Coarse-grained dynamics of an activity bump in a neural field model

C R Laing[1]**, T A Frewen**[2] **and I G Kevrekidis**[2]

[1] Institute of Information and Mathematical Sciences, Massey University, Private Bag 102-904 NSMC, Auckland, New Zealand
[2] Department of Chemical Engineering, Princeton University, NJ 08544-5263, USA

E-mail: c.r.laing@massey.ac.nz

Abstract
We study a stochastic nonlocal partial differential equation, arising in the context of modelling spatially distributed neural activity, which is capable of sustaining stationary and moving spatially localized 'activity bumps'. This system is known to undergo a pitchfork bifurcation in bump speed as a parameter (the strength of adaptation) is changed; yet increasing the noise intensity effectively slowed the motion of the bump. Here we study the system from the point of view of describing the high-dimensional stochastic dynamics in terms of the effective dynamics of a single scalar 'coarse' variable, i.e. reducing the dimensionality of the system. We show that such a reduced description in the form of an effective Langevin equation characterized by a double-well potential is quantitatively successful. The effective potential can be extracted using short, appropriately initialized bursts of direct simulation, and the effects of changing parameters on this potential can easily be studied. We demonstrate this approach in terms of (a) an experience-based 'intelligent' choice of the coarse variable and (b) a variable obtained through data-mining direct simulation results, using a diffusion map approach.

Mathematics Subject Classification: 60H15,65P30,37G35,45K05,82C31

1. Introduction

Pattern formation at large spatial scales in the cortex is a field of much recent interest [5, 9, 11, 15, 35]. Neural dynamics are intrinsically noisy, so it is natural to study the effects of noise on models for such pattern formation. In this paper we revisit a noisy model for pattern formation in a neural system [31] but this time concentrating on describing its dynamics in a
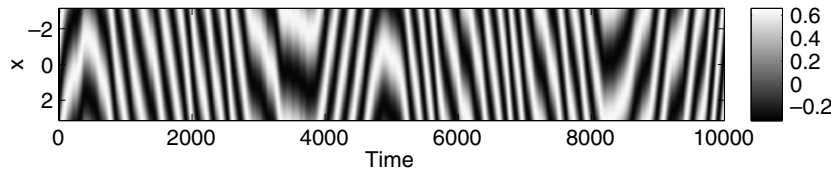
Figure 1. A typical simulation of system (3) and (4). The profile of u is shown as a function of space and time (values indicated by shadebar). Parameters are $M = 100$, $\eta = 10^{-4}$, $A = 0.17$, $\tau = 5$.

low-dimensional way. If an accurate low-dimensional description can be found, it will have obvious advantages over the original description of the system, in terms of simulation, analysis and understanding.

We consider the partial integrodifferential system

$$\frac{\partial u(x,t)}{\partial t} = -u(x,t) + \int_{-\pi}^{\pi} J(x-y) f[I + u(y,t) - a(y,t)] \, \mathrm{d}y, \tag{1}$$

$$\tau \frac{\partial a(x,t)}{\partial t} = Au(x,t) - a(x,t) \tag{2}$$

on the domain $-\pi \leqslant x \leqslant \pi$ with periodic boundary conditions, originally analysed by Laing and Longtin [31], as a simple model for pattern formation in a neural field. The variable $u(x,t)$ describes the neural activity at position x and time t. The function J describes the spatial coupling within the network, with $J(x-y)$ being the strength of coupling between neurons at position x and neurons at position y. The function $f(I)$ describes the firing rate of a single neuron with input I. The quantities A, I and τ are constant. The value of A determines the strength of adaptation, I is a background current applied to all neurons and τ is the adaptation time constant. Equation (2) represents the effects of spike frequency adaptation. Without (2) systems of this form have been used to model orientation tuning in the visual system, working memory and the head direction system [5, 15]. Variants including (2) have recently been explored [9–11, 23, 35].

In our numerical implementation we spatially discretize the domain with M equally spaced points, resulting in the $2M$ ordinary differential equations (ODEs)

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -u_i + \frac{2\pi}{M} \sum_{j=1}^{M} J_{ij} f(I + u_j - a_j), \tag{3}$$

$$\tau \frac{\mathrm{d}a_i}{\mathrm{d}t} = Au_i - a_i, \tag{4}$$

for $i = 1, \ldots, M$, where $J_{ij} = J(2\pi|i-j|/M)$. As in Laing and Longtin [31] we set $I = -0.1$, $\tau = 5$ and use $f(x) = (1 + \tanh(10x))/2$ and $J(x) = 0.05 + 0.24 \cos x$, with $M = 100$. We add noise to the system by adding a Gaussian white noise term $\xi_i(t)$ to each of (3), with $\langle \xi_i(t) \rangle = 0$ and $\langle \xi_i(t)\xi_j(s) \rangle = 2\eta v_{ij}\delta(t-s)$, where $v_{ij} = 0$ if $i \neq j$ and 1 if $i = j$. Initially we set the parameters $A = 0.17$ and $\eta = 10^{-4}$; we will study the effect of varying them below.

At the deterministic limit $\eta = 0$, for these parameter values the system dynamics ultimately exhibit a single stable activity bump that moves around the (periodic) domain in one direction (determined by the initial condition), at a constant velocity. If $\eta \neq 0$, the bump motion becomes stochastic, exhibiting occasional switches in direction. This effect, studied by Laing and Longtin [31], is illustrated in figure 1; they concentrated on the net distance travelled during a finite time interval. For small η the bump moved in only one direction, so that the net

distance travelled was quite far (effectively proportional to the time); for η greater than a certain value, switches in direction lead to a much smaller net distance travelled. This phenomenon was referred to as *noise-induced stabilization*.

Laing and Longtin [31] showed that for a coupling function of the form used here, system (1) and (2) underwent a pitchfork bifurcation in speed as A was increased. They then added Gaussian white noise to the normal form of a pitchfork bifurcation, and qualitatively reproduced the noise-induced stabilization. However, the actual derivation of a 'noisy normal form' from the original system with noise was not attempted. (Note that Curtu and Ermentrout [11] have derived normal forms for the deterministic bifurcation from stationary to moving bumps for a model similar to (1) and (2).)

In this paper we will demonstrate numerically that the noisy discretized system (3) and (4) undergoes what we will characterize as an *effective pitchfork bifurcation* in speed as A is increased, and investigate the effects of varying noise intensity. This computer-assisted bifurcation analysis will be performed assuming that the dynamics of the high-dimensional stochastic system (3) and (4) can be effectively described by the dynamics of a single, scalar coarse-grained *observable* (variable). The evolution of this variable can be determined by running many short, appropriately initialized simulations of (3) and (4), in the spirit of the 'equation-free' framework [27, 30].

The value of our scalar variable V (defined below) can be interpreted as the position of a 'particle' moving in a potential, subject to noise. Beyond the effective pitchfork bifurcation the potential is double-welled; before the bifurcation, it is single-welled. Each of the two wells correspond to persistent motion in one direction (left or right); in the case of a single well we have a stationary (on average) activity bump. Since the system is isotropic, we expect the double well to be symmetric. Although we cannot analytically derive the effective potential and noise in our hypothesized Langevin description for V, we will show how they can be estimated using appropriately initialized short bursts of simulation of (3) and (4). Once the effective potential is approximated, we can locate its maxima and minima (which correspond to unstable and stable fixed points for the particle, respectively) and estimate average transition times between minima (corresponding to bump direction switches). By determining how the potential changes as A and η are varied we can also numerically obtain a bifurcation diagram for its extrema, which takes the place of the traditional steady-state bifurcation diagram for the deterministic problem. This technique allows us to identify transition states (potential maxima, corresponding to unstable steady states), which would be extremely difficult if not impossible to do using long simulations alone. An interesting point is that we can quantify the relationship between the noise intensity in the original system, η, and the effective noise intensity, $D(V)$, to which the particle in the potential is subject.

2. Results

2.1. Reconstructing the potential

We choose as our coarse scalar variable, $V(t)$, the instantaneous difference in position between the peak of $u(x, t)$ and the peak of $a(x, t)$. This choice comes from observations of the system dynamics: practically all system profiles during a long simulation are characterized by a single peak in each of the variable fields. Furthermore, the difference in the positions of these peaks is clearly related to the instantaneous speed of the bump: the peak of a normally lags the peak of u relative to the direction of bump motion—see the bottom panel of figure 2. One can clearly see in the top two panels of figure 2 that periods of bump motion to the left (right) are characterized by effectively constant negative (positive) $V(t)$. Factoring out the small
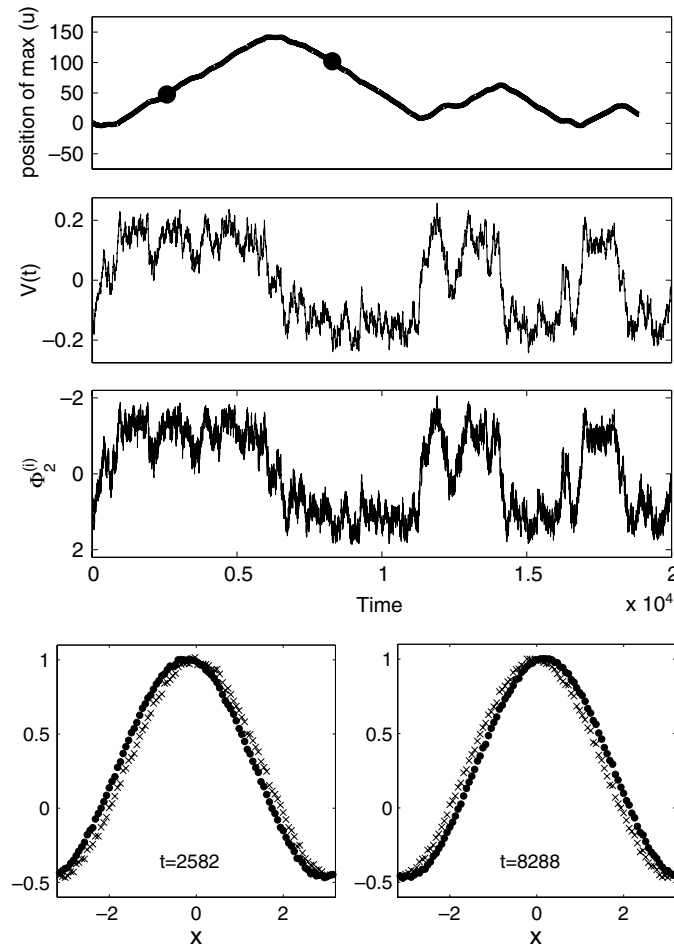
Figure 2. Simulation results for system (3) and (4). Top row: position of the peak in the u profile as a function of time; second and third rows: $V(t)$ and $\Phi_2^{(i)}$ (defined in section 3), respectively, for the same time interval; bottom row: spatial profiles of u (cross) and a (filled circles) at times marked in top panel. Parameters are the same as in figure 1.

amplitude noisy oscillations riding on the bump waveform, we identify the position of the peak of $u(x, t)$ as the location c_u that satisfies

$$\sum_{i=1}^{M} \sin(x_i - c_u)u_i(t) = 0, \tag{5}$$

where $x_i = 2\pi i/M$. (Effectively, we are finding the amount by which a simple sine wave must be shifted so that its inner product with $u(x, t)$ is zero.) Similarly, c_a satisfies

$$\sum_{i=1}^{M} \sin(x_i - c_a)a_i(t) = 0 \tag{6}$$

so that $V \equiv c_u - c_a$.

To establish the connection between V and the effective potential $\Pi(V)$ we assume that $V(t)$ satisfies an (unknown) Langevin equation. Then, the probability density $P(V, t)$ satisfies

a Fokker–Planck equation of the form

$$\frac{\partial P(V,t)}{\partial t} = \left[-\frac{\partial}{\partial V}\mu(V) + \frac{\partial^2}{\partial V^2}D(V) \right] P(V,t). \tag{7}$$

The (unknown in closed form) quantities $\mu(V)$ and $D(V)$ are estimated from ensembles of brief simulations initialized at V:

$$\mu(V) = \lim_{\Delta t \to 0} \langle \Delta V \rangle / \Delta t \qquad 2D(V) = \lim_{\Delta t \to 0} \langle [\Delta V]^2 \rangle / \Delta t, \tag{8}$$

where $\Delta V = V(t+\Delta t) - V(t)$ and the average is taken over the realization ensemble. More accurate, maximum likelihood based estimation techniques [1] can also be used. At stationarity we have $P(V) \propto \exp[-\beta\Pi(V)]$ where $\Pi(V)$ is the effective potential, related to $\mu(V)$ and $D(V)$ by [13, 39]

$$\beta\Pi(V) = \text{const} - \int_0^V \frac{\mu(s)}{D(s)} \mathrm{d}s + \log D(V). \tag{9}$$

Data for the estimation of $\mu(V_0)$ and $D(V_0)$ for a given value of V_0 can be obtained in two ways. Firstly, we find occurrences of this value in a long simulation; we then track $V(t)$ over a subsequent time interval (we use 14 time units) for each of these occurrences, and then average the appropriate quantities. This approach has been used previously to estimate drift and diffusion terms in Langevin equations [21]. Typical results of estimating $\mu(V)$ and $D(V)$ using this approach are shown in figure 3 (top and middle). Performing this for a grid of V_0 values provides enough data for the numerical estimation of the integral in (9); the results are shown in figure 3 (bottom).

   In the absence of a database from a long enough (equilibrium) simulation, a better—more economical—way to find $\mu(V)$ and $D(V)$, and thus $\Pi(V)$, is to deliberately initialize the system with a given value of V_0 and then track $V(t)$ over a subsequent time interval (we again use 14 time units), repeating and averaging as above. The advantage of this approach, as compared with the previous one, is that it need only be performed for as many V_0 values as necessary for a given accuracy in the integral evaluation. Furthermore, for each such V_0 value we can reinitialize as many independent runs as necessary for accurate estimation at will, without requiring a simulation so long that rare values of V_0 are revisited enough times. Results of the latter type of calculation are shown in figure 4. The only significant difference in this case is that initializing the system with a particular value of V seems to result in a lower estimate of both $D(V)$ and $\Pi(V)$ when $|V|$ is large. Recall, however, that these regions are rarely visited in a simulation, and thus the discrepancy may be attributed to an insufficiently long database for the first approach. As a comparison, for the amount of averaging performed here, the total amount of simulation time needed to produce figure 3 is the same as is required to produce figure 4.

   In summary, we have chosen a variable V, whose value describes the state of system (3) and (4), assumed that it obeyed a Langevin equation, and numerically obtained the effective potential for the corresponding Fokker–Planck equation.

2.2. Bifurcations

A traditional bifurcation diagram for the deterministic problem with respect to a parameter such as A would involve tracing branches of steady states and constant shape travelling waves (which can also be reduced to solutions of a fixed point problem) using pseudo-arclength and branch switching techniques (e.g. [2]). In the coarse-grained stochastic case, it is natural to trace instead the zeros of the drift $\mu(V)$ as the same parameter A is varied, using the same standard bifurcation codes. It is worth noting that for the case of state-dependent noise, the
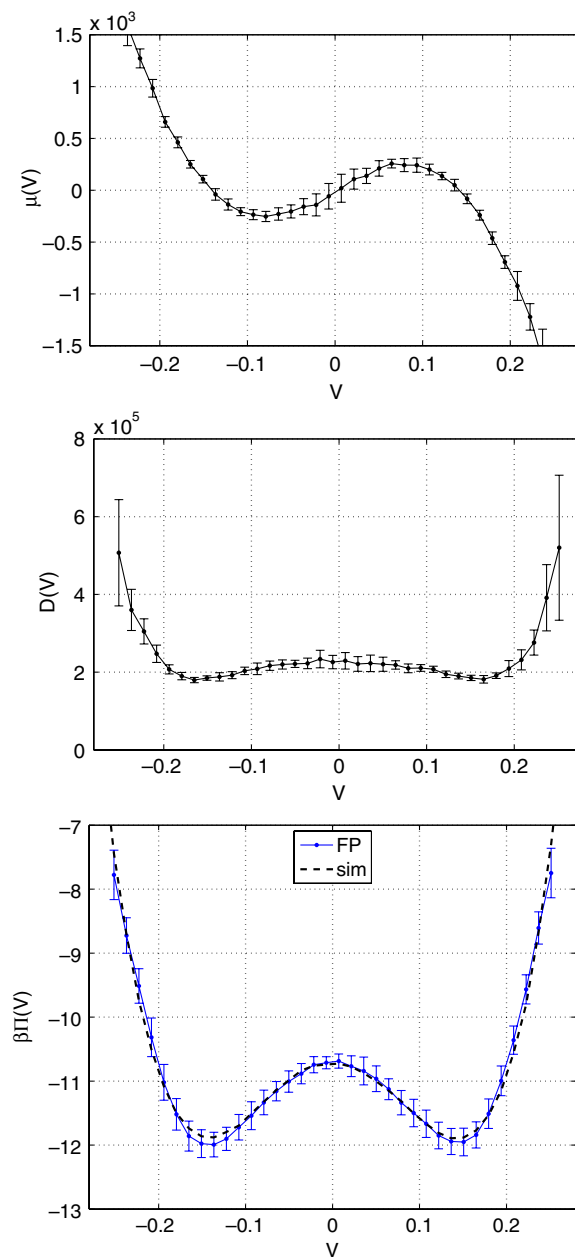
Figure 3. Estimation of $\mu(V)$, $D(V)$ and $\beta\Pi(V)$ using the occurrences of V in a long simulation. Top: estimate of $\mu(V)$. Middle: estimate of $D(V)$. Bottom: dashed line from assuming that $P(V) \propto \exp[-\beta\Pi(V)]$; points from assuming a Fokker–Planck equation and evaluating the integral (9), using the $\mu(V)$ and $D(V)$ shown in the upper panels. Parameters are $A = 0.17$, $\eta = 10^{-4}$.

local maxima of the probability density do not exactly correspond to zeros of $\mu(V)$ and one needs instead to find fixed points of the effective potential given in (9) (zeros of the right-hand side of (9) differentiated with respect to V). For the scalar case, one can implement secant-type iterative methods to converge to the zeros of the appropriate function using function estimates
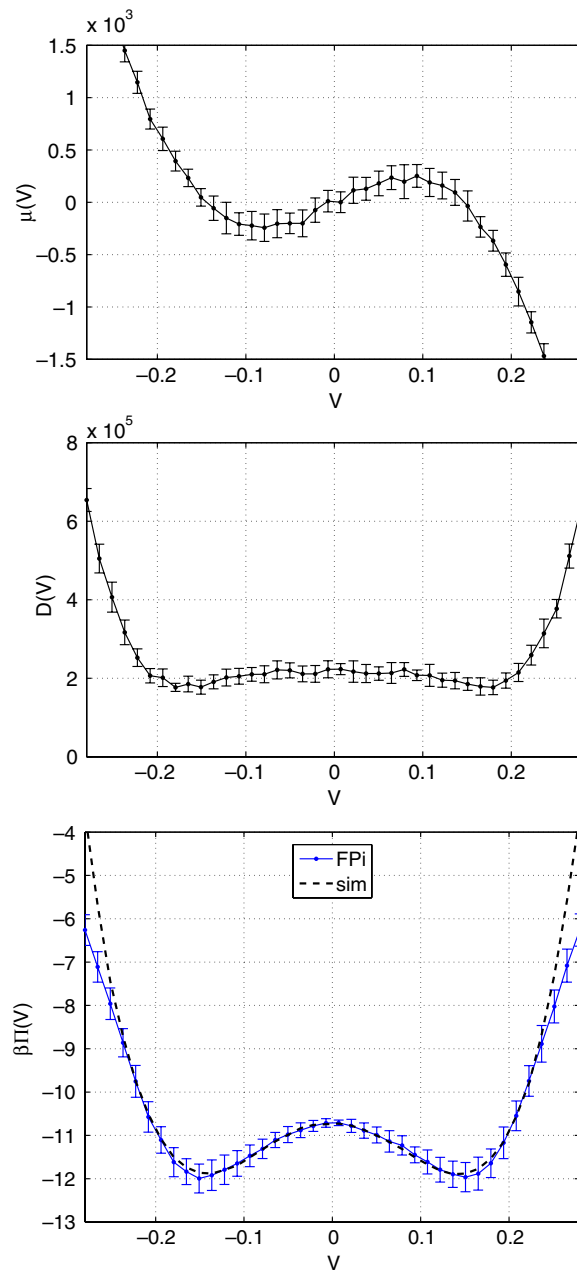
Figure 4. Estimation of $\mu(V)$, $D(V)$ and $\beta\Pi(V)$ by initializing the system at a particular value of V and then running for a short time. Top: estimate of $\mu(V)$. Middle: estimate of $D(V)$. Bottom: dashed line from assuming that $P(V) \propto \exp[-\beta\Pi(V)]$; points from assuming a Fokker–Planck equation and evaluating the integral (9), using the $\mu(V)$ and $D(V)$ shown in the upper panels. For each of 36 values of V, 4000 short bursts (duration 14 time units) were run. Parameters are $A = 0.17$, $\eta = 10^{-4}$. Note the more uniform errorbar sizes compared with those in figure 3.
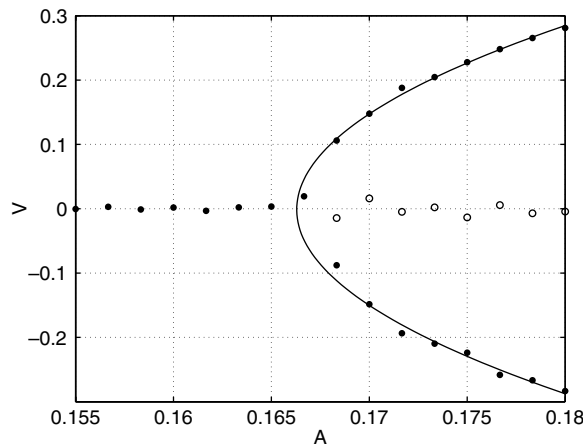
Figure 5. Zeros of $\mu(V)$ as a function of A, for $\eta = 10^{-4}$. Filled circles indicate stable solutions; empty circles, unstable. Also shown is a quadratic curve (in V) that best fits the outer branches. For each estimation of $\mu(V)$, 10 000 short bursts were run.

only; for the multivariable case matrix-free iterative techniques like Newton–Krylov GMRES can be used (see [26, 37]).

In the cases we study $\mu(V)$ is well approximated by a cubic function; we take advantage of this simplification by estimating $\mu(V)$ for just four values of V, uniquely defining this cubic whose zeros we can then easily find. Such a 'noisy' bifurcation diagram is shown in figure 5, where we vary A. Stability can be determined from the local slope, $\mu'(V)$, at the fixed point, and is indicated in the figure. Parameters of a different nature, such as the noise intensity or colour, can also be varied in this context. Figure 6 shows the 'effective pitchfork' bifurcation that occurs as η is varied. This can be thought of as either a 'noise-induced' bifurcation [17] or the delaying of the underlying pitchfork bifurcation by noise [19, 22]. The location of these effective bifurcations was verified by running long simulations in their vicinity (results not shown). Note that if we assume in advance that the potential $\Pi(V)$ is symmetric (as the underlying system is), the cubic function representing $\mu(V)$ would have no quadratic or constant terms, so that only two evaluations of $\mu(V)$ would be needed to approximate the required cubic.

2.3. Switching times

One of the most important statistics of our problem is the average time between changes of direction, or, more generally, the statistics of these switching times for the bump. Based on our effective Fokker–Planck model and Kramers' theory, once we have $\beta \Pi(V)$ we can also estimate the average time between switches in direction [13], as

$$\tau \approx \frac{2\pi \exp[\beta \Delta \Pi]}{\beta \overline{D} \sqrt{-\Pi''(V_{\min})\Pi''(0)}}, \tag{10}$$

where $V_{\min}$ is the value of V at which Π has a minimum, $\overline{D} = [D(0) + D(V_{\min})]/2$ and $\Delta \Pi = \Pi(0) - \Pi(V_{\min})$. Measurements of $\overline{D}$ and Π for $\eta = 10^{-4}$ and $A = 0.17$ give $\tau \approx 3 \times 10^3$. A simulation run at these parameter values had mean waiting time 3.5×10^3, in excellent agreement with our Kramers' approximation.

Along the same lines, we can quantify how the average switching time, τ, depends on parameters. Typical results are shown in figure 7. Here, the effective potential is not estimated
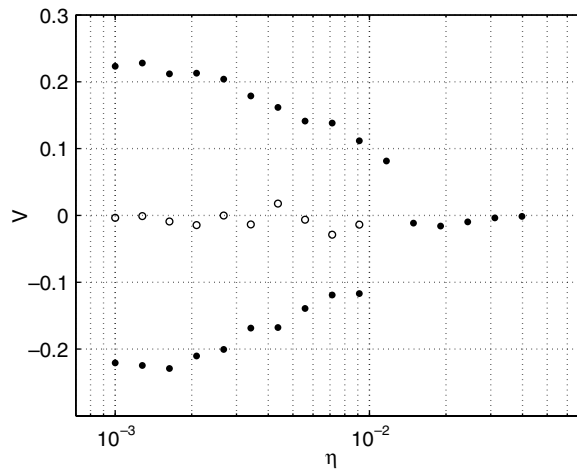
Figure 6. A noise-induced bifurcation. Zeros of $\mu(V)$ as a function of η, for $A = 0.175$. Filled circles indicate stable solutions; empty, unstable. For each estimation of $\mu(V)$, 10 000 short bursts were run.
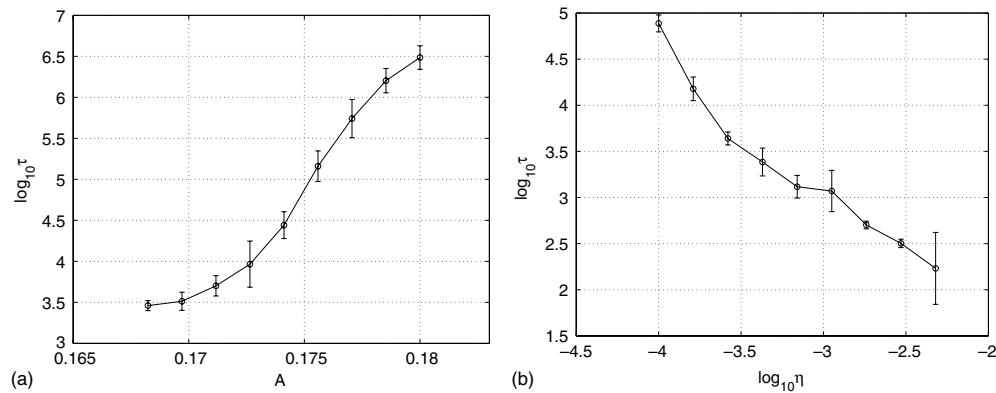

Figure 7. (a) Switching time, τ, plotted on a log scale, as a function of A. $\eta = 10^{-4}$. (b) Switching time, τ, plotted on a log scale, as a function of η (also on a log scale). $A = 0.175$. For both panels, τ was estimated from a calculation of $\beta\Pi$, which itself was computed by initializing V at appropriate values and then running to observe its short-term evolution.

from a long simulation database, but rather 'on demand' by initializing the system at particular values of V, performing short time simulation bursts, and processing their results as before. A clear advantage of this approach, as opposed to running the system for long enough to measure sufficient occurrences of switches, is that very large values of τ can be inferred from a reasonable number of short simulations. See figure 7, for example, where we can 'measure' switching times greater than 10^6.

For a simulation of a given, fixed duration T, these curves can be used to infer the parameter values for which we expect to see direction changes. If $\tau > T$, we will typically see the bump move in one direction only, whereas if $\tau < T$ we will see switching, leading to a drop in the absolute value of the distance travelled during the time T. For example, from figure 7(b) we see that for simulations of length $T = 10^3$, η must be greater than approximately 10^{-3} to observe any switching of direction.
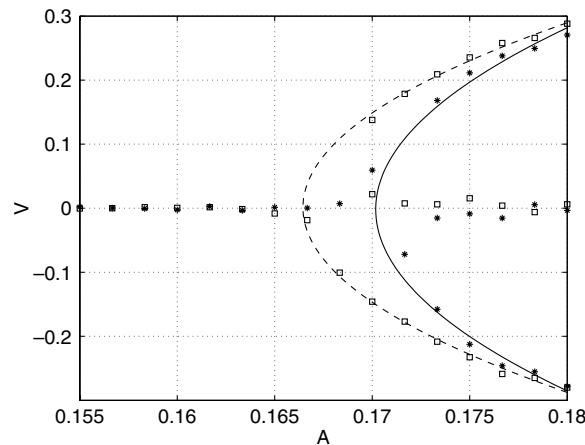
Figure 8. Zeros of $\mu(V)$ as a function of A for coloured noise with correlation time $\lambda = 100$ (solid line and asterisks) and $\lambda = 1$ (dashed line and squares). The curves are quadratic functions in V fitted to the data points.

2.4. White versus coloured noise

In [31] the authors investigated the effects of adding coloured noise to (3)-(4). They found that increasing the correlation time of the coloured noise made the bump less likely to move. This was rationalized in terms of 'frozen' noise (i.e. spatial inhomogeneity) which was likely to 'pin' the bump, preventing it from moving. Increasing the correlation time of the coloured noise was thought of as interpolating between Gaussian white noise (with delta function autocorrelation in time) to frozen, spatially structured, noise. Here we investigate the effect again, quantifying the influence of noise colour on the location of the underlying bifurcation.

As in [31] we add noise to the system by adding a term $\eta_i(t)$ to each of (3), with $\langle \eta_i(t) \rangle = 0$ and $\{\langle \eta_i(t)\eta_j(s) \rangle\} = 2\epsilon \nu_{ij} e^{-|t-s|/\lambda}$, where $\nu_{ij} = 0$ if $i \neq j$ and 1 if $i = j$. The notation $\{\cdots\}$ indicates averaging over the initial distribution of $\eta(0)$ values, taken from a Gaussian with zero mean and variance 2ϵ. We keep $\epsilon = 10^{-4}$ and vary the correlation time λ. The results are shown in figure 8 where we have plotted zeros of $\mu(V)$ as a function of A for $\lambda = 1$ and $\lambda = 100$. The figure was computed using the short runs initialized at prescribed V values. The results are consistent with those found in [31], where it was seen that coloured noise was more effective at slowing the bump than Gaussian white noise of the same power, and that the effect was stronger for longer correlation times. The results presented here show that this effect can be rationalized in terms of shifting a bifurcation point. Results for several other values of λ indicate that as λ is increased, the bifurcation moves to higher values of A (results not shown). Note that we have assumed that the dynamics of V are still governed by a Langevin equation, even though there are now temporal correlations in the noise terms. The results shown in figure 8 were verified by direct simulation.

We have thus performed effective bifurcation analysis of (3) and (4), using the dynamics of V and its associated potential $\beta \Pi(V)$.

3. Diffusion maps and the data-based detection of coarse observables

The previous results relied on our ability to choose a scalar variable, V, whose value correlates with the state of the high-dimensional system (after brief initial transients have equilibrated).
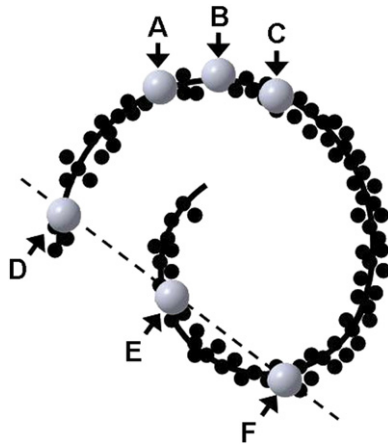
Figure 9. Schematic of datapoints (filled circles) in $\mathbb{R}^2$ that lie along a curve (solid line). Euclidean distance is a good measure of the separation between points A, B and C along the curve but is a poor measure of the separation of points D, E and F along the curve.

However, we are often faced with dynamical systems for which choosing such a variable is far from obvious. We then need a systematic procedure for determining, from the results of a simulation, a good low-dimensional representation of the system state. Such data-mining, so-called 'manifold learning' techniques have been a focus of intense research in recent years [4, 7, 8, 12, 14, 38, 40]; we will use here the recently developed diffusion map approach [32, 33]. The general idea is to 'mine' the large collection of high-dimensional data points resulting from a long simulation of our original system (3) and (4), so as to obtain a low-dimensional description of the data. We now briefly describe the general technique and then discuss the application to our particular system. More details can be found in the appendix.

3.1. General technique

A large data ensemble from direct simulations of a system can be represented as a cloud of points in a high-dimensional space; see the schematic in figure 9. When two such points are very close to one another (i.e. when the Euclidean distance between them is small, as in the case of points A, B and C in the schematic), we can consider this distance as representative of the 'intrinsic similarity' of the two configurations—in some sense, of how easy it is for the system dynamics to cause a transition from one configuration to the other. When, however, this Euclidean distance is larger than some threshold (as in the case of points D, E and F in the schematic) the Euclidean distance stops being a good measure of the 'intrinsic similarity' between configurations—the 'effort to transition' from E to F, measured by the arclength between them, is clearly less than the effort to transition from E to D, even though the Euclidean distances DE and EF are similar. The main idea underpinning diffusion maps is to perform a random walk on a graph in which data points are vertices, and connection strengths between data points are given by a Gaussian kernel of the form $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2/\sigma^2)$.

When the points are farther away than a cutoff (controlled by the parameter σ) the vertices are effectively disconnected; when they are very close, the strength of the connection is controlled by their (small) Euclidean distance. A random walk on such a graph gives rise to a *diffusion distance*—starting from the same source point and diffusing on the graph for some time, we then look for equal density contours; points on such a contour are equally easy
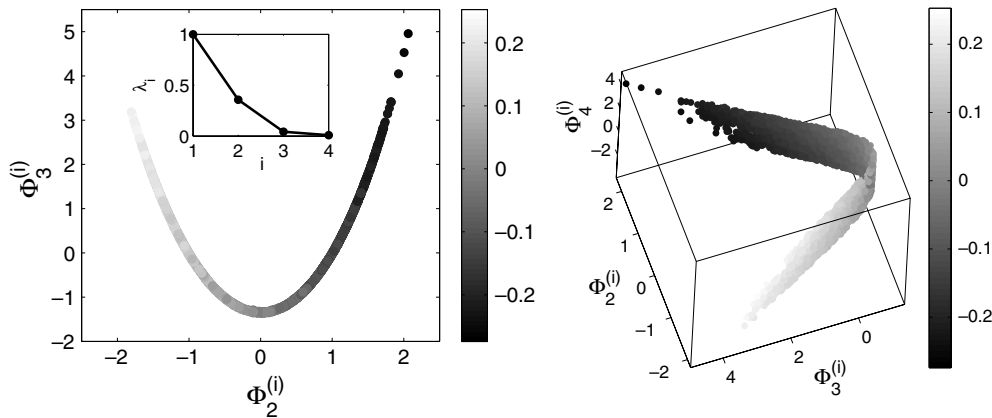
Figure 10. Left panel: diffusion map plotting components of the top two significant eigenvectors of Markov matrix *M* (defined in the appendix) constructed from simulation data. Datapoints are shaded by (and appear ordered according to) the value of the empirical coordinate $V(t)$ (defined in section 1). The inset shows the leading eigenvalue spectrum for the diffusion map. Right panel: 3D diffusion map plotting components of top three significant eigenvectors.

to access from the source point, and are therefore at equal *diffusion distance* from it, even though their Euclidean distances from the source may vary substantially. The procedure also provides a set of transformed coordinates; the Euclidean distance *in these new coordinates* is a true measure of intrinsic datapoint similarity. The procedure can be thought of as a nonlinear generalization of principal component analysis [25].

3.2. Results

In our case, we start from a long simulation (of 30 000 time units) during which we sample the $u(x)$, $a(x)$ profile every 8 time units, giving $N = 3\,750$ data points. We have spatially discretized with $M = 100$, so each of these data points is in $\mathbb{R}^{200}$. (From now on, we use the same parameter values as those in figure 1.) Using the procedure briefly outlined in the appendix we process these data forming a Markov matrix whose leading eigenvectors provide useful 'reduction coordinates' for our data. Figure 10 (left panel) shows the simulation data projected on the first two 'reduction coordinates'—the first two nontrivial eigenvectors Φ_2 and Φ_3 of the appropriate Markov matrix. The coordinates of the *i*th datapoint in this map are $(\Phi_2^{(i)}, \Phi_3^{(i)})$. The right panel of figure 10 shows a three-dimensional diffusion map, in terms of the first three nontrivial reduction coordinates. Simulation points in the diffusion map are coloured according to their associated value of the empirical coordinate V. It is clear that points are ordered along the curve by their values of this known reaction coordinate. This sorting of high-dimensional data vectors is obtained in an 'automated' fashion by the diffusion map calculation.

All data instances clearly collapse on a one-dimensional curve. Figure 11 indicates five representative data points (at values of $\Phi_2 = -1.8, -1, 0, 1$ and 1.8) on this curve. The insets, showing brief space-time plots initialized at the corresponding data points, clearly indicate that right-moving states reside on one end of this curve, left moving states on the opposite end, and transition states between the two reside in the interior of the curve. It is clear that the curve is one-to-one with the variable Φ_2 (for every value of Φ_2 there exists one point on the curve); this suggests that Φ_2 would be a good scalar observable for our system. Looking back, figure 2 (third row) shows the evolution of this 'computer-derived' observable during
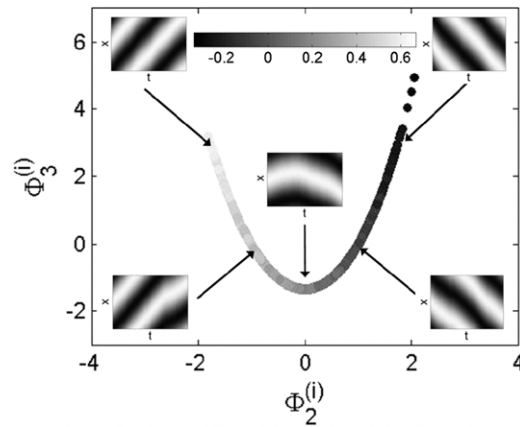
Figure 11. Two-dimensional diffusion map representation of long simulation data (as shown in figure 10). The insets show representative local solution evolution for u profiles initialized at five different regions of the map ($\Phi_2 = -1.8, -1, 0, 1$ and 1.8). Shadebar shown at top of figure indicates u values in space–time plots.
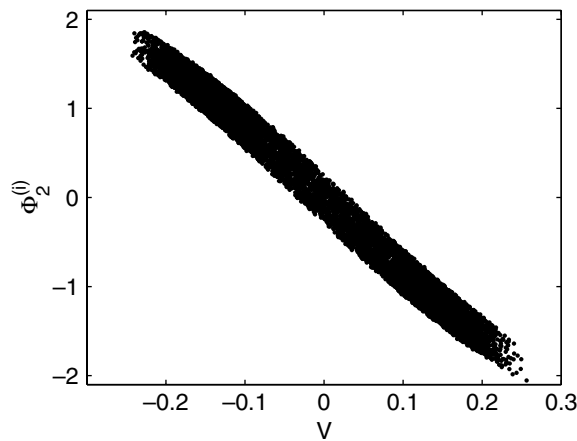

Figure 12. Coordinate $V(t)$ versus top significant eigenvector coordinate $\Phi_2^{(i)}$ for long simulation results.

our system dynamics; it is clearly capable of capturing the direction switches in the system in a manner comparable with the 'experience-based' coordinate V. Another possible scalar observable we considered is the arclength distance along the curve connecting components of the top two eigenvectors, i.e. arclength along the curve in the left panel of figure 10. The variation in this observable is comparable to that of Φ_2 and also accurately describes the direction of travel of the bump during the simulation (results not shown). Figure 12 plots the simulation data in terms of the two coarse coordinates: the experience-based V and the data-based Φ_2. At first sight, a one-to-one correspondence between the two observables is apparent, suggesting that our experience-based coordinate was indeed a good choice.

The processing of the database from a long, equilibrium run simulation that gave us an effective potential in terms of the observable V can now also be performed in terms of the computer-assisted observable Φ_2; the resulting two-well effective potential and the
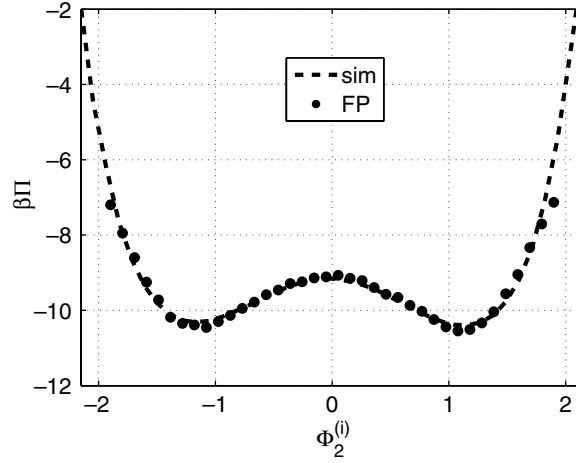
Figure 13. Effective potential $\beta\Pi$ estimated in terms of diffusion map coordinate $\Phi_2^{(i)}$. Dashed line: from assuming that $P(\Phi_2^{(i)}) \propto \exp[-\beta\Pi]$; points from assuming a Fokker–Planck equation, reconstructing $\mu(\Phi_2^{(i)})$ and $D(\Phi_2^{(i)})$ and computing the integral (9).

corresponding effective Langevin description are practically isomorphic to the V-based description. The effective potential computed in terms of the computer-assisted observable Φ_2 is shown in figure 13. Thus we have an alternative (equivalent) low-dimensional description of system (3) and (4) in terms of the variable Φ_2.

3.3. Computing with diffusion map coordinates

When we attempt to *directly* construct the effective potential in terms of the diffusion map coordinate Φ_2 (i.e. not from processing a long run database), it becomes necessary to initialize short simulation bursts consistent with particular values of Φ_2 as well as observe the Φ_2 values corresponding to results of detailed simulation. The diffusion map calculation that leads to the identification of the coordinate Φ_2 automatically provides its value on each of the data points used in the calculation (see the appendix). It therefore becomes important to establish computational protocols that routinely allow the translation between physical coordinates and diffusion map coordinates. This translation needs to be possible in both directions: to initialize a short system run we need to translate a value of Φ_2 to system initial conditions (living in $\mathbb{R}^{200}$); processing of the short simulation bursts involves extracting the Φ_2 values of the resulting states. These translation operations are termed 'lifting' and 'restriction' in the equation-free framework [20, 27]. We now briefly discuss possible implementations of such protocols, starting with the *restriction*: obtaining the Φ_2 values of new system configurations, not in the original database, that result from our short simulation bursts.

Referring to the appendix, for points *in the original dataset* (used to assemble the neighbourhood matrix $\boldsymbol{K}$) eigenvalues and eigenvectors of the symmetric kernel $\boldsymbol{M_s}$ are related by

$$\boldsymbol{M_s}\boldsymbol{\Psi}_j = \lambda_j\boldsymbol{\Psi}_j. \tag{11}$$

For a particular point $\boldsymbol{x}_i$ in the dataset we have

$$\sum_{k=1}^{N} \boldsymbol{M_s}(\boldsymbol{x}_i, \boldsymbol{x}_k)\Psi_j^{(k)} = \lambda_j\Psi_j^{(i)}. \tag{12}$$

For a new datapoint $\boldsymbol{x}^{(\text{new})}$ the same formula should be valid. However, while the right-hand side is unknown, the left-hand side may be computed for the new datapoint, which gives the following formula for the extension of the diffusion map

$$\Psi_j^{(\text{new})} = \frac{1}{\lambda_j} \sum_{k=1}^{N} \widetilde{\boldsymbol{M}}_{\mathbf{s}}(\boldsymbol{x}^{(\text{new})}, \boldsymbol{x}_k) \Psi_j^{(k)}, \tag{13}$$

where $\widetilde{\boldsymbol{M}}_{\mathbf{s}}$ is a generalized kernel which extends to new sample points and is given by

$$\widetilde{\boldsymbol{M}}_{\mathbf{s}}(\boldsymbol{x}^{(\text{new})}, \boldsymbol{x}_k) = \frac{1}{N} \frac{K(\boldsymbol{x}^{(\text{new})}, \boldsymbol{x}_k)}{\sqrt{E_{\boldsymbol{y}}\left(K(\boldsymbol{x}^{(\text{new})}, \boldsymbol{y})\right) E_{\boldsymbol{y}'}\left(K(\boldsymbol{x}_k, \boldsymbol{y}')\right)}}. \tag{14}$$

Equation (13) is called the Nyström formula [3] and allows eigenvector components associated with new data vectors $\boldsymbol{x}^{(\text{new})}$, outside of the original sample, to be computed by eigenspace interpolation instead of repeated eigendecomposition of the Markov matrix augmented by the new data vectors. The expectations appearing in the denominator of (14) are computed from the empirical data using

$$E_{\boldsymbol{y}}\left(K(\boldsymbol{x}, \boldsymbol{y})\right) = \frac{1}{N} \sum_{i=1}^{N} K(\boldsymbol{x}, \boldsymbol{y}_i). \tag{15}$$

Use of the Nyström formula (13) is central to the efficient computation of diffusion map coordinates associated with a high-dimensional vector (*restriction*) and also the preparation of a data vector $\boldsymbol{x}$ (*lifting*) with desired diffusion map coordinate values. Equation (13) is used to first compute the eigenvector components $\Psi_j^{(\text{new})}$ (associated with the symmetric matrix $\boldsymbol{M}_{\mathbf{s}}$); diffusion map coordinates (associated with $\boldsymbol{M}$) for the new data point may then be computed using

$$\Phi_j^{(\text{new})} = \frac{\Psi_j^{(\text{new})}}{\Psi_1^{(\text{new})}}. \tag{16}$$

The Nyström formula allows calculation of the diffusion map eigenvector components Φ_j^{new} associated with a new data vector $\boldsymbol{x}^{(\text{new})}$. A full eigendecomposition is typically performed first for a representative subset of simulation datapoints, identifying the relevant eigenvectors. The Nyström formula is then used to perform the restriction operation in (13) which amounts to interpolation in diffusion map space.

Our protocol for *lifting* from prescribed diffusion map coordinates to consistent system states using stochastic optimization is presented in figure 14. The main step in this lifting procedure, which produces $u(x)$ and $a(x)$ solution profiles with a specified diffusion map coordinate, Φ_2^{targ}, is the minimization of a quadratic objective function given by

$$\text{Obj}(\Phi_2^{(i)}) = \lambda_{\text{OBJ},2}(\Phi_2^{(i)} - \Phi_2^{\text{targ}})^2, \tag{17}$$

where $\lambda_{\text{OBJ},2}$ is a weighting parameter that controls the shape of the objective away from its minimum at $\Phi_2^{(i)} = \Phi_2^{\text{targ}}$. The implicit dependence of the eigenvector components $\Phi_j^{(i)}$ (associated with a data vector $\boldsymbol{x}^{(i)}$) on $\boldsymbol{x}^{(i)}$ and all other data vectors used in the construction of the neighbourhood matrix $\boldsymbol{K}$ makes this optimization problem challenging. We used here, for simplicity, the method of simulated annealing (SA) [28, 36] to minimize the objective function defined in (17), and identify a data vector $\boldsymbol{x}^{(\text{targ})}$ with the target diffusion map coordinate Φ_2^{targ}. Table 1 compares drift and diffusion coefficient values (at $\Phi_2 = -0.5$) estimated by multiple short simulation bursts, initialized using this lifting protocol, with values computed from a long time simulation. The agreement is very good.
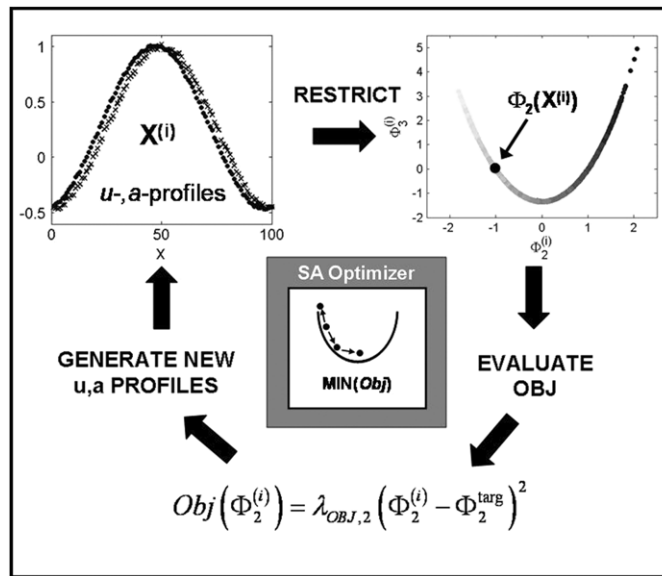
Figure 14. Protocol for lifting from diffusion map coordinates using SA. We show (top left) normalized u and a solution profiles which define the trial vector $x^{\text{(trial)}}$. This 200-dimensional vector is restricted, the 'location' of the data vector in the diffusion map is identified (top right) and the corresponding objective value computed (bottom). The objective value guides the selection of a new search direction (in $\mathbb{R}^{200}$), generating a new set of solution profiles (a new $x^{\text{(trial)}}$) and the above procedure is repeated.

Table 1. Drift and diffusion coefficients at diffusion map coordinate value $\Phi_2^* = -0.5$ estimated by finding occurrences of Φ_2^* in a long simulation (database) and by direct initialization at this value (lifting).

Method	$\mu(\Phi_2 = \Phi_2^*)$	$D(\Phi_2 = \Phi_2^*)$
Database	-3.51×10^{-3}	3.30×10^{-3}
Lifting	-3.39×10^{-3}	3.20×10^{-3}

Given these protocols for lifting and restricting, the entire effective potential and, more generally, all the bifurcation/switching time computations performed above with our empirical V coarse variable can be repeated with the Φ_2 variable—a variable extracted in an automated fashion from a single simulation of the original system. 'Coarse-grained smoothness' in the diffusion map coordinates can be exploited to guide the efficient exploration of the effective potential surface [18]. Note that due to the 'lifting/restricting' computations involved, calculations using Φ_2 instead of V generally take longer, for a particular accuracy in the estimate of Π.

4. Discussion and conclusion

In this paper we have revisited a stochastic spatio-temporal pattern forming system originally used to model activity in the cortex. Previous work [31] investigated the effects of changing two parameters (spike frequency adaptation strength and noise intensity) on the dynamics of a spatially localized 'bump' of neural activity. Here, we have concentrated on deriving and using an effective low-dimensional description of the system in terms of a single scalar coarse

variable, thus simplifying the system. This variable was assumed to satisfy an (unknown) effective Langevin equation, so that its probability density satisfies an effective Fokker–Planck equation. By appropriate processing of the results of a long simulation, or—more efficiently— by appropriately initializing short bursts of simulation of the full system, we estimated the drift and diffusion functions that appear in the Fokker–Planck equation, and thus determined an 'effective potential' for the system on demand. We have thus bypassed the difficult (if not impossible) task of deriving such a potential from the original high-dimensional, stochastic system.

In the second part of the paper we showed how a similar analysis can be performed using a variable that is extracted by data-mining a sufficiently long system simulation in an automated fashion, using the diffusion map approach [32,33]. Such an approach should prove particularly useful when applying these ideas to other systems, since in general a good coordinate (or coordinates) for a low-dimensional description of a complex system may not be known, nor easy to guess. The results obtained using this new variable showed good correspondence with those obtained in the first part of the paper.

Considering more general systems, it is known that in certain limiting cases (e.g. weak coupling) it is possible to explicitly obtain accurate reduced descriptions (e.g. phase oscillator equations) for large coupled neural systems [16,24]. Clearly, if such equations can be derived, they are much easier to use than the approach presented here; our approach is intended for cases where we believe the reduction is possible, but cannot be explicitly performed. One of the most challenging tests for a reduction—as we move away from the conditions where we can guarantee its validity analytically—is to test, on line, whether it is accurate, and— importantly—whether a *different level* reduction, with more (or even possibly with fewer) variables is in order. In our case, this would correspond to devising tests to suggest, on line, that *more than one* coarse-grained variable must take part in our effective Langevin equation. The slight 'thickness' of the line in figure 12 could be an indication that, for the parameter values used, more than one variable may be necessary in our model. Devising such tests— in effect, testing the hypothesis that the data are locally consistent with a particular model, e.g. a scalar Langevin equation—is the subject of ongoing research across several disciplines (statistics, financial mathematics). Integrating such tools in a multiscale simulation framework is only just starting.

Coarse-graining large scale, faithful neural network computations constitutes an important subject of intense current research (see, for example, the probability density approach [6, 29, 34]). Multiscale, coarse-graining numerical algorithms such as the one demonstrated here have an important role to play in elucidating the types of behaviour possible for such models and their parametric dependence.

Even though the model discussed here is rather abstract (from a computational neuroscience point of view) it adds to the body of work demonstrating that it is possible to obtain useful low-dimensional descriptions of complex systems, both in neural modelling [30] and elsewhere [13, 27, 39]. These descriptions not only provide insight into the fundamental dynamics, but enable one to simulate and analyse such systems in an efficient manner. We expect the methods described here to be widely applicable to other complex systems.

Appendix

We now give a brief description of the calculation of the coordinates $\boldsymbol{\Phi}_2$, $\boldsymbol{\Phi}_3$, . . . discussed in section 3.

We run a simulation of (3) and (4) for 30 000 time units and sample the $u(x)$, $a(x)$ profile every 8 time units, giving $N = 3\,750$ data points. We have discretized space with $M = 100$ points, so every data point is actually a vector of 200 values. Using i to index the data points, we normalize each u_i profile by its maximum value (at node $m_i \in \{1, \ldots M\}$) and normalize the a_i profile by its nodal value at the corresponding location ($a_i(m_i)$). Finally, we shift both solution profiles by a fixed number of nodes such that the alignment between u_i profiles is maximized (in a least squares sense) thereby factoring out translations. The components of the simulation data vector $\boldsymbol{x}_i$ are given by the resulting $2M$ nodal values of u_i and a_i. We define a pairwise similarity ('neighbourhood') matrix $\boldsymbol{K}$ between a representative sample of these data vectors (collected over the course of a simulation run) as follows:

$$K_{i,j} = K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left[-\left(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right)^2\right], \tag{A.1}$$

where σ is a parameter that defines the size of the local neighbourhood surrounding each high-dimensional point. Defining the diagonal normalization matrix $D_{i,i} = \sum_j K_{i,j}$, we construct the Markovian matrix $\boldsymbol{M} = \boldsymbol{D}^{-1}\boldsymbol{K}$. A few top eigenvectors of the matrix $\boldsymbol{M}$, namely $\boldsymbol{\Phi}_2$, $\boldsymbol{\Phi}_3$, . . . , are used here as a low-dimensional representation of the simulation data. Components in the eigenvector provide the low-dimensional coordinate for each simulation data vector. The diffusion map distance is equal to Euclidean distance in the *diffusion map space*. In many applications the spectrum of the matrix $\boldsymbol{M}$ possesses a spectral gap, and this diffusion distance may be approximated by using only a few top eigenvectors. This approximation has been shown to be optimal under a particular mean squared error criterion [32].

The matrix $\boldsymbol{M}$ is adjoint to the symmetric matrix $\boldsymbol{M}_{\mathbf{s}}$ defined as follows:

$$\boldsymbol{M}_{\mathbf{s}} = \boldsymbol{D}^{1/2}\boldsymbol{M}\,\boldsymbol{D}^{-1/2} = \boldsymbol{D}^{-1/2}\boldsymbol{K}\,\boldsymbol{D}^{-1/2}. \tag{A.2}$$

$\boldsymbol{M}$ and $\boldsymbol{M}_{\mathbf{s}}$ share the same eigenvalues; eigenvectors $\boldsymbol{\Phi}_j$ of $\boldsymbol{M}$ are related to those of $\boldsymbol{M}_{\mathbf{s}}$, denoted $\boldsymbol{\Psi}_j$, as follows:

$$\boldsymbol{\Phi}_j = \boldsymbol{D}^{-1/2}\boldsymbol{\Psi}_j. \tag{A.3}$$

We compute the largest eigenvalues and eigenvectors of the symmetric matrix $\boldsymbol{M}_{\mathbf{s}}$ and use (A.3) to evaluate the eigenvectors of the Markov matrix $\boldsymbol{M}$. We note that the top eigenvector $\boldsymbol{\Psi}_1$ of $\boldsymbol{M}_{\mathbf{s}}$, corresponding to the largest eigenvalue $\lambda_1 = 1$, has components equal to the diagonal entries of $\boldsymbol{D}^{1/2}$; it follows from (A.3) that the top, *trivial*, eigenvector $\boldsymbol{\Phi}_1$ of $\boldsymbol{M}$ consists entirely of ones.

The diffusion map in the left panel of figure 10 is then

$$\mathbf{x}_i \mapsto \begin{pmatrix} \boldsymbol{\Phi}_2^{(i)} \\ \boldsymbol{\Phi}_3^{(i)}, \end{pmatrix} \in \mathbb{R}^2, \tag{A.4}$$

where $\boldsymbol{\Phi}_k^{(i)}$ is the ith component of $\boldsymbol{\Phi}_k$.

References

[1] Ait-Sahalia Y 1999 Transition densities for interest rate and other nonlinear diffusions *J. Finance.* **54** 1361–95

[2] Doedel E J, Champneys A R, Fairgrieve T F, Kuznetsov Y A, Sandstede B and Wang X J AUTO97: continuation and bifurcation software for ordinary differential equations; available by FTP from ftp.cs.concordia.ca. in directory pub/doedel/auto

[3] Baker C 1977 *The Numerical Treatment of Integral Equations* (Oxford: Clarendon)

[4] Belkin M and Niyogi P 2003 Laplacian eigenmaps for dimensionality reduction and data representation *Neural Comput.* **15** 1373–96

[5] Ben-Yishai R, Bar-Or R L and Sompolinsky H 1995 Theory of orientation tuning in visual cortex *Proc. Natl Acad. Sci. USA* **92** 3844–8

[6] Cai D, Tao L, Shelley M and McLaughlin D W 2004 An effective kinetic representation of fluctuation-driven neuronal networks with application to simple and complex cells in visual cortex *Proc. Natl. Acad. Sci. USA* **101** 7757–62

[7] Coifman R, Lafon S, Lee A, Maggioni M, Nadler B, Warner F and Zucker S 2005 Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps *Proc. Natl Acad. Sci. USA* **102** 7426–31

[8] Coifman R, Lafon S, Lee A, Maggioni M, Nadler B, Warner F and Zucker S 2005 Geometric diffusions as a tool for harmonic analysis and structure definition of data: multiscale methods *Proc. Natl Acad. Sci. USA* **102** 7432–7

[9] Coombes S and Owen M R 2004 Evans functions for integral neural field equations with Heaviside firing rate function *SIAM J. Appl. Dyn. Syst.* **34** 574–600

[10] Coombes S and Owen M R 2005 Bumps, breathers, and waves in a neural network with spike frequency adaptation *Phys. Rev. Lett.* **94** 148102

[11] Curtu R and Ermentrout B 2004 Pattern formation in a network of excitatory and inhibitory cells with adaptation *SIAM J. Appl. Dyn. Syst.* **3** 191-231

[12] Donoho D L and Grimes C 2003 Hessian eigenmaps: locally linear embedding techniques for high-dimensional data *Proc. Natl Acad. Sci. USA* **100** 5591–6

[13] Erban R, Kevrekidis I G, Adalsteinsson D and Elston T C 2006 Gene regulatory networks: a coarse-grained, equation-free approach to multiscale computation *J. Chem. Phys.* **124** 084106

[14] Erban R, Frewen T A, Wang X, Elston T C, Coifman R R, Nadler B and Kevrekidis I G 2007 Variable-free exploration of stochastic models: a gene regulatory network example *J. Chem Phys.* **126** 155103

[15] Ermentrout B 1998. Neural networks as spatio-temporal pattern-forming systems *Rep. Prog. Phys.* **61** 353–430

[16] Ermentrout B, Pascal M and Gutkin B 2001 The effects of spike frequency adaptation and negative feedback on the synchronization of neural oscillators *Neural Comput.* **13** 1285–310

[17] Ermentrout B and Saunders D 2006 Phase resetting and coupling of noisy neural oscillators *J. Comput. Neurosci.* **20** 179-90

[18] Frewen T A, Hummer G and Kevrekidis I G 2006 Equation-free exploration of free energy surfaces, in preparation

[19] Fronzoni L, Mannella R, McClintock PVE, and Moss F 1987 Postponement of Hopf bifurcations by multiplicative colored noise *Phys. Rev.* A **36** 834–41

[20] Gear C W, Kevrekidis I G, and Theodoropoulos C 2002 'Coarse' integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods *Comput. Chem. Eng.* **26** 941–63

[21] Gradišek J, Siegert S, Friedrich R and Grabec I 2000 Analysis of time series from stochastic processes *Phys. Rev.* E **62** 3146–55

[22] Gutkin B, Hely T and Jost J 2004. Noise delays onset of sustained firing in a minimal model of persistent activity. *Neurocomputing* **58–60** 753–60

[23] Hansel D and Sompolinsky H 1998 Modeling feature selectivity in local cortical circuits *Methods in Neuronal Modeling: From Ions to Networks* 2nd edn ed C Koch and I Segev (Cambridge, MA: MIT Press) pp. 499-567

[24] Izhikevich E M 2000 Phase equations for relaxation oscillators *SIAM J. Appl. Math.* **60** 1789–805

[25] Jolliffe I T 1986 *Principal Component Analysis (Springer Series in Statistics)* (New York: Springer)

[26] Kelley C T 1995 *Iterative Methods for Linear and Nonlinear Equations (Frontiers in Applied Mathematics* vol 16) (Philadelphia: SIAM)

[27] Kevrekidis I G, Gear C W, Hyman J M, Kevrekidis P G, Runborg O and Theodoropoulos C 2003 Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis *Commun. Math. Sci.* **1** 715–62

[28] Kirkpatrick S, Gelatt C and Vecchi M 1983 Optimization by simulated annealing *Science* **34** 671–80

[29] Knight B W 2000 Dynamics of encoding in neuron populations: Some general mathematical features *Neural Comput.* **12** 473–518

[30] Laing C R 2006 On the application of 'equation-free' modelling to neural systems *J. Comput. Neurosci.* **20** 5–23

[31] Laing C R and Longtin A 2001 Noise-induced stabilization of bumps in systems with long-range spatial coupling *Physica* D **160** 149–72

[32] Nadler B, Lafon S, Coifman R R and Kevrekidis I G 2005 Diffusion maps, spectral clustering and eigenfunctions of Fokker–Planck operators *Advances in Neural Information Processing Systems* vol 18, ed Y Weiss *et al* (Cambridge, MA: MIT Press) pp 955-62

[33] Nadler B, Lafon S, Coifman R R and Kevrekidis I G 2006 Diffusion maps, spectral clustering and reaction coordinates of dynamical systems *Appl. Comput. Harmon. Anal.* **21** 113–27

[34] Omurtag A, Knight B W and Sirovich L 2000 On the simulation of large populations of neurons *J. Comput. Neurosci.* **8** 51-63

[35] Pinto D J and Ermentrout G B 2001 Spatially structures activity in synaptically coupled neuronal networks: II. Lateral inhibition and standing pulses *SIAM J. Appl. Math.* **62** 226–43

[36] Press W, Teukolsky S, Vetterling W and Flannery B 1992 *Numerical Recipes* (Cambridge: Cambridge University Press)

[37] Qiao L, Erban R, Kelley C T and Kevrekidis I G 2006 Spatially distributed stochastic systems: equation-free and equation-assisted preconditioned computations *J. Chem. Phys.* **125** 204108

[38] Roweis S T and Saul L K 2000 Nonlinear dimensionality reduction by locally linear embedding *Science* **290** 2323–6

[39] Sriraman S, Kevrekidis I G and Hummer G 2005 Coarse nonlinear dynamics and metastability of filling-emptying transitions: water in carbon nanotubes *Phys. Rev. Lett.* **95** 130603

[40] Tenenbaum J B, de Silva V and Langford J C 2000 A global geometric framework for nonlinear dimensionality reduction *Science* **290** 2319–23