Abstract

Recently proposed particle MCMC methods provide a flexible way of performing Bayesian inference for parameters governing stochastic kinetic models defined as Markov jump processes (MJPs). Each iteration of the scheme requires an estimate of the marginal likelihood calculated from the output of a sequential Monte Carlo scheme (also known as a particle filter). Consequently, the method can be extremely computationally intensive. We therefore aim to avoid most instances of the expensive likelihood calculation through use of a fast approximation. We consider two approximations: the chemical Langevin equation diffusion approximation (CLE) and the linear noise approximation (LNA). Either an estimate of marginal likelihood under the CLE, or the tractable marginal likelihood under the LNA can be used to calculate a first step acceptance probability. Only if a proposal is accepted under the approximation do we then run a sequential Monte Carlo scheme to compute an estimate of the marginal likelihood under the true MJP and construct a second stage acceptance probability that permits exact (simulation based) inference for the MJP. We therefore avoid expensive calculations for proposals that are likely to be rejected. We illustrate the method by considering inference for parameters governing a Lotka-Volterra system and a simple model of gene expression.

Keywords: Markov jump process, chemical Langevin equation, linear noise approximation, particle MCMC, delayed acceptance.

*andrew.golightly@ncl.ac.uk
1 Introduction

Computational systems biology (Kitano, 2002) is typically concerned with developing dynamic simulation models of biological processes. Recent recognition of the importance of stochastic effects in intra-cellular processes has motivated the need for models that incorporate intrinsic stochasticity (Wilkinson, 2009). A Markov jump process (MJP) can be used to give a formal description of a biochemical network driven by stochastic reaction events whose effect is to change biochemical species numbers by an integer amount. Our goal is to perform inference for the rate constants that govern the MJP using time course data that may be incomplete and/or subject to measurement error.

Exact (simulation based) Bayesian inference for the MJP representation of the network was the subject of Boys et al. (2008). The authors proposed two MCMC schemes that targeted the joint posterior of the rate constants and latent reaction events but found the statistical efficiency of their method to be relatively poor. It was shown in Golightly and Wilkinson (2011) how a recently proposed particle MCMC algorithm (Andrieu et al., 2010) can be applied to this class of models. In particular, the particle marginal Metropolis-Hastings (PMMH) scheme allows a joint update of the rate constants and (latent) process which can alleviate common mixing problems when sampling high-dimensional target densities that may exhibit strong correlations. The proposal mechanism involves drawing a new parameter value from an arbitrary proposal kernel and drawing new values of each latent state from a sequential Monte Carlo (SMC) approximation to the distribution of latent states conditional on the proposed new parameter value. The acceptance probability requires computation of a realisation of an unbiased estimator of marginal likelihood which can be readily obtained from the output of the SMC scheme. Consequently, at each iteration of the MH scheme, an SMC algorithm must be implemented. The method can be extremely computationally intensive as the SMC algorithm typically must generate many realisations of the MJP, with each realisation obtained from an algorithm such as the stochastic simulation algorithm (SSA) of Gillespie (1977). By using a computationally cheaper approximation to the marginal likelihood we avoid running the computationally more expensive SMC algorithm at most iterations of the MH scheme, but we still maintain the posterior under the MJP as the target distribution of the MH scheme.

The simplest approximation of the MJP is the macroscopic rate equation (MRE) which
ignores the discreteness and stochasticity of the MJP by modelling species dynamics with a set of coupled ordinary differential equations (van Kampen, 2001). The diffusion approximation or chemical Langevin equation (Gillespie, 2000) on the other hand, ignores discreteness but not stochasticity by modelling the biochemical network with a set of coupled stochastic differential equations (SDEs). Whilst inference for the parameters governing nonlinear multivariate SDEs is possible (Golightly and Wilkinson, 2008), the marginal likelihood under this model is intractable. Despite this, Golightly and Wilkinson (2011) show that inference is possible under this model using a PMMH algorithm, and this approach can result in computational savings when compared to a similar scheme targeting the posterior under the MJP.

Further computational savings can be made by considering a linear noise approximation (LNA) (van Kampen, 2001; Komorowski et al., 2009; Fearnhead et al., 2012) which is given by the MRE plus a stochastic term accounting for random fluctuations about the MRE. Under the LNA the latent process follows a multivariate Gaussian distribution and, under an assumption of Gaussian measurement error, the marginal likelihood is tractable.

Our novel contribution is to exploit the tractability of the LNA by proposing a particle analogue of the ‘delayed acceptance’ MCMC scheme described in Christen and Fox (2005). Essentially, to avoid calculating an estimate of marginal likelihood under the MJP for proposals that are likely to be rejected, proposed parameter draws are initially corrected using a cheap approximation, such as the marginal likelihood computed under the LNA. A similar approach has been proposed independently by Smith (2011) for performing inference for nonlinear economic models such as (discrete time) stochastic volatility models. In addition, we also consider a scenario in which the marginal likelihood under the approximation is intractable, but can be estimated cheaply (relative to the same calculation under the MJP) using a particle filter. Use of the CLE in the preliminary screening step falls into this category. In both cases, we show that the resulting MCMC scheme targets the correct marginal, that is, the marginal parameter posterior under the MJP.

The remainder of this paper is organised as follows. In Section 2 we describe the Markov jump process model and associated inference problem. The CLE and LNA are briefly reviewed. We describe the PMMH algorithm in Section 3.1 before considering a modification to allow delayed acceptance in Section 3.3. We apply the method to a Lotka-Volterra system and a model of gene
expression in Section 4. Finally, conclusions are drawn in Section 5.

2 Stochastic kinetic models

Consider a biochemical reaction network involving $u$ species $X_1, X_2, \ldots, X_u$ and $v$ reactions $R_1, R_2, \ldots, R_v$, with reaction $R_i$ given by

$$R_i : \ p_{i1}X_1 + p_{i2}X_2 + \cdots + p_{iu}X_u \rightarrow q_{i1}X_1 + q_{i2}X_2 + \cdots + q_{iu}X_u$$

Let $X_{j,t}$ denote the number of molecules of species $X_j$ at time $t$, and let $X_t$ be the $u$-vector $X_t = (X_{1,t}, X_{2,t}, \ldots, X_{u,t})'$. The $v \times u$ matrix $P$ consists of the coefficients $p_{ij}$, and $Q$ is defined similarly. The $u \times v$ stoichiometry matrix $S$ is defined by

$$S = (Q - P)'$$

and encodes important structural information about the reaction network. In particular, if $\Delta R$ is a $v$-vector containing the number of reaction events of each type in a given time interval, then the system state should be updated by $\Delta X$, where

$$\Delta X = S\Delta R.$$

Under the standard assumption of mass-action stochastic kinetics, each reaction $R_i$ is assumed to have an associated rate constant, $c_i$, and a propensity function, $h_i(X_t, c_i)$ giving the overall hazard of a type $i$ reaction occurring. That is, we model the system as a Markov jump process, and for an infinitesimal time increment $dt$, the probability of a type $i$ reaction occurring in the time interval $(t, t + dt]$ is $h_i(X_t, c_i)dt$. The hazard function for a particular reaction of type $i$ is often assumed to take the form

$$h_i(X_t, c_i) = c_i \prod_{j=1}^{u} \left( \frac{X_{j,t}}{p_{ij}} \right).$$

Let $c = (c_1, c_2, \ldots, c_v)'$ and $h(X_t, c) = (h_1(X_t, c_1), h_2(X_t, c_2), \ldots, h_v(X_t, c_v))'$. Values for $c$ and the initial system state $X_0 = x_0$ complete the specification of the Markov process. Although this process is rarely analytically tractable for interesting models, it is straightforward to forward-simulate exact realisations of this Markov process using a discrete event simulation method. This
is due to the fact that if the current time and state of the system are $t$ and $X_t$ respectively, then the time to the next event will be exponential with rate parameter

$$h_0(X_t, c) = \sum_{i=1}^{v} h_i(X_t, c_i),$$

and the event will be a reaction of type $R_i$ with probability $h_i(X_t, c_i)/h_0(X_t, c)$ independently of the waiting time. Forward simulation of process realisations in this way is typically referred to as the stochastic simulation algorithm (Gillespie, 1977). See Wilkinson (2012) for further background on stochastic kinetic modelling.

### 2.1 Chemical Langevin equation

We present here an informal intuitive construction of the chemical Langevin equation (CLE), and refer the reader to Gillespie (2000) for further details.

Consider an infinitesimal time interval, $(t, t + dt]$. Over this time, the reaction hazards will remain constant almost surely. The occurrence of reaction events can therefore be regarded as the occurrence of events of a Poisson process with independent realisations for each reaction type. Therefore, if we write $dR_t$ for the $v$-vector of the number of reaction events of each type in the infinitesimal time increment, it is clear that the elements are independent of one another and that the $i$th element is a Po($h_i(X_t, c_i)dt$) random quantity. From this we have that $E(dR_t) = h(X_t, c)dt$ and $\text{Var}(dR_t) = \text{diag}\{h(X_t, c)\}dt$. It is therefore clear that

$$dR_t = h(X_t, c)dt + \text{diag}\{\sqrt{h(X_t, c)}\}dW_t$$

is the Itô stochastic differential equation (SDE) which has the same infinitesimal mean and variance as the true Markov jump process (where $dW_t$ is the increment of a $v$-dimensional Brownian motion). Now since $dX_t = SdR_t$, we obtain

$$dX_t = Sh(X_t, c)dt + \sqrt{S\text{diag}\{h(X_t, c)\}S'}dW_t,$$

(1)

where now $X_t$ and $W_t$ are both $u$-vectors. Equation (1) is the SDE most commonly referred to as the chemical Langevin equation or diffusion approximation, and represents the diffusion process which most closely matches the dynamics of the associated Markov jump process, and can be shown to approximate the SKM increasingly well in high concentration scenarios (Gillespie,
In the absence of an analytic solution to (1), a numerical solution can be constructed. For example, the Euler-Maruyama approximation is

\[ \Delta X_t \equiv X_{t+\Delta t} - X_t = S h(X_t, c)\Delta t + \sqrt{S \text{diag}\{h(X_t, c)\}} S' \Delta W_t \]

where \( \Delta W_t \) is a mean zero Normal random vector with variance matrix \( \text{diag}\{ \Delta t \} \).

We require a computationally efficient approximation to the Markov jump process for use in a delayed acceptance particle MCMC scheme (described in Section 3.3). Performing exact (simulation based) inference for the diffusion approximation has been the focus of Golightly and Wilkinson (2005), Purutcuoglu and Wit (2007), and Golightly and Wilkinson (2011) among others. Although the latter find that a particle MCMC scheme based on the CLE can be more computationally efficient than a similar scheme that works with the Markov jump process directly, calculation of an estimate of marginal likelihood under the CLE (as is necessary at every iteration of a particle MCMC scheme) can be computationally expensive. To facilitate greater computational savings, we therefore also consider a linear noise approximation (LNA) (van Kampen, 2001; Komorowski et al., 2009; Fearnhead et al., 2012; Stathopoulos and Girolami, 2013) which generally possesses a greater degree of numerical and analytic tractability than the CLE (Wilkinson, 2012). This is the subject of the next section.

### 2.2 Linear noise approximation

Our derivation of the LNA closely follows the approach of Fearnhead et al. (2012) and we refer the reader to the references therein for a more detailed discussion. Essentially, we derive the LNA by partitioning \( X_t \) into a deterministic path \( z_t \) and a residual stochastic process \( M_t \). We calculate the LNA for a general SDE before formulating it as an approximation to the CLE.

Consider now a general SDE satisfied by a process \( \{X_t, t \geq 0\} \) of the form

\[ dX_t = \alpha(X_t)dt + \epsilon \beta(X_t)dW_t, \]  

(2)

where \( \epsilon \ll 1 \), and let \( z_t \) be the (deterministic) solution to

\[ \frac{dz_t}{dt} = \alpha(z_t). \]  

(3)

We assume that \( ||X_t - z_t|| \) is \( O(\epsilon) \) over a time interval of interest and substitute \( X_t = z_t + \epsilon M_t \)
into equation (2) to give

\[ d(z_t + \epsilon M_t) = \alpha(z_t + \epsilon M_t)dt + \epsilon\beta(z_t + \epsilon M_t)dW_t. \]

We then Taylor expand \( \alpha(\cdot) \) and \( \beta(\cdot) \) about \( z_t \) and collect terms of \( O(\epsilon) \) to give the SDE satisfied by \( M_t \) as

\[ dM_t = F_t M_t dt + \beta(z_t) dW_t \tag{4} \]

where \( F_t \) is the Jacobian matrix with \((i, j)\)th element \( \partial \alpha_i(z_t) / \partial z_j,t \) and \( \alpha_i(z_t) \) refers to the \( i \)th element of \( \alpha(z_t) \). We use \( \epsilon \) to indicate that the stochastic term in (2) is small and, essentially, that the drift term dominates the diffusion coefficient. However \( \epsilon \) plays no role in the evolution equations, (3) and (4). Without loss of generality, therefore, we simplify the exposition by setting \( \epsilon = 1 \). To further simplify the notation we also drop the explicit dependence of the hazard function on \( c \), and of the mean and variance of \( M_t \) on both \( c \) and \( z_t \).

For the CLE, we have

\[ \alpha(X_t) = S h(X_t), \quad \beta(X_t) = \sqrt{S \text{diag}\{h(X_t)\}}S'. \]

The linear noise approximation of the CLE is therefore defined through

\[ \frac{dz_t}{dt} = Sh(z_t) \tag{5} \]

and

\[ dM_t = F_t M_t dt + \sqrt{S \text{diag}\{h(z_t)\}}S' dW_t \tag{6} \]

where \( F_t \) has \((i, j)\)th element \( S \partial h_i(z_t) / \partial z_j,t \).

For fixed or Gaussian initial conditions, that is \( M_{t_1} \sim N(m_{t_1}, V_{t_1}) \), the SDE in (6) can be solved explicitly to give

\[ (M_t|c) \sim N(m_t, V_t) \tag{7} \]

where \( m_t \) is the solution to the deterministic ordinary differential equation (ODE)

\[ \frac{dm_t}{dt} = F_t m_t \tag{8} \]

and similarly

\[ \frac{dV_t}{dt} = V_t F_t' + S \text{diag}\{h(z_t)\}S' + F_t V_t. \tag{9} \]
Hence, the solution of equation (6) requires the solution of a system of coupled ODEs. In the absence of an analytical solution to these equations, a numerical solution can be used. The approximating distribution of $X_t$ can then be found as

$$(X_t|c) \sim N(z_t + m_t, V_t).$$

(10)

### 3 Inference

We now consider the task of performing Bayesian inference for the rate constants governing the Markov jump process. First, let us augment the rate vector, $c$, to include any additional parameters that arise from the observation process and assign to it a prior density, $p(c)$. Suppose that the MJP $X = \{X_t | 1 \leq t \leq T\}$ is not observed directly, but (perhaps partial) observations (on a regular grid) $y = \{y_t | t = 1, 2, \ldots, T\}$ are available and are conditionally independent (given $X$) with conditional probability distribution $p(y_t|x_t, c)$. In this work, we consider Bayesian inference for $c$ via the marginal posterior density

$$p(c|y) = \int p(c, x|y) dx,$$

(11)

where

$$p(c, x|y) \propto p(c) p(x|c) \prod_{t=1}^{T} p(y_t|x_t, c)$$

and $p(x|c)$ is the probability of the Markov jump process. Since the posterior in (11) will typically be unavailable in closed form, samples must usually be generated through a suitable MCMC scheme.

In what follows, for simplicity, we assume that the initial value of the MJP, $X_1 = x_1$, is a known fixed quantity, and we take $z_1 = x_1$ so that $m_1$ is the length-$u$ zero vector and $V_1$ is the $u \times u$ zero matrix. If $X_1$ were unknown then it could be assigned a prior and treated as an additional parameter in the augmented rate vector.

#### 3.1 Particle marginal Metropolis-Hastings

We consider the special case of the particle marginal Metropolis-Hastings (PMMH) scheme of Andrieu et al. (2010) and Andrieu et al. (2009) in which only samples from the marginal parameter posterior are required. Noting the standard decomposition $p(c|y) \propto p(y|c)p(c)$, we run a
Metropolis-Hastings (MH) scheme with proposal kernel \( q(c^*|c) \) and accept a move from \( c \) to \( c^* \) with probability

\[
\min \left\{ 1, \frac{\hat{p}(y|c^*)p(c^*)}{\hat{p}(y|c)p(c)} \times \frac{q(c|c^*)}{q(c^*|c)} \right\}
\]

where \( \hat{p}(y|c) \) is a sequential Monte Carlo (SMC) or ‘particle filter’ estimate of the intractable marginal likelihood term \( p(y|c) \). The PMMH scheme as described here is an example of a pseudo-marginal Metropolis-Hastings scheme (Beaumont, 2003; Andrieu and Roberts, 2009), and provided that \( \hat{p}(y|c) \) is unbiased (or has a constant multiplicative bias that does not depend on \( c \)), it is possible to verify that the method targets the marginal \( p(c|y) \). Let \( u \) denote all random variables generated by the SMC algorithm and write the SMC estimate of marginal likelihood as \( \hat{p}(y|c) = p(y|c, u) \). Augmenting the state space of the chain to include \( u \), it is straightforward to rewrite the acceptance ratio in (12) to find that the chain targets the joint density

\[
p(c, u|y) \propto p(y|c, u)p(u|c)p(c)
\]

Marginalising over \( u \) then gives

\[
\int p(c, u|y)du \propto p(c) \int p(y|c, u)p(u|c)du = p(c)p(y|c).
\]

The key insight here is that the SMC scheme can be constructed to give an unbiased estimate of the marginal likelihood \( p(y|c) \) under some fairly mild conditions involving the resampling scheme (Del Moral, 2004). The scheme therefore targets the correct marginal \( p(c|y) \). Although interest here is in the marginal \( p(c|y) \) the PMMH scheme can be used to sample the joint density \( p(c, x|y) \). At each step of the algorithm, a new path \( x^* \) is proposed from the SMC approximation of \( p(x^*|y, c^*) \). The acceptance probability is as in (12). For further details, we refer the reader to Andrieu et al. (2010). The (special case of the) PMMH algorithm and details of the SMC scheme that we use are given in Appendices A.1 and A.2.

### 3.2 Inference using the CLE and LNA

Although the marginal likelihood under the CLE is intractable, a PMMH scheme can be implemented to perform inference for this model. In the simplest version of the scheme, we replace
draws of the MJP in step 2(a) of the SMC scheme with draws of a numerical solution of the CLE, for example, using the Euler-Maruyama approximation. This is the focus of Golightly and Wilkinson (2011) and further details can be found therein.

For additive Gaussian observation regimes, the marginal likelihood under the LNA is tractable. This tractability has been exploited for the purposes of parameter inference by Komorowski et al. (2009), Fearnhead et al. (2012) and Stathopoulos and Girolami (2013). Komorowski et al. (2009) apply the LNA over the entire time interval for which observations are available. In particular, the ODE component of the LNA is solved once over the whole time-course for a given initial condition. As discussed in Fearnhead et al. (2012), this can lead to a poor approximation to the distribution of \( X_t \) as \( t \) gets large, due to the mismatch between the stochastic and ODE solution. We therefore adopt the approach proposed in Fearnhead et al. (2012) and restart the LNA at each observation time \( t \), initialising \( z_t \) to the posterior mean of \( X_t \) given all observations up to time \( t \). The algorithm for constructing the marginal likelihood under an additive Gaussian observation regime using this approach is given in Appendix A.3. Since the MJP takes integer values, the observation error is, however, unlikely to be Gaussian, and in this article we consider observations with a Poisson distribution, the mean of which is the value of the true process. Nonetheless, we may still use the LNA to obtain a tractable approximation to the marginal likelihood under the true MJP. We approximate the observation density \( p(y_t|x_t) \) by a Gaussian density with mean and variance given by the ODE solution (5). That is, we apply the algorithm in Appendix A.3 with \( \Sigma \) replaced by a diagonal matrix containing the components of \( z_t \) for which observations are made. This tractable approximation can then be used in the delayed acceptance scheme.

### 3.3 Delayed acceptance particle marginal Metropolis-Hastings

In order to improve the efficiency of the PMMH algorithm for the MJP we aim to limit the number of runs of the computationally expensive SMC scheme for the MJP. Ideally we want to run the SMC scheme only for parameter values which are likely to lead to acceptance in the PMMH algorithm. We do this by choosing a particular proposal kernel in the PMMH scheme of Appendix A.1. This proposal kernel is based on a preliminary screening step involving an approximate model which is less computationally intensive than the MJP, such as the LNA or
Our proposed algorithm for taking advantage of the CLE approximation, which we call delayed acceptance PMMH (daPMMH), is outlined in Algorithm 1; the algorithm which takes advantage of the LNA is a slight simplification of this. Both algorithms have the following basic structure. First a candidate set of parameter values is proposed, then a decision is made whether to accept or reject these values based on a MH step with target density \( p_a(c|y) \propto p_a(y|c)p(c) \), which is the posterior density of parameters under the approximate model (for example, the LNA or the CLE); here \( p_a(y|c) \) represents the marginal likelihood under the approximate model. If the proposed parameter values are accepted at this first stage then they undergo another MH step with target density \( p(c|y) \propto p(y|c)p(c) \), which is the marginal posterior density under the MJP. The idea here is that the first stage weeds out ‘poor’ parameter values. Consequently, the computationally expensive SMC algorithm for the MJP is only implemented for ‘good’ parameter values which are likely to be accepted at the second stage.

When the CLE is used as the approximate model the marginal likelihood \( p_a(y|c) \) is not available analytically, so we replace it with an unbiased estimate \( \hat{p}_a(y|c) \) obtained from an SMC scheme which targets \( p_a(x|y,c) \), the conditional density of the latent states under the approximate model, given the observed data and the parameter values. We therefore have to run a particle filter at both stages of the daPMMH algorithm, as one is always needed at stage 2 to give an unbiased estimate \( \hat{p}(y|c) \) of the MJP marginal likelihood \( p(y|c) \). We note, however, that despite the CLE requiring a run of an SMC scheme to obtain \( \hat{p}_a(y|c) \) this may still be much faster to run than the SMC scheme for the MJP (with the same number of particles).

Our daPMMH algorithm is an extension of the delayed acceptance MH (daMH) algorithm of Christen and Fox (2005), which is a version of the ‘surrogate transition method’ of Liu (2001). Specifically, we have extended the daMH algorithm by replacing all intractable marginal likelihoods by unbiased estimates obtained from appropriate SMC schemes. Our extension of the daMH algorithm to an intractable likelihood at Stage 1 is essential when the approximate model is the CLE since the marginal likelihood under the CLE is intractable. However, when the LNA is chosen as the approximate model this extra level of complexity is not necessary; we simply replace the marginal likelihood estimates \( \hat{p}_a(y|c) \) in Algorithm 1 with the exact values \( p_a(y|c) \) since these are available numerically (see Appendix A.3 for details).
Algorithm 1 Delayed acceptance PMMH (daPMMH)

1. Initialisation, \( i = 0, \)
   
   (a) set \( c^{(0)} \) arbitrarily,
   
   (b) run a particle filter targeting \( p(\mathbf{x} | \mathbf{y}, c^{(0)}) \), and let \( \hat{p}(\mathbf{y} | c^{(0)}) \) denote the marginal likelihood estimate,
   
   (c) run a particle filter targeting \( p_a(\mathbf{x} | \mathbf{y}, c^{(0)}) \), and let \( \hat{p}_a(\mathbf{y} | c^{(0)}) \) denote the marginal likelihood estimate under the approximate model.

2. For iteration \( i \geq 1, \)
   
   (a) sample \( c^* \sim q(\cdot | c^{(i-1)}) \),
   
   (b) **Stage 1**
      
      (i) run a particle filter targeting \( p_a(\mathbf{x} | \mathbf{y}, c^*) \), and let \( \hat{p}_a(\mathbf{y} | c^*) \) denote the marginal likelihood estimate under the approximate model,
      
      (ii) with probability
      
      \[
      \alpha_1(c^{(i-1)}, c^*) = \min \left\{ 1, \frac{\hat{p}_a(\mathbf{y} | c^*) p(c^*)}{\hat{p}_a(\mathbf{y} | c^{(i-1)}) p(c^{(i-1)})} \frac{q(c^{(i-1)} | c^*)}{q(c^* | c^{(i-1)})} \right\},
      \]
      
      run a particle filter targeting \( p(\mathbf{x} | \mathbf{y}, c^*) \), let \( \hat{p}(\mathbf{y} | c^*) \) denote the marginal likelihood estimate and go to 2(c); otherwise, set \( c^{(i)} = c^{(i-1)} \), \( \hat{p}(\mathbf{y} | c^{(i)}) = \hat{p}(\mathbf{y} | c^{(i-1)}) \), \( \hat{p}_a(\mathbf{y} | c^{(i)}) = \hat{p}_a(\mathbf{y} | c^{(i-1)}) \), increment \( i \) and return to 2(a).
   
   (c) **Stage 2**
      
      With probability
      
      \[
      \alpha_2(c^{(i-1)}, c^*) = \min \left\{ 1, \frac{\hat{p}(\mathbf{y} | c^*) p(c^*)}{\hat{p}(\mathbf{y} | c^{(i-1)}) p(c^{(i-1)})} \frac{\hat{p}_a(\mathbf{y} | c^{(i-1)}) p(c^{(i-1)})}{\hat{p}_a(\mathbf{y} | c^*) p(c^*)} \right\}
      \]
      
      set \( c^{(i)} = c^* \), \( \hat{p}(\mathbf{y} | c^{(i)}) = \hat{p}(\mathbf{y} | c^*) \) and \( \hat{p}_a(\mathbf{y} | c^{(i)}) = \hat{p}_a(\mathbf{y} | c^*) \) otherwise set \( c^{(i)} = c^{(i-1)} \), \( \hat{p}(\mathbf{y} | c^{(i)}) = \hat{p}(\mathbf{y} | c^{(i-1)}) \) and \( \hat{p}_a(\mathbf{y} | c^{(i)}) = \hat{p}_a(\mathbf{y} | c^{(i-1)}) \). Increment \( i \) and return to 2(a).
marginal likelihoods by unbiased estimates, our daPMMH algorithm still targets the (exact) posterior density of the parameters under the MJP, $p(c|y)$, as we outline in Section 3.3.1. Note that in an independent technical report, Smith (2011) proved that the daPMMH algorithm has $p(c|y)$ as its target density when the marginal likelihood under the approximate model is tractable. In Section 3.3.1 we generalise the argument of Smith (2011) to the case of an SMC-based marginal likelihood estimate for the approximate model.

### 3.3.1 Validity of delayed acceptance PMMH

In this Section we show that the daPMMH algorithm (Algorithm 1) is a valid MCMC scheme which targets a distribution that admits $p(c|y)$ as a marginal distribution. The validity of the daPMMH algorithm can be established by viewing it as a pseudo-marginal Metropolis-Hastings scheme with extended state space $\{c, u_1, u_2\}$, where $u_1$ and $u_2$ denote the auxiliary random variables generated in the SMC schemes for the approximate model and for the MJP, respectively.

In what follows we write the SMC estimate of marginal likelihood for the approximate model $\hat{p}_a(y|c)$ as $p_a(y|c, u_1, u_2)$, and write the SMC estimate of marginal likelihood for the MJP model $\hat{p}(y|c)$ as $p(y|c, u_1, u_2)$. Note that we condition on both $u_1$ and $u_2$ in the notation for both marginal likelihood estimates in order to stress the importance of the extended state space. However, in practice, $p_a(y|c, u_1, u_2)$ does not depend on $u_2$, and $p(y|c, u_1, u_2)$ does not depend on $u_1$.

Using the identity $\hat{p}_a(y|c) = p_a(y|c, u_1, u_2)$ for the SMC estimate of marginal likelihood for the approximate model, and explicitly incorporating the auxiliary random variables $u_1$ and $u_2$, the Stage 1 acceptance probability (13) can be written as

$$
\alpha_1(c, c^*) = \min \left\{ 1, \frac{\hat{p}_a(y|c^*)p(c^*) q(c|c^*)}{\hat{p}_a(y|c)p(c) q(c^{|c})} \right\}
= \min \left\{ 1, \frac{p_a(y|c^*, u_1^*, u_2^*)p(u_1^*|c^*)p(u_2^*|c^*)p(c^*) q(c|c^*)p(u_1|c)p(u_2|c)}{p_a(y|c, u_1, u_2)p(u_1|c)p(u_2|c)p(c) q(c^{|c})p(u_1^*|c^*)p(u_2^*|c^*)} \right\}
= \min \left\{ 1, \frac{p_a(y|c^*, u_1^*, u_2^*)p(u_1^*|c^*)p(u_2^*|c^*)p(c^*) \hat{q}(c, u_1, u_2|c^*, u_1^*, u_2^*)}{p_a(y|c, u_1, u_2)p(u_1|c)p(u_2|c) \hat{q}(c^{|c}) \hat{q}(c^*, u_1^*, u_2^*)} \right\}
= \min \left\{ 1, \frac{p_a(c^*, u_1^*, u_2^*) \hat{q}(c, u_1, u_2|c^*, u_1^*, u_2^*)}{p_a(c, u_1, u_2|c) \hat{q}(c^{|c}) \hat{q}(c^*, u_1^*, u_2^*)} \right\}
= \tilde{\alpha}_1(\{c, u_1, u_2\}, \{c^*, u_1^*, u_2^*\}).
$$

It can be seen that the target density at Stage 1 is the posterior density under the approximate
model, \( p_a(c, u_1, u_2 | y) \). Marginalising over the auxiliary variables in \( p_a(c, u_1, u_2 | y) \) gives \( p_a(c | y) \). Therefore the Stage 1 target density is the (exact) posterior density under the approximate model despite using an unbiased estimator of marginal likelihood under the approximate model, \( \hat{p}_a(y | c) \), in place of \( p_a(y | c) \). The augmented proposal density at Stage 1 is

\[
\tilde{q}(c^*, u_1^*, u_2^* | c, u_1, u_2) = q(c^* | c)p(u_1^* | c^*)p(u_2^* | c^*).
\]

Note that the generation of \( u_2^* | c^* \) is hypothetical since \( u_2 \) does not appear in the reduced form of the acceptance probability \( \alpha_1(c, c^*) \). Therefore the computationally intensive SMC scheme for the MJP need not be run at Stage 1, which potentially reduces the computational overhead.

At Stage 2, using the identities \( \tilde{p}(y | c) = p(y | c, u_1, u_2) \) and \( \tilde{p}_a(y | c) = p_a(y | c, u_1, u_2) \) for the SMC estimates of marginal likelihood, and explicitly incorporating the auxiliary random variables \( u_1 \) and \( u_2 \), the MH acceptance probability (14) can be written as

\[
\alpha_2(c, c^*) = \min \left\{ \frac{\tilde{p}(y | c^*)p(c^*)}{\tilde{p}(y | c)p(c)} \frac{\tilde{p}_a(y | c)p(c)}{\tilde{p}_a(y | c^*)p(c^*)}, \frac{p(y | c, u_1, u_2)p(u_1^* | c^*)p(u_2^* | c^*)}{p(y | c, u_1, u_2)c(p(u_1 | c)p(u_2 | c) | p(c)) \frac{p_a(y | c, u_1, u_2)p(u_1 | c)p(u_2 | c) | p(c)}{p_a(y | c^*, u_1^*, u_2^*)p(u_1^* | c^*)p(u_2^* | c^*) | p(c)}} \right\}
\]

\[
= \min \left\{ \frac{p(c^*, u_1^*, u_2^* | y)}{p(c, u_1, u_2 | y)} \frac{p_a(y | c^*, u_1^*, u_2^*)p(u_1^* | c^*)p(u_2^* | c^*)}{p_a(y | c^*, u_1^*, u_2^*)p(u_1^* | c^*)p(u_2^* | c^*)} \right\}
\]

\[
= \min \left\{ \frac{q_2(c, u_1, u_2 | c^*, u_1^*, u_2^*)}{q_2(c^*, u_1^*, u_2^* | c, u_1, u_2)} \right\}
\]

\[
= \tilde{\alpha}_2(c, u_1, u_2, \{ c^*, u_1^*, u_2^* \}).
\]

Note that because the Stage 1 transition kernel \( K(c, u_1, u_2 | c^*, u_1^*, u_2^*) \) is the proposal density at Stage 2, we must show that the ratio of proposal densities at Stage 2 is equal to the ratio of Stage 1 transition kernels in order to demonstrate that Algorithm 1 describes a valid MH scheme with the stated target density, \( p(c | y) \). Since the Stage 1 Markov chain has stationary density \( p_a(c, u_1, u_2 | y) \), the Stage 1 transition kernel \( K(c, u_1, u_2 | c^*, u_1^*, u_2^*) \) satisfies the detailed balance equation

\[
p_a(c, u_1, u_2 | y)K(c^*, u_1^*, u_2^* | c, u_1, u_2) = p_a(c^*, u_1^*, u_2^* | y)K(c, u_1, u_2 | c^*, u_1^*, u_2^*).
\]

The detailed balance equation can be rewritten to give

\[
p_a(y | c, u_1, u_2)p(u_1 | c)p(u_2 | c)K(c^*, u_1^*, u_2^* | c, u_1, u_2)
\]

\[
= p_a(y | c^*, u_1^*, u_2^*)p(u_1^* | c^*)p(u_2^* | c^*)K(c, u_1, u_2 | c^*, u_1^*, u_2^*).
\]

14
from which it follows that the ratio of Stage 1 transition kernels is
\[
\frac{K(c, u_1, u_2|c', u_1', u_2')}{K(c', u_1', u_2'|c, u_1, u_2)} = \frac{p_a(y|c, u_1, u_2)p(u_1|c)p(u_2|c)p(c)}{p_a(y|c', u_1', u_2')p(u_1'|c')p(u_2'|c')p(c')}
\]
\[
= \frac{q_2(c, u_1, u_2|c', u_1', u_2')}{q_2(c', u_1', u_2|c, u_1, u_2)}
\]
which is the ratio of Stage 2 proposal densities. Therefore, the Markov chain described by Algorithm 1 has \(p(c, u_1, u_2|y)\) as its stationary density. Marginalising over the auxiliary variables \(u_1\) and \(u_2\) gives the desired marginal posterior density \(p(c|y)\).

### 3.3.2 Comments on efficiency

Christen and Fox (2005) note that with a fast approximate model daMH algorithms are less computationally expensive — that is, they exhibit lower CPU times for the same number of iterations — than standard MH algorithms that do not employ delayed acceptance. They also note that daMH algorithms are less statistically efficient than standard MH algorithms that do not employ delayed acceptance. Here statistical efficiency relates to the mixing of the Markov chain, and can be measured by the effective sample size (ESS), the number of independent samples that are equivalent in information content to the actual number of dependent samples from the Markov chain. Clearly, computational time is dictated by the speed with which \(p_a(y|c)\) (or its estimate \(\hat{p}_a(y|c)\)) is computed, and statistical efficiency is dictated by the accuracy of the approximation \(p_a(y|c)\) or \(\hat{p}_a(y|c)\) to \(p(y|c)\). For example, \(p_a(y|c)\) under the LNA will be faster to compute than \(\hat{p}_a(y|c)\) under the CLE since the latter requires a run of an SMC algorithm. However, we might expect the CLE (at least with a small Euler timestep) to provide a better approximation to the MJP than the LNA, since the LNA is, in some sense, a simplified version of the CLE. Increasing the time-step \(\Delta t\) in the CLE will decrease the computation time but should also decrease the accuracy of the approximation; the trade-off in terms of computational efficiency between these two factors merits further investigation.

Another factor which will effect statistical efficiency is the variability associated with the SMC-based estimate of marginal likelihood \(\hat{p}_a(y|c)\). An algorithm using \(\hat{p}_a(y|c)\) will be less statistically efficient than an idealised algorithm which uses \(p_a(y|c)\) (for the same approximate model). We might expect, therefore, that using the LNA as the approximate model, with its tractable marginal likelihood, may lead to increased statistical efficiency over the CLE-based
approximation, although this depends on the accuracy of LNA.

The daPMMH scheme (using either the LNA or CLE) requires specification of a number of particles \( N \) to be used in the SMC scheme at stage 2. As noted by Andrieu and Roberts (2009), the mixing efficiency of the PMMH scheme decreases as the variance of the estimated marginal likelihood increases. This problem can be alleviated at the expensive of greater computational cost by increasing \( N \). This therefore suggests an optimal value of \( N \) and finding this choice is the subject of Pitt et al. (2012) and Doucet et al. (2013). The latter suggest that \( N \) should be chosen so that the variance in the noise in the estimated log-posterior is around 1. Pitt et al. (2012) note that the penalty is small for a value between 0.25 and 2.25. We therefore recommend performing an initial pilot run of daPMMH to obtain an estimate of the posterior mean for the parameters \( c \), denoted \( \bar{c} \). The value of \( N \) should then be chosen so that \( \text{Var}(\log \hat{p}(y|\bar{c})) \) is around 1. When the CLE is used as a surrogate model, we must also specify a number of particles (say \( N_1 \)) to be used in stage 1. For simplicity, we take \( N_1 = N \). Provided the CLE is a reasonable approximation to the MJP, we may expect that \( N_1 \) provides a suitable trade-off between computational cost and accuracy (in terms of the variance of the estimated marginal likelihood under the CLE).

In the next section we show empirically that our daPMMH algorithm (with either the CLE or the LNA as the approximate model) can lead to improvements in overall computational efficiency (in terms of ESS normalised by CPU time) over a vanilla PMMH scheme for the MJP.

4 Applications

4.1 Lotka-Volterra

Following Boys et al. (2008), we consider first a Lotka-Volterra model of predator and prey interaction comprising three reactions:

- \( \mathcal{R}_1 : X_1 \xrightarrow{c_1} 2X_1 \)
- \( \mathcal{R}_2 : X_1 + X_2 \xrightarrow{c_2} 2X_2 \)
- \( \mathcal{R}_3 : X_2 \xrightarrow{c_3} \emptyset. \)

For simplicity of notation we drop the explicit dependence of the current state \( X = (X_1, X_2)' \) and the deterministic approximation \( z = (z_1, z_2)' \) on time, \( t \). The stoichiometry matrix is given
by

\[ S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \]

and the associated hazard function is

\[ h(X, c) = (c_1X_1, c_2X_1X_2, c_3X_2)' \]

The diffusion approximation can be calculated by substituting \( S \) and \( h(X, c) \) into the CLE (1) to give respective drift and diffusion coefficients of

\[ \alpha(X, c) = \begin{pmatrix} c_1X_1 - c_2X_1X_2 \\ c_2X_1X_2 - c_3X_2 \end{pmatrix}, \quad \beta(X, c) = \begin{pmatrix} c_1X_1 + c_2X_1X_2 - c_2X_1X_2 \\ -c_2X_1X_2 + c_2X_1X_2 + c_3X_2 \end{pmatrix}. \]

For the linear noise approximation, the Jacobian matrix \( F_t \) is given by

\[ F_t = \begin{pmatrix} c_1 - c_2z_2 & -c_2z_1 \\ c_2z_2 & c_2z_1 - c_3 \end{pmatrix}. \]

We simulated a synthetic dataset by generating 50 observations at integer times using the Gillespie algorithm with initial conditions \( x_1 = (70, 80)' \) and parameter values \( c = (1.0, 0.005, 0.6)' \) taken from Wilkinson (2012). Predator values were discarded leaving 50 observations on prey only. These were then corrupted via an error distribution for which the marginal likelihood under the LNA is intractable:

\[ Y_t \sim \text{Poisson}(x_{1t}), \quad t = 1, 2, \ldots, 50. \]

A tractable approximation to the true marginal likelihood under the MJP, for use in stage 1 of the delayed acceptance scheme was obtained using the LNA as described in Section 3.2. In what follows, for simplicity, we assume that the latent initial state \( x_1 \) is known.

For brevity, we refer to the MCMC algorithm targeting the posterior under the MJP that uses the LNA inside the delayed acceptance PMMH scheme as \( daPMMH-LNA \). Similarly, when using the CLE inside the delayed acceptance scheme we refer to this as \( daPMMH-CLE \). Finally, we designate the vanilla PMMH scheme without delayed rejection as \( PMMH \). Using independent Uniform \( U(-8, 8) \) priors for each \( \log(c_i) \) we performed a pilot run of the PMMH scheme with 50 particles to give an approximate covariance matrix \( \hat{\text{Var}}(c) \) and approximate posterior mean \( \hat{c} \). Further pilot runs were then implemented with \( c \) fixed at \( \hat{c} \) and numbers of particles ranging
Table 1: Lotka-Volterra model. Marginal posterior means and standard deviations (in parentheses) for log($c_i$) from the output of the each inference scheme.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>log($c_1$)</th>
<th>log($c_2$)</th>
<th>log($c_3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True value</td>
<td>0.000</td>
<td>-5.298</td>
<td>-0.511</td>
</tr>
<tr>
<td>PMMH</td>
<td>-0.067 (0.053)</td>
<td>-5.281 (0.067)</td>
<td>-0.484 (0.075)</td>
</tr>
<tr>
<td>daPMMH-CLE (∆t = 0.2)</td>
<td>-0.063 (0.054)</td>
<td>-5.283 (0.071)</td>
<td>-0.484 (0.079)</td>
</tr>
<tr>
<td>daPMMH-CLE (∆t = 0.125)</td>
<td>-0.064 (0.056)</td>
<td>-5.286 (0.069)</td>
<td>-0.489 (0.077)</td>
</tr>
<tr>
<td>daPMMH-CLE (∆t = 0.0625)</td>
<td>-0.066 (0.056)</td>
<td>5.284 (0.070)</td>
<td>-0.487 (0.078)</td>
</tr>
<tr>
<td>daPMMH-LNA</td>
<td>-0.067 (0.057)</td>
<td>-5.281 (0.073)</td>
<td>-0.484 (0.081)</td>
</tr>
</tbody>
</table>

from 50 to 250. We found that using 200 particles gave the variance in the noise in the estimated log-posterior as 1.16. We therefore took $N = 200$ particles for the main monitoring runs, which consisted of $10^5$ iterations of each scheme, with the log($c_i$) updated in a single block using a Gaussian random walk proposal kernel. For PMMH and daPMMH-CLE, we used an innovation variance matrix given by $S = \frac{2.382}{3} \text{Var}(c)$. For daPMMH-LNA, we found that multiplying $S$ by 3 gave an improved overall efficiency (compared with simply using $S$); no such improvement was observed when using the CLE. For daPMMH-CLE, we considered three levels of discretisation, namely, ∆t = 0.2, 0.125, 0.0625. All algorithms were coded in C and were run on a desktop computer with a 2.83 GHz clock speed.

Table 1 summarises the output of each inference scheme. As expected, we see that parameter samples obtained under the delayed acceptance PMMH schemes are consistent with those obtained under the vanilla PMMH scheme, and with the true values that were used to produce the data. Table 2 shows the stage 1 acceptance probability, $\alpha_1$, the stage 2 acceptance probability $\alpha_{2|1}$, the CPU time, the minimum (over the 3 parameters) effective sample size (ESS$_{min}$) and the minimum effective sample size per second, relative to the corresponding value obtained from the vanilla PMMH scheme. Whilst the daPMMH-CLE scheme gives an improvement in overall efficiency (as measured by relative ESS$_{min}$ per second) for all values of ∆t employed, the effect of the discretisation is clear. The marginal likelihood under the CLE is closer to that under the MJP for smaller ∆t, resulting in greater statistical efficiency of the daPMMH-CLE scheme. This can also be seen by inspecting the stage 2 acceptance probability reported in Ta-
Figure 1: Log-marginal likelihood estimates under the MJP (log(\(\hat{\mathbf{p}}(\mathbf{y}|\mathbf{c})\))) against the corresponding log-marginal likelihood estimate under (a) the CLE (\(\Delta t = 0.2\)), (b) the CLE (\(\Delta t = 0.125\)), (c) the CLE (\(\Delta t = 0.0625\)) and (d) the LNA. All plots are obtained using 10,000 values of \(c\) sampled from the posterior \(p(c|\mathbf{y})\) for the Lotka-Volterra model.
Algorithm & $\alpha_1$ & $\alpha_2$ & CPU time (s) & ESS$_{min}$ & Rel. ESS$_{min}$/s \\
--- & --- & --- & --- & --- & --- \\
PMMH & 0.071 & 1.000 & 40680 & 1160 & 1.00 \\
daPMMH-CLE ($\Delta t = 0.2$) & 0.127 & 0.135 & 7606 & 251 & 1.16 \\
daPMMH-CLE ($\Delta t = 0.125$) & 0.106 & 0.286 & 7439 & 472 & 2.23 \\
daPMMH-CLE ($\Delta t = 0.0625$) & 0.106 & 0.330 & 10490 & 494 & 1.65 \\
daPMMH-LNA & 0.034 & 0.440 & 1367 & 416 & 10.66 \\

Table 2: Lotka-Volterra model. Stage 1 acceptance probability $\alpha_1$, stage 2 acceptance probability $\alpha_2$, CPU time (to the nearest second), minimum effective sample size (ESS$_{min}$, to the nearest whole number) and minimum effective sample size per second, relative to the corresponding value obtained from the vanilla PMMH scheme. All values are based on $10^5$ iterations.

Table 2. Naturally, this improvement comes at a greater computational cost suggesting an optimal value of $\Delta t$ between 0.2 and 0.0625 for this example. The CPU time for $\Delta t = 0.2$ is actually greater than that for $\Delta t = 0.125$. This is due, in part, to the relatively high stage 1 acceptance rate. Further insight into this result can be gained from Figure 1, which plots estimates of the marginal likelihood (on the log-scale) under PMMH against the corresponding value obtained under each approximation, for 10,000 values of $c$ sampled from the posterior $p(c|y)$. The stage 1 and 2 acceptance rates depend only on the estimates of the log-likelihood at the proposed and current values through their difference. Thus the efficiency of the algorithm is unaffected by any fixed shift of the points from the line through the origin with a slope of one. However, variability about a line with this slope is important and we see greater variability in the estimates obtained for $\Delta t = 0.2$ resulting in a reduction in statistical efficiency for the daPMMH-CLE ($\Delta t = 0.2$) scheme, with proposed values that were accepted at stage 1 being rejected at stage 2.

The daPMMH-LNA scheme on the other hand requires minimal tuning. The LNA gives an analytic form for the (approximate) marginal likelihood and therefore does not require implementation of a particle filter during the first stage of the delayed acceptance scheme. Moreover, the LNA solution involves solving a set of ODEs, for which standard routines, such as the lsoda package (Petzold, 1983), exist. Therefore, pre-specification of a suitable time discretisation is not required. We find for this example that the daPMMH-LNA scheme outperforms the vanilla PMMH scheme by a factor of more than 10.
4.2 Gene Expression

Finally, we consider a simple model of gene expression involving three biochemical species (DNA, mRNA, protein) and four reaction channels (transcription, mRNA degradation, translation, protein degradation):

\[ \mathcal{R}_1 : \ DNA \xrightarrow{\kappa_{R,t}} DNA + R \]
\[ \mathcal{R}_2 : \ R \xrightarrow{\gamma_R} \emptyset \]
\[ \mathcal{R}_3 : \ R \xrightarrow{\kappa_P} R + P \]
\[ \mathcal{R}_4 : \ P \xrightarrow{\gamma_P} \emptyset. \]

This system has been analysed by Komorowski et al. (2009) among others, and we therefore adopt the same notation to aid the exposition.

Let \( X_t = (R_t, P_t)' \) denote the system state at time \( t \), where \( R_t \) and \( P_t \) are the respective number of mRNA and protein molecules. As in Komorowski et al. (2009), we take \( \kappa_{R,t} \) to be the time dependent transcription rate of the gene. Specifically,

\[ \kappa_{R,t} = b_0 \exp \left( -b_1 (t - b_2)^2 \right) + b_3 \]

so that transcription rate increases for \( t < b_2 \) and tends to the baseline \( b_3 \) for \( t > b_2 \). We denote the vector of unknown parameters by

\[ c = (\gamma_R, \gamma_P, \kappa_P, b_0, b_1, b_2, b_3)' \]

and our goal is to perform inference for these parameters. The stoichiometry matrix associated with the system is given by

\[ S = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \]

and the associated hazard function is

\[ h(X_t, c) = (\kappa_{R,t}, \gamma_R R_t, \kappa_P R_t, \gamma_P P_t)' . \]

For the linear noise approximation, we have the Jacobian matrix as

\[ F_t = \begin{pmatrix} -\gamma_R & 0 \\ \kappa_P & -\gamma_P \end{pmatrix} . \]
Figure 2: A single realisation of the gene expression system obtained using the first reaction method. Protein numbers used in the artificial dataset are shown as circles.

We simulated a synthetic dataset by generating observations every 15 minutes for 25 hours (giving 100 observations in total) noting that care must be taken when simulating from the MJP representation of this system, due to the time dependent hazard of reaction $R_1$. We used initial conditions of $x_1 = (10, 150)'$ and parameter values $c = (0.44, 0.52, 10, 15, 0.4, 7, 3)'$ with units of time in hours. As in Komorowski et al. (2009) we created a challenging data-poor scenario by discarding observations on mRNA levels. Unlike Komorowski et al. (2009) we corrupt the observations via an error distribution for which the marginal likelihood under the LNA is intractable:

$$Y_t \sim \text{Poisson}(P_t), \quad t = 1, 2, \ldots, 100.$$ 

The data are shown in Figure 2. As in the previous section, to obtain a tractable approximation to the marginal likelihood under the MJP, for use in stage one of the delayed acceptance scheme, we approximate the observation density $p(y_t|p_t)$ by a Gaussian density with mean and variance given by the ODE solution (5).

For each parameter, we assumed the same prior distributions as in Komorowski et al. (2009) including informative priors for the degradation rates to ensure identifiability. Specifically, we
Figure 3: Gene expression model. Marginal posterior distributions based on the output of the PMMH scheme for the MJP (solid line) and daPMMH-LNA scheme (dashed line). True values of each $\log(c_i)$ are indicated by a straight line.

have that

$$
\gamma_R \sim \Gamma(19.36, 44), \quad \gamma_P \sim \Gamma(27.04, 52),
$$

$$
\kappa_P \sim \text{Exp}(0.01), \quad b_0 \sim \text{Exp}(0.01),
$$

$$
b_1 \sim \text{Exp}(1), \quad b_2 \sim \text{Exp}(0.1),
$$

where $\Gamma(a, b)$ denotes the Gamma distribution with mean $a/b$ and $\text{Exp}(b)$ denotes the Exponential distribution with mean $1/b)$. For simplicity, we fixed the initial latent states at their true values. We performed a pilot run of the PMMH scheme with 50 particles to give an approximate covariance matrix $\hat{\text{Var}}(c)$ and approximate posterior mean $\hat{c}$. By performing further pilot runs we found that using 200 particles gave the variance in the noise in the estimated log-posterior as 1.13. We therefore took $N = 200$ particles for the main monitoring runs, which consisted of $10^5$ iterations of each scheme, with the $\log(c_i)$ updated in a single block using a Gaussian random walk proposal kernel. For PMMH, we used an innovation variance matrix given by $S = \frac{2.38^2}{3} \hat{\text{Var}}(c)$. For daPMMH-LNA we used an innovation variance matrix of $\lambda S$ for $\lambda = 1, 2, 3$. 

23
Table 3: Gene expression model. Marginal posterior means and standard deviations (in parentheses) for each parameter from the output of the PMMH scheme for the MJP and daPMMH-LNA scheme (with $\lambda = 2$).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True value</th>
<th>PMMH</th>
<th>daPMMH-LNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log(\gamma_R)$</td>
<td>-0.821</td>
<td>-0.730 (0.207)</td>
<td>-0.736 (0.198)</td>
</tr>
<tr>
<td>$\log(\gamma_P)$</td>
<td>-0.654</td>
<td>-0.682 (0.152)</td>
<td>-0.673 (0.156)</td>
</tr>
<tr>
<td>$\log(\kappa_P)$</td>
<td>2.303</td>
<td>2.351 (0.229)</td>
<td>2.338 (0.248)</td>
</tr>
<tr>
<td>$\log(b_0)$</td>
<td>2.708</td>
<td>2.702 (0.408)</td>
<td>2.676 (0.406)</td>
</tr>
<tr>
<td>$\log(b_1)$</td>
<td>-0.916</td>
<td>-0.881 (0.368)</td>
<td>-0.847 (0.459)</td>
</tr>
<tr>
<td>$\log(b_2)$</td>
<td>1.946</td>
<td>1.864 (0.073)</td>
<td>1.865 (0.074)</td>
</tr>
<tr>
<td>$\log(b_3)$</td>
<td>1.099</td>
<td>1.064 (0.442)</td>
<td>1.076 (0.437)</td>
</tr>
</tbody>
</table>

Figure 3 and Table 3 summarise the output of the PMMH and daPMMH-LNA schemes. We find that parameter samples obtained using the PMMH and daPMMH-LNA schemes are roughly consistent with the true values that produced the data. Table 4 shows stage 1 acceptance probability $\alpha_1$, stage 2 acceptance probability $\alpha_{21}$, CPU time, minimum (over the parameters) effective sample size ($\text{ESS}_{\text{min}}$) and minimum effective sample size per second, relative to the corresponding value obtained from the PMMH scheme. The effect of increasing the scaling parameter, $\lambda$, (which in turn increases the innovation variance for the Gaussian random walk update) can clearly be seen. As computation of (an approximation to) the marginal likelihood under the LNA is extremely cheap relative to the MJP, larger moves should be tried at stage 1. When $\lambda = 2$ we see a 6 fold improvement in overall efficiency (as measured by relative $\text{ESS}_{\text{min}}$ per second). The result is relatively robust to the choice of $\lambda$, with a relative $\text{ESS}_{\text{min}}$ per second of 3.88 when $\lambda = 1$.

Figure 4 shows some mismatch between the marginal likelihood under the LNA and estimates of the marginal likelihood under the MJP, for values in the region $[-455, -440]$ perhaps suggesting a discrepancy in tails of the marginal parameter posteriors under the LNA and MJP. Despite this, the LNA provides a reasonable approximation to the MJP in regions of high posterior density and we recorded an empirical stage 2 acceptance probability of around 0.25.
Figure 4: Log-marginal likelihood estimates $\log(\hat{p}(y|c))$ under the MJP against the corresponding log-marginal likelihood estimate under the LNA, using 10,000 values of $c$ sampled from the posterior $p(c|y)$ for the gene expression model.

5 Discussion

In this paper we have proposed a particle MCMC analogue of the delayed acceptance MCMC scheme of Christen and Fox (2005) and applied it to the problem of parameter estimation in Markov jump process representations of biochemical networks. The need for such an approach is motivated by the potentially huge computational cost of performing particle MCMC for the MJP directly, where each iteration requires implementation of a particle filter with $N$ particles, and a complete run of the stochastic simulation algorithm is required for each particle.

The delayed acceptance PMMH scheme aims to avoid calculating an estimate of marginal likelihood (and therefore running the particle filter) under the MJP for proposals that are likely to be rejected, by implementing a preliminary screening step that uses a cheap approximation of the marginal likelihood. We explored two approximations, the chemical Langevin equation (CLE) and the linear noise approximation (LNA). The LNA can be viewed as an approximation to the CLE, and in order to maintain tractability of the approximate likelihood under the Poisson observation regime we have introduced a further approximation. Thus, providing the Euler time-
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>CPU time (s)</th>
<th>ESS$_{\text{min}}$</th>
<th>Rel. ESS$_{\text{min}}$/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMMH</td>
<td>0.065</td>
<td>1.000</td>
<td>190585</td>
<td>561</td>
<td>1.00</td>
</tr>
<tr>
<td>daPMMH-LNA ($\lambda = 1$)</td>
<td>0.117</td>
<td>0.261</td>
<td>13261</td>
<td>278</td>
<td>3.88</td>
</tr>
<tr>
<td>daPMMH-LNA ($\lambda = 2$)</td>
<td>0.058</td>
<td>0.246</td>
<td>6638</td>
<td>220</td>
<td>6.48</td>
</tr>
<tr>
<td>daPMMH-LNA ($\lambda = 3$)</td>
<td>0.032</td>
<td>0.234</td>
<td>3750</td>
<td>97</td>
<td>5.04</td>
</tr>
</tbody>
</table>

Table 4: Gene expression model. Stage 1 acceptance probability $\alpha_1$, stage 2 acceptance probability $\alpha_2$, CPU time (to the nearest second), minimum effective sample size (ESS$_{\text{min}}$, to the nearest whole number) and minimum effective sample size per second, relative to the corresponding value obtained from the vanilla PMMH scheme. All values are based on $10^5$ iterations.

step is not too large, the CLE leads to larger effective sample sizes. However the intractability of the likelihood under the CLE necessitates the use of a particle filter which is computationally much more costly than the LNA. In the example which we considered the speed advantage of the LNA far outweighed the disadvantage through inaccuracy, although both schemes were at least twice as efficient as the standard PMMH scheme.

The efficiency of both of the proposed delayed acceptance PMMH schemes can be improved in a number of ways. Both schemes can be parallelised and will benefit from recent work on the use of graphics cards for Monte Carlo methods (Lee et al., 2010). In addition, in high signal-to-noise scenarios, the variance of the marginal likelihood estimator under both the CLE and MJP could be reduced through implementation of an auxiliary particle filter such as that considered by Pitt et al. (2012). Providing practical advice on how to tune the random walk Metropolis step used inside the delayed acceptance scheme also remains of interest.

References


A Appendices

Recall that $x = \{ x_t | 1 \leq t \leq T \}$ denotes values of the latent MJP and $y = \{ y_t | t = 1, 2, \ldots, T \}$ denotes the collection of (noisy) observations on the MJP at discrete times. In addition, we define $x_t = \{ x_s | t-1 < s \leq t \}$ and $y_t = \{ y_s | s = 1, 2, \ldots, t \}$.

A.1 PMMH scheme

The PMMH scheme has the following algorithmic form.

1. Initialisation, $i = 0$,
   (a) set $c^{(0)}$ arbitrarily and
   (b) run an SMC scheme targeting $p(x|y, c^{(0)})$, and let $\hat{p}(y|c^{(0)})$ denote the marginal likelihood estimate

2. For iteration $i \geq 1$,
   (a) sample $c^{*} \sim q(.|c^{(i-1)})$,
   (b) run an SMC scheme targeting $p(x|y, c^{*})$, and let $\hat{p}(y|c^{*})$ denote the marginal likelihood estimate,
   (c) with probability $\min\{1, A\}$ where
      $$A = \frac{\hat{p}(y|c^{*})p(c^{*})}{\hat{p}(y|c^{(i-1)})p(c^{(i-1)})} \times \frac{q(c^{(i-1)}|c^{*})}{q(c^{*}|c^{(i-1)})}$$
      accept a move to $c^{*}$ otherwise store the current values

Note that the PMMH scheme can be used to sample the joint posterior $p(c, x|y)$. Essentially, a proposal mechanism of the form $q(c^{*}|c)\hat{p}(x^{*}|y, c^{*})$, where $\hat{p}(x^{*}|y, c^{*})$ is an SMC approximation of $p(x^{*}|y, c^{*})$, is used. The resulting MH acceptance ratio is as above. Full details of the PMMH scheme including a proof establishing that the method leaves the target $p(c, x|y)$ invariant can be found in Andrieu et al. (2010).
A.2 SMC scheme

A sequential Monte Carlo estimate of the marginal likelihood $p(y|c)$ under the MJP can be constructed using (for example) the bootstrap filter of Gordon et al. (1993). Algorithmically, we perform the following sequence of steps.

1. Initialisation.
   (a) Generate a sample of size $N$, $\{x_1^1, \ldots, x_1^N\}$ from the initial density $p(x_1)$.
   (b) Assign each $x_1^i$ a (normalised) weight given by
   \[ w_1^i = \frac{w_1^i}{\sum_{i=1}^N w_1^i}, \text{ where } w_1^i = p(y_1|x_1^i, c). \]
   (c) Construct and store the currently available estimate of marginal likelihood,
   \[ \hat{p}(y_1|c) = \frac{1}{N} \sum_{i=1}^N w_1^i. \]
   (d) Resample $N$ times with replacement from $\{x_1^1, \ldots, x_1^N\}$ with probabilities given by $\{w_1^1, \ldots, w_1^N\}$.

2. For times $t = 1, 2, \ldots, T - 1$,
   (a) For $i = 1, \ldots, N$: draw $X_{t+1}^i \sim p(x_{t+1}|x_t^i, c)$ using the Gillespie algorithm.
   (b) Assign each $x_{t+1}^i$ a (normalised) weight given by
   \[ w_{t+1}^i = \frac{w_{t+1}^i}{\sum_{i=1}^N w_{t+1}^i}, \text{ where } w_{t+1}^i = p(y_{t+1}|x_{t+1}^i, c). \]
   (c) Construct and store the currently available estimate of marginal likelihood,
   \[ \hat{p}(y_{t+1}|c) = \hat{p}(y_t|c)\hat{p}(y_{t+1}|y_t, c) \]
   \[ = \hat{p}(y_t|c) \frac{1}{N} \sum_{i=1}^N w_{t+1}^i. \]
   (d) Resample $N$ times with replacement from $\{x_{t+1}^1, \ldots, x_{t+1}^N\}$ with probabilities given by $\{w_{t+1}^1, \ldots, w_{t+1}^N\}$. 

30
A.3 Marginal likelihood under the linear noise approximation

Assume an observation regime of the form

\[ Y_t = G'X_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \Sigma) \]

where \( G \) is a constant matrix of dimension \( u \times p \) and \( \varepsilon_t \) is a length-\( p \) Gaussian random vector.

Now suppose that \( X_1 \sim N(a, C) \) \textit{a priori}. The marginal likelihood under the LNA, \( p_a(y|c) \) can be obtained as follows.

1. Initialisation. Compute

\[ p_a(y_1|c) = \phi(y_1; G'a, G'CG + \Sigma) \]

where \( \phi(\cdot; a, C) \) denotes the Gaussian density with mean vector \( a \) and variance matrix \( C \). The posterior at time \( t = 1 \) is therefore \( X_1|y_1 \sim N(a_1, C_1) \) where

\[ a_1 = a + CG(G'CG + \Sigma)^{-1}(y_1 - G'a) \]

\[ C_1 = C - CG(G'CG + \Sigma)^{-1}G'C. \]

2. For times \( t = 1, 2, \ldots, T - 1 \),

(a) Prior at \( t + 1 \). Initialise the LNA with \( z_t = a_t, m_t = 0 \) and \( V_t = C_t \). Note that (8) implies \( m_s = 0 \) for all \( s > t \). Therefore, integrate the ODEs (5) and (9) forward to \( t + 1 \) to obtain \( z_{t+1} \) and \( V_{t+1} \). Hence

\[ X_{t+1}|y_t \sim N(z_{t+1}, V_{t+1}). \]

(b) One step forecast. Using the observation equation, we have that

\[ Y_{t+1}|y_t \sim N(G'z_{t+1}, G'V_{t+1}G + \Sigma). \]

Compute

\[ p_a(y_{t+1}|c) = p_a(y_t|c)p_a(y_{t+1}|y_t, c) \]

\[ = p_a(y_t|c) \phi(y_{t+1}; G'z_{t+1}, G'V_{t+1}G + \Sigma). \]
(c) Posterior at $t + 1$. Combining the distributions in (a) and (b) gives the joint distribution of $X_{t+1}$ and $Y_{t+1}$ (conditional on $y_t$ and $c$) as

$$
\begin{pmatrix}
X_{t+1} \\
Y_{t+1}
\end{pmatrix}
\sim
\mathcal{N}
\left\{
\begin{pmatrix}
z_{t+1} \\
G'z_{t+1}
\end{pmatrix},
\begin{pmatrix}
V_{t+1} & V_{t+1}G' \\
G'V_{t+1} & G'V_{t+1}G + \Sigma
\end{pmatrix}
\right\}
$$

and therefore $X_{t+1}|y_{t+1} \sim \mathcal{N}(a_{t+1}, C_{t+1})$ where

$$
a_{t+1} = z_{t+1} + V_{t+1}G(G'V_{t+1}G + \Sigma)^{-1}(y_{t+1} - G'z_{t+1})
$$

$$
C_{t+1} = V_{t+1} - V_{t+1}G(G'V_{t+1}G + \Sigma)^{-1}G'V_{t+1}.
$$