Available online at www.sciencedirect.com
ScienceDirect
Physica D 231 (2007) 95–107
PHYSICA D
www.elsevier.com/locate/physd

Identifying stochastic basin hopping by partitioning with graph modularity

N. Santitissadeekorn*, E.M. Bollt

Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699-5815, USA

Received 4 March 2006; received in revised form 7 April 2007; accepted 23 April 2007
Available online 7 May 2007

Communicated by C.K.R.T. Jones

Abstract

It has been known that noise in a stochastically perturbed dynamical system can destroy what was the original zero-noise case barriers in the phase space (pseudobarrier). Noise can cause the basin hopping. We use the Frobenius–Perron operator and its finite rank approximation by the Ulam–Galerkin method to study transport mechanism of a noisy map. In order to identify the regions of high transport activity in the phase space and to determine flux across the pseudobarriers, we adapt a new graph theoretical method which was developed to detect active pseudobarriers in the original phase space of the stochastic dynamic. Previous methods to identify basins and basin barriers require a priori knowledge of a mathematical model of the system, and hence cannot be applied to observed time series data of which a mathematical model is not known. Here we describe a novel graph method based on optimization of the modularity measure of a network and introduce its application for determining pseudobarriers in the phase space of a multi-stable system only known through observed data.

Keywords: Multistability; Graph modularity; Noise; Basin hopping; Frobenius–Perron operator; Stochastic; Ulam's method; Phase space partition

1. Introduction

Noise in deterministic dynamic systems can play an important role in a global change in the system. The stochastic dynamics, can be dramatically different from the unperturbed deterministic analogue, even under small noise inputs. For example, additive noise in a multi-stable system can cause basin hopping between two states [1–6]. The questions of when noise induced chaos, phase space chaotic transport and mean escape times from certain basins of attractions are pressing for a given stochastic dynamic system. These questions lead us to find a way to determine basin boundaries of stochastic systems. Bollt et al. [1] have previously developed a technique to compute phase space transport without a priori knowledge of the basin boundaries based on the Ulam's method coupled with graph theory. This technique uses the Ulam matrix to approximate the Frobenius–Perron operator by a Markov operator of finite rank. In this approach the boxes covering the phase space are reindexed according to their convergence rate to a fixed point in a basin. However, this technique cannot be used if we only have experimental data, since the model is not available for certain key steps in the Ulam–Galerkin approximation. Here, we use a graph theoretical approach to reindex the matrix to identify regions that should be dynamically grouped (those in the same basin). The main advance with this new approach is that we can identify appropriate regions of phases space to call dynamically similar, and hence the transport barriers between them and corresponding transport between, without an a priori model to work with, but instead using only observed orbit data.

In a different approach, recent efforts have focused on identifying the number and location of *almost invariant* sets; those subsets of a state space where trajectories tend to stay for comparatively long periods of time before they leave into other regions. *Set oriented* methods where developed to numerically study the problem of identifying the regions [7–10]. In this language, we can say that a basin in a multi-stable system subject to stochastic noise can be considered as an almost invariant set. In the cited work, the state space of a dynamic system is first discretized into boxes. Then the *congestion* of a graph based on a multicommodity flow on the graph is employed to partition the graph to approximate the almost invariant sets. The idea of this technique is that edges with low congestion can be detected in almost invariant sets, whereas

* Corresponding author.
E-mail address: santitn@clarkson.edu (N. Santitissadeekorn).

0167-2789/$ - see front matter
doi:10.1016/j.physd.2007.04.008

those with high congestion are edges between the almost invariant sets. In the language of this paper, using notions of recent graph theory developments, we will say that those nodes in the graph representing boxes covering an almost invariant set of a dynamic system form a *community structure*. Finding a community structure of a graph is a partitioning problem of a graph, which we will describe in Section 3. We will show how this problem is related to the partitioning problem of the stochastic dynamic systems. Other approaches used for a similar purpose of detecting the community structure in a graph are also applicable to the partitioning problem of the dynamic systems, and they also approximate almost invariant sets. These are reviewed and compared in [11,12].

Applications of modelling and better understanding of systems with basin hopping due to stochastic mechanism are numerous. In particular, there has been some effort to understand the nature of the basin hopping through analysis of the Frobenius–Perron operator, or other descriptions of a discretized model of transfer operators, including in stochastic systems. Some of these researchers have studied in the literature, we find applications to celestial mechanics, [13], to molecular dynamics [14] amongst many others for deterministic analysis to biological applications including mathematical epidemiology for stochastic applications [1,2]. In this paper, we will benchmark our methods with the Duffing oscillator where transport is particularly simple to understand a priori, and then we will also analyze the stochastic dynamics of a gearbox model [5,6].

In this paper we will demonstrate an application of a new graph partition method for detecting community structure, called the modularity method [11,15,16], to a dynamic system for discovering basin boundaries of (especially stochastic) multi-stable systems known only through observed data from an orbit. For completeness we will first review the concept of the deterministic and the stochastic Frobenius–Perron operators and finite rank approximation by the Ulam–Galerkin method. Then we will describe the idea of the graph partition method based on the modularity measure of a network. In the last section we demonstrate the application of this graph partition to the Duffing oscillator model and the gearbox model [5,6].

2. Frobenius–Perron operators and approximation

Let F be a (nonsingular) discrete time dynamic system acting on initial conditions $z \in M$ (say, $M \subset \Re^n$). That is,

$$F : M \to M, \qquad x \mapsto F(x). \tag{1}$$

The Frobenius–Perron operator $P_F : L^1(M) \to L^1(M)$ is defined by [17],

$$P_F[\rho(x)] = \int_M \delta(x - F(y))\rho(y)\mathrm{d}y, \tag{2}$$

where $\rho(x)$ is a probability density function (PDF) defined in $L^1(M)$. Thus $P_F[\rho(x)]$ give us a new probability density function, which is unique a.e. and depend on F and $\rho(x)$. Note that for all measurable sets $A \subset M$, we have a continuity equation [17]

$$\int_{F^{-1}(A)} \rho(x)\mathrm{d}x = \int_A P_F(\rho(x))\mathrm{d}x. \tag{3}$$

Next, we consider a stochastic process defined by

$$x_{n+1} = \nu_n S(x_n) + T(x_n) \tag{4}$$

where ν_n are independent random vectors each having the same density g and a relatively small magnitude compared to the deterministic part S and T. Note that if we set $T \equiv 0$, we would have a process with a multiplicative perturbation, whereas when $S \equiv 1$, we have a process with an additive stochastic perturbation. Suppose the density of x_n given by ρ_n. We desire to show a relation of ρ_n and ρ_{n+1} of the above stochastic process analogous to Eq. (2) using a similar approach employed in [17]. We assume that $S(x_n), T(x_n)$, and ν_n are independent. We let $h : M \to M$ be an arbitrary, bounded, measurable function, and recall that the expectation of $h(x_{n+1})$ is given by

$$E[h(x_{n+1})] = \int_M h(x)\rho_{n+1}(x)\mathrm{d}x. \tag{5}$$

Then, using Eq. (4) we also obtain

$$E[h(x_{n+1})] = \int_M \int_M h(zS(y) + T(y))\rho_n(y)g(z)\mathrm{d}y\mathrm{d}z. \tag{6}$$

By a change of variable, it follows that

$$E[h(x_{n+1})] = \int_M \int_M h(x)\rho_n(y)g((x - T(y))S^{-1}(y)) \times |J|\mathrm{d}x\mathrm{d}y, \tag{7}$$

where $|J|$ is the absolute value of the Jacobian of the transformation $x = zS(y) + T(y)$. Since h was an arbitrary, bounded, measurable function, we can equate Eqs. (5) and (7) to conclude that

$$\rho_{n+1}(x) = \int_M \rho_n(y)g((x - T(y))S^{-1}(y))|J|\mathrm{d}y. \tag{8}$$

The Frobenius–Perron operator for this general form of a stochastic system with both parametric and additive terms can then be expressed similarly to the deterministic case as

$$P_{F_\nu}(\rho(x)) = \int_M \rho(y)g((x - T(y))S^{-1}(y))|J|\mathrm{d}y. \tag{9}$$

Notice that the original deterministic kernel δ now becomes a stochastic kernel $g|J|$. Specifically, we have for the case of the multiplicative perturbation, where $T(x) \equiv 0$, that

$$P_{F_\nu}(\rho(x)) = \int_M \rho(y)g(xS^{-1}(y))S^{-1}(y)\mathrm{d}y. \tag{10}$$

Similarly, the stochastic Frobenius–Perron operator for the additive perturbation,, where $S(x) \equiv 1$, is

$$P_{F_\nu}(\rho(x)) = \int_M \rho(y)g\left(x - T(y)\right)\mathrm{d}y. \tag{11}$$

We use the Ulam–Galerkin method, which is a particular case of Galerkin method [18], to approximate the Frobenius–Perron operator. We use the projection of the infinite dimensional linear space $L^1(M)$ with basis functions $\{\phi_i(x)_{i=1}^{\infty}\} \subset L^1(M)$ on to a finite dimensional linear subspace with a subset of the basis functions,

$$\triangle_N = \text{span}\{\phi_i(x)\}_{i=1}^{N}. \tag{12}$$

For the Galerkin method this projection,

$$\Pi : L^1(M) \to \triangle_N, \tag{13}$$

maps an operator from the infinite-dimensional space to an operator of finite rank $N \times N$ matrix by using the inner product

$$A_{i,j} = \langle P_{F_\nu}[\phi_i], \phi_j \rangle = \int_M P_{F_\nu}[\phi_i(x)]\phi_j(x)\mathrm{d}x. \tag{14}$$

The quality of this approximation is discussed in Bollt [1] and similar references can also be found in [19–24]. For the Ulam method the basis functions are a family of characteristic functions

$$\phi_i(x) = \chi_{B_i}(x) = 1 \quad \text{for } x \in B_i \text{ and zero otherwise.} \tag{15}$$

In the deterministic case, using the inner product equation (14) the matrix approximation of the Frobenius–Perron operator has the form of

$$A_{i,j} = \frac{m(B_i \cap F^{-1}(B_j))}{m(B_i)}. \tag{16}$$

where m denotes the Lebesgue measure on M and $\{B_i\}_{i=1}^{N}$ is a family of boxes or triangles of the partition that covers M and indexed in terms of nested refinements [25]. This $A_{i,j}$ can be interpreted as the ratio of the fraction of the box B_i that will be mapped inside the box B_j after an application of a map to the measure of B_i. A key observation is that the kernel form of the operator in Eq. (9) allows us to generally approximate the action of the operator with test orbits as follows. If we only have a test orbit $\{x_j\}_{j=1}^{N}$, which is actually the main interest of this paper, the Lebesgue measure can be approximated by a counting measure λ and the matrix approximation of the Frobenius–Perron operator becomes

$$A_{i,j} = \frac{\lambda(\{x_k \mid x_k \in B_i \text{ and } F(x_k) \in B_j\})}{\lambda(\{x_k \in B_i\})}. \tag{17}$$

3. Graph theory and reducible matrices

In this section we review some basic concepts of reducible matrices and corresponding relationships to the graph induced by the matrices due to discrete approximations of a Frobenius–Perron operator. In particular, we introduce a question of how to discover a permutation of the Ulam–Galerkin matrix that reveals the basin structure of a dynamic system given to us in the form of test orbit $\{x_j\}_{j=1}^{N}$.

A graph G_A associated to a matrix A consists of a set of vertices V and a set of edges $E \subset V^2$. If a disjoint collection $\{S_i\}_{i=1}^{k}$ of subsets $S_i \subset V$ consists of vertices such that there is a higher density of edges within each S_i than between them, then we say roughly that $\{S_i\}_{i=1}^{k}$ forms a *community structure* for G_A [11,15].

Let $I = \{1 \ldots N\}$ be an index set. For our specific application to the transition matrix A generated by the Ulam–Galerkin method, we define the set of vertices, $V = \{\upsilon_i\}_{i \in I}$ to label the original boxes $\{B_i\}_{i \in I}$ used to generate the matrix A, and define the edges to be the set of ordered pairs of integers $E = \{(i, j) : i, j \in I\}$ which label the vertices as their starting and ending points.

A graph is said to be *reducible* if there exists a subset $I_o \subset I$ such that there are no edges (i, j) for $i \in I_o$ and $j \in I \setminus I_o$, otherwise it is said to be *irreducible* [26,27]. This condition implies that the graph is irreducible if and only if there exists only one connected component of a graph, which is G_A itself. In term of the transition matrix A, G_A is irreducible if and only if there exist a subset $I_o \subset I$ such that $a_{ij} = 0$ whenever $i \in I_o$ and $j \in I \setminus I_o$.

Furthermore, A is said to be a reducible matrix if and only if there exists some permutation matrix P such that the result of the similarity transformation,

$$R = P^{-1}AP \tag{18}$$

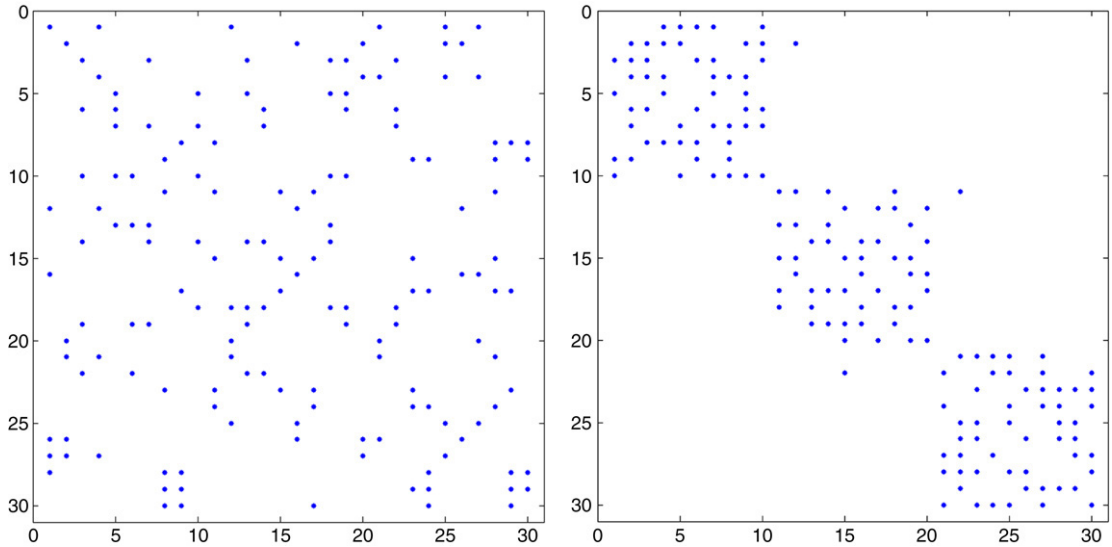
is block upper triangular

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} \\ 0 & R_{2,2} \end{bmatrix}. \tag{19}$$

This means that G_A has a decomposition $V = V_1 \bigcup V_2$, such that V_1 connects with V_1 and V_2, but V_2 connects only with itself. When $R_{1,2} = 0$, A is said to be "completely reducible" [27]. A key observation is that in the case that G_A is generated from a bistable dynamical system the transition matrix A is completely reducible, where $R_{1,1}$ and $R_{1,2}$ corresponds to the two basins of attractions of the system. Also, in a general multistable dynamic systems the transition matrix A has a similarity transformation into a block (upper) diagonal form.

A key point to this work is that most (randomly or arbitrarily realized) indexing of the configuration makes it difficult to observe even simple structure like community structure or reducibility of the graph and corresponding transition matrix. In particular, a very reasonable partitioning of the phase space, and a reasonable indexing of the boxes or triangles is not likely to reveal dynamic organization-like reducibility or a community structure. In Fig. 1(a) and (b) we show an example of a transition matrix for a 30-vertex random community-structured graph with three communities of which members are placed haphazardly, but after a proper permutation this matrix is transformed into an "almost" block diagonal form. Fig. 2(a) and (b) illustrate the associated graph of the matrix in the above example. Before sorting the vertices into three separate communities the graph looks like a random graph that has no community structure. However, after sorting we can see a community structure of the graph.

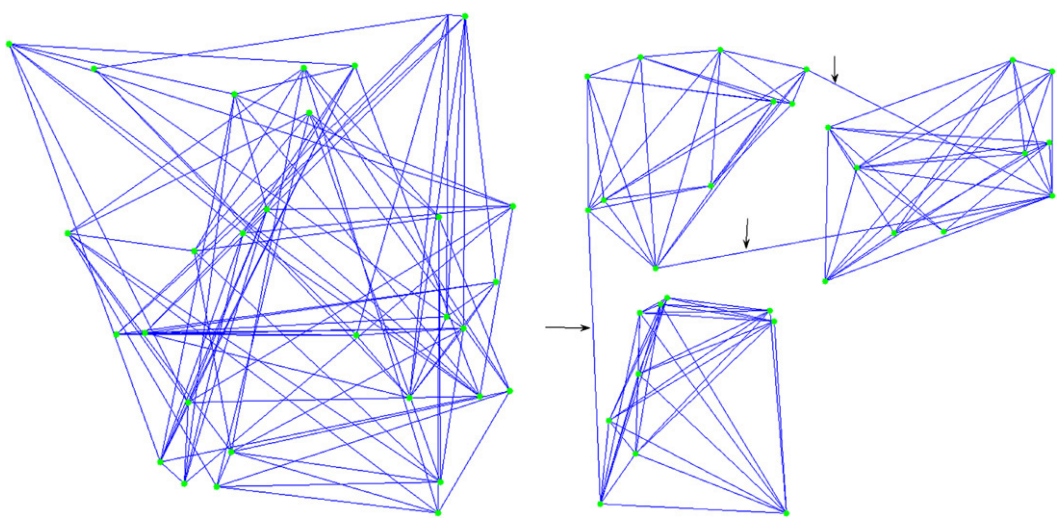
There are several techniques to find an appropriate permutation if the matrix is reducible. In the language of graph



(a) A transition matrix before sorting.

(b) The same transition matrix after sorting.

Fig. 1. The matrix in this figure is irreducible. Members of the original transition matrix before sorting are placed haphazardly and the structure is not obvious. After using the modularity method discussed in Section 21 the "almost" block diagonal form is revealed. In this case, the sorted matrix indicates the existence of three communities and the few off-block diagonal elements correspond to intrabasin diffusion.



(a) The original graph. Embedding vertices into the plane in a random configuration conceals a simple structure.

(b) Sorted graph. The same graph subject to appropriate permutation reveals "almost" block-diagonal structure community between components.

Fig. 2. An undirected graph before and after sorting. The arrows point at the intercommunity edges.

theory, we would like to discover all connected components of a graph. The betweeness methods in [11] and the local method in [28] are examples of numerous methods as reviewed in [11, 12] that can be successfully used to uncover various kinds of community structures.

In the more general case in which noise is added to a multistable dynamic system, we do not expect a reducible transition matrix because noise can cause basin hopping, which in turn adds the connection (edges) between each original connected components representing each basin in the noiseless system. The larger the magnitude of noise, the larger number of the inter-community edges. Therefore, there will be no similarity transformation of the transition matrix into a block diagonal form for a noisy system. However, G_A still has a community structure if noise is small enough. Note that in a case that a magnitude of noise is large a number of intercommunity edges may become so large that a community structure is destroyed. In other words, one might be able to discover only one community, which represents the entire phase space of the time-series data, or one may obtain a large number of very small communities that have no meaningful community structure. Bollt et al. [1] developed a technique to discover an appropriate permutation based on a convergence property of elements in B_i. However, such a technique could not be applied to the case that we merely have a test orbit $\{x_i\}_{i=1}^N$ of a real world dynamic systems, i.e. the data collected from experiments.

The question of how to discover a proper community structure of a noisy multistable dynamic system without a priori knowledge of a mathematical model of the system is the main interest of this paper. The edge betweeness method [15], based on computing shortest paths, is a good robust candidate algorithm for detecting communities, but its run time is of the order $O(m^2n)$ for a graph with m edges and n vertices, or $O(n^3)$ for a sparse graph($m \sim n$). Such expensive demands on computational resources make it infeasible for a large graph. Furthermore, a d-dimensional dynamic system with an h^d-element box (for box of order length h on a side) or triangle covering of the invariant set makes therefore an $O(h^{3d})$ algorithm, which is quite expensive for a fine grid, $h \ll 1$, in high dimension-d. In the next section we will describe a method based on a modularity measure of a graph partition developed by Newman [16]. This method is far less expensive than the betweeness method and gives useful results in many applications. In the last section of this work we demonstrate examples of how to apply this method to our specific applications in dynamic systems.

4. Modularity

As discussed in the previous section our main purpose is to discover community structure in a network derived from an orbit of a dynamic system through a grid partitioning of the phase space; such structure indicates the dynamic basin structure. Thus, our point of view is to map the problem of phase space partition of the dynamic system (approximately) into a problem of partitioning a digraph. Many methods have been proposed in recent years [11,15,16,28] to appropriately partition a digraph. Loosely, the community structure is a partition of the digraph into subgraphs such that there are relatively more connections within each defined component than between components, a sort of self clustering. In this paper, we employ a new and efficient algorithm proposed by Newman [16] for detecting community structure based on the optimization of the "modularity" measure.

First, we discuss the concept of the modularity as a measure of quality of a proposed community structure found by any graph partition algorithm. The modularity is a cost function associated with a partitioning of a given graph G,

$$Q : \mathcal{P}_G \to \mathfrak{R}, \tag{20}$$

where $\mathcal{P}_G$ is the set of all subpartitions P, of a given graph G. Given a graph G and a (test) partition, $P \in \mathcal{P}_G$, $\mathbf{P} = \cup_k \mathbf{P}_k$, each P is a set of subsets P_k, and P_k is a collection of vertices of G. $\mathbf{P}$ includes all of the vertices of G. We will refer to P_k as "community"-k. The modularity of the partition $\mathbf{P}$ is meant to reflect the quality of the split into self clustered elements P_k A modularity measure is defined by

$$Q(\mathbf{P}) = \sum (e_{ii} - a_i^2), \tag{21}$$

where e_{ij} is the fraction of edges that connect vertices in community i to those in community j and,

$$a_i = \sum e_{ij}, \tag{22}$$

represents the fraction of edges that connects to community i. Thus $Q(\mathbf{P})$ measures the difference between the fraction of the within-community edges and the expectation of the same quantity in the network with same community partition created by randomizing all connections between vertices. Therefore, $Q(\mathbf{P})$ approaches 0 for a randomly connected network and approaches $Q(\mathbf{P}) = 1$, if the network has a strong community structure. We may then optimize Q over all possible partitions to discover the best community structure

$$Q = \max_{\mathbf{P} \in \mathcal{P}_G} Q(\mathbf{P}). \tag{23}$$

The true optimization, however, is very costly to implement in practice for very large networks. Clauset, Newman, and Moore [29] proposed an approximate optimization algorithm based on greedy optimization. Suppose that we have a graph with n vertices and m edges. This algorithm starts with each vertex being the only member of one of n communities. At each step, the change in $Q(\mathbf{P})$ is computed after joining a pair of communities together and then choosing the pair that gives the greatest increase or smallest decrease in $Q(\mathbf{P})$. The algorithm stops after $n - 1$ such joins in which a single community is left. This algorithm therefore runs in time $O((m + n)n)$, or $O(n^2)$ on a sparse graph ($n \sim m$). Note that using more sophisticated data structure as introduced in [29] can reduce a run time to $O(n \log^2 n)$. Furthermore, the algorithm builds a dendrogram which represents nested hierarchy of all possible community partition of the network. Then we select the partition corresponding to local peaks of the objective function $Q(\mathbf{P})$ for our satisfactory community division. Fig. 3 shows an example of the dendrogram of the graph network shown in Fig. 2(a). We can see that the peak in the modularity corresponds to the correct identification of the community structure and the modularity is dropped from the peak when two of the three communities are joined together at the final step.

5. Examples

In this section,for benchmarking purposes, we first demonstrate an application of the modularity method to the double-well Duffing oscillator, where the solution is the expected double wells. Then we apply the method to a gearbox model from mechanical engineering [6,30] and then finally to a many basin system which is known only through data. We will observe changing basin structure with the standard deviation of the noise.

For each model, we simulate a long test orbit. Recall that we are interested in the case that we only have a given orbit, but lack a mathematical model of the map that generates the test orbit. Then as explained above, we develop a digraph transition model of the Frobenius–Perron operator, discretized on a fine triangulation grid of the phase space. Then our goal becomes the determination of an appropriate partition of the phase space using the modularity method explained in the previous section.

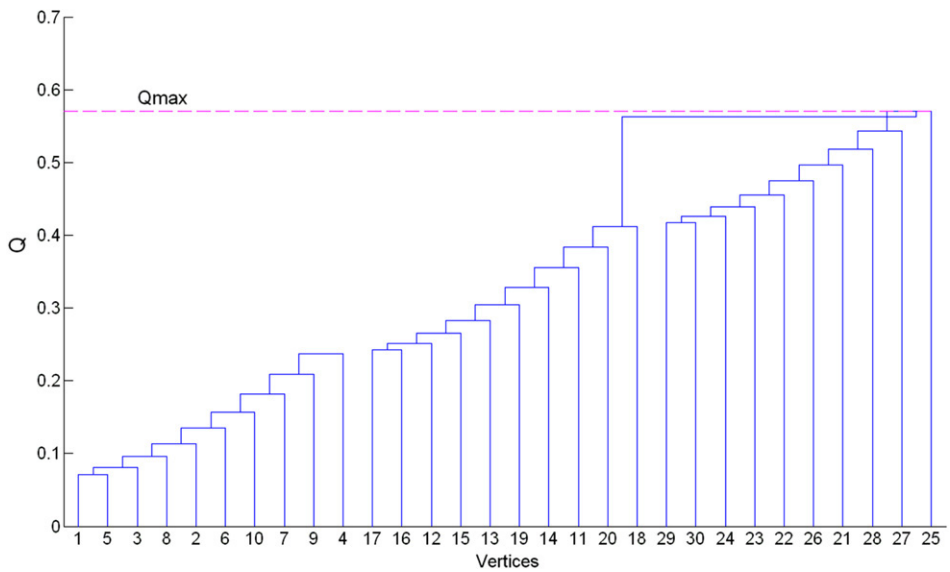


Fig. 3. Plot of dendrogram for a graph network in Fig. 2(a). As we can see the peak in the modularity (dashed line) occurs where the network is divided into three communities.

5.1. Duffing oscillator

Our first example is chosen for sake of simplicity to benchmark and clarify our method. Consider the the double-well Duffing oscillation given by

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= x - x^3 - \delta y + F\cos(2\pi\omega t), \end{aligned} \tag{24}$$

where $\delta = 0.25$, $F = 0.25$, and $\omega = 1$. To obtain a stochastic version of this model noise is added to the above equations periodically at the same phase having mean zero and standard deviation σ. Thus this model may be viewed as a stochastic map

$$F : \mathbb{R}^2 \to \mathbb{R}^2, \qquad (x, y)(t+1) = F[(x, y)(t)] + \nu(t), \tag{25}$$

where $\nu(t)$ is a discrete noise term. This additive noise induces chaos-like oscillations since it causes noise-induced basin hopping between two basins of the noise-free system. In order to study where in the phase space these basin hopping are most likely to occur we should first determine the basin boundaries.

For the particular experiment shown below, we randomly choose an initial condition and iterate it 50,000 times using the time-T map, $T = 1$, of the Duffing model with normal noise of zero mean and standard deviation $\sigma = 0.01$, which is added after each iteration of the map. Note that in the noiseless case, for the particular chosen parameter values of the deterministic system, if the initial point belongs to one of the wells it will converge to the bottom of the well ($x = -1$ or $x = 1$). However, when noise is added, hopping between the two basins may occur sporadically, depending on the nature of the noise. Of course, given just the data of a sampled orbit, it is not immediately obvious where are the basins and where the jumping between the basins occurs. Of course, we only know this information since we have chosen a particularly simple benchmark example. We will now demonstrate that our algorithms defined in the previous sections can automatically discover what we already know for this problem.

With the test orbit in hand we choose a bounded phase space that covers every points in the orbit and divide it into small triangular regions (3200 triangles are used in this example). After (arbitrarily) indexing each triangle we then generate a graph whose edges and vertices correspond to the transition of points in the orbit and the index of these triangles, respectively. Then we can apply the modularity method to find a partition of this graph.

The results of the digraph partitioning are shown in Fig. 4, and the correspondingly indexed triangles are embedded back in their native phase space positions, and coloured according to the partition results. We can see from Fig. 4 that there are two basins coloured by magenta and yellow in accordance with left and right wells, as expected. Furthermore, hopping between the two basins occurs many times. The triangles are automatically coloured according to the partition by the modularity method, which agrees very well with a priori known behaviour described in the above paragraph. We see that the two basins are each automatically coloured as their own connected components in the phase space. Fig. 5(a) and (b) shows the Galerkin matrices before and after the sorting by using the partition due to the modularity method. Of particular interest when inspecting the transitions matrices in their almost canonical reducible form is the elements in the off-diagonal blocks. These off-block diagonal elements indicate the hopping between two the basins, which is approximately in agreement with Eq. (19). Accordingly the corresponding triangles are the more active transport regions in the phase space.

5.2. Gearbox model

Our second example is a model for rattling in a single-stage gearbox systems with a backlash, comprising of two wheels with a sinusoidal driving [6,30]. The relative displacement between the gears due to the backlash is

$$s = \frac{AR_e}{\nu}\sin\omega t - \frac{R}{\nu}\varphi, \tag{26}$$

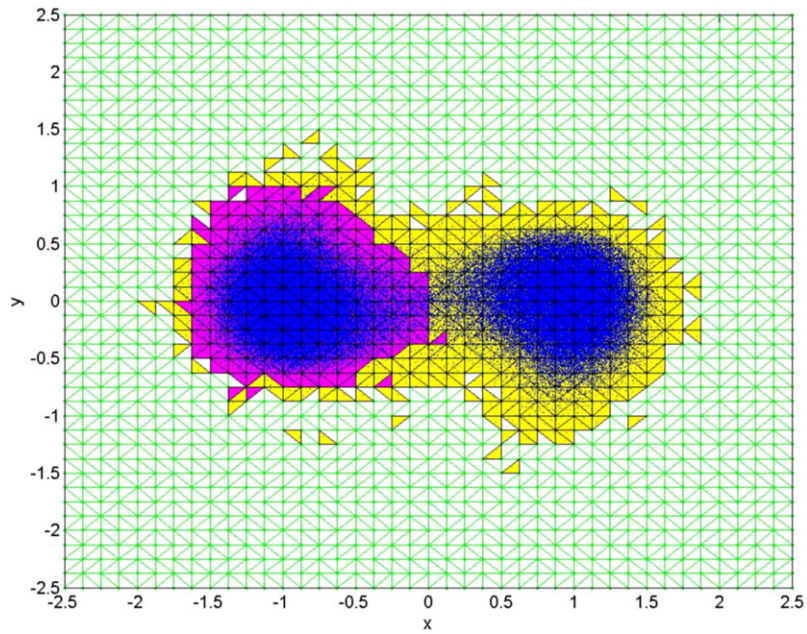
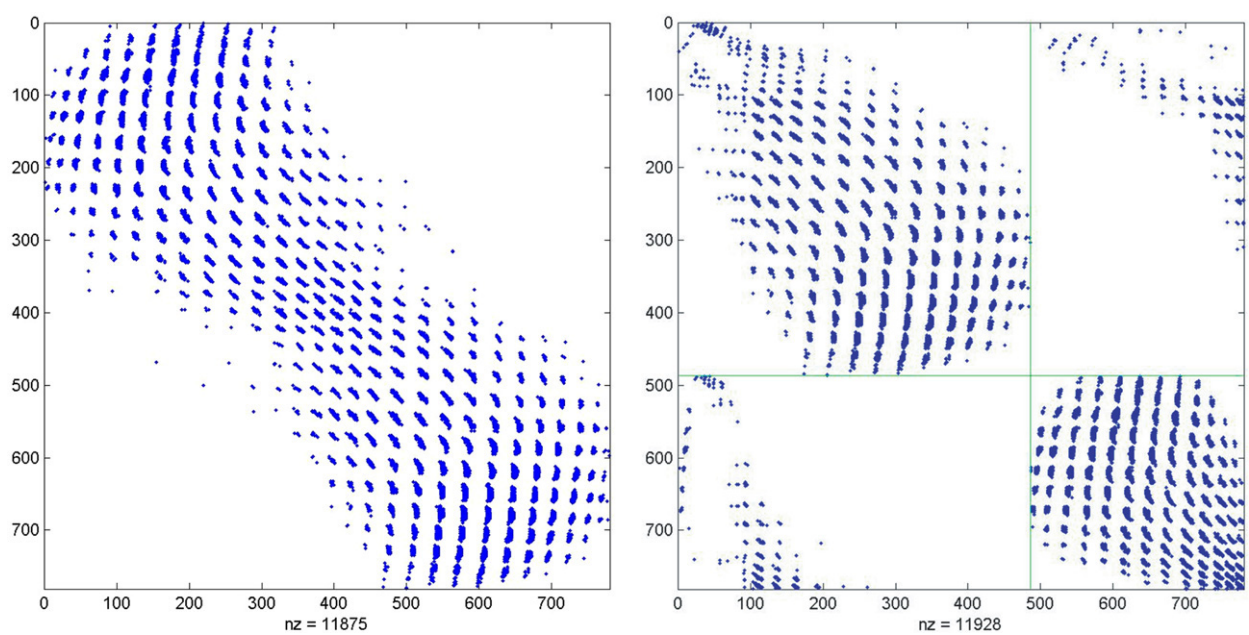


Fig. 4. A partition of an orbit generated from the Duffing model with $\delta = 0.095$ and normal noise, $\sigma = 0.01$. The colors, magenta and yellow, are according to the two communities found by the modularity method applied to the transition matrix, and we see that the colors of these corresponding triangles agree well with the a priori known basins of the deterministic system. The blue dots shown are the test orbit used in the process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) The Ulam–Galerkin matrix from the Duffing oscillator before sorting. No obvious basin structure or transport in this operator.

(b) The Ulam–Galerkin matrix from the Duffing oscillator after sorting into into approximate reducible form. We see the basin structure of off-diagonal elements showing relative transport between basins.

Fig. 5. Galerkin matrices before and after sorting of the Duffing oscillator.

where A and ω are angular amplitude and frequency of the driving gear, respectively, R and R_e are the radii of two gears with a backlash ν, and φ is the angular displacement of the second gear. The s variable varies from -1 to 0 and the impact occurs when $s = 0$ or -1. Let $s_n, \dot{s_n}$, and τ_n represent the displacement, velocity, and time (modulo 2π just before the nth impact. Then the dynamic variable for $(n+1)$th impact can be derived as a function of the nth impact [5];

$$\begin{aligned} s_{n+1} &= s_n + \alpha(\sin\tau_{n+1} - \sin\tau_n) + \frac{\gamma}{\beta}(\tau_{n+1} - \tau_n) \\ &\quad - \frac{1}{\beta}(1 - \exp[-\beta(\tau_{n+1} - \tau_n)]) \\ &\quad \times \left(r\dot{s_n} + \alpha\cos\tau_n + \frac{\gamma}{\beta}\right), \end{aligned} \tag{27}$$

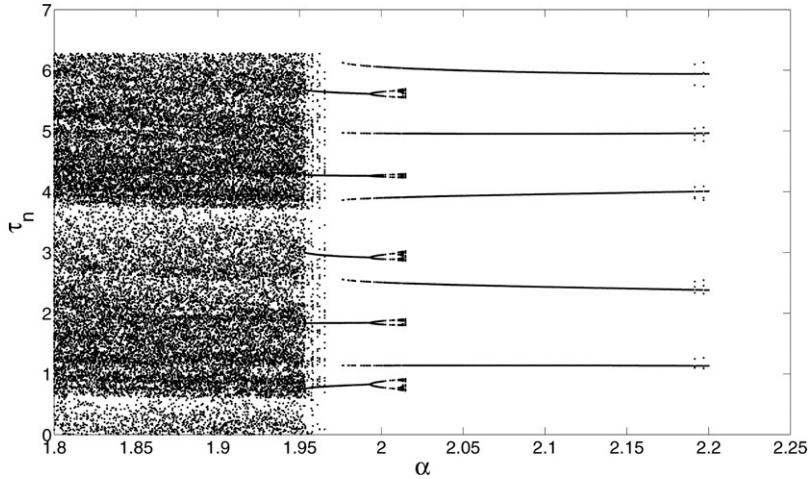


Fig. 6. Bifurcation diagram for the time of impact.

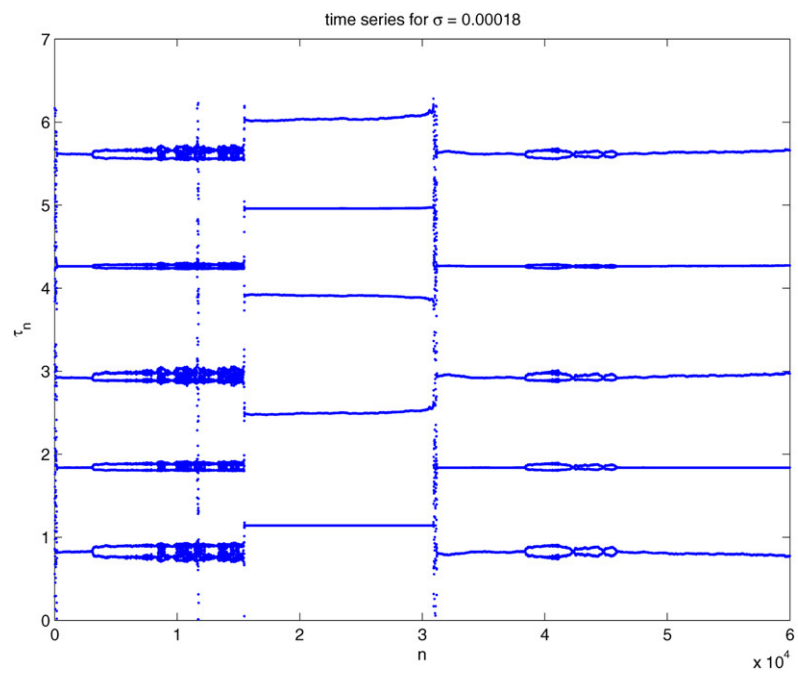


Fig. 7. A time series of the τ_n for $\alpha = 1.985$ at noise level $\sigma = 0.0018$.

$$\dot{s}_{n+1} = \alpha \cos \tau_{n+1} + \left(\frac{\gamma}{\beta} - \exp[-\beta(\tau_{n+1} - \tau_n)]\right) \times \left(r\dot{s}_n + \alpha \cos \tau_n + \frac{\gamma}{\beta}\right). \tag{28}$$

Pictures showing the physical meaning of these variables in more detail can be found in [6,30].

We fix values, $s_0 = -0.5$, $\beta = 0.1$, $\gamma = 0.1$, and $r = 0.95$. The bifurcation diagram shown in Fig. 6 shows that two attractors coexist near the $\alpha = 2$ and these attractors appear as the successive time of five impacts. Following Souza et al. [6], we add noise to the control parameter α. That is

$$\alpha_n = \alpha + \eta(n), \tag{29}$$

where $\eta(n)$ is a discrete noise term, with a normal distribution, with zero mean, and with standard deviation σ. Note that in this example we have the case of the stochastic Frobenius–Perron operator Eq. (9), in which the perturbation is neither additive nor multiplicative but a combination of both. We simulate a time series using 60,000 iterations for $\alpha = 1.985$ and set the noise standard deviation at $\sigma = 0.0018$. A time series for this case is shown in Fig. 7, where the basin hopping can be noticed.

Following the same procedures as described in the preceding example we can generate a graph associated to the orbit of the gearbox model. The result from the modularity method for this case is shown in Fig. 8, where the large squares and dots indicate each attractor for the noiseless case. The transition matrices before and after sorting according to the partition from the modularity method are shown in Fig. 9(a) and (b).

5.3. A stochastic model with many basins

Let's consider the following Hamiltonian system with a stochastic white noise:

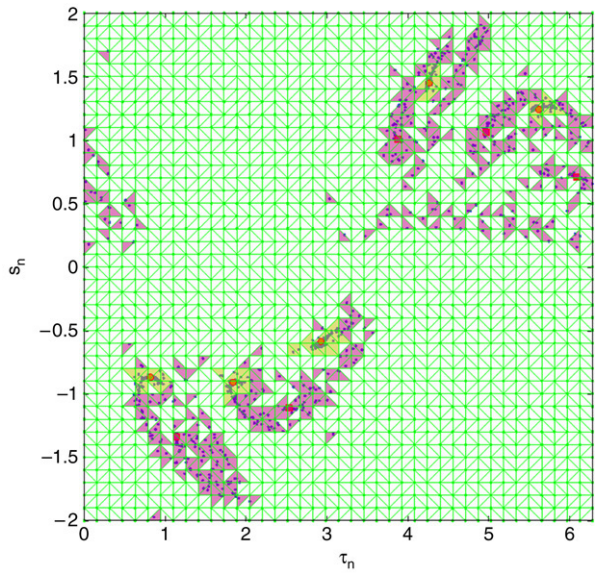


Fig. 8. A partition of an orbit generated from the gearbox model with normal noise, $\sigma = 0.0018$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

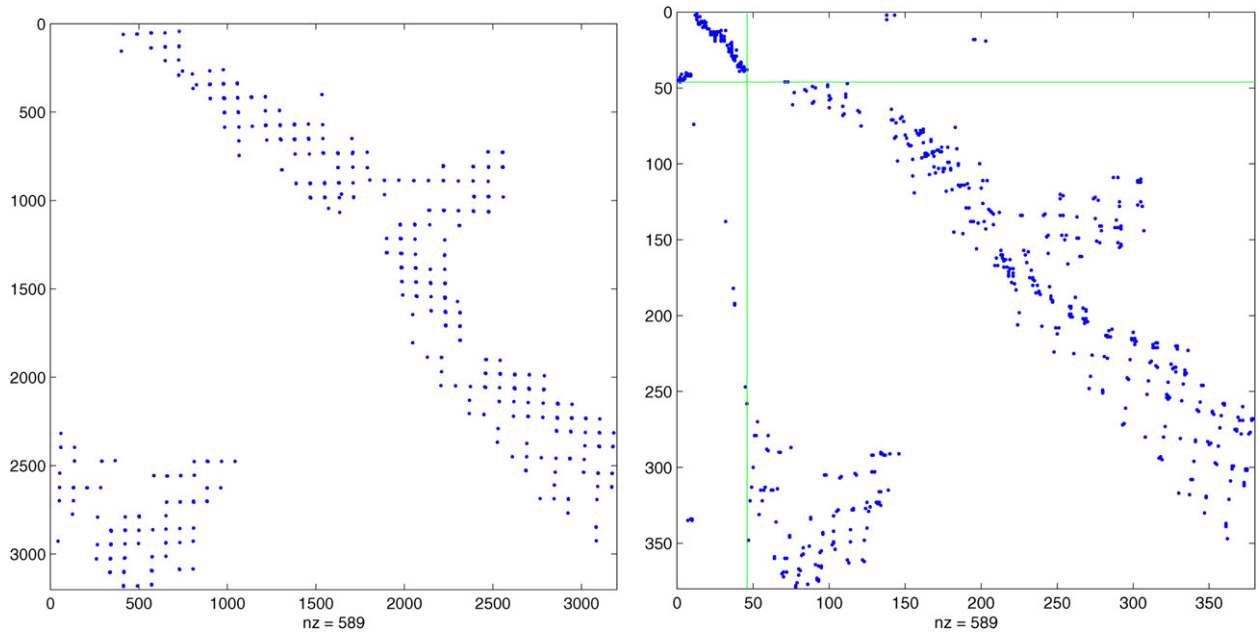
$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial \psi}{\partial y} + \eta_x(t) \\ \frac{dx}{dt} &= -\frac{\partial \psi}{\partial x} + \eta_y(t) \end{aligned} \tag{30}$$

where the streamfunction ψ is given numerically as shown in Fig. 10 and η_x and η_y are the normal noises with zero mean. We then generate a time-series data based on the time-T map of Eq. (30) for a normal noise with zero mean and standard deviation $\sigma = 0.001$ as shown in Fig. 11. One may notice bursting in the time series, which suggests transitions between the almost invariant sets.

To investigate how the basin structure changes as the standard variation of noise is varied, we compare the result obtained from the modularity method for four cases of $\sigma = 0.001, 0.005, 0.01$, and 0.05 as shown in Figs. 12 and 13 In all four cases we start with an initial point $(x, y) = (-0.2, 0.6)$. Note that the regions with no colour do not contain any point in a test orbit and so they are not used in a process of generating a graph. One may observe from Fig. 12 that the bandwidth of the Ulam–Galerkin matrix before sorting becomes larger as σ increases. Moreover, the modularity measure as defined in Eq. (23) becomes lower when noise becomes greater ($Q = 0.7267, 0.7183, 0.6968$, and 0.6361 for $\sigma = 0.001, 0.005, 0.01$ and 0.05, respectively). Recall that the higher the value of the modularity measure, the stronger the community structure (the lower value of the ratio of the intercommunity edges and the within-community edges).

6. On computational complexity

In this section, we briefly discuss the computational complexity of the methods described in this paper. Although increasing the number of boxes or triangles coverings, and hence the number of vertices in the graph, can make a graph partition that reveal more refined structure of basin, there exists a necessary number of boxes or triangles coverings to discover a basin structure. This minimum requirement can be related to the Lyapunov exponent of a given dynamical system from which the data comes from. This means that refining a covering of a phase space beyond the necessary requirement will not reveal more structure of basins, but any over-refinement still



(a) Before. (b) After.

Fig. 9. Transition matrix before and after sorting of the gearbox model. Notice in the sorted matrix that two distinct basins become clear as those triangles indexed to associate to the diagonal blocks, and the regions causing hopping are clear as those few off diagonal elements. Each of the two basins, the smaller and larger blocks, are the pink and yellow coloured regions in Fig. 8.

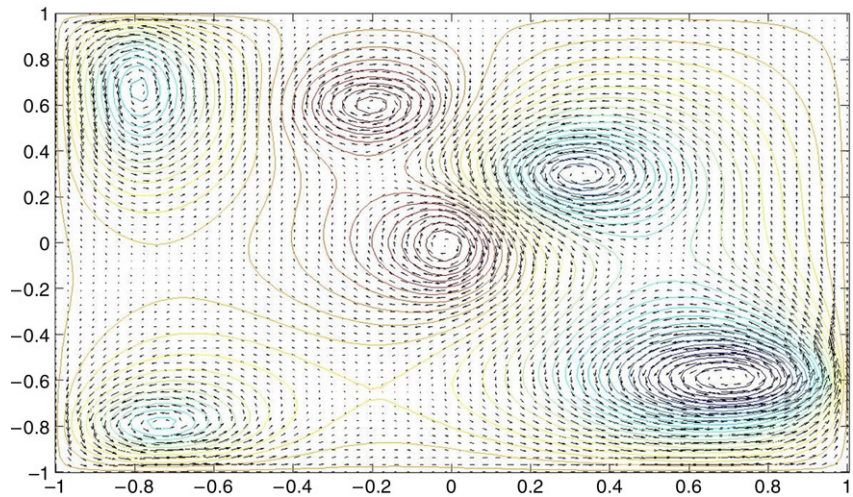


Fig. 10. The velocity filed for the noiseless case of Eq. (30) is computed from the above streamfunction by the relation $(\mathrm{d}x/\mathrm{d}t, \mathrm{d}y/\mathrm{d}t) = \nabla \times \psi$.

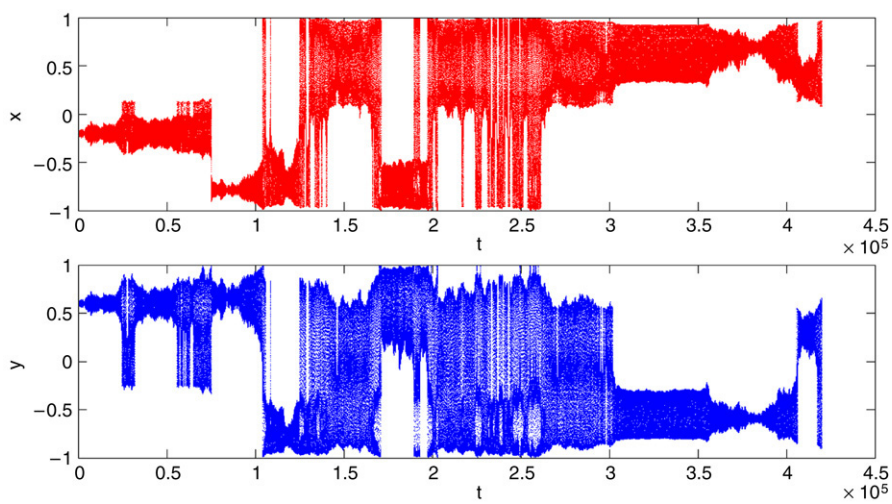


Fig. 11. A test orbit of the time-T map of Eq. (30) for a normal noise with standard deviation $\sigma = 0.001$.

requires more computer storage and more computer work. The following analysis assumes the zero noise, and so this would seem to imply it may be relevant at best for only small noise profiles. We remind however that there does exist a concept of stretching and Lypanov exponent for a stochastic system [31], and so we offer the following as a heuristic model which is suggestive of the necessary refinement of the grid to gain a given knowledge of the system's evolution. We find that these considerations give a useful starting point of the noise profile is not overwhelming.

We use the Baker's map as an example to describe the trade-off between the number of box coverings and the Lyapunov exponent. Recall that a Baker's map on the unit square $[0, 1] \times [0, 1]$ maybe defined by [32],

$$\begin{aligned} x_{n+1} &= \begin{cases} \lambda_1 x_n & \text{if } y_n < \alpha, \\ (1-\lambda_2) + \lambda_2 x_n & \text{if } y_n > \alpha, \end{cases} \\ y_{n+1} &= \begin{cases} y_n/\alpha & \text{if } y_n < \alpha, \\ (y_n - \alpha)/(1-\alpha) & \text{if } y_n > \alpha, \end{cases} \end{aligned} \tag{31}$$

where $\lambda_1 + \lambda_2 < 1$ and $0 < \alpha < 1$. The forward-time invariant set of this map is a Cantor set of parallel vertical lines. Consider learning these structures with box covering methods, and what can be discovered by such finite computations. Suppose that we want to cover all parallel stripes left in the unit box *after n iterations* by square boxes of size ϵ so that two stripes are not contained in the same box, see Fig. 14. The size of the square box required to do this is

$$\epsilon(n) = \min\{\lambda_1^n, \lambda_2^n\}. \tag{32}$$

Therefore, the required number of boxes to reveal the fractal structure of these parallel stripes is

$$N(\epsilon) \geq \epsilon(n)^{-2}, \tag{33}$$

and $d = 2$ is the dimension of the embedding phase space. Using more boxes than the minimum requirement (33) would only increase required storage and work but does not give us a better knowledge of the stripes left in the unit box.

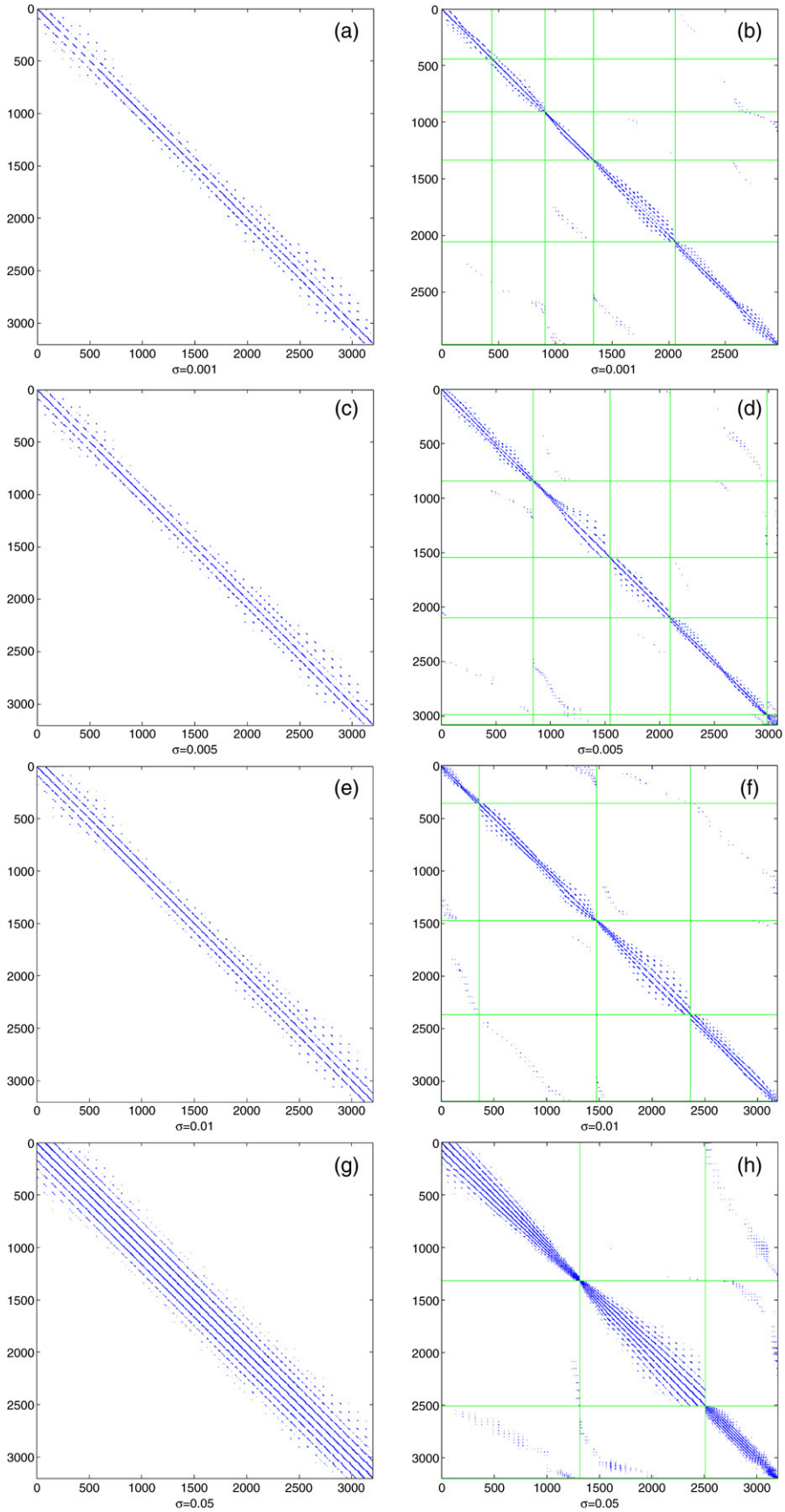


Fig. 12. Transition matrix before and after sorting of the model Eq. (30). Notice decreasing the number of basins as the noise increases.

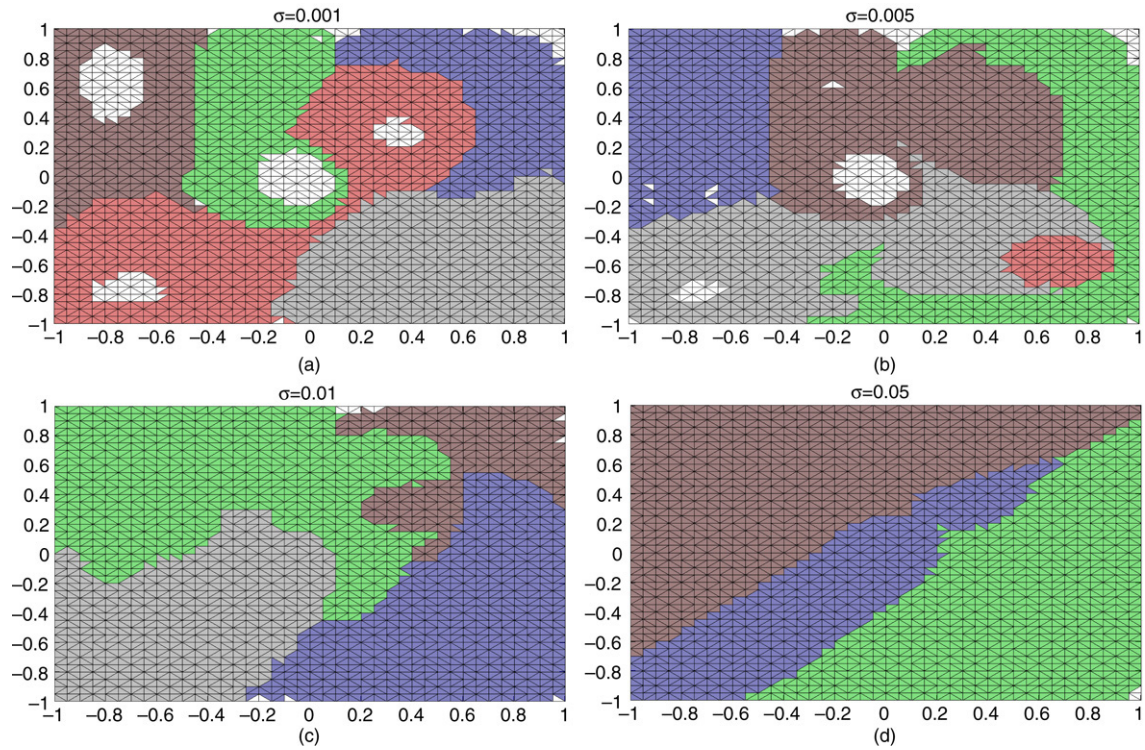
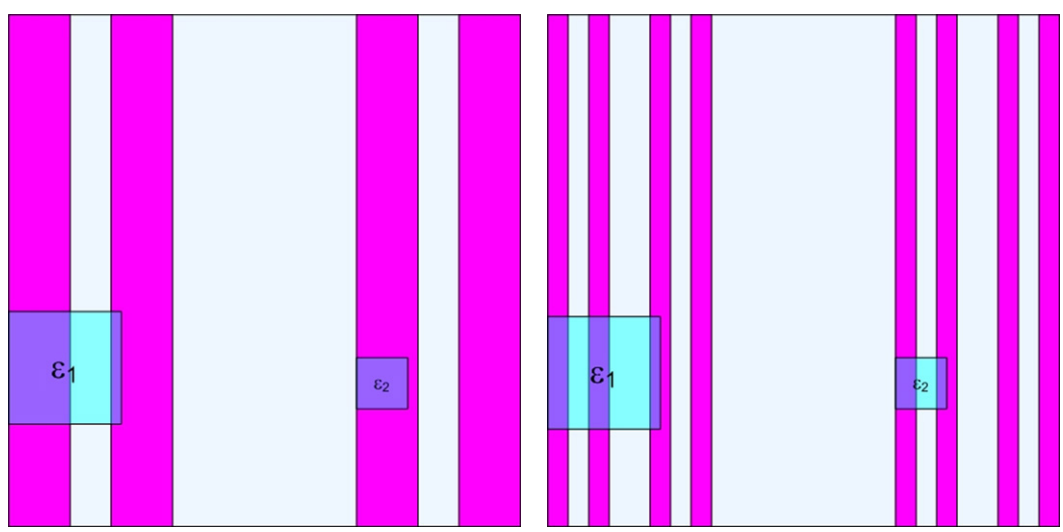


Fig. 13. The phase space partition of the duffing-like model with various values of σ using the graph modularity method. The values of the modularity measure as defined in Eq. (23) in each case are $Q = 0.7267, 0.7183, 0.6968$, and 0.631 for $\sigma = 0.001, 0.005, 0.01$ and 0.05, respectively. Similarly, the basin structure is changing with σ.



(a) All stripes after $n = 2$. A square box of size ϵ_1 cannot resolve a structure finer than ϵ_1. But we see that a smaller square box of size ϵ_2 is able to resolve a stripe of width up to ϵ_2.

(b) Here, however, the box of size ϵ_2 cannot resolve the stripe left in the unit box after $n = 3$.

Fig. 14. Two iterations of the Baker map with $\lambda_1 = \lambda_2$.

Thus we see that there is a trade-off between Lyapunov exponents, λ_i, time, n, box size, ϵ, the size of the memory requirements and size of the resulting digraph, $N(\epsilon)$ and the dimension of the underlying space, d.

7. Conclusion

In this paper, we have described how a concept of community structure in a graph network relates to a

stochastically perturbed multistable system through the (almost) reducibility of the matrix representing the finite-rank approximation of the Frobenius–Perron operator by the Ulam–Galerkin method. A new graph algorithm based on the optimization of the modularity measure for extracting a community structure in a network can be applied to detect pseudo-barriers and transport mechanisms in phase space from a time series data that comes from a stochastic multistable dynamic systems. Optimizing the modularity of a large network, which varies between 0 and 1, for all possible network partitions is computationally infeasible. Instead, an approximate optimization based on greedy algorithm [29] is used, which runs in time $O(n \log^2 n)$ on a sparse graph with n vertices. We demonstrate the utility of this method in three examples; the Duffing oscillator, the gearbox model with normal noise and a stochastic system with many basins. For our benchmark models, we found that the resulting community structures correspond closely to the priori knowledge of the basin structures in both cases.

Although the examples we demonstrated are two-dimensional, the extension of the methods to extract a basin structure from a time series data from a stochastic multistable system of higher dimensions proceeds in a straightforward manner. The box dimension will be the moderating factor for the computational complexity of our methods.The network derived from such cases can be very large, but using the algorithm that has $O(n \log^2 n)$ running time will make community structure analysis of a very large network possible in practice.

Acknowledgements

We thank Jim Bagrow for sharing his expertise in random graph theory and network partition methods. We would also like to thank Aaron Clauset and Mark Newman for sharing their codes used in [29] for modularity method computations. We are grateful to the National Science Foundation of the USA for their support of both of the authors of this project through DMS-0404778.

References

[1] E. Bollt, L. Billings, I. Schwartz, A manifold independent approach to understanding transport in stochastic dynamical systems, Physica D 173 (2002) 153–177.
[2] L. Billings, E. Bollt, I.B. Schwartz, Phase-space transport of stochastic chaos in population dynamics of virus spread, Phys. Rev. Lett. 88 (2002) 234101.
[3] S. Kraut, U. Feudel, Multistability, noise and attractor-hopping: The crucial role of chaotic saddles, Physica D 181 (2003) 222–234.
[4] S. Kraut, U. Feudel, Enhancement of noise-induced escape through the existence of a chaotic saddle, Phys. Rev. E 67 (2003) 1–4.
[5] S.L.T. de Souza, I.L. Caldas, Basin of attraction and transient chaos in a gear-rattling model, J. Vib. Control. 7 (2004) 849–862.
[6] Silivio L.T. de Souza, Ibereĕ L. Caldas, Ricardo L. Viana, Antŏnio M. Batista, Tomasz Kapitaniak, Noise-induced basin hopping in a gearbox model, Chaos Solitons Fractals 26 (2005) 1523–1531.
[7] M. Dellnitz, O. Junge, On the approximation of complicated dynamical behavior, SIAM J. Numer. Anal. 36 (2) (1999) 491–515.
[8] R. Preis, M. Dellnitz, Congestion and almost invariant sets in dynamical systems, in: Proceedings of Symbolic and Numerical Scientific Computation, SNSC'01, Springer, 2003.
[9] R. Preis, M. Dellnitz, M. Hessel, Ch. Schtte, E. Meerbach, Dominant paths between almost invariant sets of dynamical systems. Preprint 154 of the DFG Schwerpunktprogramm 1095, 2004.
[10] K. Padberg, R. Preis, M. Dellnitz, Integrating multilevel graph partitioning with hierarchical set oriented methods for the analysis of dynamical systems. Preprint 152 of the DFG Schwerpunktprogramm 1095, 2004.
[11] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.
[12] L. Dannon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification. Preprint cond-mat/0505245, 2005.
[13] M. Dellnitz, O. Junge, W.S. Koon, F. Lekien, M.W. Lo, J.E. Marsden, K. Padberg, R. Preis, S.D S.D. Ross, B. Thiere, Transport in dynamical astronomy and multibody problems, Internat. J. Bifur. Chaos Appl. Sci. Engrg. 15 (2005) 699–727.
[14] R. Preis, M. Dellnitz, M. Hessel, Ch. Schtte, E. Meerbach, Dominant paths between almost invariant sets of dynamical systems. Preprint cond-mat/0505245, 2004.
[15] M.E.J. Newman, M. Girvan, Statistical Mechanics of Complex Networks, Springer, Berlin, 2004.
[16] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (2004) 066133.
[17] A. Lasota, M.C. Mackey, Chaos, fractals, and noise, Stochastic Aspects of Dynamics, 2nd edition, Springer, New York, 1994.
[18] T.Y. Li, Finite approximation for the Frobenius–Perron operator. a solution to Ulam's conjecture, J. Approx. Theory 17 (2) (1976) 177–186.
[19] G. Froyland, Estimating physical invariant measures and space averages of dynamical systems indicators. Ph.D. Thesis, University of Western Australia, 1996.
[20] G. Froyland, Finite approximation of Sinai–Bowen–Ruelle measure for Anosov systems in two dimension, Random Comput. Dynam. 3 (4) (1995) 251–263.
[21] C. Chui, Q. Du, T. Li, Error estimates of the Markov finite approximation of the Frobenius Perron operator, Nonlinear Anal. 19 (4) (1992) 291–308.
[22] J. Ding, A. Zhou, The projection method for computing multidimensional absolutely continuous invariant measures, J. Stat. Phys. 77 (3–4) (1994) 899–908.
[23] F. Hunt, Approximating the invariant measures of randomly perturbed dissipative maps, J. Math. Anal. Appl. 198 (2) (1996) 534–551.
[24] A. Boyarsky, Y.S. Lou, Approximating measure invariant under higher-dimensional chaotic transformation, J. Approx. Theory 65 (2) (1991) 231–244.
[25] S.M. Ulam, Problems in Modern Mathematics, Science editions, Wiley, New York, 1970.
[26] Béla Bollobás, Modern Graph Theory, Springer, NewYork, 1998.
[27] N.G. Van Kampen, Stochastic Processes in Physics and Chemistry, 4th edition, North-Holland, 2003.
[28] J.P. Bagrow, E.M. Bollt, A local method for detecting communities, Phys. Rev. E 72 (2005) 046108.
[29] A. Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (2004) 066111.
[30] F. Pfeiffer, A. Kunert, Rattling models from deterministic to stochastic precesses, Nonlinear Dynam. 1 (1991) 63–74.
[31] L. Arnold, Random Dynamical Systems, 2nd edition, Springer, New York, 2003.
[32] E. Ott, Chaos in Dynamical Systems, 1st edition, Cambridge University Press, 1993.