

Towards characterisation of chaotic
attractors in terms of embedded coherent
structures.

Thesis submitted for the degree of

Doctor of Philosophy

at the University of Leicester

by

Daniel Crane

Department of Mathematics

University of Leicester

June 2017

Abstract

The central theme of this thesis is the development of general methods for the modelling of the dynamics on chaotic attractors by a coarse-grained representation constructed through the use of embedded periodic orbits & other coherent structures. Our aim is to develop tools for constructing two types of reduced representations of chaotic attractors: Markov-type models, and symbolic dynamics. For Markov models, we present construction of a minimal cover of chaotic attractors of maps and high-dimensional flows by embedded coherent structures such as periodic orbits from which a Markov chain of the dynamics can be constructed. For the symbolic dynamics, we investigate the utility of unstable periodic orbits for the construction of an approximate generating partition of a chaotic attractor.

In the first section of Part 1 we present an original method by which chaotic attractors of discrete-time dynamical systems can be covered using a small set of unstable periodic orbits (UPOs) following an iterative selection algorithm that only chooses those UPOs that provide additional covering of the attractor to be included into the cover. We then show how this representation can be used to represent trajectories in the system as a series of transition between cover elements, using which as a basis for the construction of a Markov chain representation of the dynamics. In the second section we extend this method to continuous-time dynamical systems, introducing methods by which covers of high-dimensional attractors can be constructed in low dimensional projections with as little information loss as possible, and also giving an example of how group symmetries of the system can be dealt with.

In Part 2 we change our focus to the construction of symbolic dynamics of discrete-time systems, presenting an extension to an existing method for the computational construction of approximate generating partitions that increases the applicability of the method to a wider range of systems, and also significantly improving the results for more complex maps.

Acknowledgements

Over an enlightening 8 years at the University of Leicester I've accrued many debts.

My supervisor Prof. Ruslan Davidchack has been unequivocally supportive over the past 5 years, offering invaluable advice about both academic and personal issues; without him, I have no doubt that the completion of this PhD wouldn't have been possible.

I'd also like to thank Dr. Ivan Tyukin who mentored me for several years during my undergraduate studies, and who continued to support me throughout my doctoral studies; the internship opportunity he provided me set me down this path, and so without him I would undoubtedly not be writing this today.

Prof. Alexander Gorban has been an incredible source of inspiration during my PhD project. During the first year his advice solidified the foundations of the project, with his continued support has keeping the project on track.

Additionally, I'd like to extend my heartfelt thanks to all members of the Department of Mathematics at the University of Leicester; especially the administrative staff, including Dan Carter, Ushma Chauhan, & Melissa Goodwin.

I would also like to express my sincere gratitude to my family members for their continued support over the past 8 years without whom the completion of my PhD wouldn't have been possible. I'd like to extend specific thanks to my parents Adam & Sue, my brother James, and my uncle and his partner Matthew & Gemma.

Last but not least, I'd like to extend my thanks to my partner Xiaoying Ren whose patience and support helped me considerably over the past several years. I look forward to starting the next chapter of my life with her upon completion of my doctoral studies.

Contents

Introduction	5
Periodic Orbit Theory	5
Computational Method for Detection of Unstable Periodic Orbits	6
1 Cover of 2-dimensional chaotic maps by embedded periodic orbits	8
1.1 Hausdorff Distance	9
1.2 Constructing the Cover	10
1.3 Application to Ikeda Map	12
1.4 Shadowing	16
1.5 Markov Chain Model	19
1.6 Summary & Future Work	28
2 Cover of high-dimensional chaotic flows with symmetries by embedded coherent structures	30
2.1 Preliminaries	30
2.2 Constructing the Cover	31
2.3 Kuramoto-Sivashinsky Equation	32
2.4 Application to the KSE	34
2.5 Shadowing	43
2.6 Markov Chain Model	51
2.7 Minimising the Cover	56
2.8 Segment-Based Cover	59
2.9 Summary & Future Work	64
3 Construction of approximate generating partitions of chaotic systems by embedded periodic orbits.	66
3.1 Motivation	69
3.2 Maps	70
3.3 DLBD Method	76
3.4 Dynamic Addition of Symbols	80

3.5	Normalisation of Z .	85
3.6	Staggered Construction	90
3.7	Gaussian Proximity Kernel	92
3.8	Approximate Generating Partitions	95
3.9	Standard Map	100
3.10	Summary & Future Work	103
4	Concluding Remarks	107
	Bibliography	110

Introduction

Periodic Orbit Theory

Periodic orbits play a fundamental role in the analysis of various types of dynamical systems. This importance was perhaps first noticed by Poincaré who hypothesised that any motion of a dynamical system can be approximated by those of periodic type (which was later proven by Birkhoff[1] in 1927); this trait is especially applicable to systems exhibiting chaotic dynamics, where it is well-proven that unstable periodic orbits must be dense in the chaotic set[2]. This profound importance of periodic orbits to chaotic systems led to the recent popular description by Cvitanović[3] of periodic orbits as “the skeleton of classical and quantum chaos”.

Poincaré’s above hypothesis led to the development of the branch of dynamical systems theory known as ‘periodic orbit theory’, which revolves around determining geometric and dynamical properties of chaotic systems, such as natural measure, Lyapunov exponents, fractal dimensions, entropies, etc. by using the properties of embedded unstable periodic orbits (UPOs)[4]. An example of the usage of periodic orbits in the calculation of dynamical properties of low-dimensional chaotic systems is given by Cvitanović[5] where cycle expansions[6] are used to calculate dynamical averages such as Lyapunov exponents, diffusion constants, and power spectra of chaotic systems.

This approach has also been shown to work well for low-dimensional attractors of high-dimensional systems[7, 8], however when the dimensionality of the attractor is large (several positive Lyapunov exponents) even a qualitative description of the attractor becomes difficult, an issue we attempt to breach in Chapter 2.

Computational Methods for Detection of Unstable Periodic Orbits

In recent years a considerable amount of progress has been made in locating periodic orbits and other coherent structures in high dimensional chaotic systems. Examples include the work of López et al.[9], who presented a method for finding relative periodic solutions for differential equations with continuous symmetries, and used this method to find relative periodic solutions of the one-dimensional complex Ginzburg-Landau equation; they acknowledged, however, that without modifications their method would not be suitable for use in systems of higher dimension. Hof et al.[10] found experimental evidence of the existence of travelling waves in turbulent pipe flow, in agreement with the numerical studies of Faisst et al.[11] & Wedin et al.[12]. Additionally, Zoldi and Greenside[13] used a damped-Newton method to find unstable periodic orbits in the Kuramoto-Sivashinsky (KS) equation.

More recently, Cvitanović et al.[14] used multiple shooting and the Levenberg-Marquardt algorithm to locate over 60 000 coherent structures in the KS equation on a periodic domain. In this paper, Cvitanović et al.[14] claim that in comparison to the method of Zoldi & Greenside[13], their method converges faster due to the removal of artificial constraints (added by Zoldi & Greenside to avoid their system of equations from being underdetermined) which is enabled by the use of the LM algorithm which solves the system instead as an optimisation problem. It's also worth mentioning that Zoldi & Greenside acknowledged in their work that natural progressions to their method may include multiple shooting.

In the case of Cvitanović et al.[14], they also acknowledged that due to the presence of discrete and continuous symmetries, these coherent structures include not only the traditional equilibria and periodic orbits (such as those found by Zoldi & Greenside), but also travelling waves and relative periodic orbits[9, 14] (also known as 'modulated travelling waves'[15]).

Another popular method is the stabilising transformations method first proposed by Schmelcher and Diakonov[16] and further developed by Crofts & Davidchack[4, 17] which revolves around finding a transformation for the system such that the unstable periodic orbits become stable. Upon doing so, locating the periodic orbits becomes as simple as following the dynamics until a limit cycle is reached.

Throughout this thesis, we will make extensive use of the set of over 60 000 coherent structures for the Kuramoto-Sivashinsky equation found by Cvitanović et al.[14], as well as sets of unstable periodic orbits for the Ikeda, Hénon, Duffing, and Standard maps found by Davidchack et al. using the stabilising transformations method[4, 18, 19].

1 Cover of 2-dimensional chaotic maps by embedded periodic orbits

Poincaré's aforementioned hypothesis that any motion of a dynamical system can be approximated by those of periodic type opened a path into the previously unbreachable territory of chaotic dynamics, and paved the way for a periodic orbit theoretical approach to the calculation of many dynamical properties & averages of such systems[5]. Despite this, few attempts have been made to expand on the true essence of this hypothesis - the approximation of trajectories by periodic orbits (be it the approximation of periodic orbits by other periodic orbits, or of non-periodic trajectories by periodic orbits).

The main reason for the lack of exploration in this direction is down to the fact that (unstable) periodic orbits are dense in any chaotic attractor, and so the finding of all UPOs even up to some fixed low period was, until recently[4, 9, 14], an extremely difficult task both computationally and theoretically. Following the mantra of Cvitanović[5] that short cycles give the overall structure of the dynamics, with longer cycles giving quickly decaying adjustments, we proceed under the assumption that so long as we have all periodic orbits up to a sufficiently high period then we will be able to approximate the dynamics reasonably well. However, even by restricting our periodic orbits to those of up to a certain period, for higher-dimensional systems the number of periodic orbits of even lower periods can become unmanageably high.

With this in mind, for now, we will focus on the case of 2-dimensional maps exhibiting chaotic dynamics. We propose a selection process by which only periodic orbits that provide unique information about the dynamics in their vicinity are chosen to represent the chaotic attractor - in other words, we will construct a cover of the attractor. We then proceed to explore how this cover can be used to represent generic trajectories in the attractor as a sequence of transitions between the chosen periodic orbits. Finally, we present how this transition information can be used in order to construct a Markov

Chain representation of the system, and how this can be used in order to rank periodic orbits in the cover in terms of importance to the overall dynamics.

Throughout this part of the thesis we will exclusively focus the application of the presented methods and analyses in the context of the Ikeda map (which will be defined later) with parameters chosen such that chaotic behaviour is exhibited. Despite this, in order to ensure generality of the presented methods we will make use of no specific traits of the two-dimensional Ikeda map; the methods can equally be applied to maps of any dimension.

1.1 Hausdorff Distance

In order to calculate the distance between segments of orbits (a key theme of our methods), we will make use of the directed Hausdorff distance between sets endowed with the Euclidean norm (for simplicity). Roughly speaking, the directed Hausdorff distance describes the distance from orbit segment A to orbit segment B as the distance between the furthest point in A to its closest point on B - if this distance is small, it means that A stays close to B for its whole duration; below we give a more thorough definition and explanation.

The Hausdorff distance **between** sets A & B is defined as $H(A, B) = \max(h(A, B), h(B, A))$, where $h(A, B)$ corresponds to the directed Hausdorff distance **from** A to B, and is defined for a chosen norm $\|\cdot\|$ as:

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

Intuitively speaking, the directed Hausdorff distance gives the supremum of the set of distances of each element of A to their closest neighbour in B; it follows that the undirected Hausdorff distance returns the maximum of these two suprema. In Fig. 1 we show an example of how $h(A, B) \neq h(B, A)$. Due to this property, it's worth noting that the directed Hausdorff *distance* is not a distance in the strict Mathematical sense, as it doesn't satisfy the symmetry axiom.

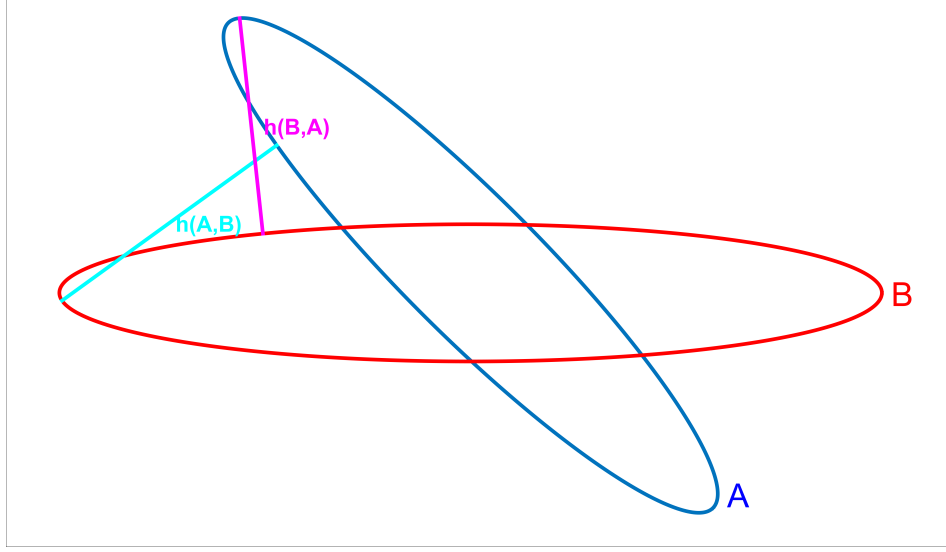


Figure 1: Directed Hausdorff distances between sets A and B.

Both the directed and undirected Hausdorff distances have a long history of use in object & pattern recognition [e.g. 20, 21, 22]. An example of how they would be used in such a field is as follows: Assume that you have an object with edges defined by set A, that you wish to match to two images with edges defined by B and C (depicted in Fig. 2). In this example $h(A, B)$ would be relatively small as image A can be found inside image B; however the converse isn't true, image B cannot be found as a whole in image A, so $h(B, A)$ would be much larger - since $h(B, A)$ is large, that also means the undirected Hausdorff distance $H(A, B)$ would be large. An example of where the undirected Hausdorff distance would be small would be in the case of $H(A, C)$, where A is similar to C, and C is also similar to A.

1.2 Constructing the Cover

Before presenting the method by which we propose one can construct a cover of a chaotic attractor by using unstable periodic orbits, a definition of what a cover is seems prudent: For a set X with distance function $d(\cdot, \cdot)$, a *closed cover* Y of X is defined as a family of closed subsets whose union covers X [23]; in other words, $X \subseteq \bigcup_{y \in Y} y$, where each $y \subseteq X$ is closed.

In our case, we iteratively construct a cover of the attractor (the 'set') by ε -neighbourhoods

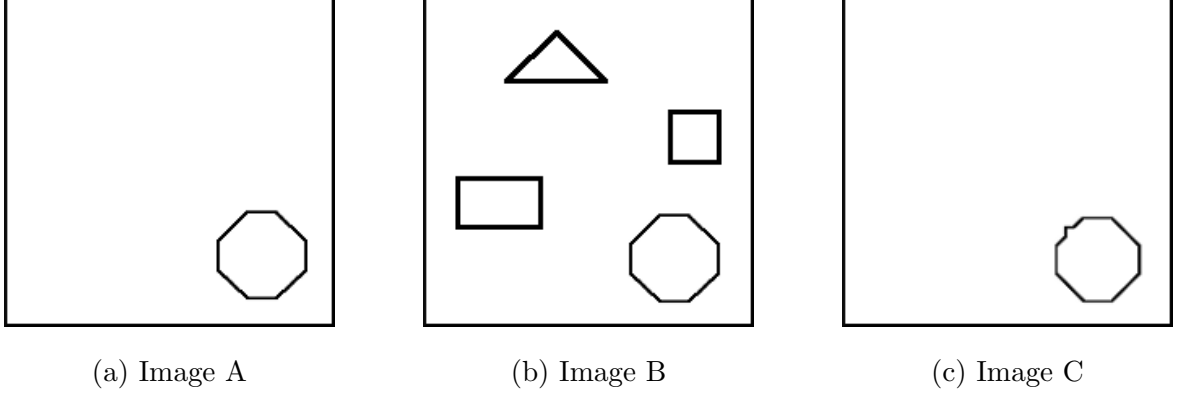


Figure 2: Images A, B, and C.

of certain unstable periodic orbits in the attractor (the ‘subsets’). The way these orbits are chosen to be included in the cover is outlined below.

The selection process proceeds as follows: Let $P = \{p_i\}_{i=1}^{N_P}$ be a set of unstable periodic orbits listed in order of increasing period. Starting with the empty set as the initial basis of the cover $W = \emptyset$, iteratively proceed through the orbits in order of period, adding into the cover those orbits whose distance to W is above some threshold, ε .

For $i = 1, 2, \dots, N_P$:

$$W \leftarrow \begin{cases} W \cup \{p_i\} & \text{if } d(p_i, W) > \varepsilon \\ W & \text{if } d(p_i, W) \leq \varepsilon \end{cases} \quad (1.1)$$

where $d(p_i, W)$ denotes the directed Hausdorff distance between sets p_i and W , given as follows:

$$d(p_i, W) = \max_{t \in \{1, 2, \dots, T^{p_i}\}} \min_{\substack{w_j \in W \\ \tau \in \{1, 2, \dots, T^{w_j}\}}} \|p_i(t) - w_j(\tau)\| \quad (1.2)$$

where $T^{p_i}, T^{w_j} \in \mathbb{Z}$ are the periods of orbits p_i and w_j respectively; p_i is a given orbit in set P , w_j is an orbit in the cover W ; and $\|\cdot\|$ is the chosen norm in the dynamical system phase space with a given metric. What this distance gives us is the maximum of the minimum distances of each point on $p_i(t)$ to the closest orbit in W at each time step t ; in other words, the distance from the furthest point on p_i to the closest point in set W . Intuitively speaking, we only include a new periodic orbit into the set W if it

explores a part of the attractor that is not yet covered by the current W .

By including orbits into the cover who satisfy $d(p_i, W) \leq \varepsilon$, we are effectively constructing a closed ε -ball around the set W that fully contains the set P , in other words: $P \subseteq B_\varepsilon(W)$.

The choice of ε can have quite a significant effect on the resulting covering of the attractor. For larger ε , one can expect to see a reduction in the number of elements of the resultant cover, at the cost of potentially poorer accuracy of approximation of the dynamics on the attractor. Conversely, for smaller ε one can expect a larger number of cover elements, with the benefit of a more accurate approximation of the dynamics on the attractor.

While the distance threshold ε can vary depending on the outcome one seeks (i.e. a smaller cover set, or more accurate approximation), for the sake of consistency throughout this section we will adopt the following way of choosing this parameter: ε will be chosen to be 1% of the length of the lead diagonal of the rectangle bounding the attractor; in other words, 1% of the norm of the vector containing the length of the attractor in the x and y directions. This sets the *resolution* of the cover, with smaller ε giving a higher resolution (i.e. a more accurate representation) but increasing the amount of elements in W .

1.3 Application to Ikeda Map

In order to give a concrete example of the results of the above processes, we will apply the methods to the Ikeda-Hammel-Jones-Moloney map[24, 25] (which will henceforth be referred to simply as the Ikeda Map):

$$x_{n+1} = a + b(x_n \cos \phi_n - y_n \sin \phi_n) \quad (1.3)$$

$$y_{n+1} = b(x_n \sin \phi_n + y_n \cos \phi_n) \quad (1.4)$$

where $\phi_n = k - \eta/(1 + x_n^2 + y_n^2)$, with parameters $a = 1.0$, $b = 0.9$, $k = 0.4$ and $\eta = 6.0$ chosen such that the map exhibits chaotic dynamics. We will use a set that contains an exhaustive list of unstable periodic orbits ranging from period 1 to period 20 (provided by Davidchack et al.[26]) totalling 20,061 orbits, in order to construct the cover. As a side note, we will exclude one of the two equilibria located at $(-1.1143, -2.2857)$ from this set as it lies outside of the actual attractor (on the border of the basin of attraction), and so has no influence on the dynamics within the attractor itself; as such, we'll only use 20,060 of the 20,061 orbits in the set. A portrait of the attractor of the Ikeda map can be seen in Fig. 3.

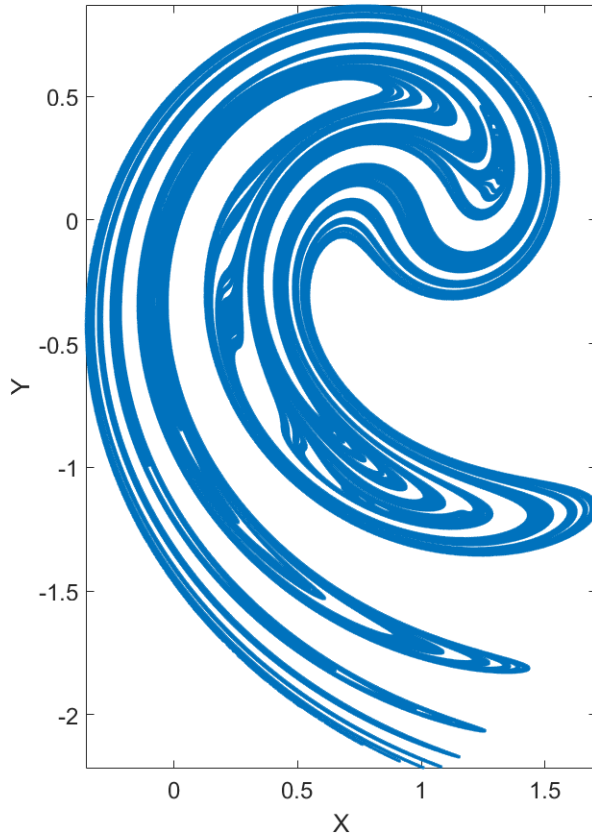


Figure 3: Portrait of the Ikeda map's attractor.

In the construction of the cover we will choose ε as previously described, by taking it to be 1% of the length of the main diagonal of the attractor; since the attractor is bounded by the rectangle: $x_n \in [-0.3516, 0.7082]$, and $y_n \in [-2.2857, 0.8645]$, we choose $\varepsilon \approx 0.0376$. By applying the covering algorithm (Eq. 1.1) with this ε , our set of 20,060 unstable periodic orbits is reduced to a cover containing only 271 orbits.

A visualisation of the covering of the Ikeda map can be seen in Fig. 4; the points of the 20,061 unstable periodic orbits of the attractor are shown in black, and the points of the unstable periodic orbits included in the cover are in red with their ε -neighbourhood shown by surrounding red disks. The covering provided by using periodic orbits up to period 5, 10, 15, and 20 (corresponding to 12, 115, 1,378, and 20,061 UPOs respectively) are shown from left to right. By the time we reach period 20, it's clearly observable that the set of 20,060 periodic orbits are completely covered by the ε -neighbourhoods of the 271 cover orbits (by construction); it's also quite clear that there is quite a lot of redundancy in the set, this can be most clearly seen in the rightmost subfigure of Fig. 4 where we can see some dense red areas where multiple neighbourhoods are intersecting. The resolution of this redundancy is, however, beyond the scope of this thesis, and will be left as a problem for future research.

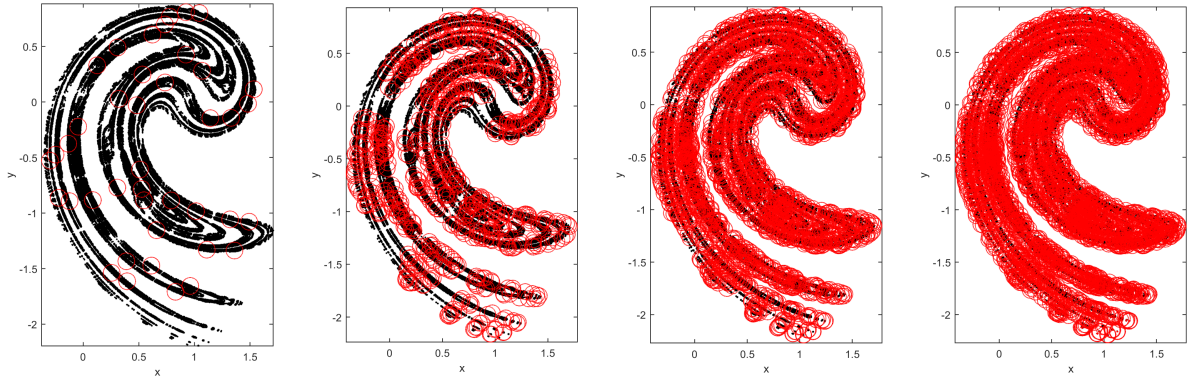


Figure 4: Covering of the Ikeda map using unstable periodic orbits of period up to 5, 10, 15, and 20 (from left to right), with the red circles corresponding to the ε -neighbourhood around chosen cover elements.

By construction, the set P is fully contained by the closed ε -ball surrounding W . It's safe to assume, however, that the set P doesn't contain every single point in the attractor, and so we will assume that in order to cover the whole attractor an ε -ball must also be taken around P .

As such, to ensure that the whole attractor C is covered we must henceforth deal with a set that covers $C \subseteq B_\varepsilon(P)$. Using the fact that by construction $P \subseteq B_\varepsilon(W)$, the

following must hold true:

$$C \subseteq B_\varepsilon(P) \subseteq B_\varepsilon(B_\varepsilon(W)) = B_{2\varepsilon}(W) \quad (1.5)$$

and so by taking a closed 2ε -ball around W we can ensure that the whole attractor is now covered.

An illustration of the covering provided by $B_\varepsilon(W)$ and $B_{2\varepsilon}(W)$ can be seen in Fig. 5; in red are the ε and the 2ε balls surrounding W respectively, and in black are the points of the chaotic attractor that were generated by taking a long (approximately 10^9 time steps) trajectory in the attractor, which represent the above described set C . By focusing our attention on the bottom of the attractor in each case, it's clear to see that indeed $B_\varepsilon(W)$ doesn't cover the whole of C , an issue which is resolved by using $B_{2\varepsilon}(W)$.

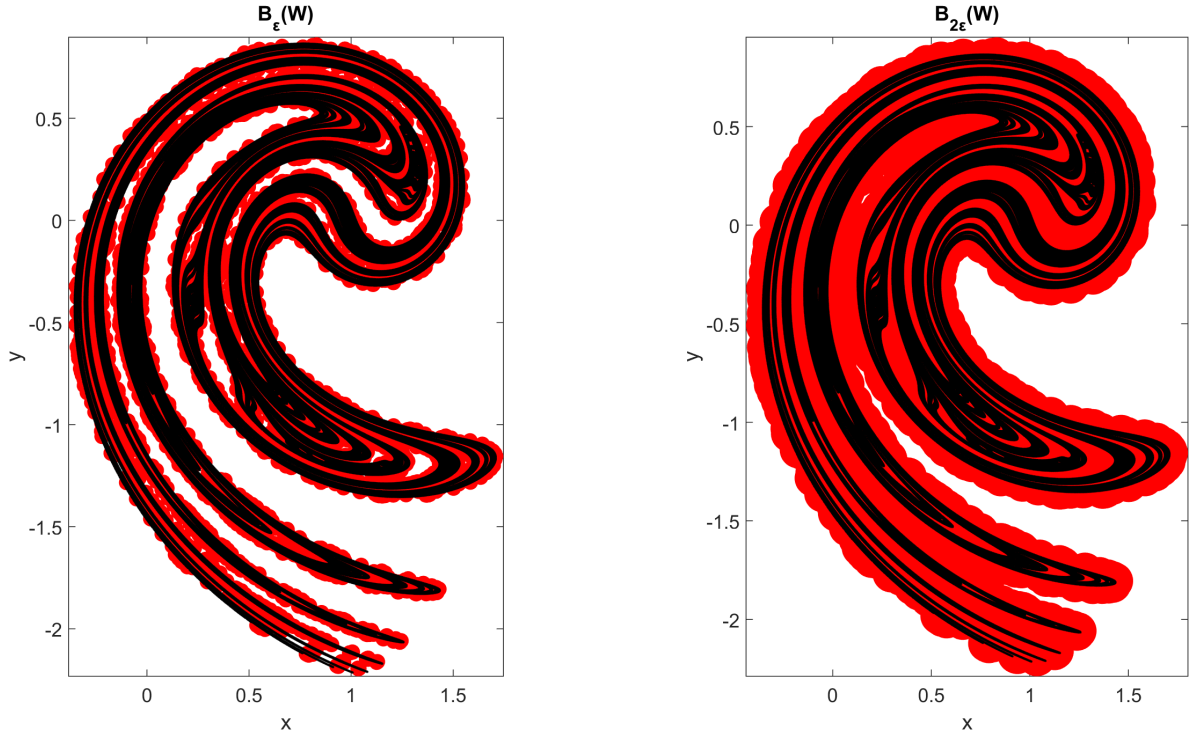


Figure 5: Here in red we show the ε -ball surrounding W (left), and the 2ε -ball surrounding W (right); in black are the points of the chaotic attractor of the Ikeda map.

With this in mind, we will instead use 2ε in the following applications of the cover W .

1.4 Shadowing

Now that we've reduced P to the set W , whose 2ε -neighbourhood provides at least the same covering of the chaotic attractor as the ε -neighbourhood of P , it is possible to represent any trajectory in the attractor as a sequence of transitions between nearby elements (i.e. within 2ε distance) of W .

As a starting point we will first seek to create such a representation in the case of periodic orbits in the attractor that don't belong to W (i.e. in $P \setminus W$), after which we will generalise this method to represent any trajectory in the attractor in a similar manner.

Finding such a sequence of transitions can be done in a multitude of ways (discussed in more detail on page 45); the simplest such way being for some $p_i \in P \setminus W$ finding the closest $w_i \in W$ to each point on p_i . This method can however, produce sequences that jump frequently between different cover elements; as we would like a more *natural* representation with less frequent jumps, and smoother transitions. An example of this is the itinerary produced by applying the above shadowing scheme to the period 10 orbit $p_{78} \notin W$: $w_6(2) \rightarrow w_{36}(1) \rightarrow w_{17}(4) \rightarrow w_{36}(1) \rightarrow w_6(2)$, where $w_i(t_i) \rightarrow w_j$ corresponds to a transition from w_i to w_j with a t_i waiting time before transitioning. Taking into account the periodicity of the orbit, this period 10 orbit of the above described Ikeda map requires 4 transitions each with relatively short *waiting times* (time spent on the orbit before transitioning, shown in brackets). While this result may seem acceptable, for longer trajectories this problem becomes more apparent.

In order to try to resolve this issue, we sought an algorithm that prioritises longer close approaches over short but closer approaches. The best solution to this problem would require solving a minimisation problem to minimise the number of segments (all of which must be within 2ε distance), however for long trajectories the computation times may become prohibitively large; as such, we will assume a naïve, greedy approach where we start from a given point and proceed forward, finding the cover element that

stays within 2ε distance for the longest (hence greedy). This will unfortunately make the result dependant upon starting point, at the benefit of reducing computation times significantly; the true benefit of this particular algorithm will become clear in the second part of the thesis where we will discuss the application of this method to projections of higher-dimensional attractors of chaotic flows.

Starting with $\tilde{W}_0 = W$, our proposed naïve, segment-based algorithm for shadowing of a periodic orbit or trajectory is as follows:

1. Starting from $\mathbf{x}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$, find $\tilde{W}_0 \supseteq \tilde{W}_1 = \left\{ w \in \tilde{W}_0 \mid d(\mathbf{x}_0, w) \leq 2\varepsilon \right\}$.
2. Repeat this process, finding $\tilde{W}_{i-1} \supseteq \tilde{W}_i = \left\{ w \in \tilde{W}_{i-1} \mid d(\mathbf{x}_{i-1}, w) \leq 2\varepsilon \right\}$.
3. When k is found such that $\tilde{W}_k = \emptyset$, label the segment $\varphi_1 = \{\mathbf{x}_j, j \in [0, k-1]\}$ by the index j of $w_j \in \tilde{W}_{k-1}$. In the case that \tilde{W}_{k-1} contains multiple orbits, label the segment by the lowest period orbit in \tilde{W}_{k-1} .
4. Repeat this process starting from \mathbf{x}_k , until the whole trajectory is labelled.

Intuitively speaking, starting from a given point, this algorithm finds the periodic orbit in W that stays within 2ε distance of this point for the longest, labelling the segment by the index of the periodic orbit. This process is then repeated until the whole trajectory is labelled. In the case of p_{78} , the itinerary obtained from this algorithm is: $w_6(5) \rightarrow w_{17}(5)$; compared to the aforementioned closest-distance shadowing which required 4 transitions for this orbit we see a marked shift towards longer shadowing with fewer jumps.

To give another example, we will look at the first orbit not included into W by the covering algorithm, the period-9 orbit p_{54} . Using the naïve segment-based shadowing algorithm we can represent p_{54} as the transitions between three different cover elements, with itinerary: $w_{24}(4) \rightarrow w_{21}(3) \rightarrow w_{17}(2)$. In Fig. 6 we can see how each of the points on the trajectory p_{54} (shown as blue crosses, with each point in the itinerary numbered accordingly) are within 2ε -distance (shown by circles) of the relevant point

in the itinerary.

This figure gives an illustration of how our algorithm should be working: points 1 and 2 are both within 2ε -distance of w_{24} **and** w_{17} , however points 3 & 4 are within 2ε -distance of w_{24} but not w_{17} , and so w_{24} is chosen to represent the first 4 points. This also brings to light one of the drawbacks of the algorithm, since the orbit is periodic there are actually 7 points in a row within 2ε -distance of w_{24} : 7, 8, 9, 1, 2, 3 & 4, however points 8, & 9 are labelled by w_{17} as it's a shorter orbit than w_{24} ; if we were to change the starting point of the shadowing instead to point 7, we would get a different (and arguably better) shadowing of our trajectory, requiring only two cover elements. Resolving this is an issue that will be left for future research, as we would first like to present the results in this simple case of shadowing before making such improvements.

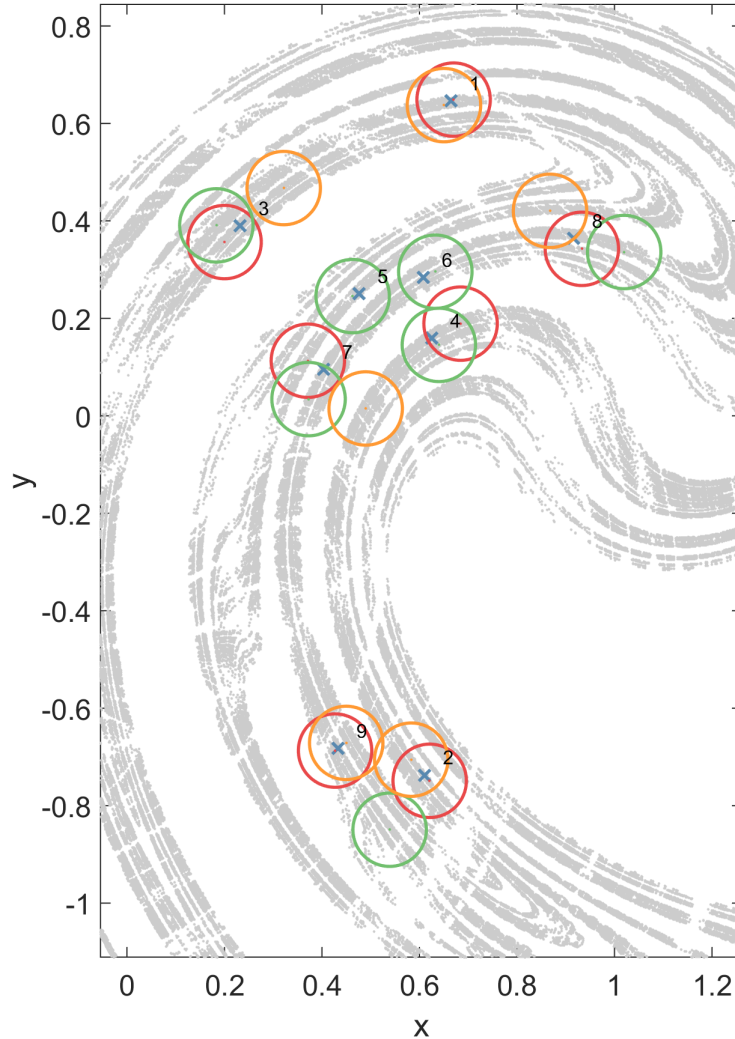


Figure 6: Shadowing of p_{54} (blue crosses) by w_{24} (red), w_{21} (green), and w_{17} (orange). Labels 1-9 to the top right of each point represent the order of the points in the itinerary.

This procedure isn't restricted to only periodic orbits, it can equally be applied to any chaotic trajectory. By applying this process to a long trajectory in the attractor and observing the transitions and waiting times between cover elements we hope to be able to gain insight into the workings of the chaotic attractor; such a representation of the dynamics lends itself quite handily to a Markov-type model, as we will discuss in the following section.

1.5 Markov Chain Model

Given a finite set of states $S = \{s_1, s_2, \dots, s_n\}$, a finite-state Markov chain describes the process by which the system jumps from state to state by the associated transition probability matrix, \mathbf{A} . In order to be considered a Markov chain, the Markov property must be satisfied. This property stipulates that the probability \mathbf{A}_{ij} of transitioning from the current state s_i to the state s_j must be independent of the previous states, also known as *memorylessness*[27].

As a consequence of the Markov property, the waiting times of a discrete-time Markov chain should follow a geometric distribution[28, 29], as this is the only discrete probability distribution that has the memoryless property[30]; similarly, for continuous-time Markov chains (which we will deal with in the next part of the thesis), the waiting times should follow an exponential distribution, being the only continuous probability distribution with the memoryless property[30]. As such, when ascertaining the appropriateness of a transition-based process for representation as a Markov chain, the distribution of waiting times is a good place to begin.

As the result of the previously described shadowing algorithm is precisely given as the set of waiting times on each cover element and the transition to the next cover element it makes sense to model our dynamics by a Markov chain; specifically, as we

have a finite number of cover elements, and are working with a discrete-time map, we will use a discrete-time finite-state Markov chain. Assuming that the memoryless property holds (which we will check later), such a model will allow us to gain more detailed information about the dynamics of the system under the shadowing regime with our given cover.

1.5.1 Construction

The construction of a Markov chain approximation for 2-dimensional maps is a relatively simple task for two reasons: generating and shadowing long orbits in 2-dimensions is extremely fast, and since maps are discrete by nature we can use a simple discrete Markov chain. By performing shadowing on a long piece of trajectory, started from a random point within the attractor, we reduce the trajectory to a set of waiting times (the duration of the stay on a given cover element) and transitions from the current cover element to the next; the transition matrix for the Markov chain can be constructed quite simply using these transitions and waiting times. Using the shadowing data on the long trajectory, the diagonal elements of the transition matrix \mathbf{A}_{ii} can be set as the sum of waiting times on each cover element w_i ; the non-diagonal elements \mathbf{A}_{ij} will be the sum of all occurrences when the shadowing changes from w_i to w_j . It's worth noting that such a matrix will have all positive elements. Upon completion of this process the matrix should be normalised to be *right-stochastic*, in other words such that all rows sum to 1, $\sum_j \mathbf{A}_{ij} = 1$.

To give a hypothetical example, if we were to perform shadowing on a trajectory started from a random point in the attractor using a cover of two elements for 6 time steps with the resultant itinerary: $w_2 \rightarrow w_2 \rightarrow w_2 \rightarrow w_1 \rightarrow w_2 \rightarrow w_1$, we would begin the construction of our transition matrix \mathbf{A} by increasing $\mathbf{A}_{2,2}$ by the number of self-transitions from our starting point, in other words $\mathbf{A}_{2,2} = 2$. Next, as we see a transition from w_2 to w_1 we would set $\mathbf{A}_{2,1} = 1$; similarly, for the next transition we would set $\mathbf{A}_{1,2} = 1$. Following this is a transition back from w_2 to w_1 , so for our final step we would increase $\mathbf{A}_{2,1}$ by 1 giving $\mathbf{A}_{2,1} = 2$. In other words, $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 2 & 2 \end{pmatrix}$. Finally, we must normalise \mathbf{A} to

ensure that the rows sum to 1, giving $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$.

By using our cover W to shadow a trajectory started from a random point in the attractor of duration 1.9×10^8 (compared to the 3,297 total time of cover elements) we computed the 271×271 transition matrix \mathbf{A} : a sparse matrix with around 74% zero elements.

1.5.2 Properties

In order to represent the transition data accurately by a finite-state Markov chain, the *Markov property* (also known as *memorylessness*) must be satisfied; roughly speaking, memorylessness means that the next state of the system depends only upon the current state, and not any previous states. The easiest way to check whether observed data satisfies this property is by checking whether or not the frequency of the waiting times (in our case the amount of time the shadowed trajectory stays each cover element before transitioning) follow a geometric distribution. Fig. 7 shows the distribution of waiting times for our system with the waiting times on the horizontal axis, and their log-scaled frequencies on the vertical axis.

Were the distribution to be geometric, we'd expect a log-scaled frequency plot to decrease linearly; while this is not the case for waiting times below 5, for the waiting times above 5 we do indeed see the behaviour expected of a geometric distribution[28, 29]. These first few waiting times that don't fit the pattern are likely in part an artefact of our greedy shadowing algorithm that avoids creating segments with short waiting times.

The main factor that affects the segments with short waiting times not following the geometric distribution however, is the fact that we're dealing not with a stochastic system but with a deterministic chaotic dynamical system, and so there is in fact memory in the system. Due to the combination of sensitive dependence on initial conditions of chaotic dynamics and rounding errors, after a certain time all memory of where the trajectory came from will be lost, as this initial error will be propagated exponentially fast;

an example of this in the context of the Ikeda attractor can be seen in Fig. 8 where the distance between two points initially separated by machine precision of $\epsilon = 2^{-52}$ grows exponentially until approximately the 71st iterate at which point the distance stops growing exponentially and begins to resemble noise. The time it takes to reach this point is determined by the maximum Lyapunov exponent of the system. As such, while there is memory in the system, due to rounding error this memory is extremely limited, and so on longer time scales the dynamics can be approximated with a stochastic process. This particular issue is discussed in detail in many different contexts, including research into distinguishing chaotic processes from stochastic processes/noise[31, 32], and also in the context of the applicability of Markov models to chaotic systems[33, 29].

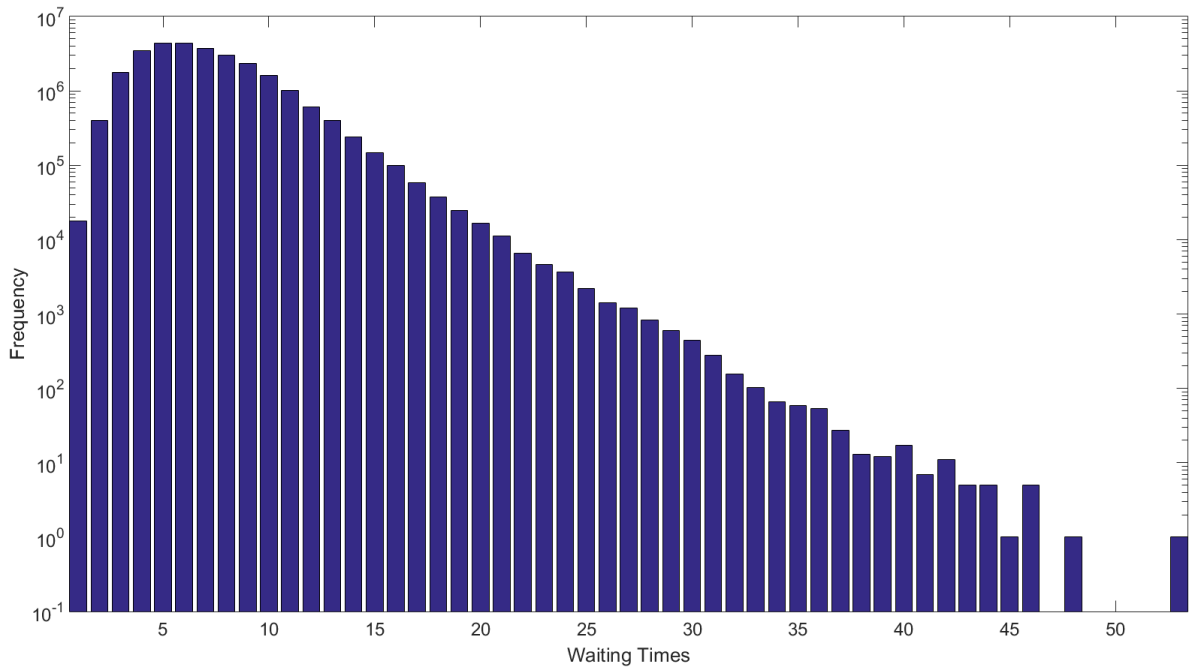


Figure 7: Log-Scaled Frequency of Waiting Times.

As these segments with short waiting times constitute a relatively small portion of the total time (approximately 10%), we can proceed relatively safely under the assumption that our data can be suitably represented by a Markov model on longer time scales.

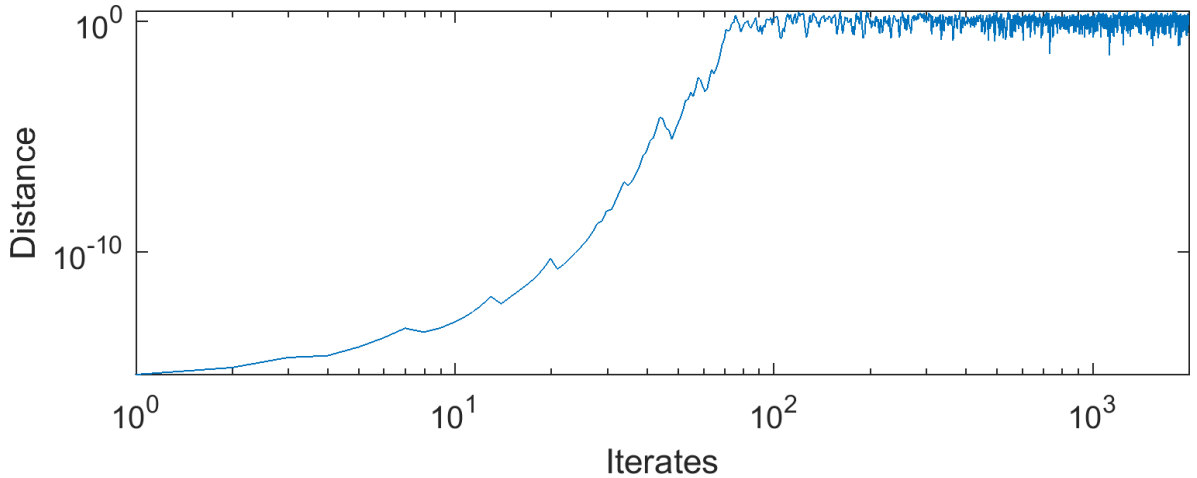


Figure 8: Propagation of distance between two points initially separated by machine precision $\epsilon = 2^{-52}$ over successive iterates of the Ikeda map.

One of the fundamental properties of Markov chains is that of *ergodicity*. A finite-state Markov chain is considered to be ergodic if it's possible to reach any state from any other state (in any number of steps)[34] (i.e. $\forall i, j \exists t \in \mathbb{N} : \mathbf{A}_{ij}^t > 0$). By considering the transition matrix \mathbf{A} of the Markov chain in the context of a directed graph[35], it's possible to determine ergodicity of the Markov chain by finding the number of connected components in the graph - if this number is one, the graph is fully connected, and hence the Markov chain is ergodic. Using MATLAB's `graphconncomp` command \mathbf{A} , we determined that the Markov Chain defined by \mathbf{A} is fully connected (having only one connected component), and so it is indeed ergodic.

1.5.3 Stationary Distribution

Another useful piece of information that we can gain from the transition matrix is the *stationary distribution* of the Markov chain. The stationary distribution of the transition matrix \mathbf{A} is the row vector π such that $\pi\mathbf{A} = \pi$, i.e. the left eigenvector of \mathbf{A} corresponding to eigenvalue 1. Such a stationary distribution exists and is unique for every ergodic finite-state Markov chain[34] (which as stated earlier \mathbf{A} is). Since π gives the probability of being at any w_i at any given point in time, by sorting the values of π we can effectively rank the cover elements in order of importance (in the

sense that a cover element is more important if the probability of being there is higher).

Calculation of π can be done in two ways. The first is through the power method, by finding n such that $\mathbf{A}^n = \mathbf{A}^{n-1}$, taking π to be any row of \mathbf{A} ; the second is by calculating the left eigenvector of the matrix of \mathbf{A} corresponding to eigenvalue 1, and rescaling it such that $\sum \pi = 1$ (with all $\pi_i > 0$, since they must be non-zero probability values (due to ergodicity)). We first calculated π using the former method, requiring iterating up to $n = 608$ in order to calculate π to MATLAB's machine precision; for comparison we then calculated the left eigenvector using `[~,D,W]=eig(A)`, and selecting the column of \mathbf{W} corresponding to $D(i,i)=1$, transposing (from column to row vector), and then dividing by a factor of 14.18. As expected, both of the resulting row vectors were identical.

To give a visualisation of how π can be used to rank the states of our Markov chain, Fig. 9 shows each π_i relative to the mean ($\mu_\pi = \frac{1}{271} \approx 0.00369$), along with horizontal lines depicting one standard deviation ($\sigma(\pi) \approx 0.0022$) above and below the mean. Judging by eye alone we can see that w_{223} is clearly ranked highest in importance, with a probability of over 1.4% at almost 5 standard deviations above the mean. By contrast, w_1 , w_2 , w_4 , & w_{124} rank lowest with more than 1.6 standard deviations below the mean, which corresponds to a probability of around 0.005%.

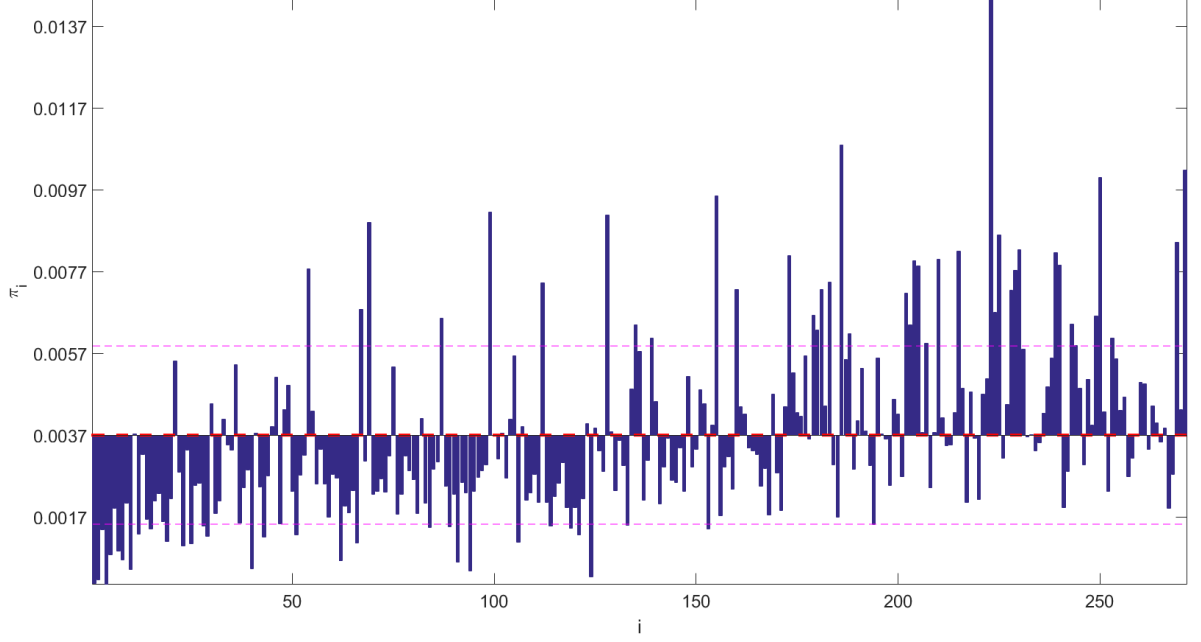


Figure 9: π_i of each cover element i , with the middle red line corresponding to the mean μ_π and the thin red lines corresponding to the one standard deviation above and below the mean respectively ($\mu_\pi \pm \sigma(\pi)$).

In Fig. 10 the probabilities π_i are sorted in descending order, with s_i being the corresponding list of sorted indices. By taking the cumulative sum of these probabilities (sorted in descending order), we can see how many orbits in W are required to shadow varying percentages of the dynamics on the attractor (results can be seen in Table 1). From this we can glean information such as that the system spends over 50% of its time on less than 30% of the orbits in W .

Cumulative Sum	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Number of Orbits	10	23	38	56	78	102	130	163	203	271
Percent of Orbits	3.7%	8.5%	14%	20.7%	28.8%	37.6%	48%	60.1%	74.9%	100%

Table 1: Number of orbits in W required to account for varying amounts of the dynamics.

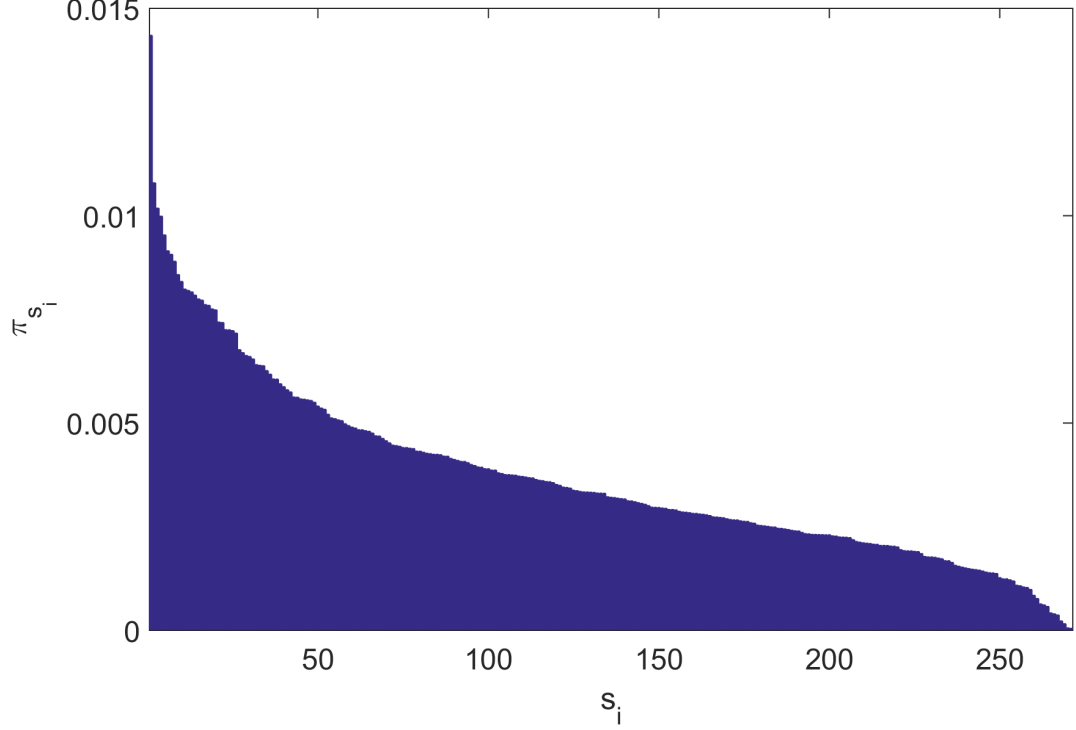


Figure 10: Distribution of probabilities of π , sorted in descending order.

1.5.4 Mean First Passage Time

While the stationary distribution gives the long-time probability distribution of being in each state at any given time, for a more short-term measure of the dynamics of the Markov chain we will turn to the measure known as the *Mean First Passage Time*.

The mean first passage time is the average time required to go from a given state to another given state for the first time, and was defined for an N -state discrete Markov chain by Kemeny & Snell[36] as the matrix \mathbf{M} , where:

$$\mathbf{M} = (I - Z + EZ_{dg})D \quad (1.6)$$

where I is the $N \times N$ identity matrix, E is the $N \times N$ matrix of ones, D is the diagonal matrix with elements $D_{ii} = \frac{1}{\pi_i}$, and Z is the fundamental matrix (where Z_{dg} is the matrix containing only the diagonal elements of Z), defined as:

$$Z = (I - \mathbf{A} + \Pi)^{-1}$$

where Π is the matrix with each row equal to the stationary distribution π .

In the resultant matrix \mathbf{M} , element \mathbf{M}_{ij} corresponds to the mean time required to reach w_j from w_i for the first time, the *mean first passage time from w_i to w_j* , and \mathbf{M}_{ii} corresponds to the mean time required to return to w_i from w_i for the first time, the *mean first return time for w_i* .

In order to try to find useful patterns from \mathbf{M} , we define m_j as the $(N - 1) \times 1$ column vector containing the j^{th} column of \mathbf{M} with the diagonal element \mathbf{M}_{jj} excluded. By calculating the relative distance of each element i of m_j to the mean \bar{m}_j given by $d_j^i = (m_j^i - \bar{m}_j) / \bar{m}_j$, we can search for pairs in \mathbf{M} that have uncharacteristically large or small mean first passage times. (Note: When converting the (i, j) values from m to \mathbf{M} one must take care to account for the fact that the diagonal elements of \mathbf{M} were removed in order to calculate m).

For example, using the transition matrix \mathbf{A} and stationary distribution π for the Ikeda map as calculated above, we can calculate the matrix \mathbf{M} with relative ease using Eq. 1.6. By calculating the above distances d_j^i for each (i, j) we found examples of orbits with mean first passage times significantly below the mean for that column. For example, $\mathbf{M}_{124,256} \approx 975$, compared to $\bar{m}_{256} \approx 1623$; intuitively this means that starting from most orbits would require an average of approximately 1623 time steps in order to reach w_{256} for the first time, however when starting from w_{124} it only takes an average of 975 time steps to reach w_{256} . This example, along with many others can be seen in Fig. 11 where on the horizontal axis is the index j of m_j , and on the vertical axis is d_j^i ; the dimension corresponding to index i has been collapsed for easier viewing.

The previously described point $\mathbf{M}_{124,256}$ can be seen in the bottom right of the image (indicated by a red circle) with $d_{256}^{124} \approx -0.4$.

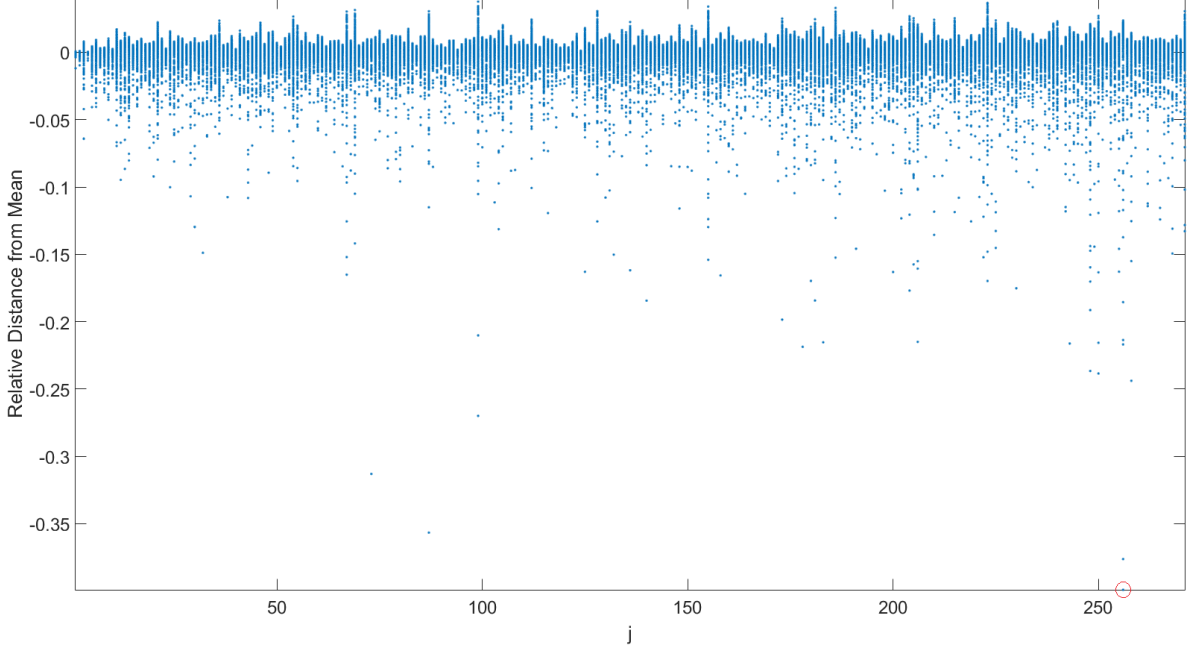


Figure 11: Relative distance of m_j^i from mean \bar{m}_j for each i, j . Dimension i has been collapsed for ease of viewing, and so all m_j^i are shown in the j^{th} column.

By isolating points with large (positive or negative) relative distances from the mean, we can potentially find pairs of cover elements between which there is some kind of dynamical link, opening another avenue for future exploration.

1.6 Summary & Future Work

In this part of the thesis we presented a method by which one may cover a chaotic attractor of a discrete-time dynamical system by a subset of a larger set of periodic orbits, whilst providing the same amount of covering as the larger set.

Using this, we proceeded to present a method by which the dynamics of any trajectory on the attractor can be coarse-grained to a set of transitions between the elements of the aforementioned cover, also known as *shadowing*, in line with Poincaré’s vision of approximating any motion of a dynamical system by those of periodic type.

We then presented an example of one of the potential applications of this new representation of the dynamics in the form of a discrete (in time and state) Markov chain model, using this model to calculate the *stationary distribution* in the context of the *ergodic* Markov chain given by the Ikeda map which allowed us to rank the cover elements in order of importance in the long-term dynamics of the system. Following this, we calculated the *mean first passage time* between each pair of states of this Markov chain which gave us insight into the short-term dynamics of the system, providing examples of pairs of states that have irregularly high or low mean first passage times, opening a new avenue for exploration.

Future work will involve finding methods by which any redundancy in the cover can be reduced, experimenting with the effects that different shadowing algorithms have, exploring further the applications of the Markov chain model of the dynamics, and exploration into other alternative applications of our method.

Before doing so, however, it would be prudent to first test the goodness of the approximation of the attractor given by the Markov chain model, for example by comparing the mean-first passage times and probabilities of being on each cover element at any given time given by the stationary distribution to real data obtained from trajectory segments in the attractor.

In a similar vain to Diego[37], it’s worth mentioning that the Markov model presented here isn’t intended to be a perfect representation of the dynamics, however it may be possible to use it in order to uncover more general information about the system.

This sentiment is further echoed by the word of warning mentioned in [38]:

“... As a result, unlike the near-equilibrium processes of traditional thermodynamics, complex systems do not forget their initial conditions: they carry their history on their backs (Prigogine, Spring 1995, U. S. Naval Academy). Their origin constrains their trajectory.”

2 Cover of high-dimensional chaotic flows with symmetries by embedded coherent structures

In this section we extend the methods described in the previous section to the case of chaotic attractors of flows. We present a method by which the dynamics of high-dimensional systems can be reduced to a cover constructed in a lower-dimensional projection with as little loss of information as possible; we also present an example of a system with symmetries, and show how it's possible to construct a cover for such a system.

In the construction of the cover in the previous part of the thesis, the set P contained only equilibria and unstable periodic orbits. In the case of systems with symmetries, the set P of interesting features of the system may also include such structures as travelling waves, relative periodic orbits, pre-periodic orbits, etc. Such features will henceforth be referred to as *coherent structures*.

2.1 Preliminaries

Throughout this part of the thesis we will almost exclusively focus on the application of our methods and analyses in the context of the Kuramoto-Sivashinsky equation[39, 40], with parameters chosen such that chaotic behaviour is exhibited. We make every effort, however, to ensure that the methods presented make no use of specific traits of the Kuramoto-Sivashinsky equation, so that they can be applied in the contexts of other systems.

Due to the presence of discrete & continuous symmetries in the Kuramoto-Sivashinsky Equation, we will calculate distance in a symmetry-reduced projection where all points related by the group symmetries are mapped to the same point; as such, coherent structures such as travelling waves and relative periodic orbits map to simple equilibria and periodic orbits respectively. To construct such a space, we will project the system dynamics from the full state space onto a subspace which is invariant under the symmetry

transformations of the system. Such a projection can reduce the dimensionality of the system considerably, allowing for much faster computations.

In order to calculate the distance between segments of orbits we will make extensive use of the Hausdorff Distance between sets, which was described in full detail in section 1.1, endowed with the Euclidean norm. Roughly speaking, this distance will describe the distance between two segments as the distance between the two furthest points on the segments - and so if this distance is small, then that means that the two segments stay close for all time.

The use of this particular distance measure is especially important to our methods as we will later deal with low-dimensional projections of the system with phase information removed through the symmetry reductions described above. By calculating the distance between segments of orbits rather than points, we can in a sense get back some of the information that is lost through the symmetry reduction and dimension reduction.

The reason for this is roughly as follows: as lower and lower dimensional projections of the space are taken, it becomes harder to tell whether or not two points are close to one another in the full phase space. By instead taking the distance between segments of orbits rather than individual points, due to the sensitivity to initial conditions of chaotic dynamics, two segments that stay close for a long time in the projection must also be close in the full phase space.

2.2 Constructing the Cover

The cover of potentially high-dimensional flows can be constructed in precisely the same manner as was shown for 2-dimensional maps in section 1.2, the only differences being that we will now deal with a system with continuous time, and that our set P may now contain not only unstable periodic orbits, but also other types of coherent structures.

The selection process proceeds as follows: Let $P = \{p_i\}_{i=1}^{N_P}$ be a set of coherent struc-

tures listed in the order of increasing complexity (e.g., equilibria followed by periodic orbits & relative periodic orbits of increasing period). Starting with the empty set as the initial basis of the cover $W = \emptyset$, iteratively proceed through the structures in order of complexity, adding into the cover those structures whose *distance* to W is above some threshold, ε .

For $i = 1, 2, \dots, N_P$:

$$W \leftarrow \begin{cases} W \cup \{p_i\} & \text{if } d(p_i, W) > \varepsilon \\ W & \text{if } d(p_i, W) \leq \varepsilon \end{cases} \quad (2.1)$$

where $d(p_i, W)$ denotes the directed Hausdorff distance between coherent structure p_i and the set of coherent structures W , given as follows:

$$d(p_i, W) = \max_{t \in [0, T^{p_i}]} \min_{\substack{w_j \in W \\ \tau \in [0, T^{w_j}]}} \|p_i(t) - w_j(\tau)\| \quad (2.2)$$

where T^{p_i} and T^{w_j} are the periods of coherent structures p_i and w_j respectively, p_i is a given coherent structure in set P , w_j is a coherent structure in the cover W , $p_i(t)$ corresponds to t time of the coherent structure, and $\|\cdot\|$ is the chosen norm in the dynamical system phase space with a given metric.

In the case of coherent structures with no period (e.g. equilibrium, travelling wave solutions), the distance of Eq. 2.2 is reduced to:

$$d(p_i, W) = \min_{\substack{w_j \in W \\ \tau \in [0, T^{w_j}]}} \|p_i - w_j(\tau)\| \quad (2.3)$$

In the following sections, we will present the construction and analysis of such a cover in the context of the chaotic attractor of the Kuramoto-Sivashinsky equation. As such, we will first proceed by introducing this equation and its properties.

2.3 Kuramoto-Sivashinsky Equation

The Kuramoto-Sivashinsky equation[39, 40] provides a description of the flutter of a flame front on a one-dimensional spatially periodic domain, and is one of the simplest

nonlinear PDEs that exhibit spatiotemporally chaotic behaviour[14]. The KS equation is defined as follows:

$$u_t = -u_{xxxx} - \frac{1}{2} (u^2)_x, \quad x \in \left[-\frac{L}{2}, \frac{L}{2}\right], \quad t \geq 0 \quad (2.4)$$

with the spatially periodic boundary condition: $u(x + L, t) = u(x, t)$.

Throughout this thesis we follow in the footsteps of Cvitanović et al.[14] who chose to use system size $L = 22$ due to the structurally stable chaotic attractor observed for this system size. Due to the aforementioned spatial periodicity, it's convenient to work instead with the Fourier representation of the solutions of (2.4), given by:

$$u(x, t) = \sum_{k=-\infty}^{+\infty} a_k(t) e^{\frac{2\pi k x}{L}} \quad (2.5)$$

where $a_k \in \mathbb{C}$ and $a_{-k} = a_k^*$ (since $u(x, t)$ must be real). This Fourier representation allows us to rewrite the solution of $u(x, t)$ defined by the PDE (2.4) in terms of the Fourier coefficients a_k in (2.5), as the infinite set of ODEs:

$$\dot{a}_k = (q_k^2 - q_k^4) a_k - i \frac{q_k}{2} \sum_{m=-\infty}^{+\infty} a_m a_{k-m}, \quad q_k = \frac{2\pi k}{L} \quad (2.6)$$

where $a_0(t) = 0$ for all t due to Galilean invariance[14].

This representation allows for solutions to the truncated version of the Fourier series (2.5) to be found. For the system size under consideration ($L = 22$), Cvitanović et al.[14] posited that due to the damping caused by the u_{xxxx} term of Eq. 2.4 (which in turn causes a_k to drop off fast for large k) taking truncations to the Fourier series of between 16 and 128 complex modes gives accurate solutions. Due to the fact that $a_0 = 0$, this can be reduced further to 15 modes; as such, in our below applications we will deal exclusively with the Fourier series of Eq. 2.6 truncated at the 15th mode.

2.3.1 Symmetries

The Kuramoto-Sivashinsky equation (2.4) has both translational and reflective symmetries. Due to the translational symmetry, if $u(x, t)$ is a solution, then so is $u(x + \ell, t)$

for $\ell \in [-\frac{L}{2}, \frac{L}{2}]$; similarly, the reflectional (anti-)symmetry relates $u(x, t)$ to $-u(-x, t)$. In terms of the effect of these symmetries on the Fourier coefficients, the translational symmetry relates the point represented by the complex Fourier coefficients a to the points $g(\ell)a$ for $\ell \in [-\frac{L}{2}, \frac{L}{2}]$ where $g(\ell) = \text{diag}(e^{iq_k \ell})$, corresponding to a rotation of each mode in the complex plane by the angle $\frac{k\ell}{L}$; the reflective symmetry simply relates a to $-a^*$ [14].

These symmetries allow for a more diverse range of coherent structures; in addition to the traditional coherent structures, equilibria and periodic orbits, we also have their symmetry-group related counterparts *relative equilibria*, *relative periodic orbits*, and *pre-periodic orbits*.

Relative equilibria are solutions where $u(x, t)$ is translationally equivalent to $u(x, \tau)$ for all τ , and can also be thought of as travelling waves; due to the reflective symmetry, these relative equilibria come in pairs (travelling in opposite directions). *Relative periodic orbits* are solutions where there exists some τ (the orbit's period) such that $u(x, 0)$ is translationally equivalent to $u(x + \ell, \tau)$, for non-zero ℓ ; it's also possible to have relative periodic orbits with translation *and* reflection, i.e. there exists some τ such that $u(x, 0)$ is equivalent to $-u(-x - \ell, \tau)$, this type of orbit is known as a *pre-periodic orbit*; this name comes from the fact that by repeating this orbit twice we get a relative periodic orbit without reflection, and so it is *pre-periodic* to a relative periodic orbit of period 2τ .

For a more thorough description of these symmetries and coherent structures, see Cvitanović et al.[14].

2.4 Application to the KSE

In order to put this method into practice, one must first have a set of coherent structures for their chosen system. In our case, we used the set of approximately 60,000 coherent structures of the Kuramoto-Sivashinsky Equation found by Cvitanović et al.[14]; this

set contains 3 equilibrium solutions (not including the trivial solution, $u(x, t) = 0$), 2 travelling wave solutions, and relative periodic orbits and pre-periodic orbits ranging from period 10 to 200, all of which are stored up to the 15th complex Fourier mode with step size $h = 0.25$.

As we are dealing with a representation of a continuous-time flow on a computer, we must instead work with a discrete-time approximation of the original flow. In our case, we will be working with coherent structures stored discretely in such a way that after the initial point there's a fixed interval of $h = 0.25$ between each successive time step. As such, in calculations such as the directed Hausdorff distance of Eq. 2.2 which rely on the time parameter t (and τ) these parameters don't in fact lie in the continuous space, but are finitely represented as: $t_0 = 0, t_1 = 0 + h, \dots$.

In practice, the time step $h = 0.25$ is rather large, and as the distance $d(\cdot, \cdot)$ of Eq. 2.2 relies explicitly on this time step, the accuracy of the results is therefore affected. As such, for the sake of accuracy we will in all cases throughout this section apply piecewise cubic Hermite interpolation between each set of $h = 0.25$ spaced points, adding an additional 5 time steps between each set of points (giving a h of effectively $\frac{0.25}{5+1} = 0.0417$); this will be done through the use of MATLAB's built-in `pchip` function.

Typically, trajectories of the KSE are viewed in terms of x , t , and $u(x, t)$; an example of a typical trajectory within the attractor can be seen in Fig. 12 where $x \in [-\frac{L}{2}, \frac{L}{2}]$, $t \in [0, 200]$, and where the colours represent the value of $u(x, t)$ (with blue to red indicating low to high values of u). As we'll be dealing with a symmetry-reduced truncated Fourier representation of the KSE, it will be convenient to instead view trajectories as in Fig. 13 - that is, in terms of the magnitude of the two most expanding Fourier modes (which will be discussed later).

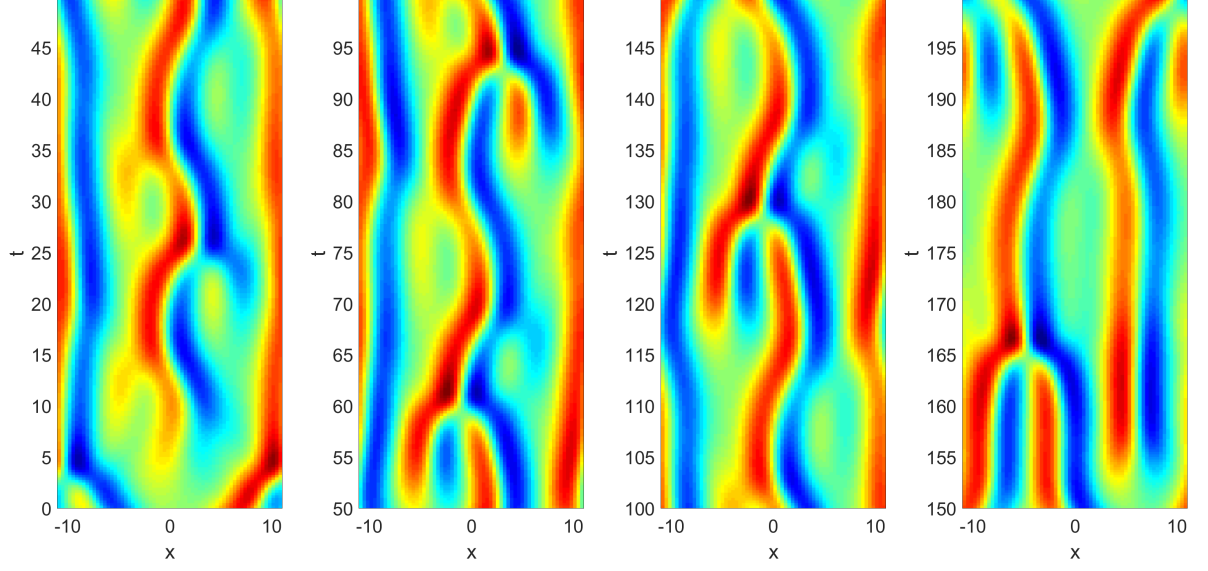


Figure 12: Trajectory segment of duration 200 started from a random point in the attractor, depicted in the full state space. The figures from left to right represent times in the respective intervals $[0, 50]$, $[50, 100]$, $[100, 150]$, and $[150, 200]$.

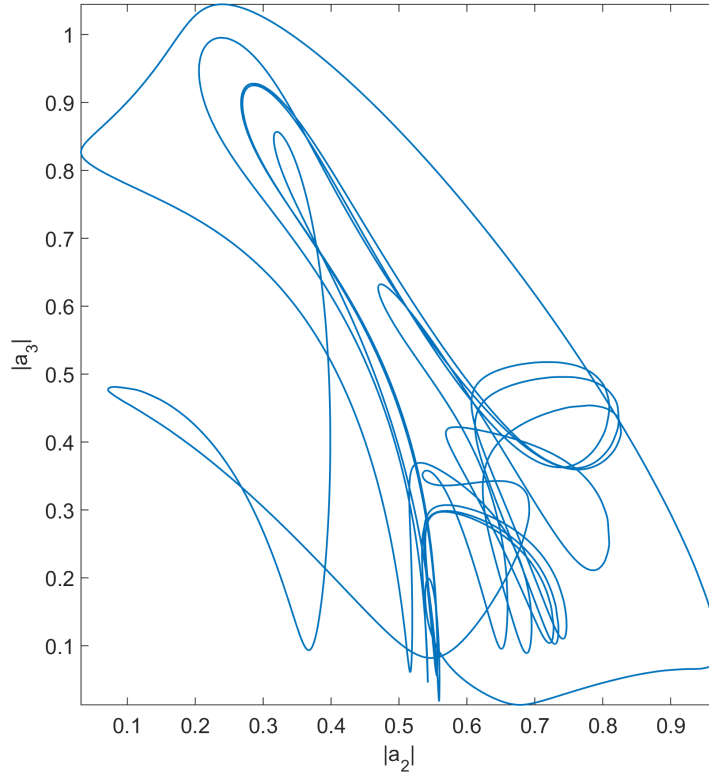


Figure 13: Trajectory segment of duration 200 started from a random point in the attractor (the same point as in Fig. 12), projected in the space spanned by the magnitude of the two leading Fourier modes.

A visualisation of the first 10 coherent structures can be seen in Fig. 14, where the three equilibria are represented by a black cross, the two travelling wave solutions are represented by red asterisks, and the five shortest relative periodic orbits (with periods ranging from approximately 10 to 33) are represented by the different colour lines.

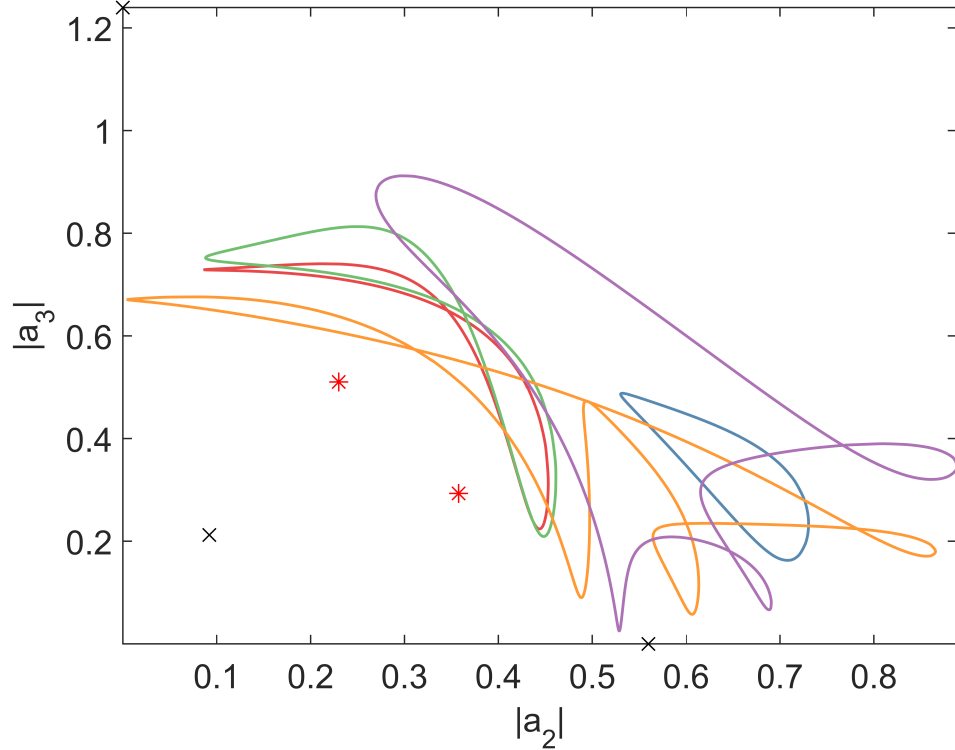


Figure 14: Visualisation of the first 10 coherent structures in 2 dimensions.

2.4.1 Model Reduction

Calculating distances in such high-dimensional systems can quickly become a computationally unwieldy task. As such we will focus a large part of our analysis on how far we can reduce the dimensionality of the system with as little loss of information as possible. In other words, we will explore how well covers constructed in lower dimensional projections cover the ‘full’ 15-dimensional system.

The first projection that we will take of our system is a projection of the points on the

attractor to a subspace in which the symmetries of the system are no longer present. In the context of the KSE, the simplest way that this can be done is by considering only the magnitude of the complex Fourier modes, $|a_k|$; this projection effectively removes the phase information of any given point in the trajectory, and so all points related by the translational and reflective symmetries (which are represented by a complex plane rotation and conjugation respectively) are therefore represented by a single point in the projected space. A side-effect of taking such a projection is that two points with different phases but matching magnitudes will be represented by the same point in the projected space, however this projection reduces the computational complexity of calculating distances in the attractor quite significantly. Had we not taken such a projection, in Eq. 2.2 we would have two additional parameters on the minimisation: a binary reflection parameter, and a translation parameter that varies between $[-\frac{L}{2}, \frac{L}{2}]$ [14]; additionally, all calculations would have to be done in the complex domain, further doubling the dimensionality of calculations.

The other projection that we will make is a projection of the space from 15-dimensions to a lower-dimensional subspace. In order to decide which dimensions an n -dimensional projection should include, we determined the size of the chaotic attractor along each of the Fourier modes by measuring the size of each mode, $m_k = \max_t |a_k(t)|$, on a long ergodic trajectory in the attractor, $t \in [0, 10^7]$. From Table 2 we can see that for a 2-dimensional projection we should use $k = 2, 3$; for a 3-dimensional projection, $k = 2, 3, 4$; and for a 4-dimensional projection or above simply take the first n modes. It's worth noting that the reason that the m_k decay so quickly for larger k is due to the damping effect of the u_{xxxx} term of the KSE (Eq. 2.4).

To counteract the significant loss of information caused by the two above projections throughout this section we will exclusively make use of the Hausdorff distance between segments of orbits. By focusing on the distance between segments of orbits rather than between individual points, due to the sensitivity to initial conditions of chaotic systems if two segments stay close for a long time in any projection, then they must indeed be

k	1	2	3	4	5	6	7	8
m_k	0.5714	0.9807	1.2386	0.5786	0.3264	0.2238	0.0763	0.0357
k	9	10	11	12	13	14	15	
m_k	0.0191	0.0063	0.0027	0.0013	0.0004	0.0002	0.0001	

Table 2: Size of the attractor along each Fourier mode.

close in the full phase space. By using the Hausdorff distance we can get a faithful covering of the whole phase space with even low-dimensional projections of the system, as our following analysis will show.

In order to both accommodate and distinguish between different n -dimensional projections of the attractor in the construction of the cover, we will modify Eq. 2.1 & 2.2 as follows:

For $i = 1, 2, \dots, N_P$:

$$W_n \leftarrow \begin{cases} W_n \cup \{p_i\} & \text{if } d_n(p_i, W_n) > \varepsilon \\ W_n & \text{if } d_n(p_i, W_n) \leq \varepsilon \end{cases} \quad (2.7)$$

$$d_n(p_i, W_n) = \max_{t \in [0, T^{p_i}]} \min_{\substack{w_j \in W \\ \tau \in [0, T^{w_j}]}} \|p_i(t) - w_j(\tau)\|_n \quad (2.8)$$

where $\|\cdot\|_n$ is the Euclidean norm in the n -dimensional projection, and W_n corresponds to the cover constructed in the n -dimensional projection.

Before constructing the cover for the KSE, there is one final parameter to decide upon - the distance threshold, ε . ε can be thought of as the *resolution* of the cover, with lower ε values giving a higher resolution, which results in a more accurate cover containing more cover elements; while the choice of this parameter is arbitrary, we opted to adopt a consistent way of choosing ε based upon the ‘size’ of the n -dimensional projection of the attractor; this approach will allow us to more easily compare the covers obtained with various attractors. In the n -dimensional projection, we will choose ε_n to be 1% of the length of the main diagonal of the hypercuboid spanned by the n largest modes k_j ,

n	2	3	4	5	6	7	8
ε_n	1.5798	1.6825	1.7769	1.8066	1.8204	1.8220	1.8224
n	9	10	11	12	13	14	15
ε_n	1.8225	1.8225	1.8225	1.8225	1.8225	1.8225	1.8225

Table 3: ε_n

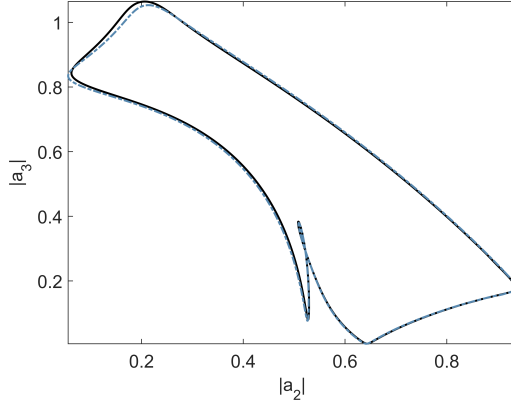
where $k_j = 3, 2, 4, 1, 5, 6, \dots, 15$ for $j = 1, \dots, n$. In other words,

$$\varepsilon_n = 0.01 \sqrt{\sum_{j=1}^n m_{k_j}^2} \quad (2.9)$$

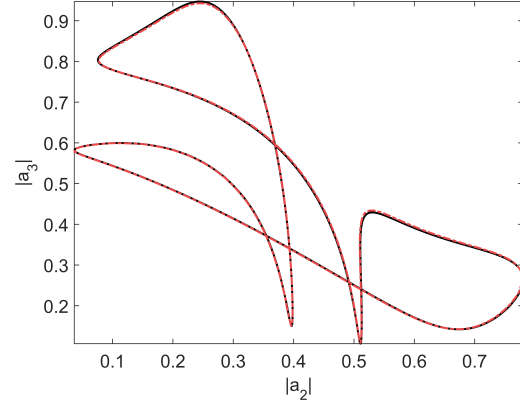
The values of ε_n for $n = 2, \dots, 15$ can be seen in Table 3 (rounded to 4 d.p.), we exclude the case where $n = 1$ as it makes little sense to represent trajectories in anything less than 2 dimensions. After $n = 7$ we see very little increase in ε_n (on the order of 10^{-4} from $7 \rightarrow 8$, decreasing to 10^{-9} from $14 \rightarrow 15$), whereas the computational requirements to compute these covers increases significantly with each added dimension, as such we will focus the following analysis on $n = 2, \dots, 7$.

Having applied the algorithm described in Eq. 2.7 & 2.8 for $n = 2, \dots, 7$, we obtain six different covers, $W_{2,\dots,7}$, containing 91, 513, 747, 809, 825, and 830 coherent structures respectively; this means that by coarse graining the attractor with a resolution parameter of 1% we can approximate every coherent structure in our set using a subset containing only approximately 0.15-1.38% of the original coherent structures (depending on n). An example of how coherent structures not in the cover can be approximated by those in it can be seen in Fig. 15a-15d where the first 4 coherent structures **not** included in W_2 are shown, along with their counterparts **in** W_2 to whom they lie wholly within ε_2 -distance of. This is a notion known as *shadowing*, which will be fully discussed later; for now it suffices to say that not all coherent structures can be represented by **a single** coherent structure in the cover. In fact, it's rather peculiar that such pairs with almost identical trajectories (in the projection) exist, and while outside the scope of this thesis an investigation into these pairs may reveal interesting results.

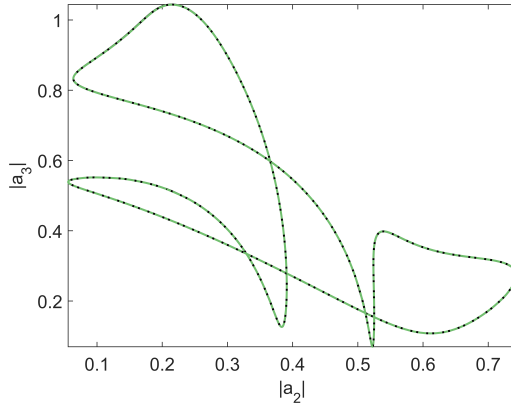
Before discussing this further we must address the issue of the dimensionality of projection, and try to understand in the case of the KSE how many dimensions are required to cover the *full* 15-dimensional system with sufficient accuracy.



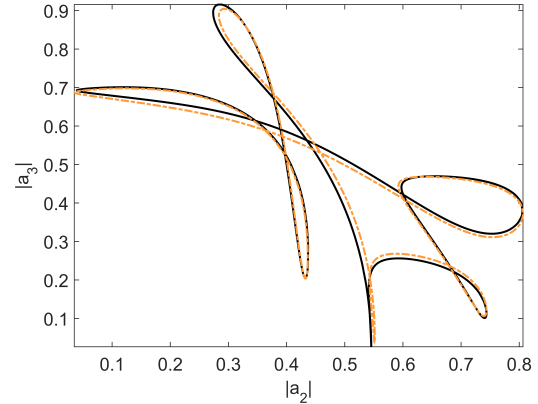
(a) p_{18} (black) & p_{17} (dashed, blue).



(b) p_{20} (black) & p_{19} (dashed, red).



(c) p_{27} (black) & p_{26} (dashed, green).



(d) p_{30} (black) & p_{28} (dashed, orange).

Figure 15: The first 4 coherent structures not included in the set W_2 , represented by solid, black lines; and the structures in W_2 that're within ε_2 distance of them, represented by dashed, coloured lines.

2.4.2 Dimensionality of Projection

In order to gauge how well the 15-dimensional attractor is covered by the covers $W_{2,\dots,7}$, created using lower-dimensional projections of the attractor, we can check how well each

subset $P \setminus W_n$ is covered by W_n in the 15-dimensional space. In other words, we want to know how many structures $p_i \in P \setminus W_n$ satisfy:

$$d_{15}(p_i, W_n) \leq \varepsilon_{15}$$

This approach can be seen as finding out how many of these coherent structures spend 100% of their time within ε_{15} distance of each cover; in order to see more detail about how these covers perform, we will extend this approach to see how many coherent structures spend varying fractions, $m \leq 1$, of their time within ε_{15} distance of their respective covers. More rigorously, we want to know for some m how many coherent structures $p_i \in P \setminus W_n$

$$\exists \tilde{p}_i \subseteq p_i \text{ with } T^{\tilde{p}_i} \geq mT^{p_i} \text{ such that } d_{15}(\tilde{p}_i, W_n) \leq \varepsilon_{15}. \quad (2.10)$$

Intuitively speaking, if for some n a coherent structure p_k satisfies the above for $m = 0.5$, then p_k must spend at least 50% of its period within ε_{15} distance of W_n .

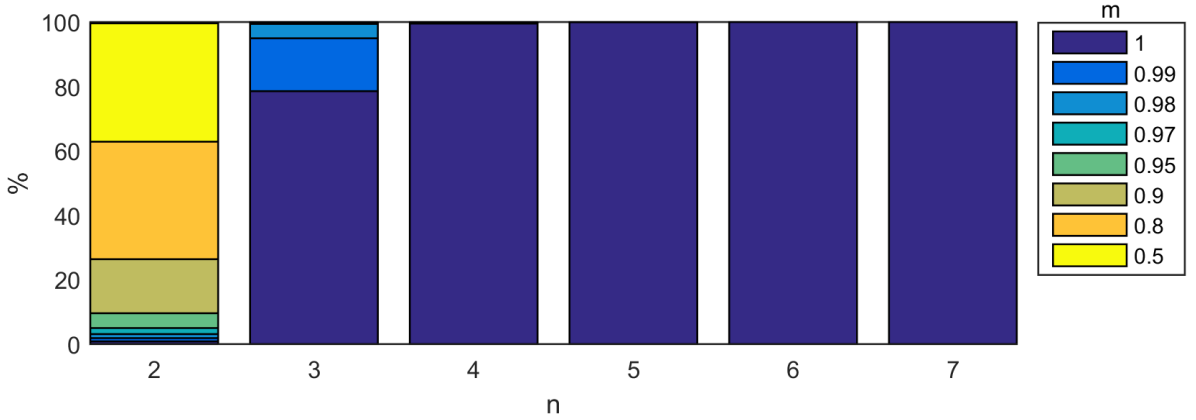


Figure 16: Percent of structures in $P \setminus W_n$ that satisfy Eq. 2.10 for various m (in legend), for each n .

The results of applying this process for each n can be seen in Fig. 16 where we calculated what percent of coherent structures in $P \setminus W_n$ satisfy Eq. 2.10 for various m (represented by different colours, as described in legend). For $n = 2$, only 0.84% of coherent structures in $P \setminus W_2$ lie wholly within ε_{15} distance of W_2 , slowly increasing to only 26.36% of structures satisfying $m = 0.9$. When compared to the case of $n = 3$,

$n = 2$ is found wanting; 78.54% of structures are wholly within ε_{15} distance of W_3 , 94.96% satisfy $m = 0.99$, increasing steadily to 100% for $m = 0.9$. While this looks promising, $n = 4$ offers the last significant improvement, with 99.52% of structures in $P \setminus W_4$ satisfy the case $m = 1$, increasing to 100% for $m = 0.99$. By the time we reach $n = 7$ 100% of structures satisfy $m = 1$, however it seems unlikely that this minor increase in accuracy is worth the extra computational cost of including three additional dimensions in the projection.

From this, it seems reasonable to assume that by taking up to $n = 4$ we will be able to construct a cover that faithfully covers the full 15-dimensional space. As such, the next section will focus on the ways we can make use of such covers, with examples given in the context of W_4 .

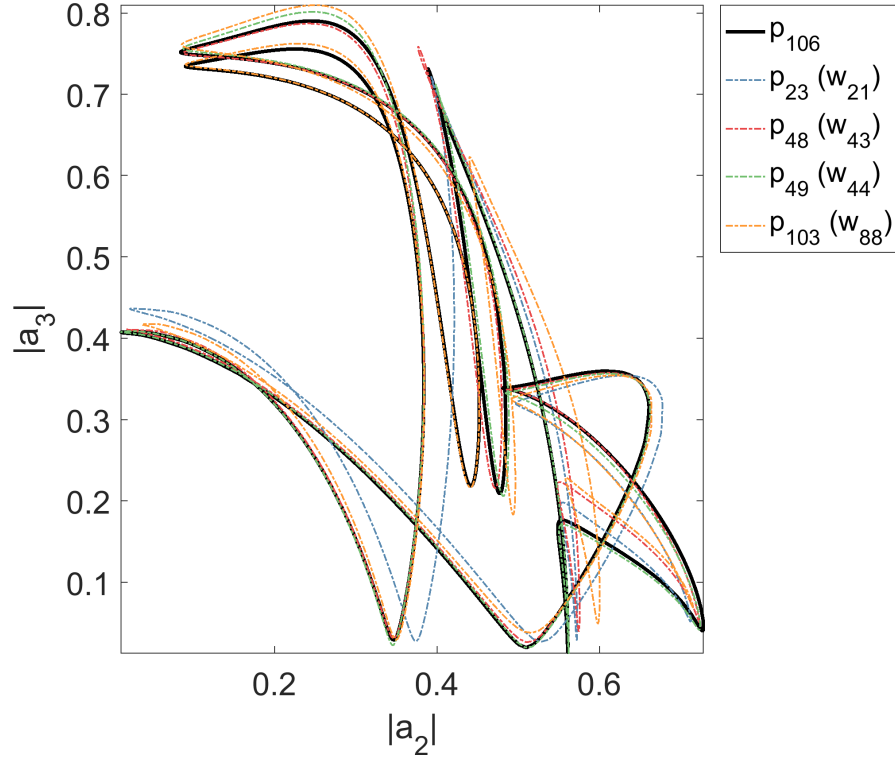
2.5 Shadowing

We saw in Fig. 15a-15d that it's possible to approximate structures in P by those in W_n with ε_n resolution. By construction, it's possible to represent **any** coherent structure in P with ε_n resolution by a subset of structures from W_n ; in other words,

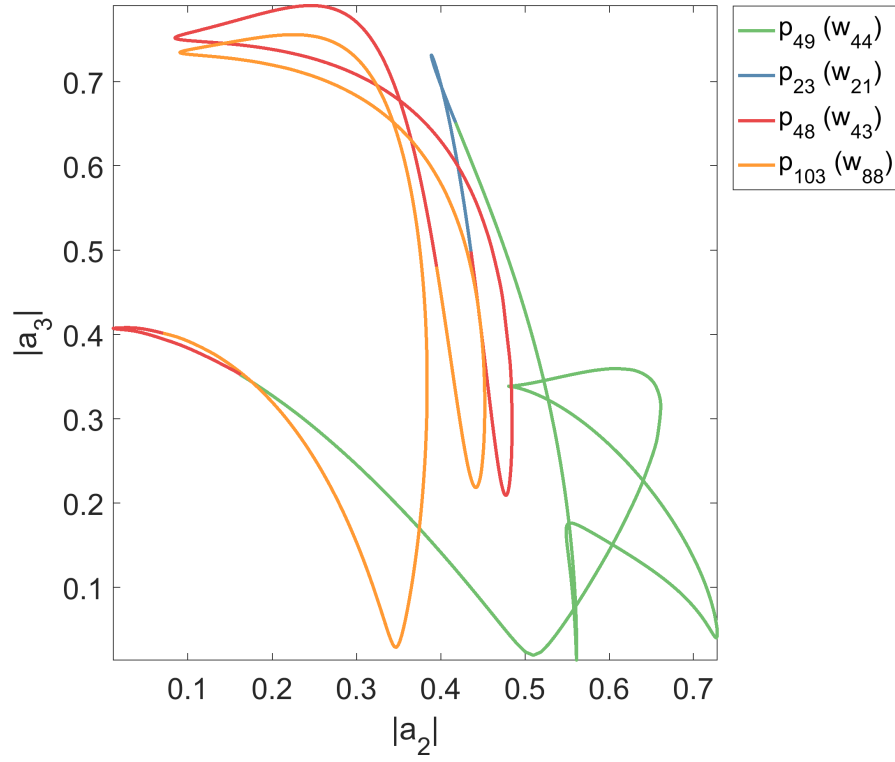
$$\forall p_i \in P \exists \tilde{W} \subseteq W : d_n(p_i, \tilde{W}) \leq \varepsilon_n$$

In the case of the RPOs in Figs. 15a-15d these subsets consisted of only one structure each from W_2 . In a typical case, however, \tilde{W} will contain several structures; an example of which can be seen in Fig. 17a where 4 coherent structures from W_4 are required to fully approximate the whole trajectory of p_{106} with ε_4 resolution.

Such an approximation isn't limited to structures within P , one can approximate any ergodic trajectory in the attractor in much the same way; however, while structures in P are by construction within ε distance of the cover W_n , for general trajectories in the attractor this may not be true. Here we will use the assumption made in section 1 that all trajectories should lie within ε distance of P - meaning that they lie within 2ε distance of W_n .



(a) Structures in W_4 (dashed, coloured) required to approximate p_{106} (black, solid).



(b) Encoding of p_{106} by coherent structures in W_4 .

Figure 17: Two visualisations of the approximation of p_{106} by coherent structures in W_4 .

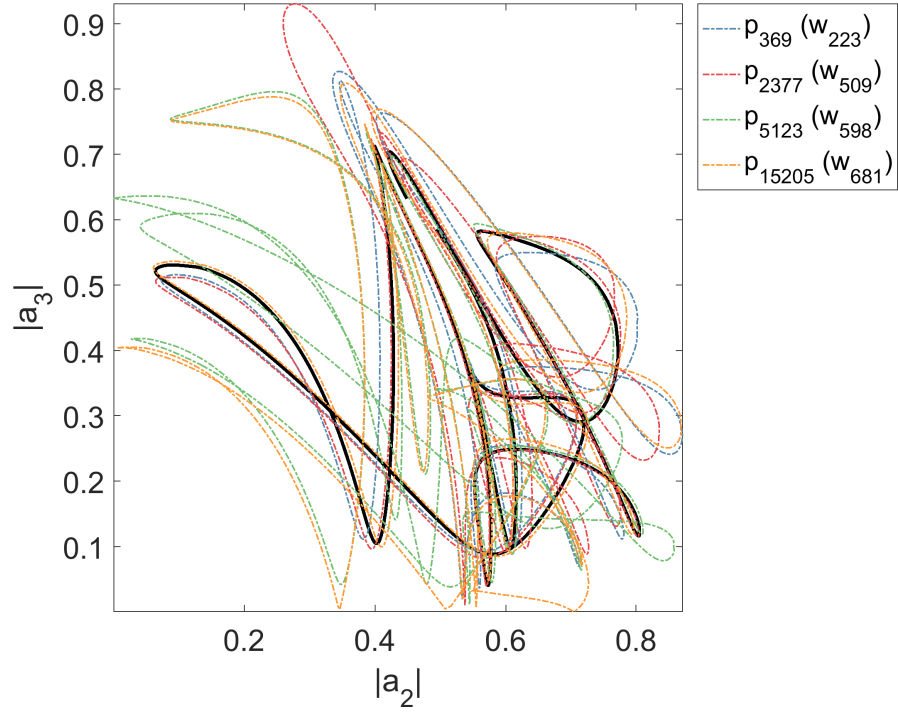
An example of the shadowing of a typical trajectory in the attractor can be seen in Fig. 18a, where a segment of the trajectory of length 60 requires a subset of only 2 coherent structures from W_4 to be fully covered with $2\varepsilon_4$ resolution. As such, when performing shadowing on trajectories not in P we will use instead the resolution parameter 2ε .

The next logical progression from this kind of approach is to label specifically which points on any given trajectory are approximated by which coherent structures in the cover; this notion, known as *shadowing*, allows us to reduce the dynamics of any trajectory on the attractor to a series of transitions between cover elements (a notion closely related to *symbolic dynamics*, the focus of the second part of this thesis). Examples of such shadowing can be seen in Fig. 17b & 18b, where the segments of the black trajectories from Fig. 17a & 18a are encoded by the appropriate elements of W_4 (the algorithm for the construction of such encoding will be detailed in the coming paragraphs, for now I present these examples purely for motivational purposes).

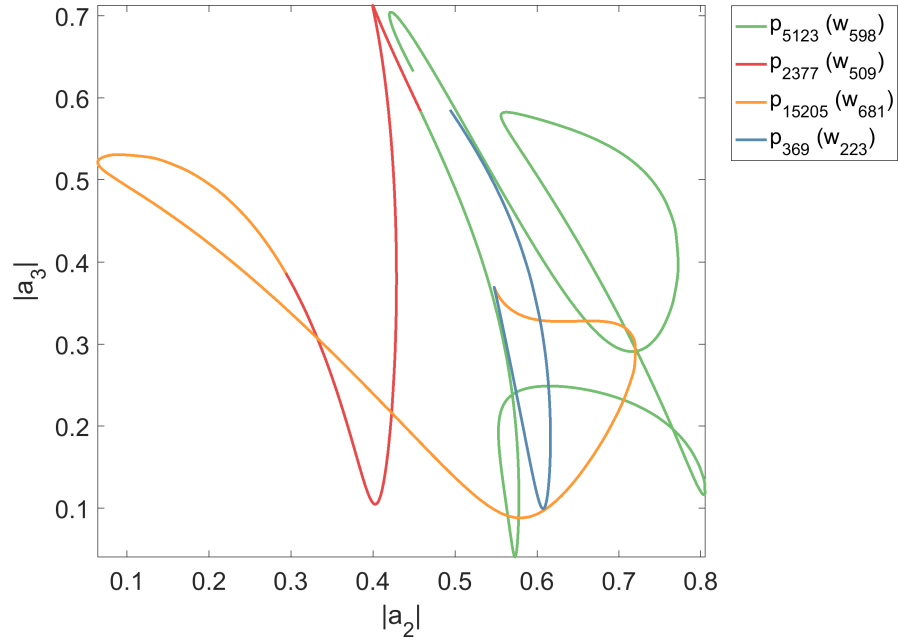
As any given point on a trajectory may have multiple elements of W_n within $2\varepsilon_n$ distance (due to some redundancies in the cover, which will be fully discussed in the following sections), different shadowing algorithms may give different results. Possible shadowing algorithms include:

1. For each point on the trajectory, choose the **closest** coherent structure in W to encode that point.
2. Starting from a given point on a trajectory, choose the coherent structure in W that stays within ε -distance of this point for the **longest** to encode the segment; repeat this process with the point immediately after this segment taken as the new starting point until the whole trajectory is encoded.
3. Given a trajectory, find the configuration of encodings that results in the least number of segments/transitions (all of which must be within 2ε -distance of W).

Following our earlier mantra that in lower-dimensional projections segments that stay



(a) Structures in W_4 (dashed, coloured) required to approximate the trajectory (black, solid).



(b) Encoding of trajectory by coherent structures in W_4 .

Figure 18: Two visualisations of the approximation of a trajectory of duration 60 (started from a random point in the attractor) by coherent structures in W_4 .

close for longer are more likely to be close in the full space, as well as the intuition that when shadowing trajectories it's more desirable to have fewer, longer segments, algorithms 2 & 3 seem to be the better choices. By comparison to algorithm 2, algorithm 1 often results in many short shadowing segments (or many transitions), which for lower dimensional projections results in poor shadowing in the full space. The results of algorithm 1 are also more sensitive to the specific choice of metric, which goes against the fundamental idea of our approach that trajectories that are close in the full state space will remain close for a relatively long time in any reasonable metric.

A side-by-side visualisation of the differences between algorithms 1 & 2 can be seen in Fig. 19a & 19b respectively, where we see that the segment-based shadowing (algorithm 2) encodes the whole trajectory of p_{113} using only 2 segments, whereas for the same trajectory the point-based shadowing (algorithm 1) encodes the trajectory with 23 short segments of 10 different elements of W_4 .

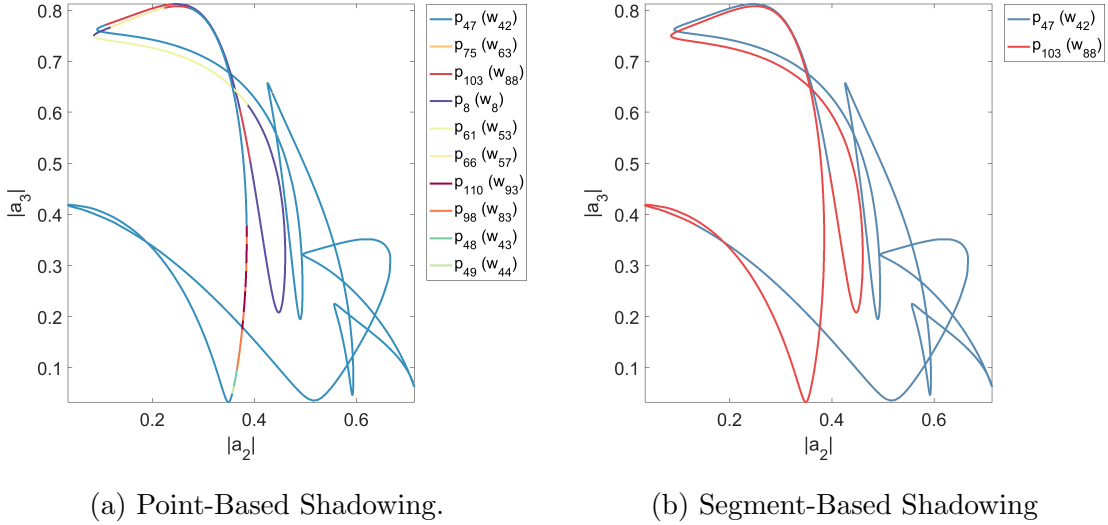


Figure 19: Encoding of p_{113} by coherent structures in W_4 as a result of shadowing algorithm 1 (left) and 2 (right).

To give a rough idea of the difference in sensitivity of each method to the choice of metric, we can look at the number of segments required in order to shadow p_{113} for both methods. In the case of the point-based shadowing algorithm, for $n = 2, \dots, 7$ the

number of required segments is: 593, 60, 25, 18, 18, 18.

By comparison, for the segment-based shadowing algorithm the number of segments required is: 3, 3, 2, 2, 2, 2. Additionally, in the case of this algorithm, the coherent structures used remain relatively constant for each n .

By comparison to both of these, algorithm 3 would be the ideal solution that combines the best parts of each of the two previous algorithms: the independence of solution to starting point of algorithm 1, and the longer segments and more intuitive results of algorithm 2. However, due to the extra computational effort required to compute such a solution (especially for longer trajectories) we will leave this as an avenue for further research, and for now focus on algorithm 2. Since the second algorithm starts from a given point (rather than optimising over the whole set), and takes the cover element that stays close for the longest, we will refer to this algorithm as both *naïve* and *greedy*.

2.5.1 Shadowing Algorithm

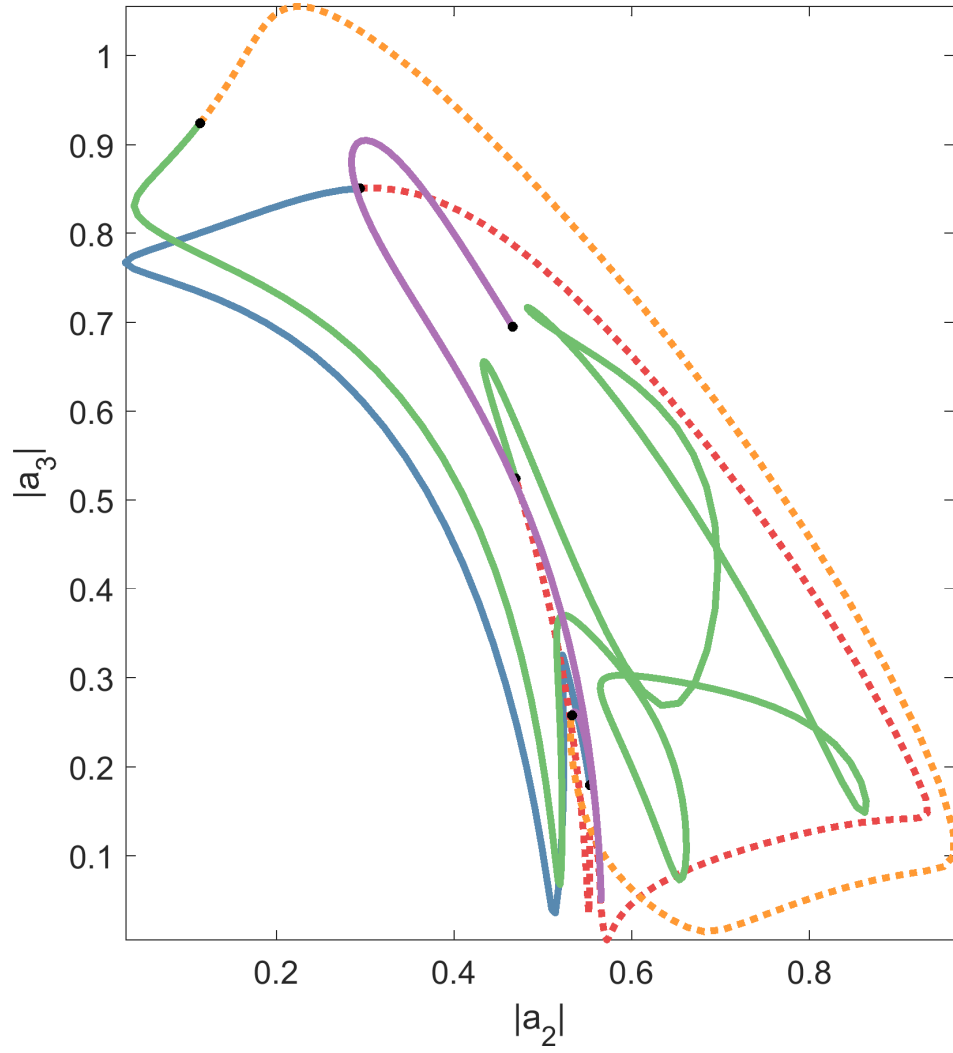
The algorithm for the greedy naïve segment-based shadowing of a trajectory by coherent structures in W_n with step size h is as follows:

1. Starting from $\mathbf{a}(0)$, find $W_n \supseteq \tilde{W}_0 = \{w \in W_n \mid d_n(\mathbf{a}(0), w) \leq 2\varepsilon_n\}$.
2. Repeat this, finding $\tilde{W}_{i-1} \supseteq \tilde{W}_i = \{w \in \tilde{W}_{i-1} \mid d_n(\mathbf{a}(0 + ih), w) \leq 2\varepsilon_n\}$.
3. When k is found such that $\tilde{W}_k = \emptyset$, label the segment $\{\mathbf{a}(t), t \in [0, 0 + (k - 1)h]\}$ by the index j of $w_j \in \tilde{W}_{k-1}$. In the case that \tilde{W}_{k-1} contains multiple such w , choose the least complex coherent structure (i.e. the lowest period).
4. Repeat this process from step 1, starting from $\mathbf{a}(kh)$, until the whole trajectory is labelled.

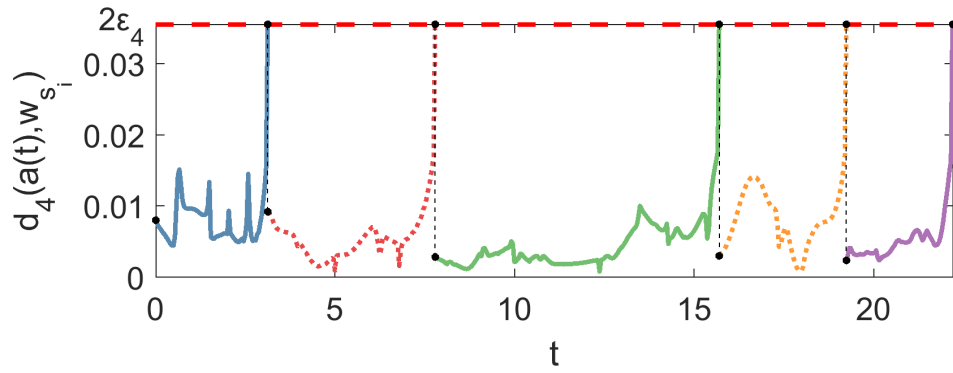
Intuitively speaking, starting from a given point, this algorithm finds the coherent structure in W_n that stays within $2\varepsilon_n$ distance of this point for the longest, encoding the segment by that structure's index. This process is then repeated until the whole trajectory is covered. The result is an itinerary of visits that the trajectory makes to structures

within W_n . An example of this is shown in Fig. 20a where a trajectory with $t \in [0, 22.21]$ starting from a random point in the attractor is encoded by 5 different structures in W_4 , giving the trajectory: $w_8(3.125) \rightarrow w_{237}(4.67) \rightarrow w_{242}(7.92) \rightarrow w_{70}(3.54) \rightarrow w_{319}(2.96)$, where the number in brackets corresponds to the time spent at that coherent structure before moving to the next - also known as *residence time*.

Fig. 20b gives a visualisation of the inner workings of the algorithm; starting from w_8 , the trajectory stays within $2\varepsilon_4$ distance of the w_8 until $t = 3.125$, at which point $d_4(\mathbf{a}(3.125), w_8) > 2\varepsilon_4$, and so this shadowing segment is finalised (and can be seen in Fig. 20a as the solid blue line segment). We then move onto w_{237} which stays within $2\varepsilon_4$ distance until $t = 7.78$, and so on.



(a) Encoding of a trajectory by W_4 .



(b) Distance from $\mathbf{a}(t)$ to the chosen structure in W_4 . These structures are: w_8 , w_{237} , w_{242} , w_{70} , and w_{319} respectively.

Figure 20: Encoding of a trajectory of duration 22.21 by W_4 , along with the distances of each point of the trajectory to the encoded coherent structure in W_4 . The different coloured and styled segments from (b) correspond to their similarly styled counterparts in (a).

2.6 Markov Chain Model

By using this shadowing algorithm we can reduce any trajectory in the attractor to a relatively small set of transitions and accompanying residence times. As in the first part of the thesis, using these transitions and residence times it's possible to model the data by a Markov chain, the only difference being that now that we're dealing with continuous in time flows we'll instead use a finite-state continuous-time Markov Chain (CTMC).

2.6.1 Construction

Construction of a finite-state continuous-time Markov chain is slightly more involved than its discrete-time counterpart. In order to construct the transition probability matrix, one must first construct the transition *rate* matrix (also known as the *generator*) of the Markov chain. The transition rate matrix \mathbf{Q} takes the form:

$$\mathbf{Q} = \begin{bmatrix} -q_1 & q_{12} & q_{13} & \cdots & q_{1n} \\ q_{21} & -q_2 & q_{23} & \cdots & q_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & \cdots & -q_n \end{bmatrix} \quad (2.11)$$

where n is the number of states.

In the 2007 review paper *Generator estimation of Markov jump processes* by Metzner et al.[41] two key methods for estimation of the generator of continuous-time Markov chains using transition data are reviewed.

The first, known as the *Quadratic optimisation* method, was first introduced by Crommelin & Vanden-Eijnden[42]. This method involves calculating an estimate for the generator through the solution a quadratic minimisation problem using the fact that many properties of a CTMC depend strongly on the eigenvalues and eigenvectors of its generator[41]. One of the drawbacks of this method, however, is that if the number of states d of the system is too high to store the tensor $H \in \mathbb{R}^{d \times d \times d \times d}$ then the problem

must first be reformulated for use with (for example) MATLAB's `quadprog` as shown in [43].

Alternatively, there is the *Expectation maximisation* method introduced by Holmes & Rubin[44] which involves using data to find an estimate of the generator such that the given continuous log-likelihood function is maximised. While Holmes & Rubin first introduced this method as a tool for estimating a generator for continuous-time observations, Bladt & Sørensen[45] later extended the method to allow for the construction of an approximate generator using data observed at discrete time intervals (as we have).

Metzner et al.'s[41] analysis concluded that in most cases, the difference between the Quadratic programming and Expectation maximisation methods is negligible, with a slight edge given to the Quadratic programming method. Were we to use the Quadratic programming method however, due to our number of states being far too high to store the tensor H (which would require storing a $747 \times 747 \times 747 \times 747$ variable, requiring approximately 2.5TB of memory), we would have to reformulate the problem as described in [43] to not require H to be stored explicitly. As such, for this example we decided to use the Expectation maximisation method for the construction of the approximate generator, due to the simplicity of construction; note, however, that for future research purposes in this direction, outside of this example, we will endeavour to more accurately analyse the differences between these methods (and others) and choose which to use accordingly.

Using this method, construction of the transition rate matrix \mathbf{Q} uses the maximum likelihood estimates of q_{ij} defined as $\hat{q}_{ij} = \frac{N_{ij}}{R_i}$, where N_{ij} is the number of transitions from state i to j , and R_i is the total time spent in state i [45]. The diagonal elements q_i are then defined as the sum of all non-diagonal elements of each row, i.e. $\sum_{i \neq j} q_{ij}$. Note that due to the negative symbol on the diagonal of \mathbf{Q} of Eq. 2.11, each row sums to 0.

Having calculated \mathbf{Q} , the transition probability matrix \mathbf{A} can be computed using the matrix exponential:

$$\mathbf{A}^t = e^{\mathbf{Q}t} \quad (2.12)$$

where \mathbf{A}_{ij}^t corresponds to the probability of transitioning from state i to j after t time[45]. Note that when we refer to simply \mathbf{A} below, we are referring to the transition probability matrix after unit time \mathbf{A}^1 .

In order to construct the transition probability matrix \mathbf{A} for our system, we first performed shadowing on a trajectory of duration approximately 10^{10} using W_4 . By applying the above procedure, the result is the 747×747 matrix \mathbf{A} .

2.6.2 Properties

As mentioned in Section 1.5, in order for a jump process to be considered Markovian, the Markov property must be fulfilled. This property stipulates that the probability of jumping from state i to state j depends **only** upon the current state, and not on any previous states; as such, this property is also known as *memorylessness*. As a consequence of this property, the frequency of waiting times of the system must follow a distribution that possesses the memoryless property. In the case of continuous probability distributions, the only distribution that has this property is the exponential distribution[30].

In order to check whether our data can suitably be represented by a Markov chain model, we must therefore first check that the transition-based representation of the dynamics that we've constructed through the use of the cover and shadowing algorithms result in waiting times whose frequency distribution is at least almost of exponential type[28, 29].

Fig. 21 shows the distribution of waiting times, with a log scale on the horizontal axis. As in the case of the attractor of the Ikeda map in the previous part of the thesis, due in part to the greedy nature of the shadowing algorithm (preferring longer

waiting times), but mainly due to the fact that the KSE isn't a stochastic system but a deterministic chaotic system, the exponential distribution of waiting times doesn't become apparent until longer waiting times. As explained previously, this is due to the combination of rounding error and sensitivity to initial conditions causing the system to effectively lose memory after a certain time. As these short segments constitute a relatively small portion of the total time (around 13%), we can proceed relatively safely under the assumption that our data can suitably be represented by a Markov model on longer time scales.

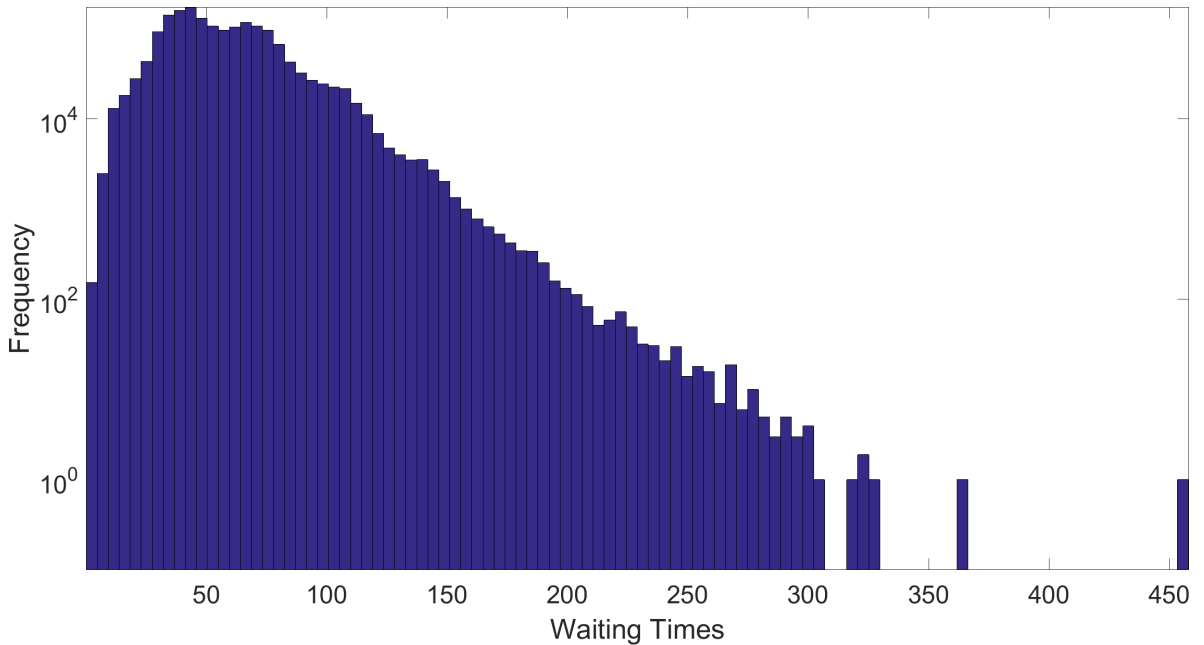


Figure 21: Log-Scaled Frequency of Waiting Times.

In order to calculate measures such as the stationary distribution of the Markov chain, we must first ensure that the chain is *ergodic* - meaning that it's possible to reach any state from any other state within a finite time[34]. By analysing the waiting times, it became clear that cover elements 1-8 were not used at all during the shadowing of the long trajectory. In the case of the equilibrium (states 1,2 & 3) and travelling wave solutions (states 4 & 5), this is due to the fact that they lie on the outskirts of the attractor - and so they are very seldom visited, and only for very short times. In the case of the relative periodic orbits constituting states 6, 7, & 8, all points on these orbits are completely covered by points from other, longer orbits; as such, the shadow-

ing algorithm chooses the longer orbits in place of these shorter ones, as the trajectory follows these longer orbits for more time.

By removing states 1-8 from the Markov model we reduce \mathbf{A} to a 739×739 matrix, representing an *ergodic* Markov chain.

One example of the potential applications of the Markov chain model is ranking cover elements in order of importance using the *stationary distribution* of the Markov chain.

2.6.3 Stationary Distribution

In order to calculate the stationary distribution of the Markov chain, we will follow precisely the same procedure as in section 1.5.3; that is, by finding the row vector π satisfying $\pi\mathbf{A} = \pi$. We do this by finding the left eigenvector of \mathbf{A} corresponding to eigenvalue 1 using MATLAB's `eig` command (and then rescaling the vector to sum to 1).

The stationary distribution π gives the probability of being in each state at any given time, and so by analysing these probabilities we can determine which states are most and least important to the dynamics. Fig. 22 shows the probabilities π_i for each cover element i as deviations from the mean μ_π , with one standard deviation above and below the mean indicated by the red dotted lines. From this figure we get the sense that under the current shadowing regime the shorter orbits play a generally less important role in the dynamics. The most important cover element is w_{675} , which has the highest π value of around 0.86%; in contrast, the least important element is w_{42} , with a π value of less than 0.00022%.

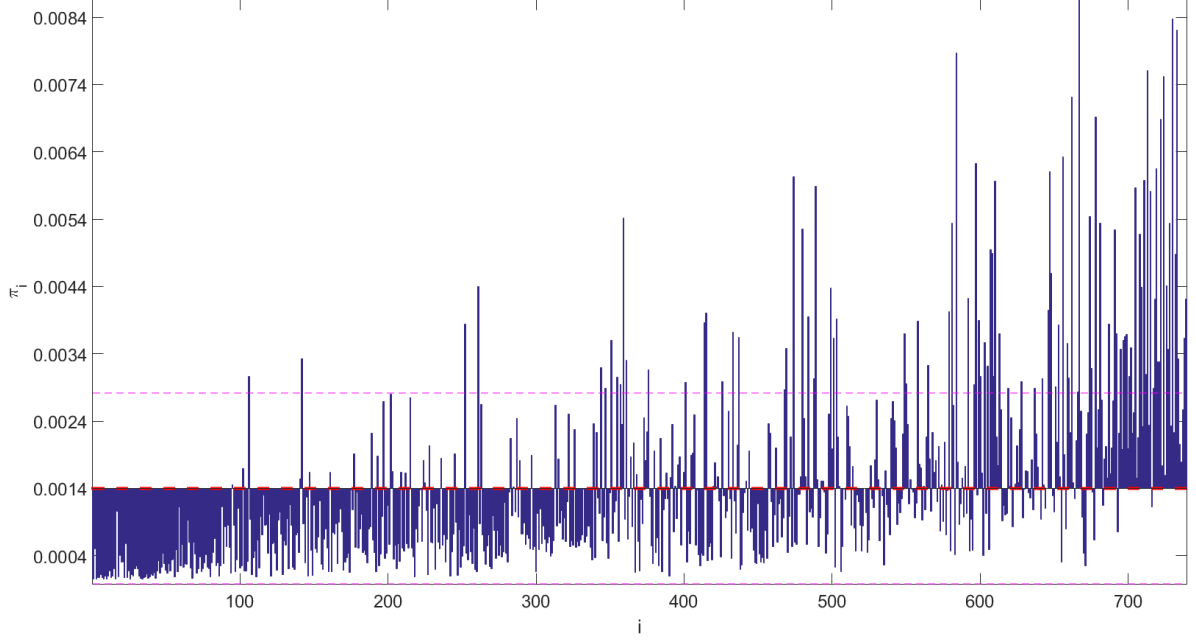


Figure 22: π_i of each cover element, with the middle red line corresponding to the mean μ_π and the outer red lines corresponding to one standard deviation above and below the mean respectively ($\mu_\pi \pm \sigma(\pi)$).

2.7 Minimising the Cover

In section 2.2 we proposed a method by which one can construct a cover of a chaotic attractor using coherent structures. In this section we will discuss and give examples of redundancies present in such a cover, and provide a method by which some of these redundancies can be reduced.

2.7.1 Redundancies

Due to the iterative nature of the construction of the cover proposed in section 2.2, it's possible that a coherent structure that's included into the set early (say, w_i) could be completely covered by one added later (w_j , $j > i$). If w_j completely covers w_i , and explores some additional part of the attractor, then w_i would (by construction) not cover w_j ; in other words, it's possible that:

$$\exists i < j \ (i, j \in [1, |W|]) : d(w_i, w_j) \leq \varepsilon \ \& \ d(w_j, w_i) > \varepsilon \quad (2.13)$$

in which case, the covering provided by $W \setminus w_i$ is the same as that provided by W

as $\varepsilon \rightarrow 0$. A more tangible example of this can be seen in Fig. 23 where (for W_4) w_8 is completely covered by w_{53} , which also explores additional parts of the chaotic attractor. Since $d_4(w_8, w_{53}) \leq \varepsilon_4$, and it's clearly observable that $d_4(w_{53}, w_8) > \varepsilon_4$, w_8 can be considered redundant in that almost no covering would be lost by removing it from the set.

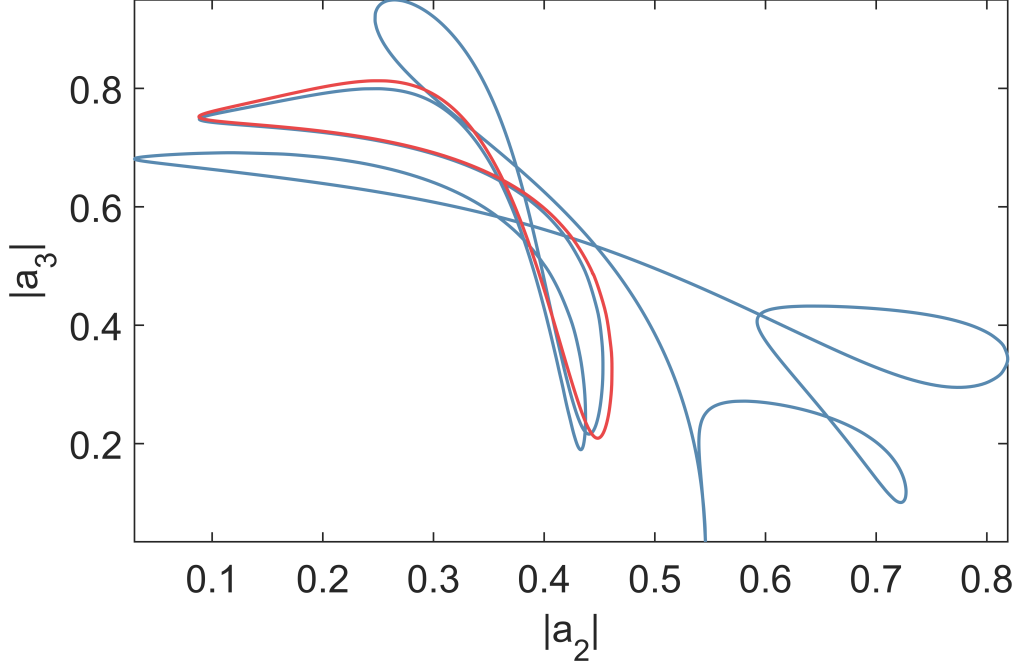


Figure 23: Example of redundancy in W_4 : w_8 (red) is completely within ε_4 distance of w_{53} (blue).

2.7.2 Second Pass

With this in mind we propose an additional stage to the construction of the cover of section 2.2, a second pass, by which such redundancies can be removed from the set. Starting with W as previously constructed, for $i = 1, 2, \dots, |W|$:

$$W \leftarrow \begin{cases} W \setminus w_i & \text{if } d(w_i, W \setminus w_i) \leq \varepsilon \\ W & \text{if } d(w_i, W \setminus w_i) > \varepsilon \end{cases} \quad (2.14)$$

Intuitively, this second pass goes through all coherent structures in W and removes those that have no effect on the covering of the attractor, in a sense *minimising* the cover.

One drawback of this method is that by removing orbits satisfying $d(w_i, W \setminus w_i) \leq \varepsilon$, it's possible that we may lose some of the covering provided by the ε -neighbourhood of w_i . If we denote the original cover W_0 , then the resulting cover after the second pass W satisfies: $B_\varepsilon(W) \supseteq W_0$. This unfortunately means that if we wish to provide the same covering as P we need to take $B_{2\varepsilon}(W) \supseteq B_\varepsilon(W_0) \supseteq P$; further, if we wish to provide the same covering of the attractor as $B_\varepsilon(P)$ (as previously), then we must take a 3ε ball around W : $B_{3\varepsilon}(W) \supseteq B_{2\varepsilon}(W_0) \supseteq B_\varepsilon(P)$. As such, the resolution of the cover is further reduced by applying the second pass.

The result of applying this second pass to our previously constructed $W_{2,\dots,7}$ using the same $\varepsilon_{2,\dots,7}$ as previously used (and as can be seen in Table 3) are new covers $W_{2,\dots,7}$ containing 29, 201, 306, 333, 339, and 342 coherent structures respectively, a significant reduction in size for each cover. In order to test if this second pass had any effect on the covering of $P \setminus W_n$, we repeated the same analysis as earlier (see Eq. 2.10), the results of which can be seen in Fig. 24. For this minimal cover, W_4 still seems like the best choice of cover, however there has been a noticeable loss of covering for each projection. In the case of $n = 4$ far fewer (approx. 75%) coherent structures satisfy $m = 1.0$; all structures in $P \setminus W_4$ do however satisfy the case where $m = 0.98$, meaning that at least 98% of each coherent structure is covered by W_4 in the full space - which is still an acceptable result for the reduction in redundancy we achieve (over 50% less structures in W_4). Using this new minimal cover we can perform precisely the same kind of shadowing shown in section 2.5 with precisely the same algorithm, as such I will forego any explanations or examples. The only difference between shadowing of the old cover and the new minimal cover is that the minimal cover has fewer elements - this fact will be useful when attempting to create a Markov-type representation of the attractor as the transition matrix will be significantly smaller and more meaningful due to the reduction of redundancy.

Unfortunately, this method has an additional drawback: the resulting cover is a little counter-intuitive. Short periodic orbits are often considered to form the fundamental

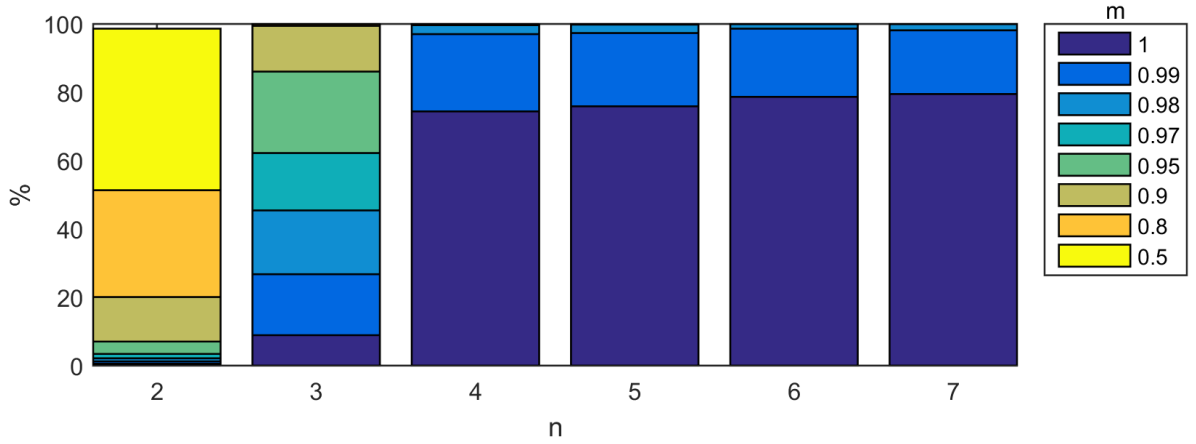


Figure 24: Percent of structures in $P \setminus W_n$ that satisfy Eq. 2.10 for various m (in legend), for each n .

structure of the ‘skeleton of chaos’, with longer periodic orbits giving small refinements to this structure[3]; as such, we would ideally like to represent trajectories in the attractor by shorter coherent structures when possible. This method unfortunately puts more emphasis on longer structures (that contain the shorter structures within them).

In the following section we propose an alternative method of constructing a cover from scratch that removes all redundancy without any of the above outlined issues (loss in accuracy, emphasis on longer structures).

2.8 Segment-Based Cover

In the previously created set W each element of the cover was a coherent structure from the set P . While minimal in the sense that the same covering of the attractor as P is given by a much smaller set of coherent structures, there may be sections of coherent structures in W that overlap with one another, causing redundancy in the set. An example of this can be seen in Fig. 25 where $p_{137}, p_{138} \in W_4$ spend a significant portion of their trajectories within very close proximity of one another (most obviously seen in the top-right of the trajectories).

In order to reduce this redundancy, we will modify algorithm 2.1 to only include into

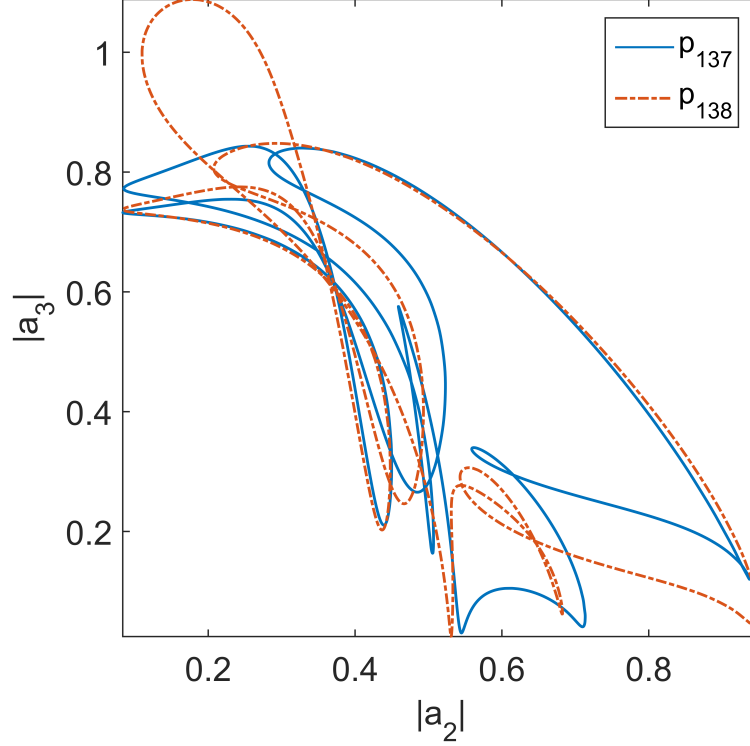


Figure 25: Trajectory of $p_{137}, p_{138} \in W_4$.

the cover the **segments** of the coherent structure that lie outside of ε distance, rather than including the whole coherent structure. This method unfortunately opens up the possibility of our new cover being composed of disjoint singletons from various coherent structures which is undesirable as we're trying to find qualitative characteristics of the system in terms of finding patterns or behaviours that may be dynamically important; as such, we'll add the constraint that in order for a segment to be included in the cover W it must spend at least duration τ outside ε distance of W . Due to the dynamical importance of equilibria and travelling waves, we'll automatically include them into the set, despite the aforementioned constraint.

In order to formulate the new segment-based cover construction algorithm with more rigour, we must first define a segment of a coherent structure of at least length τ . We will define ϕ_j^i , the j^{th} segment of coherent structure p_i , as follows:

$$\phi_j^i = \{a_i(t) \in \{p_i\}, t \in [t_0^{i,j}, t_1^{i,j}] : t_1^{i,j} - t_0^{i,j} \geq \tau\}$$

Within any structure many (let us say N_ϕ) such segments exist, with this number depending on the time step h of the discretisation of the continuous-time dynamics, the period of the structure in question, and the choice of τ ; however, we're only interested in choosing the segments that stay outside ε distance of W in such a way that the points immediately preceding and following the segment are within ε distance of W - in other words, we only want to choose the (connected) portion of the coherent structure that's not covered by W .

Starting with $W = \bigcup_{i=1}^{N_e} p_i$, where N_e is the number of equilibrium & travelling wave solutions, the algorithm is as follows:

For $i = N_e + 1, \dots, N_P$, $j = 1, \dots, N_\phi$:

$$W \leftarrow \begin{cases} W \cup \{\phi_i^j\} & \text{if } \begin{cases} d(\phi_i^j, W) > \varepsilon \\ d(a_i(t_0^{i,j} - h), W) \leq \varepsilon \\ d(a_i(t_1^{i,j} + h), W) \leq \varepsilon \end{cases} \\ W & \text{otherwise} \end{cases} \quad (2.15)$$

For our initial tests of the new algorithm, we used the same 4-dimensional projection as in the previous sections, using the same ε_4 , and we'll arbitrarily choose $\tau = 5$ (roughly half of the duration of the shortest relative periodic orbit). Using these parameters we obtained a cover containing 646 segments from 533 different coherent structures. The total time of the elements of the new segment-based cover is approximately 4,623.5, less than 10% of the 49,177 time-span of the coherent structure-based cover created in the previous sections.

The first example of a coherent structure from P that is only partially included into W using the new algorithm is p_8 ; in Fig. 26 the red segment of p_8 is within ε distance of W (in black), and so only the cyan segment is included into the cover.

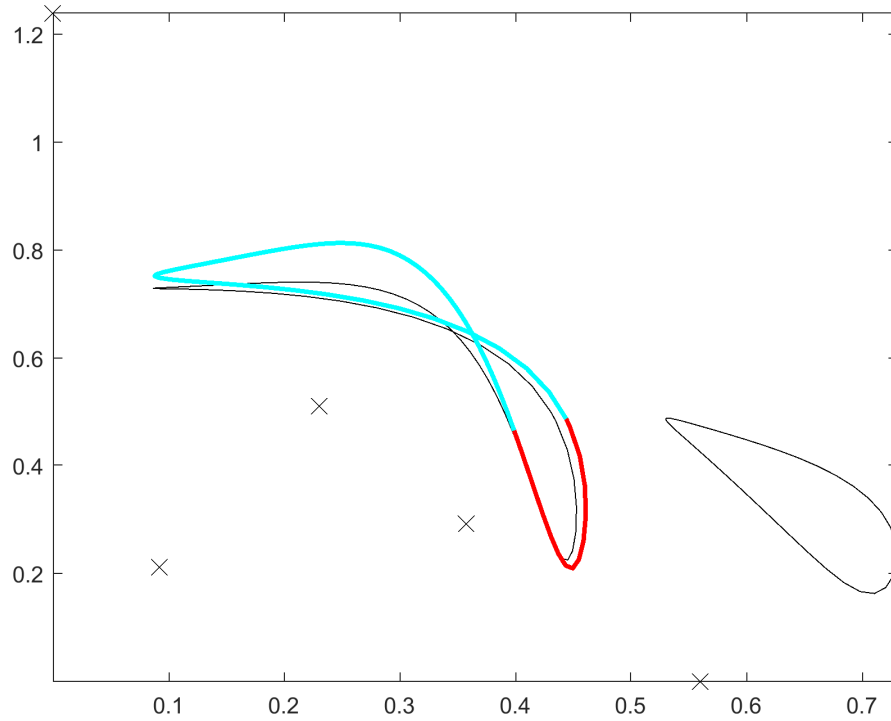


Figure 26: Comparison of p_8 to $p_{1,\dots,7}$, all of which are included in W . Black crosses represent equilibria/travelling waves, black curves represent segments of orbits included in W , the red segment corresponds to the segment of p_8 within ε distance of W , and the cyan segment corresponds to the segment of p_8 outside ε distance of W .

The first coherent structure to not be included in the cover is p_{18} , a relative periodic orbit of period 36.2058. As seen in Fig. 27, all but a few small sections of duration $< \tau$ are within ε distance of $p_{12}, p_{15} \in W$, and so no segments from this orbit are included in W .

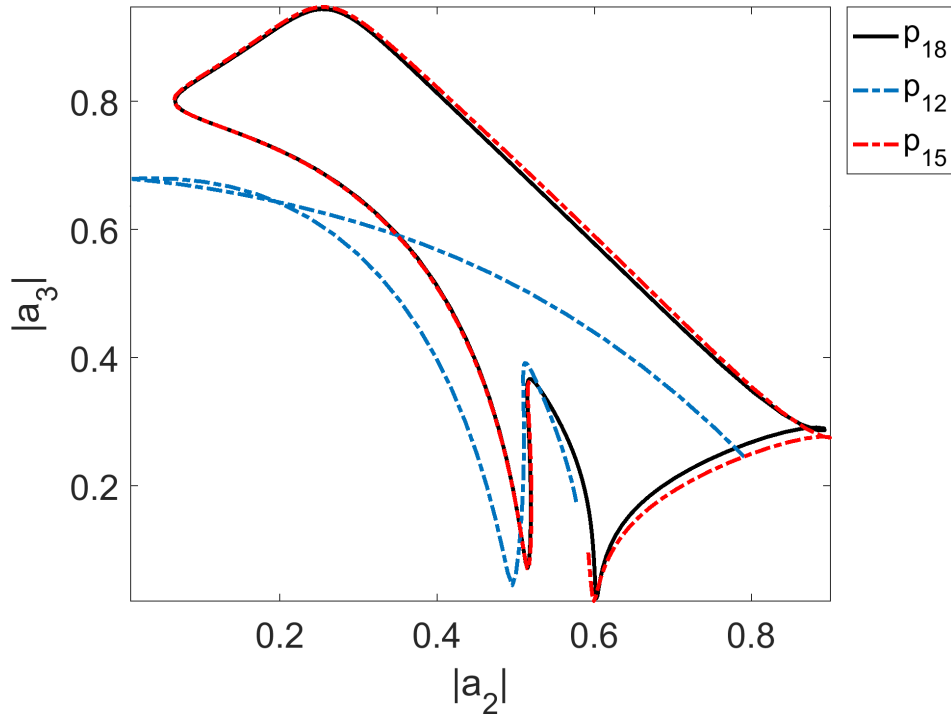


Figure 27: Comparison of p_{18} to $p_{12}, p_{15} \in W$.

Creating a cover in such a way will undoubtedly reduce the amount of points needed to cover the attractor, at the cost of losing the simplicity and intuitiveness of representing parts of the attractor simply by whole coherent structures.

One of the key issues with creating such a cover is the selection of parameter τ - higher values of which lead to more loss of covering of the attractor. A preliminary analysis showed that for a long trajectory, the amount of the trajectory within $2\varepsilon_4$ distance of the covers using $\tau = 0, 1, 2, 3, 4, 5$ was 100%, 100%, 100%, 99.99%, 99.95%, and 99.89% respectively - all still within reasonably acceptable ranges. It however remains to be seen how the covering of this low-dimensional projection will work in the 15-dimensional space.

In order to capture the benefit of higher τ values (the creation of a cover formed by long segments), and of lower τ values (covering of more of the attractor), an algorithm that starts with high τ which then repeats with increasingly lower τ values upon completion could give desirable results. This algorithm would at first try to cover the attractor

with as long segments as possible, and then slowly fill in the attractor with segments of decreasing duration where necessary. This method would provide a nice middle ground between the original method and the segment-based covering in terms of intuitiveness (prioritising long segments over shorter), covering (would cover the whole attractor as in the original method), and redundancy (no redundancy, as with the segment-based covering).

For now, this segment-based cover is provided as a concept to motivate further development of the covering algorithm. The analysis and further development of this method will be left to future research.

2.9 Summary & Future Work

In this part of the thesis we extended the previously presented method for covering chaotic attractors of maps to the case of high-dimensional continuous systems with symmetries.

We presented a method by which both the symmetries and the dimension of the system can be reduced in the construction of the cover, using the Hausdorff distance between trajectories in order to reduce the adverse effects of the dimension and symmetry reductions.

Using the shadowing algorithm introduced in the previous part of the thesis, we shadowed a long trajectory in a Kuramoto-Sivashinsky chaotic attractor using the cover created in the 4-dimensional projection, and used this data to construct a Markov model of the dynamics. As motivation, we gave an example of how the Markov model can be used (the *stationary distribution*).

Finally, we presented two methods by which it's possible to further minimise the cover, reducing redundancies.

Future work will involve improving the selection of lower-dimensional projections of the space. For example, by using dimensionality reduction techniques such as principal component analysis or linear discriminant analysis, rather than the current method of choosing the leading Fourier modes. Other further steps could include exploring different shadowing algorithms and their effects, refining and analysing the segment-based covering algorithm, and further exploring the properties and potential applications of the Markov chain model of the dynamics. Additionally, we hope to find alternative ways to reduce the redundancies in the cover (with fewer drawbacks the second pass algorithm).

3 Construction of approximate generating partitions of chaotic systems by embedded periodic orbits.

Symbolic dynamics is a tool developed for the purpose of analysing dynamical systems through the discretisation of the state space into partitions through which it is possible to identify trajectories in the system not by a set of points, but instead by a sequence of transitions between these partitions (each of which is given a unique label, known as a *symbol*). If a partition is constructed such that each unique trajectory is associated with a unique symbol sequence, then such a partition is known as a *generating partition*[46].

To give an example of how symbolic dynamics and generating partitions can look, Fig. 28 shows the symbolic dynamics given by a generating partition of the Hénon map found by Grassberger et al.[47] in their 1985 paper, and which is also corroborated by our method (which is to be discussed later). Here, the generating partition is given by the boundary between the red and cyan regions which correspond to symbols **0** and **1** respectively, with the black points being the points on the chaotic attractor of the Hénon map for the parameters $a = 1.4$ & $b = 0.3$ (which will be introduced in more detail later). In white are the four points of a period 4 orbit on the attractor, going in order from the point labelled 1 this trajectory would be given the binary symbol sequence **0100** - a much simpler representation than the 8 real numbers that would otherwise be required to represent the trajectory. Due to the periodicity of this particular orbit, the sequence **0100** can be repeated infinitely in both directions to give the representation of the whole orbit: ...**01000100**...

To be more precise, a generating partition creates a conjugacy between the (invertible) map $x_{n+1} = f(x_n)$ and the shift operator $\sigma(S)$; here, S corresponds to the bi-infinite symbol sequence $S = \dots s_{-2}s_{-1} \cdot s_0s_1s_2\dots$ where each s_i corresponds to a specific symbol, the points after the \cdot correspond to the points obtained by iterating the map forwards (starting from the current point's symbol s_0), and the points

before the \cdot correspond to the points obtained by iterating the map backwards (using the inverse). By applying the shift operator σ to S , we get the shifted sequence: $\sigma(S) = \dots s_{-2}s_{-1}s_0 \cdot s_1s_2\dots$, where if s_0 corresponds to some x_0 , s_1 has a direct correspondence to $x_1 = f(x_0)$ [48]. In the context of the example above, if we have the sequence $S = \dots \mathbf{0100} \cdot \mathbf{01000100} \dots$ then $\sigma(S) = \dots \mathbf{01000} \cdot \mathbf{1000100} \dots$.

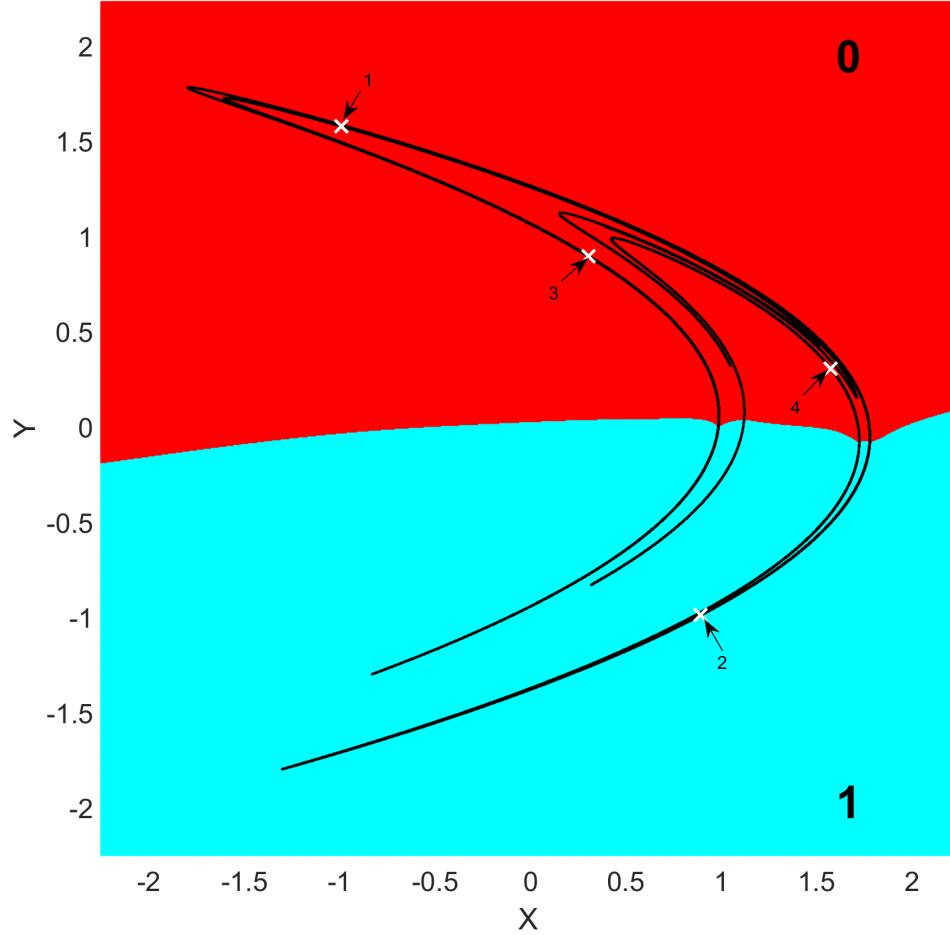


Figure 28: Generating partition of the Hénon map depicted as the boundary between the region corresponding to symbol **0** (red) and symbol **1** (cyan), with the black points corresponding to the points on the attractor of the Hénon map, and the white crosses corresponding to the points of a period 4 orbit on the attractor.

The construction of generating partitions has a long history, despite which a general method of construction of generating partitions is only known for one-dimensional maps, and uniformly hyperbolic systems of any dimension[49]. In the case of non-hyperbolic systems of dimension greater than one, no general method for the construction of them

is yet known; over the past century however, various approaches have been adopted in order to construct generating partitions of a small set of low-dimensional systems.

This history (arguably[50]) began with Hadamard's 1898 paper *Les surfaces à courbures opposées et leurs lignes géodésiques*[51], with his analysis of geodesic flows in which he gave codings for the closed geodesics. This idea was further developed by Morse in 1921[52], and later became the titular topic of Morse & Hedlund's 1938 paper entitled simply *Symbolic dynamics*[53] where symbolic dynamics are treated in a very abstract sense.

In recent years, the construction of generating partitions of symbolic dynamics has led researchers in the field to branch in many different directions.

For example, in the 1985 paper *Generating partitions for the dissipative Hénon map*[47], Grassberger & Kantz use *primary* tangencies between stable and unstable manifolds in order to find generating partitions of the Hénon map for two different sets of parameters. As the images or pre-images of the primary tangencies are also good candidates for partition boundaries, they acknowledge that their choice of generating partition isn't unique. Following this same method, Giovannini & Politi in their 1991 paper[54] find a generating partition of a Poincaré section of the Duffing attractor.

Unstable periodic orbits have also been a key focus in determining generating partitions, primarily due to the finding that in order for a partition to be considered generating all periodic orbits must have a unique symbol sequence[55]. Two examples of the different approaches using this information are given by Lefranc et al. in their 1994 paper[56] and Davidchack et al. in their 2000 paper[26]. Where Lefranc et al. use topological analysis on information extracted from UPOs to create symbolic dynamics of up to 2-dimensional maps, Davidchack et al. present a numerical approach in which they use UPOs in order to iteratively build an approximate generating partition. In their paper, Davidchack et al. confirm that their method works for the well-known

Hénon map, and also present a thereunto unknown generating partition of the Ikeda-Hammel-Jones-Moloney map; in theory, this method isn't restricted to systems of a specific dimension provided the data of the UPOs of the system.

Numerical approaches using experimental data that don't rely on unstable periodic orbits also appeared. An example of which is Hirata et al.'s 2004 paper in which they present a new method which they call *Symbolic shadowing*[57]; this method relies on solving a minimisation problem using an iterative algorithm in which they tessellate the state space with so-called *representatives*. They apply their method to precisely the same systems as in Davidchack et al.[26], obtaining the same partitions despite using a fundamentally different method.

Due to the significant advantages of analysing systems in their symbol space (in terms of speed, memory usage, and robustness to noise), the application of symbolic dynamics can be found throughout the experimental sciences, including the fields of: astrophysics, fluid dynamics, biology, medicine, chemistry, engineering, computer science, and data mining. A thorough review of the applications in each of these fields can be found in Daw et al.'s 2002 review paper, *A review of symbolic analysis of experimental data*[46].

3.1 Motivation

In this part of the thesis we strive to extend the method of the construction of approximate generating partitions presented by Davidchack et al.[26] to a wider class of systems. In their paper, Davidchack et al. apply their method to the case of a Hénon and Ikeda map. While the method as described in this paper gives good results for these two maps, for more complicated maps a few obstacles must be overcome in order to construct good symbolic dynamics. One such obstacle is that in their method, the number of symbols required must be given *a priori*, and so for systems for which the number of symbols required isn't known the method may construct more partitions than necessary.

The results of the application of the method of Davidchack et al.[26] don't necessarily

always give a clear divide between symbol sets; depending on the attractor, isolated points and regions can appear that on observation seem to be labelled with the wrong symbol. As such, when we talk about aiming to construct *good* symbolic dynamics, we mean that we wish to construct partitions that are as well-connected as possible, without any disconnected points or sub-regions.

Our objective is thus to overcome these obstacles, and provide motivation for future research in this direction. In order to do so, we will make use of the available sets of periodic orbits detected by Crofts & Davidchack using the stabilising transformations approach[4]; in particular, of two Hénon maps, an Ikeda map, two Duffing attractors, and the Standard map. A description of each of these maps and their corresponding sets of unstable periodic orbits is given below.

3.2 Maps

Throughout this part of the thesis we will make use of the sets of unstable periodic orbits for the Hénon, Ikeda, Duffing, and Standard maps that were found by Davidchack et al. using the stabilising transformations method[4, 18, 19].

Hénon Map

The Hénon map is given by:

$$\begin{aligned}x_{n+1} &= a - x_n^2 + by_n \\ y_{n+1} &= x_n\end{aligned}\tag{3.1}$$

and was first defined by Hénon in his 1976 paper[58] *A Two-dimensional Mapping with a Strange Attractor*. In this paper, we will focus on two different sets of parameters.

Hénon I. The first, $a = 1.4$ & $b = 0.3$ is the *classic* set of parameters defined by Hénon[58] as forming a chaotic attractor. The portrait of this attractor can be seen in

Fig. 29. For this set of parameters we have a set of 1,614 UPOs ranging from period 1 to period 20.

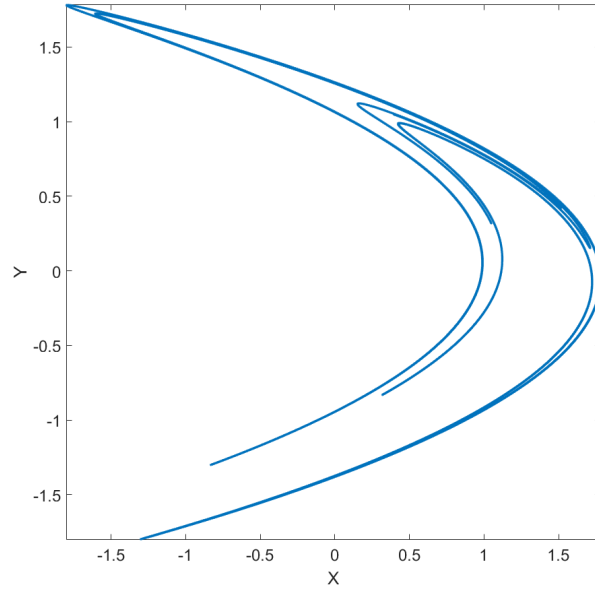


Figure 29: Portrait of the Hénon map's attractor for parameters $a = 1.4$, $b = 0.3$.

Hénon II. The second set of parameters we will use, $a = 1$ & $b = 0.54$ also exhibit chaotic behaviour, and the portrait of the attractor in this case can be seen in Fig. 30. For this set of parameters we have a set of 1,250 UPOs ranging from period 1 to period 29.

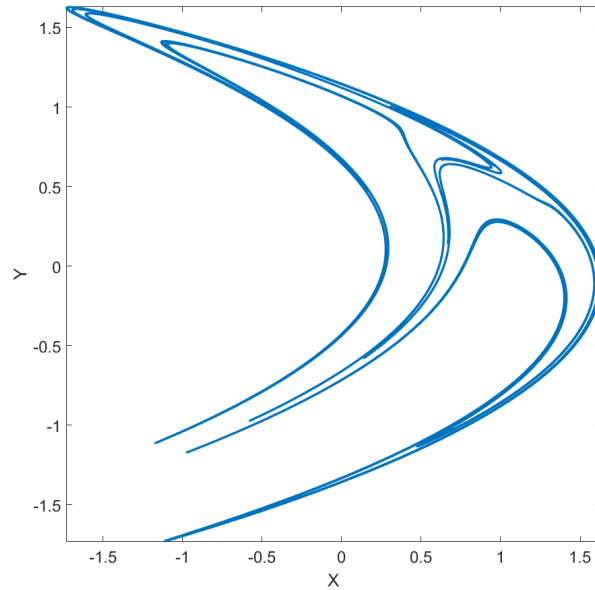


Figure 30: Portrait of the Hénon map's attractor for parameters $a = 1$, $b = 0.54$.

Both of these sets of parameters were used by Grassburger & Kantz in their 1985 paper[47] *Generating Partitions for the Dissipative Hénon map*. Their results will form a good point of comparison for the results obtained in this thesis.

Ikeda Map

The Ikeda-Hammel-Jones-Moloney map[24, 25] (henceforth referred to simply as the Ikeda Map) is given by:

$$\begin{aligned}x_{n+1} &= a + b(x_n \cos \phi_n - y_n \sin \phi_n) \\y_{n+1} &= b(x_n \sin \phi_n + y_n \cos \phi_n)\end{aligned}\tag{3.2}$$

where $\phi_n = \kappa - \eta/(1 + x_n^2 + y_n^2)$.

In order to present our results in the same context as Davidchack et al.[26], we will use the canonical choice of parameters of the Ikeda map for which there is a chaotic attractor: $a = 1$, $b = 0.9$, $\kappa = 0.4$, & $\eta = 6$. We will also use the same set of UPOs for this map that was used by Davidchack et al.[26] which contains 20,061 UPOs ranging from period 1 to period 20. A portrait of the chaotic attractor exhibited by the map for these parameters can be seen in Fig. 31.

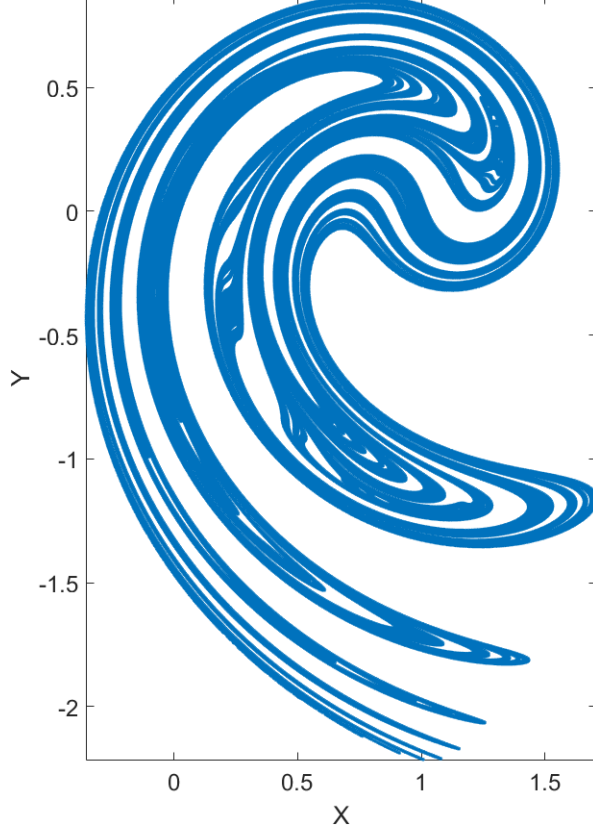


Figure 31: Portrait of the Ikeda map's attractor for parameters $a = 1$, $b = 0.9$, $\kappa = 0.4$, $\eta = 6$.

Duffing Attractor

The Duffing flow is given by the system of ODEs:

$$\begin{aligned} x' &= y \\ y' &= -\delta y + \beta x - x^3 + \gamma \cos(ft) \end{aligned} \tag{3.3}$$

As we are focusing only on two-dimensional maps in this part of the thesis, we will use the Poincaré section of the flow which is constructed by taking the solution of Eq. 3.3 when the phase ft is a multiple of π , and inverting the co-ordinates $(x, y) \rightarrow (-x, -y)$ (consistent with the construction used in Giovannini & Politi[54]).

Duffing I. As in Giovannini & Politi's 1991 paper *Homoclinic tangencies, generating partitions and curvature of invariant manifolds*[54] we will take the parameters as: $\delta = 0.25$, $\beta = 1$, $\gamma = 0.4$, & $f = 1$. For this map we will use a set of 7,757 UPOs

ranging from period 1 to period 20. The portrait of the chaotic attractor of this map can be seen in Fig. 32.

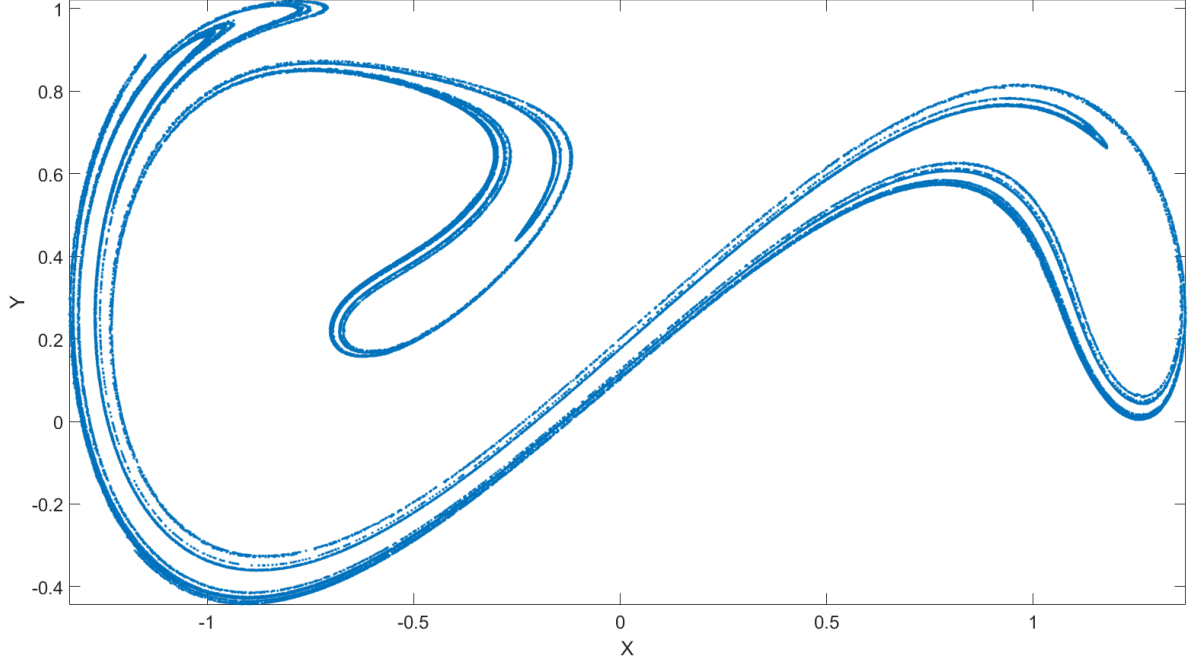


Figure 32: Portrait of the Poincaré section of the Duffing map's attractor for parameters $\delta = 0.25$, $\beta = 1$, $\gamma = 0.4$, $f = 1$.

Duffing II. In addition to the above, we will also experiment with a slightly fuller (in that it's 'more 2-dimensional') form of the Duffing map, with parameters: $\delta = 0.02$, $\beta = 0$, $\gamma = 34$, & $f = 1$. For this map we have a set of 4,276 UPOs ranging from period 1 to period 20. The portrait of the map for this set of parameters can be seen in Fig. 33; note that throughout this section we will use square axes for ease of visualisation of the attractor, but in reality the range of the y -direction is much larger than that of the x -direction.

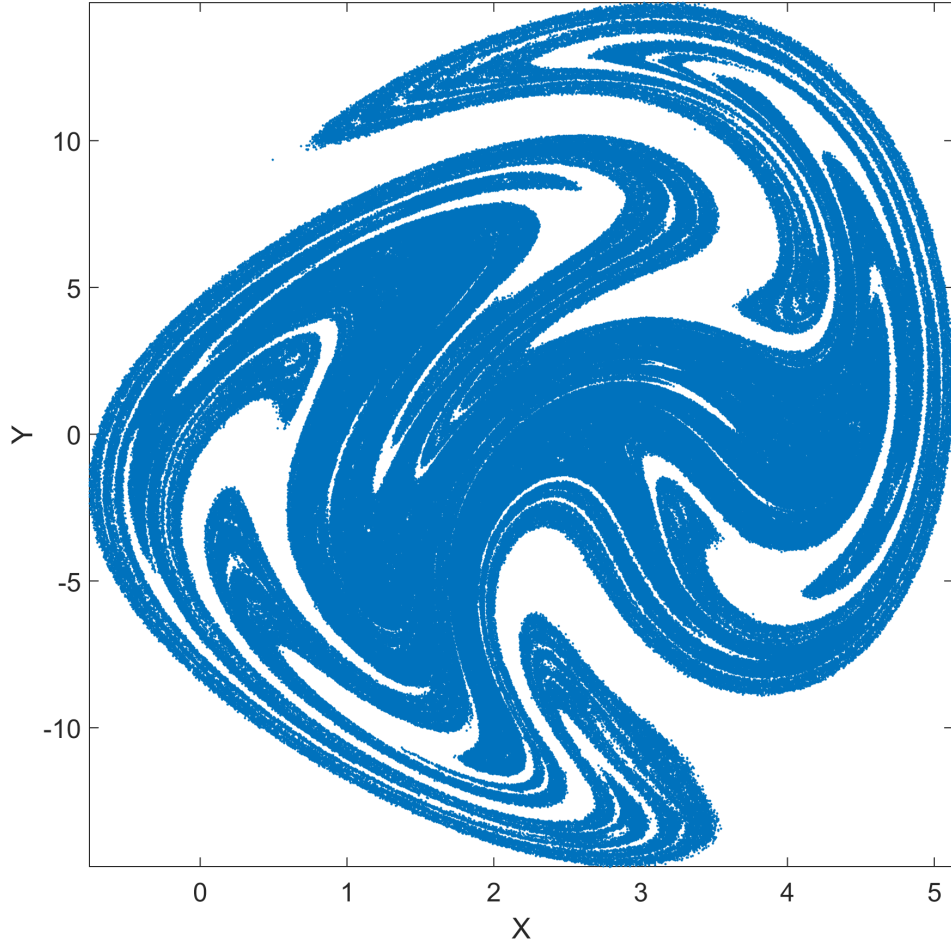


Figure 33: Portrait of the Poincaré section of the Duffing II attractor for parameters $\delta = 0.02$, $\beta = 0$, $\gamma = 34$, $f = 1$, using square axes.

Standard Map

The standard map is given by:

$$\begin{aligned} x_{n+1} &= y_n \\ y_{n+1} &= -x_n + 2y_n - \alpha \cos(y_n) \bmod 2\pi \end{aligned} \tag{3.4}$$

As we wish to present our results in the same context as Christiansen & Politi[59], we will use $\alpha = 6$. The portrait of the chaotic attractor of this map can be seen below in Fig. 34, where the points of periodic orbits up to period 9 are plotted; note that due to the modulo 2π in the equation of the map, the map is periodic of period 2π in both the x & y directions (representing dynamics on a torus), and so Fig. 34 can be tiled infinitely in either direction.

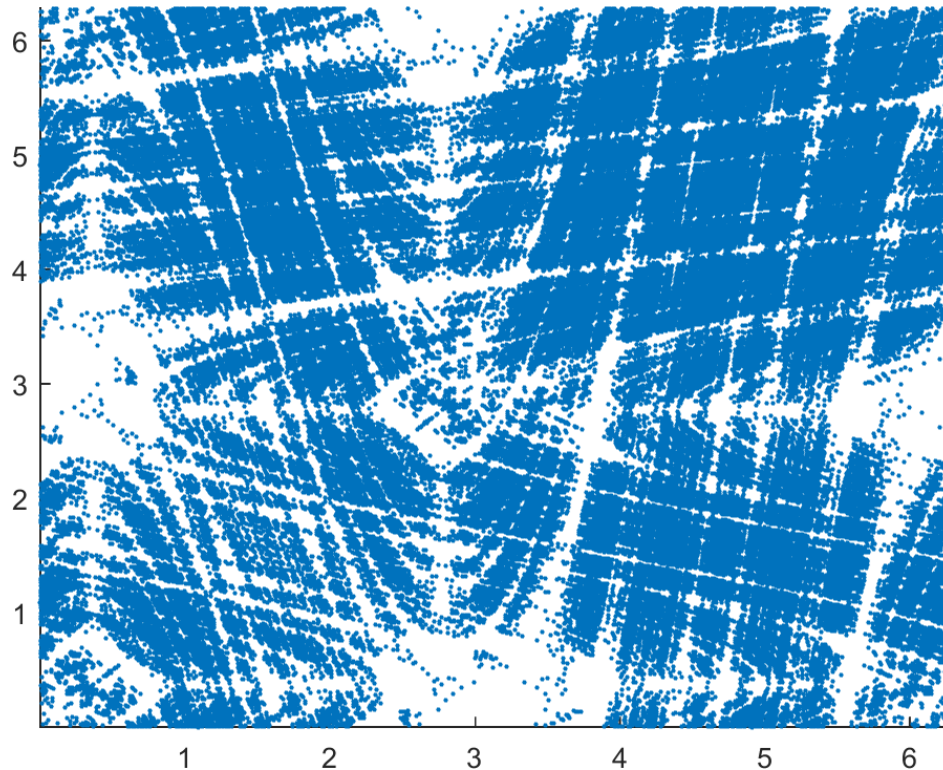


Figure 34: Portrait of the Standard map's attractor for $\alpha = 6$.

For this map we have a set of 50,499 periodic orbits ranging from period 1 to period 9.

3.3 DLBD Method

The method devised by Davidchack, Lai, Bollt, & Dhamala[26] (henceforth referred to as the DLBD method) uses unstable periodic orbits of the Hénon and Ikeda maps in order to construct approximate generating partitions of these attractors. The key underlying principles of this method are that in order for a partition to be generating, all periodic orbits must have a unique symbol sequence[55], and that periodic points in close proximity to one another should be encoded with the same symbol (excluding those near the partition boundary).

Using the former principle, the method starts by assigning different symbols to each fixed point; it then proceeds through the periodic orbits in order of ascending period assigning to each point the symbol to the symbol set to which the point is *closest*. After all points of a given period are assigned a symbol, a check is performed to make sure

that no orbit has neither a periodic substring, nor the same symbol sequence as another orbit of the same period.

In the case of the former, the point which is closest to another symbol set is relabelled. The above check is then repeated; if this relabelling causes no further conflicts then the procedure continues, if another conflict is created then this relabelling is undone and the next closest candidate is relabelled (and so on).

In the case of the latter, points on both orbits are taken into consideration, and the single point that's closest to another symbol is relabelled. If this relabelling solves the conflict and causes no further conflicts, the procedure continues; if another conflict is created, then this relabelling is undone and the next closest candidate is relabelled (and so on).

Assuming that we have a finite set of n symbols: $S = \{S_1, S_2, \dots, S_n\}$, where S_i is the set containing all periodic orbit points assigned to symbol i , the proximity from a point \mathbf{x} to a symbol set S_i is defined as follows[26]:

$$Z^{(i)}(\mathbf{x}) = \sum_{\mathbf{x}_k \in S_i} |\mathbf{x} - \mathbf{x}_k|^{-2} \quad (3.5)$$

(note that this choice of proximity function is not unique, and we will later experiment with alternatives).

To give an example in the context of the Ikeda map, the equilibrium point $(0.5328, 0.2469)$ is assigned the symbol **1**, and the second equilibrium point $(1.1143, -2.2857)$ is assigned the symbol **2**. After this, we calculate the Z-values for each point of period two to each of the two symbol sets, assigning to each point the symbol with the highest Z-value (corresponding to lowest distance); this, however, results in both of the period two points being assigned the symbol **1**. As such, we compare the Z-values of each point to the other symbols, choosing to relabel the point to symbol with the highest corresponding Z-value. In this case, since we only have two symbols, we choose to relabel the point $(0.5098, -0.6084)$ to **2**, as it has a $Z^{(2)}$ value of 0.3146 compared to the point $(0.6216, 0.6059)$'s $Z^{(2)}$ value of 0.1162.

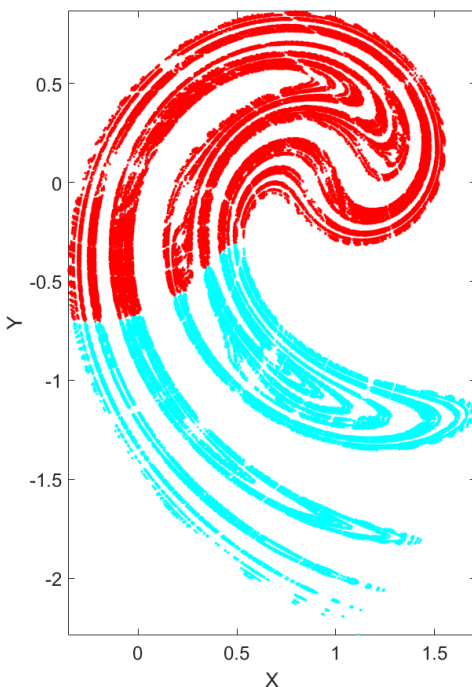


Figure 35: Results of applying the above method to the UPOs of the Ikeda map up to period 20.

By proceeding in this fashion, the code only has to change 17 points in order to encode all points up to period 20 (a total of 373,006 points) in such a way that the each UPO is encoded by a unique symbol string. The result of this process can be seen in Fig. 35, where points belonging to symbol **1** are coloured red, and points belonging to symbol **2** are coloured cyan. While the generating partition itself isn't obtained from this process, its position can be approximated by the position of the two differently coloured symbol sets. By including orbits of higher period in the construction of the symbolic dynamics, the accuracy of the approximation can be improved.

Another example of where this method works well is the Hénon I map where the results of applying this process to UPOs of up to period 20 (Fig. 36) match well with the generating partition obtained by Grassberger & Kantz[47], despite being constructed through very different means. There is however one period-13 point in the cyan region that is mislabelled which can be seen more clearly in the zoomed-in plot on the right of Fig. 36.

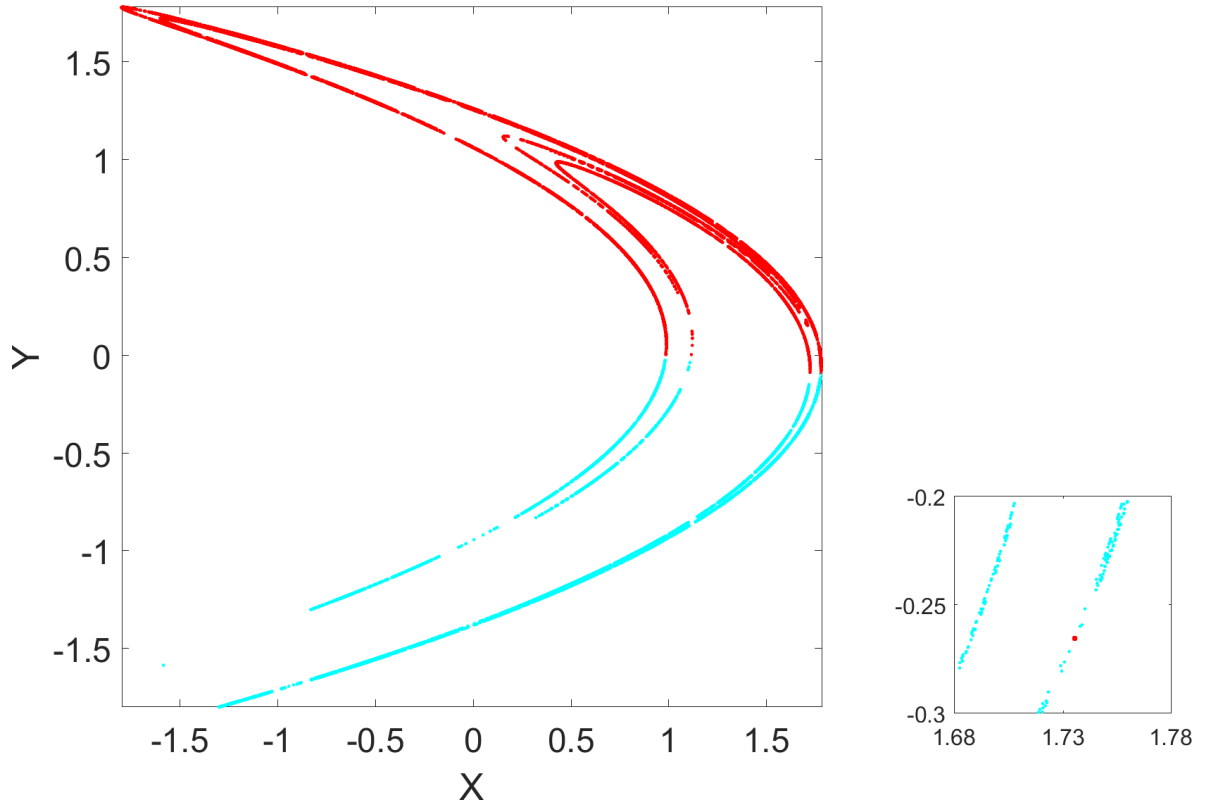


Figure 36: Constructed symbolic dynamics of the Hénon map's attractor for parameters $a = 1.4$, $b = 0.3$ along with zoomed in sub-plot around mislabelled point.

An example of where this method doesn't work so well is the Hénon map for the slightly different parameters: $a = 1$, $b = 0.54$, henceforth referred to as Hénon II, for which we will use UPOs of up to period 29. It's clear from observation of Fig. 37 alone that the symbol sets created by the DLBD algorithm don't allow for any good (connected) partitions between the red and cyan regions due to the interlacing of symbol sets towards the middle of the attractor. Grassberger & Kantz's 1985 paper[47] showed that a simply connected generating partition does indeed exist for this attractor, and so in our research we use this example to motivate the further improvement of the DLBD method.

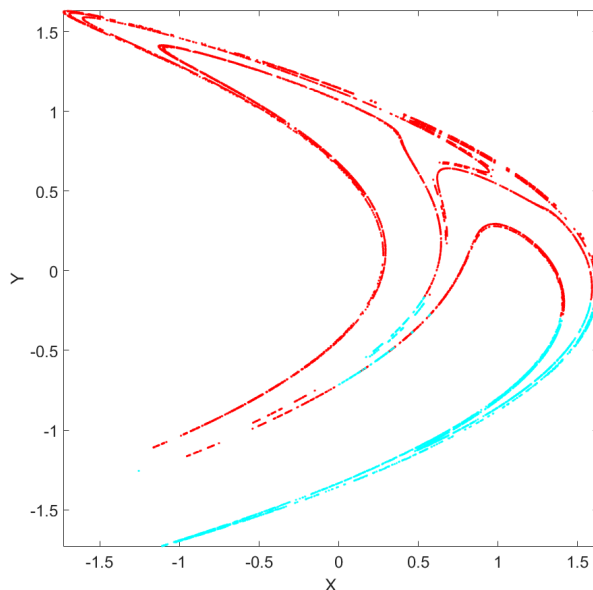


Figure 37: Constructed symbolic dynamics of the Hénon map's attractor for parameters $a = 1$, $b = 0.54$.

All of the three above examples have been well-researched, and so the number of required symbols was known in advance. In general, this may not be the case (for example in the case of the Duffing II and Standard map which we will discuss later). As we hope to make our method as general as possible, changes must be made to the algorithm in order to accommodate such circumstances.

3.4 Dynamic Addition of Symbols

To address this problem, we must modify the method in order to add additional symbols when required.

Assuming that the set of UPOs of a given period n is complete, the minimum number of symbols m required to represent all UPOs of period n can be determined by calculating the number of *circular permutations of length n with repetition* - that is, the number of ways a set of objects can be arranged in order to create a sequence of length n , including repetition of objects. For example, if we have the alphabet of 2 symbols **1**, **2**, the number of ways they can be arranged to form a sequence of length 4 is 6:

1111, 2222, 1222, 2111, 1212, 1221

Sequences that are the same after some rotation, such as **1222** \rightarrow **2122** shouldn't be included in this count, as they represent the same circular permutation of objects.

The solution of the problem of finding this number was first solved by Major Percy A. MacMahon in his 1892 memoirs[60], and was later explained with more clarity in Rosen et al.'s 1999 *Handbook of discrete and combinatorial mathematics*[61]. The number of circular permutations of length n where the elements are chosen from an alphabet of size m is defined by Rosen et al.[61] as:

$$\sum_{k|n} \left(\frac{1}{k} \sum_{d|k} \mu(d) m^{k/d} \right) \quad (3.6)$$

where μ is the *Möbius function* given by Rosen et al. as[61]:

$$\mu(d) = \begin{cases} 1 & \text{if } d = 1 \\ (-1)^k & \text{if } d = p_1 p_2 p_3 \dots p_k, \text{ i.e. the product of } k \text{ distinct primes.} \\ 0 & \text{if } d \text{ is divisible by the square of a prime.} \end{cases} \quad (3.7)$$

Indeed, in the case of the previous example with $m = 2$ symbols and $n = 4$ elements, the result of Eq. 3.6 is 6.

In the context of symbolic dynamics, however, this particular formulation doesn't quite fit. In order for the symbolic dynamics to be well-defined, a UPO of period n may not have any periodic substrings. In the case of the above example **1111** & **2222** both have periodic substrings of period 1, given by **1** & **2** respectively; additionally, **1212** has a periodic substring of period 2, given by **12**. As such, for this m and n we desire the outcome of our modified Eq. 3.6 to be 3, corresponding to the sequences: **1222**, **2111**, **1221**.

Performing such a modification is thankfully quite simple, requiring only the adjustment of the outermost sum of Eq. 3.6. Instead of summing over all $k|n$, we will instead

only take the case $k = n$, resulting in the new expression:

$$\frac{1}{n} \sum_{d|n} \mu(d) m^{n/d} \quad (3.8)$$

with μ defined as previously. Evaluation this expression for $m = 2$ and $n = 4$ results in the expected answer of 3 cyclic permutations.

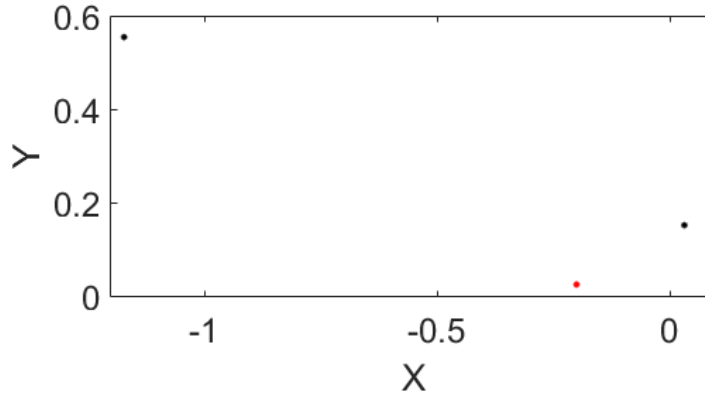
In order to make use of this method for the purpose of dynamically adding symbols, the problem must be reformulated a bit. In our case, we don't seek the number of cyclic permutations, but instead we seek to find the smallest m such that the number of cyclic permutations is greater or equal to the number of UPOs of period n . In other words:

$$m_n = \min \left\{ m : \frac{1}{n} \sum_{d|n} \mu(d) m^{n/d} \geq N_n \right\} \quad (3.9)$$

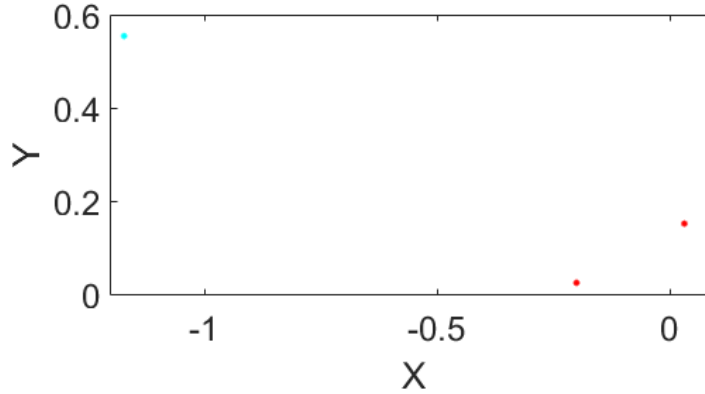
where N_n is the number of UPOs of period n . Using this, if m_n is greater than the number of symbols used in the previous step, then a new symbol must be added. In order to determine how to add the new symbol, we will choose the point in the set of points given by all UPOs of period n that is furthest (for example in terms of the Z-distance in Eq. 3.5) from the current existing symbols as the first point of the new symbol. After adding this new symbol, the construction of the symbolic dynamics can proceed as normal.

This addition to the DLBD method allows us to extend the applications to maps whose numbers of required symbols isn't known *a priori*. For example, in the case of the Duffing map with parameters $\delta = 0.25$, $\beta = 1$, $\gamma = 0.4$, & $f = 1$ (henceforth referred to as Duffing I) there is only one equilibrium; as there is one UPO of period 2 at least 2 symbols are required, and so without the above modification the algorithm wouldn't be able to proceed past period 1.

Upon reaching period 2, the algorithm recognises that the number of permutations



(a) Points of UPOs of the Duffing I attractor of period 1 (in red), and period 2 (in black).



(b) The leftmost point of period 2 is assigned to the new symbol **2** (cyan), and the rightmost is assigned to symbol **1**.

Figure 38: Encoding of points of the period 2 UPO of the Duffing I attractor.

possible with 1 symbol isn't enough, and so it chooses the point furthest from the symbol set **1** (in this case meaning furthest from the equilibrium), and assigns it to the new symbol set **2**. This process can be seen visually in Fig. 38. In Fig. 38a the equilibrium point (symbol set **1**) is shown in red, and the two points of the period 2 orbit are in black. As the leftmost point of the period 2 UPO is furthest from the symbol set **1**, it is assigned the new symbol **2** (in cyan, as seen in Fig. 38b). The process then continues as normal, with the remaining point being assigned to symbol **1** before moving onto the next period.

When reaching period 3 a similar situation occurs, and so a third symbol is added. After this, no more symbols are required for the UPOs of up to period 20, and so the algorithm proceeds as described in the previous section. The results of this process

can be seen in Fig. 39. While on the whole these results seem good, the small strip of green in the otherwise red region doesn't result in such a clear-cut partition between the symbol sets as the generating partition constructed by Giovannini & Politi[54], which we will later show is achievable by making further modifications to the DLBD method.

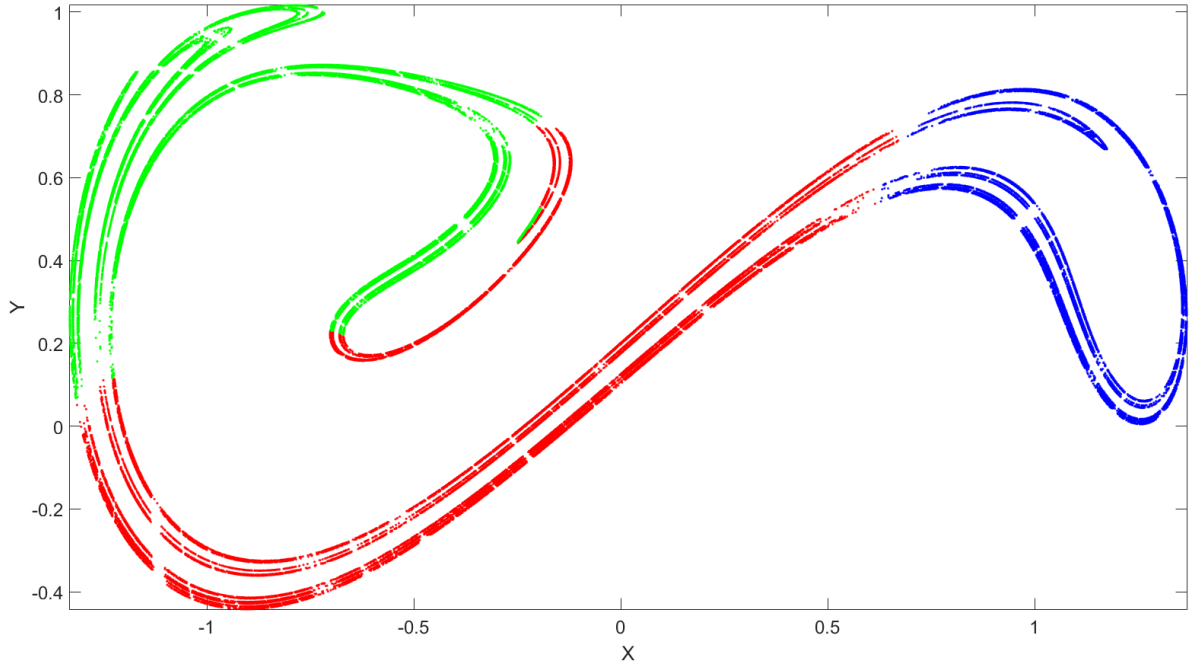


Figure 39: Constructed symbolic dynamics of Duffing I's attractor, using UPOs of up to period 20.

For the alternative set of parameters of the Duffing map $\delta = 0.02$, $\beta = 0$, $\gamma = 34$, $f = 1$, (henceforth referred to as Duffing II) two more symbols are also required. The resulting symbolic encodings aren't quite as nice as in the previous case, as seen in Fig. 40, with many points being disconnected from their symbol sets.

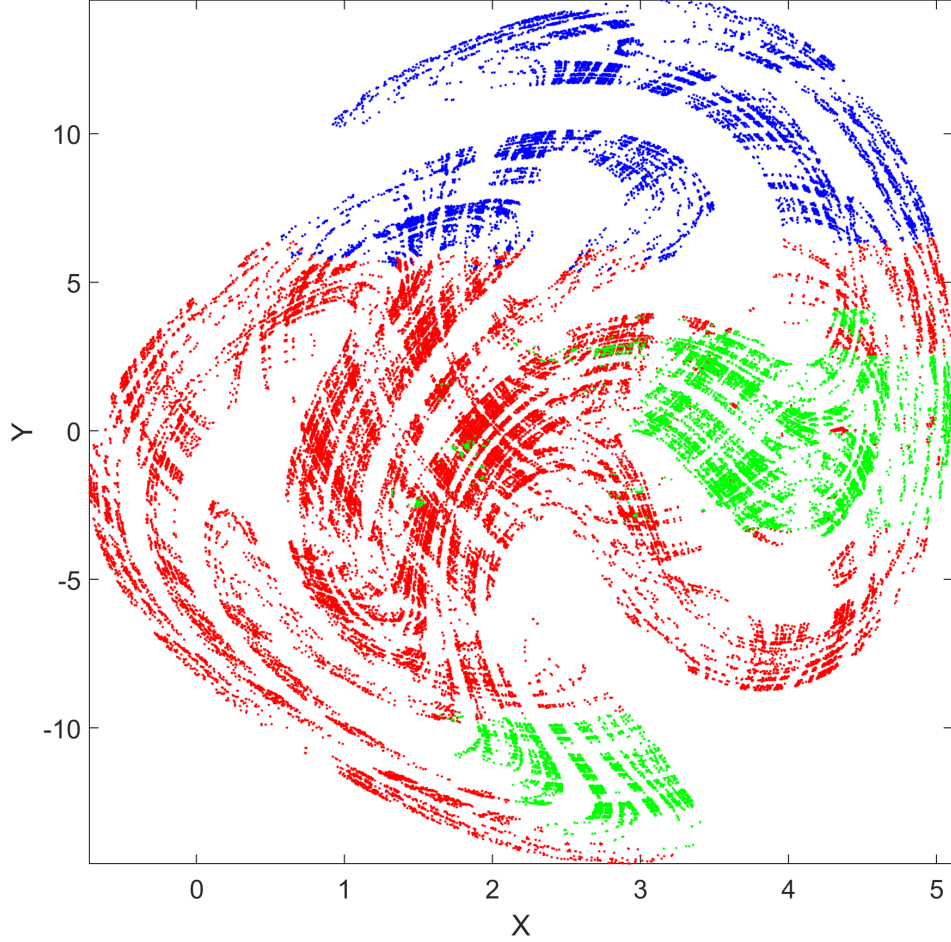


Figure 40: Constructed symbolic dynamics of Duffing II's attractor, using UPOs of up to period 20.

The disconnected points and regions shown in the previous two examples and the Hénon II attractor of the previous section give us motivation for making further improvements to the method. Over the next few sections we will experiment with different proximity kernels and other modifications to the DLBD method to the end of obtaining *good* partitions with as few disconnected points and regions as possible.

3.5 Normalisation of Z .

One such modification that can be made in order to improve the connectedness of the results is through the normalisation of the proximity kernel of Eq. 3.5:

$$Z^{(i)}(\mathbf{x}) = \sum_{\mathbf{x}_k \in \mathcal{S}_i} |\mathbf{x} - \mathbf{x}_k|^{-2}$$

As the proximity kernel above simply sums the distance $|\mathbf{x} - \mathbf{x}_k|^{-2}$ over **all** points in the symbol set, sets with more points will likely return higher Z -values for different \mathbf{x} . This effect can result in sets that are assigned more points early on expanding quickly, with sets that are assigned few points early on barely growing at all.

By normalising $Z^{(i)}(\mathbf{x})$ with respect to the number of elements in S_i ,

$$\hat{Z}^{(i)}(\mathbf{x}) = \frac{1}{|S_i|} \sum_{\mathbf{x}_k \in S_i} |\mathbf{x} - \mathbf{x}_k|^{-2} \quad (3.10)$$

this issue can be alleviated. To give a concrete example of the effect that this can have on the outcome, we will take the Hénon II attractor as an example. By using the DLBD method up to UPOs of period 7, we get the approximated generating partition shown in Fig. 41. In this figure, the red region corresponds to the *basin* of symbol **1** (i.e. the points \mathbf{x} where $Z^{(1)}(\mathbf{x}) > Z^{(2)}(\mathbf{x})$), and the cyan region corresponds to the basin of symbol **2**; the black dots correspond to points of UPOs of period ≤ 7 which have already been assigned symbols; the black circles correspond to points on the UPOs of period 8 which have yet to be assigned a symbol.

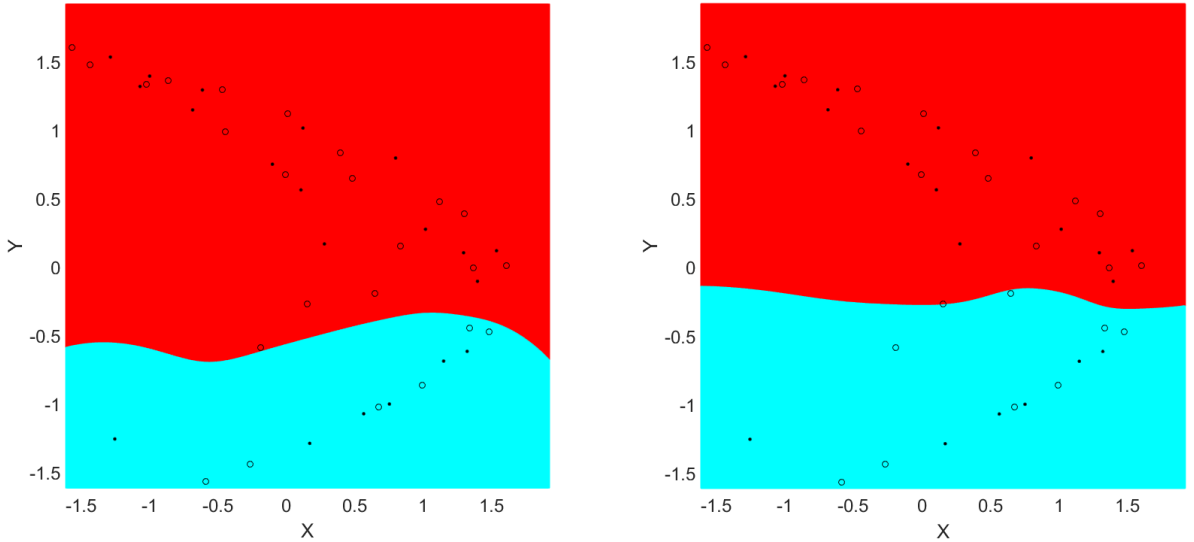


Figure 41: Basin of symbols **1** (red) and **2** (cyan) using: Z of Eq. 3.5 (*left*), \hat{Z} of Eq. 3.10 (*right*).

The left plot shows what happens in the case of the DLBD method without normalisation of Z . From this we can see that (especially on the left) the red region extends much further south than one might expect, despite having no points (black dots) in

the vicinity. This is due to the fact that symbol **1** has more points, and so it in a sense overpowers symbol **2**. The right plot shows the effect of normalising Z , and using instead \hat{Z} . The end result of using these two different distance functions can be seen in Fig. 42; while there are still a couple of mislabelled points, the resulting symbol sets obtained by using \hat{Z} over Z are more well-connected, which makes the positioning of the generating partition much clearer.

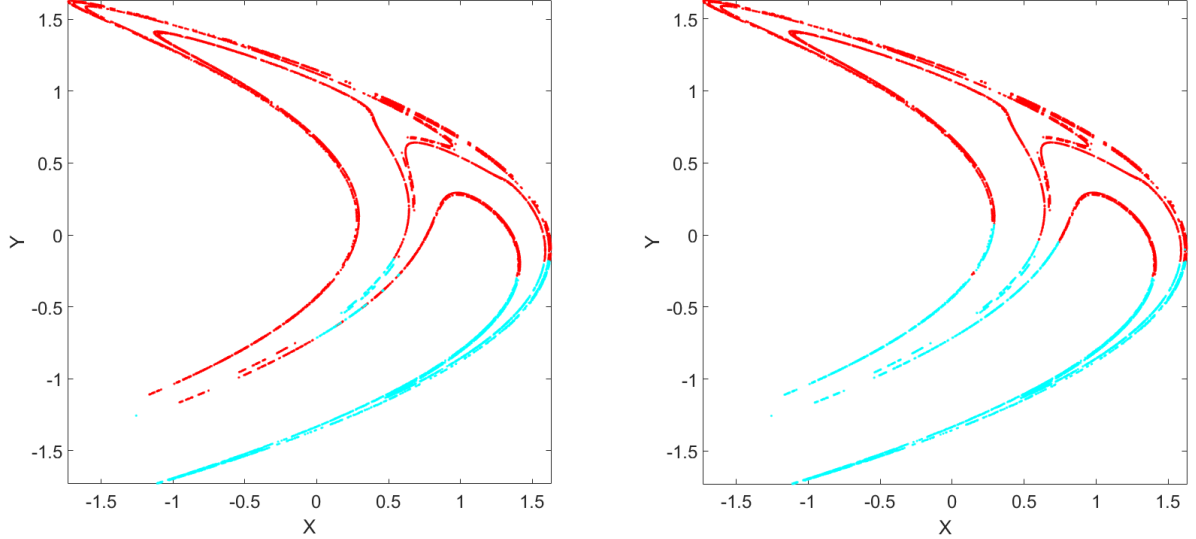


Figure 42: Resulting symbol sets of Hénon II using: Z of Eq. 3.5 (*left*), \hat{Z} of Eq. 3.10 (*right*).

In terms of the effect that using \hat{Z} has on the Ikeda map and Hénon I attractor used in Davidchack et al.'s[26], the results remain largely the same. In the case of the Ikeda map, there's no difference at all in the labelling of any of the points on UPOs up to period 20. In the case of the Hénon I attractor two points are labelled differently; from Fig. 43 it can be seen that the two changes from **1** (red) to **2** (cyan) result in a clearer, more connected partition.

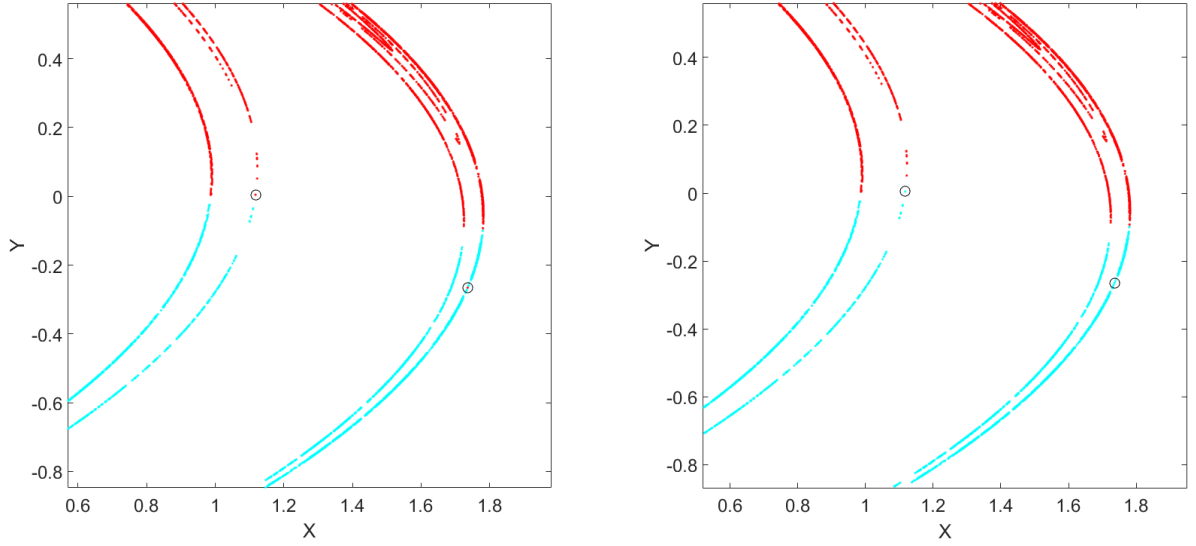


Figure 43: Resulting symbol sets of Hénon I using: Z of Eq. 3.5 (*left*), \hat{Z} of Eq. 3.10 (*right*). Differences in assigned symbols are represented by black circles. (This figure is zoomed in around the changes, for ease of viewing).

In the case of the Duffing I attractor, we can see quite a substantial change between the resulting symbol sets (Fig. 44). Overall we see symbol **1** (red) ceding some of its coverage to symbols **2** (green) and **3** (blue), with symbol **3** seeing a slight extension towards the left; and symbol **2** expanding all the way south, and replacing its coverage of the upper fold with coverage only of the bend; we also see at least one green point that is still mislabelled (slightly above the bend).

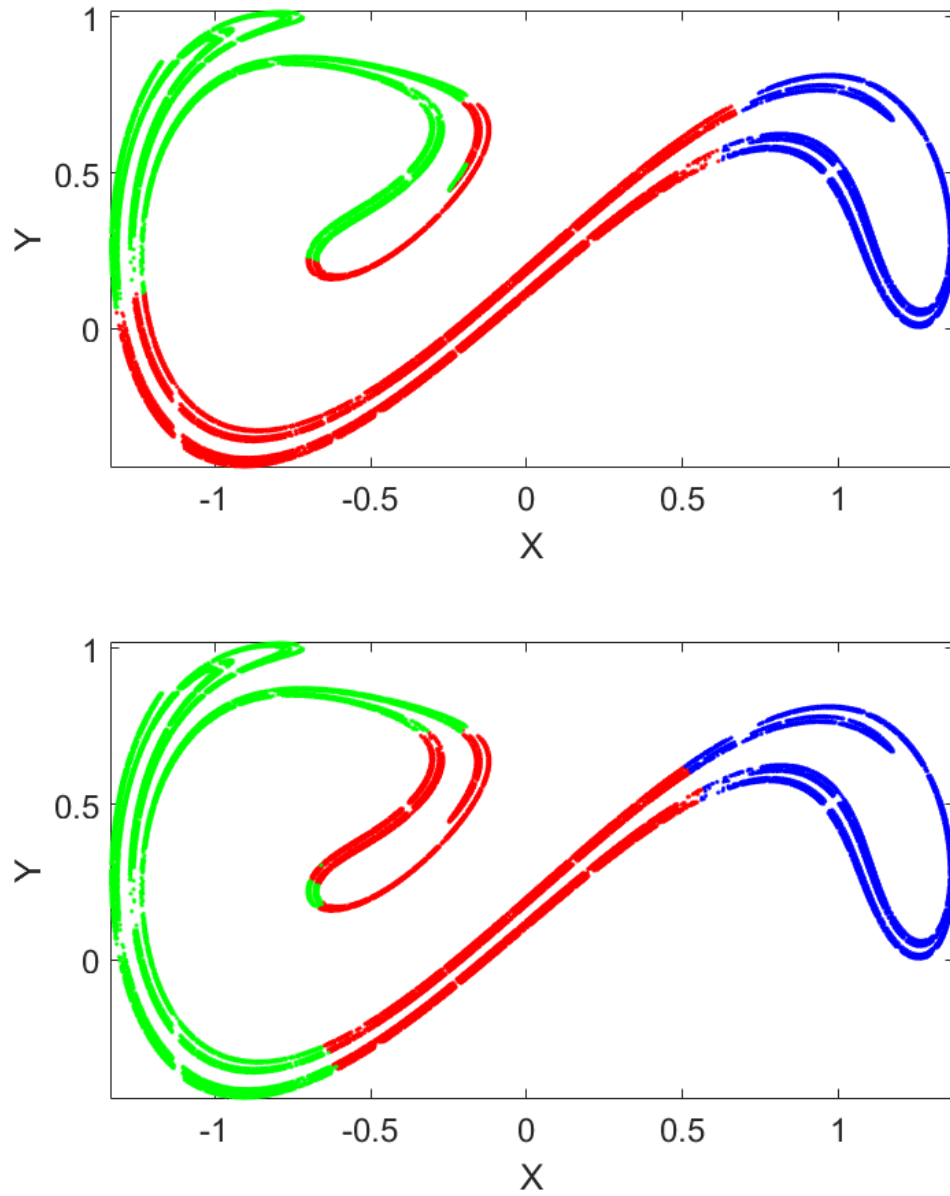


Figure 44: Resulting symbol sets of Duffing I using: Z of Eq. 3.5 (*top*), \hat{Z} of Eq. 3.10 (*bottom*).

In the case of the Duffing II attractor we also see slight improvements in connectedness, but there are still too many disconnected regions for the results to be considered an acceptable approximation of a generating partition.

3.6 Staggered Construction

Even when using the normalised proximity kernel \hat{Z} , the problem of isolated points and regions being mislabelled is still apparent in the more complex Duffing attractors. This issue is in part due to the fact that at the beginning of the construction of the partition the number of points is relatively low, and so some of the points may be assigned a symbol that is inconsistent with the labelling of higher-period points. As the algorithm doesn't currently allow for the relabelling of lower period points, these points then go on to become isolated from their symbol set, or can even go on to develop into larger isolated regions of points isolated from their symbol set.

In order to alleviate this problem, we devised a *staggered method* of the construction of symbol sets. The method is largely the same as the DLBD method, with the addition that once a period is complete we recalculate all previous periods using the current period's data, and repeat this process until convergence before continuing.

The staggered method assigns symbols to periods 1 and 2 as normal; and for periods $n = 3$ onwards we apply the following process:

1. Compute symbol sets for period n using data of symbols for orbits of period 1 to $n - 1$ (as usual).
2. Recompute the symbols for UPOs of period 2 to $n - 1$ using the symbol sets of period n as a basis.
3. Recompute the symbol set for period n using the newly calculated symbol sets of period 1 to $n - 1$ as a basis.
4. Repeat the above two steps until the symbol set of period n converges before moving on to period $n + 1$.

This repetition until convergence was introduced in the hope of reducing the number of isolated points. Fig. 45 shows a comparison between the results of the DLBD method and the staggered construction method in the context of Duffing I; the circled points on the top subfigure represent the differences between the results of the two methods.

In this example the staggered method seems to work well in ridding the encoding of mislabelled points, as hoped.

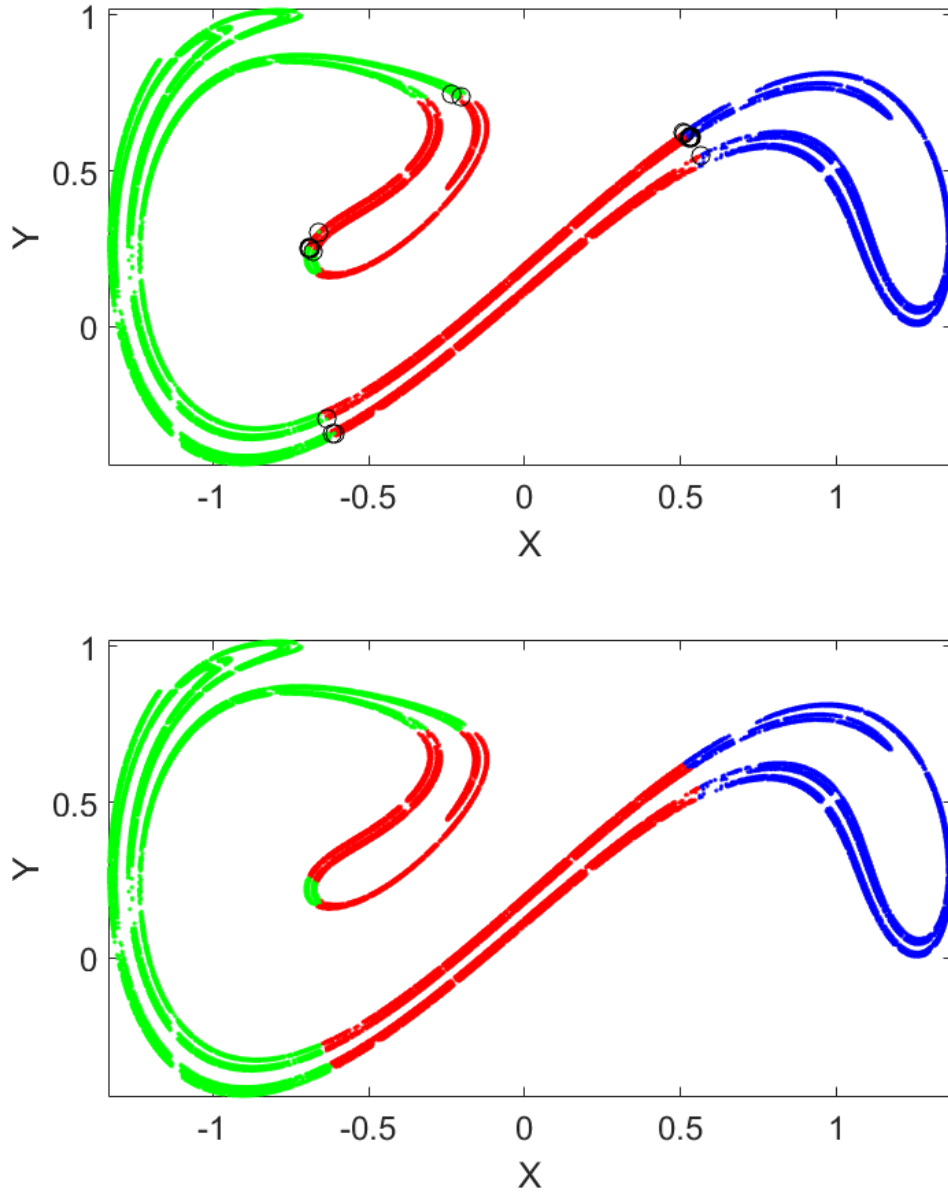


Figure 45: Resulting symbol sets of Duffing I using \hat{Z} with the DLBD method (*top*) & staggered method (*bottom*). The differences between the sets are represented by black circles in the top subfigure.

In the case of the Duffing II attractor, the results unfortunately aren't as promising (Fig. 46). The changes made to the lower half of the attractor seem to give better partitioning overall, with the red region transformed into a more consolidated region; however as a result the top half of the attractor seems to have suffered, with a green region appearing

in the middle of the blue domain, and the partition boundaries between each of the symbols becoming less clear.

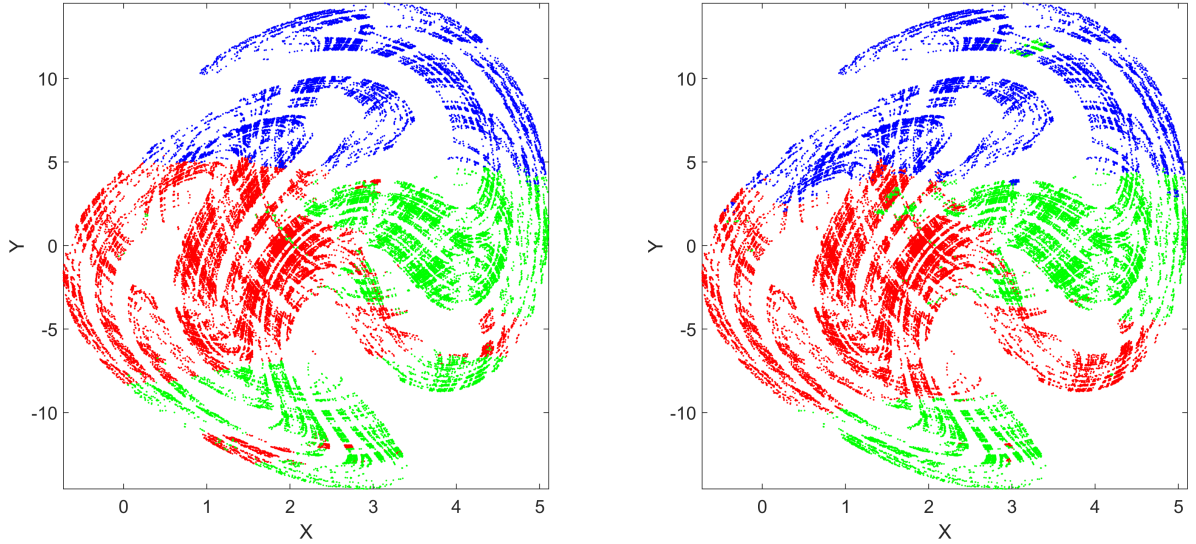


Figure 46: Resulting symbol sets of Duffing II using \hat{Z} with the DLBD method (*left*) & staggered method (*right*).

3.7 Gaussian Proximity Kernel

Thus far, the proximity kernels that we've used have been relatively slowly decaying with respect to the distance between the point \mathbf{x} and the points in the symbol set, denoted by \mathbf{x}_k in Eqs. 3.5 & 3.10. In order to increase the connectedness of symbol sets, we sought a faster decaying proximity kernel; by giving points that are much closer a much higher score, a heavier focus is put on the local neighbourhood around each given point - reducing the probability of an isolated point being mislabelled.

The new function that we experimented with was the Gaussian-inspired kernel:

$$\tilde{Z}^{(i)}(\mathbf{x}) = \sum_{\mathbf{x}_k \in S_i} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_k|^2}{2\sigma^2}\right) \quad (3.11)$$

where σ is a parameter to be determined, corresponding to the width of the Gaussian kernel. By increasing σ the width of the bell curve is increased, this increase in width corresponds to a decrease in gradient of the curve, which in turn decreases the sensitivity of the Z-value to the distance $|\mathbf{x} - \mathbf{x}_k|$. In other words the higher the σ value, the

less local the behaviour of the method becomes.

For each map different σ values are suitable, due to the varying sizes of the dimensions of each attractor. In the case of the Duffing I attractor through trial and error we found that for the staggered method of the previous section σ values in the range of 0.02 to 0.1 give good results. Of particular interest are the two cases $\sigma \in [0.02, 0.022]$ and $\sigma \in [0.023, 1]$; in each of these two cases we see two different possible generating partitions being constructed, with the transition somewhere in the interval $(0.022, 0.023]$. Two archetypal examples of the different regimes can be seen in 47, with the top subfigure being a result of using $\sigma = 0.02$, and the bottom subfigure using $\sigma = 0.03$.

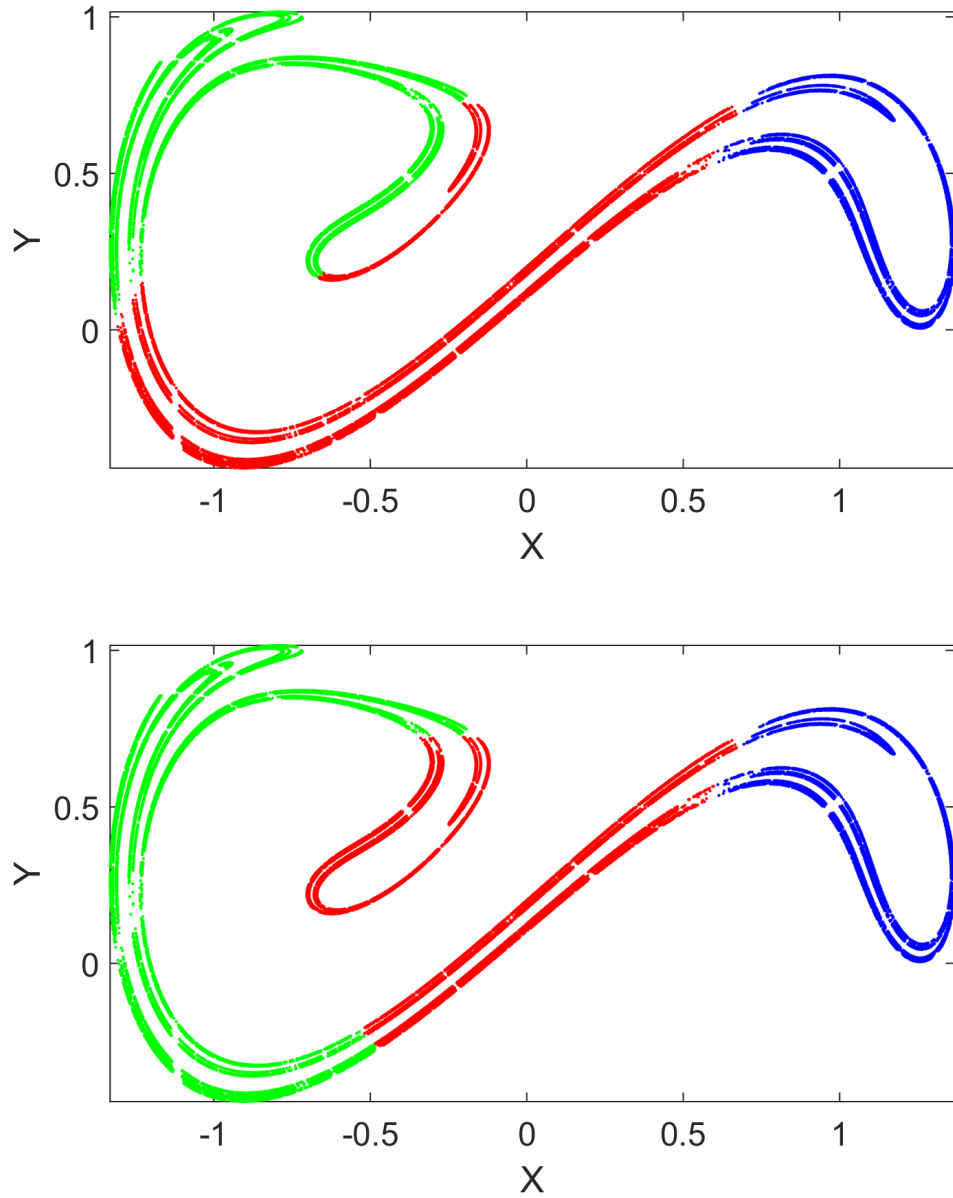


Figure 47: Resulting symbol sets of Duffing I using $\sigma = 0.02$ (*top*) & $\sigma = 0.03$ (*bottom*).

In both cases clear partitioning between the different symbol sets without any isolated points or regions is present. Despite the fact that the above two estimates of generating partitions are slightly different to the results found by Giovannini et al.[54], it's probable that those found by Giovannini et al. are either images or pre-images of those found by our method.

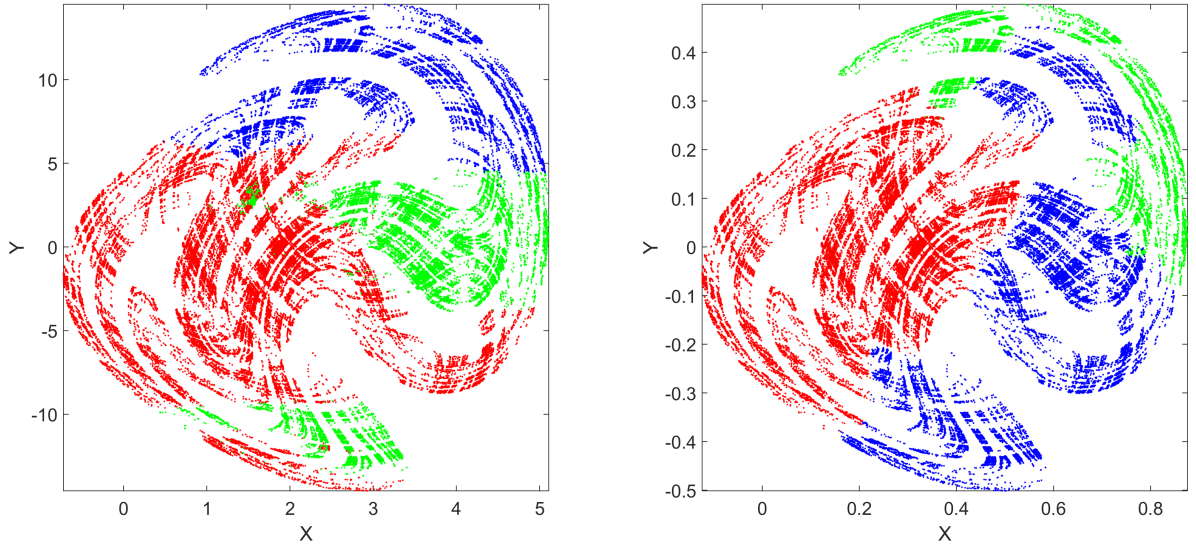


Figure 48: Resulting symbol sets of Duffing II using $\sigma = 0.34$ (*left*), using normalised space coordinates with $\sigma = 0.02$ (*right*).

The results for Duffing II are unfortunately not as promising as for the previous set of parameters when directly applied, as seen left sub-figure of Fig. 48 where $\sigma = 0.34$ is used. While in general the divisions between the partitions has become clearer, there are still far too many isolated points being mislabelled to make a clear judgement of the boundaries of the generating partition.

By adding the additional step of first normalising the spatial coordinates to a (shifted) unit square before applying the method, we get the more promising results shown in the right sub-figure with $\sigma = 0.02$. In this case, there are only a very small handful of isolated points, with the boundaries between the partitions being extremely clear.

3.8 Approximate Generating Partitions

In order to more clearly visualise the partitioning of the attractor provided by the modified DLBD algorithm it's possible to use the proximity kernel in conjunction with the resultant symbolic encoding of each periodic point in order to plot the *basin* of each symbol. By calculating the Z-score of each point in (and around) the attractor and assigning to each point the symbol with the highest score, we can get a clearer view of the partition boundaries of each of the maps.

In the case of the attractor of the Ikeda map, by using $\sigma = 0.05$ we get the following partitioning:

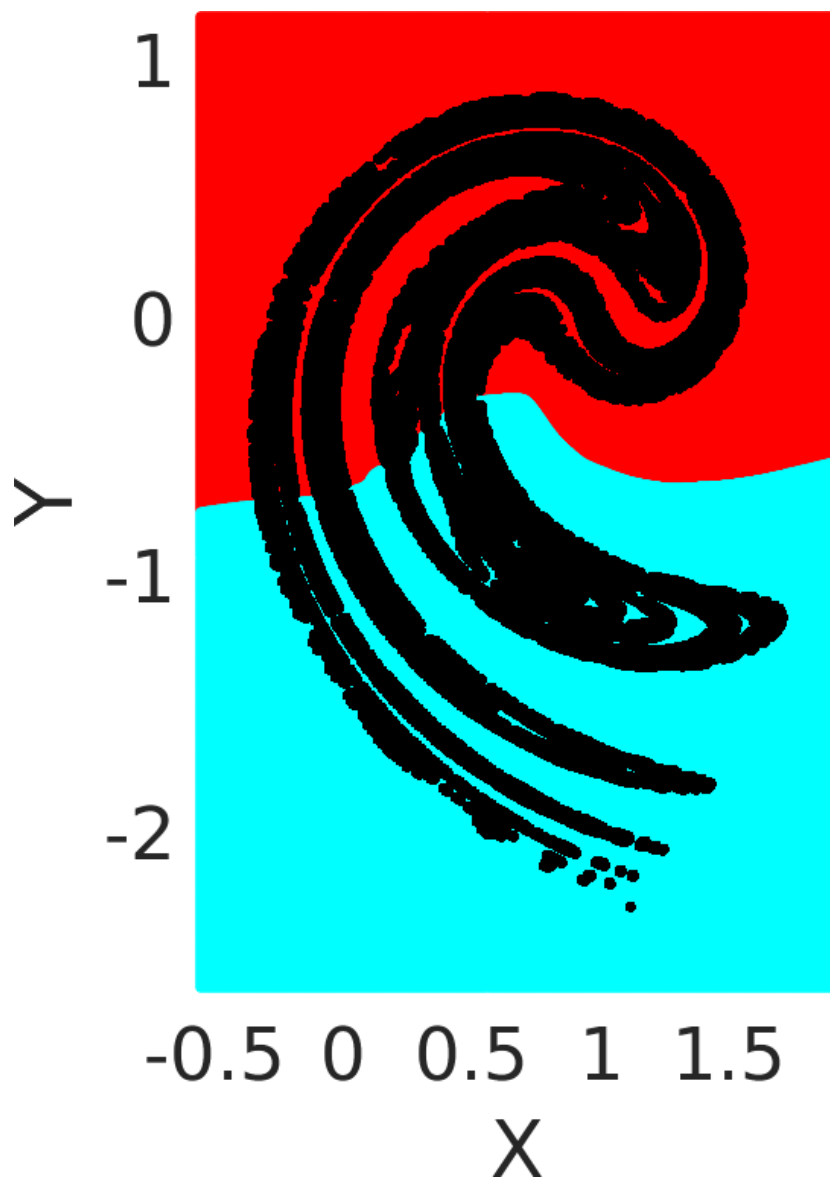


Figure 49: Partitioning of the Ikeda attractor using $\sigma = 0.05$.

In the case of the Hénon I attractor using $\sigma = 0.09$, we get the following partitioning:

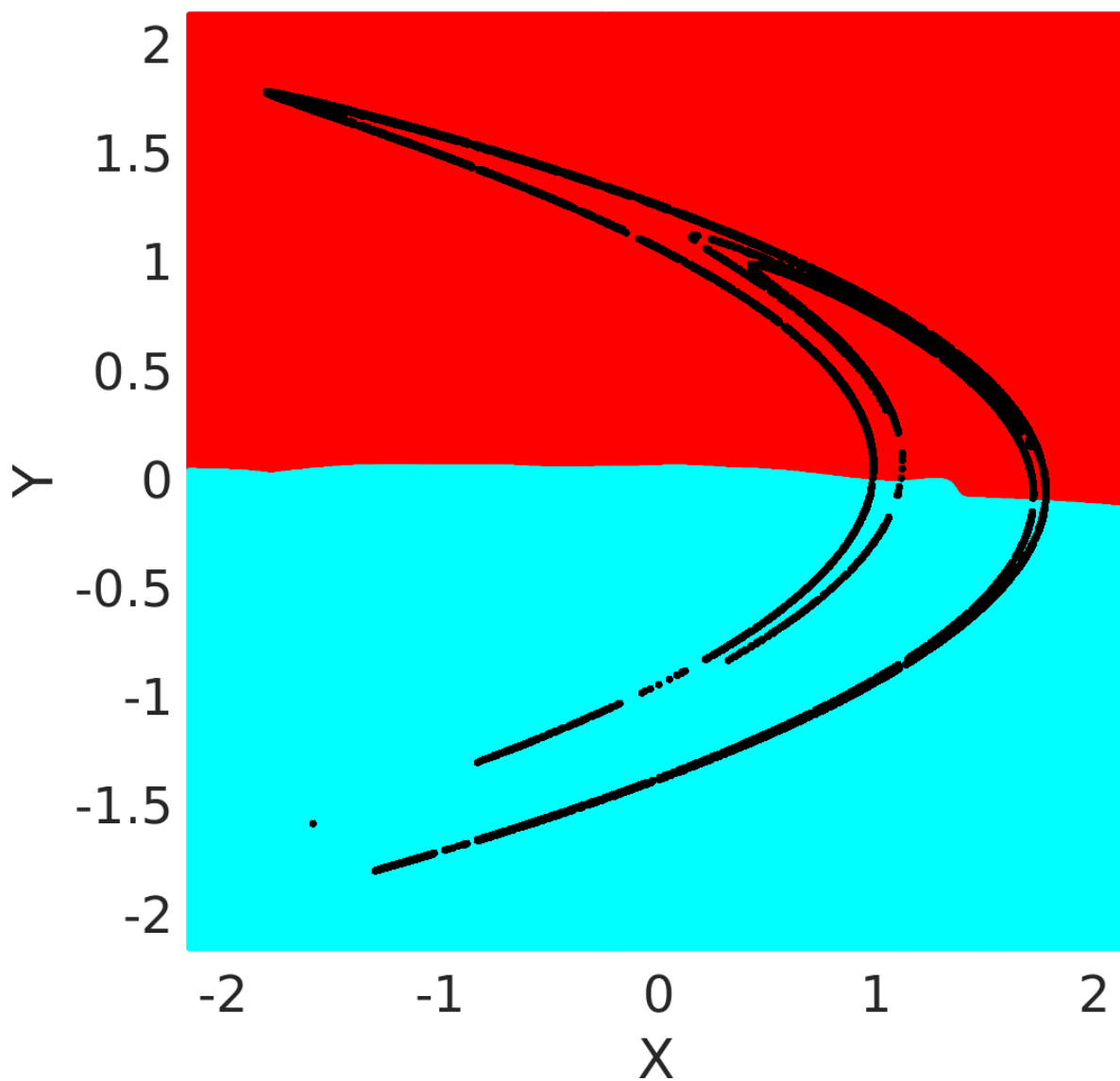


Figure 50: Partitioning of the Hénon I attractor using $\sigma = 0.09$.

In the case of Hénon II attractor with $\sigma = 0.07$:

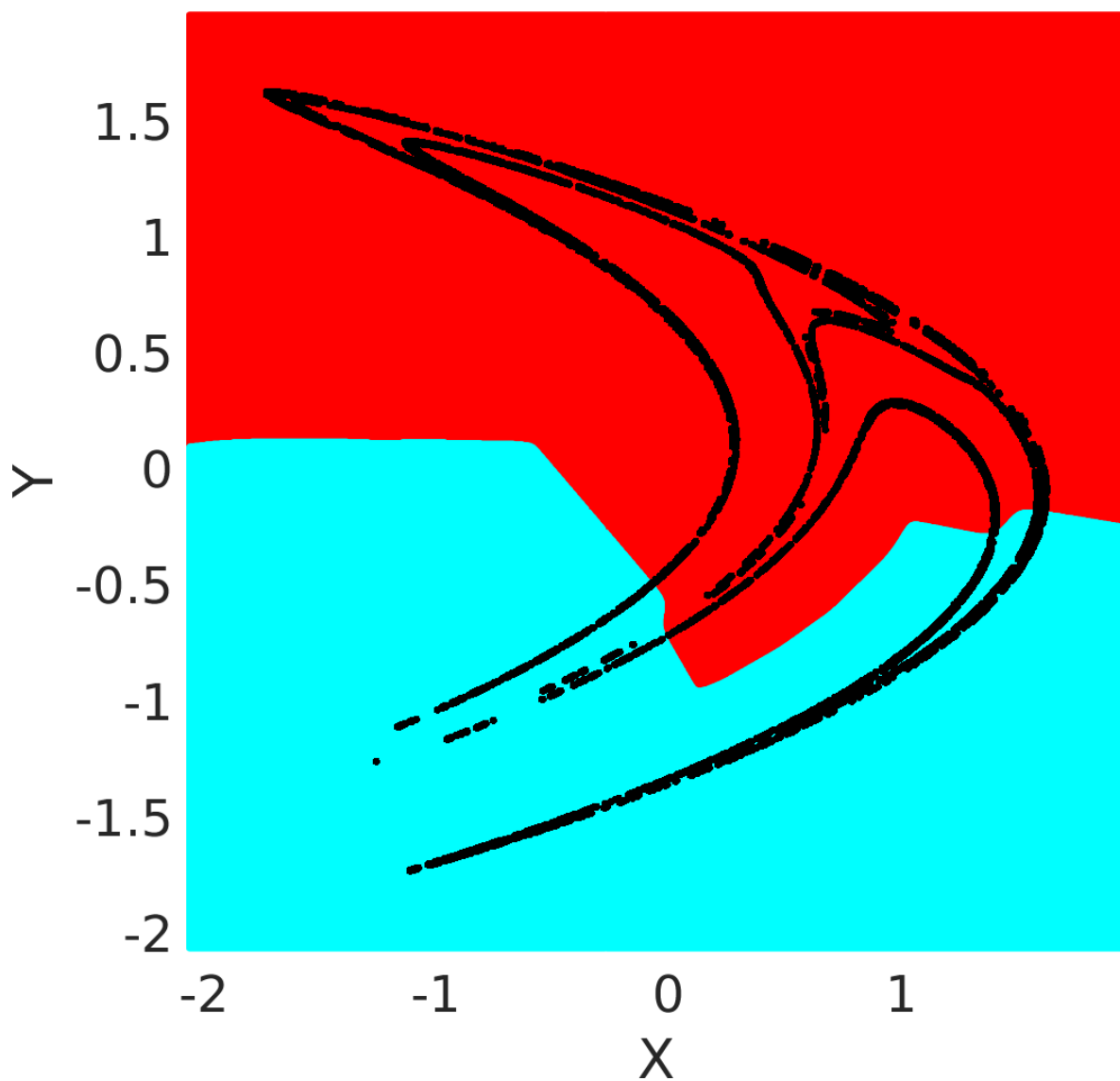


Figure 51: Partitioning of the Hénon II attractor using $\sigma = 0.07$.

In the case of Duffing I with $\sigma = 0.02$:

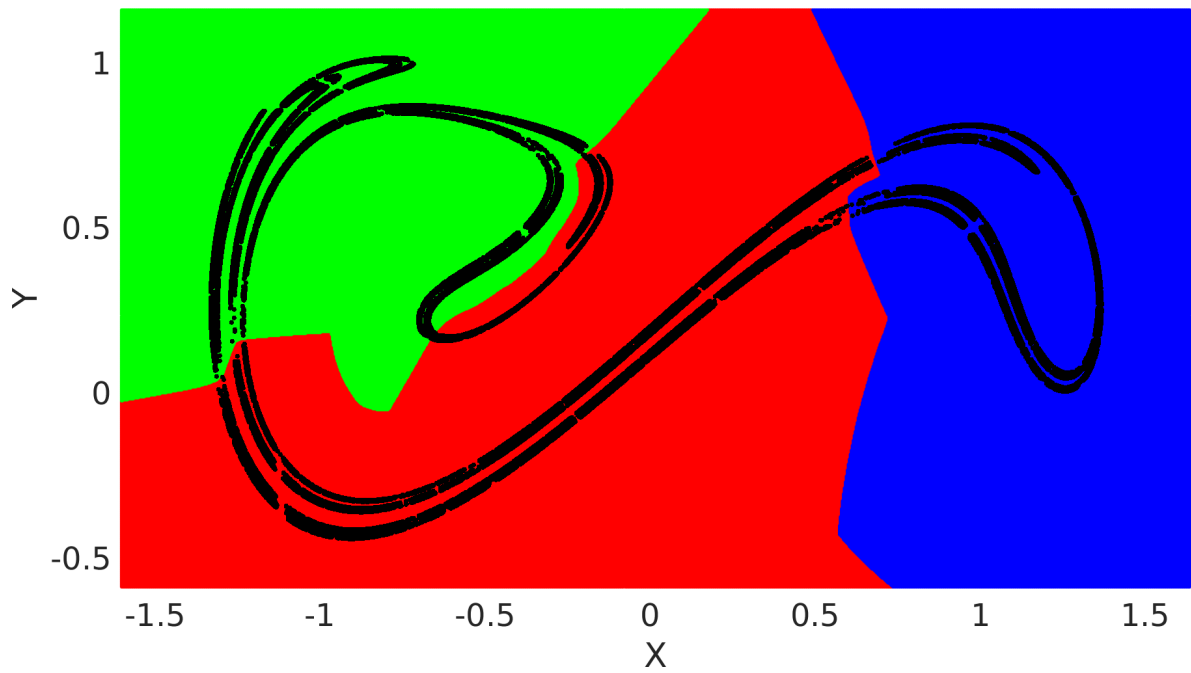


Figure 52: Partitioning of the Duffing I attractor using $\sigma = 0.02$.

And Duffing I with the alternative parameter $\sigma = 0.03$:

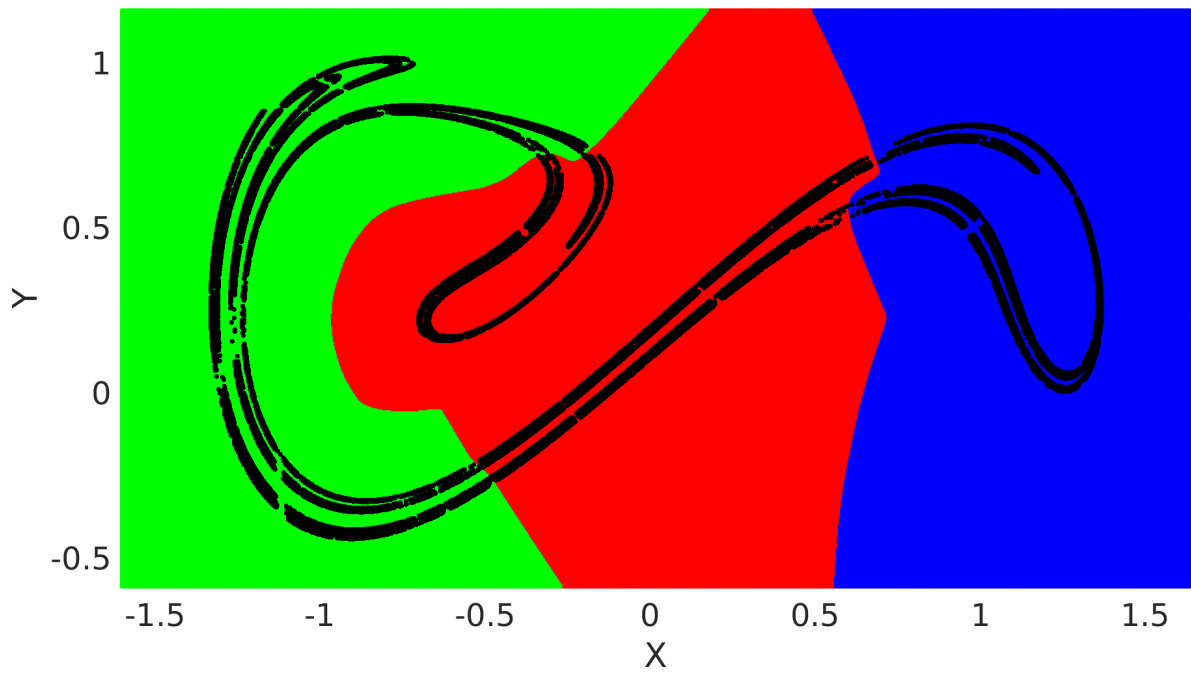


Figure 53: Partitioning of the Duffing I attractor using $\sigma = 0.03$.

Finally, in the case of Duffing II using normalised co-ordinates with the parameter $\sigma = 0.02$:

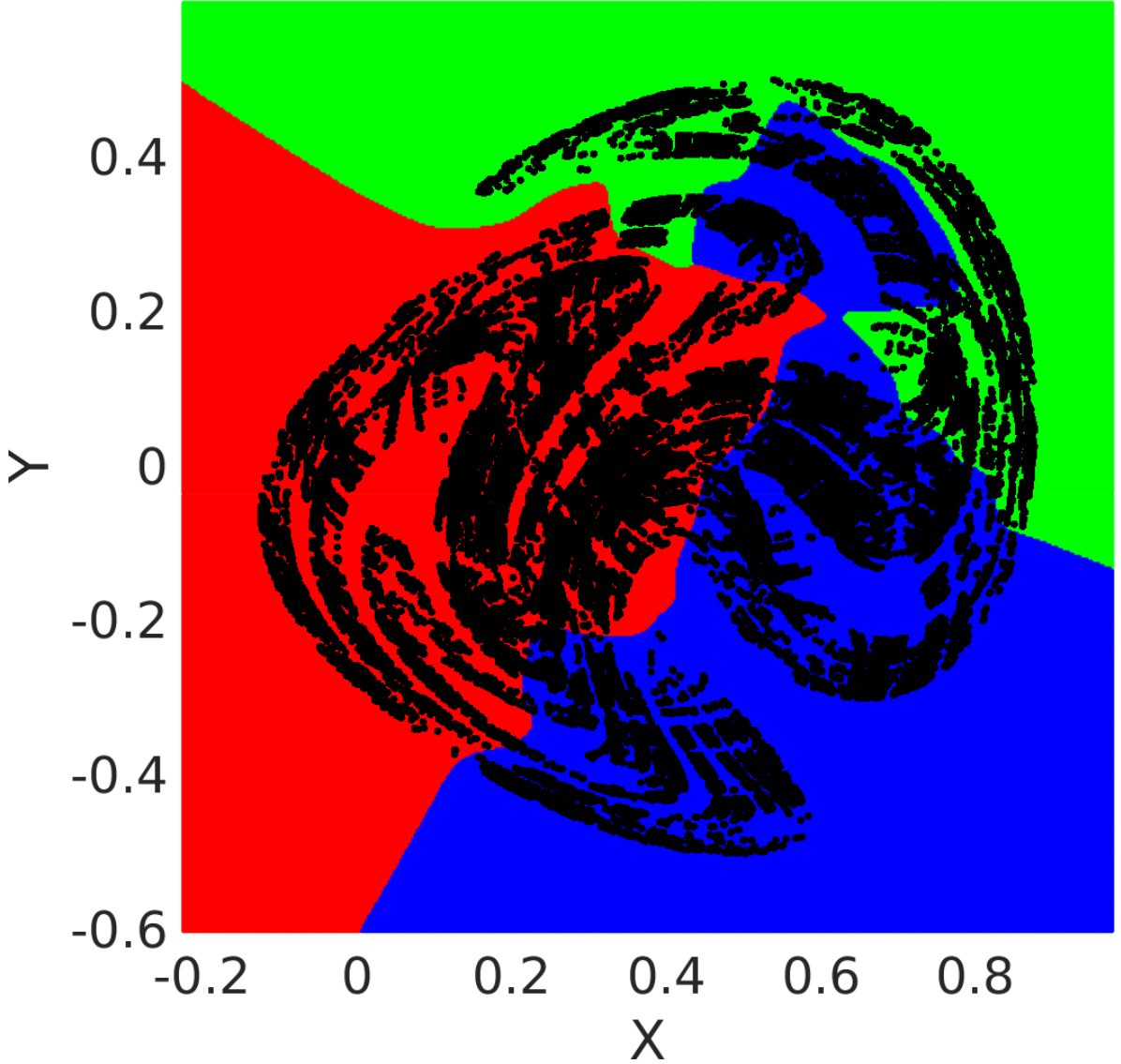


Figure 54: Partitioning of the Duffing II attractor using $\sigma = 0.02$ with spatial co-ordinates normalised to a unit square.

Until now, no partitioning of the Duffing II attractor has been given, and so we present this 3-symbol partition as a new result.

3.9 Standard Map

In this section we present the results of the implementation of the above methods in order to attempt to locate a generating partition of the standard map which Christiansen

& Politi discussed in their 1995 paper *Generating partition for the standard map*[59]. Using the method of homoclinic tangencies they found a partition with seven symbols using which all periodic orbits (in their set of approximately 30,000) up to period 9 were uniquely coded, with the exception of one period 6 orbit and four period 8 orbits.

The standard map is formulated by Christiansen & Politi as follows:

$$\begin{aligned}x_{n+1} &= y_n \\y_{n+1} &= -x_n + 2y_n - \alpha \cos(y_n) \bmod 2\pi\end{aligned}\tag{3.12}$$

with parameter $\alpha = 6$. As we wish to present our results in the framework, we will also assume this parameter value; additionally, we will also consider periodic orbits up to period 9, however our set of periodic orbits contains 50,499 orbits - approximately 20,000 more orbits than Christiansen & Politi's set.

In Eq. 3.12, the modulo 2π results in the map being periodic with period 2π in both the x & y directions. Before applying our method, we must be sure to account for this periodicity in the distance function. This can be done by making a slight modification to the calculation of the norm:

$$|\mathbf{x} - \mathbf{x}_k|^2 = \sum_i (\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)})^2\tag{3.13}$$

where $\mathbf{x}^{(i)}$ corresponds to the i^{th} component of the vector \mathbf{x} . In the case where any $\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)} > \pi$, the shortest distance between the two components can be found by instead using the symmetry-related point $\mathbf{x}^{(i)} - 2\pi$ - meaning that in this case one should replace the above with $\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)} - 2\pi$. Similarly, in the case where $\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)} < -\pi$, the component's sum should be increased by 2π , resulting in the new sum: $\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)} + 2\pi$.

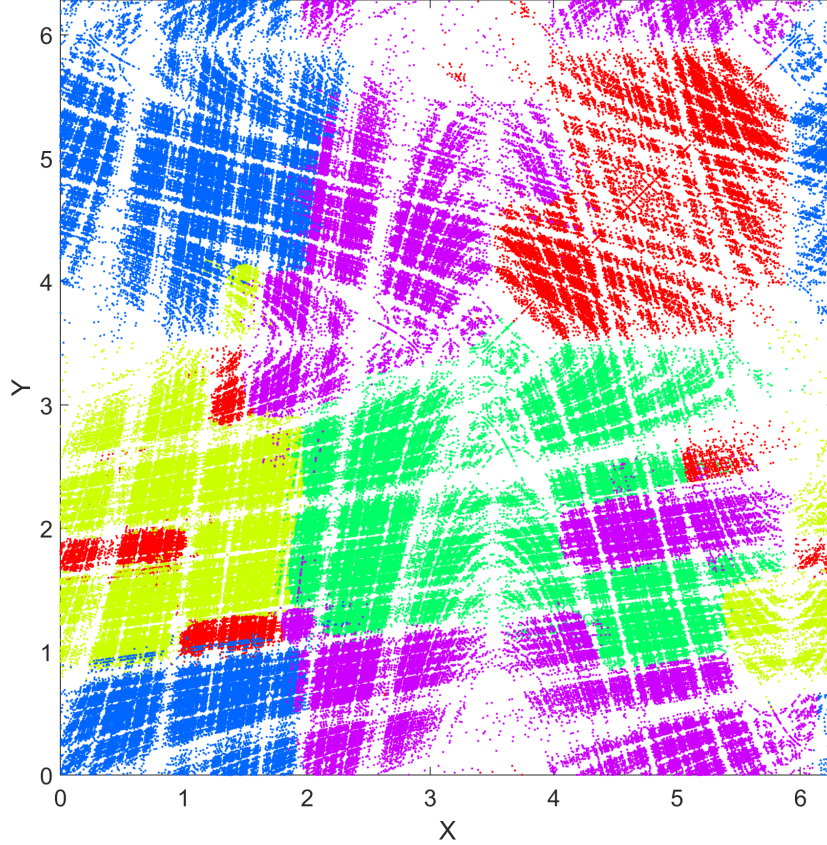


Figure 55: Resulting symbol sets using $\sigma = 0.021$.

By applying this tweak to the distance \hat{Z} of Eq. 3.11, we can safely apply the above staggered method to the set of 50,499 periodic orbits of this map, ranging from period 1 to 9. After some trial and error, we found that the value $\sigma = 0.021$ gives the best overall results. Using this method with $\sigma = 0.021$ we get an encoding that uses only 5 symbols (compared to Christiansen & Politi's[59] 7 symbols) to uniquely encode **all** 50,042 orbits up to period 9. Unfortunately, the results of our method don't provide such clear partition boundaries at this stage, but at the very least we see the main partition boundaries starting to form (Figs. 55 & 56). We hope that with further research the method will be able to provide as clear partitions for the standard map as it does for those discussed previously.

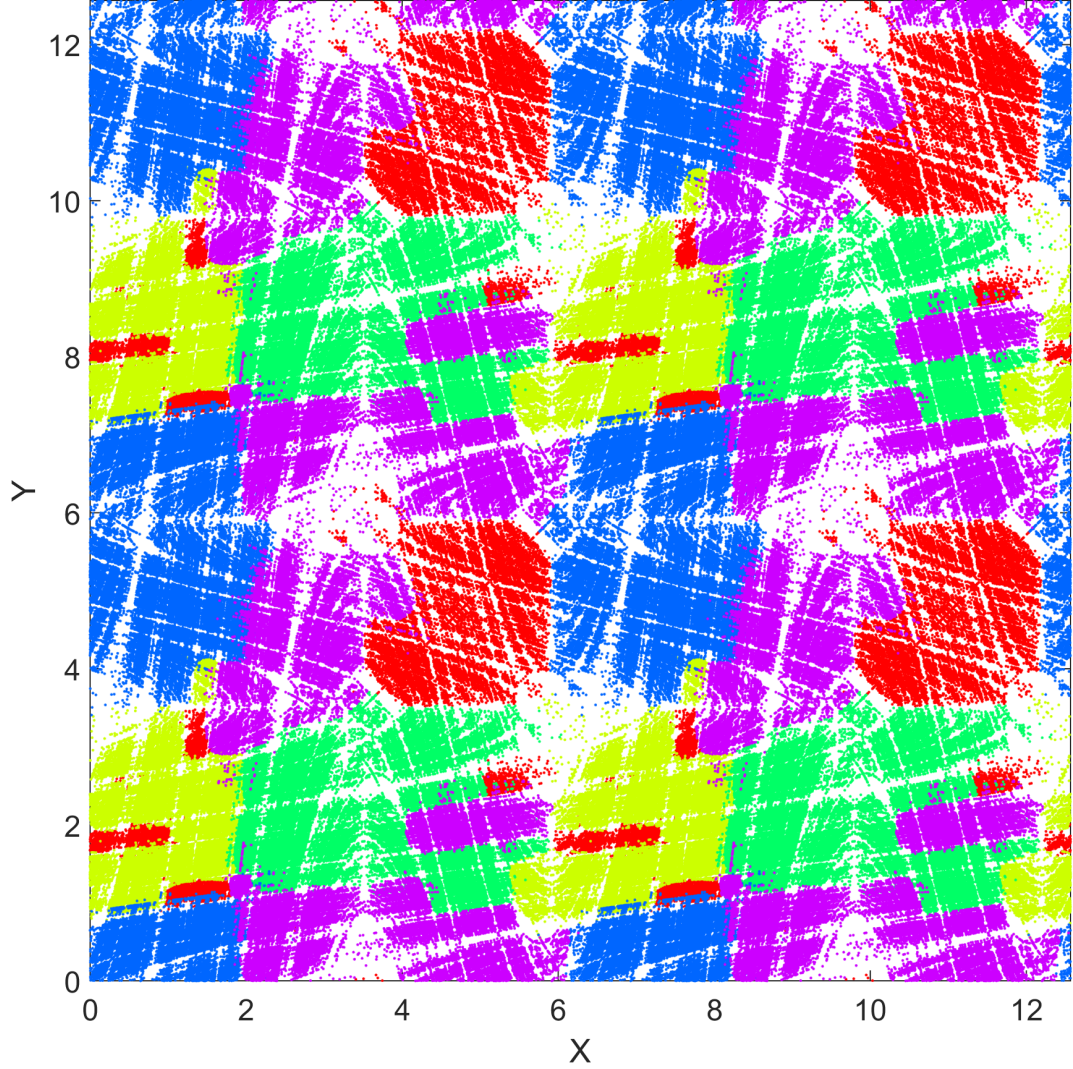


Figure 56: Resulting symbol sets using $\sigma = 0.021$; the original $[0, 2\pi]^2$ cell has been tiled to create a 2×2 grid of size $[0, 4\pi]^2$, allowing us to see the behaviour across the periodic boundaries.

3.10 Summary & Future Work

In this part of the thesis we first presented the method of approximation of generating partitions using periodic orbits developed by Davidchack et al.[26], and highlighted some limitations & pitfalls of the method, namely the inability to add more symbols when necessary, and drawbacks of the proximity kernel Z of Eq. 3.5.

The inability to add new symbols meant that this method could only effectively be

applied to systems whose required number of symbols was known *a priori* - an extremely prohibitive limitation. In order to resolve this issue we added an extra step to the DLBD method which, before moving on to the next period, calculated the number of symbols required by calculating the possible number of circular permutations with repetition using the current number of symbols - if this number is smaller than the number of UPOs of the given period, more symbols are added until the issue is resolved. This addition allows the DLBD method to be applied to a much broader set of systems, which we demonstrated by applying the method to the Duffing map for two different sets of parameters.

In the case of the attractors of the Hénon & Ikeda maps presented in Davidchack et al.'s paper[26], the symbol sets given by the method are (as one would hope) completely connected with no isolated points or regions. Unfortunately, in the case of the Hénon map for the second set of parameters and the Duffing map for both sets of parameters, this wasn't the case; many isolated points and regions were observed, resulting in symbol sets that weren't completely connected.

In an attempt to resolve this we experimented with a slight modification of the proximity function Z in which we normalise the Z -value of a point to a given symbol set by the number of points with that symbol, we called this new function \hat{Z} . This normalisation was done in order to stop the symbol sets that were assigned more points in the early periods from expanding far beyond the region of eventual convergence of the partition boundary, which often led to the creation of isolated points and regions. This new proximity function saw success with the Hénon II and Duffing I attractors, resulting in symbol sets that were almost completely connected, with only one or two points being mislabelled.

In order to deal with these mislabelled points, we introduced a *staggered* method of construction, which (roughly speaking) repeats the DLBD method for each period until convergence before continuing. The aim of this method was to fix mislabelled points

in early periods before they have the chance to propagate, causing more points to be mislabelled later on. In the case of Hénon II and Duffing I, all of the mislabelled points were indeed corrected, as hoped.

In the case of Duffing II, neither the use of \hat{Z} nor the use of the staggered method displayed good results. As such, we decided to try a new proximity function that focused on more local behaviour. To this end, we chose to use the Gaussian-inspired proximity function \tilde{Z} of Eq. 3.11 with which the proximity score exponentially decays with increasing distance. Combining this new function with the staggered method resulted in further improvements to Duffing I, with two different generating partitions being given for different parameter values σ of \tilde{Z} .

Duffing II however didn't initially see such great success with the new proximity function \tilde{Z} in combination with the staggered method. By taking the additional step of first normalising the spatial coordinates such that the attractor lies within a unit square before applying these methods resulted in a very well-connected set of distinct symbol regions, by which an approximate generating partition could be inferred. Until now, no generating partition for this attractor has been given, and so we present this 3-letter symbolic encoding as a new & original result.

As a final step, we applied this method to the Standard map, after first ensuring that the norm $|\cdot|$ accounted for the periodic boundaries. In the paper by Christiansen & Politi[59], they give a 7-letter encoding of the attractor. Interestingly, the application of our method results in an encoding that requires only 5 symbols; unfortunately however, our results have several mislabelled points & isolated regions.

Our immediate plans for future research in this direction will primarily focus on improving the partitioning provided for the standard map, with the hope of revealing a new partition requiring only 5 symbols. To this end we will focus our research on alternative proximity kernels, further tweaking the staggered method, and developing a *second pass*

that will go through the symbol sets obtained through the staggered method once more and try to further reduce the number of mislabelled points.

Another area of focus will be the application of our method to higher-dimensional chaotic attractors. Despite the fact that all of the examples presented in this section were of 2-dimensions, given a set of periodic orbits this method can be applied without modification to maps of any dimension; as seen in this section, however, further changes may need to be made in order to ensure that the partitions given for such maps are as good as one would hope (for example to the proximity kernel).

In the long term, we plan on adapting the methods presented by Patil & Cusumano[62] for use with embedded coherent structures. Their method revolves around applying Hirata et al.'s[57] symbolic shadowing method to a large set of initially random partitions, and then choosing the partition that minimises the so-called Lempel-Ziv complexity[63] in order to select the *best* partition. By implementing a similar process to our algorithm, it may be possible to further improve the resulting partitions for more complicated attractors.

4 Concluding Remarks

Throughout this thesis we presented two methods for the coarse-graining of chaotic attractors into partitions using unstable periodic orbits and other coherent structures.

The first method we introduced involved the construction of a finite cover of chaotic attractors, and was presented in the context of discrete-time maps (Chapter 1), and high-dimensional continuous-time flows with symmetries (Chapter 2). By constructing such a cover, we showed that it's possible to reduce the dynamics of trajectories in the attractor to a series of transitions between elements of the cover, known as *shadowing*. By then performing shadowing on a long trajectory segment in the attractor, we were able to construct a Markov chain representation of the dynamics of the system under the given shadowing regime.

As this method makes no usage of the periodicity of the structures being used to create the cover, it's equally applicable to any system for which there exists a large enough data set of trajectories, including data for which the underlying equations determining the dynamics are unknown. This opens the possibility for the method to not only be used with chaotic dynamical systems in an abstract sense, but also with experimental data of potentially high-dimensional processes.

Future research could focus on resolving some of the limitations of this method. For example, due to the redundancies in the cover when performing shadowing at each point there are many options to choose from, and so the dynamics encoded within the Markov chain may be heavily influenced by different choices of shadowing regime. Additionally, in the case of flows, choosing the n leading Fourier modes for the n -dimensional projection may not be the optimal choice of projection; it may be preferable to instead use other dimensionality reduction techniques such as principal component analysis or linear discriminant analysis in order to choose the directions of projection. If this is the case, then it may be possible to construct better covers with fewer dimensions in

the projection, potentially increasing the accuracy whilst also decreasing computation times. Further research into the Markov chain representation of the dynamics is also needed, beyond the toy examples given in the concluding sections of Chapters 1 & 2; if constructed correctly, the Markov chain representation could become a powerful tool for determining various properties of the underlying systems.

The second method we presented (Chapter 3) was an extension of the algorithm of Davidchack et al.[26] which constructed approximate generating partitions of chaotic attractors of discrete-time dynamical systems using unstable periodic orbits. By introducing a series of modifications to this method we removed the necessity of pre-determining the number of partitions and also significantly improved the results for more complex maps. One of the key results of this section was the construction of a well-connected approximation of a generating partition of the so-called Duffing II map, for which until now to the best of our knowledge no generating partition has been found. Following this, we applied our method to the Standard map for which an approximate generating partition was found by Christiansen & Politi[59], for which we had mixed results; our partition successfully encoded all orbits up to period 20 unlike that of Christiansen & Politi[59] using 2 less symbols, however our partitions weren't well-connected, with many isolated points and regions.

By dedicating further research into different distance functions or other modifications of the method it should be possible to further improve the approximation given for systems such as the Standard map, for which finding a connected approximate generating partition using only 5 symbols would be a significant discovery. Since the presented method does not rely on the two-dimensional nature of the chaotic attractors, it can be equally well applied to higher-dimensional maps, which could be another direction of future research.

Two examples of related fields that it may be interesting to apply the methods presented in this thesis to are chaos control and random number generation.

In terms of chaos control, the methods of Chapters 1 & 2 may be a good fit. The control of chaos revolves around the stabilisation of a trajectory in the system by locating a nearby unstable periodic orbit to the trajectory, and applying perturbations (either in discrete intervals[64] or continuously[65]) to the control parameter of the system such that the unstable periodic orbit becomes stable, resulting in the trajectory being pulled to the newly-stable periodic orbit. By using such a cover it may be possible to simplify the process of finding the closest unstable periodic orbit for control by using the representative subset provided by the cover in a lower-dimensional projection of the full space.

The fast generation of random numbers is extremely important for many applications, notably in cryptography and Monte-Carlo methods. Recently, the use of chaos in the generation of random numbers has received much attention from the research community. A brief review of the use of chaos to generate random numbers is given by Stojanovski et al.[66], along with their own generation method. While clearly not a new idea, we believe that by using such a cover as in Chapter 1 & 2, or by using the symbolic dynamics of Chapter 3 it may be possible to find a new, novel way of generating random numbers.

Additionally, in the field of chaotic dynamics & periodic orbit theory itself, by carefully selecting a representative subset of unstable periodic orbits of the system it may be possible to find accurate estimates for properties of the attractor (such as the Lyapunov exponents, fractal dimension, and natural measure) using this small subset of periodic orbits rather than having to find and use **all** of them up to a given period, as in [67].

Bibliography

- [1] George D Birkhoff. On the periodic motions of dynamical systems. *Acta Mathematica*, 50(1):359–379, 1927.
- [2] Robert L Devaney et al. *An introduction to chaotic dynamical systems*, volume 13046. Addison-Wesley Reading, 1989.
- [3] Predrag Cvitanović. Periodic orbits as the skeleton of classical and quantum chaos. *Physica D: Nonlinear Phenomena*, 51(1):138–151, 1991.
- [4] Jonathan Crofts and Ruslan Davidchack. Efficient detection of periodic orbits in chaotic systems by stabilizing transformations. *SIAM Journal of Scientific Computing*, 28(4), 2006.
- [5] Predrag Cvitanović. Dynamical averaging in terms of periodic orbits. *Physica D: Nonlinear Phenomena*, 83(1):109–123, 1995.
- [6] Predrag Cvitanovic. Invariant measurement of strange sets in terms of cycles. *Phys. Rev. Lett.*, 61:2729, 1988.
- [7] Itamar Procaccia. Complex or just complicated? *Nature*, 333:498–499, 1988.
- [8] Yueheng Lan and Predrag Cvitanović. Unstable recurrent patterns in kuramoto-sivashinsky dynamics. *Physical review E*, 78(2):026208, 2008.
- [9] Vanessa López, Philip Boyland, Michael T Heath, and Robert D Moser. Relative periodic solutions of the complex ginzburg–landau equation. *SIAM Journal on Applied Dynamical Systems*, 4(4):1042–1075, 2005.
- [10] Björn Hof, Casimir WH van Doorne, Jerry Westerweel, Frans TM Nieuwstadt, Holger Faisst, Bruno Eckhardt, Hakan Wedin, Richard R Kerswell, and Fabian Waleffe. Experimental observation of nonlinear traveling waves in turbulent pipe flow. *Science*, 305(5690):1594–1598, 2004.
- [11] Holger Faisst and Bruno Eckhardt. Traveling waves in pipe flow. *Physical Review Letters*, 91(22):224502, 2003.

- [12] Hakan Wedin and RR Kerswell. Exact coherent structures in pipe flow: travelling wave solutions. *Journal of Fluid Mechanics*, 508:333–371, 2004.
- [13] Scott M Zoldi and Henry S Greenside. Spatially localized unstable periodic orbits of a high-dimensional chaotic system. *Physical Review E*, 57(3):R2511, 1998.
- [14] Predrag Cvitanović, Ruslan L Davidchack, and Evangelos Siminos. On the state space geometry of the kuramoto-sivashinsky flow in a periodic domain. *SIAM Journal on Applied Dynamical Systems*, 9(1):1–33, 2010.
- [15] Ioannis G Kevrekidis, Basil Nicolaenko, and James C Scovel. Back in the saddle again: a computer assisted study of the kuramoto-sivashinsky equation. *SIAM Journal on Applied Mathematics*, 50(3):760–790, 1990.
- [16] P Schmelcher and FK Diakonov. Detecting unstable periodic orbits of chaotic dynamical systems. *Physical Review Letters*, 78(25):4733, 1997.
- [17] Jonathan J Crofts and Ruslan L Davidchack. On the use of stabilizing transformations for detecting unstable periodic orbits in high-dimensional flows. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(3):033138, 2009.
- [18] Ruslan L Davidchack and Ying-Cheng Lai. Efficient algorithm for detecting unstable periodic orbits in chaotic systems. *Physical Review E*, 60(5):6172, 1999.
- [19] Ruslan L Davidchack, Ying-Cheng Lai, Aaron Klebanoff, and Erik M Bollt. Towards complete detection of unstable periodic orbits in chaotic systems. *Physics Letters A*, 287(1):99–104, 2001.
- [20] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993.
- [21] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 566–568. IEEE, 1994.

- [22] Oliver Jesorsky, Klaus J Kirchberg, and Robert W Frischholz. Robust face detection using the hausdorff distance. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 90–95. Springer, 2001.
- [23] Mark Anthony Armstrong. *Basic topology*. Springer Science & Business Media, 2013.
- [24] Kensuke Ikeda. Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics communications*, 30(2):257–261, 1979.
- [25] SM Hammel, JV Moloney, and CKRT Jones. Global dynamical behavior of the optical field in a ring cavity. *JOSA B*, 2(4):552–564, 1985.
- [26] Ruslan L Davidchack, Ying-Cheng Lai, Erik M Bollt, and Mukeshwar Dhamala. Estimating generating partitions of chaotic systems by unstable periodic orbits. *Physical Review E*, 61(2):1353, 2000.
- [27] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.
- [28] Daniel T Gillespie. *Markov processes: an introduction for physical scientists*. Elsevier, 1991.
- [29] C Nicolis, W Ebeling, and C Baraldi. Markov processes, dynamic entropies and the statistical prediction of mesoscale weather regimes. *Tellus A*, 49(1):108–118, 1997.
- [30] B Ramachandran. On the “strong memorylessness property” of the exponential and geometric probability laws. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 244–251, 1979.
- [31] M Cencini, M Falcioni, E Olbrich, H Kantz, and Angelo Vulpiani. Chaos or noise: Difficulties of a distinction. *Physical Review E*, 62(1):427, 2000.
- [32] Holger Kantz, Wolfram Just, Nilüfer Baba, Katrin Gelfert, and Anja Riebert. Fast chaos versus white noise: entropy analysis and a fokker–planck model for the slow dynamics. *Physica D: Nonlinear Phenomena*, 187(1):200–213, 2004.

- [33] C Nicolis. Chaotic dynamics, markov processes and climate predictability. *Tellus A*, 42(4):401–412, 1990.
- [34] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.
- [35] JP Jarvis and Douglas R Shier. Graph-theoretic analysis of finite markov chains. *Applied mathematical modeling: a multidisciplinary approach*, 1999.
- [36] John G Kemeny, James Laurie Snell, et al. *Finite markov chains*, volume 356. van Nostrand Princeton, NJ, 1960.
- [37] R Diego. On markov chains, attractors, and neural nets. *Complex Systems*, 12: 339–355, 2000.
- [38] Alicia Juarrero. *Dynamics in action: Intentional behavior as a complex system*. MIT press Cambridge, MA, 1999.
- [39] Yoshiki Kuramoto and Toshio Tsuzuki. Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Progress of theoretical physics*, 55(2):356–369, 1976.
- [40] GI Sivashinsky. Large cells in nonlinear marangoni convection. *Physica D: Non-linear Phenomena*, 4(2):227–235, 1982.
- [41] Philipp Metzner, Evelyn Dittmer, Tobias Jahnke, and Ch Schütte. Generator estimation of markov jump processes. *Journal of Computational Physics*, 227(1): 353–375, 2007.
- [42] DT Crommelin and Eric Vanden-Eijnden. Fitting timeseries by continuous-time markov chains: A quadratic programming approach. *Journal of Computational Physics*, 217(2):782–805, 2006.
- [43] Daan Crommelin and Eric Vanden-Eijnden. Data-based inference of generators for markov jump processes using convex optimization. *Multiscale Modeling & Simulation*, 7(4):1751–1778, 2009.

- [44] I Holmes and GM Rubin. An expectation maximization algorithm for training hidden substitution models. *Journal of molecular biology*, 317(5):753–764, 2002.
- [45] Mogens Bladt and Michael Sørensen. Statistical inference for discretely observed markov jump processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):395–410, 2005.
- [46] C Stuart Daw, Charles Edward Andrew Finney, and Eugene R Tracy. A review of symbolic analysis of experimental data. *Review of Scientific Instruments*, 74(2):915–930, 2003.
- [47] Peter Grassberger and Holger Kantz. Generating partitions for the dissipative hénon map. *Physics Letters A*, 113(5):235–238, 1985.
- [48] John Guckenheimer and Philip J Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer Science & Business Media, 2013.
- [49] Michael Eisele. Comparison of several generating partitions of the hénon map. *Journal of Physics A: Mathematical and General*, 32(9):1533, 1999.
- [50] Ethan M Coven and Zbigniew H Nitecki. On the genesis of symbolic dynamics as we know it. *arXiv preprint math/0611322*, 2006.
- [51] Jacques Hadamard. Les surfaces à courbures opposées et leurs lignes géodésique. *J. Math. Pures Appl.*, 4:27–73, 1898.
- [52] Harold Marston Morse. Recurrent geodesics on a surface of negative curvature. *Transactions of the American Mathematical Society*, 22(1):84–100, 1921.
- [53] Marston Morse and Gustav A Hedlund. Symbolic dynamics. *American Journal of Mathematics*, 60(4):815–866, 1938.
- [54] Franco Giovannini and Antonio Politi. Homoclinic tangencies, generating partitions and curvature of invariant manifolds. *Journal of Physics A: Mathematical and General*, 24(8):1837, 1991.

- [55] R Badii, E Brun, M Finardi, L Flepp, R Holzner, J Parisi, C Reyl, and J Simonet. Progress in the analysis of experimental chaos through periodic orbits. *Reviews of Modern Physics*, 66(4):1389, 1994.
- [56] M Lefranc, P Glorieux, F Papoff, F Molesti, and E Arimondo. Combining topological analysis and symbolic dynamics to describe a strange attractor and its crises. *Physical review letters*, 73(10):1364, 1994.
- [57] Yoshito Hirata, Kevin Judd, and Devin Kilminster. Estimating a generating partition from observed time series: Symbolic shadowing. *Physical Review E*, 70(1):016215, 2004.
- [58] Michel Hénon. A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, 50(1):69–77, 1976.
- [59] Freddy Christiansen and Antonio Politi. Generating partition for the standard map. *Physical Review E*, 51(5):R3811, 1995.
- [60] PA MacMahon. Memoir on the theory of the compositions of numbers. *Proceedings of the Royal Society of London*, 52(315-320):290–294, 1892.
- [61] Kenneth H Rosen. *Handbook of discrete and combinatorial mathematics*. CRC press, 1999.
- [62] Navendu Patil and Joseph Cusumano. Empirical generating partitions of driven oscillators using optimized symbolic shadowing. 2016. (Unpublished manuscript).
- [63] Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Transactions on information theory*, 22(1):75–81, 1976.
- [64] Edward Ott, Celso Grebogi, and James A Yorke. Controlling chaos. *Physical review letters*, 64(11):1196, 1990.
- [65] Kestutis Pyragas. Continuous control of chaos by self-controlling feedback. *Physics letters A*, 170(6):421–428, 1992.

- [66] Toni Stojanovski and Ljupco Kocarev. Chaos-based random number generators-part i: analysis [cryptography]. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(3):281–288, 2001.
- [67] P. Cvitanović, R. Artuso, R. Mainieri, G. Tanner, and G. Vattay. *Chaos: Classical and Quantum*. Niels Bohr Institute, Copenhagen, 2012. ChaosBook.org.