sigma_G$ is a dynamical system (see e.g. [Rob95]). Sofic subshifts given with a (finite) vertex shift presentation G are particularly nice for extracting dynamics. For example, if one is looking for a period n orbit, then one checks that there is a symbol sequence $\boldsymbol{a}^* = (a_1, a_2, \ldots) \in \Sigma_G$ such that $a_{i+n} = a_i$ for all $i \in \mathbb{N}$. This periodic symbol sequence corresponds to a cycle in G.

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 $f : S \to S$ be a continuous map. A set $W \subset S$ 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*

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

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

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

Theorem 2.5 (Robinson, [Rob95]). *Let G be a right-resolving vertex shift presentation of the sofic subshift $\sigma : \Sigma_G \to \Sigma_G$. Then*

$$h_{\text{top}}(\sigma|_{\Sigma_G}) = \lim_{N \to \infty} \frac{|\mathcal{B}_N(\Sigma_G)|}{N} = \log(\text{sp}(G))$$

where sp(G) is the spectral radius of the adjacency matrix A with

$$A(i, j) = \begin{cases} 1 & \text{if } (v_i, v_j) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

for G.

In essence, (n, ϵ, σ)-separation is encoded in the representation of the system and may be computed directly from the vertex shift presentation G.

2.2. The Itinerary Function and Topological Semi-conjugacy. Topological conjugacies and topological semi-conjugacies link two systems, preserving information about dynamics. The *itinerary function* defined below allows us to re-write a system $f : S \to S$ as a subshift. Our methods are designed to ensure that the itinerary function serves as a topological semi-conjugacy between the two systems.

Definition 2.6. *A continuous map $\rho : X \to Y$ is a* topological semi-conjugacy *from $\psi : Y \to Y$ to $\phi : X \to X$ if $\rho \circ \phi = \psi \circ \rho$ and ρ is surjective (onto). If, in addition, ρ is injective (one-to-one), then ρ is a* topological conjugacy*.*

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

$$\gamma_x := (\ldots, x_{-1}, x_0, x_1, \ldots) \tag{2.2}$$

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

$$\text{Inv}(N, f) := \{x \in N | \text{ there exists a trajectory } \gamma_x \text{ with } \gamma_x \subset N\}. \tag{2.3}$$

By construction, Inv(N, f) is in the domain of the itinerary function ρ given below in Definition 2.7.

Topological conjugacies preserve many properties of dynamical systems. For example, if ρ is a topological conjugacy between $\phi : X \to X$ and $\psi : Y \to Y$, then $y \in Y$ is a periodic point of period n under ψ (i.e. $\psi^n(y) = y$ and $\psi^k(y) \neq y$ for any positive integer $k < n$) if and only if $\rho^{-1}(y)$ is a periodic point of period n under ϕ.

If $f : S \to S$ is topologically conjugate to a sofic subshift, 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 sofic subshifts that we prove are topologically semi-conjugate to $f : S' \to S'$ for some appropriately defined subset S'.

Definition 2.7. *Suppose $N \subset X$ may be partitioned into a finite number of disjoint, closed subsets indexed by alphabet $\mathcal{A}$. That is, in addition to each N_a being closed, we have that $N =$*

$\cup_{a\in\mathcal{A}} N_a$ *and* $N_a \cap N_{a'} = \emptyset$ *for all* $a \neq a'$ *and* $a, a' \in \mathcal{A}$*. For* $S := \mathrm{Inv}(N, f)$*, the* itinerary function $\rho : S \to \mathcal{A}^{\mathbb{N}}$ *is given by* $\rho(x) = a_0a_1\ldots$*, where* $a_j = a$ *for* $f^j(x) \in N_a$*. The above assumptions ensure that* ρ *is well-defined.*

The itinerary function is continuous under an appropriate choice of metric and naturally satisfies the commutativity condition required for topological semi-conjugacy (that is, $\rho \circ f = \sigma \circ \rho$). (See [Dev89], [Rob95] for more details.) In what follows, we describe a procedure based on Conley index theory that allows us to construct a sofic subshift $\sigma : \Sigma_G \to \Sigma_G$ with vertex presentation G and $\Sigma_G \subset \mathcal{A}^{\mathbb{N}}$ so that for some $S' \subseteq S$, $\rho : S' \to \Sigma_G$ is surjective. The surjectivity condition completes the proof that ρ is a topological semi-conjugacy from $\sigma : \Sigma_G \to \Sigma_G$ to $f : S' \to S'$.

The following theorem also allows us to use a semi-conjugate sofic subshift to obtain a lower bound for the topological entropy of the system under study. In particular, since ρ being a semi-conjugacy from f to g ensures that $h_{\mathrm{top}}(f) \geq h_{\mathrm{top}}(g)$ [Rob95, Theorem IX.1.7], and $h_{\mathrm{top}}(\sigma) = \log(\mathrm{sp}(G))$ for sofic shift $\sigma : \Sigma_G \to \Sigma_G$ with vertex shift presentation G [LM95], we have the following.

Theorem 2.8. *Suppose that the itinerary function* ρ *is a semi-conjugacy from sofic subshift* $\sigma : \Sigma_G \to \Sigma_G$ *with vertex shift presentation* G *to* $f : S' \to S'$ *for some* $S' \subset X$*. Then*

$$h_{\mathrm{top}}(f) \geq \log(\mathrm{sp}(G))$$

where sp(G) *is the spectral radius of the adjacency matrix for* G*.*

2.3. Conley Index Theory and Ważewski's Principle. We now define the Conley index and state a homological version of the Ważewski Principle for maps (Theorem 2.14) which offers one basic mechanism for using the index to conclude the existence of dynamics.

We begin with some basic definitions.

Definition 2.9. *A compact set* $N \subset \mathbb{R}^n$ *is an* isolating neighborhood *if*

$$\mathrm{Inv}(N, f) \subset \mathrm{int}(N) \tag{2.4}$$

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

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

Definition 2.10. *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*

$$f_P(x) := \begin{cases} f(x) & \textit{if } x, f(x) \in P_1 \setminus P_0 \\ [P_0] & \textit{otherwise} \end{cases} . \tag{2.5}$$

Definition 2.11. *([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* $S = \mathrm{Inv}(\overline{P_1 \setminus P_0}, f)$*.*

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

Definition 2.12. *Two group homomorphisms,* $\phi : G \to G$ *and* $\psi : G' \to G'$ *on abelian groups* G *and* G' *are* shift equivalent *if there exist group homomorphisms* $r : G \to G'$ *and* $s : G' \to G$ *and a constant* $m \in \mathbb{N}$ *(referred to as the 'lag') such that*

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

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 2.13. *Let* $P = (P_1, P_0)$ *be an index pair for the isolated invariant set* $S = \mathrm{Inv}(\overline{P_1 \setminus P_0}, f)$ *and let* $f_{P*} : H_*(P_1, P_0) \to H_*(P_1, P_0)$ *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*}

$$\mathrm{Con}(S, f) := [f_{P*}]_s. \tag{2.6}$$

The Conley index for the isolated invariant set S given in Definition 2.13 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]). It is also computable given an appropriate computational framework, as we discuss in Section 2.5. Note that through our computational approach, there can be only finitely many generators on each level of homology, and only finitely many levels of nontrivial homology. Thus throughout the paper, given a basis of generators, we will write f_{P*} as a matrix; see Definition 2.16.

So far we have passed from continuous maps to induced maps on relative homology. Our overall goal, however, is to describe the dynamics of the original map. One theorem in this direction is a version of Ważewski's Principle for flows, reworked for the context of Conley index theory for maps. See, e.g., [KMM04] for a discussion of Ważewski's Principle and Theorem 10.91 in [KMM04] for the extension to the map context.

Theorem 2.14. *If* $\mathrm{Con}(S, f) \neq [0]_s$, *then* $S \neq \emptyset$.

In other words, a nontrivial index indicates that the associated isolated invariant set is nonempty. ∎

2.4. The Labeled Conley Index, Cocyclic Shifts, and Surjectivity. By recording additional information in the computation of the Conley index, we can use a modification of Theorem 2.14 to study finer structure. This extends the construction of the itinerary function ρ in Definition 2.7 to the simultaneous encoding of algebraic topological map information and follows closely the approach in [Szy97]. We then use our definition of the *labeled Conley index* to build *cocyclic shifts*, a class defined by Kwapisz [Kwa00], and use this framework to discuss surjectivity of the itinerary map. This step requires the following corollary to Theorem 2.14.

Corollary 2.15. *Let* $N = \cup_{a \in \mathcal{A}} N_a \subset X$ *be the union of pairwise disjoint, compact sets indexed by the alphabet* $\mathcal{A}$ *and let* $S := \mathrm{Inv}(N, f)$ *be the isolated invariant set relative to* N. *For* $\boldsymbol{b} = a_1 a_2 \ldots a_n$, $a_i \in \mathcal{A}$, *set*

$$f^{\boldsymbol{b}} := f_{N_{a_n}} \circ \cdots \circ f_{N_{a_1}}$$

where $f_{N_{a_i}}$ *denotes the restriction of the map* f *to the region* N_{a_i}. *Then for* $S_{\boldsymbol{b}} := \mathrm{Inv}(N, f^{\boldsymbol{b}}) \subset S$, *if*

$$\mathrm{Con}(S_{\boldsymbol{b}}, f^{\boldsymbol{b}}) \neq [0]_s, \tag{2.7}$$

then $S_{\boldsymbol{b}} \neq \emptyset$. *More specifically, there exists a point in* S *whose trajectory under* f *travels through the regions* $N_{a_1}, \ldots, N_{a_n}$ *in the prescribed order. Equivalently, the point* $\boldsymbol{b}^{\mathbb{N}} := a_1 a_2 \ldots a_n a_1 a_2 \ldots a_n \ldots$ *is in the image of the itinerary function* ρ *given in Definition 2.7.*

Indices of the type listed in (2.7) may be computed using the following additional information. (See [KMM04] for further detail about the support operator.)

Definition 2.16. *Consider an index pair* $P = (P_1, P_0)$ *with associated isolating neighborhood* $N = \overline{P_1 \setminus P_0}$ *and homology maps* $f_{P*} : H_*(P_1, P_0) \to H_*(P_1, P_0)$ *as described in Definition 2.13. Let*

$\{N_a : a \in \mathcal{A}\}$ be a decomposition of N into pairwise disjoint sets. For a generator $g \in H_*(P_1, P_0)$, let $|g|$ denote the support *of g. That is, $|g|$ is the topological object associated to the algebraic object g. Suppose that $\{g_i\}_{i=1}^n$ is a basis generating $H_*(P_1, P_0)$ such that for each i, $|g_i| \subset N_a$ for exactly one $a \in \mathcal{A}$. For ease of notation in what follows, we consider a matrix representation $M = M(f_{P*}, \{g_i\}_{i=1}^n)$ of the map f_{P*} in the basis $\{g_i\}_{i=1}^n$. The* labeling function*, $\ell : \{1, \ldots, n\} \to \mathcal{A}$, is given by $\ell(i) = a$ where $|g_i| \subset N_a$. We refer to $(f_{P*}, \{g_i\}_{i=1}^n, \ell)$ as a* labeled index map *and (M, ℓ) as a* labeled index representative.

A labeled index map $(f_{P*}, \{g_i\}_{i=1}^n, \ell)$, and, in particular, a labeled index representative (M, ℓ), may be used to compute $\mathrm{Con}(S', f^{\boldsymbol{b}})$ for $\boldsymbol{b} = a_1 \ldots a_n$, $a_i \in \mathcal{A}$ and $S' := \mathrm{Inv}(N_{a_1}, f^{\boldsymbol{b}})$. Using an approach developed by Szymczak [Szy95], we set

$$f_P^{ab}(x) := \begin{cases} f(x) & \text{if } x \in N_a \text{ and } f(x) \in N_b \\ [P_0] & \text{otherwise} \end{cases} , \tag{2.8}$$

and let $f_{P*}^{ab} : H_*(P_1, P_0 \cup (\cup_{c \neq a} N_c)) \to H_*(P_1, P_0 \cup (\cup_{c \neq b} N_c))$ be the corresponding map induced in relative homology. For $\boldsymbol{b} = a_1 a_2 \ldots a_n$, $(P_1, P_0 \cup (\cup_{c \neq a_1} N_c))$ is an index pair for the isolated invariant set $S_{\boldsymbol{b}} = \mathrm{Inv}(N_{a_1}, f^{\boldsymbol{b}})$ with index map $f_{P*}^{a_n a_1} \circ \cdots \circ f_{P*}^{a_1 a_2} : H_*(P_1, P_0 \cup (\cup_{c \neq a_1} N_c)) \to H_*(P_1, P_0 \cup (\cup_{c \neq a_1} N_c))$. Therefore,

$$\mathrm{Con}(S_{\boldsymbol{b}}, f^{\boldsymbol{b}}) = [f_{P*}^{\boldsymbol{b}}]_s \tag{2.9}$$

where $f_{P*}^{\boldsymbol{b}} := f_{P*}^{a_n a_1} f_{P*}^{a_{n-1} a_n} \circ \cdots \circ f_{P*}^{a_1 a_2}$. Using the matrix M given in Definition 2.16 as the representation of f_{P*} in the prescribed basis we have that f_{P*}^{ab} is the submatrix M^{ab} given by restricting M to the columns corresponding to the generators in N_a and rows corresponding to the generators in N_b. For example, if g_2 and g_{10} are the generators in N_a and g_3 is the generator in N_b then M^{ab} is the 1×2 matrix $M^{ab} = [M_{3,2} \quad M_{3,10}]$. The map $f^{\boldsymbol{b}}$ may now be written as the matrix product

$$f_{P*}^{\boldsymbol{b}} = M^{a_n a_1} M^{a_{n-1} a_n} \cdots M^{a_1 a_2}. \tag{2.10}$$

Note that since $f_{Pk} : H_k(P_1, P_0) \to H_k(P_1, P_0)$, these maps and the matrices in the product preserve homology level. In other words, grouping the generators $\{g_i\}$ by level of homology H_k yields a block diagonal form for M. This approach leads to the following, broader definition of a *labeled matrix*, a classification that will include labeled Conley index representatives.

Definition 2.17. *Given a matrix $M \in \mathbb{R}^{n \times n}$ and a labeling $\ell : \{1, \ldots, n\} \to \mathcal{A}$ for alphabet $\mathcal{A}$, we call the pair (M, ℓ) a* labeled matrix. *As above we write M^{ab} to denote the submatrix of M with rows $\ell^{-1}(a)$ and columns $\ell^{-1}(b)$, and let $M^{\boldsymbol{b}} := M^{a_{n-1} a_n} \cdots M^{a_2 a_3} M^{a_1 a_2}$, for $\boldsymbol{b} = a_1 \cdots a_n$.*

In order to apply Corollary 2.15 to $f^{\boldsymbol{b}}$ where $\boldsymbol{b} = a_1 \ldots a_n$, we must determine whether the linear map $f_{P*}^{\boldsymbol{b}} = M^{\boldsymbol{b} a_1}$ is shift equivalent to 0. For this we use the fact that for any finite-dimensional linear map A, we have $[A]_s = [0]_s$ if and only if A is nilpotent (see Proposition 10.93 in [KMM04]). We therefore design an algorithm that identifies symbol sequences for which no power of the corresponding matrix product may be 0. This allows us to verify the hypotheses of Corollary 2.15 and conclude surjectivity of the itinerary function onto the prescribed symbol sequences. This approach is a modification of the approach proposed in [DFT08] in which we developed methods to check that matrix products maintained a nonzero trace and were therefore not shift equivalent to 0 since trace is also preserved by shift equivalence. The approach we propose here is stronger since a nonzero trace is a necessary but not sufficient condition for a matrix being non nilpotent.

Kwapisz [Kwa00, Kwa04] defines *cocyclic shifts*, a useful framework in this context. As we discuss below, we adapt the notation and terminology to better fit our representations of the Conley index.

DEFINITION 2.18. *For the labeled matrix (M, ℓ), define the* cocyclic shift *to be*

$$\Sigma(M, \ell) := \{\boldsymbol{a} \in \mathcal{A}^{\mathbb{N}} | M^{[\boldsymbol{a}]_m} \neq 0 \textit{ for all } m\}. \tag{2.11}$$

Note that if $M^{\boldsymbol{b}a_1}$ is nilpotent for $\boldsymbol{b} = a_1 \dots a_m$, then $\boldsymbol{a} := \boldsymbol{b}^{\mathbb{N}}$ constructed by repeating the block $\boldsymbol{b}$, will be excluded from the cocyclic shift since for n the length of $\boldsymbol{b}$ and k sufficiently large, $M^{[\boldsymbol{a}]_{mk}} = (M^{[\boldsymbol{a}]_m})^k = (M^{\boldsymbol{b}})^k = 0$.

Definition 2.18 is a slight departure from the works introducing cocyclic subshifts [Kwa00, Kwa04]. In these works, cocyclic subshifts are defined by a collection of square matrices $\{\Phi_a \in \mathbb{R}^{n \times n} : a \in \mathcal{A}\}$; letting $\Phi^{\boldsymbol{b}} := \Phi_{b_m} \cdots \Phi_{b_1}$, the subshift is given by sequences $\boldsymbol{a} \in \mathcal{A}^{\mathbb{N}}$ with $\Phi^{[\boldsymbol{a}]_m} \neq 0$ for all m. It is straightforward to check that the definitions cover the same set of subshifts, as we now briefly argue. Given M and ℓ, we may let Φ_a be the $n \times n$ matrix with the only nonzero columns being those corresponding to $\ell^{-1}(a)$. Then $\Phi^{\boldsymbol{b}} \neq 0$ if and only if $M^{\boldsymbol{b}a} \neq 0$ for some $a \in \mathcal{A}$, and thus the corresponding shifts are the same. Conversely, given $\{\Phi_a\}_{a \in \mathcal{A}}$, we simply let M be the $n^2 \times n^2$ matrix which assigns exactly n rows/columns to each symbol, and set $M_{ab} = \Phi_a$ for all symbols $a, b \in \mathcal{A}$.

We will routinely visualize labeled matrices as a graph with vertices $\mathcal{A}$ and edges (a, b) labeled by the matrix M^{ab}; see Figure 2.1 for an example. The corresponding cocyclic subshift is therefore the set of labels of infinite walks in the graph which maintain nonzero matrix products. This graphical representation of a labeled matrix is a special case of a much more general object introduced by Kwapisz [Kwa00], called a *colored graph with propagation*. The latter is a directed graph with edges labeled by both symbols and arbitrary linear maps, which can be thought of as a generalization of sofic shifts with an extra constraint that the composition of maps be nonzero. In contrast, motivated by the Conley index, we merely take a labeled square matrix and write its blocks as edges in a complete graph (though we often drop edges labeled by a zero matrix).

$$M = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 5 & 0 & 0 & 2 & 0 \\ 0 & 4 & 1 & 0 & 2 \end{bmatrix}, \qquad \ell(i) = \begin{cases} 0 & i \in \{1, 2, 3\} \\ 1 & i \in \{4, 5\} \end{cases}$$

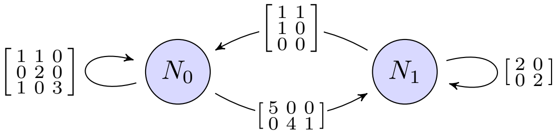

$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & 3 \end{bmatrix}$
N_0
N_1
$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$
$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 4 & 1 \end{bmatrix}$

FIG. 2.1. *A labeled matrix for alphabet $\mathcal{A} = \{0, 1\}$. The corresponding cocyclic shift is the full 2-shift, $\Sigma(M, \ell) = \{0, 1\}^{\mathbb{N}}$, as evidenced by the fact that the $(1, 1)$ entry of any matrix product is always positive.*

We can now give a basic result which connects labeled index computations to a semiconjugacy between the closure of periodic points of a particular subshift Σ and the original system.

DEFINITION 2.19. *Given a subshift $\Sigma \subseteq \mathcal{A}^{\mathbb{N}}$, we denote the set of its periodic points by*

$$\mathcal{P}(\Sigma) = \{a_1 a_2 \ldots \in \Sigma : \textit{ for some } n, \; a_{n+i} = a_n \textit{ for all } i \in \mathbb{N}\}.$$

In the notation of Corollary 2.15, a sequence in $\mathcal{P}(\Sigma)$ may be written as $\boldsymbol{b}^{\mathbb{N}}$ where $\boldsymbol{b} = a_1 a_2 \dots a_n$

is a repeated block in the sequence. We define the periodic closure*, $\overline{\mathcal{P}}(\Sigma)$ of Σ, to be the subshift given by the closure of $\mathcal{P}(\Sigma)$.*

When applied to cocyclic subshifts, the periodic closure corresponds to the set of non-transient points [Kwa00, Section 5]. These points can be decomposed into a union of (possibly overlapping) topologically transitive and irreducible systems, which may be further decomposed into aperiodic systems [Kwa00, Section 6]. Thus, the periodic closure of a cocyclic subshift, which plays a central role in this paper, has considerable structure.

We now state the key result which allows us to build symbolic systems from Conley index information.

PROPOSITION 2.20. *Following Definition 2.7, consider a partition $\{N_a\}_{a\in\mathcal{A}}$ of isolating neighborhood $N = \cup_{a\in\mathcal{A}} N_a$ into disjoint, closed sets with labels in the alphabet $\mathcal{A}$. For $S := \mathrm{Inv}(N, f)$ let $\rho : S \to \mathcal{A}^{\mathbb{N}}$ on $S := \mathrm{Inv}(N, f)$ be the corresponding itinerary function. Suppose that for a collection of periodic points $\Sigma \subseteq \mathcal{P}(\mathcal{A}^{\mathbb{N}})$, each point $\boldsymbol{b}^{\mathbb{N}} \in \Sigma$ satisfies $\mathrm{Con}(S_{\boldsymbol{b}}, f^{\boldsymbol{b}}) \neq [0]_s$. Then for the closure of the set of periodic orbits, $\Sigma' := \overline{\Sigma}$, the (restricted) itinerary function $\rho : S' \to \Sigma'$, where $S' := \rho^{-1}(\Sigma')$, is a topological semi-conjugacy.*

Proof. By Corollary 2.15, $\mathrm{Con}(S_{\boldsymbol{b}}, f^{\boldsymbol{b}}) \neq [0]_s$ for periodic point $\boldsymbol{b}^{\mathbb{N}} \in \Sigma$ implies that $\boldsymbol{b}^{\mathbb{N}} \in \rho(S)$, the image of ρ. Hence, $\Sigma \subseteq \rho(S)$. Since ρ is continuous and S is compact, $\rho(S)$ must be compact. Therefore $\Sigma' := \overline{\Sigma} \subseteq \rho(S)$. Using $S' := \rho^{-1}(\Sigma')$, $\rho : S' \to \Sigma'$ is surjective. Since ρ is continuous and $\rho \circ f = \sigma \circ \rho$ by construction, $\rho : S' \to \Sigma'$ is a topological semi-conjugacy. □

In general, our constructed subshift $\Sigma' := \overline{\Sigma}$ will be a subshift of $\Sigma(M, \ell)$, the cocyclic shift for the labeled index. The maximal subshift obtainable by this approach would be $\Sigma' = \overline{\mathcal{P}}(\Sigma(M, \ell))$ since the Proposition produces subshifts that are closures of collections of periodic points and a point $\boldsymbol{a}$ approximated by a periodic point $\boldsymbol{b}^{\mathbb{N}}$ having a nontrivial Conley index under (M, ℓ) requires that each block has a nonzero matrix product. (Note that once we fix a labeled index pair, this maximal subshift $\overline{\mathcal{P}}(\Sigma(M, \ell))$ does not depend on the choice of labeled index representative (M, ℓ).) Proposition 2.20 underpins our approach, but leaves an important question unresolved: *How can one verify that all periodic points in Σ have nontrivial Conley indices, when in general there are infinitely many of them?* This is precisely our goal in Section 3.

2.5. Computational Conley Index Theory. Importantly, the isolating neighborhoods, index pairs, and labeled Conley indices required to construct the necessary cocyclic sub shifts are computable given an appropriate combinatorial framework. This process can be made fairly general, see e.g. [KMV14, KMV16]. For illustration, we now outline one constructive approach that we used to produce the sample results throughout the paper. Building the framework typically begins with a discretization of the phase space. One approach is to produce a cubical grid over a rectangular subdomain. For example, a *uniformly-subdivided cubical grid* on a rectangular set $W = \prod_{k=1}^{n} [x_k^-, x_k^+] \subset \mathbb{R}^n$ is given by

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

where $r_k = x_k^+ - x_k^-$ is the radius of W in the kth coordinate and the depth d is a nonnegative integer. In order to formalize the connection between sets of grid elements and subsets of the phase space, we define the *topological realization* of a collection of grid elements $G \subset \mathcal{G} = \mathcal{G}^{(d)}$ as $|G| := \cup_{B\in G} B \subset \mathbb{R}^n$.

We next produce a discrete version of the continuous map f acting on the grid. By construction, this discretization records outer bounds on images under f. Typically, one uses outward rounding interval arithmetic and analysis to produce outer bounds on images $f(|G|)$ for individual grid elements $G \in \mathcal{G}$, which are then covered by grid elements to produce the combinatorial outer bound. The end result is a *combinatorial outer approximation* $\mathcal{F} : \mathcal{G} \rightrightarrows \mathcal{G}$, mapping a grid element $G \in \mathcal{G}$ to a collection of grid elements $\mathcal{F}(G) \subset \mathcal{G}$ and satisfying $f(|G|) \subset |\mathcal{F}(G)|$.

If, in addition, we require that the topological realization $|\mathcal{F}(G)|$ is *acyclic* for every $G \in \mathcal{G}$, that is has the topology of a point, then $\mathcal{F}$ is amenable to algorithms for computing isolating neighborhoods, index pairs, and maps on relative homology. For these computations, it is helpful to view the combinatorial outer approximation as a directed graph, with a vertex set corresponding to the collection of grid elements in $\mathcal{G}$ and edge set $E = \{(G_1, G_2) \in \mathcal{G} \times \mathcal{G} \mid G_2 \in \mathcal{F}(G_1)\}$. In this setting, isolating neighborhoods and index pairs are given as topological realizations of computed subsets of grid elements. Algorithms for using combinatorial outer approximations to produce these sets and compute Conley indices are presented in more detail in [DFT08, DJM05].

Figure 2.2 shows a depiction of an image under a combinatorial outer approximation and Figure 2.3 shows an isolating neighborhood and index pair computed using a combinatorial outer approximation of the Hénon map. Other sample results for the Hénon map as well as the LPA model and Kot-Schaffer equation follow in Sections 2.6 and 4. The combinatorial outer approximations for the low dimensional systems – the Hénon map (see [DJM05, DFT08]) and the LPA model, are constructed via a straightforward application of outward rounding interval arithmetic, while the construction for the Kot-Schaffer model requires more sophisticated analysis to incorporate truncation and dimension reduction into the procedure (see [DK13]).

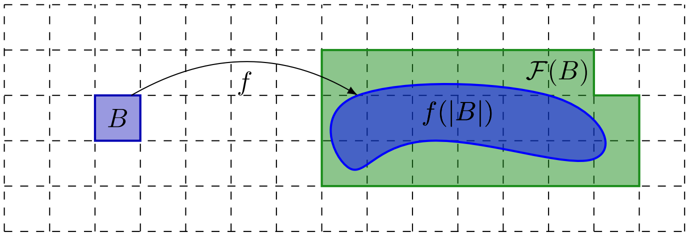


Fig. 2.2. *A depiction of an outer approximation image, $\mathcal{F}(B)$, for grid element B and continuous map f.*

2.6. Examples. We conclude with several examples illustrating the above definitions and motivating the need for an automated approach. The first, in Figure 2.3, is a simple example from the Hénon system (see Section 4.1) showing a period-two orbit. Here we can see how the labeled Conley index representative is split into submatrices on the edges corresponding to the maps defined in Corollary 2.15 and Definition 2.16. In this example, as in Figure 2.1, the matrices are simple enough that one can easily verify that the matrix products M^{121} and M^{212} are non-nilpotent, thus yielding a nontrivial Conley index for the maps f_P^{12} and f_P^{21}. Hence, from Proposition 2.20 we can conclude that the vertex presentation on the right, which is its own periodic closure, is semi-conjugate to the Hénon map on a portion of the phase space. The second example (Figure 2.4), also from Hénon, gives rise to a more complicated Conley index, but still the corresponding cocyclic shift is easy to compute by hand, and again we may apply Proposition 2.20.

In contrast to these simple cases, and that of Figure 2.1, we give a more complicated example in Figure 2.5. Here several regions have multiple generators of homology, and the resulting labeled Conley index representative is much larger. As a consequence, manually enumerating the periodic symbol sequences yielding nonnilpotent matrix products, and therefore nontrivial indices, would

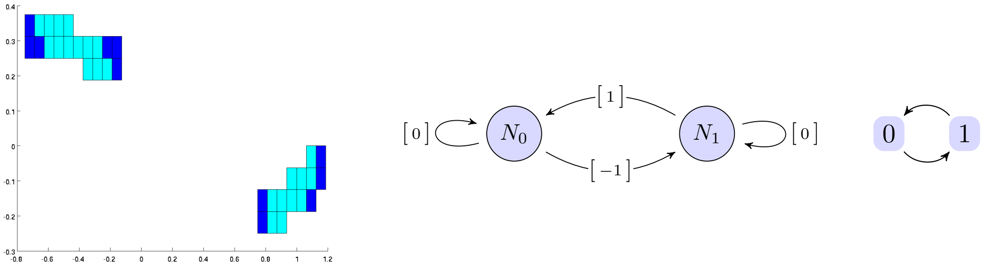


FIG. 2.3. *A periodic orbit from the Hénon map on the domain $[-2,2]^2$ at depth 6. The index pair (left) contains two disjoint regions, giving rise to a labeled Conley index representative (middle) whose cocyclic shift is that of a simple period-two orbit (right).*

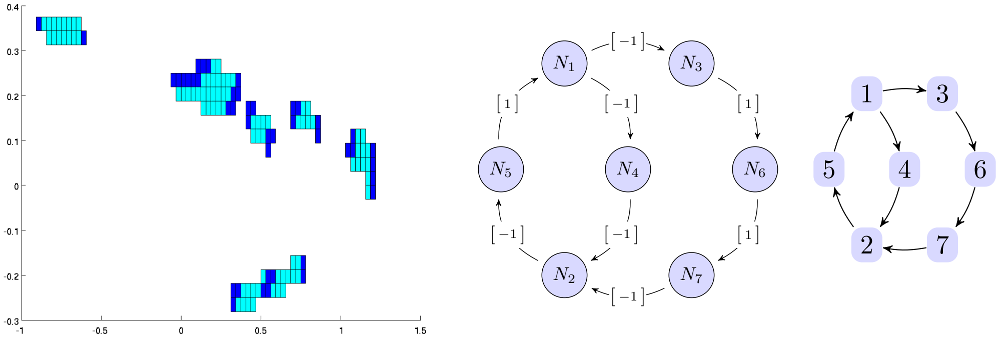


FIG. 2.4. *Chaotic dynamics in the Hénon map on domain $[-2,2]^2$ at depth 7. The index pair (left) is grown from the isolating neighborhoods of period-4 and period-6 orbits (see Section 4.2), giving a labeled Conley index representative (middle) whose cocyclic shift is the vertex shift shown (right). Omitted edges in the index representative correspond to the zero matrix, in this case $[\,0\,]$.*

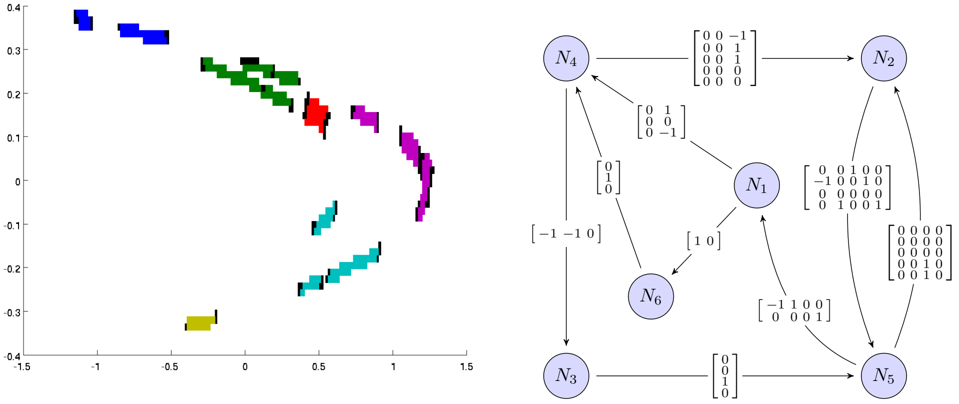

0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5
N_4
$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
N_2
$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
N_1
$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$
$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
$\begin{bmatrix} -1 & -1 & 0 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \end{bmatrix}$
N_6
$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
N_3
$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$
N_5

FIG. 2.5. *A more complicated example of chaos in the Hénon map on domain $[-2,2]^2$ at depth 8, constructed from the isolating neighborhoods of three periodic orbits (see Section 4.2). Attaching labels to the 6 disjoint colored regions of the index pair (left) results in the index map representative (right). The complexity of the index map representative renders manual calculations tedious. See Figure A.1 for the result of Algorithm 2, which gives a vertex presentation of this cocyclic shift (which is sofic in this case).*

be extremely tedious, and impractical for even more complicated examples. A main focus of this project is the development of the automated methods presented in Section 3.1 to handle these larger examples. See the Appendix (Figure A.1) for results obtained using this automated approach on the example presented in Figure 2.5.

3. Processing Conley index information.

As seen in Section 2.6, constructing semi-conjugacies to symbolic dynamics from a labeled Conley index (M, ℓ) becomes very difficult by hand as the number of regions and generators increases. Day et al. [DFT08] present an algorithm which processes the labeled Conley index and proves a semi-conjugacy to a subshift of finite type. In this section, we develop a new algorithm for constructing semi-conjugacies to symbolic dynamics, focusing instead on the broader class of sofic shifts. Specifically, we will construct a sofic subshift of the cocyclic shift $\Sigma(M, \ell)$. In Section 3.2 we show that this new algorithm is more powerful and efficient than the previous algorithm of [DFT08]. In addition, this approach achieves the optimal semi-conjugacy, $\overline{\mathcal{P}}(\Sigma(M, \ell))$ (see discussion following Proposition 2.20), whenever it terminates within the specified number of iterations, a case which arises overwhelmingly in practice.

3.1. Computing sofic subshifts of cocyclic shifts.

We first discuss the computation of a sofic subshift Σ of a cocyclic shift $\Sigma(M, \ell)$. Naturally, the algorithm given here will then be applied to labelled Conley index representatives (M, ℓ), but the algorithm itself is agnostic to the origin of M and ℓ. As $\Sigma(M, \ell)$ is only defined formally in Definition 2.18, the constructed subshift $\Sigma = \Sigma_G$, given by a vertex or edge shift presentation G, offers a more useful, searchable, catalog of dynamics (see Section 2.1). Our algorithm is closely related to existing constructions for cocyclic subshifts [Kwa04], as we describe below.

To construct our sofic subshift of $\Sigma(M, \ell)$, we iteratively compute matrix products until some redundancy is detected. The redundancy we search for is when the image space of the matrix product has been previously seen and recorded. As the matrices are labelled according to ℓ, we must also record the relevant symbol corresponding to the matrix product; specifically, for a block $\boldsymbol{b} = a_1, \ldots, a_n$, we represent the corresponding matrix product $M^{\boldsymbol{b}}$ as the pair $\{a_n, \mathsf{ech}(M^{\boldsymbol{b}})\}$, where ech denotes the column echelon form, which uniquely encodes the matrix image (column) space.[1]

Referring to a (symbol, image space) pair as a *state*, we iteratively construct a *state graph*. The state graph is initialized by constructing states (symbol, full image space) for each symbol in the alphabet. The algorithm then considers a matrix product resulting from a matrix multiplication out of an existing state S. This product may yield a new state S', which is then added to the graph, or may yield a previously observed state S'; in both cases an edge is added from S to S'. In this way, the state graph grows by adding edges out of existing states and possibly adding new states when new pairs (symbol, image space) are observed. When a new edge is added between existing states, the algorithm has discovered "redundancy" in the matrix products. Because images that build on that state have been, or will be, computed separately, there is no need to track further products along this separate path. Often, this procedure reduces the problem of enumerating a typically infinite list of nonzero matrix products, to a finite set of computations to check nontrivial matrix image spaces and find redundancies. We describe this procedure formally in Algorithm 1.[2]

As an example, consider the two-region cocyclic shift $\Sigma(M, \ell)$ over alphabet $\mathcal{A} = \{0, 1\}$, where

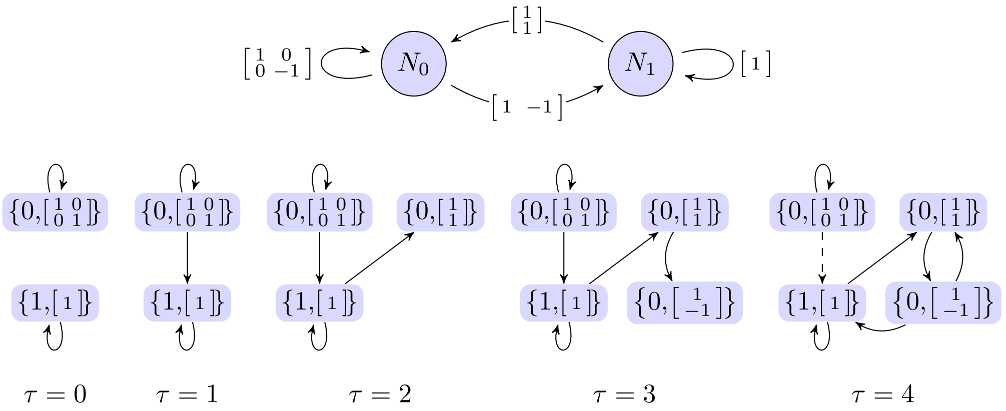
[1] While we assume infinite precision floating-point arithmetic for now, we will ultimately work with integer-valued matrices in § 3.2, and thus the implementation of ech in Algorithm 1 will produce rational-valued matrices via exact computations. One could also use the Hermite normal form, which directly computes an integer-valued representation.

[2] Code for Algorithms 1 and 2 available at `https://github.com/caosuomo/sofproc`.

Algorithm 1 Computing a Sofic Subshift from a Cocyclic Shift

Input: Matrix $M \in \mathbb{R}^{n\times n}$, labels $\ell : \{1, \dots, n\} \to \mathcal{A}$, and number of iterations τ
Output: A right-resolving vertex presentation $G^{(\tau)}$ of a sofic subshift of $\Sigma(M, \ell)$

```
procedure SoficProcessor($M$, $\ell$, $\tau$)
    $G \leftarrow$ empty graph with states $\{a, I_a\}$ for all $a \in \mathcal{A}$
                        # Here $I_a$ is the $n_a \times n_a$ identity matrix, where $n_a = |\ell^{-1}(a)|$.
    $Q \leftarrow$ queue initially containing all states of $G$
    while $Q$ is not empty, or until $\tau$ iterations, do
        dequeue state $\{a, A\}$ from $Q$
        for $b \in \mathcal{A}$ do
            $B \leftarrow \mathsf{ech}(M^{ab} A)$
            if $B \neq 0$ then
                if $\{b, B\} \notin G$ then
                    add state $\{b, B\}$ to $G$ and enqueue $\{b, B\}$ into $Q$
                add edge $(\{a, A\}, \{b, B\})$ to $G$
    return $G^{(\tau)} = G$                 # Vertex labels are given by $\mathcal{L}(\{a, A\}) = a$
```



N_0
N_1
$\left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right]$
$\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$
$\left[\begin{smallmatrix}1&-1\end{smallmatrix}\right]$
$[1]$
$\{0,[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}]\}$
$\{1,[1]\}$
$\{0,[\begin{smallmatrix}1\\1\end{smallmatrix}]\}$
$\{0,[\begin{smallmatrix}1\\-1\end{smallmatrix}]\}$
$\tau = 0$
$\tau = 1$
$\tau = 2$
$\tau = 3$
$\tau = 4$

Fig. 3.1. *A run of Algorithm 1 on a simple two-state example $\Sigma(M, \ell)$ (above). The algorithm initializes with states $\{0,[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}]\}$ and $\{1,[1]\}$, creates state $\{0,[\begin{smallmatrix}1\\1\end{smallmatrix}]\}$ in iteration 2, creates state $\{0,[\begin{smallmatrix}1\\-1\end{smallmatrix}]\}$ in iteration 3, and then terminates after iteration 4 with an empty queue. Note that there is no edge from $\{0,[\begin{smallmatrix}1\\1\end{smallmatrix}]\}$ to a state of the form $\{1,\cdot\}$, as in iteration 3 the algorithm detects that such a transition would yield a 0 matrix product.*

If we had run Algorithm 2 instead, the only difference would be the removal of the dashed edge in the final step. For this example, this final step does not change the shift space, as both graphs are vertex presentations of the even shift (see Section 4.3 and Figure 4.4).

$M = \left[\begin{smallmatrix}1&0&1\\0&-1&1\\1&-1&1\end{smallmatrix}\right]$ and $\ell(1) = \ell(2) = 0$, $\ell(3) = 1$, depicted in Figure 3.1. To see how the algorithm captures "redundancy" in the matrix product image spaces, consider a block $\boldsymbol{b} = a_1 \dots a_n$ with $a_1 = 1$ and $a_n = 0$. The output of Algorithm 1 (Figure 3.1, $\tau = 4$) encodes the fact that the corresponding matrix product $M^{\boldsymbol{b}}$ must either be in the space spanned by $\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$ or by $\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]$. In this simple example, Algorithm 1 terminates after only 4 iterations through the while loop. Even on very large graphs that arise in practice, however, the algorithm often terminates within $2|\mathcal{A}|$ iterations.

Fixing labeled matrix (M,ℓ), let $G^{(\infty)}$ be the countably infinite graph given by running Algorithm 1 on input (M,ℓ,∞), that is, with $\tau=\infty$. Viewing $\mathsf{ech}(A)$ as a representation of $\mathrm{im}(A)$, we see that $G^{(\infty)}$ is precisely the graph $G_{\mathbf{P}}$ constructed in the proof of Theorem 1 in [Kwa04]. Thus, one may view Algorithm 1 as performing a breadth-first search of $G_{\mathbf{P}}$ starting from the full image states. Its correctness is established by the following Proposition, which follows from the observation that $\Sigma_{G_{\mathbf{P}}}=\Sigma(M,\ell)$ [Kwa04, Theorem 1].

Proposition 3.1. *Let $G^{(\tau)}$ be the labeled graph output from Algorithm 1 on input (M,ℓ,τ). Then $G^{(\tau)}$ is a right-resolving vertex presentation of $\Sigma_{G^{(\tau)}}\subseteq\Sigma(M,\ell)$. Moreover, if the algorithm terminates with an empty queue, then $\Sigma_{G^{(\tau)}}=\Sigma(M,\ell)$, and in particular, $\Sigma(M,\ell)$ is sofic.*

Proof. For any τ, we have by construction that $G^{(\tau)}$ has finitely many states (at most $(\tau+1)|\mathcal{A}|$). Defining $\mathcal{L}(\{a,A\})=a$, we see that $G^{(\tau)}$ is right-resolving, and thus is a right-resolving vertex shift presentation of the sofic shift $\Sigma_{G^{(\tau)}}$. (See Definitions 2.1 and 2.2.) From [Kwa04, Theorem 1] we have $\Sigma_{G_{\mathbf{P}}}=\Sigma(M,\ell)$, and as observed above, we also have $\Sigma_{G^{(\infty)}}=\Sigma_{G_{\mathbf{P}}}=\Sigma(M,\ell)$. The results now follow from the following two observations. First, for all $\tau<\infty$, we have $\Sigma_{G^{(\tau)}}\subseteq\Sigma_{G^{(\infty)}}$ as no states are ever removed. Second, if the algorithm terminates with empty queue after τ iterations, then $G^{(\tau)}=G^{(\infty)}$, as no new states will ever be added. □

The works [Kwa00, Kwa04] establish many interesting and useful properties of cocyclic subshifts. For example, it is shown that cocyclic subshifts can be decomposed into subshifts that have the *specification* property [Kwa00, Theorem 7.1]. (A subshift X has specification if there is some k_0 such that for all $\boldsymbol{a},\boldsymbol{a}'\in\mathcal{B}(X)$ and all $k\geq k_0$ there exists $\boldsymbol{b}\in\mathcal{B}_k(X)$ such that $\boldsymbol{aba}'\in\mathcal{B}(X)$.) As specification implies that the shift is *almost sofic*, meaning it can be approximated from within by sofic subshifts whose entropy approaches the true entropy [Boy00, p. 66], one may ask whether the sequence of sofic shifts presented by $G^{(\tau)}$ in Algorithm 1 achieve this approximation. We can see that the answer is yes, using a construction in [Kwa04, Lemma 1] of a sequence of finite subgraphs of $G^{(\infty)}=G_{\mathbf{P}}$ whose entropy approximates that of $\Sigma(M,\ell)$; as $G^{(\infty)}$ is reachable from the full image states, Algorithm 1 eventually reaches these finite subgraphs, and the entropy converges.

Proposition 3.2. $\lim_{\tau\to\infty}h(G^{(\tau)})=h(G^{(\infty)})=h(\Sigma(M,\ell))$.

Proof. For any state s in $G^{(\infty)}$ we can define $\tau(s)$ to be the iteration at which s was added to the graph. Given any finite subgraph G of $G^{(\infty)}$, clearly G is a subgraph of $G^{(\tau)}$ where $\tau=\max_{s\in G}\tau(s)$. As $G^{(\infty)}$ is isomorphic to $G_{\mathbf{P}}$ as observed above, the result follows from [Kwa04, Lemma 1]: for any $\epsilon>0$, there exists a finite subgraph $G_1(\epsilon)$ of $G_{\mathbf{P}}$ such that $h(\Sigma_{G_1(\epsilon)})\geq h(\Sigma(G_{\mathbf{P}}))-\epsilon=h(\Sigma(M,\ell))-\epsilon$. □

Theoretically, Algorithm 1 could produce $G^{(\infty)}$ which is countably infinite even in cases when $\Sigma(M,\ell)$ is sofic, though this does not happen for any of our computations involving the Conley index. For example, Figure A.2 shows a cocyclic representation of the full 2-shift where $G^{(\infty)}$ has infinitely many nodes. (Indeed, as the entries are nonnegative, [Kwa00, Theorem 10.2] shows that the cocyclic subshift must be sofic.) In this example, however, one may note that starting Algorithm 1 from the image space $\left[\begin{smallmatrix}1\\0\\0\end{smallmatrix}\right]$ terminates with the minimal presentation of the full 2-shift. It therefore may be of interest, should this situation arise frequently in practice, to choose the initial image spaces more carefully. More generally, we would like an algorithm which can more directly determine if $\Sigma(M,\ell)$ is sofic, and if so, give a finite presentation of it.

3.2. Semi-conjugate sofic subshifts from the labeled Conley index. The algorithm given in the previous section produces a vertex presentation of a sofic subshift of a given cocyclic shift $\Sigma(M,\ell)$. We now return to our goal of producing semi-conjugate symbolic dynamics from Conley index information. If the cocyclic shift is generated by a labeled Conley index representative (M,ℓ)

Algorithm 2 Producing a Semi-conjugate Sofic Shift

Input: Labeled Conley index representative $(M \in \mathbb{Z}^{n\times n},\ \ell : \{1,\dots,n\} \to \mathcal{A})$
Output: A right-resolving vertex presentation G' of a semiconjugate sofic shift

```
procedure SemiConjugateSoficShift($M$, $\ell$, $\tau$)
    $G \leftarrow$ SoficProcessor($M$, $\ell$, $\tau$)          # Algorithm 1
    $G' \leftarrow \mathsf{cyc}(G)$          # restriction to cycles in $G$; see Def 3.3
    return $G'$
```

for a map f, then Proposition 2.20 allows us to define a shift Σ' that is topologically semi-conjugate to f on an appropriate subset. Note however that an extra step is required to relate the subshift produced by Algorithm 1 to our original map f. Specifically, in order to invoke Proposition 2.20 we restrict the computed shift to cycles, as described below. The complete procedure is given as Algorithm 2.

Definition 3.3. *The graph H is a* cyclic subgraph of G *if $E(H)$ is a cycle in G and $V(H)$ is the set of vertices appearing in $E(H)$. Then $\mathsf{cyc}(G)$ is the graph union of all cyclic subgraphs of G. In other words, $\mathsf{cyc}(G)$ is given by removing all edges and vertices from G which are not contained in cycles.*

We will need this restriction to cycles in order to use Proposition 2.20 to prove a semi-conjugacy to a closure of periodic points. As the following lemma shows, restricting to the union of cycles in the presentation of a sofic shift restricts the shift space to its periodic closure.

Lemma 3.4. *For a (finite) right-resolving vertex presentation G of Σ_G, we have $\Sigma_{\mathsf{cyc}(G)} = \overline{\mathcal{P}}(\Sigma_G)$.*

Proof. For $\boldsymbol{a} \in \mathcal{A}^{\mathbb{N}}$, we have $\boldsymbol{a} \in \overline{\mathcal{P}}(\Sigma_G)$ if and only if there is a sequence of periodic points $\{\boldsymbol{a}^{(i)}\}_{i\in\mathbb{N}} \subseteq \mathcal{P}(\Sigma_G)$ such that for all $k \in \mathbb{N}$ there exists $i \in \mathbb{N}$ such that $[\boldsymbol{a}]_k = [\boldsymbol{a}^{(i)}]_k$. Thus, we conclude $\mathcal{B}(\mathcal{P}(\Sigma_G)) = \mathcal{B}(\overline{\mathcal{P}}(\Sigma_G))$. Now, note that $w \in \mathcal{B}(\mathcal{P}(\Sigma_G))$ if and only if w can be extended to a cycle in G, which can happen if and only if $w \in \mathcal{B}(\Sigma_{\mathsf{cyc}(G)})$. As $\Sigma_{\mathsf{cyc}(G)}$ and $\overline{\mathcal{P}}(\Sigma_G)$ are both subshifts, we are done. □

Combining this lemma with Propositions 2.20 and 3.1 gives the following corollary showing the correctness of Algorithm 2.

Corollary 3.5. *Let (M, ℓ) be a labeled Conley index representative for f on isolated invariant set S, and let $\Sigma' := \Sigma_{G'}$ be the sofic shift presented by the output G' of Algorithm 2 on input (M, ℓ, τ) for some maximum iteration number τ. Then (Σ', σ) is topologically semi-conjugate to f on an invariant set $S' \subseteq S$.*

Proof. Since $\mathcal{P}(\Sigma') \subset \Sigma(M, \ell)$, $\mathrm{Con}(S_{\boldsymbol{b}}, f^{\boldsymbol{b}}) \neq [0]_s$ for each $\boldsymbol{b}^{\mathbb{N}} \in \mathcal{P}(\Sigma')$ (see discussion following Definition 2.18). By Proposition 2.20, $\Sigma' = \Sigma_{G'} = \Sigma_{\mathsf{cyc}(G')} = \overline{\mathcal{P}}(\Sigma')$ is topologically semi-conjugate to f. □

When Algorithm 2 terminates with an empty queue, the resulting subshift $\Sigma' = \overline{\mathcal{P}}(\Sigma(M, \ell))$ is the maximal subshift obtainable from (M, ℓ) via Proposition 2.20. That is, Σ' is the closure of *all* periodic symbol sequences corresponding to nontrivial Conley indices from (M, ℓ). Moreover, we can conclude that this maximal subshift is sofic.

3.3. Advantages over previous methods. The approach we have presented above offers two important advantages over the approach from [DFT08]. The first is that, by focusing on the broader class of sofic shifts rather than the subshifts of finite type produced in [DFT08], Algorithm 2 can extract strictly more information from the labeled Conley index. Intuitively, this follows as sofic shifts are more expressive than subshifts of finite type. Returning to the conceptual example in Figure 3.1, the algorithm of [DFT08] would discover that a transition from region N_2 to N_1 and back is not allowed in general, and determine that one of the edges $(1, 2)$ or $(2, 1)$ would need to but removed. After the $\mathsf{cyc}()$ operation, this would yield a pair of fixed points, i.e. the subshift $\Sigma = \{\overline{1}, \overline{2}\}$, which has zero entropy. Algorithm 2, on the other hand, produces the even shift, which has entropy roughly 0.4812.

To illustrate this difference in practice, consider the standard map, studied in [FT12] using the index processing methods of [DFT08]. One of the main results in [FT12], Theorem 4.4, concerns the index pair depicted in Figure 3.2, which has 41 regions. Using the [DFT08] approach for constructing a subshift of finite type from the computed index representative (M, ℓ), produces a vertex shift Σ_1 with 41 vertices and 55 edges, with topological entropy roughly 0.5451. Using Algorithm 2 on the same labeled Conley index representative, we obtain a semi-conjugate sofic shift Σ_2, which is not of finite type but contains Σ_1, implying that $\Sigma_1 \subsetneq \Sigma_2$. Moreover, the topological entropy of Σ_2 is significantly higher at 0.5715, implying that Σ_2 has exponentially more periodic orbits than Σ_1. In this example, as happens overwhelmingly in practice, Algorithm 1 terminated with an empty queue, meaning we have $\Sigma_2 = \overline{\mathcal{P}}(\Sigma(M, \ell))$, the maximal subshift obtainable from Proposition 2.20. Using minimization techniques described later in Section 4.3, we see that Σ_2 has an edge presentation with 45 vertices and 71 edges, only slightly larger in description size than the original vertex shift Σ_1.

Another advantage comes when considering generators on multiple levels of homology. Recall from the discussion following Definition 2.13, that $M = M(f_{P*}, \{g_i\}_{i=1}^n)$ is a block-diagonal matrix, with a blocks corresponding to each homology level. Thus, after running Algorithm 1, we would expect to see a union of disjoint graphs, where the states in each represent matrix images within different blocks. This is a major improvement over the algorithms in [DFT08]; consider for example the matrix $\left[\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right]$, where generators g_1, g_2 are on level 1, g_3, g_4 are on level 2, and there are two regions, N_1 containing g_1, g_3 and N_2 containing g_2, g_4. Using Algorithm 2, we would obtain a graph with three disjoint components, corresponding to the four points $(\overline{1}, \overline{2}, \overline{12}, \overline{21})$. In contrast, the methods of [DFT08] would be unable to express this system; one could have both $\overline{1}$ and $\overline{2}$ or both $\overline{12}$ and $\overline{21}$, but any larger and spurious connections would appear; for example, adding all four would give the full 2-shift. Note that even though the shift returned by Algorithm 2 is actually of finite type in this case, it is not a vertex shift, which is required by [DFT08].

The final advantage of our approach is speed. In practice, Algorithm 2 is considerably faster than the algorithm given in [DFT08], as one avoids the branch-and-bound search of an exponentially large solution space, namely the set of edges in the vertex shift to cut. The difference can be quite dramatic; for the large Hénon example given in Section 4.2, processing time dropped from hours to well under a minute on the same computer.

3.4. Extensions and generalizations. We used the column echelon form in Algorithm 1, as a unique way to represent matrix images and check nonnilpotency of the associated matrix products, but other invariants may be used. For example, we may wish to use the Lefschetz number, a fixed point invariant computable from $\mathrm{Con}(S, f^{\boldsymbol{b}})$ (see [DFT08]). The Lefschetz number is a stronger invariant and results in the additional property that periodic orbits in the constructed semiconjugate subshift have preimages under the itinerary function that are periodic orbits of the same period. We can modify Algorithm 1 to output a sofic system whose periodic orbits consist entirely of those with nontrivial Lefschetz numbers by simply replacing ech with another normalization function which preserves the Lefschetz numbers of the final matrix products. One such function which performs

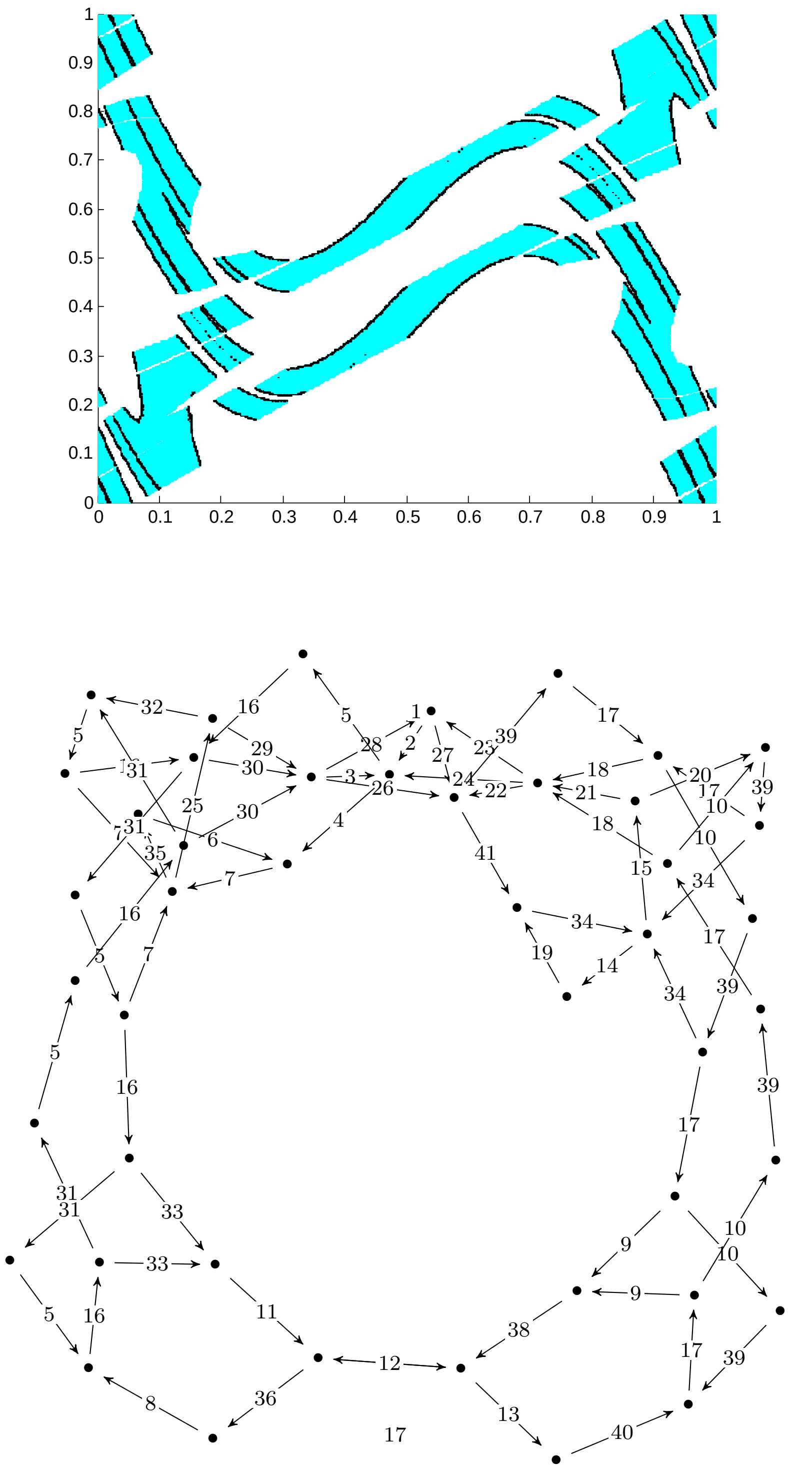


Fig. 3.2. *Index pair from [FT12], and the minimized right-resolving edge presentation of $\Sigma' = \Sigma(M, \ell)$. We can see that Σ' is not a subshift of finite type by looking at the symbol '12': we have $38(12)^k13 \in \mathcal{B}(\Sigma')$ when k is even but not when k is odd. Note that we use an edge instead of vertex presentation as a result of the minimization procedure in Section 4.3.*

well in practice is to divide the matrix by the first nonzero entry, an approach similar to [DFT08].

In terms of the shift space, we would be modifying the definition of a cocyclic shift to (additionally) disallow blocks of symbols which correspond to cycles with 0 Lefschetz number; that is, we disallow points containing a block $\boldsymbol{b} = a_1 a_2 \dots a_m$ if $\mathrm{Lef}(M^{\boldsymbol{b}}) = 0$. Replacing `ech` in Algorithm 1 with a normalization which preserves the Lefschetz number will generally create many more (symbol,matrix) states, but this is necessary to track the Lefschetz number, which is a stronger invariant. In general, any such invariant of interest could be tracked as long as `ech` is replaced by a normalization which preserves the invariant.

4. Applications. As mentioned in Section 2.5, computational Conley index methods have been successfully applied to a variety of systems, including the two-dimensional Hénon map, the three-dimensional LPA model, and the infinite-dimensional Kot-Schaffer equation. We now use combinatorial outer approximations for these systems (computed following techniques described in [DJM05, DFT08, DK13]) to compute and analyze Conley index information capturing increasingly complicated dynamics. This presents new challenges over what had been faced in previously studies. Most notably, systematic methods for both constructing isolating neighborhoods and processing the resulting labeled Conley indices become essential to computing increasingly complicated dynamics. In what follows, we first introduce the three models we use to illustrate the methods, then describe techniques for constructing isolating neighborhoods containing complicated dynamics and amalgamating symbols to simplify computed sofic shifts.

4.1. The models. The *Hénon map* is a map $h : \mathbb{R}^2 \to \mathbb{R}^2$ given by

$$h(x, y) = (1 + y - ax^2, bx) \ . \tag{4.1}$$

with the classical parameter values $a = 1.4$, $b = 0.3$. This map has become the benchmark system for measurements of complicated dynamics, see e.g. [NBGM08, Gal02, DJM05, DFT08]. Continuing the study described in [DFT08], we now briefly describe several updated techniques for growing isolating neighborhoods, and illustrate them together with Algorithm 2 on the Hénon map.

The *Larvae-Pupae-Adult (LPA) map*, describes the evolution of a population of flour beetles with three developmental stage classes. The LPA map is given as $T : \mathbb{R}^3 \to \mathbb{R}^3$,

$$T(x, y, z) = ((f_1 x + f_2 y + f_3 z) \cdot e^{-\lambda(x+y+z)}, p_1 x, p_2 y) \tag{4.2}$$

where f_i is the *per capita* fertility of stage class i at small population sizes, λ is an additional fertility parameter incorporating nonlinear effects at larger population sizes, and $0 \leq p_i \leq 1$ is the probability that an individual in stage-class i survives and enters the next stage in one time step (one application of the map T). Following [UW04], we set $f = f_1 = f_2 = f_3 = 37.5$, the fertility parameter to be $\lambda = 0.1$ and the survival and transition probabilities to be $p_1 = 0.8$ and $p_2 = 0.6$. In what follows, we study (4.2) in the domain $[0, 400]^3$.[3] In addition to having a higher dimensional phase space, the LPA model also exhibits slower recurrence times than the Hénon map. The sample results presented for this model in Section 4.2 give what we believe to be the highest known lower bound on topological entropy for this model.

The *Kot-Schaffer integrodifference operator* is a spatially explicit discrete dynamical system that was introduced in [KS86] to describe populations with distinct growth and dispersal phases. One

[3] To compute the outer approximation, we used a simple refinement of interval arithmetic, wherein we subdivide each box uniformly into 27 smaller boxes (3 slices in each dimension), compute the interval images of each smaller box using interval arithmetic, and then taking the union as the image of the box.

example is a population of plants that have a growth phase, producing seeds, and then the seeds are dispersed during a distinct dispersal phase, giving rise to the next population of plants. In general form, the Kot-Schaffer model consists of an integrodifference operator $\Phi : L^2([-\pi, \pi]) \to L^2([-\pi, \pi])$ of the form

$$\Phi[a](y) := \frac{1}{2\pi} \int_{-\pi}^{\pi} b(x, y) G[a](x) dx, \tag{4.3}$$

with smooth dispersal kernel $b(x, y) = b(x - y)$ and growth operator G. In [DK13], Day and Kalies describe an approach using Chebyshev interpolants, Galerkin projection, and error analysis to construct an outer approximation for (4.3) with G an L^∞-bounded operator that is well-approximated by a polynomial Nemytskii operator with computable error bounds. The example from [DK13] which we now study here uses the Ricker growth operator $G[a](x) := \mu \ a(x)e^{-a(x)}$ where μ is a fitness parameter and $c \in C^2([-\pi, \pi])$, a nonnegative function with $\|c\|_\infty = 1$, models heterogeneous variation in the fitness of the environment. For the sample results, we set $\mu = 30$ and the Fourier expansions of b and c to be $\hat{b} = [1, 1 - \Sigma_{n=2}^{\infty} \lambda^n, \lambda^2, \lambda^3, \ldots]$ where $\lambda = 1/15$ and $\hat{c} = [0.5, 0.25, 0, 0, \ldots]$ respectively. The outer approximation for this infinite-dimensional map is constructed in Fourier space with subdivision only in a finite number of modes, giving a grid of the form $Z = R \times V = \prod_{n=0}^{M-1} \frac{A_{s_n}}{n^{s_n}}[-1, 1] \times \prod_{n \geq M} \frac{A_{s_*}}{n^{s_*}}[-1, 1]$, where $s_* = 10$ and s_n, A_{s_n}, A_{s*} are all positive constants. Modes $n \geq M = 14$ are handled analytically while modes $n < M$ are tracked explicitly using interval arithmetic. Nontrivial subdivision, that is subdividing rather than just updating the bound on a particular mode, is carried out in the first 6 modes. For more detail on the combinatorial outer approximation for this model, see [DK13].

4.2. Preprocessing: isolating neighborhood construction. We now present methods for computing isolating neighborhoods containing complicated dynamics. This approach is a refinement of one given in [DFT08] and utilizes isolating neighborhoods of periodic orbits as a base for the construction. Motivated by the idea that periodic orbits, and especially low period periodic orbits, may be used to approximate mixing on an attractor and topological entropy, we begin by computing isolating neighborhoods for cycles in a directed graph representation for the combinatorial outer approximation. Taking the union of some, or all, of these isolating neighborhoods yields a larger set whose isolating neighborhood often contains not only the periodic orbits corresponding to the originally computed cycles, but also other orbits that weave in and out of these.

Generally speaking, if we view the combinatorial outer approximation as a directed graph, we expect a period-k orbit to correspond to a length-k cycle in the graph. Thus, to locate candidate regions for periodic orbits, we can examine the cycles in the combinatorial outer approximation, as has been done in previous work (see e.g. [DFT08] and references therein). Here, we introduce a slightly more sophisticated approach, where we instead calculate the *first return time* of each grid element B, which is the length of the shortest cycle containing B. First return times have several advantages, including efficiency and the lack of duplicates among different periods. Note that a grid element with first return time k does not necessarily contain any k-periodic points, but by the nature of the combinatorial outer approximation, we are guaranteed to have captured all k-periodic points in boxes whose first return times are k or factors of k (e.g. a period-7 orbit may fit entirely within a single grid element). The computation is done using a simple breadth-first search (BFS), and is given as Algorithm B.1.

Given periodic orbit candidates of period k, we use the Conley index to *verify* them. Specifically, we take a candidate grid element, compute a cycle within the candidate set in the graph of length k, grow an isolating neighborhood for these k boxes, and check that the Conley index is consistent with a period k orbit. If successful, we add the isolating neighborhood to a list of verified orbits. We then remove the neighborhood from the candidate boxes, and proceed until all boxes are processed. (See Algorithm B.2.)

To build more complicated dynamics, we take the union of the neighborhoods of a subset of the verified (usually low-period) orbits, and use the result to grow our final neighborhood. For example, one may grow a neighborhood from all periodic orbits up to period 8. This approach often produces more dynamics than the sum of its parts; in addition to the union of the periodic orbits, disjoint regions may grow together (or "glue"), and the larger isolating neighborhood may include additional trajectories not seen by restricting to any of the smaller isolating neighborhood. This leads to additional edges in the resulting sofic shift. As an illustration, Figure 2.4 depicts an index pair from the Hénon map at depth 7, which is grown from the isolating neighborhoods of a period-2 and a period-4 orbit. One can see that the resulting semiconjugate subshift is the "gluing" of these two orbits together into a simple horseshoe. Similarly, the index pair in Figure 2.5, from Hénon at depth 8, is the union of neighborhoods for period-4, period-6, and period-7 orbits; these orbits can be read off easily from the sofic shift generated by Algorithm 2, shown in Figure A.1.

Taking this approach to the extreme gives the following example, again on the Hénon map but at depth 14 instead of 8, which gives us a 64-fold increase in resolution. Here we verify 700 orbits with periods less than 19, and using a simple randomized binary search algorithm (see Algorithm B.4), select 350 of them to grow an index pair. The resulting isolating neighborhood has 185030 boxes (with 3193 additional exit set boxes), consisting of 342 disjoint regions (Figure 4.1). Computing the Conley index yields a labeled index representative with 2062 generators of homology, and applying Algorithm 2 to this representative yields a semiconjugate sofic shift with a right-resolving vertex presentation with 388 vertices and 586 edges (Figure 4.2(L)). As a subroutine, Algorithm 1 terminates with an empty queue in 395 iterations,[4] which by Proposition 3.1 implies that the maximal subshift $\overline{\mathcal{P}}(\Sigma(M,\ell))$ is sofic in this case. The entire run of Algorithm 2 took under 5 seconds on a commodity laptop.

To measure the complexity of the dynamics we capture on the Hénon attractor, we can examine the topological entropy of the computed sofic shift. As mentioned in Section 2, the topological entropy of a sofic subshift produced by Algorithm 2 is readily attainable as the log of the spectral radius of its returned vertex presentation. Since the sofic subshift is semi-conjugate to the system that generated the input labeled Conley index, this computed topological entropy serves as a lower bound for the original system. Putting all of the above together yields the following theorem.

Theorem 4.1. *The topological entropy of the Hénon map is at least* 0.4555.

The bound we achieve here is considerably higher than the Conley index and SFT-based bound of 0.4320 computed in [DFT08]. This improvement is due to our more principled method for building the index pair, i.e. as a union of verified periodic orbits rather than computing pairwise connections in the outer approximation, as well as the transition from SFTs to sofic shifts made possible by Algorithm 2. Interestingly, the improvement is certainly not from advances in computational power, as the computations were performed on the same commodity laptop as the previous study [DFT08]. Increasing the depth from 14 to 16 or 17 should yield a higher entropy bound.

Other techniques may also be used to compute symbolic dynamics for Hénon and similar maps. One leading approach based on *trellises* applies to diffeomorphisms and requires careful calculations of finer grain information about the system. Pieter Collins first defined trellises in [Col04, Col05] and

[4] The reader may have noted that when Algorithm 1 terminates with an empty queue, the number of iterations must equal the number of vertices, in this case 395. The discrepancy between 395 and 388 is due to 7 vertices being orphaned during the periodic closure step, and thus removed.

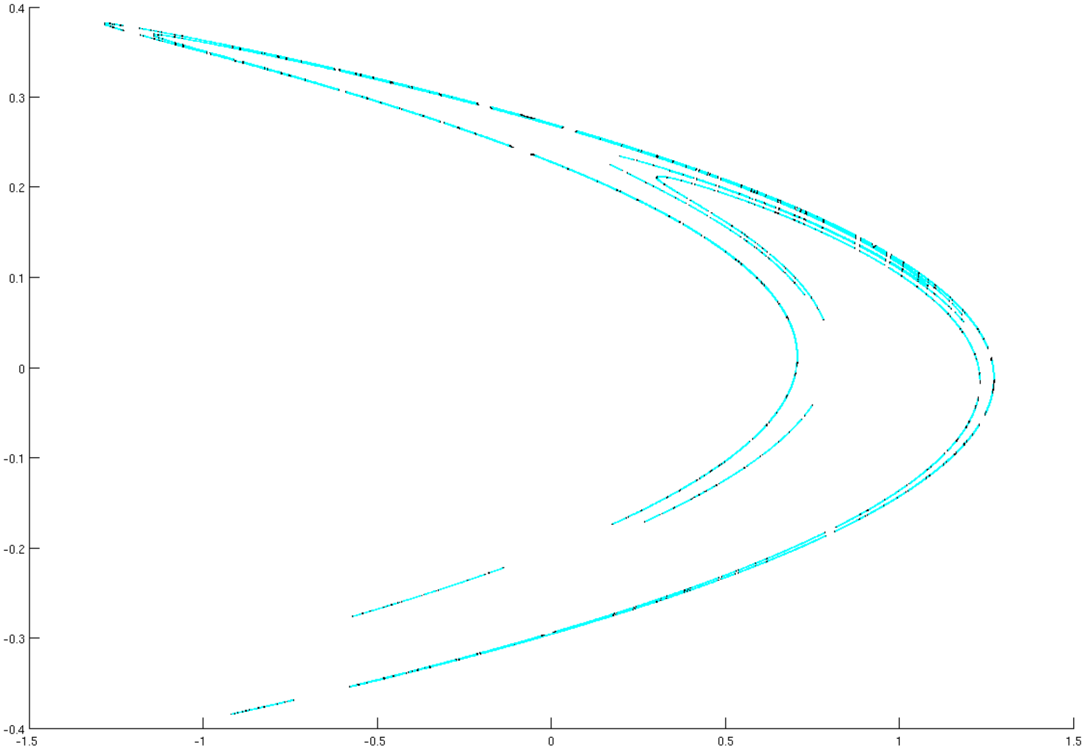


FIG. 4.1. *The index pair for the Hénon map at depth 14 used in Theorem 4.1. The entropy of the resulting sofic subshift is approximately 0.45558.*

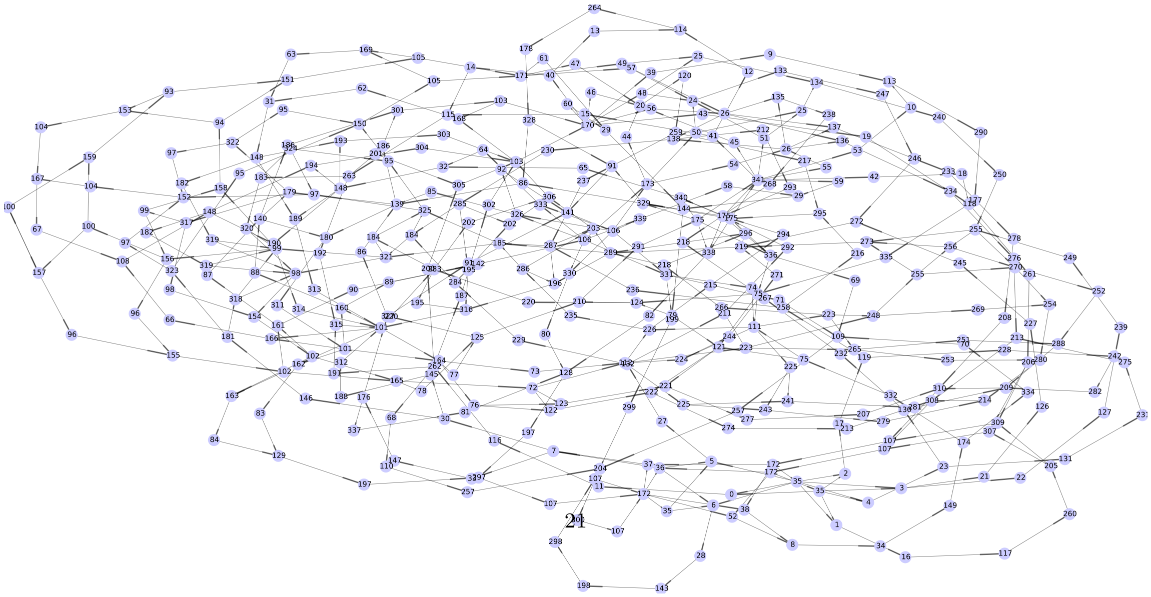


FIG. 4.2. *The right-resolving vertex presentation of the sofic shift produced for Theorem 4.1, with 388 vertices and 586 edges. The thick segment at the end of an edge denotes its direction. See Figure A.3 for the significantly smaller minimized edge presentation of this vertex presentation.*

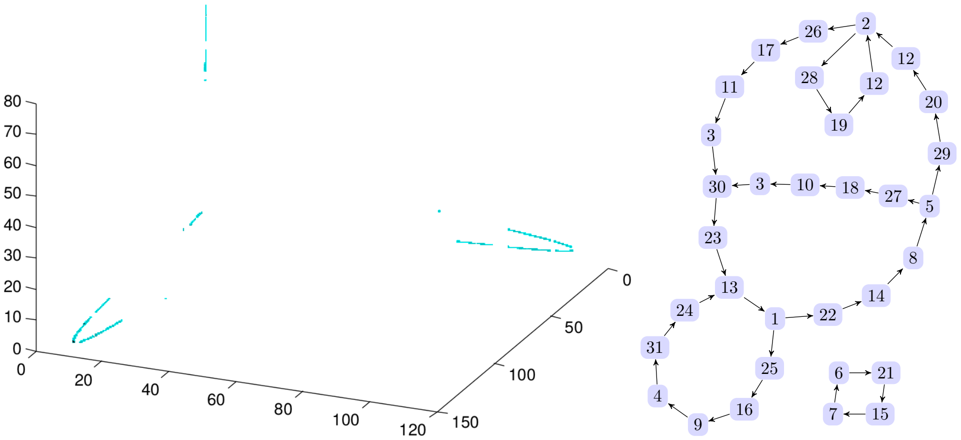


FIG. 4.3. *(L) The index pair for the LPA map* (4.2) *at depth 14 used in Theorem 4.2. (R) The resulting vertex shift presentation of the sofic shift returned by Algorithm 2, the labels for which correspond to a labeling of the 31 disjoint regions of the isolating neighborhood.*

their construction uses information about pieces of stable and unstable manifolds to force dynamics. Following this construction, Newhouse et al. [NBGM08] developed an algorithm to calculate regions bounded by pieces of the one-dimensional stable and unstable manifolds for the hyperbolic fixed point on the Hénon attractor as well as how these regions map across one another. The resulting symbolic dynamics yields the best known lower bound for the Hénon system of 0.46469. Methods of this type rely on careful calculations of stable and unstable manifolds, the regions they bound, and how these regions map across each other. By comparison, our approach requires only construction of an outer approximation (i.e. the ability to compute reasonable outer bounds on the images of cubical grid elements) and a coarse level of hyperbolicity necessary to produce a nontrivial Conley index. It would be interesting to study the use of adaptive grids in the outer approximation construction and whether such an approach, motivated by the idea of approximating the regions used in trellis calculations, could be used to further optimize the methods presented here.

To emphasize that the approach presented here is not restricted to planar diffeomorphisms or maps for which we can compute stable and unstable manifolds, we now apply our techniques to the LPA model and, in Section 4.3, the Kot-Schaffer model. The next sample result gives, to our knowledge, the first nonzero lower bound on topological entropy for the LPA map. We begin by gathering all periodic orbits up to period 28 at depth 14 and grow an index pair whose isolating neighborhood has 60974 boxes (with 702 additional exit set boxes), consisting of 31 disjoint regions. The labeled index representative has 36 generators of homology. Running Algorithm 2 on this labeled representative, we compute the semi-conjugate sofic subshift with vertex presentation shown in Figure 4.3. Finally, computing the topological entropy of this vertex presentation yields the following theorem.

THEOREM 4.2. *The topological entropy of the LPA map* (4.2) *is at least* 0.1203.

4.3. Postprocessing: Minimization and Amalgamation. As one can see from the sample results in Section 4.2, the sofic shifts produced by Algorithm 2 can be quite complicated, with hundreds of symbols, states, and transitions. While some of this complexity is of course inherent in the dynamics, some of it may be unnecessary, arising from the particular discretization of the phase

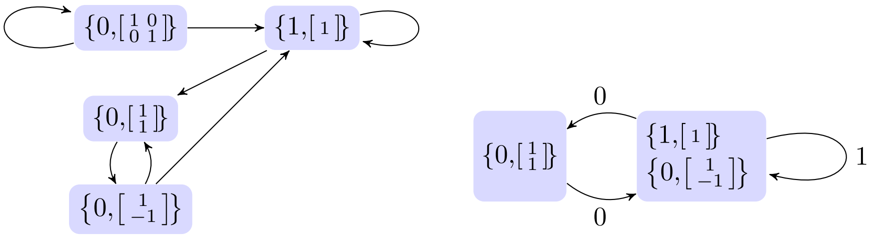

$\{0,\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]\}$
$\{1,[1]\}$
$\{0,\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\}$
$\{0,\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]\}$
0
$\{0,\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\}$
$\{1,[1]\}$
$\{0,\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]\}$
1
0

FIG. 4.4. *The minimization of the final vertex presentation from Figure 3.1, yielding the standard edge presentation of the even shift.*

space. A natural question is thus whether we can find a more compact representation, which is only as complex as required to describe the dynamics. Here we discuss two techniques to reduce the size of the representation while preserving the underlying dynamics. The first technique computes the minimal right-resolving edge presentation for a sofic shift, which combines redundant states/nodes together while exactly preserving the original subshift. The second modifies the shift by combining *symbols* in the alphabet, via a process called *amalgamation*, in a way which preserves the dynamics up to conjugacy.

It is well-known that sofic shifts are closely related to deterministic finite automata (DFA) from information theory and computer science: both have a finite state graph with transitions labeled by symbols, and both have sets of blocks associated with them, the *language* of a DFA and the allowed blocks of a sofic shift. The main difference between the two is that DFAs have explicit start, accept, and reject states, but allow transitions on any symbol, whereas sofic shifts have only accept states, and disallow certain transitions; sofic shifts can thus be simulated by a DFA by filling in missing transitions and directing them to a new reject state. The DFA minimization problem, that of finding a DFA accepting the same language but with the fewest possible states, is well-studied, and enjoys fast algorithms for this task such as Hopcroft's algorithm [Hop71]. Given an *irreducible* sofic shift,[5] a property guaranteed in Algorithm 2 by the $\mathsf{cyc}(\cdot)$ operation,[6] Hopcroft's algorithm will find its unique minimal right-resolving presentation [BKM85, Jon96]. The runtime remains efficient at $O(s \cdot |\mathcal{A}| \log s)$, where s is the number of states in the original right-resolving presentation. Figure 4.4 shows the minimization of the example in Figure 3.1, which results in the classic form of the even shift; one can easily verify that the two graphs represent the same sofic shift. Applying the minimization technique to the sample Hénon result in Theorem 4.1 yields an edge presentation with only 251 vertices and 383 edges (Figure A.3(R)), a reduction in size of roughly 35%.

In addition to reducing the number of states in a sofic shift presentation, one could also try to reduce the number of symbols. The process of *amalgamating* symbols a and b to a new symbol $[ab]$ corresponds to taking regions N_a and N_b and replacing them with a single region $N_{[ab]} = N_a \cup N_b$. As a side effect, one loses some information in the resulting shift with regard to itineraries of trajectories (for a point containing symbol $[ab]$, it is no longer clear which of N_a or N_b it passes through), but

[5] A subshift Σ is irreducible if for every $u, v \in \mathcal{B}(\Sigma)$ there is some $w \in \mathcal{B}(\Sigma)$ such that $uwv \in \mathcal{B}(\Sigma)$. In the case of sofic shifts, irreduciblility is equivalent to the existence of an irreducible (strongly-connected) presentation.

[6] Technically, $\mathsf{cyc}(\cdot)$ may return a disconnected graph, each component of which presents an irreducible sofic shift. In practice, we may simply minimize each presentation individually, or choose the component with the highest entropy, depending on the application.

of course this information remains in the original presentation. Not all symbol amalgamations yield a conjugate subshift, and the problem of determining how many amalgamations are possible while preserving conjugacy is a challenging one. In fact, this problem is NP-hard (computationally intractable for large instances) even for SFTs [Fro17], so in practice we simply perform a brute-force search with a few heuristics to find a smaller conjugate subshift (cf. [Fro14, Algorithm 1]). Amalgamations can significantly reduce the number of symbols needed to express a subshift, but are especially useful when trying to compare the produced subshift to another, as in [Fro14, Theorem 5.2].

For our final example, we show a sample result for the Kot-Schaffer model that uses symbol amalgamation to present a simplified, conjugate shift to one produced by Algorithm 2. The isolating neighborhood of the index pair from [DK13] had 61167 boxes (with 1031 additional exit set boxes) in 16 disjoint regions; see Figure 4.5(L). The vertex presentation of the sofic shift resulting from Algorithm 2, which happens to be an SFT in this case, is shown in Figure 4.5(R) along with an overlaid amalgamation down to just 4 symbols $\{a, b, c, d\}$. As the shifts are conjugate, both yield the same rigorous lower bound on the topological entropy of 0.2406.

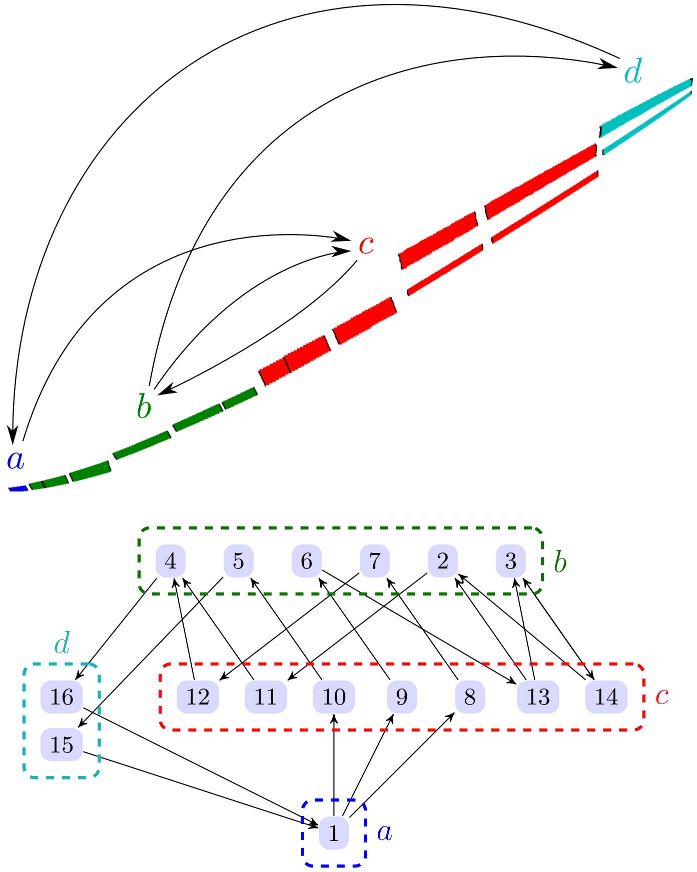


Fig. 4.5. *The index pair from [DK13] (top) along with the resulting sofic shift computed from Algorithm 2 (bottom). The amalgamation of the symbols of the sofic shift is shown with dashed lines surrounding the four final symbols, which correspond to the larger regions above labeled a, b, c, d.*

REFERENCES

[AKK$^+$09] Zin Arai, William Kalies, Hiroshi Kokubu, Konstantin Mischaikow, Hiroe Oka, and PawełPilarczyk, *A database schema for the analysis of global dynamics of multiparameter systems*, SIAM J. Appl. Dyn. Syst. **8** (2009), no. 3, 757–789. MR 2533624

[BKM85] Mike Boyle, Bruce Kitchens, and Brian Marcus, *A note on minimal covers for sofic systems*, Proceedings of the American Mathematical Society (1985), 403–411.

[Bow71] Rufus Bowen, *Periodic points and measures for Axiom A diffeomorphisms*, Trans. Amer. Math. Soc. **154** (1971), 377–397. MR MR0282372 (43 #8084)

[Boy00] Mike Boyle, *Algebraic aspects of symbolic dynamics*, Topics in symbolic dynamics and applications (Temuco, 1997), London Math. Soc. Lecture Note Ser **279** (2000), 57–88.

[Col04] Pieter Collins, *Dynamics of surface diffeomorphisms relative to homoclinic and heteroclinic orbits*, Dyn. Syst. **19** (2004), no. 1, 1–39. MR 2038270

[Col05] ———, *Entropy-minimizing models of surface diffeomorphisms relative to homoclinic and heteroclinic orbits*, Dyn. Syst. **20** (2005), no. 4, 369–400. MR 2182474

[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)

[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)

[DFT08] Sarah Day, Rafael Frongillo, and Rodrigo Treviño, *Algorithms for rigorous entropy bounds and symbolic dynamics*, SIAM J. Appl. Dyn. Syst. **7** (2008), no. 4, 1477–1506. MR 2470974 (2010b:37023)

[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

[DJM05] ———, *Towards automated chaos verification.*, Proceeding of Equadiff 2003 (2005), 157–162.

[DK13] Sarah Day and William D. Kalies, *Rigorous computation of the global dynamics of integrodifference equations with smooth nonlinearities*, SIAM J. Numer. Anal. **51** (2013), no. 6, 2957–2983. MR 3124898

[Fro14] Rafael Frongillo, *Topological Entropy Bounds for Hyperbolic Plateaus of the Henon Map*, SIAM Undergraduate Research Online **7** (2014), 142–161.

[Fro17] ———, *Optimal state amalgamation is NP-hard*, Ergodic Theory and Dynamical Systems (2017), 1–13.

[FT12] Rafael Frongillo and Rodrigo Trevino, *Efficient automation of index pairs in computational conley index theory*, SIAM Journal on Applied Dynamical Systems **11** (2012), no. 1, 82–109.

[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.

[Hop71] John Hopcroft, *An n log n algorithm for minimizing states in a finite automaton*, Theory of machines and computations, Elsevier, 1971, pp. 189–196.

[Jon96] Nataa Jonoska, *Sofic shifts with synchronizing presentations*, Theoretical Computer Science **158** (1996), no. 1, 81–115.

[KMM04] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek, *Computational homology*, Applied Mathematical Sciences, vol. 157, Springer-Verlag, New York, 2004. MR 2028588

[KMV14] William D. Kalies, Konstantin Mischaikow, and Robert C. A. M. Vandervorst, *Lattice structures for attractors I*, J. Comput. Dyn. **1** (2014), no. 2, 307–338. MR 3415257

[KMV16] ———, *Lattice structures for attractors II*, Found. Comput. Math. **16** (2016), no. 5, 1151–1191. MR 3552843

[KS86] Mark Kot and William M. Schaffer, *Discrete-time growth-dispersal models*, Math. Biosci. **80** (1986), no. 1, 109–136. MR 88a:92030

[Kwa00] Jaroslaw Kwapisz, *Cocyclic subshifts*, Mathematische Zeitschrift **234** (2000), no. 2, 255–290.

[Kwa04] ———, *Transfer operator, topological entropy and maximal measure for cocyclic subshifts*, Ergodic Theory and Dynamical Systems **24** (2004), no. 4, 1173–1197.

[LM95] Douglas Lind and Brian Marcus, *An introduction to symbolic dynamics and coding*, Cambridge University Press, Cambridge, 1995. MR 1369092 (97a:58050)

[MM95] Konstantin Mischaikow and Marian Mrozek, *Chaos in the Lorenz equations: a computer-assisted proof*, Bull. Amer. Math. Soc. (N.S.) **32** (1995), no. 1, 66–72. MR 1276767

[MM98] ———, *Chaos in the Lorenz equations: a computer assisted proof. II. Details*, Math. Comp. **67** (1998), no. 223, 1023–1046. MR 1459392

[MM02] ———, *Conley index*, Handbook of dynamical systems, Vol. 2, North-Holland, Amsterdam, 2002, pp. 393–460. MR MR1901060 (2003g:37022)

[MMS01] Konstantin Mischaikow, Marian Mrozek, and Andrzej Szymczak, *Chaos in the Lorenz equations: a computer assisted proof. III. Classical parameter values*, J. Differential Equations **169** (2001), no. 1, 17–56, Special issue in celebration of Jack K. Hale's 70th birthday, Part 3 (Atlanta, GA/Lisbon, 1998). MR 1808460

[NBGM08] S. Newhouse, M. Berz, J. Grote, and K. Makino, *On the estimation of topological entropy on surfaces*, Discrete and Continuous Math **469** (2008), 243–270.

[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**sp $*$ (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)

[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)

[UW04] Ilie Ugarcovici and Howard Weiss, *Chaotic dynamics of a nonlinear density dependent population model*, Nonlinearity **17** (2004), no. 5, 1689–1711. MR 2086145

Appendix A. Additional Examples.

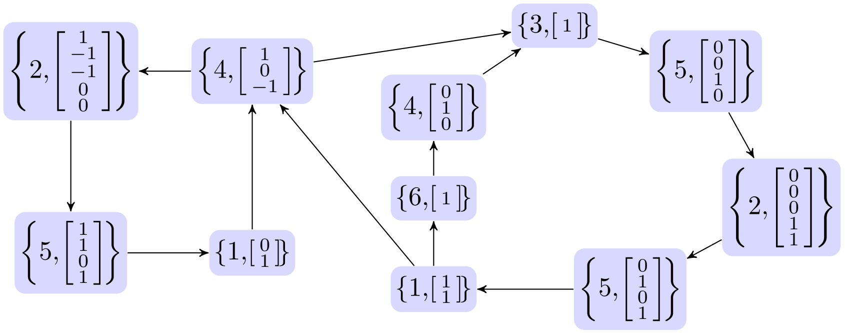

$\left\{2,\begin{bmatrix}1\\-1\\-1\\0\\0\end{bmatrix}\right\}$
$\left\{4,\begin{bmatrix}1\\0\\-1\end{bmatrix}\right\}$
$\{3,[\,1\,]\}$
$\left\{5,\begin{bmatrix}0\\0\\1\\0\end{bmatrix}\right\}$
$\left\{4,\begin{bmatrix}0\\1\\0\end{bmatrix}\right\}$
$\{6,[\,1\,]\}$
$\left\{2,\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix}\right\}$
$\left\{5,\begin{bmatrix}1\\1\\0\\1\end{bmatrix}\right\}$
$\{1,[\begin{smallmatrix}0\\1\end{smallmatrix}]\}$
$\{1,[\begin{smallmatrix}1\\1\end{smallmatrix}]\}$
$\left\{5,\begin{bmatrix}0\\1\\0\\1\end{bmatrix}\right\}$

Fig. A.1. *The vertex presentation of the sofic shift given by applying Algorithm 2 to the labeled Conley index representative from Figure 2.5. Each node is a pair* $\{$*symbol, matrix image space*$\}$*, with label equal to the symbol. One can clearly identify the original periodic orbits in the sofic shift, corresponding to points* $\overline{4251}$, $\overline{435251}$, *and* $\overline{4352516}$, *respectively.*

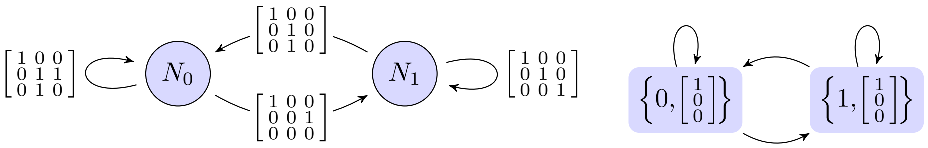

$\begin{bmatrix}1&0&0\\0&1&1\\0&1&0\end{bmatrix}$
N_0
$\begin{bmatrix}1&0&0\\0&1&0\\0&1&0\end{bmatrix}$
$\begin{bmatrix}1&0&0\\0&0&1\\0&0&0\end{bmatrix}$
N_1
$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$
$\left\{0,\begin{bmatrix}1\\0\\0\end{bmatrix}\right\}$
$\left\{1,\begin{bmatrix}1\\0\\0\end{bmatrix}\right\}$

Fig. A.2. *A simple example where* $G^{(\infty)}$ *from Algorithm 1 contains countably infinitely many nodes, yet starting from smaller image spaces (right) presents* $\Sigma(M,\ell)$*, here the full 2-shift, in just two nodes. Specifically,* $G^{(\infty)}$ *contains nodes of the form* $\{0,\begin{bmatrix}1&0&0\\0&F_{n+1}&F_n\\0&F_n&F_{n-1}\end{bmatrix}\}$ *for all* $n \geq 1$*, where* F_n *is the nth Fibonacci number.*

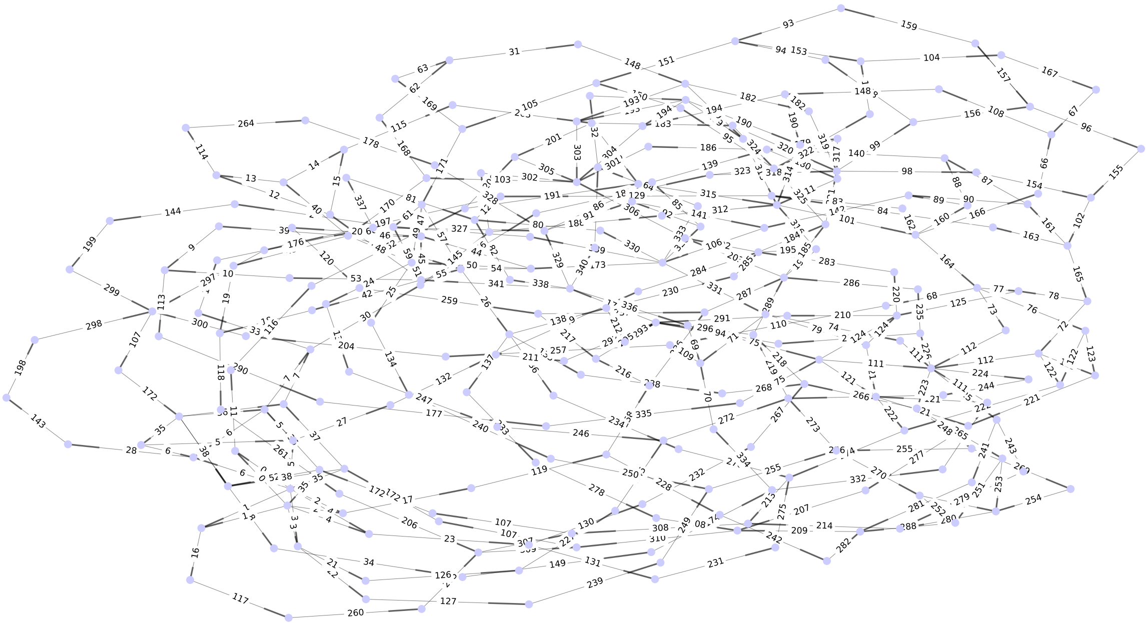


FIG. A.3. *The edge presentation resulting in minimizing the vertex presentation in Figure 4.2 for Theorem 4.1. The thick segment at the end of an edge denotes its direction. This minimized presentation has 251 vertices and 383 edges, significantly fewer than the 388 vertices and 586 edges of the vertex presentation.*

Appendix B. Omitted Algorithms.

Algorithm B.1 Compute the first return times of all nodes in a directed graph.

Input: Directed graph G

Output: An array C of sets, with $C[k] \subseteq V(G)$ being the subset of vertices of G with first return time k

# We will invoke this routine with G being the enclosure $\mathcal{F}$ with vertices $V(G) = \mathcal{G}$, the grid elements, and edges $E(G) = \{(g_1, g_2) \in \mathcal{G} \times \mathcal{G} \mid g_2 \in \mathcal{F}(g_1)\}$.

```
procedure FirstReturnTimes($G$)
    $C \leftarrow \emptyset$
    for vertices $v$ in $V(H)$ do
        $H' \leftarrow H$ with vertex $v$ copied to $v'$, with the same neighborhood
        $D \leftarrow \mathrm{BFS}(H', v)$          # Discover times of breadth-first search starting from $v$
        $k \leftarrow D[v']$                        # First return time of vertex $v$
        $C[k] \leftarrow C[k] \cup \{v\}$
    return $C$
```

Algorithm B.2 Verify periodic orbits from a collection of candidate boxes.

Input: Enclosure $\mathcal{F}$, grid adjacency graph A, first return time array C

Output: An array P of isolating neighborhoods of verified periodic orbits

```
procedure VerifyPeriodicOrbits($\mathcal{F}$, $A$, $C$)
    $P \leftarrow \emptyset$
    for periods $k = 1, \dots,$ length($C$) do
        while $C[k] \neq \emptyset$ do
            $v \leftarrow$ an arbitrary element of $C[k]$
            $S \leftarrow$ FindCycle($v$, $\mathcal{F}$, $k$)          # A length-$k$ cycle in $\mathcal{F}$ through $v$
            if $S = \emptyset$ then
                $C[k] \leftarrow C[k] \setminus \{v\}$
            else
                $\mathcal{N} \leftarrow$ GrowIsolatingNeighborhood($S$, $\mathcal{F}$, $A$)
                $(M, \ell) \leftarrow$ ConleyIndexRepresentative($\mathcal{N}$, $\mathcal{F}$, $A$)
                $G \leftarrow$ SemiConjugateSoficShift($M$, $\ell$, 10000)
                if $G$ presents the subshift of a $k$-cycle then
                    $P \leftarrow P \cup \{\mathcal{N}\}$          # In practice, we also record the period $k$
                    $C[k] \leftarrow C[k] \setminus \mathcal{N}$
    return $P$
```

Algorithm B.3 Simple union approach to growing isolating neighborhoods from verified periodic orbits.

Input: Enclosure $\mathcal{F}$, grid adjacency graph A, array P of verified periodic orbits

Output: Vertex presentation G of a semiconjugate sofic shift

```
procedure SimpleUnionApproach($\mathcal{F}$, $A$, $P$)
    $\mathcal{S} \leftarrow \cup_{S \in P} S$                                  # Union of all grid elements in $P$
    $\mathcal{N} \leftarrow$ GrowIsolatingNeighborhood($\mathcal{S}$, $\mathcal{F}$, $A$)
    $(M, \ell) \leftarrow$ ConleyIndexRepresentative($\mathcal{N}$, $\mathcal{F}$, $A$)
    $G \leftarrow$ SemiConjugateSoficShift($M$, $\ell$, 10000)
    return $G$
```

Algorithm B.4 Heuristic to find a subset of verified orbits of high entropy.

Input: Enclosure $\mathcal{F}$, grid adjacency graph A, array P of verified periodic orbits

Output: Vertex presentation $G_{\max}$ of a semiconjugate sofic shift (of high entropy)

```
procedure FindTippingPoint($\mathcal{F}$, $A$, $P$)
    $(k_{\text{lo}}, k_{\text{hi}}) \leftarrow (1, |P|)$
    $(h_{\max}, k_{\max}, G_{\max}) \leftarrow (0, 1, \emptyset)$
    while $k_{\text{lo}} \leq k_{\text{hi}}$ do
        $k \leftarrow \lfloor (k_{\text{lo}} + k_{\text{hi}})/2 \rfloor$
        $G \leftarrow$ SimpleUnionApproach($\mathcal{F}$, $A$, $P[1..k]$)         # First $k$ periodic orbits
        $h \leftarrow$ ComputeEntropy($\Sigma_G$)                # Log of the spectral radius of
                                                            the adjacency matrix of $G$
        if $h > h_{\max}$ then
            $(h_{\max}, k_{\max}, G_{\max}) \leftarrow (h, k, G)$
        if $k > k_{\max}$ and $h < h_{\max}$ then
            $k_{\text{hi}} \leftarrow k - 1$
        else
            $k_{\text{lo}} \leftarrow k + 1$
    return $G_{\max}$
```