Gradient-particle solutions of Fokker-Planck equations for noisy delay bifurcations

R. Kuske
School of Mathematics
University of Minnesota

February 12, 1999

Abstract

A gradient particle method is used to compute probability densities for processes with delay bifurcations that are sensitive to very small noise. The method is particularly useful for computing the probability density in the transition region, where asymptotic approximations may not be valid. For a one dimensional steady bifurcation problem and a two-dimensional FitzHugh-Nagumo model we solve the Fokker-Planck equation, and compare the results with direct simulations and asymptotic approximations.

1 Introduction

Certain dynamical systems are very sensitive to noise, such as models of chemical kinetics [1]-[2], neuron firing [2]-[3], chaotic wave mode interactions [4], and lasers [5]. The deterministic dynamics of these systems, described by systems of ordinary differential equations, can change substantially when very small noise is introduced. The probability density for the process describes the sensitivity of the process to noise. It can be used to find a variety of properties of the stochastic process, including time and location of state transitions and moments. Even though the probability density function holds all of the information for the stochastic process, much of the previous work on noisy nonlinear dynamics does not attempt to compute or approximate it. This is because it can be difficult to find the probability density when the underlying dynamical system is complicated and the noise is small.

In general, the goal is to find \( p(u,t) \), the probability that the process takes the value \( u \) at a given time \( t \). Given a system of stochastic differential equations (SDE’s) describing the process,

\[
du = a(u,t)dt + \sqrt{2\epsilon}dW, \quad (\epsilon \text{ is a constant})
\]

with \( W \) a vector of independent Brownian motions, the probability density \( p(u,t) \) satisfies the partial differential equation,

\[
\frac{\partial p}{\partial t} = \epsilon^2 \nabla^2 p - \nabla \cdot (ap),
\]

known as the Fokker-Planck equation (FPE) [6]. Note that the white noise in the SDE’s corresponds to the diffusion in (1.2) while the drift \( a(u,t) \) describes the convection. In this paper we consider problems in which a slow variation of a control parameter through a critical point results in a delay in the transition between states for the underlying noiseless dynamics ((1.1) with \( \epsilon = 0 \)). This delay can be changed significantly by very small noise (e.g. \( 0 < \epsilon \leq 10^{-4} \)). This is discussed in detail in Section 2.

There are several issues to consider in trying to find a solution to (1.2). For small noise \( (\epsilon \ll 1) \) it is expected that \( p(u,t) \) has large gradients, as is typical of solutions of problems with small diffusion.

* Work partially supported by NSF grant DMS-9704589
Then one might look for an asymptotic approximation, rather than a numerical solution. However, in the problems we study in this paper, the probability density has large gradients in some regions of space or time but not in others. Then an asymptotic approach may not be uniformly applicable. Therefore a numerical solution is necessary both to validate the asymptotic approximation where it is correct, and to provide a solution when the asymptotic approximation breaks down. The asymptotic expansions for (1.2) are discussed in detail in [7]. In this paper we focus on numerical solutions for the probability density, and illustrate their complementary relationship to the asymptotic results.

A variety of numerical methods can be considered for solving (1.2). For very small diffusion, grid-based numerical methods introduce numerical diffusion which obscures the effects of the small noise. Since the main purpose of the calculation is to determine the effect of the small noise, these methods are not appropriate. Another approach is to find the probability density from simulations of (1.1), which avoids the issue of numerically solving (1.2). While this is a direct and intuitive way to find \( p(u, t) \) given the SDE (1.1), a very large number of simulations is necessary to obtain a smooth result for the density function. The number of simulations necessary can increase with the dimension of \( u \), and it also depends on the concentration of the density.

In this paper we use a gradient-particle method (GPM) to find the probability densities from (1.2) for problems with noisy delay bifurcations. Even with very small noise, the concentration and the gradient of the probability density function for these processes can vary significantly. The GPM incorporates the natural idea of simulation of (1.1) into the numerical solution of (1.2) so that the effects of the noise are not obscured and the result for the density is smooth.

In Section 2 we give a description of a one-dimensional problem which illustrates why GPM’s are appropriate for this problem. In Section 3 we outline the algorithm, which uses ideas from [8] and [9]. In Section 4 we compare the GPM results for the one-dimensional problem of Section 2 with results from direct simulation (DS) and the asymptotic method of [7]. In Section 5 we compare the probability density function for the noisy FitzHugh-Nagumo model, and compare the gradient-particle method (GPM) results with direct simulation (DS) and the asymptotic results. In this paper we focus on the computational issues in finding the probability density function, and refer the reader to [7] for a discussion of the complementary asymptotic methods.

2 Motivation for the gradient-particle method

A description of a simple example of a delay bifurcation illustrates the important issues. The deterministic problem is

\[
\frac{dy}{dt} = cy - y^3.
\]

This equation was studied in [10] as a canonical model of the delay of a steady bifurcation. We summarize the results. If the control or bifurcation parameter, \( c \), is a constant, there are two steady state solutions: \( y = 0 \) for all \( c \) and \( y = \sqrt{c} \) for \( c > 0 \). The solution \( y = \sqrt{c} \) is stable for \( c > 0 \). That is, there is a steady bifurcation from the zero solution to the solution \( y = \sqrt{c} \) at the critical parameter value \( c = 0 \).

Now consider \( c = \mu t \), for \( 0 < \mu \ll 1 \). When the system is sub-critical, \( (\mu t < 0) \), \( y(t) \) decays exponentially. As \( t \) increases so that \( t > 0 \), \( y(t) \) remains small for a long interval of time. For \( \sqrt{\mu t} = O(1) \), \( y(t) \) grows rapidly, approaching \( y(t) \sim \sqrt{\mu t} \). This is known as a delay bifurcation, since the solution does not immediately increase rapidly as the coefficient \( \mu t \) varies slowly through the critical point \( \mu t = 0 \). This behavior can be seen from the exact solution of (2.1).

Now we consider the effect of noise on the delay bifurcation. Previous studies have considered the reduction of the delay with the addition of sinusoidal oscillations [10]. Numerical simulations of

\[
\frac{dy}{dt} = (\mu t y - y^3)dt + \sqrt{2}edW,
\]

where \( W \) is standard Brownian motion, demonstrate that the delay is reduced with the addition of noise [11], even though \( \epsilon = 10^{-4} \) (see Figure 2.1). The simulations of Figure 2.1 show that the noisy dynamics are close to the deterministic dynamics, except during the transition from \( y \ll 1 \) to \( y \sim \pm \sqrt{\mu t} \).

These simulations suggest that the probability density function is sharply peaked about either \( y = 0 \)
or $y = \pm \sqrt{\mu t}$, except in the transition region. This variation suggests that the concentration of the probability density function changes significantly over time.

A quantitative description of the delay can be obtained by considering the Fokker-Planck equation for the probability density $p(y, t)$, defined as

$$p(y, t)dy = P(y(t) \in (y, y + dy) \text{ at time } t),$$

for the process $y(t)$ described in (2.2). For $\epsilon \neq 0$ we look for the solution of,

$$p_t = \epsilon^2 p_{yy} - ((\mu ty - y^3)p)_y = 0, \quad p(0) = p_0(y).$$

An asymptotic approximation can be made for this density when the random dynamics are close to the deterministic dynamics [7], but from the few typical realizations in Figure 2.1, that leaves a significant region (in this case between $t = 30$ and $t = 50$) in which this approximation may fail. Therefore we are especially interested in computing $p(y, t)$ in this transition region. As mentioned in the Introduction, $p(y, t)$ gives the probability that the process has made the transition from $y \sim 0$ to $y \sim \pm \sqrt{\mu t}$ at time $t$, and can be used to compute moments of the process $y(t)$ or the expected time until this transition is made [7].

In Section 4 we use a GPM to compute the probability density function for this one-dimensional example of a delay bifurcation. In Section 5 we extend this method to a two-dimensional model, the FitzHugh-Nagumo model. There the noise also reduces the time of a delay bifurcation. We have several reasons for using GPM’s which follow from the discussion above.

1. The diffusion in the Fokker-Planck equation is very small for the cases of interest, that is $\epsilon \ll 1$. (In particular, $\epsilon \leq .0001$ in this paper.) Grid-based methods give numerical diffusion which swamps the diffusion due to the noise.

2. The probability density has steep gradients in some regions in space and time, and not others. This is due to the fact that the noisy dynamics follow the deterministic dynamics closely except in the transition region. Outside of the transition region the gradients are large, but within this region the gradients are no longer large. This variation in shape, and the transition of the process itself, make adaptive methods impractical, since frequent regridding would be required.

3. A straightforward simulation method involving only direct simulation of the SDE would require a large number of realizations in order to resolve the probability density in all regions of space and time of interest. The GPM uses the natural intuition of a simulation method, but requires much fewer realizations, since both particle positions and gradients are computed. We compare the GPM results to the direct simulation results in Sections 4 and 5.
3 The method

We outline the GPM for computing the probability density function \( p(x,t) \) for the process,

\[
\frac{dx}{dt} = a(x,t)dt + \sqrt{2}\sigma dW.
\]

Here \( x \) and \( a(x,t) \) are vectors (in one and two dimensions, for the applications in this paper) and \( dW \) is a vector of independent white noise processes. Then the probability density satisfies the Fokker-Planck equation

\[
p_t = \epsilon^2 \nabla^2 p - \nabla \cdot (a p), \quad p(x,0) = p_0(x).
\]

The equation for the gradient is obtained by differentiating (3.2). In one dimension \( x = x \) and \( a(x,t) = a(x,t) \), and the equation for \( p_x(x,t) \) is,

\[
(p_x)_t = \epsilon^2 \frac{\partial^2 p_x}{\partial x^2} - (ap_x)_x - a_{xx}p - a_x p_x.
\]

In two dimensions we use the notation of the application in Section 5, taking \( x = (u,v) \), \( g = (g_1,g_2) \equiv (p_u,p_v) \) and \( a = (a_1,a_2) \) and differentiating (3.2),

\[
\frac{\partial}{\partial t} \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \epsilon^2 \nabla^2 \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} - \frac{\partial}{\partial u} \left[ a_1 \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \right] - \frac{\partial}{\partial v} \left[ a_2 \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \right]
\]

\[= \begin{pmatrix} (a_{1u})_u & (a_{2u})_u \\ (a_{1v})_v & (a_{2v})_v \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} - \left[ \begin{pmatrix} (a_{1u})_u + (a_{2u})_u \\ (a_{1v})_v + (a_{2v})_v \end{pmatrix} \right] p.
\]

These equations are similar to those in [8], where the convection-diffusion equation for a chemical concentration in an incompressible flow was solved with GPM’s. The main difference between [8] and this application is that the equation for the gradient in this paper involves both \( \nabla p \) and \( p \) itself, while the equation for the gradient in [8] involves only the derivatives of \( p \). This is because the flow was incompressible (\( \nabla \cdot a = 0 \)) for the convection-diffusion equation studied in [8]. This is not true for (1.1) in general, and in particular, \( \nabla \cdot a \neq 0 \) for the delay bifurcation applications which we study here. It follows that there is a reconstruction of \( p \) from \( \nabla p \) in each time step. A similar reconstruction was required in [9] for the solution of a reaction-diffusion system using a gradient random walk method. We outline the steps of the algorithm, pointing out differences and similarities to [8] and [9]. Both [8] and [9] contain excellent discussions of GPM’s and comparisons with previous work in deterministic and random particle methods. We do not repeat their discussion here, but mention a few alternative approaches at the end of this section.

The algorithm is based on the observation that both the particle motion and the evolution of the gradient vectors contribute to changes in \( \nabla p \) and thus in \( p \) as well [8]. Specifically, in (3.3) the terms \( \epsilon^2 (p_x)_{xx} - (ap_x)_x \) correspond to changes in concentration due to the motion of the particles as described by (3.1). The terms \(-a_{xx}p - a_x p_x \) give the changes in shape due to the evolution of \( p_x \). Similarly in the two-dimensional case, the terms \( \epsilon^2 \nabla^2 (g_j) - \nabla \cdot (ag_j) \) for \( j = 1,2 \) are due to the motion of the particles, and the remaining terms on the right hand side of (3.4) describe changes in the gradient vectors. The motion of the particles is obtained by solving (3.1).

One additional note is that the probability densities described above are defined on an infinite domain. However, the probability densities are exponentially small outside a bounded region. Therefore we compute on a rectangular region which contains all but the tails of the probability density whose mass is essentially zero and set \( p(x,t) = 0 \) on the boundary of this region.

The algorithm:

Initial conditions: To each of \( N \) particles assign position vectors \( x_k \), the probability density \( p_k \equiv p_0(x_k) \) from the initial condition (3.2), and gradient vectors \( g_k \equiv \nabla p_k \) for \( k = 1, \ldots, N \).

At each time step:

1. Find the new position of the particles by solving (3.1). In this paper a second order Runge-Kutta method for SDE’s [12] was used.
2. Evolve the gradient vectors using the remaining terms from the equation. For the one-dimensional example, \( g_k = (p_k)_x \) and \((g_k)_t = -a_{xx}p_k - a_xg_k,\) and \(a_{xx}\) and \(a_x\) are evaluated at \(x_k.\) In two dimensions the equation is

\[
\frac{\partial g_k}{\partial t} = - \left[ \begin{array}{cc} (a_1)_u & (a_2)_u \\ (a_1)_v & (a_2)_v \end{array} \right] \left( \begin{array}{c} g_{1k} \\ g_{2k} \end{array} \right) - \left( \begin{array}{c} (a_1)_{uu} + (a_2)_{uv} \\ (a_1)_{uv} + (a_2)_{vv} \end{array} \right) p_k. \tag{3.5} \]

This step is similar to one used in [8], but there is an extra term involving \(p_k\) that was not present in the applications studied in [8], as discussed above. A second order Runge-Kutta method was used to advance \(g_k.\)

Note that this step is not present in the method used for reaction-diffusion equations in [9] and [13]. Instead changes in the shape of the density were obtained by either creating or destroying particles, depending on the sign of terms analogous to the right hand side of (3.5). In this paper the number of particles is constant throughout the computation, and changes in the shape of the density enter through the evolution of the gradient computed in this step.

3. Reconstruct \(p_k\) from its gradient vector \(g_k.\) In one dimension this is accomplished by sorting the particles and integrating \(g_k = (p_k)_x\) to get \(p_k.\) In two dimensions \(x_k = (u_k, v_k)\) and \(\nabla p(u, v; t)\) is determined on a lattice with points given by \((u_j, v_m),\) for \(j = 1, \ldots, N_u, m = 1, \ldots, N_v,\) as follows,

\[
\left( \begin{array}{c} p_u(u_j, v_m, t) \\ p_v(u_j, v_m, t) \end{array} \right) = \sum_{k=1}^{N} g_k(t) \delta_{x}((u - u_j), (v - v_m)), \tag{3.6} \]

where \(\delta_{x}\) approximates the two dimensional Dirac delta function. Then \(p(u, v, t)\) is obtained using the method outlined in [8], by solving two Poisson’s equations. Note that this reconstruction must be performed at every time step, since the evolution equation for the gradient, as in (3.5), depends on \(p.\) Briefly, the reconstruction of \(p(u, v, t)\) is computed on the rectangle \(a \leq u \leq b,\) and \(c \leq v \leq d,\)

\[
\begin{align*}
\nabla \phi &= p_u, \quad \phi(a, v) = \phi(b, v) = \phi(u, c) = \phi(u, d) = 0, \\
\nabla \psi &= p_v, \quad \psi(a, v) = \psi(b, v) = \psi(u, c) = \psi(u, d) = 0, \\
\psi(u, v; t) &= \phi_u + \psi_v + \text{constant}. \tag{3.7}
\end{align*}
\]

Here the rectangle is sufficiently large so that \(p(u, v, t)\) is essentially zero on the boundaries. This reconstruction method is discussed in [8]. As noted there, we found that this reconstruction gives a nonnegative result for the density. The constant in (3.7) was chosen in order to properly normalize \(p(x, y, t)\) for any \(t,\) so that the integral over space of the density is 1. We found that this constant approached zero as the number of points in the reconstruction lattice was increased, and was negligible as compared to the maximum of the density function.

In [9] the reconstruction step used a method similar to random vortex methods [14]. The method suggested by Anderson [15], uses Poisson’s formula to obtain a point-gradient formulation for the recovery of \(p(u, v, t).\) This method is expensive, but fast multipole methods of [16] can be used to reduce the computation time. Some smoothing problems were encountered in [9] in using the fast multipole methods. Since we found the implementation straightforward and the method (3.7) sufficiently fast we used the method of [8]. As noted there, this reconstruction method does introduce some errors when the gradients are very large. In our problem these errors are apparent in the tails of the density, but they did not affect the bulk of the density. These effects are discussed further in the following sections. We used the fishpack subroutines to solve (3.7). Future work could compare this reconstruction method with that of [15] for the noisy delay bifurcation problems described here.

## 4 The one-dimensional model of a steady bifurcation

First we give the results for the probability density for the one-dimensional steady delay bifurcation example as discussed in Section 2. The equation for \(p(y, t)\) is given by

\[
\begin{align*}
p_t &= e^y p_{yy} - ((ut)^2 - y^3)p_y = 0, \tag{4.1} \\
p(0) &= p_0(y). \tag{4.2}
\end{align*}
\]
Figure 4.1: Comparison of results for the density at $t = 24$, $t = 36$ and $t = 40$, with $\mu = .01$ and $\epsilon = 10^{-4}$ and initial condition (4.3). a) The solid line is the GPM solution with 10,000 particles, the dash-dotted line is the GPM solution for 15,000 particles, and the dotted line is the result from the DS of (2.2) with 60,000 realizations (particles). Although it is not shown, the asymptotic approximation of [7] is in good agreement with all of these results. b) The solid line is the GPM solution with 10,000 particles and the dotted line is the result for DS of (2.2) with 60,000 realizations (particles). The dash-dotted line is the asymptotic (Gaussian-type) approximation, which does not capture the behavior of the tails for this value of $t$, when the density is less concentrated. c) The solid line is the GPM solution with 10,000 particles, and the dotted line is the result from the DS of (2.2) with 60,000 particles. The numerical results are in good agreement as far as location and general shape of the density, but the DS results are not smooth.
As an initial condition, we take

$$p_0(y) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left( -\frac{(y - y_0)^2}{2\sigma^2} \right). \tag{4.3}$$

As discussed above, $y$ decays exponentially for $t < 0$. Then starting with (4.3) is the same as taking an initial condition at $t < 0$ so that $y(0) < \epsilon$ and the noise dominates the dynamics at the critical point $t = 0$. The figures compare computations of $p(y, t)$ using 10,000 and 15,000 particles in the GPM with $p(y, t)$ obtained from a direct simulation (DS) of the SDE (2.2). The results at $t = 36$ are also compared with an asymptotic approximation which is valid through a part of the transition region, as discussed in detail in [7].

In Figure 4.1a, we see good agreement between all methods, noting that the direct simulation with 60,000 realizations (particles) has statistical errors, while the GPM method gives a much smoother solution. In Figure 4.1b this difference is even more obvious. Figure 4.1 shows that the GPM can be used to compute the probability density through the transition region, and gives the probability density before and after this transition has occurred (mean transition time is shown to be $t \approx 28$ in [7] for this case). Specifically, Figure 4.1a shows the probability density while the process is still concentrated around $y = 0$. Figure 4.1b shows the density within the time interval in which the process makes the transition from $y \ll 1$ to $[y] \sim \sqrt{\mu t}$, and Figure 4.1c shows the density at a time when the transition is likely to have occurred. Thus the GPM result is valid throughout the transition interval, which is a longer time range than the validity interval for the asymptotic approximation. The reconstruction of the density from its gradient provides the smoothing which gives good results even though the GPM uses only 10,000 particles, as compared to the non-smooth results from the DS of the particle motion, using 60,000 particles. Even though the GPM requires additional computations of the evolution of the gradients and the reconstruction of $p(y, t)$ from $p_y(y, t)$, the computation times for the GPM with 10,000 particles and the DS with 60,000 are comparable (overnight on a Pentium II workstation to compute to $t = 40$). Note that the time step can not be too large, in order to keep computational errors below the noise level.

Recalling that the density $p(y, t)$ is reconstructed from $p_y(y, t)$ at each step in the GPM method, one might expect that errors in the reconstruction can feedback into the computation. We found that these errors could be significant, especially in the transition zone. Then the gradient is not large, except near $y = \pm \sqrt{\mu t}$, where the density function drops off sharply. The reconstruction errors for $p(y, t)$ near $y = \pm \sqrt{\mu t}$ can appear in the computation near this drop in the density function. By decreasing the time-step and using a sufficiently fine grid, we reduced this error substantially.

5 The two-dimensional model: FitzHugh-Nagumo

Next we consider the two dimensional model of FitzHugh-Nagumo [2] [17],

$$du = b(v - \gamma u)dt, \tag{5.1}$$

$$dv = (-f(v) - u + I(t))dt + \sqrt{2\epsilon}dW, \tag{5.2}$$

$$f(v) = v(v - 1)(v - a),$$

where $a$, $b$, and $\gamma$ are physical parameters. This is a reduced model of the propagation of neural impulses in the giant axon of a squid, with $v$ the potential difference across the membrane of the axon, and $u$ the recovery current. $I(t)$ is an applied current, and plays the role of the bifurcation parameter. When $I$ is constant, there is a Hopf bifurcation at $I = I_c$; for $I < I_c$ the stable solution is constant, while for $I > I_c$ the stable solution is periodic. Figure 5.1 shows that there is delay in the bifurcation from constant to periodic behavior as the applied current $I(t) = \mu t$ is increased slowly ($\mu \ll 1$). In the realizations with small noise ($\epsilon = 10^{-5}$ in Figure 5.1) this delay is significantly reduced. Only the solution for $v$ is shown in Figure 5.1, since the variable $u$ is virtually slaved to $v$. That is, the behavior of $u$ follows that of $v$, with a time lag. Then a transition to oscillatory behavior in $v$ is followed by a transition to oscillatory behavior in $u$. Since the noise causes a reduction in the delay until the transition to the oscillatory state, we anticipate that the gradient of the probability density varies significantly in the transition region. The
Probability density for this process satisfies (1.2) with 
\[ a_1 = b(v - \gamma u) \text{ and } a_2 = -f(v) - u + I(t) \] 
so that
\[ \frac{\partial p}{\partial t} = \varepsilon^2 \nabla^2 p - ((b(v - \gamma u)p)_u - ((-f(v) - u + I(t))p)_v. \] 

We determine \( p(u, v, t) \) using the GPM, and compare to the results from the direct simulation of (5.1) and the asymptotic approximation of [7]. In the computations the initial condition was
\[ p(u, v, t_0) = p_{\text{asymp}}(u, v, t_0), \]
where \( p_{\text{asymp}}(u, v, t_0) \) is the asymptotic approximation to the density obtained in [7] and \( t_0 > 0 \) is some time before the transition time. This choice of initial condition is justified by the dynamics of the system. For a typical realization, as shown in Figure 5.1, the stochastic and deterministic behavior are close until \( t \) approaches the transition time. The asymptotic approximation and the numerical computations agree for \( t \) below the transition interval, which Figure 5.1 suggests is \( t < 370 \) for the parameter values used there. We verified this agreement by starting at different initial times \( t_0 < 350 \) with initial conditions of the type (5.4). As expected, the asymptotic approximation is valid until large values of \( t \) where the probability function is less concentrated around the deterministic dynamics.

The use of the initial condition (5.4) also illustrates the complementary nature of the asymptotic approximation and the computational method; the asymptotic approximation is used as an initial condition at \( t_0 \) before the transition time, and then the computational method is used to find the probability density in the transition region, where the validity of the asymptotic method is suspect. This is a very practical reduction of the amount of computation necessary, since the computational method does not perform as well as the asymptotic method for values of \( t_0 \) before the transition time, where the density function has large gradients. The computation time is significantly reduced, since the computations are done for \( t > t_0 > 0 \), rather than for all \( t > 0 \).

In Figure 5.2 we compare the results from GPM and the asymptotic method of [7] for \( t = 375 \) and the same parameter values as in Figure 5.1. These figures show good agreement between the two methods. This is expected, since the process has not reached the expected transition time (calculated as \( t \approx 383 \) in [7]), and the density is still concentrated around the deterministic behavior. In Figure 5.3 we compare the GPM and asymptotic method for \( t = 390 \) and \( t = 410 \) for the same parameter values as in 5.2, and in Figure 5.4 we compare the results from the GPM and asymptotic method for a different value of \( \mu = .003 \).

In both of these cases the results are compared for times before and after the transition to oscillatory behavior. Again good agreement is seen for the location of the concentration of the density and the shape of the peak. The shape and location of the density is consistent with the simulations and the dynamics.
Figure 5.2: Comparison of the probability density $p(u, v, t)$ computed with the GPM of this paper (top row) with the asymptotic result from [7] (bottom row). The parameter values are the same as in Figure 5.1, and $\epsilon = 10^{-5}$. The left column is the surface plot, and the right column is the corresponding contour plot. Good agreement in both location and shape is obtained from the two methods. Note that this is for a time before the transition has occurred.
$t = 375$

$t = 390$

$t = 410$

Figure 5.3: Comparison of the contour plots for $p(u, v, t)$ for the asymptotic result [7] with GPM results for $t = 375, t = 390, t = 410$. The parameters are the same as in Figure 5.2. The top row are the results from the GPM. The bottom row are the asymptotic results.
of (5.1). The density remains centered around the deterministic dynamics, but the variable $v$ makes the transition to the oscillatory state first, so that the density is less concentrated about the deterministic value of $v$. The gradients are larger in the variable $u$ since $u$ follows $v$ with a time lag. That is, the gradients in $u$ are also decreasing, but not as fast as in the $v$ direction.

Figure 5.4: Comparison of the contour plots for $p(u, v, t)$ for the asymptotic result [7] with GPM results for the parameter values $\epsilon = 10^{-5}$, $a = .2$, $b = .05$, $\gamma = .4$ and $f(t) = \mu t$ with $\mu = .003$. The plots are compared for times before $(t = 130)$ and after $(t = 145)$ the transition. Again there is a large change in the shape of the density. Accumulation of reconstruction errors is reduced as described in the text. The top row are the results from the GPM. The bottom row are the asymptotic results.

Errors in the reconstruction of $p$ can accumulate in our computations, since the reconstruction must be carried out at each time step, and the result for $p$ is used in the evolution of the gradient vectors (3.5) at the following time step. The computations in [8] did not have this accumulation of error, since the concentration (analogous to $p(u, v, t)$) did not appear in the equation for the gradient (see Section 3). There is some oscillation in the tails of the numerical result which is especially evident at $t = 410$ in Figure 5.3. This is due to the reconstruction process, which has difficulty resolving the very steep gradients. We found an increased effect of these errors when $|p_u|$ is much larger than $|p_v|$, which is far into the transition zone. Since this $p$ is reconstructed from $\nabla p$ at every time step, these errors can accumulate over long computation times, as is required for these computations. As in the one-dimensional problem of Section 4, these errors can be reduced by decreasing the time step and adjusting the reconstruction lattice. We chose the lattice and time step so that the computations would not take longer than 10-20 hours on a Pentium II workstation, times which are comparable to direct simulation times. The largest reconstruction lattice that we used was 512x256, and the smallest time step was .00001. Even with these extreme cases the
error in the tails of the density was not completely eliminated. We also reduced the errors by using the asymptotic approximation for an initial condition at larger values of $t$, where the approximation is still valid and agrees with the computation. (For Figure 5.3, we used (5.4) with $t_0 = 390$, and for Figure 5.4 we used (5.4) with $t_0 = 135$.) This adjustment reduces the computing time and the accumulation of error. The resulting reduction of the error can be seen in comparing Figures 5.3 and 5.4. In Figure 5.4, the parameter $\mu$ is larger ($\mu = .003$) and the transition occurs at a faster rate. Then the probability density after the transition is obtained from computing over a shorter time interval, and there is less accumulation of reconstruction errors.

In Figure 5.5 we compare the results for $p(u, v, t)$ from solving (5.3) with a GPM and from the direct simulation (DS) of (5.1). The results from the DS use 80,000 realizations, while the GPM uses 10,000. The statistical errors in the DS are increasingly evident as time increases since the density is spreading out. An increasing number of realizations would be necessary to properly approximate the density for later times. The results for the GPM method are smooth and the number of particles is sufficient for resolving the peak in the density. Other solutions were obtained using a larger number of particles in the GPM, but they gave essentially the same results.
In Figure 5.6 we compare the result for the marginal density

\begin{equation}
    p(v, t) = \int_{-\infty}^{\infty} p(u, v, t) du.
\end{equation}

The dynamics of \( v \) indicate when the transition occurs, since the transition in \( u \) follows that of \( v \).

![Figure 5.6: Comparison of the marginal density \( p(v, t) \) obtained by the DS of (5.1) (80,000 realizations) and the GPM for \( t = 377 \) and \( t = 415 \). Here \( a = .2, b = .05, \gamma = A, \mu = .001, \) and \( \epsilon = 10^{-5} \).

Therefore \( p(v, t) \) itself may be used to determine when the transition has occurred, as in [7]. Figure 5.6 compares the results for the marginal density, using \( p(u, v, t) \) for the GPM and the DS of (2.2). Note that even with the extra step of smoothing, that is, integration in \( u \), the results from the DS do not give a smooth result for the marginal density.

6 Conclusion

A gradient particle method is used to solve Fokker-Planck equations (FPE’s) for noisy delay bifurcations. The solution of the FPE gives the time-dependent probability density function for the stochastic process.

In these problems, a delay in the transition between states occurs in the deterministic dynamics when a bifurcation parameter varies slowly through the bifurcation point. This delay is reduced by the presence of noise, even if it is very small. The probability density gives a complete description of the process, and in particular it describes the transition of the process. Since there is considerable variation in the gradient of the density, numerical methods must be used to solve the FPE, yielding results which are complementary to those obtained asymptotically. The small diffusion of the cases of interest (very small noise) and the variation in the concentration of the density suggest that the gradient particle methods are practical methods for computing the solution to the FPE. These methods incorporate simulation of the noisy process (motion of the particles) with the evolution of the gradients of the probability density. The reconstruction of the density from its gradient introduces a smoothing so that fewer realizations are necessary than in computing the density with a direct simulation.

We compute probability densities through the transition regions and compare the gradient particle method results to asymptotic results and direct simulations. The GPM’s yields satisfactory results throughout the transition regions, where the asymptotic method is no longer valid, and yield results which are much smoother than those obtained with direct simulation. Some caution must be used in applying the gradient particle methods, since the density must be reconstructed from its gradient at each time step, and errors can accumulate over long time periods. These errors can be reduced by adjusting the
time step and reconstruction grid, and by using the asymptotic approximation as an “initial condition”, which reduces the length of the time of computation.

Acknowledgements

The author would like to thank Christoph Börgers for many helpful discussions.

References


