

Managing heterogeneity in the study of neural oscillator dynamics

Carlo R Laing*¹ , Yu Zou² , Ben Smith^{1,3} and Ioannis G Kevrekidis²

¹Institute of Information and Mathematical Sciences, Massey University, Private Bag 102-904 North Shore Mail Centre, Auckland 0632, New Zealand

²Department of Chemical and Biological Engineering and Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, 08544, USA

³Research Centre for Cognitive Neuroscience, Department of Psychology, University of Auckland, Private Bag 92019, Auckland 1142, New Zealand

*Corresponding author: c.r.laing@massey.ac.nz

Email: CRL: c.r.laing@massey.ac.nz

YZ: yzou@princeton.edu

BS: benjsmith@gmail.com

IGK: yannis@princeton.edu

;

*Corresponding author

Abstract

We consider a coupled, heterogeneous population of relaxation oscillators used to model rhythmic oscillations in the pre-Bötzinger complex. By choosing specific values of the parameter used to describe the heterogeneity, sampled from the probability distribution of the values of that parameter, we show how the effects of heterogeneity can be studied in a computationally efficient manner. When more than one parameter is heterogeneous, full or sparse tensor product grids are used to select appropriate parameter values. The method allows us to effectively reduce the dimensionality of the model, and it provides a means for systematically investigating the effects of heterogeneity in coupled systems, linking ideas from uncertainty quantification to those for the study of network dynamics.

Keywords: heterogeneity; neural oscillators; pre-Bötzinger complex; model reduction; bifurcation; computation.

1 Introduction

Networks of coupled oscillators have been studied for a number of years [1–7]. One motivation for these studies is that many neurons, when isolated (and possibly injected with a constant current), either periodically fire action potentials [8, 9] or periodically move between quiescence and repetitive firing (the alternation being referred to as bursting [10, 11]). In either case, the isolated neuron can be thought of

as an oscillator. Neurons are typically coupled with many others via either gap junctions [12] or chemical synapses [13–15]; hence, a group of neurons can be thought of as a network of coupled oscillators.

As an idealisation, one might consider identical oscillators; in which case, the symmetry of the network will often determine its possible dynamics [16, 17]. However, natural systems are never

ideal, and thus, it is more realistic to consider *heterogeneous* networks. Also, there is evidence in a number of contexts that heterogeneity within a population of neurons can be beneficial. Examples include calcium wave propagation [18], the synchronisation of coupled excitable units to an external drive [19,20], and the example we study here: respiratory rhythm generation [13,21].

One simple way to incorporate heterogeneity in a network of coupled oscillators is to select one parameter which affects the individual dynamics of each oscillator and assign a different value to this parameter for each oscillator [3,15,22,23]. Doing this raises natural questions such as from which distribution should these parameter values be chosen, and what effect does this heterogeneity have on the dynamics of the network?

Furthermore, if we want to answer these questions in the most computationally efficient way, we need a procedure for selecting a (somehow) optimal representative set of parameter values from this distribution. In this paper, we will address some of these issues.

In particular, we will show how - given the distribution(s) of the parameter(s) describing the heterogeneity - the representative set of parameter values can be chosen so as to accurately incorporate the effects of the heterogeneity without having to fully simulate the entire large network of oscillators.

We investigate one particular network of coupled relaxation oscillators, derived from a model of the pre-Bötzinger complex [13,14,24], and show how the heterogeneity in one parameter affects its dynamics. We also show how heterogeneity in more than one parameter can be incorporated using either full or sparse tensor product grids in parameter space.

Our approach thus creates a bridge between computational techniques developed in the field of uncertainty quantification [25,26] involving collocation and sparse grids on the one hand, and network dynamics on the other. It also helps us build accurate, reduced computational models of large coupled neuron populations.

One restriction of our method is that it applies only to states where all oscillators are synchronised

(in the sense of having the same period) or at a fixed point. Synchronisation of this form typically occurs when the strength of coupling between oscillators is strong enough to overcome the tendency of non-identical oscillators to desynchronise due to their disparate frequencies [2,3,27] and is often the behaviour of interest [6,13,14,23].

We present the model in Section 2 and show how to efficiently include parameter heterogeneity in Section 3. In Section 4, we explore how varying heterogeneity modifies bifurcations and varies the period of the collective oscillation. Sections 5 and 6 show how to deal with two and more, respectively, heterogeneous parameters. We conclude in Section 7.

2 The model

Our illustrative example is a network of model neurons thought to describe at some level the dynamics of the pre-Bötzinger complex, governed by the following equations:

$$C \frac{dV_i}{dt} = -g_{\text{Na}} m(V_i) h_i (V_i - V_{\text{Na}}) - g_l (V_i - V_l) + I_{\text{syn}}^i + I_{\text{app}}^i \quad (1)$$

$$\frac{dh_i}{dt} = \frac{h_{\infty}(V_i) - h_i}{\tau(V_i)} \quad (2)$$

for $i = 1 \dots N$, where

$$I_{\text{syn}}^i = \frac{g_{\text{syn}}(V_{\text{syn}} - V_i)}{N} \sum_{j=1}^N s(V_j), \quad (3)$$

as considered in the work of Rubin and Terman [14]. Here, V_i is the membrane potential of cell i , and h_i is a channel state variable for neuron i governing the inactivation of persistent sodium. Equations 1 and 2 were derived from the model in the works of Butera et al. [13,24] by blocking currents responsible for action potentials. A similar model with $N = 2$ was considered in the work of Rubin [28], and Dunmyre and Rubin [29] considered synchronisation in the case $N = 3$, where one of the neurons was quiescent, another was tonically firing, and the third one could be either quiescent, tonically firing or bursting. The neurons are all-to-all coupled via the term I_{syn}^i ; when $g_{\text{syn}} = 0$ the neurons are uncoupled. The various functions involved in the model equations are

the following:

$$s(V) = \frac{1}{1 + \exp[-(V + 40)/5]} \quad (4)$$

$$\tau(V) = \frac{1}{\epsilon \cosh[(V + 44)/12]} \quad (5)$$

$$h_\infty(V) = \frac{1}{1 + \exp[(V + 44)/6]} \quad (6)$$

$$m(V) = \frac{1}{1 + \exp[-(V + 37)/6]} \quad (7)$$

The functions $\tau(V)$, $h_\infty(V)$ and $m(V)$ are a standard part of the Hodgkin-Huxley formalism [8], and synaptic communication is assumed to act instantaneously through the function $s(V)$. The parameter values we use initially are $V_{Na} = 50$, $g_l = 2.4$, $V_l = -65$, $V_{syn} = 0$, $C = 0.21$, $\epsilon = 0.1$, $g_{syn} = 0.3$ and $g_{Na} = 2.8$.

Note that the synaptic coupling is excitatory. These parameters are the same as those used in the work of Rubin and Terman [14] except that they used $\epsilon = 0.01$ and $g_l = 2.8$, and their function $s(V)$ had a more rapid transition from approximately 0 to 1 as V was increased. These changes in parameter values were made to speed up the numerical integration of Equations 1 and 2, and the methods presented here do not depend on the particular values of these parameters.

If the values of the applied current I_{app}^i are taken from a uniform distribution on the interval [10, 25], the behaviour is as shown in Figure 1. After a transient, we see synchronous behaviour, i.e. all neurons oscillate periodically with the same period, although the heterogeneity in the I_{app}^i means that each neuron follows a slightly different periodic orbit in its own (V, h) phase space. (Because spiking currents have been removed in the derivation of Equations 1 and 2, these oscillations are interpreted as burst envelopes, i.e. neuron i is assumed to be spiking when V_i is high and quiescent when V_i is low.) It is this stable synchronous periodic behaviour that is of interest: In what parameter regions does it exist, and how does the period vary as parameters are varied? Butera et al. [13] observed that including parameter heterogeneity in a spiking model for the pre-Bötzinger complex increased both the range of parameters over which bursting occurred and the range of burst frequencies (this being functionally advantageous for respiration), and this was the motivation

for the study of Rubin and Terman [14].

3 Managing heterogeneity

3.1 The continuum limit

The key observation behind our approach can be seen in Figure 2, where we plot the V_i and $s(V_i)$ as functions of I_{app}^i at one instant in time. Once the neurons have synchronised, V_i values (and h_i and any smooth functions of these variables) appear to vary smoothly when plotted as a function of the heterogeneous parameter I_{app}^i . This is also the case when the I_{app}^i values are chosen randomly from the interval [10, 25] rather than uniformly (not shown). This suggests that in the limit of $N \rightarrow \infty$, at any instant in time, V and h will be smooth functions of the continuous variable I_{app} . We now consider this case where I_{app} is a continuous random variable with a uniform density on the interval [10, 25]. We parametrise I_{app} as $I_{app} = I_m + I_s \mu$, where the probability density function for μ is as follows:

$$p(\mu) = \begin{cases} 1/2, & -1 \leq \mu \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$V_i(t)$ and $h_i(t)$ become $V(\mu, t)$ and $h(\mu, t)$, respectively, and the points in Figure 2 ‘fill in’ to form continuous functions. In the given example, we had $I_m = 17.5$ and $I_s = 7.5$. Thus, the ordinary differential equations (ODEs) 1 and 2 become the following:

$$C \frac{\partial V(\mu, t)}{\partial t} = -g_{Na} m(V(\mu, t)) h(\mu, t) (V(\mu, t) - V_{Na}) - g_l (V(\mu, t) - V_l) + I_{syn}(\mu, t) + I_m + I_s \mu \quad (9)$$

$$\frac{\partial h(\mu, t)}{\partial t} = \frac{h_\infty(V(\mu, t)) - h(\mu, t)}{\tau(V(\mu, t))} \quad (10)$$

where

$$I_{syn}(\mu, t) = g_{syn} (V_{syn} - V(\mu, t)) \int_{-1}^1 s(V(\mu, t)) p(\mu) d\mu. \quad (11)$$

The results for $N \rightarrow \infty$ should provide a good approximation to the behaviour seen when N is large but finite, which is the realistic (although difficult to simulate) case. The continuum limit presented in this section was first introduced by Rubin and Terman [14], but their contribution was largely analytical, whereas ours will be largely numerical.

3.2 Stochastic Galerkin

One approach to studying Equations 9 and 11, motivated by techniques developed in the context of uncertainty quantification [25, 26], is to expand the functions $V(\mu, t)$ and $h(\mu, t)$ in orthogonal polynomials in μ , with the choice of particular polynomials determined by the probability density of μ , i.e. the distribution of the heterogeneous parameter. For the uniform density $p(\mu)$, one would choose Legendre polynomials, writing $V(\mu, t)$ and $h(\mu, t)$ as follows:

$$V(\mu, t) = \sum_{i=0}^{\infty} a_i(t) P_i(\mu), \quad h(\mu, t) = \sum_{i=0}^{\infty} b_i(t) P_i(\mu), \quad (12)$$

where P_i is the i th Legendre polynomial; this is known as a ‘polynomial chaos’ expansion [3]. Substituting Equation 12 into Equation 9, multiplying both sides by $P_j(\mu)p(\mu)$ and integrating over μ between -1 and 1 , the orthogonality properties of Legendre polynomials with uniform weight allows one to obtain the ODE satisfied by $a_j(t)$. Similarly, one can use Equation 10 to obtain the ODEs governing the dynamics of $b_j(t)$. Having solved (a truncated set of) these ODEs, one could reconstruct $V(\mu, t)$ and $h(\mu, t)$ using Equation 12. This is referred to as the stochastic Galerkin method [25]. However, the integrals just mentioned cannot be performed analytically. They must be calculated numerically at each time step in the integration of the ODEs for a_i and b_i ; this is computationally intensive. Note that the optimal choice of orthogonal polynomials is determined by the distribution of the heterogeneous parameter: for a uniform distribution, we use Legendre polynomials; for other distributions, other families of orthogonal polynomials are used [25, 26].

3.3 Stochastic collocation

An alternative, motivated by the stochastic collocation method [25], is to simply discretise in the μ direction, obtaining N different μ_i values, and then solve Equations 9 and 10 at each of the μ_i , using the values of $s(V(\mu_i, t))$ to approximate the integral in Equation 11.

It is important to realize that the number (N) of neurons simulated in this approach may well be much smaller than the number of neurons in the ‘true’ system, considered to be in the thousands. Notice also that these neurons are ‘mathematically’ coupled to one another via the discretisation of the

integral (Equation 11), which is an approximation of the continuum limit.

Using the values of $s(V(\mu_i, t))$ to approximate the integral in Equation 11, we are in fact including the influence of *all* other neurons (an infinite number of them in the continuum limit), not just those that we have retained in our reduced approximation. We now examine how different discretisation schemes affect several different calculations.

3.3.1 Period calculation

Firstly, we consider the period of the collective oscillations seen in Figure 1. The analogue of finite differences, or the method of lines, is to uniformly discretise the interval $[-1, 1]$ into N values, μ_i , and to solve Equations 9 and 10 at each of the μ_i . Defining $\mu_i = -1 + 2(i - 1/2)/N$ for $i = 1, 2 \dots N$, we approximate the integral in Equation 11 using the composite midpoint rule:

$$\int_{-1}^1 s(V(\mu, t)) p(\mu) d\mu \approx \frac{1}{N} \sum_{i=1}^N s(V(\mu_i, t)) \quad (13)$$

which, after defining $V_i(t) = V(\mu_i, t)$, is nothing more than the sum in Equation 3, where $I_{\text{app}}^i = I_m + I_s \mu_i$. To show convergence of the calculation of the period with N , we plot the error in Figure 3 with red stars; the error is defined to be the absolute value of the difference between the calculated period and the true period (defined at the end of this section). We see that the error scales as N^{-2} as expected from numerical analysis [30]. (All numerical integration was performed using Matlab’s `ode113` with an absolute tolerance of 10^{-10} and a relative tolerance of 10^{-12} .)

However, by choosing non-uniformly spaced values of μ_i , we can evaluate the integral in Equation 13 much more accurately. (By ‘more accurately’, we mean either that for a fixed N , using the non-uniformly spaced μ_i will result in a smaller error than that obtained using uniform spacing, or that to obtain a fixed accuracy, using non-uniform spacing will require a smaller N than that needed for uniform spacing.) Specifically, for a fixed N , if we choose μ_i to be the i th root of $P_N(\mu)$, where P_N is the N th Legendre polynomial, normalised so that

$P_N(1) = 1$, and the weights

$$w_i = \frac{1}{(1 - \mu_i^2) [P'_N(\mu_i)]^2}, \quad (14)$$

then the Gauss-Legendre quadrature rule [31] is

$$\int_{-1}^1 s(V(\mu, t)) p(\mu) d\mu \approx \sum_{i=1}^N w_i s(V(\mu_i, t)). \quad (15)$$

Convergence of the error in the period with N is shown in Figure 3 (blue circles), where we see the very rapid convergence expected from a spectral method. For $50 \lesssim N$, the error in the period calculation using this method is dominated by errors in the numerical integration of the Equations 9 and 10 in time, rather than in the approximate evaluation of the integral in Equation 11. (The true period was calculated using Gauss-Legendre quadrature with N significantly larger than 10^4 and is approximately 8.040104851819.) The rapid convergence of Gauss-Legendre quadrature is a consequence of the fact that the function $s(V(\mu))$ is a sufficiently smooth function of μ (see Figure 2). This smoothness will arise only when the oscillators become fully synchronised.

3.3.2 Hopf bifurcations

By decreasing or increasing I_m (the mean of the I_{app}^i), we find that the oscillations in Figure 1 terminate in Hopf bifurcations. We now examine the effects of the different discretisations mentioned on the detection of these Hopf bifurcations. In Figure 4, we see the error in calculating the value of I_m at which the upper Hopf bifurcation occurs as a function of N , the number of points used, for the two different schemes (the true value, again calculated using Gauss-Legendre quadrature with a large N , is approximately $I_m = 33.1262$).

The expected behaviour (very rapid convergence for Gaussian quadrature and the error scaling as N^{-2} for the composite midpoint rule) is seen (compare with Figure 3). Figure 5 shows a similar calculation but for the lower Hopf bifurcation which occurs at $I_m \approx 6.064$. Several interesting points in contrast with the results in Figure 4 are evident: The error in the composite midpoint rule appears to decay as N^{-1} , while the error using Gaussian quadrature appears to decay as N^{-2} . The reason for these differences is not clear.

3.4 Summary

In this section, we have shown that a judicious choice of the values of the heterogeneous parameter, combined with a scheme for Gaussian quadrature, allows us to calculate quantities of interest (such as the period of oscillation and the parameter value at which a Hopf bifurcation occurs) much more parsimoniously than a naive implementation of uniformly spaced I_i values for a uniform distribution. Effectively, we have simulated the behaviour of a large network of oscillators by actually simulating a much smaller one, carefully choosing *which* oscillators to simulate (and how to couple them so as to also capture the effect of the omitted ones).

Having demonstrated this, we now fix $N = 10$ and use the quadrature rule given in Equation 15. Note that our discretisation in μ can be thought of in two different ways. Firstly, we can consider the continuum limit ($N \rightarrow \infty$) as the true system, whose dynamics will be close to the real system which consists of a large number of neurons. Our scheme is then an efficient way of simulating this true system. The other interpretation is that the true system consists of a large, finite number of neurons with randomly distributed parameter(s), and our scheme is a method for simulating such a system but using far fewer oscillators.

In the next section, we investigate the effects of varying I_m, I_s and g_{syn} . In a later section, we consider more than one heterogeneous parameter and show how tensor product grids and sparse tensor product grids can be used to accurately calculate the effects of further, independently distributed, heterogeneities.

4 The effects of heterogeneity

4.1 A single neuron

In order to investigate the effects of heterogeneity, we first examine a single uncoupled neuron (i.e. $N = 1$ and $g_{\text{syn}} = 0$). The behaviour as I_m is varied as shown in Figure 6 (left panel). For this range of I_m , there is always one fixed point, but it undergoes two Hopf bifurcations as I_m is varied, leading to a family of stable periodic orbits. The period decreases monotonically with increasing I_m . The lower

Hopf bifurcation results in a canard periodic solution [32] which very rapidly increases in amplitude as I_m is increased. This is related to the separation of time scales between the V dynamics (fast) and the h dynamics (slow). In the left panel of Figure 6, we see that some of the neurons in the network whose behaviour is shown in Figure 1 would be quiescent *when uncoupled*, while most would be periodically oscillating.

The behaviour in the left panel of Figure 6 can also be understood by looking at the (V, h) phase plane for different values of I_m - see Figure 7. The behaviour of one self-coupled neuron ($N = 1, g_{\text{syn}} = 0.3$) is shown in Figure 6 (right panel). We see that the main effect of self-coupling is to move both Hopf bifurcations to lower values of I_m .

4.2 A coupled population of neurons

Now, consider a coupled heterogeneous population with $N = 10$ neurons. Parameter values are $g_{\text{syn}} = 0.3$ and $I_s = 7.5$. (Note that if $I_s = 0$, we recover the results for one self-coupled neuron.) The results from varying I_m are shown in Figure 8. Comparing with the right panel of Figure 6, we see that including heterogeneity widens the range of I_m values for which oscillations occur. The periodic orbit cannot be followed below $I_m \approx 8$, as more complex oscillations than purely periodic occur (not shown), as discussed below. Note that the mean voltage at the fixed point is easily calculated as $\bar{V} \equiv \sum_{i=1}^{10} w_i V_i$, where V_i is the steady state value of neuron i , and the variance of the V_i 's is simply $\sum_{i=1}^{10} w_i (V_i - \bar{V})^2$. (Recall that the weights w_i are given in Equation 14.)

To better understand these results, we can follow the Hopf bifurcations as *two* parameters are varied. Figure 9 (top) shows the two curves of Hopf bifurcations in the I_m, I_s plane for $g_{\text{syn}} = 0.3$. Increasing the ‘spread’ of the heterogeneity, i.e. increasing I_s , increases the range of values of I_m for which periodic oscillations are possible (between the Hopf bifurcations), but there may not necessarily exist stable periodic orbits over the entire range. For I_s larger than about 6, i.e. for very heterogeneous neurons, the synchronous behaviour created in the rightmost Hopf bifurcation shown in Figure 9 (top) breaks up as I_m is decreased at constant I_s , leading to complex oscillations (not shown). The break-up of the

synchronous behaviour always involves the neurons with the lowest values of μ , i.e. the lowest values of I_{app} . The curve in Figure 9 (top) where synchronous behaviour breaks up was found by slowly decreasing I_m at constant I_s until the break-up was observed. In principle, it could be found by numerical continuation of the stable periodic orbit created in the rightmost Hopf bifurcation, monitoring the orbit’s stability.

Now, consider varying g_{syn} and I_m for a fixed $I_s = 7.5$. As seen in Figure 10, the range of values of I_m for which oscillations may arise decreases as g_{syn} increases (both Hopf bifurcations move to lower values of I_m), and for small g_{syn} (i.e. weak coupling) the neurons are no longer synchronous, due to break-up as discussed. The conclusion is that, in order to obtain robust synchronous oscillations, we need moderate to large coupling (g_{syn}) and a not-too-heterogeneous population (I_s not too large). This is perhaps not surprising, but our main point here is to demonstrate how the computation of the effects of heterogeneity can easily be accelerated. We now consider more than one heterogeneous parameter.

5 Two heterogeneous parameters

Now, consider the case where both I_{app} and g_{Na} for each neuron are randomly (independently) distributed. We keep the uniform distribution for the I_{app} , choosing $I_m = 25$, $I_s = 7.5$ so that the I_{app} come from a uniform distribution on $[17.5, 32.5]$. We choose the g_{Na} from a normal distribution with a mean of 2.8, and standard deviation σ and set $g_{\text{syn}} = 0.3$. We keep 10 points in the μ direction and use the values of μ_i and w_i given in Equation 14 and the text immediately preceding it to perform integration in the μ direction. The quantity M refers to the number of different g_{Na} values chosen, and we thus simulate $10M$ appropriately coupled neurons.

The values of I_{app} and g_{Na} for the different neurons are selected based on the tensor product of the vectors formed from I_{app} and g_{Na} . Similarly, the weights in a sum of the form given by Equation 15 will be formed from a tensor product of the w_i associated with the I_{app} direction and those associated with the g_{Na} .

We initially choose $\sigma = 0.25$ and write $g_{\text{Na}} =$

$2.8 + \sigma\lambda$, where λ has the probability density function

$$q(\lambda) = \frac{1}{\sqrt{2\pi}} e^{-\lambda^2/2}, \quad (16)$$

i.e. λ is normally distributed. Then, as mentioned, the continuum variables V and h are written in the form $V(\mu, \lambda, t)$ and $h(\mu, \lambda, t)$, respectively, and the sum in Equation 3 becomes

$$\int_{-\infty}^{\infty} \int_{-1}^1 s(V(\mu, \lambda, t)) p(\mu) q(\lambda) d\mu d\lambda. \quad (17)$$

Keeping the Gauss-Legendre rule in the μ direction, this gives

$$\int_{-\infty}^{\infty} \sum_{i=1}^{10} w_i s(V(\mu_i, \lambda, t)) q(\lambda) d\lambda. \quad (18)$$

The simplest approach to this integral is the Monte Carlo method [30], where we simply randomly choose M values of λ from the unit normal distribution and calculate an approximation to the integral as the following:

$$\frac{1}{M} \sum_{j=1}^M \sum_{i=1}^{10} w_i s(V(\mu_i, \lambda_j, t)). \quad (19)$$

Here, the weights in the λ direction are all equal to $1/M$. An example of the μ_i and λ_j for $M = 15$ is shown in Figure 11 (top). Another approach is to transform the integral to one over $[0, 1]$ and use the composite midpoint rule on that new variable. Specifically, if we define

$$z = Q(\lambda) \equiv \int_{-\infty}^{\lambda} q(s) ds, \quad (20)$$

i.e. Q is the cumulative density function for λ , and then for a general function f , the integral

$$\int_{-\infty}^{\infty} f(\lambda) q(\lambda) d\lambda \quad (21)$$

can be written as

$$\int_0^1 f(Q^{-1}(z)) dz. \quad (22)$$

Thus, we define

$$\lambda_j = Q^{-1}\left(\frac{j}{M} - \frac{1}{2M}\right), \quad (23)$$

for $j = 1, \dots, M$ and use the approximation in Equation 19. An example of the μ_i and λ_j for $M = 15$ is shown in Figure 11 (middle). It is better still to use Gaussian quadrature (specifically, Gauss-Hermite quadrature) in the λ direction. We approximate the integral

$$\int_{-\infty}^{\infty} f(\lambda) q(\lambda) d\lambda \approx \sum_{j=1}^N v_j f(\lambda_j), \quad (24)$$

where λ_j is the j th root of H_N , the N th ‘probabilists’ Hermite polynomial’, and the weights v_j are given by

$$v_j = \frac{N!}{[N H_{N-1}(\lambda_j)]^2}. \quad (25)$$

(The first few probabilists’ - as opposed to physicists’ - Hermite polynomials are $H_0(x) = 1$, $H_1(x) = x$, $H_2(x) = x^2 - 1, \dots$) Thus, we approximate the integral in Equation 17 by the double sum:

$$\begin{aligned} & \int_{-\infty}^{\infty} \int_{-1}^1 s(V(\mu, \lambda, t)) p(\mu) q(\lambda) d\mu d\lambda \\ & \approx \sum_{j=1}^M \sum_{i=1}^{10} v_j w_i s(V(\mu_i, \lambda_j, t)). \end{aligned} \quad (26)$$

An example of the μ_i and λ_j for $M = 15$ is shown in Figure 11 (bottom).

The result of using these three different methods to allocate the g_{Na} (and thus, to select the reduced number of appropriately coupled neurons we simulate) is shown in Figure 12. This figure shows the error in the calculated period as M is varied. (The true period was calculated using Gauss-Hermite quadrature with a large M in the g_{Na} direction.)

We see that as expected, the Gauss-Hermite quadrature performs the best, with the error saturating between $M = 10$ and $M = 20$. (Recalling that we are using 10 points in the μ direction, this is consistent with the idea that roughly the same number of points should be used in each random direction.) Using the Monte Carlo method, i.e. randomly choosing, the g_{Na} gives convergence that scales as $M^{-1/2}$. Uniformly sampling the inverse cumulative distribution function gives an error that appears to scale as M^{-1} . This is at variance with the expected scaling of M^{-2} for the composite midpoint rule applied to a

function with a bounded second derivative, but the inverse CDF of a normal distribution (i.e. $Q^{-1}(z)$) does not have a bounded second derivative, and an error analysis of Equation 22 (not shown) predicts a scaling of M^{-1} , as observed.

6 Sparse grids

The process described in the previous section can obviously be generalised to more than two randomly, but independently, distributed parameters. The distribution of each parameter determines the type of quadrature which should be used in that direction, and the parameter values and weights are formed from tensor products of the underlying one-dimensional rules. However, the curse of dimensionality will restrict how many random parameters can be accurately sampled. If we use N points in each of D random dimensions, the number of neurons we need to simulate is N^D .

One way around this problem is to use sparse grids [33, 34], as introduced by Smolyak [35]. The basic idea is to use sparse tensor products, chosen in such a way as to have similar accuracy to the corresponding full tensor product, but with fewer grid points, and thus (in our case) fewer neurons to simulate. A general theory exists [33, 34], but to illustrate the idea, suppose we have two uncorrelated random parameters, each is distributed uniformly between -1 and 1 . A full tensor product for the Gauss-Legendre quadrature using 11 points in each direction is shown in Figure 13.

To form a two-dimensional sparse grid using the Gauss-Legendre quadrature, we first write the one-dimensional integration rule for integrating a function f as

$$\int_{-1}^1 f(x) dx \approx U^i(f) \equiv \sum_{j=1}^{N_i} w_j f(x_j), \quad (27)$$

where $i \in \mathbb{N}$; w_j are the weights, and x_j are the nodes. We form a nested family of such rules with index i where the correspondence between i and N_i is given in the following:

i.e. $N_i = 2^{i+1} - 1$. Then, the level L rule in two

spatial dimensions is

$$A(L, 2) = \sum_{|\mathbf{i}|=L} (U^{i_1} \otimes U^{i_2}) - \sum_{|\mathbf{i}|=L-1} (U^{i_1} \otimes U^{i_2}), \quad (28)$$

where $\mathbf{i} \in \mathbb{N}^2$ and $|\mathbf{i}| = i_1 + i_2$. The approximation of the integral of f over the domain $[-1, 1]^2$ is $A(L, 2)(f)$. So for example, the level 2 rule (in 2 spatial dimensions and using Gauss-Legendre quadrature) is

$$A(2, 2) = U^0 \otimes U^2 + U^1 \otimes U^1 + U^2 \otimes U^0 - (U^0 \otimes U^1 + U^1 \otimes U^0) \quad (29)$$

The grid for this rule is shown in Figure 14 (top), along with grids corresponding to several of its components.^a Figure 14 (bottom) shows the grid for rule $A(3, 2)$.

Rules such as these can be constructed for an arbitrary number of spatial dimensions, using a variety of quadrature rules (and possibly different rules in different dimensions). Their advantage becomes apparent as the dimension of the space to be integrated over (or in our case, the number of heterogeneous parameters) is increased. To illustrate this, we consider as an example the model given by Equations 1 and 2 with I_{app} uniformly spread between 17.5 and 32.5, the g_{Na} uniformly spread between 2.55 and 3.05, V_{syn} uniformly spread between -1 and 1 , and V_{Na} uniformly spread between 49 and 51, i.e. 4 independent random dimensions. A comparison of the error in calculating the period of collective oscillation using full and sparse grids is shown in Figure 15.

We see that for fixed N , the sparse grid calculation is approximately two orders of magnitude more accurate than the full grid - implying, in turn, that *the way* we select the reduced number of neurons we retain to simulate the full system is critical. This relative advantage is expected to increase as the number of distributed parameters increases. As an example of the growth in the number of grid points, a level 6 calculation in 10 dimensions uses fewer than one million points, and the resulting system can be easily simulated on a desktop PC. (Note that the grid points and weights are calculated before the numerical integration starts, so the computational cost in producing data like that shown in Figure 15 is almost entirely due to numerical integration of the

ODEs, which is proportional to the number of grid points, i.e. neurons, used.)

7 Discussion

In this paper, we have presented and demonstrated the use of a computationally efficient method for systematically investigating the effects of heterogeneity in the parameters of a coupled network of neural oscillators. The method constitutes a model reduction approach: By only considering oscillators with parameter values given by roots of families of orthogonal polynomials (Legendre, Hermite, ...), we can use Gaussian quadrature to accurately evaluate the term coupling the oscillators, which can be thought of as the discretisation of an integral over the heterogeneous dimension(s).

Effectively, we are simulating the behaviour of an infinite number of oscillators by only simulating a small number of judiciously selected ones, modifying appropriately the way they are coupled. When the oscillators are synchronised, or at a fixed point, the function to be integrated is a smooth function of the heterogeneous parameter(s), and thus, convergence is very rapid. The technique is general (although subject to the restriction immediately above) and can be used when there is more than one heterogeneous parameter, via full or sparse tensor products in parameter space. For a given level of accuracy, we are simulating far fewer neurons than might naively be expected. The emphasis here has been on computational efficiency rather than a detailed investigation of parameter dependence.

The model we considered involved coupling only through the mean of a function, s , of the variable V_i which, in the limit $N \rightarrow \infty$, can be thought of as an integral or, more generally, as a functional of $V(\mu)$. Thus, the techniques demonstrated here could also be applied to networks coupled through terms which, in the continuum limit, are integrals or functions of integrals. A simple example is diffusive coupling [3]; another possibility is coupling which is dependent upon the correlation between some or all of the variables. As mentioned, the technique will break down once the oscillators become desynchronised, as the dependence of state on parameter(s) will no longer be smooth. However, if the oscillators form several clusters [14,36], it may be possible to apply the ideas

presented here to each cluster, as the dependence of state on parameter(s) *within* each cluster should still be smooth. Ideally, this reparametrisation would be done adaptively as clusters form, in the same way that algorithms for numerical integration adapt as the solution varies [30]. Alternatively, if a single oscillator ‘breaks away’ [27], the methods presented here could be used on the remaining synchronous oscillators, with the variables describing the state of the rogue oscillator also fully resolved. More generally, there are systems in which it is not necessarily the *state* of an oscillator that is a smooth function of the heterogeneous parameter, but the *parameters describing the distribution of states* [37,38], and the ideas presented here could also be useful in this case.

The primary study with which we should compare our results is that of Rubin and Terman [14]. They considered essentially the same model as Equations 1 and 2 but with heterogeneity only in the I_{app} and, taking the continuum limit, referred to the curve in (V, h) space describing the state of the neurons at any instant in time as a ‘snake’. By making various assumptions, such as an infinite separation of time scales between the dynamics of the V_i and the h_i , and that the dynamics of the h_i in both the active and quiescent phases is linear, they derived an expression for the snake at one point in its periodic orbit and showed that such a snake is unique and stable. They also estimated the parameter values at which the snake ‘breaks’ and some oscillators lose synchrony. In contrast with their mainly analytical study, ours is mostly numerical and thus does not rely on any of the assumptions just mentioned. Using the techniques presented here, we were able to go beyond the work of Rubin and Terman, exploring parameter space. Our approach can be thought of as a particular parametrisation of this snake, which takes into account the probability density of the heterogeneity parameter(s); we also showed a systematic way of extending this one-dimensional snake to two and higher dimensions.

Another paper which uses some of the same ideas as presented here is that of Laing and Kevrekidis [3]. There, the authors considered a finite network of coupled oscillators and used a polynomial chaos expansion of the same form as Equation 12. However, instead of integrating the equations for the polynomial chaos coefficients directly, they used projective integration [39] to do so, in an ‘equation-free’ ap-

proach [40] in which the equations satisfied by the polynomial chaos coefficients are never actually derived. They also chose the heterogeneous parameter values randomly from a prescribed distribution and averaged over realisations of this process in order to obtain ‘typical’ results. Similar ideas had been explored earlier by Moon et al. [27], who considered a heterogeneous network of phase oscillators.

Assisi et al. [22] considered a heterogeneous network of coupled neural oscillators, deriving equations of similar functional form to Equations 9 and 11. Their approach was to expand the variables in a way similar to Equation 12 but using a small number of arbitrarily chosen ‘modes’ rather than orthogonal polynomials. Their choice of modes, along with the fact that their neural model consisted of ODEs with polynomial right hand sides, allowed them to analytically derive the ODEs satisfied by the coefficients of the modes. This approach allowed them to qualitatively reproduce some of the behaviour of the network such as the formation of two clusters of oscillators. However, in the general case modes should be chosen as orthogonal polynomials, the specific forms of which are determined by the distribution of the heterogeneous parameter(s) [25, 26].

The network we considered was all-to-all coupled, and the techniques presented should be applicable to other similar systems. The only requirement is that the relationship between the heterogeneity parameter(s) and the state of the system (possibly after transients) be smooth (or possibly piecewise smooth). An interesting extension is the case when the network under consideration is not all-to-all coupled. Then, the effects of degree distribution may affect the dynamics of individual oscillators [38, 41, 42], and if we have a way of parametrising this type of heterogeneity, it might be possible to apply the ideas presented here to such networks. Degree distribution is a discrete variable, and corresponding families of orthogonal polynomials exist for a variety of discrete random variables [25, 26].

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The works of CRL and BS were supported by the Marsden Fund Council from government funding, administered by the Royal Society of New Zealand. The works of IGK and YZ were supported by the AFOSR and the US DOE (DE-SC0005176 and DE-SC00029097).

Endnotes

^aThese sparse grids were computed using software from <http://people.sc.fsu.edu/~jburkardt/>

References

1. Baesens C, Guckenheimer J, Kim S, MacKay RS: **Three coupled oscillators: mode-locking, global bifurcations and toroidal chaos.** *Physica D: Nonlinear Phenomena* 1991, **49**(3):387–475.
2. Matthews PC, Mirollo RE, Strogatz SH: **Dynamics of a large system of coupled nonlinear oscillators.** *Physica D: Nonlinear Phenomena* 1991, **52**(2-3):293–331.
3. Laing CR, Kevrekidis IG: **Periodically-forced finite networks of heterogeneous globally-coupled oscillators: a low-dimensional approach.** *Physica D: Nonlinear Phenomena* 2008, **237**(2):207 – 215.
4. Martens EA, Laing CR, Strogatz SH: **Solvable model of spiral wave chimeras.** *Phys. Rev. Lett.* 2010, **104**(4):044101.
5. Ermentrout B, Pascal M, Gutkin B: **The effects of spike frequency adaptation and negative feedback on the synchronization of neural oscillators.** *Neural Computation* 2001, **13**(6):1285–1310.
6. Pikovsky A, Rosenblum M, Kurths J: *Synchronization.* Cambridge: Cambridge University Press; 2001.
7. Acebrón JA, Bonilla LL, Pérez Vicente CJ, Ritort F, Spigler R: **The Kuramoto model: a simple paradigm for synchronization phenomena.** *Rev. Mod. Phys.* 2005, **77**:137–185.
8. Hassard B: **Bifurcation of periodic solutions of the Hodgkin-Huxley model for the squid giant axon.** *Journal of Theoretical Biology* 1978, **71**(3):401–420.
9. Ermentrout GB, Terman D: *Mathematical Foundations of Neuroscience.* New York: Springer Verlag; 2010.
10. Coombes S, Bressloff PC: *Bursting: The Genesis of Rhythm in the Nervous System.* Singapore: World Scientific Pub Co Inc; 2005.

11. Izhikevich EM: **Neural excitability, spiking and bursting.** *International Journal of Bifurcation and Chaos* 2000, **10**(6):1171–1266.
12. Coombes S: **Neuronal networks with gap junctions: a study of piecewise linear planar neuron models.** *SIAM Journal on Applied Dynamical Systems* 2008, **7**:1101.
13. Butera RJ, Rinzel J, Smith JC: **Models of respiratory rhythm generation in the pre-Bötzinger complex. II. Populations of coupled pacemaker neurons.** *Journal of Neurophysiology* 1999, **82**:398.
14. Rubin J, Terman D: **Synchronized activity and loss of synchrony among heterogeneous conditional oscillators.** *SIAM Journal on Applied Dynamical Systems* 2002, **1**:146–174.
15. Golomb D, Rinzel J: **Dynamics of globally coupled inhibitory neurons with heterogeneity.** *Phys. Rev. E* 1993, **48**(6):4810–4814.
16. Golubitsky M, Stewart I, Buono PL, Collins JJ: **Symmetry in locomotor central pattern generators and animal gaits.** *Nature* 1999, **401**(6754):693–695.
17. Ashwin P, Swift J: **The dynamics of n weakly coupled identical oscillators.** *Journal of Nonlinear Science* 1992, **2**:69–108.
18. Gosak M: **Cellular diversity promotes intercellular Ca^{2+} wave propagation.** *Biophysical Chemistry* 2009, **139**:53–56.
19. Pérez T, Mirasso CR, Toral R, Gunton JD: **The constructive role of diversity in the global response of coupled neuron systems.** *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2010, **368**(1933):5619–5632.
20. Tessone CJ, Mirasso CR, Toral R, Gunton JD: **Diversity-induced resonance.** *Phys. Rev. Lett.* 2006, **97**:194101.
21. Purvis LK, Smith JC, Koizumi H, Butera RJ: **Intrinsic bursters increase the robustness of rhythm generation in an excitatory network.** *Journal of Neurophysiology* 2007, **97**(2):1515–1526.
22. Assisi CG, Jirsa VK, Kelso JAS: **Synchrony and clustering in heterogeneous networks with global coupling and parameter dispersion.** *Phys. Rev. Lett.* 2005, **94**:018106.
23. White JA, Chow CC, Ritt J, Soto-Treviño C, Kopell N: **Synchronization and oscillatory dynamics in heterogeneous, mutually inhibited neurons.** *Journal of Computational Neuroscience* 1998, **5**:5–16.
24. Butera RJ, Rinzel J, Smith JC: **Models of respiratory rhythm generation in the pre-Bötzinger complex. I. Bursting pacemaker neurons.** *Journal of Neurophysiology* 1999, **82**:382.
25. Xiu D: **Fast numerical methods for stochastic computations: a review.** *Communications in Computational Physics* 2009, **5**(2-4):242–272.
26. Xiu D, Karniadakis GE: **Modeling uncertainty in flow simulations via generalized polynomial chaos.** *Journal of Computational Physics* 2003, **187**:137–167.
27. Moon SJ, Ghanem R, Kevrekidis IG: **Coarse graining the dynamics of coupled oscillators.** *Phys. Rev. Lett.* 2006, **96**:144101.
28. Rubin JE: **Bursting induced by excitatory synaptic coupling in nonidentical conditional relaxation oscillators or square-wave bursters.** *Phys. Rev. E* 2006, **74**(2):021917.
29. Dunmyre JR, Rubin JE: **Optimal intrinsic dynamics for bursting in a three-cell network.** *SIAM Journal on Applied Dynamical Systems* 2010, **9**:154–187.
30. Quarteroni A, Sacco R, Saleri F: *Numerical Mathematics, Volume 37.* New York: Springer Verlag; 2007.
31. Trefethen LN: **Is Gauss quadrature better than Clenshaw-Curtis?** *SIAM Review* 2008, **50**:67.
32. Moehlis J: **Canards in a surface oxidation reaction.** *Journal of Nonlinear Science* 2002, **12**(4):319–345.
33. Gerstner T, Griebel M: **Numerical integration using sparse grids.** *Numerical Algorithms* 1998, **18**(3):209–232.
34. Barthelmann V, Novak E, Ritter K: **High dimensional polynomial interpolation on sparse grids.** *Advances in Computational Mathematics* 2000, **12**:273–288.
35. Smolyak SA: **Quadrature and interpolation formulas for tensor products of certain classes of functions.** *Dokl. Akad. Nauk SSSR* 1963, **4**:240–243.
36. Somers D, Kopell N: **Waves and synchrony in networks of oscillators of relaxation and non-relaxation type.** *Physica D: Nonlinear Phenomena* 1995, **89**(1-2):169–183.
37. Abrams DM, Strogatz SH: **Chimera states in a ring of nonlocally coupled oscillators.** *Int. J. Bifur. Chaos* 2006, **16**:21–37.
38. Ko TW, Ermentrout GB: **Partially locked states in coupled oscillators due to inhomogeneous coupling.** *Phys. Rev. E* 2008, **78**:016203.

39. Kevrekidis IG, Gear CW, Hyman JM, Kevrekidis PG, Runborg O, Theodoropoulos C: **Equation-free, coarse-grained multiscale computation: enabling macroscopic simulators to perform system-level analysis.** *Communications in Mathematical Sciences* 2003, **1**(4):715–762.
40. Xiu D, Kevrekidis IG, Ghanem R: **An equation-free, multiscale approach to uncertainty quantification.** *Computing in Science & Engineering* 2005, **7**(3):16–23.
41. Rajendran K, Kevrekidis IG: **Coarse graining the dynamics of heterogeneous oscillators in networks with spectral gaps.** *Phys. Rev. E* 2011, **84**:036708.
42. Tsoumanis AC, Siettos CI, Bafas GV, Kevrekidis IG: **Computations in social networks: from agent-based modeling to coarse-grained stability and bifurcation analysis.** *International Journal of Bifurcation and Chaos* 2010, **20**(11):3673–3688.

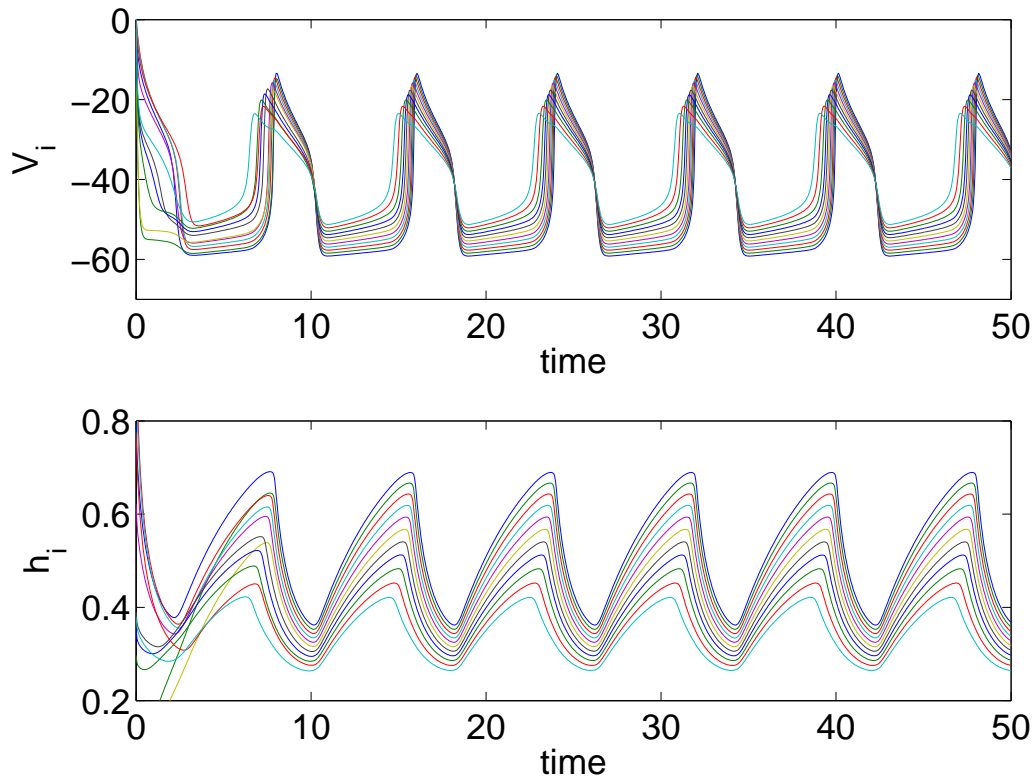


Figure 1: **Solutions of Equations 1 and 2.** These are the solutions when the I_{app}^i values are uniformly sampled from a uniform distribution on $[10, 25]$. Top: V_i as functions of time. Bottom: h_i as functions of time. $N = 101$. Different line colours correspond to different neurons (only every 10th neuron is shown).

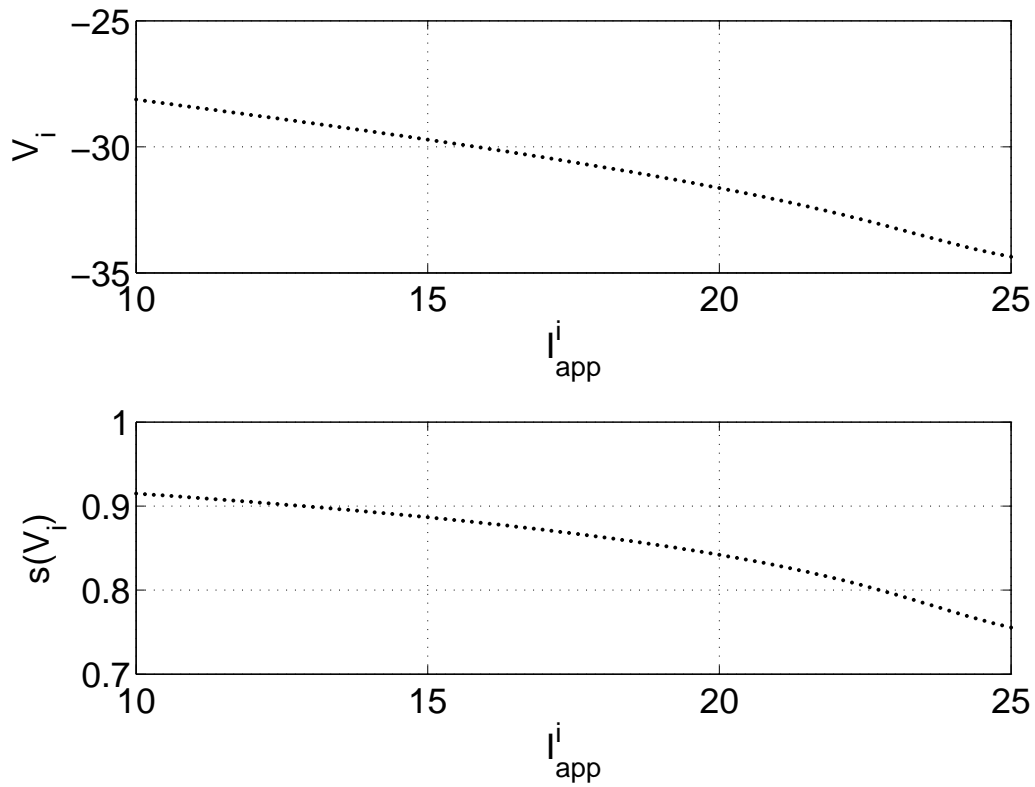


Figure 2: **Solutions of Equations 1 and 2 at one instant in time.** V_i (top) and $s(V_i)$ (bottom) as functions of I_{app}^i , $N = 101$. This shows a state where all neurons are active (see Figure 1). If the network was switching from active to quiescent or vice versa, there would be a steep ‘front’ where the V_i changed rapidly with i although they would still form a continuous curve.

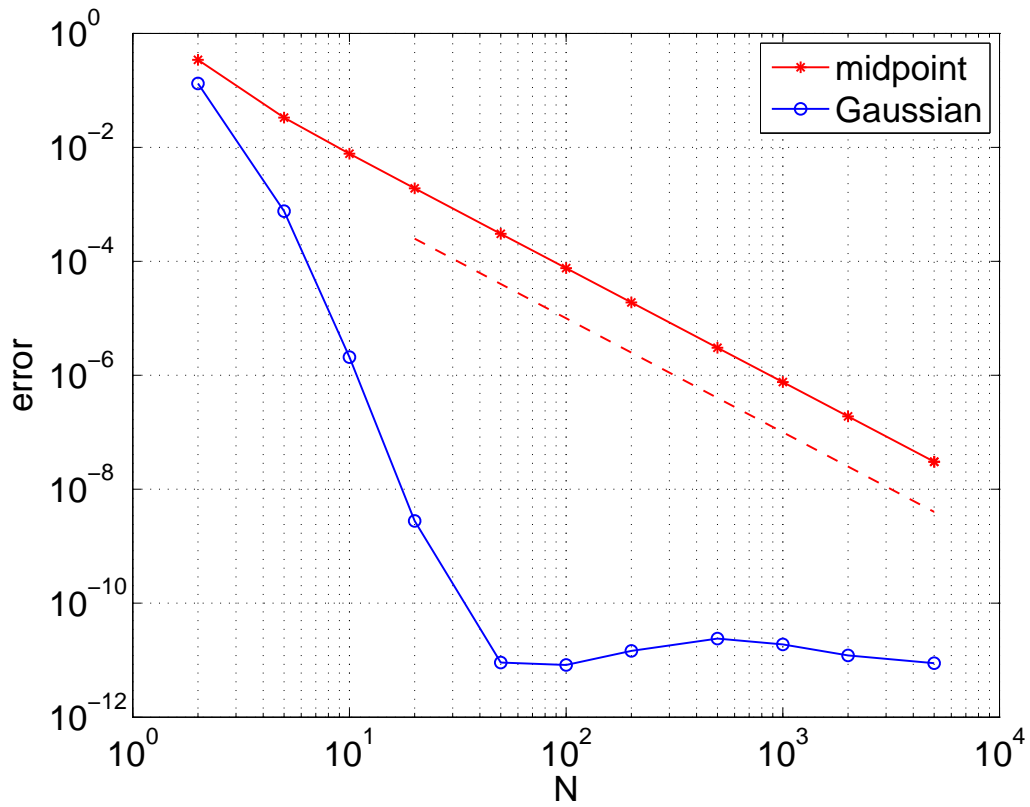


Figure 3: **Error in the calculated period of the synchronised oscillators.** Error in the calculated period of the synchronised oscillators as a function of the number of neurons simulated (N) for the midpoint rule (red stars) and Gaussian quadrature (blue circles). Also shown (dashed) is a line corresponding to error scaling as N^{-2} , to guide the eye.

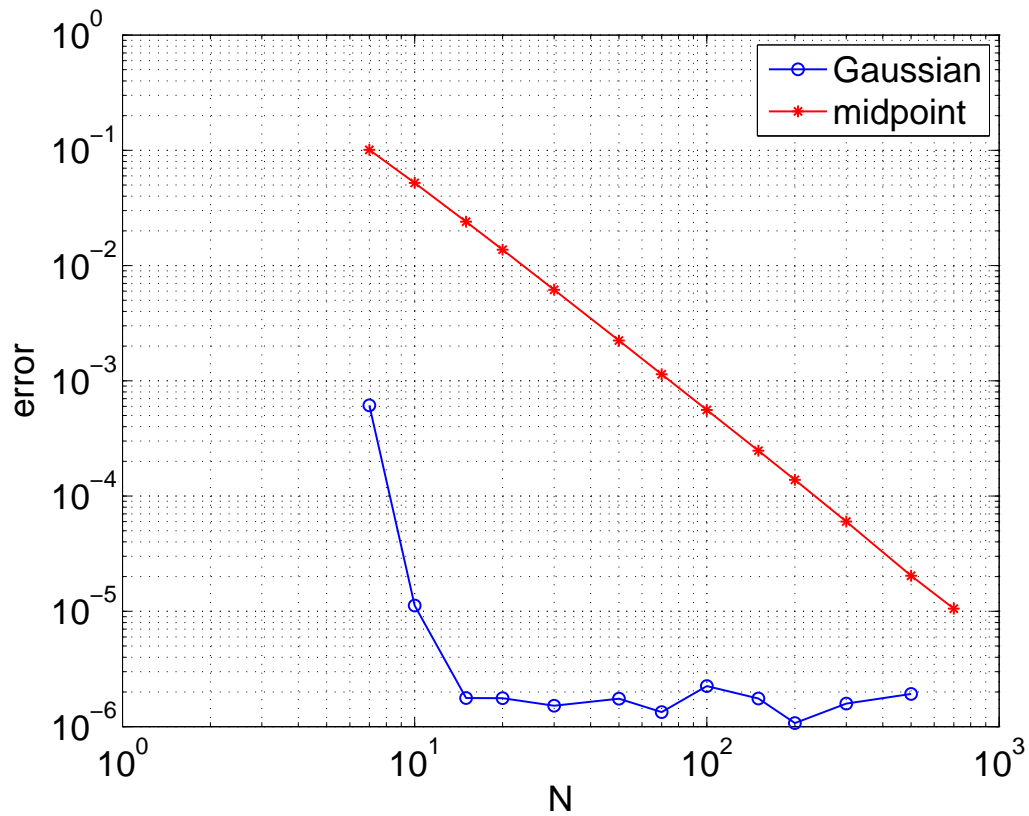


Figure 4: **Error in calculation of the value of I_m where upper Hopf bifurcation occurs.** Error in the calculation of the value of I_m at which the upper Hopf bifurcation occurs using the midpoint rule (red stars) and Gaussian quadrature (blue circles). Other parameters: $g_{\text{syn}} = 0.3, I_s = 7.5$. The midpoint rule error decays as $1/N^2$. For $10 < N$, the error using Gaussian quadrature is dominated by the precision with which the Hopf bifurcation can be located, hence the plateau.

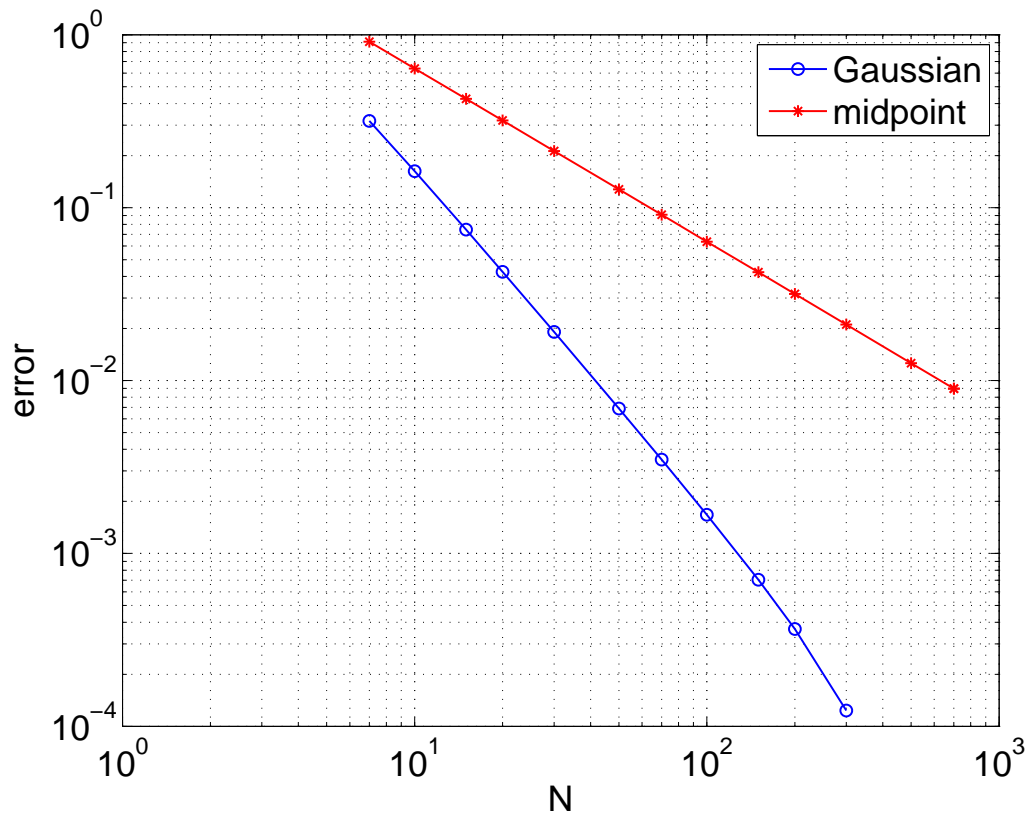


Figure 5: **Error in calculation of the value of I_m where lower Hopf bifurcation occurs.** Error in calculation of the value of I_m at which the lower Hopf bifurcation occurs using the midpoint rule (red stars) and Gaussian quadrature (blue circles). The error for the midpoint rule appears to decay as $1/N$. Other parameters: $g_{\text{syn}} = 0.3, I_s = 7.5$.

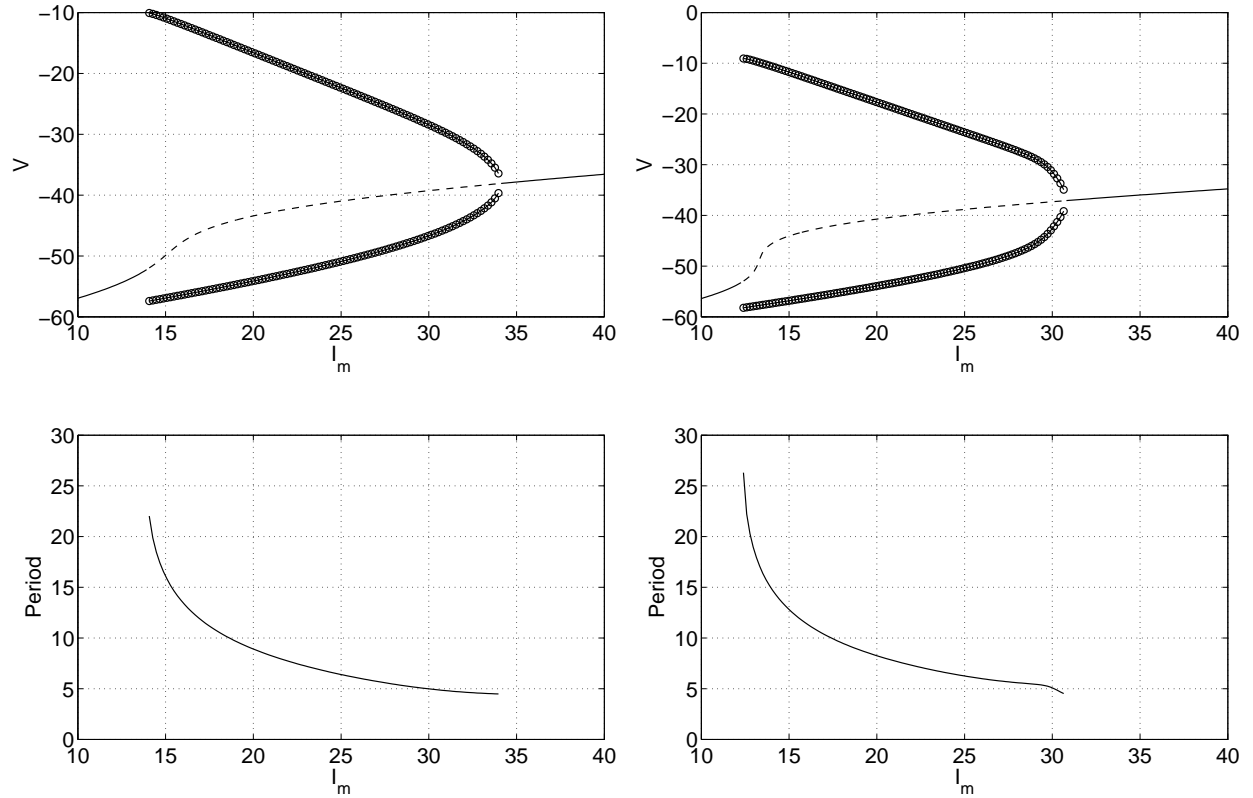


Figure 6: **The bifurcation behaviour, V as functions of I_m and period of the stable periodic orbit.** Left: the bifurcation behaviour of a single uncoupled neuron ($N = 1, g_{\text{syn}} = 0$). Top left: voltage V at a fixed point (solid, stable; dashed, unstable) and the maximum and minimum of V over one period of oscillation (circles), as a function of I_m . Bottom left: period of the stable periodic orbit for a single uncoupled neuron. The apparent discontinuity in the periodic orbit towards low I_m is because of the canard nature of the oscillations (mentioned in the text). Right: the bifurcation behaviour of a single self-coupled neuron ($N = 1, g_{\text{syn}} = 0.3$). Top right: voltage V at a fixed point (solid stable, dashed unstable) and the maximum and minimum of V over one period of oscillation (circles), as a function of I_m . Bottom right: period of the stable periodic orbit for a single self-coupled neuron.

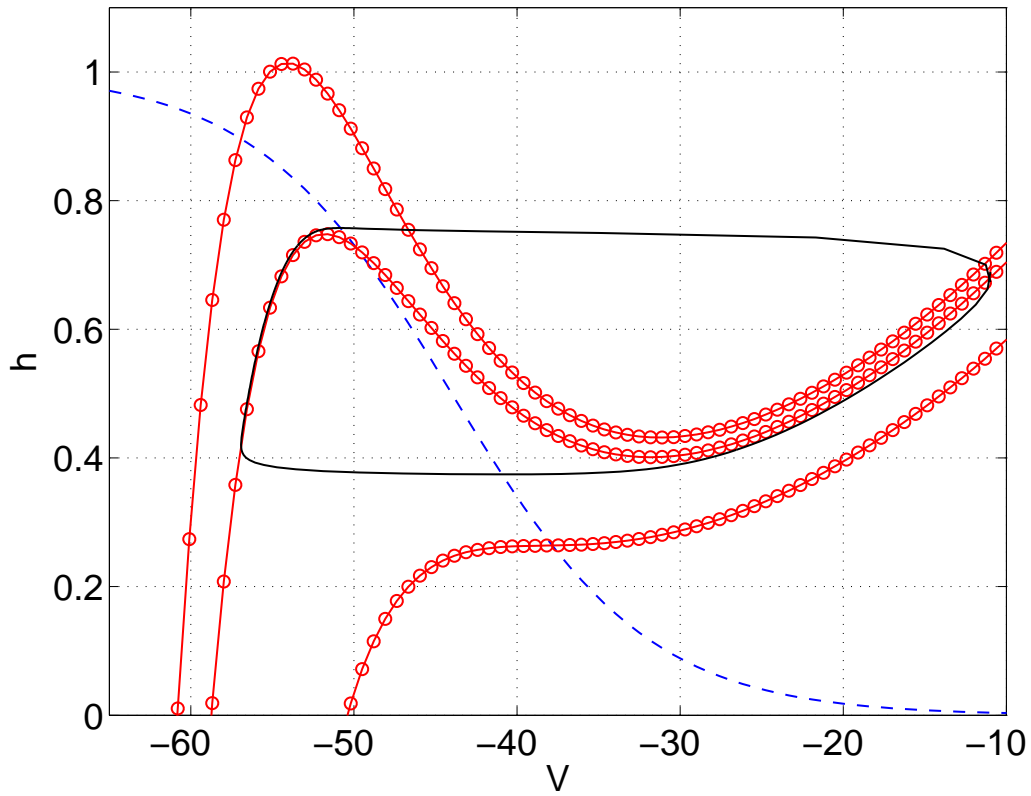


Figure 7: **The phase plane for a single uncoupled neuron.** h -nullcline (dashed, on which $dh/dt = 0$) and the V -nullclines (circles, on which $dV/dt = 0$) for $I_m = 10, 15$ and 35 (top to bottom). Also shown (solid) is the stable periodic orbit that exists when $I_m = 15$.

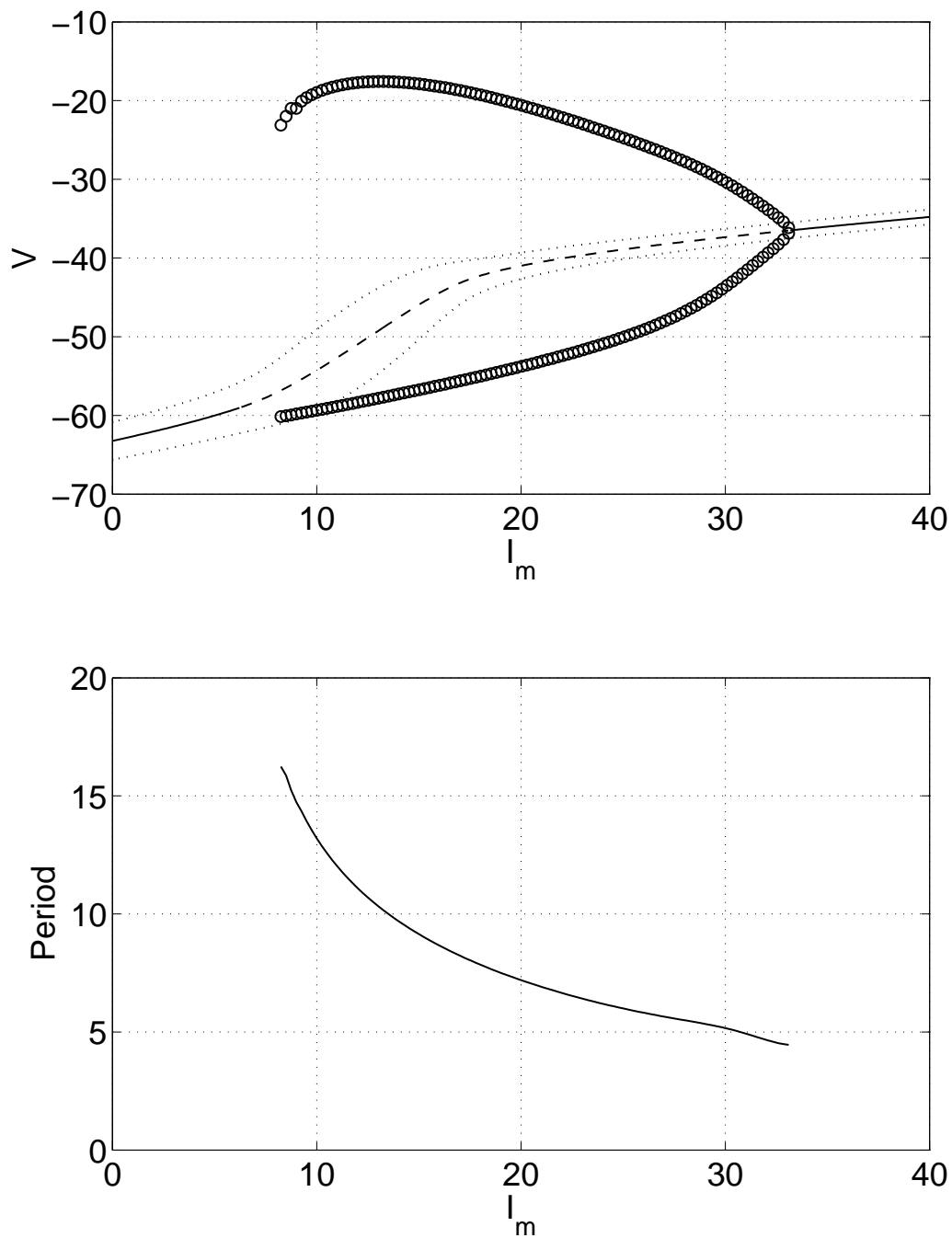


Figure 8: **The bifurcation behaviour of a heterogeneous population.** Top: mean voltage at a fixed point (solid stable, dashed unstable), mean \pm one standard deviation (dotted), and the maximum and minimum of the mean of V over one period of oscillation (circles), as a function of I_m . Bottom: period of the stable periodic orbit. $N = 10, g_{\text{syn}} = 0.3, I_s = 7.5$.

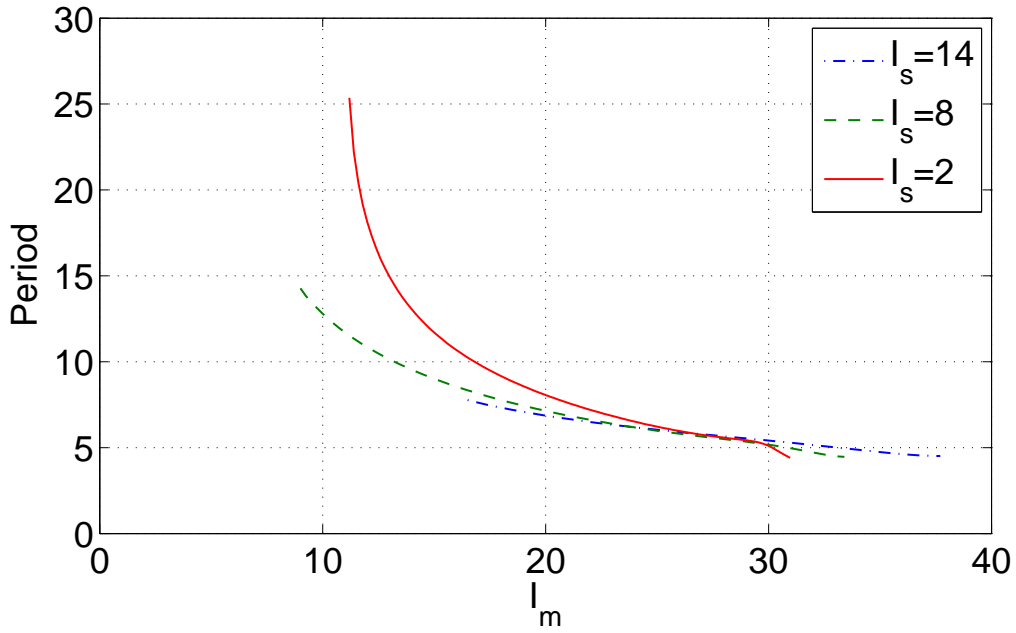
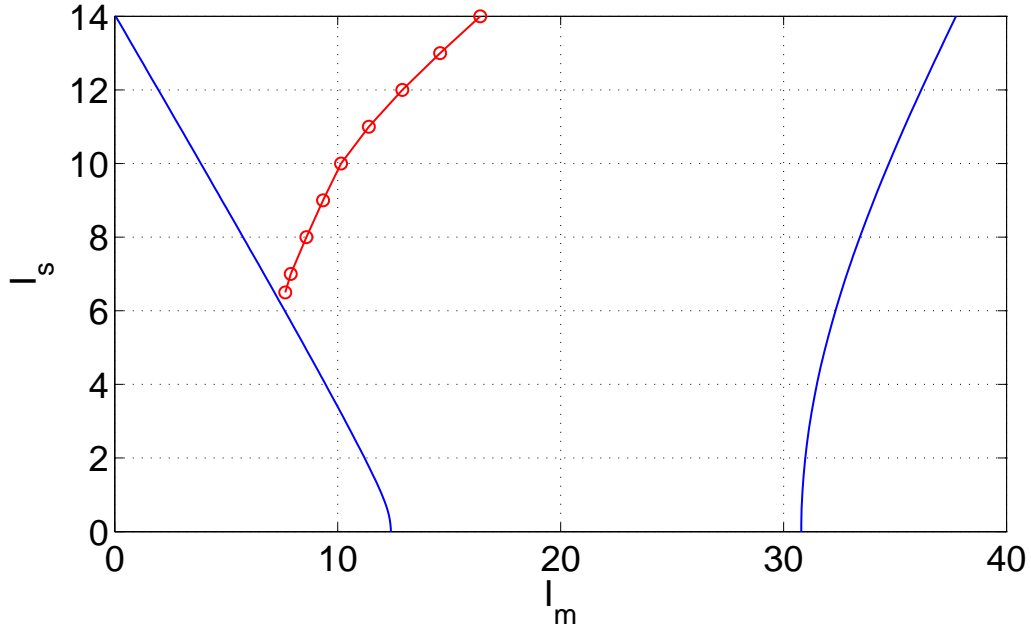


Figure 9: **Hopf bifurcation curves and period of the stable periodic orbit for three different values of I_s .** Top: Hopf bifurcation curves (solid) and the curve on which the periodic orbit created in the rightmost Hopf bifurcation loses stability (circles, found from direct simulation). Bottom: period of the stable periodic orbit for three different values of I_s , the spread of the heterogeneity. For $I_s = 8$ and 14 , the curves are terminated at low I_m when the periodic orbit loses stability to a more complex oscillation. $g_{\text{syn}} = 0.3$, $N = 10$.

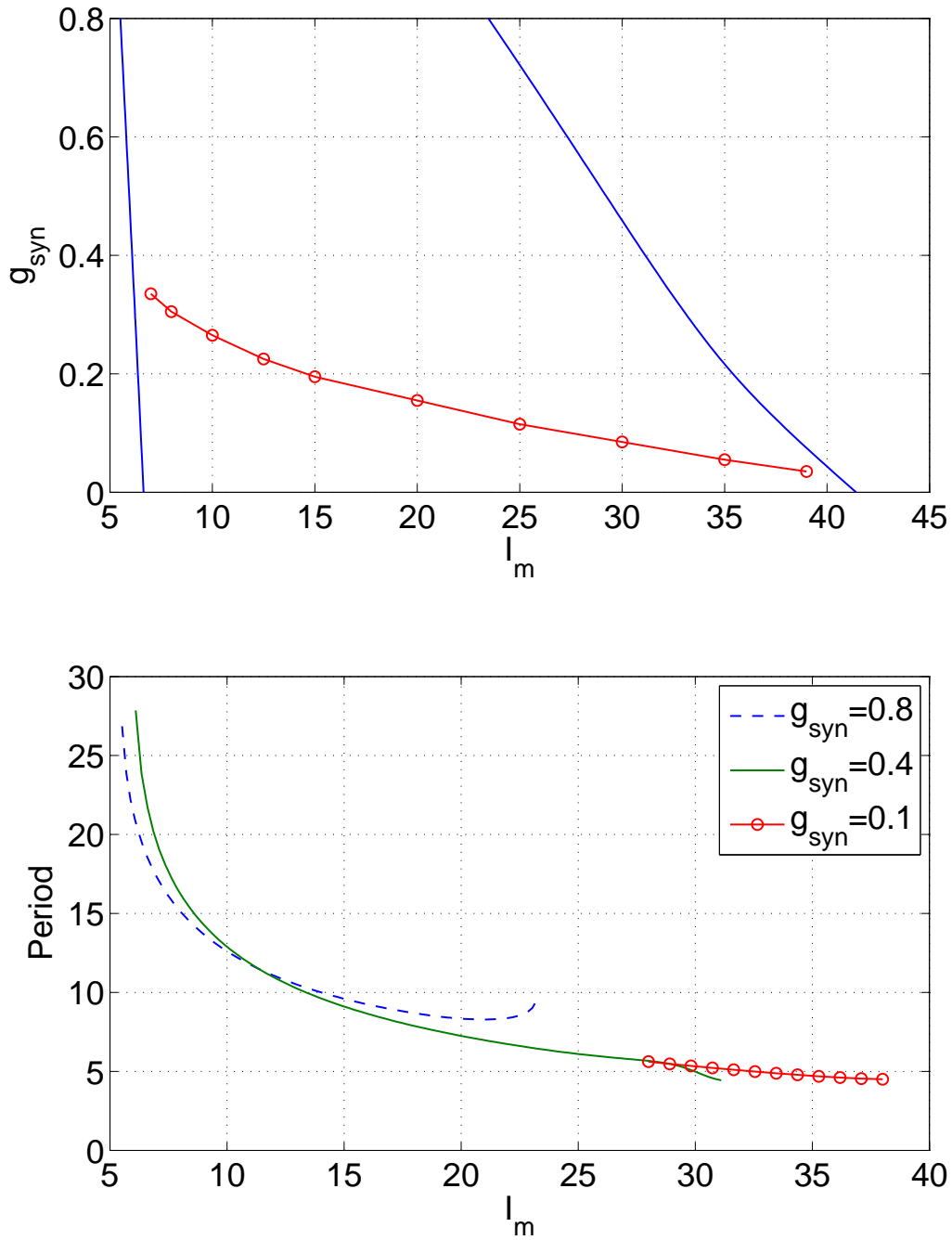


Figure 10: **Hopf bifurcation curves and period of the stable periodic orbit for three different values of g_{syn} .** Top: Hopf bifurcation curves (solid) and the curve on which the periodic orbit created in the rightmost Hopf bifurcation loses stability (circles, obtained by direct simulation). Synchronous oscillations occur only *above* the curve shown with red circles. Bottom: period of the stable periodic orbit for three different values of g_{syn} . The curve for $g_{\text{syn}} = 0.1$ is terminated at low I_m when the periodic orbit loses stability to a more complex oscillation. $I_s = 7.5$, $N = 10$.

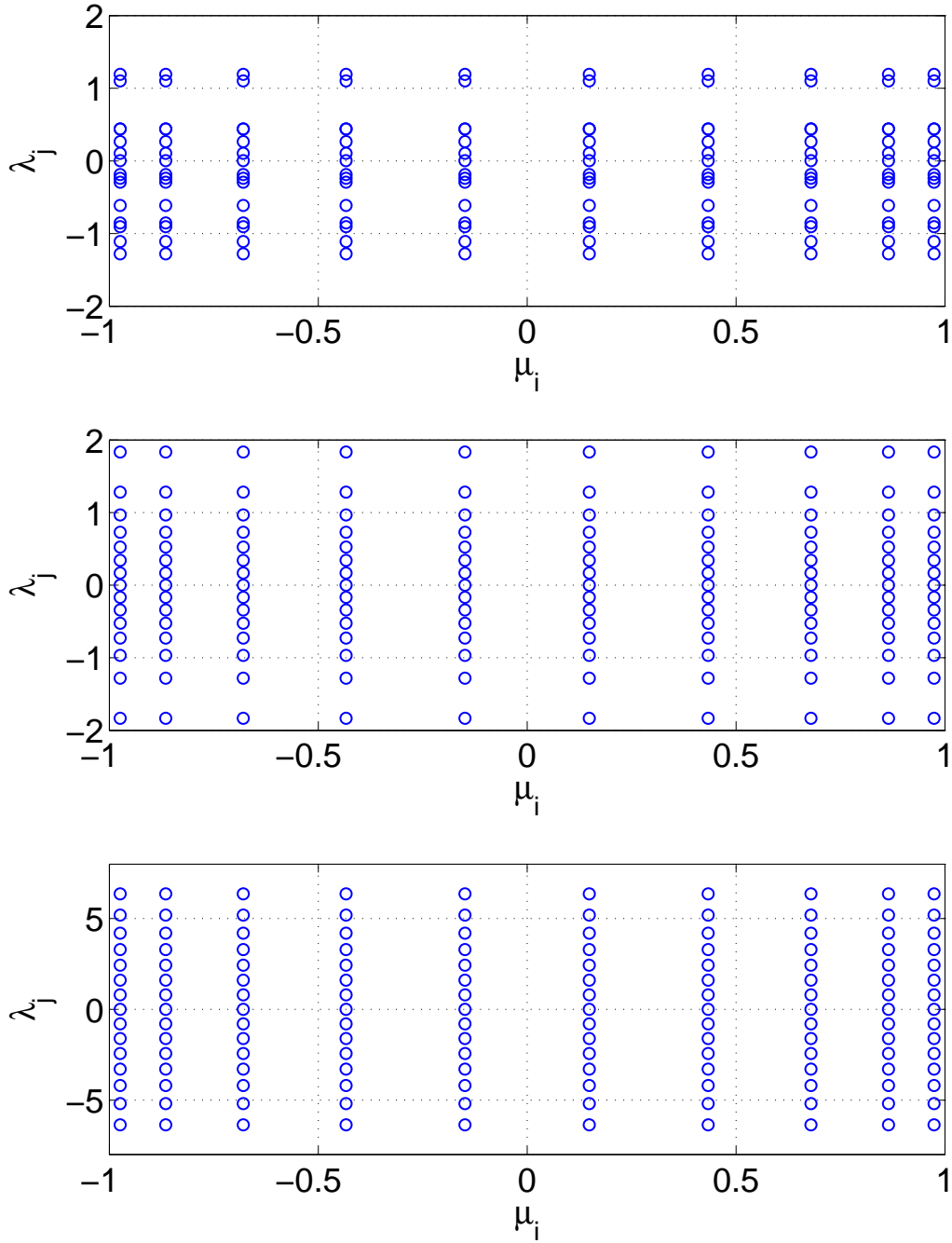


Figure 11: **Examples of the heterogeneity grid values of μ_i and λ_j for $M = 15$.** Top: the λ_j values are randomly chosen from a unit normal distribution. Middle: the λ_j values are chosen by uniformly sampling the inverse cumulative distribution function of a unit normal distribution. Bottom: the λ_j values are the roots of H_{15} , the 15th Hermite polynomial. In all cases, the μ_i are roots of P_{10} , the 10th Legendre polynomial.

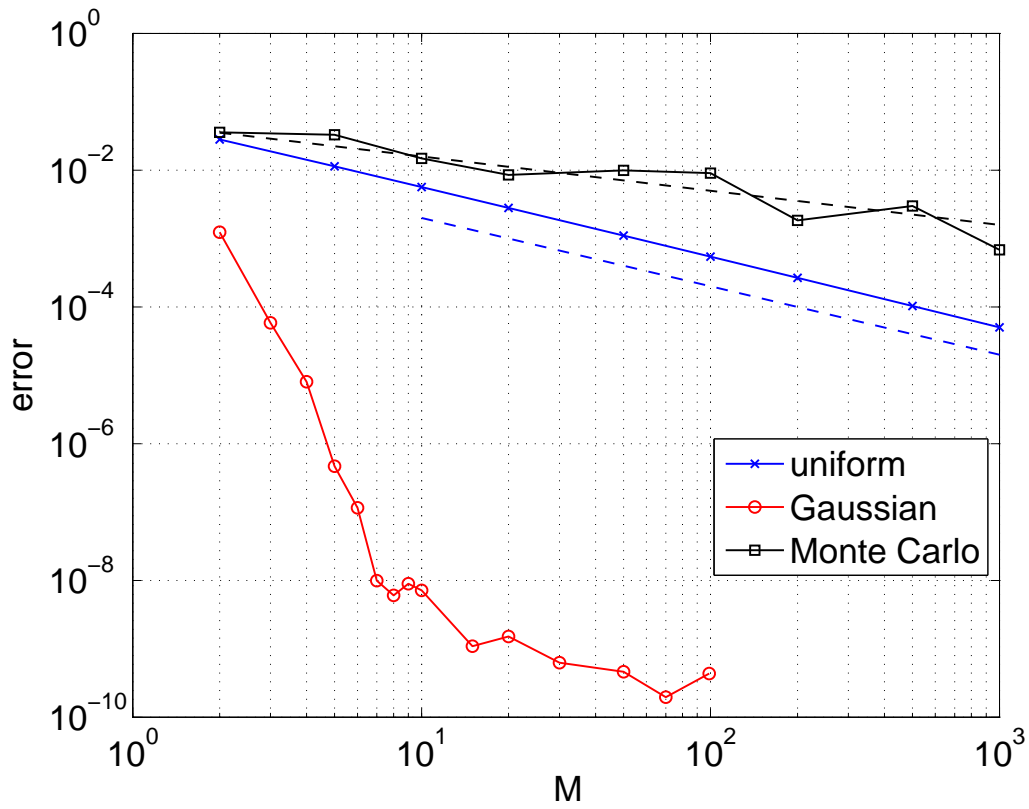


Figure 12: **Error in the calculated period using three different methods (see text) for $\sigma = 0.25$.** The dashed lines, drawn to guide the eye, have slopes $-1/2$ (upper) and -1 (lower). For the Monte Carlo simulations, the average of 10 calculations for each M is shown.

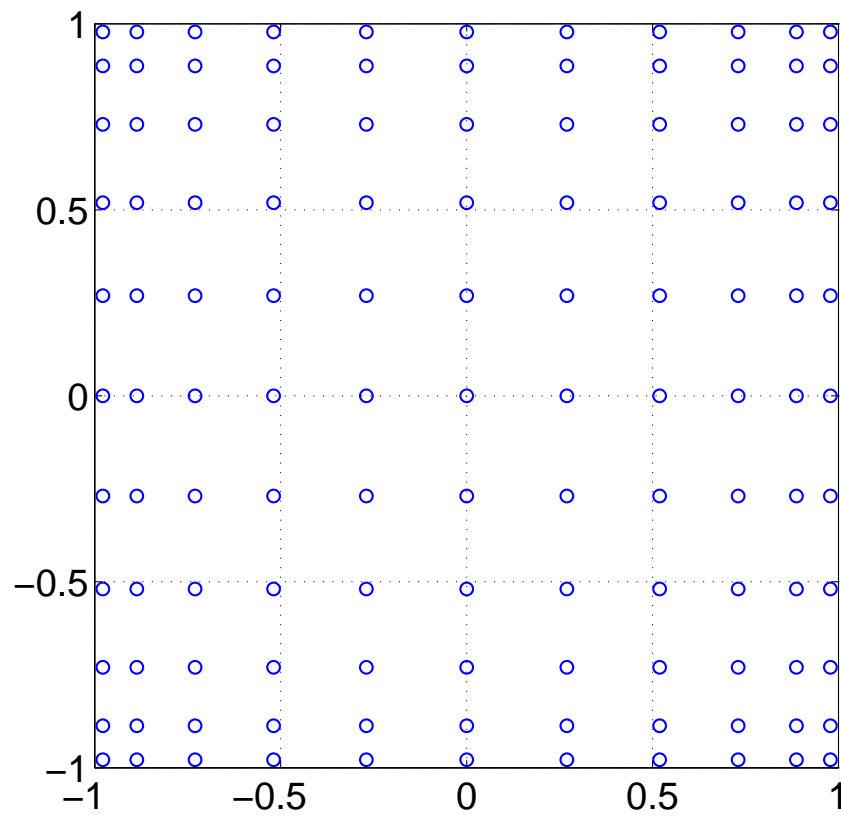


Figure 13: **Full tensor product using 11 points in each direction (121 points in total).** The points are the roots of P_{11} , the 11th Legendre polynomial.

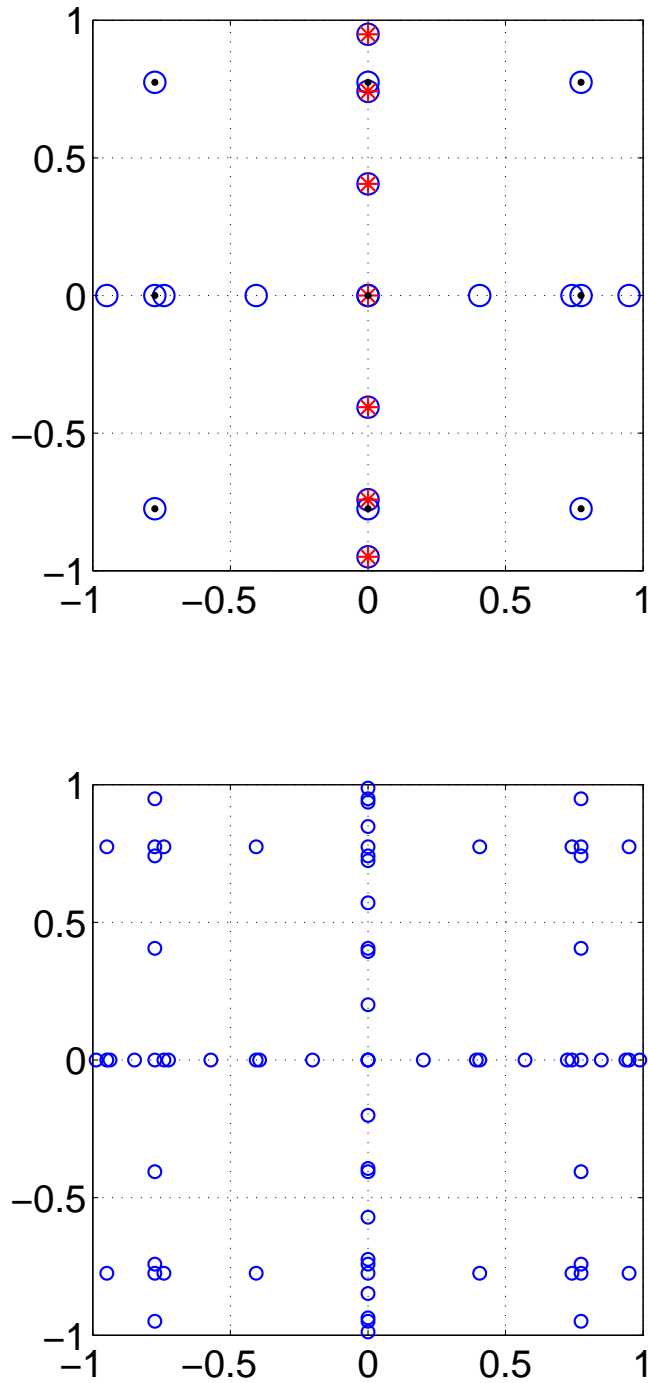


Figure 14: **Grids for rules A(2,2) and A(3,2).** Top: blue circles: the grid for rule $A(2,2)$ (i.e. level 2 in 2 spatial dimensions) using Gauss-Legendre quadrature. Red crosses: grid corresponding to $U^0 \otimes U^2$ (one point horizontally, 7 vertically). Black dots: grid corresponding to $U^1 \otimes U^1$ (3 points both horizontally and vertically). The three black dots on the y -axis correspond to $U^0 \otimes U^1$, while the three black dots on the x -axis correspond to $U^1 \otimes U^0$. Bottom: the grid for rule $A(3,2)$ (i.e. level 3 in 2 spatial dimensions). Rule $A(2,2)$ has 21 grid points, and rule $A(3,2)$ has 73.

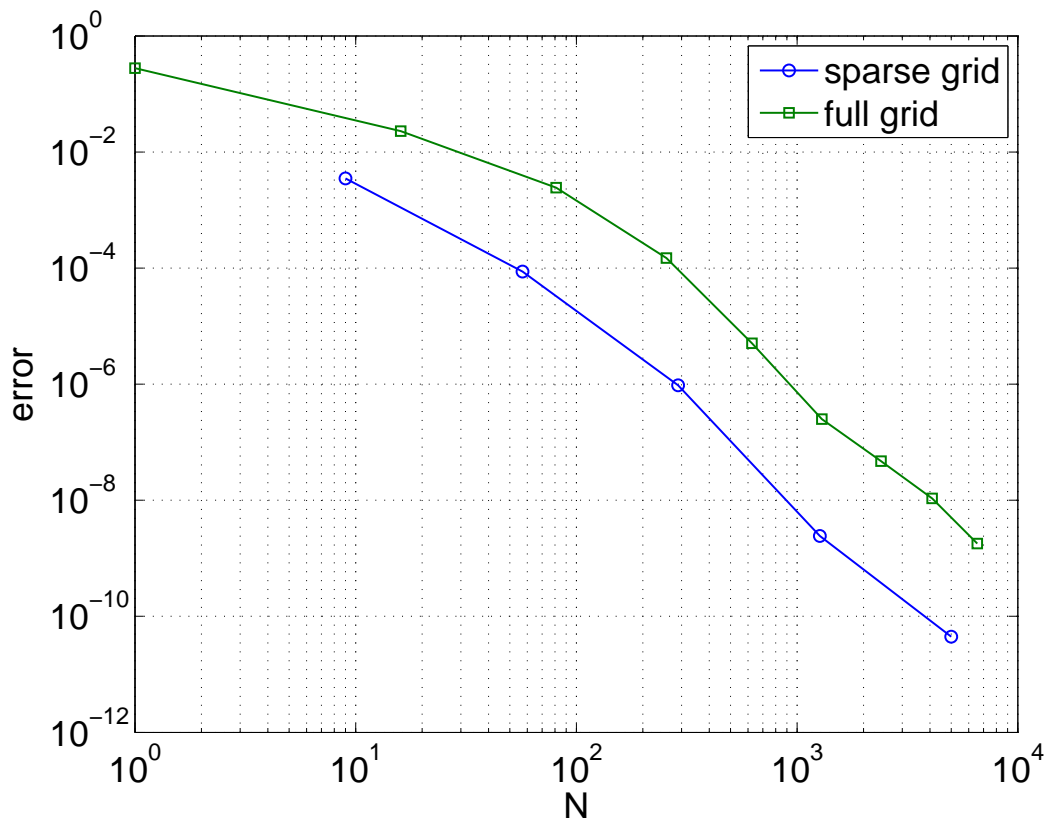


Figure 15: **Error in calculation of period.** Four distinct parameters are simultaneously heterogeneous (independently of one another) and we consider both full and sparse grids. See text for details. N is the number of neurons simulated.